Sri Lanka Institute of Information Technology

Web Security - IE2062

Journal Report

PERERA A.P.J

IT22280992

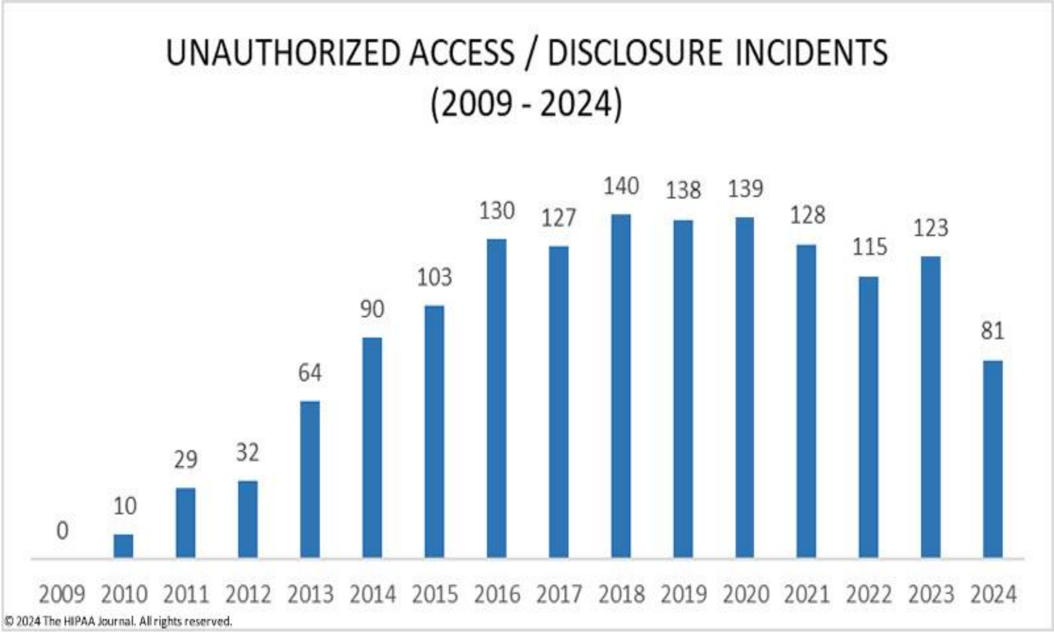
Group Y2S2.CS

Table of Contents

1. INTRODUCTION

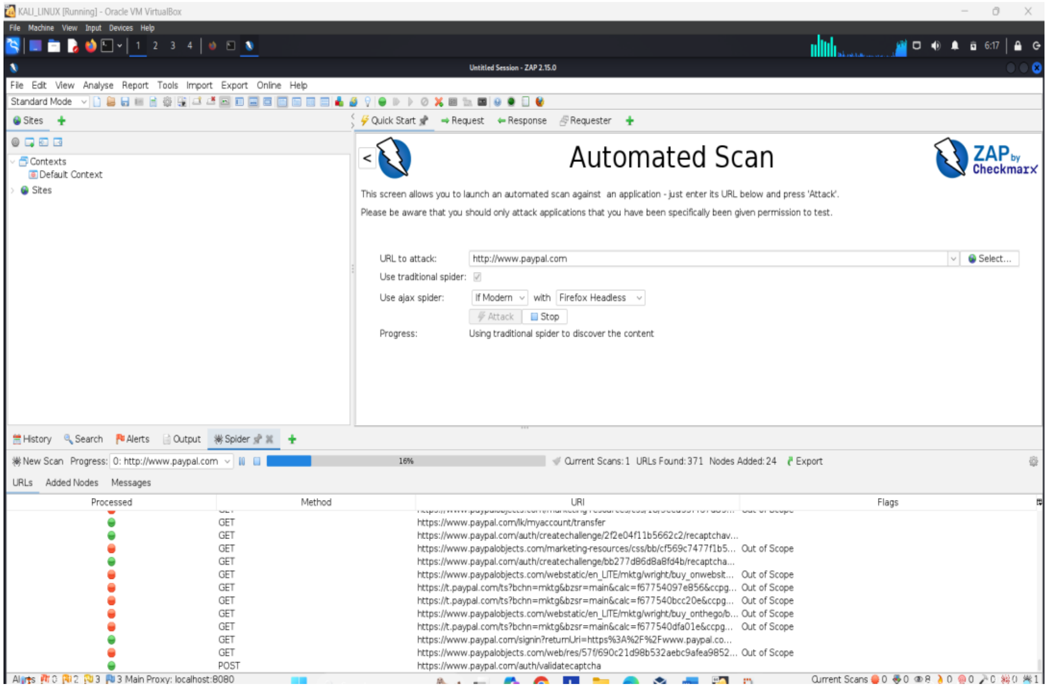
- In the journey of understanding and mastering cybersecurity, particularly in the realms of web security and Bug Bounty Hunting, each day brings forth unique challenges, insights, and learning experiences. This report chronicles my daily experiences, from uncovering vulnerabilities like PII disclosure and cross-domain script inclusion to exploring advanced tools such as OWASP ZAP, Burp Suite, and Nikto. These daily reflections not only highlight the technical aspects of the vulnerabilities discovered but also emphasize the significance of implementing effective mitigation strategies. Through these experiences, I have gained a deeper appreciation of the complexities involved in safeguarding information systems and the critical role of continuous learning in staying ahead of emerging threats.

2. Day 1

Date	2024/10/06
Summary Of the day's activities	▪ Today was a productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 1. I dedicated time to learning about data privacy and particularly PII disclosure, Vulnerable js Libarary, Cookie without secure flag vulnerabilities.  UNAUTHORIZED ACCESS / DISCLOSURE INCIDENTS (2009 - 2024) © 2024 The HIPAA Journal. All rights reserved.
Vulnerabilities discovered or explored	▪ PII Disclosure. ▪ Vulnerable JS Libarary. ▪ Cookie with Secure Flag.
Challenges faced and how they were overcome	▪ Time Consuming – Use Automation tools. ▪ Access to Resource and tools – Coloboration with Security Vendors.

