Confidential Audit Report Generated by RidgeBot™

JP EXT Site

Mar 1, 2023 at 19:38

Agreement

CONFIDENTIALITY

This document contains proprietary and confidential information of a highly sensitive nature. Reproduction or distribution without the express written permission of Ridge Security Technology Corp. or the Client named above is strictly prohibited. This document should be marked “CONFIDENTIAL” and therefore we suggest that this document be disseminated on a 'need to know' basis.

DISCLAIMERS

The information presented in this document is provided as is and without warranty. Vulnerability assessments are a 'point in time' analysis and as such it is possible that something in the environment could have changed since the tests reflected in this report were run. Also, it is possible that new vulnerabilities may have been discovered since the tests were run. For this reason, this report should be considered a guide, not a 100% representation of the risk threatening your systems, networks and applications. This report cannot and does not protect against personal or business loss as the result of use of the applications or systems described. By using this information, you agree that Ridge Security shall be held harmless in any event.

9 Critical Successful Exploits

Report Generated by RidgeBot™

JP EXT Site

QUICKLINKS

- Executive Summary
- Configuration at a Glance
- Asset Details
- Website Fingerprints
- Host Open Ports
- Exploit Details
- Vulnerability Details
- Attack Surface Details

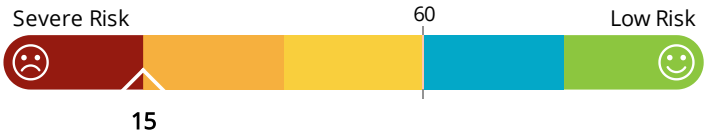
Executive Summary

System Version: **V4.2.0-20221130** Plugin Library Version: **V4.16.0**

TASK NAME	START TIME	END TIME	TOTAL TIME	STATUS
JP EXT Site	Mar 1, 2023 at 19:38	Mar 1, 2023 at 20:08	0 hours and 30 minutes	Success

Total Health Score

Policy: Minimum Score 60

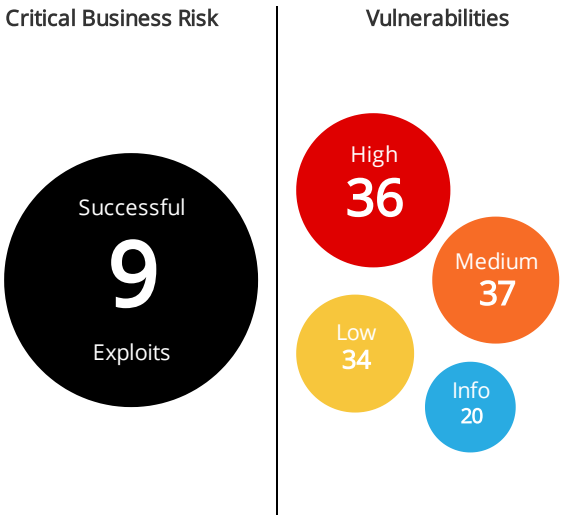


In this task, we have tested 1 IPs and 1 web servers, the Total Health Score of the target system is 15, this score is based on 100 scale. It is a comprehensive evaluation based on multiple factors such as percentage of vulnerability, attack surface, encrypted traffic etc.This test system is considered as in a "Risky"(Risky<60; 60<=normal<85; good>=85) condition with the score of 15. The vulnerability found on each asset can be found in "Asset Detail".

The platform successfully performed 9 exploits. These 9 exploited risks are critical and require immediate attention. It means a real hacker can easily achieve the same result. In the "Exploit Details", we provided information on how it attacked - path, techniques and actions etc for security team to replicate and fix the issue.

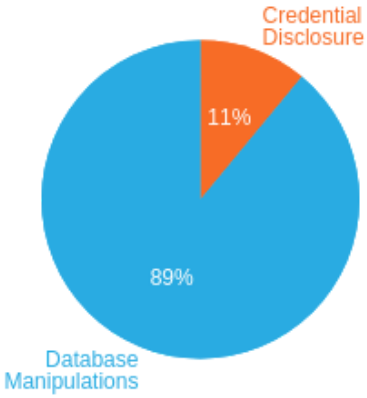
Among 9 exploits, 11.0% credential disclosure. 89.0% database manipulations.

Risk Weighted Assessment



Total number of targets :	1
Number of active assets :	1
Number of active Domains :	1
Number of attack surface(s) :	52

Exploit Results by Type



Understanding the health and risk charts

In addition, the platform found 36 high vulnerabilities, 37 medium and 34 low vulnerabilities. These vulnerabilities are possible risks, it might be exploitable, however it may take bigger risk or larger efforts for a hacker. It shall be attended to achieve a comprehensive defense system. Please refer to the "Vulnerability Details" for more information and remediation suggestion.

Penetration Test Action Distribution

The Penetration Test Action Distribution chart

Breakdown of total jobs spent within each of the three core functions for the total count of jobs.

Business Risk Summary

INDEX	RISK TYPE	RELATED VULNERABILITY	TARGET	DETAILS
1	Credential Disclosure	Backend Weak Password	http://testphp.vulnweb.com/login.php	→
2	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/search.php?test=query	→
3	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/search.php?test=query	→
4	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/listproducts.php?cat=1	→
5	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/artists.php?artist=1	→
6	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/product.php?pic=1	→
7	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/secured/newuser.php	→
8	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/userinfo.php	→
9	Database Manipulations	SQL Injection	http://testphp.vulnweb.com/listproducts.php?artist=1	→

Configuration at a Glance

SYSTEM TEMPLATE	CUSTOMIZED TEMPLATE	PLUGINS SELECTED	SCAN TYPE	SCRAPING MODE	STEALTH LEVEL
Website Penetration	N/A	4161	Web application	Crawling	Normal

OS TYPE	SEVERITY	RISK
WINDOWS (618) OTHER (4032) LINUX (3943)	HIGH (1127) MEDIUM (1133) LOW (1768) INFO (133)	IMPACTFUL (442) LOW IMPACT (3719)

Asset Details

TARGET	OS TYPE	EXPLOITED	HIGH	MEDIUM	LOW
44.228.249.3	Ubuntu	0	0	0	0
testphp.vulnweb.com		0	0	0	0

SITE	IP/DOMAIN	EXPLOITED	HIGH	MEDIUM	LOW
http://testphp.vulnweb.com/	testphp.vulnweb.com	9	36	37	34

Website Fingerprints

INDEX	SITE	CMS	LANGUAGE	FRAMEWORK	WAF/CDN TYPE
1	http://testphp.vulnweb.com/	Nginx 1.19.0	PHP 5.6.40	-	-

Exploit Details

9 Critical Business Risks

1 web credentials obtained via Backend Weak Password vul Critical

Type	Rank	CVSS Score
Credential Disclosure	Critical	8.6

CVSS Vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L

Description:

When the application permits weak passwords for users or admins, hacker can brute-forced into backend and gain the exposure of private data.

Solution:

1.Implement multi-factor authentication to prevent automated attacks. 2. Encourage (or force) the user to adopt a good password policy. 3. Limit failed logins. 4. Use efficient algorithm hash. When choosing an algorithm, consider the max password length. 5. Test the session timeout system and make sure the session token is invalidated after logout.

Reference:

https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication

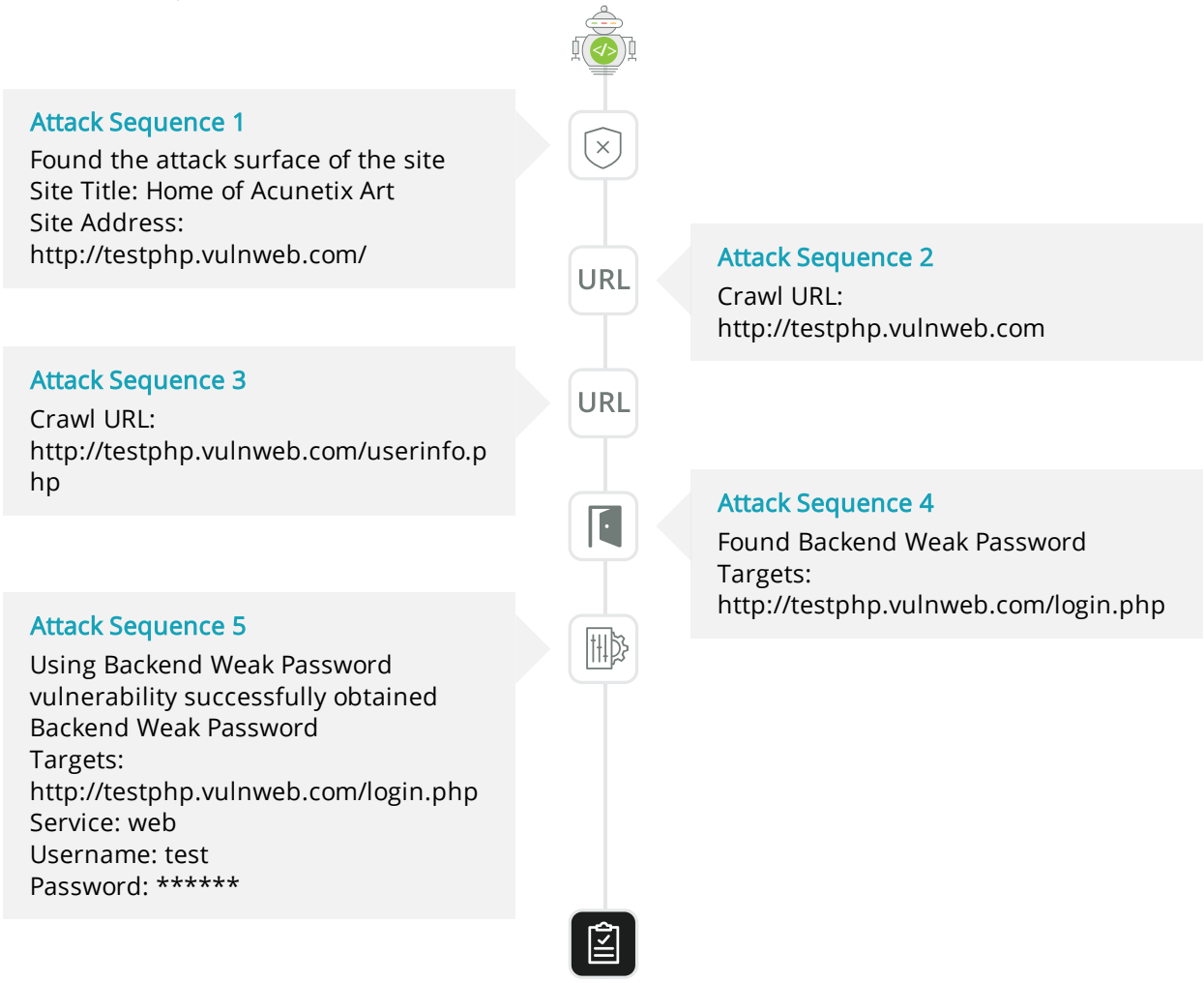
https://geekflare.com/web-backend-security-risk/

Detail(Total 1):

#1/1 Vulnerability Target: http://testphp.vulnweb.com/login.php

Current User: test
Service: web
Port:
Username: test
Password: ******

Kill Chain Analysis



2 - 9 database information disclosed via SQL Injection vul

Critical

Type	Rank	CVSS Score
Database Manipulations	Critical	8.6

CVSS Vector:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L

Description:

An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server, or others. Hackers may use it to gain unauthorized access to your sensitive data: customer information, personal data, trade secrets, intellectual property, and more.