New tools,techniques or concepts learned

- OWASP ZAP



KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
6:17
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
ZAP by Checkmarx

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://www.paypal.com Select...

Use traditional spider: ☑

Use ajax spider: If Modern with Firefox Headless

Attack Stop

Progress: Using traditional spider to discover the content

New Scan Progress: 0: http://www.paypal.com 16% Current Scans: 1 URLs Found: 371 Nodes Added: 24 Export

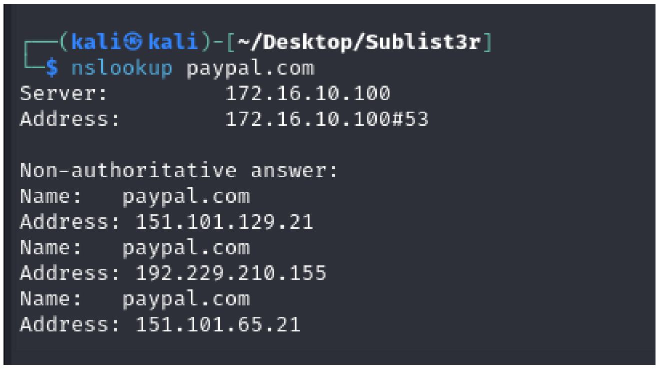
URLs Added Nodes Messages

Processed	Method	URI	Flags
🟢	GET	https://www.paypal.com/lk/myaccount/transfer	
🟢	GET	https://www.paypal.com/auth/createchallenge/2f2e04f11b5662c2/recaptchav...	
🔴	GET	https://www.paypalobjects.com/marketing-resources/css/bb/cf569c7477f1b5...	Out of Scope
🟢	GET	https://www.paypal.com/auth/createchallenge/bb277d86d8a8fd4b/recaptcha...	
🔴	GET	https://www.paypalobjects.com/webstatic/en_LITE/mktg/wright/buy_onwebsit...	Out of Scope
🔴	GET	https://t.paypal.com/ts?bchn=mktg&bzsr=main&calc=f67754097e856&ccpg...	Out of Scope
🔴	GET	https://t.paypal.com/ts?bchn=mktg&bzsr=main&calc=f677540bcc20e&ccpg...	Out of Scope
🔴	GET	https://www.paypalobjects.com/webstatic/en_LITE/mktg/wright/buy_onthego/b...	Out of Scope
🔴	GET	https://t.paypal.com/ts?bchn=mktg&bzsr=main&calc=f677540dfa01e&ccpg...	Out of Scope
🟢	GET	https://www.paypal.com/signin?returnUri=https%3A%2F%2Fwww.paypal.co...	
🔴	GET	https://www.paypalobjects.com/web/res/57f/690c21d98b532aebc9afea9852...	Out of Scope
🟢	POST	https://www.paypal.com/auth/validatecaptcha	

Alerts 0 2 3 3 Main Proxy: localhost:8080

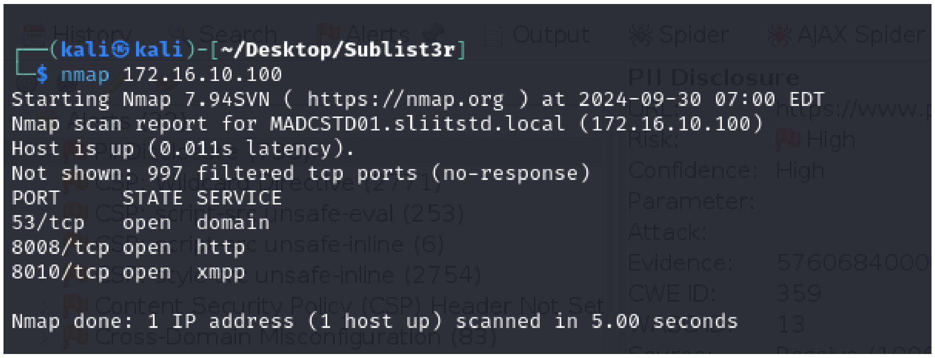
Current Scans 0 0 8 0 0 0 0 1

- Nslookup



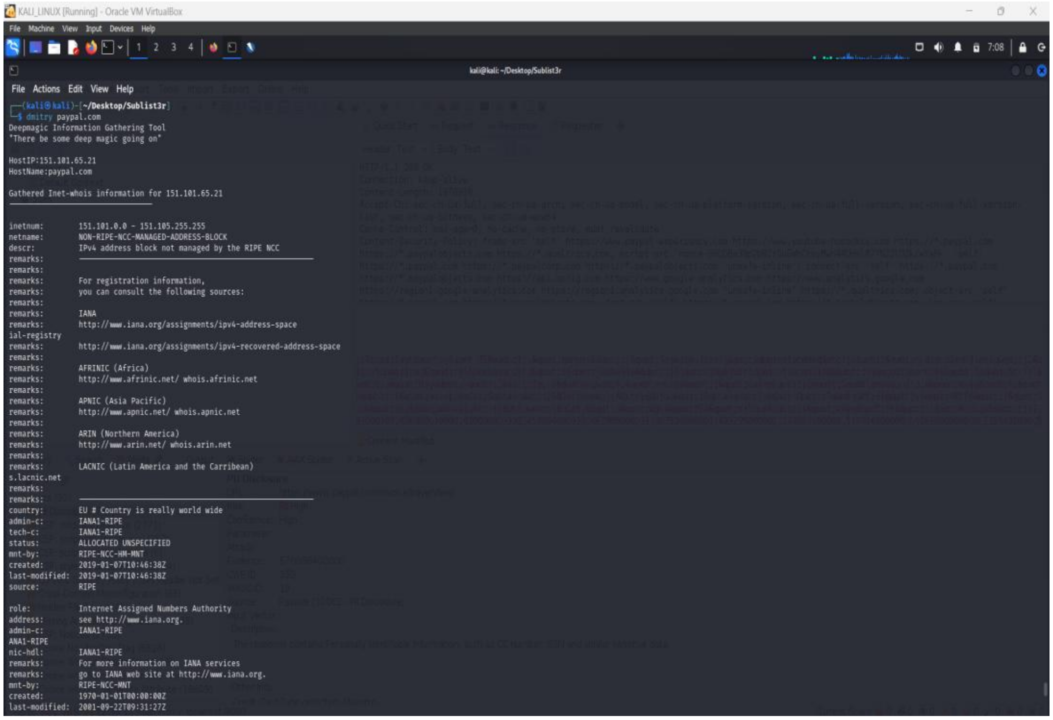
```
┌──(kali㉿kali)-[~/Desktop/Sublist3r]
└─$ nslookup paypal.com
Server:         172.16.10.100
Address:        172.16.10.100#53

Non-authoritative answer:
Name:   paypal.com
Address: 151.101.129.21
Name:   paypal.com
Address: 192.229.210.155
Name:   paypal.com
Address: 151.101.65.21
```

- Nmap



```
┌──(kali㉿kali)-[~/Desktop/Sublist3r]
└─$ nmap 172.16.10.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-30 07:00 EDT
Nmap scan report for MADCSTD01.sliitstd.local (172.16.10.100)
Host is up (0.011s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
8008/tcp open  http
8010/tcp open  xmpp

Nmap done: 1 IP address (1 host up) scanned in 5.00 seconds
```

- Dmitry

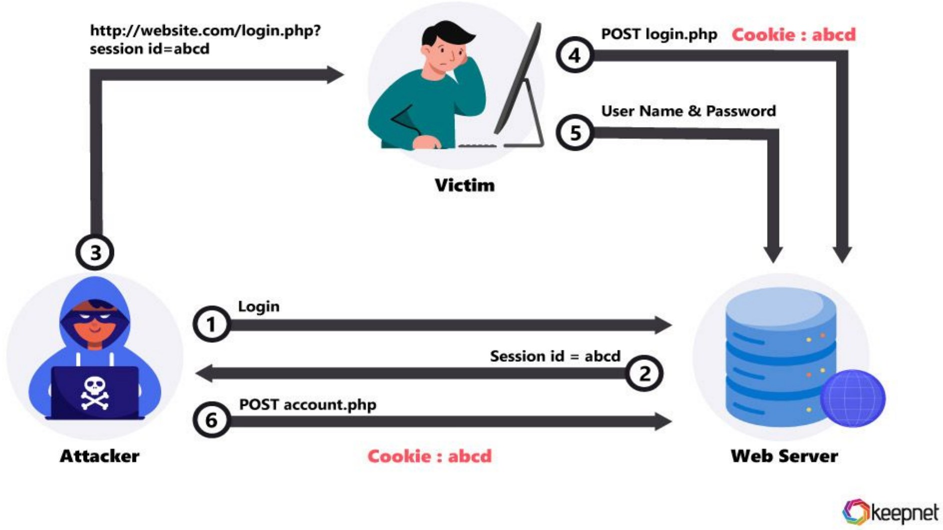


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
7:08
kali@kali: ~/Desktop/Sublist3r
File Actions Edit View Help

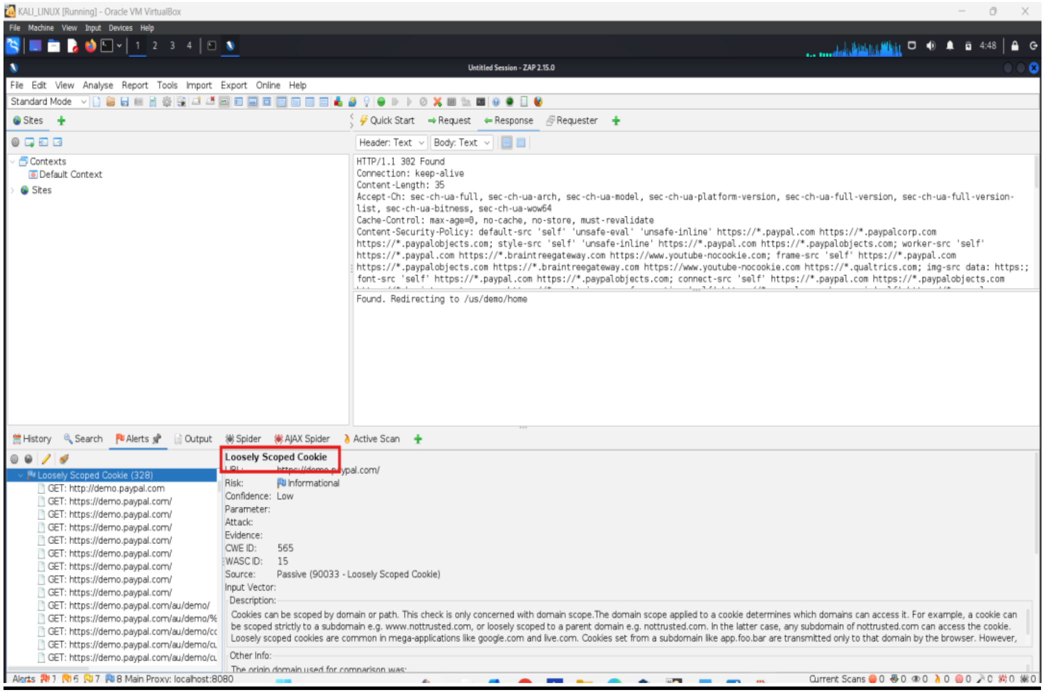
```
┌──(kali㉿kali)-[~/Desktop/Sublist3r]
└─$ dmitry paypal.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP:151.101.65.21
HostName:paypal.com

Gathered Inet-whois information for 151.101.65.21
---------------------------------


inetnum:        151.101.0.0 - 151.105.255.255
netname:        NON-RIPE-NCC-MANAGED-ADDRESS-BLOCK
descr:          IPv4 address block not managed by the RIPE NCC
remarks:        ------------------------------------------------------
remarks:
remarks:        For registration information,
remarks:        you can consult the following sources:
remarks:
remarks:        IANA
remarks:        http://www.iana.org/assignments/ipv4-address-space
ial-registry
remarks:        http://www.iana.org/assignments/ipv4-recovered-address-space
remarks:
remarks:        AFRINIC (Africa)
remarks:        http://www.afrinic.net/ whois.afrinic.net
remarks:
remarks:        APNIC (Asia Pacific)
remarks:        http://www.apnic.net/ whois.apnic.net
remarks:
remarks:        ARIN (Northern America)
remarks:        http://www.arin.net/ whois.arin.net
remarks:
remarks:        LACNIC (Latin America and the Carribean)
s.lacnic.net
remarks:
remarks:        ------------------------------------------------------
country:        EU # Country is really world wide
admin-c:        IANA1-RIPE
tech-c:         IANA1-RIPE
status:         ALLOCATED UNSPECIFIED
mnt-by:         RIPE-NCC-HM-MNT
created:        2019-01-07T10:46:38Z
last-modified:  2019-01-07T10:46:38Z
source:         RIPE

role:           Internet Assigned Numbers Authority
address:        see http://www.iana.org.
admin-c:        IANA1-RIPE
ANA1-RIPE
nic-hdl:        IANA1-RIPE
remarks:        For more information on IANA services
remarks:        go to IANA web site at http://www.iana.org.
mnt-by:         RIPE-NCC-MNT
created:        1970-01-01T00:00:00Z
last-modified:  2001-09-22T09:31:27Z
```