Solution:

The only sure way to prevent SQL Injection attacks is input validation and parameterized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.

Reference:

https://www.owasp.org/index.php/Blind_SQL_Injection

https://en.wikipedia.org/wiki/SQL_injection

http://www.websec.ca/kb/sql_injection

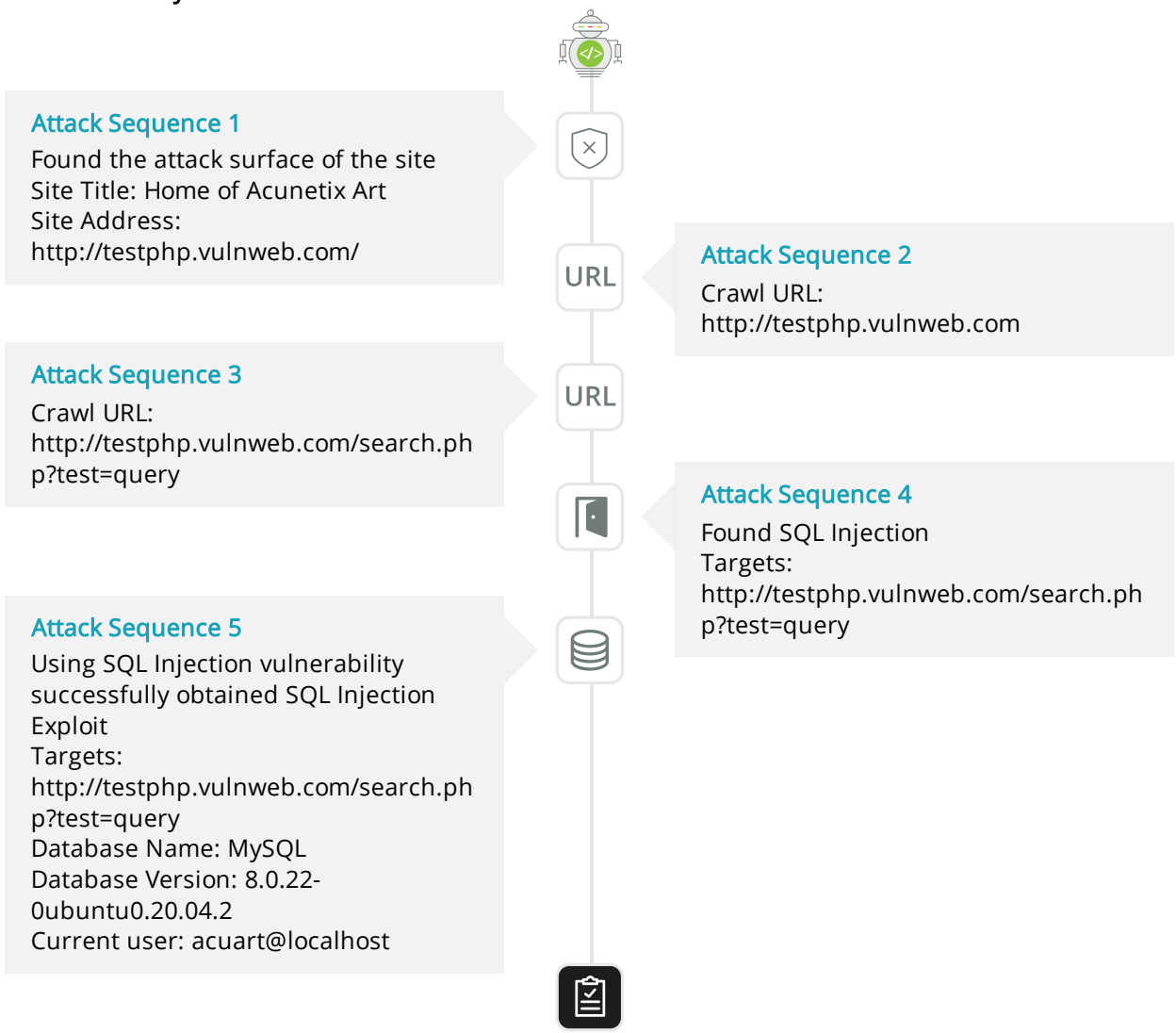
https://www.owasp.org/index.php/SQL_Injection

Detail(Total 8):