Reflections and takeaways	▪ The sharing of info that traces a person's identity with the government or any institition (PII) creates a lot of threats leading to for instance stealing a person's identity among other risks. In order to protect PII when embedded within documents, organizations must adopt strong policies such use as encryption and regular audits whilst user sensiation regarding the protection of the information must also be encouraged. On top of that, using 3rd party JavaScript libraries that are outdated or poorly written may open an application to a wide range of threats including, x cross-site forging (CSF) which can also be referred to as cross-site scripting. However, it is important because tonics to treat XSS do not work on some of the libraries and changes to the libraries have proven rather elusive. In addition, cookies that are not secure flagged tend to be dropped during activities and that exposes information they held to external parties. To mitigate this threat, every organization should implement secure flag, HttpOnly and SameSite attributes out of the cookies. In a nutshell, this means that there should be no complacency in dealing with these weaknesses, the combination of proper data handling, software engineering and education of the end users should be employed in dealing with the situation.

3. Day 2

Date	2024/10/07
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 2. I dedicated time to learning about data privacy and Cookie poisoning, Loosely Scooped Cookie vulnerabilities.  http://website.com/login.php? session id=abcd Victim 4 POST login.php Cookie : abcd 5 User Name & Password 3 Attacker 1 Login Session id = abcd 2 6 POST account.php Cookie : abcd Web Server keepnet
Vulnerabilities discovered or explored	▪ Cookie poisoning ▪ Loosely Scooped Cookie
Challenges faced and how they were overcome	▪ Human Error and oversights - Establish clear procedures and checklists for critical tasks to minimize errors.

New tools,techniques or concepts learned


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
History
Search
Alerts
Output
Spider
AJAX Spider
Active Scan
Loosely Scoped Cookie (328)
Alerts 1 5 7 8 Main Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0

```
HTTP/1.1 302 Found
Connection: keep-alive
Content-Length: 35
Accept-Ch: sec-ch-ua-full, sec-ch-ua-arch, sec-ch-ua-model, sec-ch-ua-platform-version, sec-ch-ua-full-version, sec-ch-ua-full-version-
list, sec-ch-ua-bitness, sec-ch-ua-wow64
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Security-Policy: default-src 'self' 'unsafe-eval' 'unsafe-inline' https://*.paypal.com https://*.paypalcorp.com
https://*.paypalobjects.com; style-src 'self' 'unsafe-inline' https://*.paypal.com https://*.paypalobjects.com; worker-src 'self'
https://*.paypal.com https://*.braintreegateway.com https://www.youtube-nocookie.com; frame-src 'self' https://*.paypal.com
https://*.paypalobjects.com https://*.braintreegateway.com https://www.youtube-nocookie.com https://*.qualtrics.com; img-src data: https:;
font-src 'self' https://*.paypal.com https://*.paypalobjects.com; connect-src 'self' https://*.paypal.com https://*.paypalobjects.com
```

```
Found. Redirecting to /us/demo/home
```

```
GET: http://demo.paypal.com
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/
GET: https://demo.paypal.com/au/demo/
GET: https://demo.paypal.com/au/demo/%
GET: https://demo.paypal.com/au/demo/cc
GET: https://demo.paypal.com/au/demo/cu
GET: https://demo.paypal.com/au/demo/cu
```

Loosely Scoped Cookie

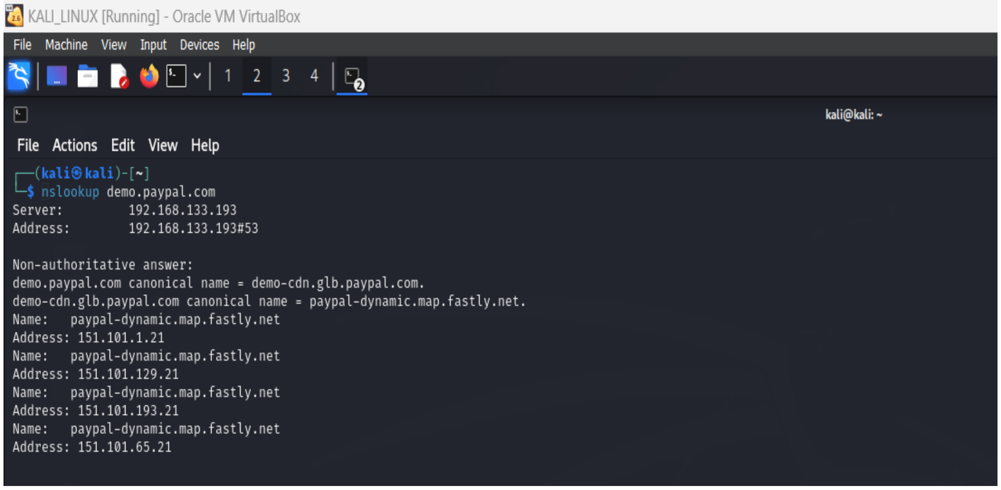
URL: https://demo.paypal.com/
Risk: Informational
Confidence: Low
Parameter:
Attack:
Evidence:
CWE ID: 565
WASC ID: 15
Source: Passive (90033 - Loosely Scoped Cookie)
Input Vector:

Description:

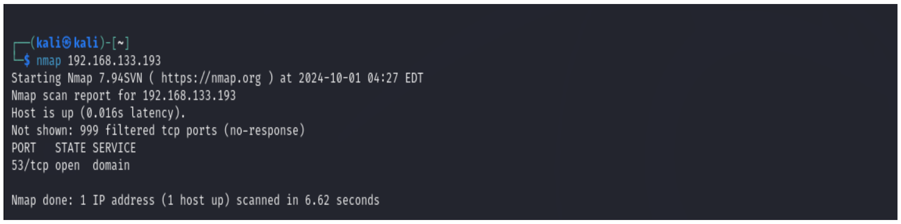
Cookies can be scoped by domain or path. This check is only concerned with domain scope.The domain scope applied to a cookie determines which domains can access it. For example, a cookie can be scoped strictly to a subdomain e.g. www.nottrusted.com, or loosely scoped to a parent domain e.g. nottrusted.com. In the latter case, any subdomain of nottrusted.com can access the cookie. Loosely scoped cookies are common in mega-applications like google.com and live.com. Cookies set from a subdomain like app.foo.bar are transmitted only to that domain by the browser. However,

Other Info:

The origin domain used for comparison was:


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nslookup demo.paypal.com
Server:         192.168.133.193
Address:        192.168.133.193#53

Non-authoritative answer:
demo.paypal.com canonical name = demo-cdn.glb.paypal.com.
demo-cdn.glb.paypal.com canonical name = paypal-dynamic.map.fastly.net.
Name:   paypal-dynamic.map.fastly.net
Address: 151.101.1.21
Name:   paypal-dynamic.map.fastly.net
Address: 151.101.129.21
Name:   paypal-dynamic.map.fastly.net
Address: 151.101.193.21
Name:   paypal-dynamic.map.fastly.net
Address: 151.101.65.21
```

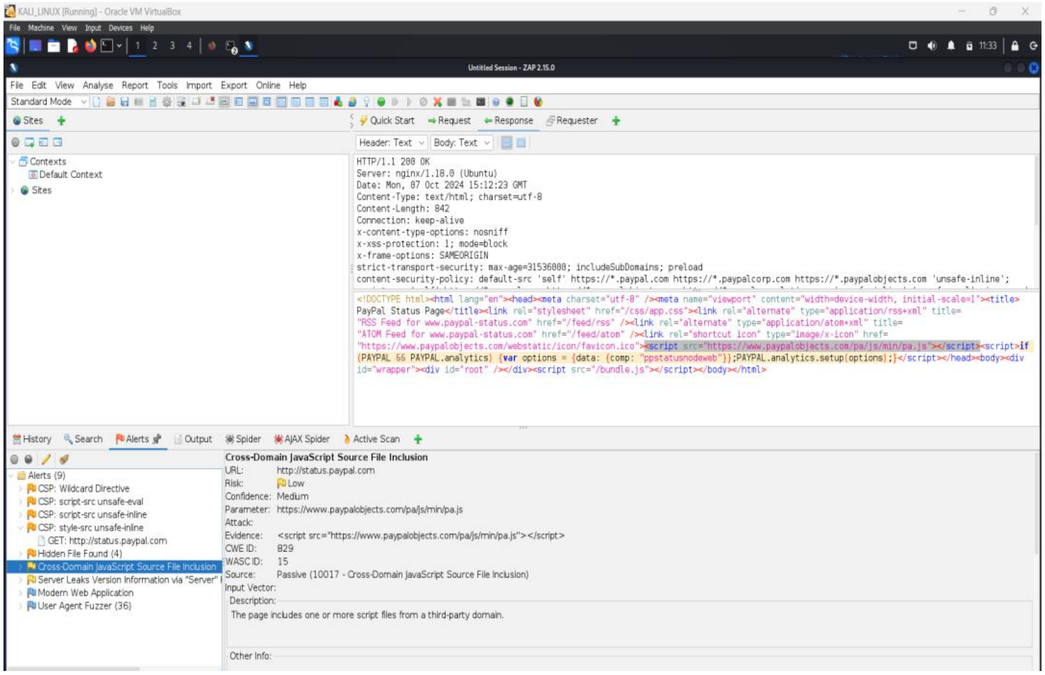