#1/8 Vulnerability Target: http://testphp.vulnweb.com/search.php?test=query

Current User: acuart@localhost
Database Count: 2
Table Count: 19

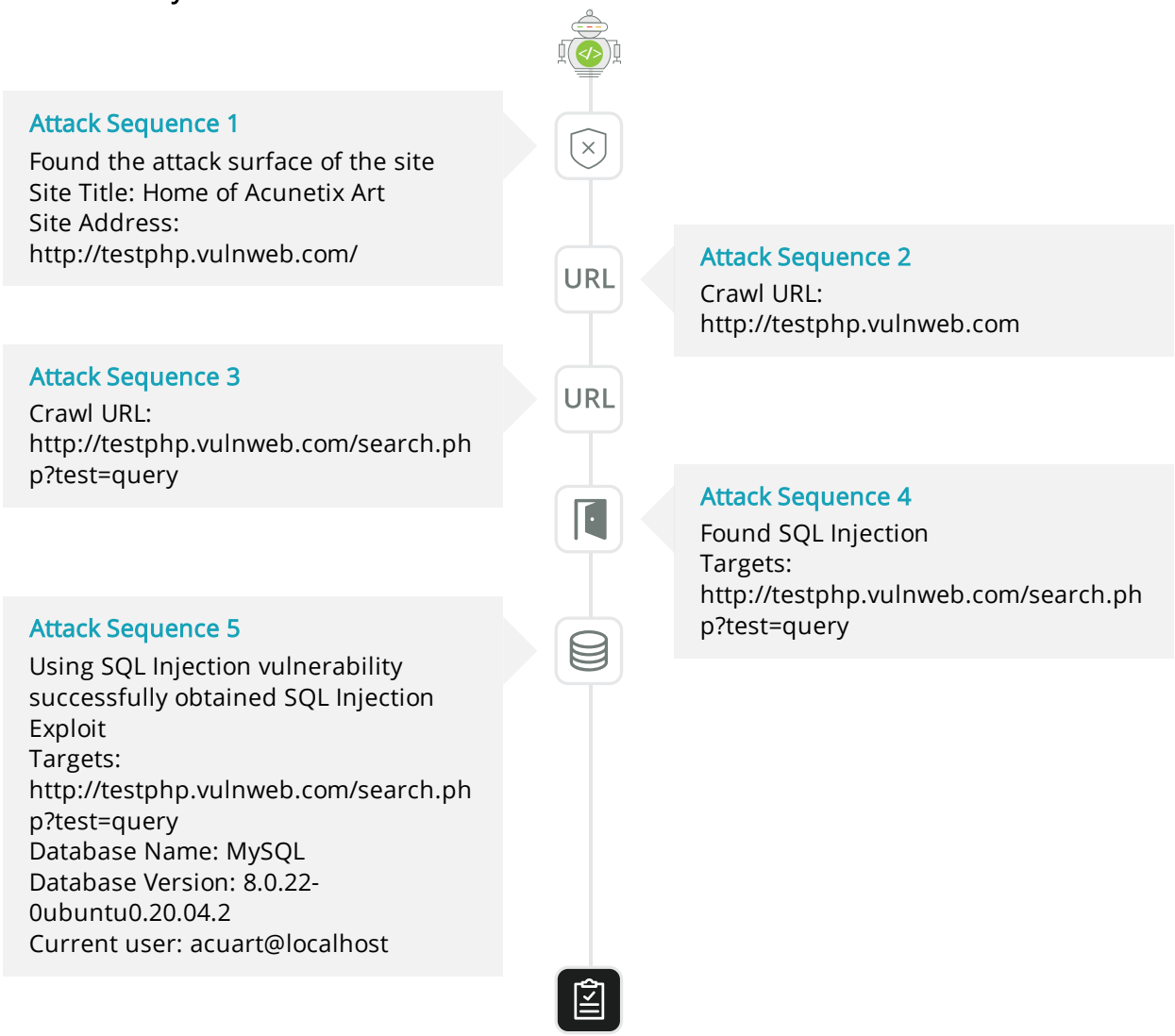
Kill Chain Analysis



#2/8 Vulnerability Target: http://testphp.vulnweb.com/search.php?test=query

Current User: acuart@localhost
Database Count: 2
Table Count: 19

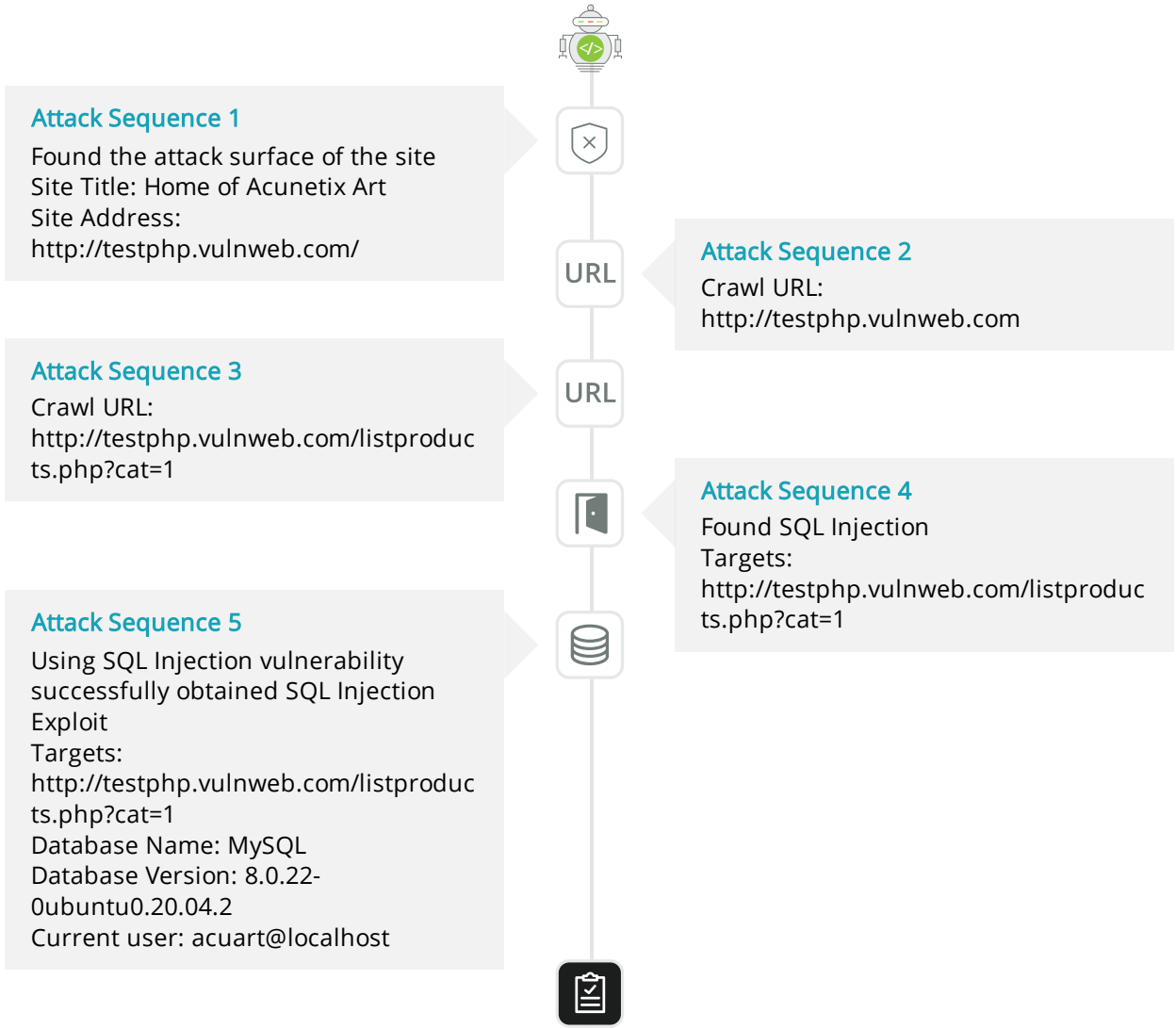
Kill Chain Analysis



#3/8 Vulnerability Target: http://testphp.vulnweb.com/listproducts.php?cat=1

Current User: acuart@localhost
Database Count: 2
Table Count: 87

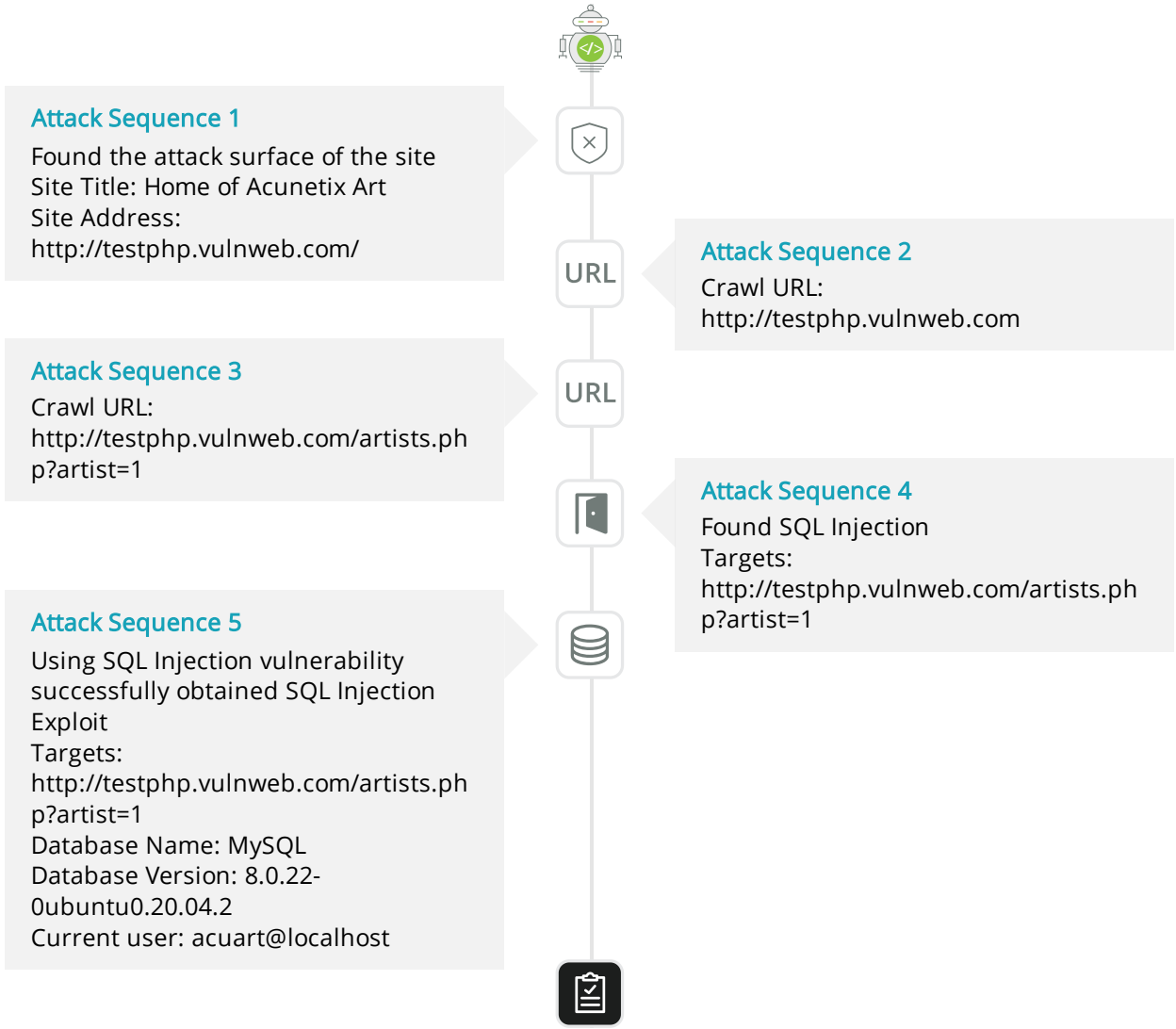
Kill Chain Analysis



#4/8 Vulnerability Target: http://testphp.vulnweb.com/artists.php?artist=1

Current User: acuart@localhost
Database Count: 2
Table Count: 87

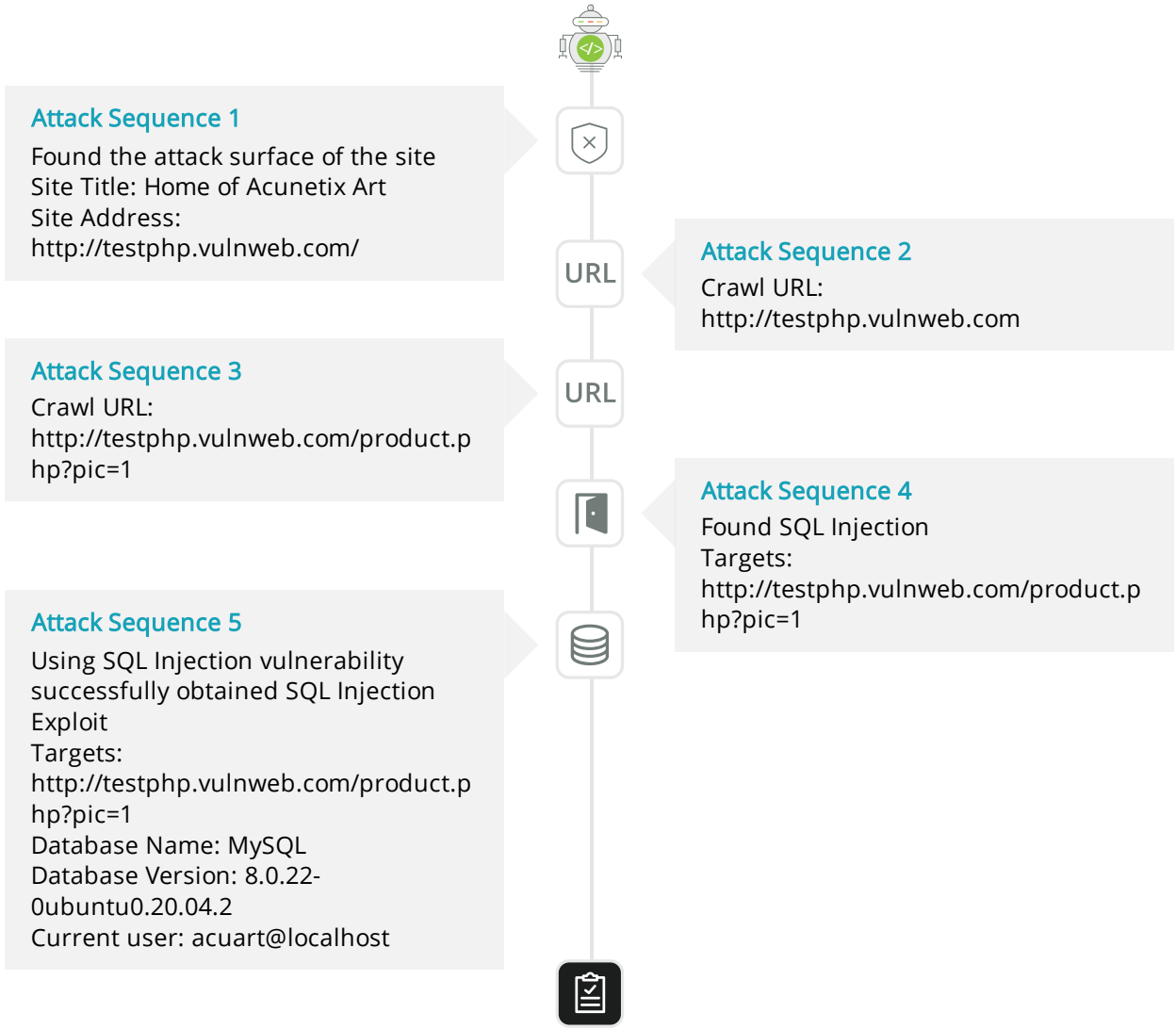
Kill Chain Analysis



#5/8 Vulnerability Target: http://testphp.vulnweb.com/product.php?pic=1

Current User: acuart@localhost
Database Count: 2
Table Count: 87

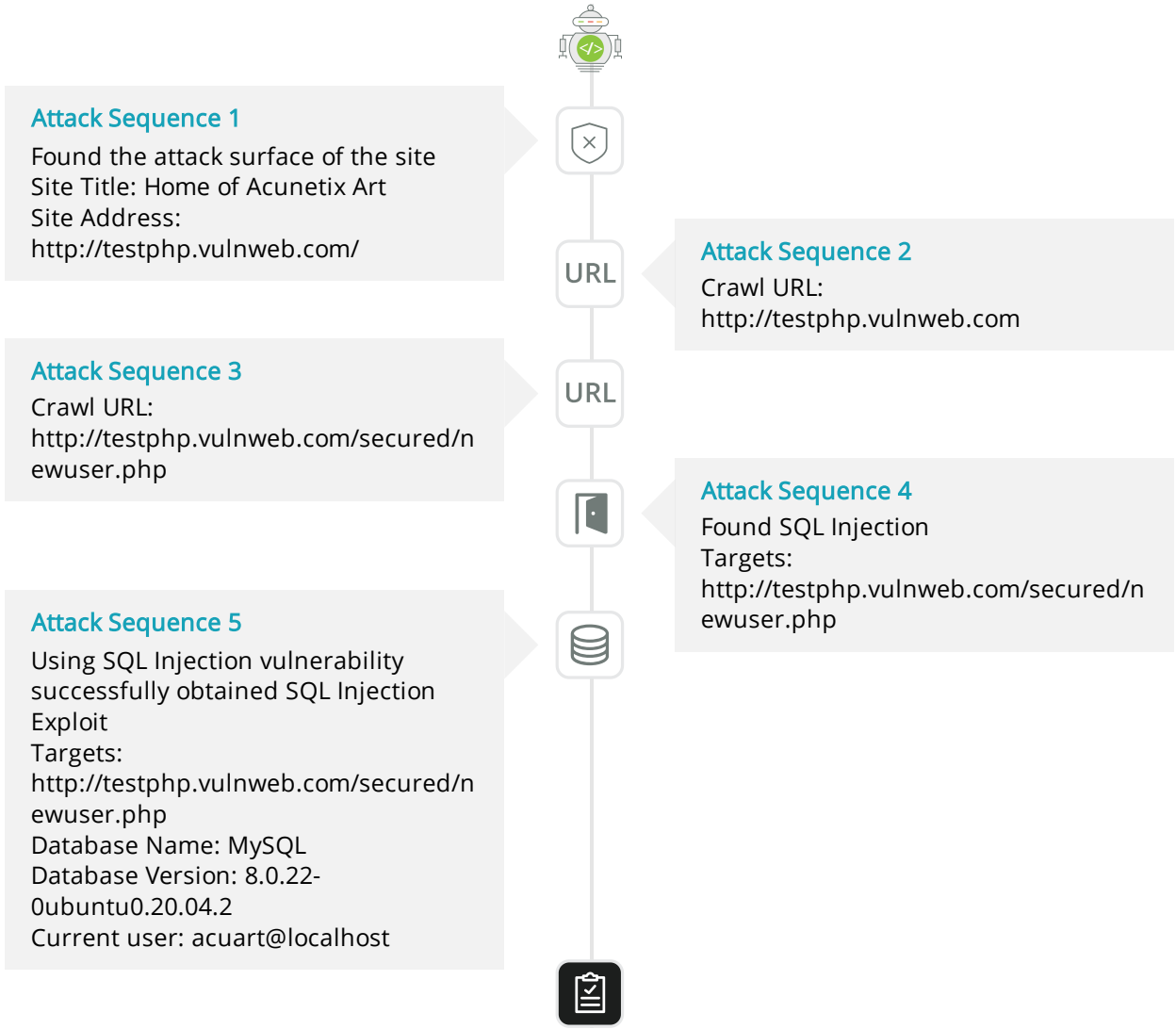
Kill Chain Analysis



#6/8 Vulnerability Target: http://testphp.vulnweb.com/secured/newuser.php

Current User: acuart@localhost
Database Count: 2
Table Count: 87

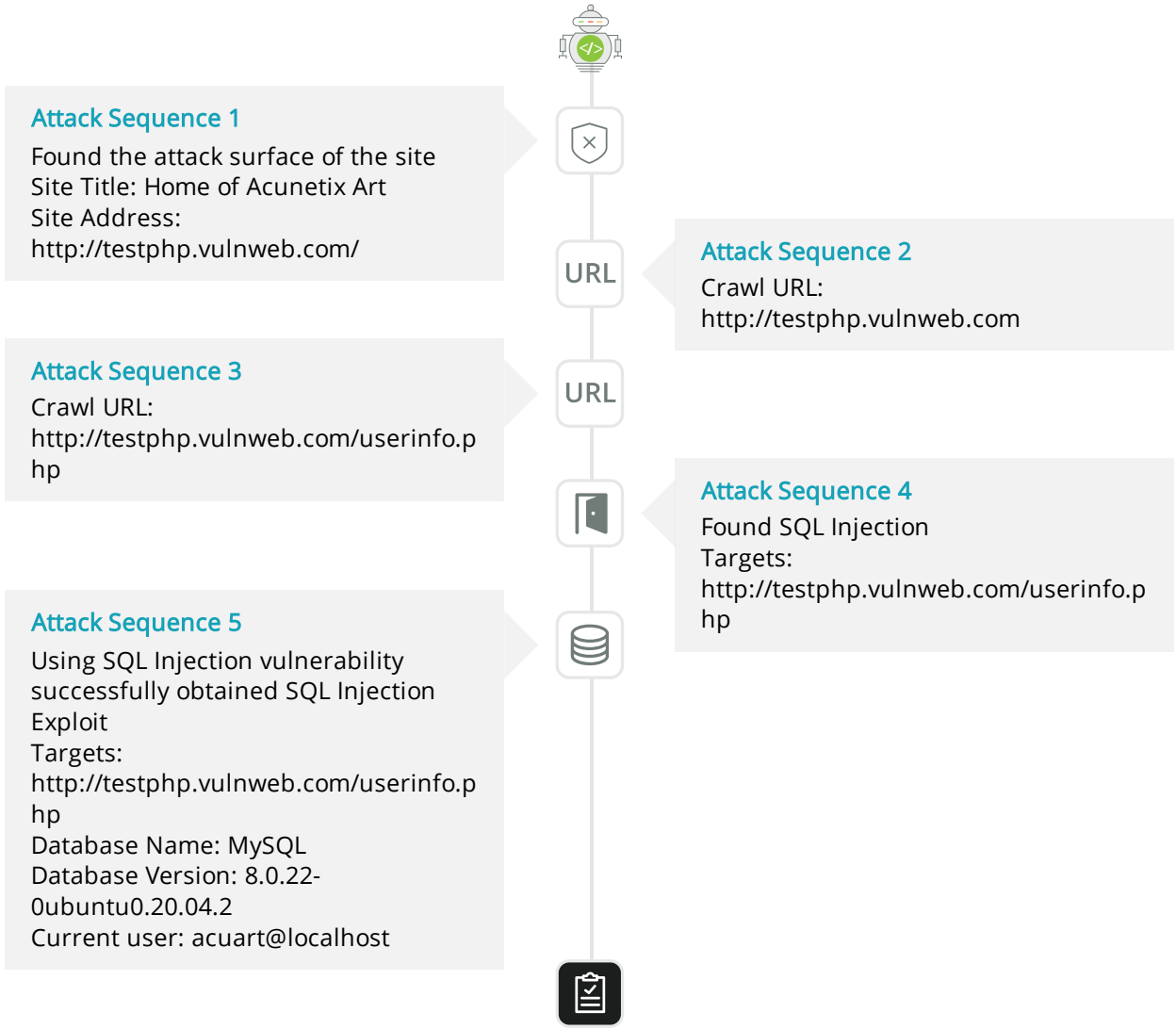
Kill Chain Analysis



#7/8 Vulnerability Target: http://testphp.vulnweb.com/userinfo.php

Current User: acuart@localhost
Database Count: 2
Table Count: 87

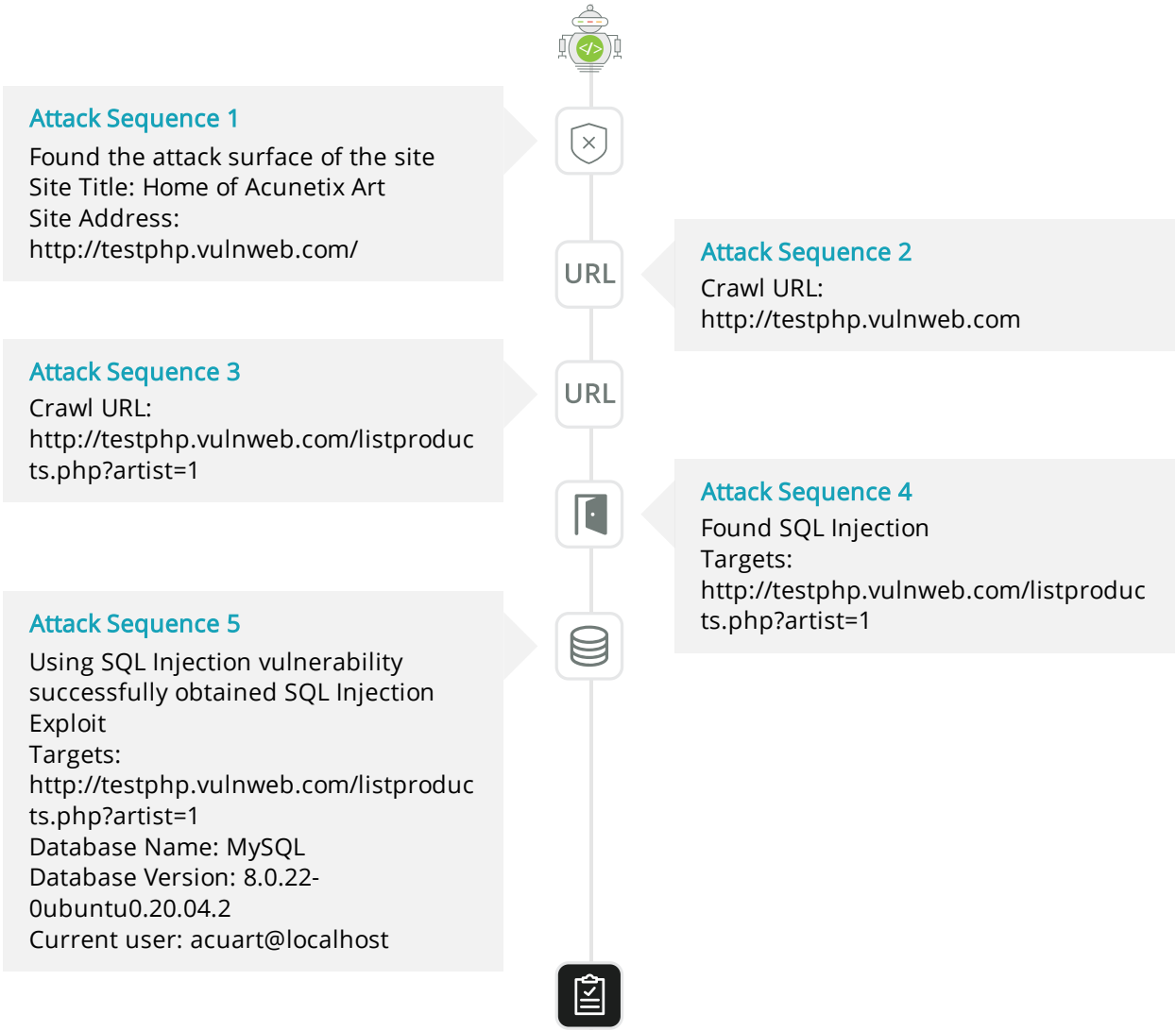
Kill Chain Analysis



#8/8 Vulnerability Target: http://testphp.vulnweb.com/listproducts.php?artist=1

Current User: acuart@localhost
Database Count: 2
Table Count: 87

Kill Chain Analysis



Vulnerability Details

36 High Vulnerabilities

1 Relative Path Traversal

Description:

Due to business requirements, some websites often need to provide file view or file download functions. However, if there are no restrictions on the files users view or download, malicious users can view or download any sensitive files, which is a file view and download vulnerability * there is a function to read files * the path to read files is controllable and unchecked or not strictly verified by users * Output the file content download any file of the server, such as script code, service, system configuration file and other available codes for further code audit and more exploitable vulnerabilities

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/showimage.php?file=showimage.php
	Vulnerability details	Target http://testphp.vulnweb.com/showimage.php?file=showimage.php has Relative Path Traversal vulnerability

	Parameter names	file
	Payload	showimage.php

References

REFERENCES
https://owasp.org/www-community/attacks/Path_Traversal
https://cwe.mitre.org/data/definitions/23.html

Vulnerability Solution:

1.Strictly control the input parameters of users, and filter the response of functions affected by parameters

Public Poc:

Not Available

2 - 9 SQL Injection

Description:

An SQL Injection vulnerability may affect any website or web application that uses an SQL database such as MySQL, Oracle, SQL Server, or others. Hackers may use it to gain unauthorized access to your sensitive data: customer information, personal data, trade secrets, intellectual property, and more.

Affected Nodes:

1/8	Target	http://testphp.vulnweb.com/artists.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php?artist=1 has SQL Injection vulnerability
	Parameter names	artist
	Payload	%E6%5C'"\(
2/8	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has SQL Injection vulnerability
	Parameter names	test
	Payload	'
3/8	Target	http://testphp.vulnweb.com/listproducts.php?cat=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?cat=1 has SQL Injection vulnerability
	Parameter names	cat
	Payload	%E6%5C'"\(
4/8	Target	http://testphp.vulnweb.com/product.php?pic=1
	Vulnerability details	Target http://testphp.vulnweb.com/product.php?pic=1 has SQL Injection vulnerability

	Parameter names	pic
	Payload	%E6%5C'"\(
5/8	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has SQL Injection vulnerability
	Parameter names	uuname
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
6/8	Target	http://testphp.vulnweb.com/listproducts.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?artist=1 has SQL Injection vulnerability
	Parameter names	artist