```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.133.193
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-01 04:27 EDT
Nmap scan report for 192.168.133.193
Host is up (0.016s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 6.62 seconds
```

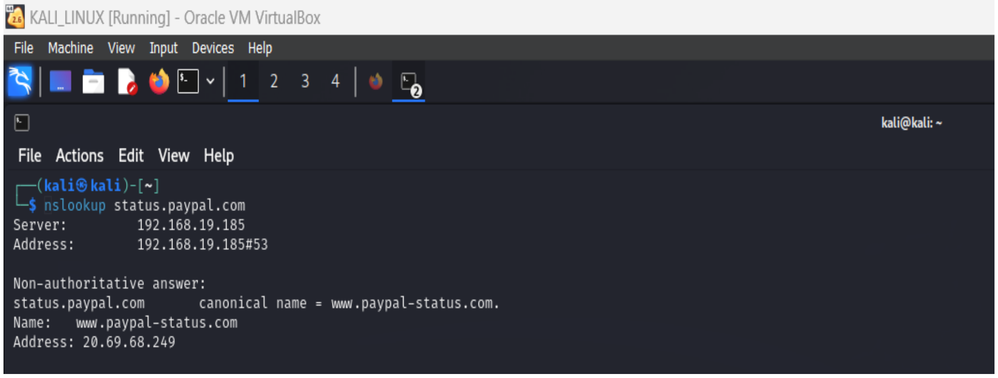
Reflections and takeaways

- poisoning and poorly scoped cookies are two web security issues that pose serious threats resulting into access and information compromise. In other words cookie poisoning is manipulation of the cookie data by a malicious user in an attempt to obtain unwarranted access and usurp another user's authority in order to breach the user's sensitive information and the application as a whole. Loosely scoped cookies are referred to as cookies that do not have strict domain or path limitation; making them available for cross application or across different subdomains usage, hence increasing risks of csrf and other attacks. In order to protect from such threats, it is very important for developers to make sure that cookie data is correctly validated and sanitized and only safe data is allowed to be used. Application of such secure attributes for cookies such as Secure, HttpOnly, SameSite and others is also effective in improving the security of the cookies. On the whole, vigilance and preventive actions are key in preventing exploitation of cookies and ensuring that user information remains secure.

4. Day 3

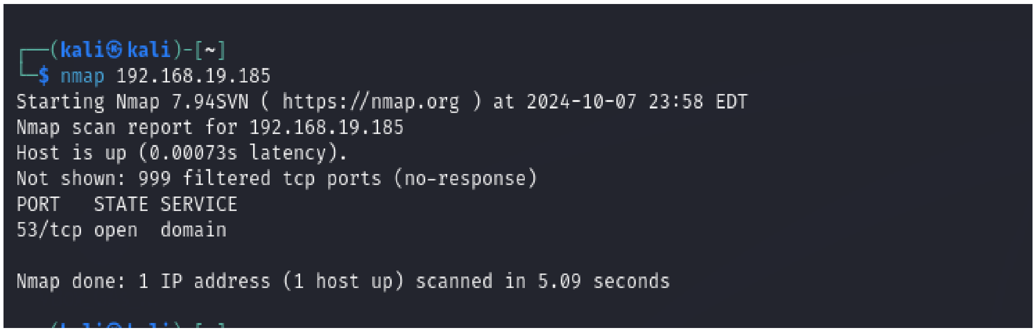
Date	2024/10/08
Summary Of the day's activities	Today was a another productive day focused on enhancing my skills in Bug Bounty Hunting day 2. I dedicated time to learning about data privacy and Cross domain java script source file inclution vulnerabilities.
Vulnerabilities discovered or explored	▪ Cross domain java script source file inclution
Challenges faced and how they were overcome	
New tools,techniques or concepts learned	▪ OWASP ZAP  KALI_LINUX [Running] - Oracle VM VirtualBox File Machine View Input Devices Help Untitled Session - ZAP 2.15.0 File Edit View Analyse Report Tools Import Export Online Help Standard Mode Sites Contexts Default Context Sites Quick Start Request Response Requester Header: Text Body: Text History Search Alerts Output Spider AJAX Spider Active Scan HTTP/1.1 200 OK Server: nginx/1.18.0 (Ubuntu) Date: Mon, 07 Oct 2024 15:12:23 GMT Content-Type: text/html; charset=utf-8 Content-Length: 842 Connection: keep-alive x-content-type-options: nosniff x-xss-protection: 1; mode=block x-frame-options: SAMEORIGIN strict-transport-security: max-age=31536000; includeSubDomains; preload content-security-policy: default-src 'self' https://*.paypal.com https://*.paypalcorp.com https://*.paypalobjects.com 'unsafe-inline'; <!DOCTYPE html><html lang="en"><head><meta charset="utf-8" /><meta name="viewport" content="width=device-width, initial-scale=1"><title>PayPal Status Page</title><link rel="stylesheet" href="/css/app.css"><link rel="alternate" type="application/rss+xml" title="RSS Feed for www.paypal-status.com" href="/feed/rss" /><link rel="alternate" type="application/atom+xml" title="ATOM Feed for www.paypal-status.com" href="/feed/atom" /><link rel="shortcut icon" type="image/x-icon" href="https://www.paypalobjects.com/webstatic/icon/favicon.ico"><script src="https://www.paypalobjects.com/pa/js/min/pa.js"></script><script>if (PAYPAL && PAYPAL.analytics) {var options = {data: {comp: "ppstatusnodeweb"}};PAYPAL.analytics.setup(options);}</script></head><body><div id="wrapper"><div id="root" /></div><script src="/bundle.js"></script></body></html> Alerts (9) CSP: Wildcard Directive CSP: script-src unsafe-eval CSP: script-src unsafe-inline CSP: style-src unsafe-inline GET: http://status.paypal.com Hidden File Found (4) Cross-Domain JavaScript Source File Inclusion Server Leaks Version Information via "Server" Modern Web Application User Agent Fuzzer (36) Cross-Domain JavaScript Source File Inclusion URL: http://status.paypal.com Risk: Low Confidence: Medium Parameter: https://www.paypalobjects.com/pa/js/min/pa.js Attack: Evidence: <script src="https://www.paypalobjects.com/pa/js/min/pa.js"></script> CWE ID: 829 WASC ID: 15 Source: Passive (10017 - Cross-Domain JavaScript Source File Inclusion) Input Vector: Description: The page includes one or more script files from a third-party domain. Other Info:

- Nslookup


KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nslookup status.paypal.com
Server:         192.168.19.185
Address:        192.168.19.185#53

Non-authoritative answer:
status.paypal.com       canonical name = www.paypal-status.com.
Name:   www.paypal-status.com
Address: 20.69.68.249
```