	Payload	%E6%5C'"\(
7/8	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has SQL Injection vulnerability
	Parameter names	test
	Payload	goButton=go&searchFor=data
8/8	Target	http://testphp.vulnweb.com/userinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/userinfo.php has SQL Injection vulnerability
	Parameter names	uname
	Payload	pass=g00dPa%24%24w0rD&uname=data

References

REFERENCES
https://www.owasp.org/index.php/Blind_SQL_Injection
https://en.wikipedia.org/wiki/SQL_injection
http://www.websec.ca/kb/sql_injection
https://www.owasp.org/index.php/SQL_Injection

Vulnerability Solution:

The only sure way to prevent SQL Injection attacks is input validation and parameterized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.

Public Poc:

Not Available

10 - 13 Blind Cross-Site Scripting

Description:

Blind Cross-site Scripting is a form of persistent XSS. It generally occurs when the attacker's payload saved on the server and reflected back to the victim from the backend application. For example in feedback forms, an attacker can submit the malicious payload using the form, and once the backend user/admin of the application will open the attacker's submitted form via the backend application, the attacker's payload will get executed.

Affected Nodes:

1/4	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Blind Cross-Site Scripting vulnerability
	Parameter names	text
	Payload	name=anonymous+user&submit=add+message&text=data
2/4	Target	http://testphp.vulnweb.com/comment.php
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php has Blind Cross-Site Scripting vulnerability
	Parameter names	name
	Payload	Submit=Submit&comment=data&name=%3Cyour+name+here%3E&phpaction=echo+%24_POST%5Bcomment%5D%3B
3/4	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has Blind Cross-Site Scripting vulnerability
	Parameter names	searchFor
	Payload	goButton=go&searchFor=data
4/4	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Blind Cross-Site Scripting vulnerability
	Parameter names	uphone
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data

References

REFERENCES
https://owasp.org/www-community/attacks/xss/
https://owasp.org/www-community/xss-filter-evasion-cheatsheet

Vulnerability Solution:

Edit the source code to ensure that input is properly validated. Where is possible, it is recommended to make a list of accepted filenames and restrict the input to that list.

Public Poc:

Not Available

14 - 18 XSS via Remote File Inclusion

Description:

This script is possibly vulnerable to remote XSS inclusion. The path to the XSS file can be controlled by the attacker. Therefore, it's possible to include malicious XSS files.

Affected Nodes:

1/5	Target	http://testphp.vulnweb.com/listproducts.php?cat=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?cat=1 has XSS via Remote File Inclusion vulnerability
	Parameter names	cat
	Payload	http://66.220.31.40/p/body?content=<script>prompt(87394581)</script>
2/5	Target	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12 has XSS via Remote File Inclusion vulnerability
	Parameter names	pp
	Payload	aaaa%2F=data
3/5	Target	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12 has XSS via Remote File Inclusion vulnerability
	Parameter names	pp
	Payload	http://66.220.31.40/p/body?content=<script>prompt(87394581)</script>
4/5	Target	http://testphp.vulnweb.com/listproducts.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?artist=1 has XSS via Remote File Inclusion vulnerability
	Parameter names	artist
	Payload	http://66.220.31.40/p/body?content=<script>prompt(87394581)</script>
5/5	Target	http://testphp.vulnweb.com/hpp/?pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/?pp=12 has XSS via Remote File Inclusion vulnerability
	Parameter names	pp
	Payload	http://66.220.31.40/p/body?content=<script>prompt(87394581)</script>

References

REFERENCES
https://owasp.org/www-community/xss-filter-evasion-cheatsheet

Vulnerability Solution:

Edit the source code to ensure that input is properly validated. Where is possible, it is recommended to make a list of accepted filenames and restrict the input to that list.

Public Poc:

Not Available

19 - 34 Cross-Site Scripting

Description:

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

Affected Nodes:

1/16	Target	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12 has Cross-Site Scripting vulnerability
	Parameter names	p
	Payload	'"()%26%25<acx><ScRiPt%20>cfljM(cfljM)</ScRiPt>
2/16	Target	http://testphp.vulnweb.com/listproducts.php?cat=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?cat=1 has Cross-Site Scripting vulnerability
	Parameter names	cat
	Payload	'"()%26%25<acx><ScRiPt%20>xsitq(xsitq)</ScRiPt>
3/16	Target	http://testphp.vulnweb.com/showimage.php?file=
	Vulnerability details	Target http://testphp.vulnweb.com/showimage.php?file= has Cross-Site Scripting vulnerability
	Parameter names	file
	Payload	'"()%26%25<acx><ScRiPt%20>juYFT(juYFT)</ScRiPt>
4/16	Target	http://testphp.vulnweb.com/listproducts.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?artist=1 has Cross-Site Scripting vulnerability
	Parameter names	artist
	Payload	'"()%26%25<acx><ScRiPt%20>ByTLv(ByTLv)</ScRiPt>

5/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	uemail
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
6/16	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Cross-Site Scripting vulnerability
	Parameter names	name
	Payload	name=anonymous+user&submit=add+message&text=data
7/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	uphone
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
8/16	Target	http://testphp.vulnweb.com/comment.php
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php has Cross-Site Scripting vulnerability
	Parameter names	name
	Payload	Submit=Submit&comment=data&name=%3Cyour+name+here%3E&phpaction=echo+%24_POST%5Bcomment%5D%3B
9/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	urname
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
10/16	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Cross-Site Scripting vulnerability
	Parameter names	text
	Payload	name=anonymous+user&submit=add+message&text=data

11/16	Target	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12 has Cross-Site Scripting vulnerability
	Parameter names	pp
	Payload	'"()%26%25<acx><ScRiPt%20>cfljM(cfljM)</ScRiPt>
12/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	ucc
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
13/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	uuname
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
14/16	Target	http://testphp.vulnweb.com/hpp/?pp=12
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/?pp=12 has Cross-Site Scripting vulnerability
	Parameter names	pp
	Payload	'"()%26%25<acx><ScRiPt%20>YVKgM(YVKgM)</ScRiPt>
15/16	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has Cross-Site Scripting vulnerability
	Parameter names	searchFor
	Payload	goButton=go&searchFor=data
16/16	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Cross-Site Scripting vulnerability
	Parameter names	uaddress
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data

References

REFERENCES
https://owasp.org/www-community/attacks/xss/
https://www.owasp.org/index.php/Reflected_DOM_Injection
https://www.owasp.org/index.php/Testing_for_Stored_Cross_site_scripting_(OTG-INPVAL-002)
https://www.owasp.org/index.php/Testing_for_Reflected_Cross_site_scripting_(OTG-INPVAL-001)
https://portswigger.net/web-security/cross-site-scripting
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

Vulnerability Solution:

1. Filter input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input.

2. Encode data on output. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.

3. Use appropriate response headers. To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

4. Content Security Policy. As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

Public Poc:

Not Available

35 Backend Weak Password

Description:

When the application permits weak passwords for users or admins, hacker can brute-forced into backend and gain the exposure of private data.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has Backend Weak Password vulnerability
	Parameter names	Null
	Payload	

References

REFERENCES
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication
https://geekflare.com/web-backend-security-risk/

Vulnerability Solution:

1.Implement multi-factor authentication to prevent automated attacks.

2. Encourage (or force) the user to adopt a good password policy.

3. Limit failed logins.

4. Use efficient algorithm hash. When choosing an algorithm, consider the max password length.

5. Test the session timeout system and make sure the session token is invalidated after logout.

Public Poc:

Not Available

36 PHP phpinfo Page Information Disclosure

Description:

In PHP environment, variables and other information can be obtained through the phpinfo page. The disclosure of these information combined with some other vulnerabilities can exposed the system to infiltrate and attack.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/secured/phpinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/phpinfo.php has PHP phpinfo Page Information Disclosure vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/information-disclosure-phpinfo/

Vulnerability Solution:

1. Delete phpinfo file

Public Poc:

Not Available

37 Medium Vulnerabilities

1 - 22 Cross Site Request Forgery (CSRF)

Description:

Cross-site request forgery, also known as one-click attack or session riding and abbreviated as CSRF or XSRF, is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the web application trusts. There are many ways in which a malicious website can transmit such commands; specially-crafted image tags, hidden forms,
and JavaScript XMLHttpRequests, for example, can all work without the user's interaction or even knowledge. Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser. In a CSRF attack, an innocent end user is tricked by an attacker into submitting a web request that they did not intend. This may cause actions to be performed on the website that can include inadvertent client or server data leakage, change of session state, or manipulation of an end user's account.

Affected Nodes:

1/22	Target	http://testphp.vulnweb.com/listproducts.php?cat=1

	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?cat=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
2/22	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
3/22	Target	http://testphp.vulnweb.com/product.php?pic=1
	Vulnerability details	Target http://testphp.vulnweb.com/product.php?pic=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
4/22	Target	http://testphp.vulnweb.com/artists.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php?artist=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
5/22	Target	http://testphp.vulnweb.com/disclaimer.php
	Vulnerability details	Target http://testphp.vulnweb.com/disclaimer.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
6/22	Target	http://testphp.vulnweb.com/categories.php
	Vulnerability details	Target http://testphp.vulnweb.com/categories.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
7/22	Target	http://testphp.vulnweb.com/userinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/userinfo.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	

8/22	Target	http://testphp.vulnweb.com/cart.php
	Vulnerability details	Target http://testphp.vulnweb.com/cart.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
9/22	Target	http://testphp.vulnweb.com/userinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/userinfo.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	pass=g00dPa%24%24w0rD&uname=data
10/22	Target	http://testphp.vulnweb.com/comment.php?pid=1
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php?pid=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
11/22	Target	http://testphp.vulnweb.com/signup.php
	Vulnerability details	Target http://testphp.vulnweb.com/signup.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
12/22	Target	http://testphp.vulnweb.com/artists.php
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
13/22	Target	http://testphp.vulnweb.com/cart.php
	Vulnerability details	Target http://testphp.vulnweb.com/cart.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	addcart=1&price=500
14/22	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null

	Payload	goButton=go&searchFor=data
15/22	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	name=anonymous+user&submit=add+message&text=data
16/22	Target	http://testphp.vulnweb.com/listproducts.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?artist=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
17/22	Target	http://testphp.vulnweb.com/comment.php
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
18/22	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
19/22	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
20/22	Target	http://testphp.vulnweb.com/comment.php?aid=1
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php?aid=1 has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	
21/22	Target	http://testphp.vulnweb.com/userinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/userinfo.php has Cross Site Request Forgery (CSRF) vulnerability

	Parameter names	Null
	Payload	
22/22	Target	http://testphp.vulnweb.com/index.php
	Vulnerability details	Target http://testphp.vulnweb.com/index.php has Cross Site Request Forgery (CSRF) vulnerability
	Parameter names	Null
	Payload	

References

REFERENCES
https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

Vulnerability Solution:

Most CSRF prevention techniques work by embedding additional authentication data into requests that allows the web application to detect requests from unauthorized locations. Following are example:

1. Synchronize token pattern;

2. Cookie-to-header token;

3. Double Submit Cookie;

4. SameSite cookie attribute;

5. Client-side safeguards.

Public Poc:

Not Available

23 - 24 Test File Disclosure

Description:

Information disclosure refers to a website or a file which reveals sensitive information unintentionally to its users. Sensitive information includes password, key, session ID, license, personal data such as text messages, authorization credentials, personal identifiable information(name, address, telephone number and etc), program files, configuration files, log files, backup files and database

Affected Nodes:

1/2	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html/test.html?NOMBuJXovDGcZUNkJePHaMKyRxNNSUhJ
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html/test.html?NOMBuJXovDGcZUNkJePHaMKyRxNNSUhJ has Test File Disclosure vulnerability
	Parameter names	test.html
	Payload	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html/test.html?NOMBuJXovDGcZUNkJePHaMKyRxNNSUhJ
2/2	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html/test.html?cgqGWTRCgfUDbZiChSoHNLClVLnyZwiQ

	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html/test.html?cgqGWTRCgfUDbZiChSoHNLClVLnyZwiQ has Test File Disclosure vulnerability
	Parameter names	test.html
	Payload	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html/test.html?cgqGWTRCgfUDbZiChSoHNLClVLnyZwiQ

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

Vulnerability Solution:

Enhance server configuration

Public Poc:

Not Available

25 HTTP Parameter Pollution

Description:

This script is possibly vulnerable to HTTP Parameter Pollution attacks. HPP attacks consist of injecting encoded query string delimiters into other existing parameters. If the web application does not properly sanitize the user input, a malicious user can compromise the logic of the application to perform either client-side or server-side attacks.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/hpp/?pp=12%26n945389%3Dv964683
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/?pp=12%26n945389%3Dv964683 has HTTP Parameter Pollution vulnerability
	Parameter names	url
	Payload	pp=12%26n945389%3Dv964683

References

REFERENCES
https://owasp.org/www-pdf-archive/AppsecEU09_CarettoniDiPaola_v0.8.pdf

Vulnerability Solution:

The application should properly sanitize user input (URL encode) to protect against this vulnerability.

Public Poc:

Not Available

26 Directory Traversal

Description:

Directory traversal is a vulnerability that allows an attacker to access a restricted directory and read files outside the Web server's root directory.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/showimage.php?file=../../../../../../../../../../../../../../proc/version
	Vulnerability details	Target http://testphp.vulnweb.com/showimage.php?file=../../../../../../../../../../../../../../proc/version has Directory Traversal vulnerability
	Parameter names	url
	Payload	../../../../../../../../../../../../../../proc/version

References

REFERENCES
https://www.acunetix.com/websitesecurity/directory-traversal/

Vulnerability Solution:

Filters user input parameters.

Public Poc:

Not Available

27 - 28 JetBrains Idea File Disclosure

Description:

Information disclosure refers to a website or a file which reveals sensitive information unintentionally to its users. Sensitive information includes password, key, session ID, license, personal data such as text messages, authorization credentials, personal identifiable information(name, address, telephone number and etc), program files, configuration files, log files, backup files and database

Affected Nodes:

1/2	Target	http://testphp.vulnweb.com/showimage.php?file=.idea/workspace.xml
	Vulnerability details	Target http://testphp.vulnweb.com/showimage.php?file=.idea/workspace.xml has JetBrains Idea File Disclosure vulnerability
	Parameter names	url
	Payload	.idea/workspace.xml
2/2	Target	http://testphp.vulnweb.com/.idea/workspace.xml
	Vulnerability details	Target http://testphp.vulnweb.com/.idea/workspace.xml has JetBrains Idea File Disclosure vulnerability
	Parameter names	url
	Payload	.idea/workspace.xml

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

Vulnerability Solution:

Enhance server configuration

Public Poc:

Not Available

29 - 30 Website Backup File Disclosure

Description:

Information disclosure refers to a website or a file which reveals sensitive information unintentionally to its users. Sensitive information includes password, key, session ID, license, personal data such as text messages, authorization credentials, personal identifiable information(name, address, telephone number and etc), program files, configuration files, log files, backup files and database

Affected Nodes:

1/2	Target	http://testphp.vulnweb.com/index.bak
	Vulnerability details	Target http://testphp.vulnweb.com/index.bak has Website Backup File Disclosure vulnerability
	Parameter names	url
	Payload	index.bak
2/2	Target	http://testphp.vulnweb.com/index.zip
	Vulnerability details	Target http://testphp.vulnweb.com/index.zip has Website Backup File Disclosure vulnerability
	Parameter names	url
	Payload	index.zip

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

Vulnerability Solution:

Enhance server configuration

Public Poc:

Not Available

31 PHP display_errors Enabled

Description:

The display_errors directive determines whether error messages should be sent to the browser. These messages frequently contain sensitive information about your web application environment, and should never be presented to untrusted sources. display_errors is on by default.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/secured/phpinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/phpinfo.php has PHP display_errors Enabled vulnerability

	Parameter names	url
	Payload	

References

REFERENCES
http://www.php.net/manual/en/errorfunc.configuration.php#ini.error-reporting

Vulnerability Solution:

You can disable display_errors from php.ini or .htaccess.php.ini display_errors = 'off'.

Public Poc:

Not Available

32 PHP allow_url_fopen Enabled

Description:

The PHP configuration directive allow_url_fopen is enabled. When enabled, this directive allows data retrieval from remote locations (web site or FTP server). A large number of code injection vulnerabilities reported in PHP-based web applications are caused by the combination of enabling allow_url_fopen and bad input filtering. allow_url_fopen is enabled by default.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/secured/phpinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/phpinfo.php has PHP allow_url_fopen Enabled vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
http://www.php.net/manual/en/filesystem.configuration.php

Vulnerability Solution:

You can disable allow_url_fopen from php.ini or .htaccess.php.ini allow_url_fopen = 'off'.

Public Poc:

Not Available

33 Improper Configuration of crossdomain.xml

Description:

The browser security model normally prevents web content from one domain from accessing data from another domain. This is commonly known as the "same origin policy". URL policy files grant cross-domain permissions for reading data. They permit operations that are not permitted by default. The URL policy file is located, by default, in the root directory of the target server, with the name crossdomain.xml (for example, at www.example.com/crossdomain.xml). [break][break] When a domain is specified in crossdomain.xml file, the site declares that it is willing to allow the operators of any servers in that domain to obtain any document on the server where the policy file resides. The crossdomain.xml file deployed on this website opens the server to all domains (use of a single asterisk "*" as a pure wildcard is supported). This practice is suitable for public servers, but should

not be used for sites located behind a firewall because it could permit access to protected areas. It should not be used for sites that require authentication in the form of passwords or cookies. Sites that use the common practice of authentication based on cookies to access private or user-specific data should be especially careful when using cross-domain policy files.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/crossdomain.xml
	Vulnerability details	Target http://testphp.vulnweb.com/crossdomain.xml has Improper Configuration of crossdomain.xml vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://www.acunetix.com/vulnerabilities/web/insecure-crossdomain-xml-file/

Vulnerability Solution:

Carefully evaluate which sites will be allowed to make cross-domain calls. Consider network topology and any authentication mechanisms that will be affected by the configuration or implementation of the cross-domain policy.

Public Poc:

Not Available

34 .htaccess File Disclosure

Description:

Information disclosure refers to a website or a file which reveals sensitive information unintentionally to its users. Sensitive information includes password, key, session ID, license, personal data such as text messages, authorization credentials, personal identifiable information(name, address, telephone number and etc), program files, configuration files, log files, backup files and database

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess has .htaccess File Disclosure vulnerability
	Parameter names	url
	Payload	.htaccess

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html

Vulnerability Solution:

Enhance server configuration

Public Poc:

Not Available

35 Invalid Page Text Search

Description:

Improper Error Handling of webpage may leak path information and other sensitive information.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/showimage.php?file=
	Vulnerability details	Target http://testphp.vulnweb.com/showimage.php?file= has Invalid Page Text Search vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://owasp.org/www-community/Improper_Error_Handling

Vulnerability Solution:

Follow OWASP recommendation to have a common error handling policy and apply the policy consistently to a website.

Public Poc:

Not Available

36 Cross Domain Data Hijacking

Description:

This page is possibly vulnerable to Cross domain data hijacking. If an attacker can create/upload a malicious Flash (SWF) file or control the top part of any page he can perform an attack known as Cross domain data hijacking. The Content-Type of the response doesn't matter. If the file is embedded using an tag, it will be executed as a Flash file as long as the content of the file looks like a valid Flash file.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/hpp/params.php?p=CWS%07%0E000x%9C=%8D1N%C3%40%10E%DF%AE%8D%BDI%08)%D3%40%1D%A0%A2%05%09%11%89HiP"%05D%8BF%8E%0BG%26%1B%D9%8E%117%A0%A2%DC%82%8A%1Br%04X;!S%8C%FE%CC%9B%F9%FF%AA%CB7Jq%AF%7F%ED%F2.%F8%01>%9E%18p%C9c%9Al%8B%ACzG%F2%DC%BEM%EC%ABdkj%1E%AC%2C%9F%A5(%B1%EB%89T%C2Jj)%93"%DBT7%24%9C%8FH%CBD6)%A3%0Bx)%AC%AD%D8%92%FB%1F%5C%07C%AC%7C%80Q%A7Nc%F4b%E8%FA%98%20b_%26%1C%9F5%20h%F1%D1g%0F%14%C1%0A]s%8D%8B0Q%A8L<%9B6%D4L%BD_%A8w%7E%9D[%17%F3/[%DCm{%EF%CB%EF%E6%8D:n-%FB%B3%C3%DD.%E3d1d%EC%C7%3F6%CD0%09&pp=12

	Vulnerability details	Target http://testphp.vulnweb.com/hpp/params.php?p=CWS%07%0E000x%9C=%8D1N%C3%40%10E%DF%AE%8D%BDI%08)%D3%40%1D%A0%A2%05%09%11%89HiP"%05D%8BF%8E%0BG%26%1B%D9%8E%117%A0%A2%DC%82%8A%1Br%04X;!S%8C%FE%CC%9B%F9%FF%AA%CB7Jq%AF%7F%ED%F2.%F8%01>%9E%18p%C9c%9Al%8B%ACzG%F2%DC%BEM%EC%ABdkj%1E%AC%2C%9F%A5(%B1%EB%89T%C2Jj)%93"%DBT7%24%9C%8FH%CBD6)%A3%0Bx)%AC%AD%D8%92%FB%1F%5C%07C%AC%7C%80Q%A7Nc%F4b%E8%FA%98%20b_%26%1C%9F5%20h%F1%D1g%0F%14%C1%0A]s%8D%8B0Q%A8L<%9B6%D4L%BD_%A8w%7E%9D[%17%F3/[%DCm{%EF%CB%EF%E6%8D:n-%FB%B3%C3%DD.%E3d1d%EC%C7%3F6%CD0%09&pp=12 has Cross Domain Data Hijacking vulnerability
	Parameter names	p
	Payload	CWS%07%0E000x%9C=%8D1N%C3%40%10E%DF%AE%8D%BDI%08)%D3%40%1D%A0%A2%05%09%11%89HiP"%05D%8BF%8E%0BG%26%1B%D9%8E%117%A0%A2%DC%82%8A%1Br%04X;!S%8C%FE%CC%9B%F9%FF%AA%CB7Jq%AF%7F%ED%F2.%F8%01>%9E%18p%C9c%9Al%8B%ACzG%F2%DC%BEM%EC%ABdkj%1E%AC%2C%9F%A5(%B1%EB%89T%C2Jj)%93"%DBT7%24%9C%8FH%CBD6)%A3%0Bx)%AC%AD%D8%92%FB%1F%5C%07C%AC%7C%80Q%A7Nc%F4b%E8%FA%98%20b_%26%1C%9F5%20h%F1%D1g%0F%14%C1%0A]s%8D%8B0Q%A8L<%9B6%D4L%BD_%A8w%7E%9D[%17%F3/[%DCm{%EF%CB%EF%E6%8D:n-%FB%B3%C3%DD.%E3d1d%EC%C7%3F6%CD0%09

References

REFERENCES
https://developer.mozilla.org/en-US/docs/Web/Security/Types_of_attacks#click-jacking

Vulnerability Solution:

For file uploads: It is recommended to check the file's content to have the correct header and format. If possible, use "Content-Disposition: attachment; filename=Filename.Extension;" header for the files that do not need to be served in the web browser. Isolating the domain of the uploaded files is also a good solution as long as the crossdomain.xml file of the main website does not include the isolated domain. For other cases: For JSONP abuses or other cases when the attacker control the top part of the page, you need to perform proper input filtering to protect against this type of issues.

Public Poc:

Not Available

37 PHP session.use_only_cookies Disabled

Description:

When use_only_cookies is disabled, PHP will pass the session ID via the URL. This makes the application more vulnerable to session hijacking attacks. Session hijacking is basically a form of identity theft wherein a hacker impersonates a legitimate user by stealing his session ID. When the session token is transmitted in a cookie, and the request is made on a secure channel (that is, it uses SSL), the token is secure.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/secured/phpinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/phpinfo.php has PHP session.use_only_cookies Disabled vulnerability
	Parameter names	url

	Payload

References

REFERENCES
http://www.php.net/session.configuration

Vulnerability Solution:

You can enabled session.use_only_cookies from php.ini or .htaccess.php.ini session.use_only_cookies = 'on' .htaccessphp_flag session.use_only_cookies on.

Public Poc:

Not Available

34 Low Vulnerabilities

1 Clickjacking due to X-Frame-Options Not Being Set in Response Header

Description:

Clickjacking, also known as a “UI redress attack”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page. Thus, the attacker is “hijacking” clicks meant for their page and routing them to another page, most likely owned by another application, domain, or both.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/
	Vulnerability details	Target http://testphp.vulnweb.com/ has Clickjacking due to X-Frame-Options Not Being Set in Response Header vulnerability
	Parameter names	X-Frame-Options
	Payload	

References

REFERENCES
https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html#Defending_with_Content_Security_Policy_frame-ancestors_directive
https://owasp.org/www-community/attacks/Clickjacking

Vulnerability Solution:

1. Restrict iframe busting via javascript.

2. Restrict iframe loading via setting in a response header X-Frame-Options. DENY: browser denies any frame loading pages; SAMEORIGEN: only allow the frame page from the same domain; ALLOW-FROM: customized permission, specify IP address that allows frame pages from.

3. In addition, some browser uses extension to combat clickjacking , such as Firefox extension 'Content-Security-Policy' and 'No-script'

Public Poc:

Not Available

2 - 30 Possible Relative Path Overwrite

Description:

Security researcher introduced a technique to take advantage of CSS imports with relative URLs by overwriting their target file. This technique can be used by an attacker to trick browsers into importing HTML pages as CSS Style Sheets. If the attacker can control a part of the imported HTML pages he can abuse this issue to inject arbitrary CSS rules.

Affected Nodes:

1/29	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
2/29	Target	http://testphp.vulnweb.com/index.php
	Vulnerability details	Target http://testphp.vulnweb.com/index.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
3/29	Target	http://testphp.vulnweb.com/AJAX/showxml.php
	Vulnerability details	Target http://testphp.vulnweb.com/AJAX/showxml.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
4/29	Target	http://testphp.vulnweb.com/AJAX/categories.php
	Vulnerability details	Target http://testphp.vulnweb.com/AJAX/categories.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
5/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
6/29	Target	http://testphp.vulnweb.com/AJAX/titles.php
	Vulnerability details	Target http://testphp.vulnweb.com/AJAX/titles.php has Possible Relative Path Overwrite vulnerability

	Parameter names	url
	Payload	
7/29	Target	http://testphp.vulnweb.com/signup.php
	Vulnerability details	Target http://testphp.vulnweb.com/signup.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
8/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
9/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
10/29	Target	http://testphp.vulnweb.com/secured/newuser.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/newuser.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	signup=signup&uaddress=data&ucc=data&uemail=data&upass=data&upass2=data&uphone=data&urname=surname&uuname=data
11/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
12/29	Target	http://testphp.vulnweb.com/AJAX/index.php

	Vulnerability details	Target http://testphp.vulnweb.com/AJAX/index.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
13/29	Target	http://testphp.vulnweb.com/cart.php
	Vulnerability details	Target http://testphp.vulnweb.com/cart.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
14/29	Target	http://testphp.vulnweb.com/AJAX/artists.php
	Vulnerability details	Target http://testphp.vulnweb.com/AJAX/artists.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
15/29	Target	http://testphp.vulnweb.com/hpp/
	Vulnerability details	Target http://testphp.vulnweb.com/hpp/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
16/29	Target	http://testphp.vulnweb.com/comment.php
	Vulnerability details	Target http://testphp.vulnweb.com/comment.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	Submit=Submit&comment=data&name=%3Cyour+name+here%3E&phpaction=echo+%24_POST%5Bcomment%5D%3B
17/29	Target	http://testphp.vulnweb.com/categories.php
	Vulnerability details	Target http://testphp.vulnweb.com/categories.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
18/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url

	Payload	
19/29	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
20/29	Target	http://testphp.vulnweb.com/userinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/userinfo.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
21/29	Target	http://testphp.vulnweb.com/
	Vulnerability details	Target http://testphp.vulnweb.com/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
22/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
23/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
24/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	

25/29	Target	http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php
	Vulnerability details	Target http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
26/29	Target	http://testphp.vulnweb.com/artists.php
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
27/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
28/29	Target	http://testphp.vulnweb.com/disclaimer.php
	Vulnerability details	Target http://testphp.vulnweb.com/disclaimer.php has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	
29/29	Target	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
	Vulnerability details	Target http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/ has Possible Relative Path Overwrite vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
http://www.thespanner.co.uk/2014/03/21/rpo/

Vulnerability Solution:

it's recommended to use absolute links for CSS imports. The problem can be partially mitigated by preventing framing. To prevent framing configure your web server to include an X-Frame-Options: deny header on all pages.

Public Poc:

Not Available

31 Hidden Input Form Found

Description:

A hidden form input named price was found. It's not recommended to hide sensitive information in hidden form fields.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/product.php?pic=1
	Vulnerability details	Target http://testphp.vulnweb.com/product.php?pic=1 has Hidden Input Form Found vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Vulnerability Solution:

Check if the script inputs are properly validated.

Public Poc:

Not Available

32 - 33 User Credentials in Plain Text

Description:

User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by malicious users.

Affected Nodes:

1/2	Target	http://testphp.vulnweb.com/signup.php
	Vulnerability details	Target http://testphp.vulnweb.com/signup.php has User Credentials in Plain Text vulnerability
	Parameter names	url
	Payload	
2/2	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has User Credentials in Plain Text vulnerability
	Parameter names	url

	Payload	

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Vulnerability Solution:

Because user credentials are considered sensitive information, should always be transferred to the server over an encrypted connection (HTTPS).

Public Poc:

Not Available

34 HTTP Content-Security-Policy Header Not Set

Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/
	Vulnerability details	Target http://testphp.vulnweb.com/ has HTTP Content-Security-Policy Header Not Set vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
N/A

Vulnerability Solution:

It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

Public Poc:

Not Available

20 Info Vulnerabilities

1 - 2 Password Type Input with auto-complete Enabled

Description:

When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved. Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the clear-text password from the browser cache.

Affected Nodes:

1/2	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has Password Type Input with auto-complete Enabled vulnerability
	Parameter names	url
	Payload	
2/2	Target	http://testphp.vulnweb.com/signup.php
	Vulnerability details	Target http://testphp.vulnweb.com/signup.php has Password Type Input with auto-complete Enabled vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration

Vulnerability Solution:

The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: INPUT TYPE="password" AUTOCOMPLETE="off".

Public Poc:

Not Available

3 - 17 Email Address Information Disclosure

Description:

One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.

Affected Nodes:

1/15	Target	http://testphp.vulnweb.com/index.php
	Vulnerability details	Target http://testphp.vulnweb.com/index.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
2/15	Target	http://testphp.vulnweb.com/login.php
	Vulnerability details	Target http://testphp.vulnweb.com/login.php has Email Address Information Disclosure vulnerability
	Parameter names	url

	Payload	
3/15	Target	http://testphp.vulnweb.com/listproducts.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?artist=1 has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
4/15	Target	http://testphp.vulnweb.com/cart.php
	Vulnerability details	Target http://testphp.vulnweb.com/cart.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
5/15	Target	http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php
	Vulnerability details	Target http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
6/15	Target	http://testphp.vulnweb.com/categories.php
	Vulnerability details	Target http://testphp.vulnweb.com/categories.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
7/15	Target	http://testphp.vulnweb.com/artists.php
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
8/15	Target	http://testphp.vulnweb.com/artists.php?artist=1
	Vulnerability details	Target http://testphp.vulnweb.com/artists.php?artist=1 has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
9/15	Target	http://testphp.vulnweb.com/

	Vulnerability details	Target http://testphp.vulnweb.com/ has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
10/15	Target	http://testphp.vulnweb.com/signup.php
	Vulnerability details	Target http://testphp.vulnweb.com/signup.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
11/15	Target	http://testphp.vulnweb.com/disclaimer.php
	Vulnerability details	Target http://testphp.vulnweb.com/disclaimer.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
12/15	Target	http://testphp.vulnweb.com/guestbook.php
	Vulnerability details	Target http://testphp.vulnweb.com/guestbook.php has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
13/15	Target	http://testphp.vulnweb.com/product.php?pic=1
	Vulnerability details	Target http://testphp.vulnweb.com/product.php?pic=1 has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	
14/15	Target	http://testphp.vulnweb.com/search.php?test=query
	Vulnerability details	Target http://testphp.vulnweb.com/search.php?test=query has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	goButton=go&searchFor=data
15/15	Target	http://testphp.vulnweb.com/listproducts.php?cat=1
	Vulnerability details	Target http://testphp.vulnweb.com/listproducts.php?cat=1 has Email Address Information Disclosure vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Vulnerability Solution:

Remove sensitive information such as email addresses to prevent internal account information from leaking

Public Poc:

Not Available

18 Insecure Referrer Policy

Description:

Referrer Policy controls behavior of the Referer header, which indicates the origin or web page URL the request was made from. The web application uses insecure Referrer Policy configuration that may leak user's information to third-party sites.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/
	Vulnerability details	Target http://testphp.vulnweb.com/ has Insecure Referrer Policy vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy

Vulnerability Solution:

Consider setting Referrer-Policy header to 'strict-origin-when-cross-origin' or a stricter value

Public Poc:

Not Available

19 Error Page Web Server Version Disclosure

Description:

By requesting a page that doesn't exist, an error page was returned. This error page contains the web server version number and a list of modules enabled on this server. This information can be used to conduct further attacks.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/IMcsSyTR4W.aspx
	Vulnerability details	Target http://testphp.vulnweb.com/IMcsSyTR4W.aspx has Error Page Web Server Version Disclosure vulnerability
	Parameter names	url

	Payload

References

REFERENCES
https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Vulnerability Solution:

Custom exception page

Public Poc:

Not Available

20 PHP open_basedir isnt Set

Description:

The open_basedir configuration directive will limit the files that can be opened by PHP to the specified directory-tree. When a script tries to open a file with, for example, fopen() or gzopen(), the location of the file is checked. When the file is outside the specified directory-tree, PHP will refuse to open it. open_basedir is a good protection against remote file inclusion vulnerabilities. For a remote attacker it is not possible to break out of the open_basedir restrictions if he is only able to inject the name of a file to be included. Therefore the number of files he will be able to include with such a local file include vulnerability is limited.

Affected Nodes:

1/1	Target	http://testphp.vulnweb.com/secured/phpinfo.php
	Vulnerability details	Target http://testphp.vulnweb.com/secured/phpinfo.php has PHP open_basedir isnt Set vulnerability
	Parameter names	url
	Payload	

References

REFERENCES
https://www.php.net/ini.core

Vulnerability Solution:

You can set open_basedir from php.ini open_basedir = your_application_directory.

Public Poc:

Not Available

Attack Surface Details

Total 51

INDEX	METHOD	URL	PARAMETERS
1	POST	http://testphp.vulnweb.com/userinfo.php	
2	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-2/	
3	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-1/	
4	GET	http://testphp.vulnweb.com/guestbook.php	

INDEX	METHOD	URL	PARAMETERS
5	POST	http://testphp.vulnweb.com/secured/newuser.php	signup, uaddress, ucc, uemail, upass, upass2, uphone, urname, uuname
6	GET	http://testphp.vulnweb.com/AJAX/artists.php	
7	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/	
8	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/	
9	GET	http://testphp.vulnweb.com/Templates/main_dynamic_template.dwt.php	
10	GET	http://testphp.vulnweb.com/showimage.php?file=	file
11	GET	http://testphp.vulnweb.com/	N/A
12	GET	http://testphp.vulnweb.com/artists.php?artist=1	artist
13	POST	http://testphp.vulnweb.com/guestbook.php	name, submit, text
14	GET	http://testphp.vulnweb.com/secured/	N/A
15	POST	http://testphp.vulnweb.com/search.php?test=query	test, goButton, searchFor
16	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/	
17	GET	http://testphp.vulnweb.com/comment.php?aid=1	aid
18	GET	http://testphp.vulnweb.com/search.php?test=query	test
19	GET	http://testphp.vulnweb.com/AJAX/index.php	
20	GET	http://testphp.vulnweb.com/hpp/	
21	GET	http://testphp.vulnweb.com/AJAX/titles.php	
22	GET	http://testphp.vulnweb.com/listproducts.php?artist=1	artist
23	GET	http://testphp.vulnweb.com/AJAX/categories.php	
24	GET	http://testphp.vulnweb.com/hpp/	N/A
25	GET	http://testphp.vulnweb.com/userinfo.php	
26	GET	http://testphp.vulnweb.com/hpp/?pp=12	pp
27	POST	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12	p, pp, aaaa%2F
28	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-2.html	
29	GET	http://testphp.vulnweb.com/listproducts.php?cat=1	cat
30	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-1.html	
31	GET	http://testphp.vulnweb.com/index.php	
32	GET	http://testphp.vulnweb.com/artists.php	
33	GET	http://testphp.vulnweb.com/signup.php	
34	GET	http://testphp.vulnweb.com/disclaimer.php	
35	GET	http://testphp.vulnweb.com	N/A
36	GET	http://testphp.vulnweb.com/categories.php	
37	GET	http://testphp.vulnweb.com/comment.php?pid=1	pid
38	GET	http://testphp.vulnweb.com/product.php?pic=1	pic
39	POST	http://testphp.vulnweb.com/userinfo.php	pass, uname
40	GET	http://testphp.vulnweb.com/login.php	
41	GET	http://testphp.vulnweb.com/	
42	GET	http://testphp.vulnweb.com/AJAX/showxml.php	
43	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/	
44	GET	http://testphp.vulnweb.com/cart.php	
45	POST	http://testphp.vulnweb.com/cart.php	addcart, price
46	GET	http://testphp.vulnweb.com/comment.php	

INDEX	METHOD	URL	PARAMETERS
47	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/BuyProduct-3/	
48	GET	http://testphp.vulnweb.com/Mod_Rewrite_Shop/RateProduct-3.html	
49	POST	http://testphp.vulnweb.com/comment.php	Submit, comment, name, phpaction
50	GET	http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12	p, pp
51	GET	http://testphp.vulnweb.com/secured/newuser.php	