- Nmap

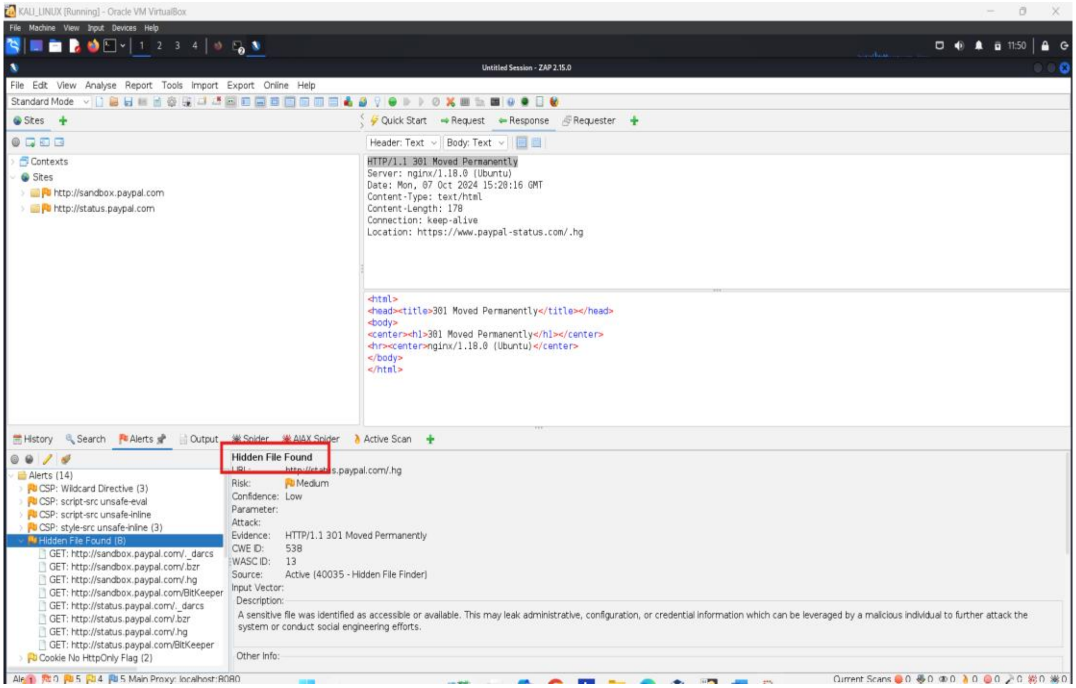


```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.19.185
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-07 23:58 EDT
Nmap scan report for 192.168.19.185
Host is up (0.00073s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 5.09 seconds
```

Reflections and takeaways

- Cross-domain JavaScript source file incorporation poses grave security risks when a web application is built to incorporate and run scripts from different domains. Such vulnerabilities can result in several forms of attacks, including but not limited to, Cross-Site Scripting (XSS), and information theft whereby an attack uses included scripts to run a code against a victim session. The central scope that is to be emphasized is the fact that very strict measures ought to be put in place regarding the use of external scripts; applications should allow the use of external scripts only from trusted sources and even that, it is best if a Content Security Policy (CSP) is put in place to define the permitted sources. Moreover, the application of risk reduction principles, such as, checking and cleaning any data engaged in incorporating a script can be of assistance. In addition, it is necessary for the developers to conduct audits on the external sources and the dependencies on a regular basis in order to maintain them patched and risk-free. In conclusion, ensuring regards development as an activity that must expose no risks and safe coding measures are employed while being aware of cross domain risks, would help curb the misuse of applications.

5. Day 4

Date	2024/10/09
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 4. I dedicated time to learning about data privacy and Hidden file found vulnerabilitiy.
Vulnerabilities discovered or explored	▪ Hidden file found.
Challenges faced and how they were overcome	▪ Technical understanding – Documentation and Resources
New tools,techniques or concepts learned	▪ OWASP ZAP 

KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Sites
http://sandbox.paypal.com
http://status.paypal.com
Quick Start
Request
Response
Requester
Header: Text
Body: Text
History
Search
Alerts
Output
Spider
AJAX Spider
Active Scan
Current Scans 0 0 0 0 0 0 0 0

```
HTTP/1.1 301 Moved Permanently
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 07 Oct 2024 15:20:16 GMT
Content-Type: text/html
Content-Length: 178
Connection: keep-alive
Location: https://www.paypal-status.com/.hg
```

```
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.18.0 (Ubuntu)</center>
</body>
</html>
```

```
Alerts (14)
  CSP: Wildcard Directive (3)
  CSP: script-src unsafe-eval
  CSP: script-src unsafe-inline
  CSP: style-src unsafe-inline (3)
  Hidden File Found (8)
    GET: http://sandbox.paypal.com/._darcs
    GET: http://sandbox.paypal.com/.bzr
    GET: http://sandbox.paypal.com/.hg
    GET: http://sandbox.paypal.com/BitKeeper
    GET: http://status.paypal.com/._darcs
    GET: http://status.paypal.com/.bzr
    GET: http://status.paypal.com/.hg
    GET: http://status.paypal.com/BitKeeper
  Cookie No HttpOnly Flag (2)
```

Hidden File Found

URL: http://status.paypal.com/.hg
Risk: Medium
Confidence: Low
Parameter:
Attack:
Evidence: HTTP/1.1 301 Moved Permanently
CWE ID: 538
WASC ID: 13
Source: Active (40035 - Hidden File Finder)
Input Vector:

Description:

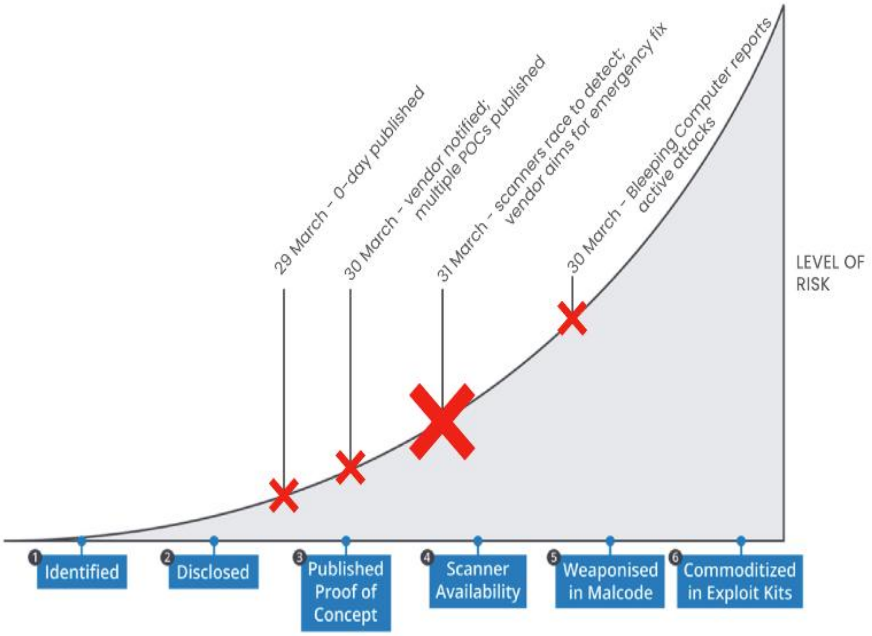
A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.

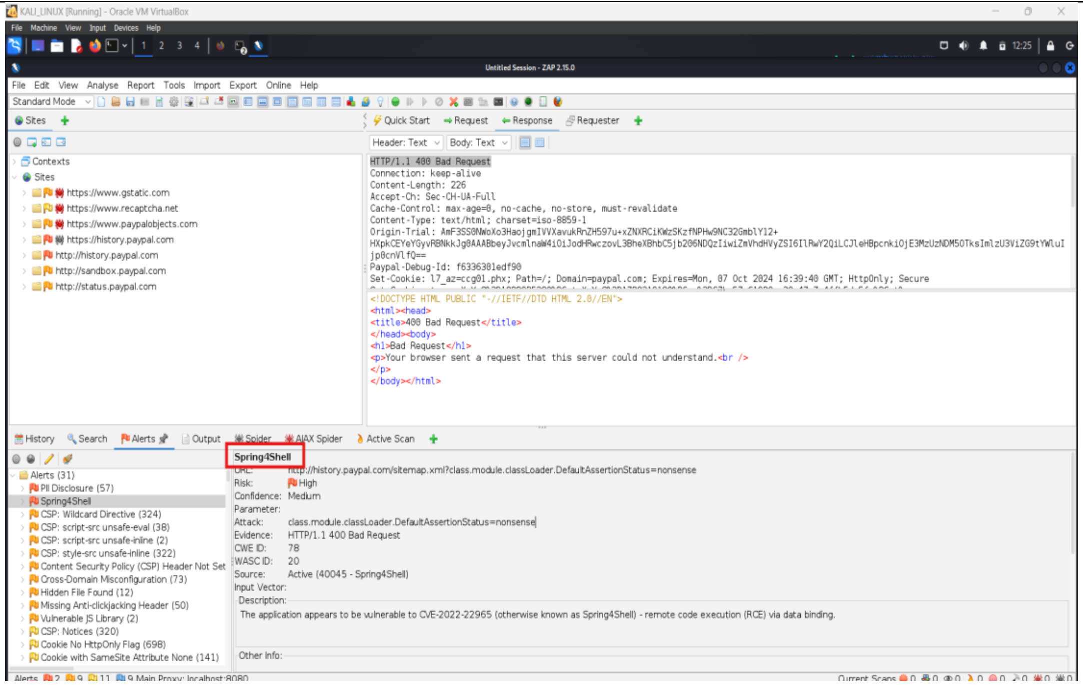
Other Info:

Alerts 1 0 5 4 5 Main Proxy: localhost:8080

Reflections and takeaways	▪ Finding any concealed files in a web application or a server is very serious in terms of security. Such files may include sensitive data or even configuration settings that can be exploited by attackers. One of the critical issues that come to mind is whether there should be security audits carried out after a certain amount of time. Also, regular scans should be done in order to find such files that would not be available to anyone unelevated. As hidden files may be a way through which an attacker can access secured areas, other measures, such as access control, and ensuring that the elevations are done correctly, are also necessary. It is also worth reminding any reader that maintaining security in such cases entails certain measures, such as updating the system's server software from time to time or installing a reliable firewall. Most importantly, it should be emphasized that the so-called least privileged access policy must be followed, as it is very important that a user is able to access only those files that are needed in their work. In general, it is commendable to educate programmers and network administrators about the dangers posed by hidden files in order to promote a culture of protecting information from possible leaks.

6. Day 5

Date	2024/10/10
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 5. I dedicated time to learning about data privacy and Hidden file found vulnerabilitiy
Vulnerabilities discovered or explored	▪ Spring 4 shell. Spring4Shell Exploit Weaponization Timeline*  * Figure based on model from the Recorded Future *Threat Intelligence Handbook*
Challenges faced and how they were overcome	
New tools,techniques	▪ OWASP ZAP

or concepts learned	
Reflections and takeaways	▪ Spring 4 Shell is an interactive command line application creation framework for Spring based applications that allows for quick iteration and efficient management of Spring based applications. One thing to note about it is that it eases testing and debugging of applications without having to employ an entire user interface. One can interactively probe the application context and manipulate it, making it easier to understand how the application works and help in troubleshooting. One critical assumption is that introducing Spring Shell will not be enough without knowing what Spring can do. This includes learning about Spring's beans and dependency injection which makes development easier and enables more advanced command implementations. In addition, Spring Shell fosters good software engineering practices as it advocates for writing easily maintainable and testable command with high level of clarity and succinctness.

KALI_LINUX [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.15.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Sites
https://www.gstatic.com
https://www.recaptcha.net
https://www.paypalobjects.com
https://history.paypal.com
http://history.paypal.com
http://sandbox.paypal.com
http://status.paypal.com
History
Search
Alerts
Output
Spider
AJAX Spider
Active Scan

```
HTTP/1.1 400 Bad Request
Connection: keep-alive
Content-Length: 226
Accept-Ch: Sec-CH-UA-Full
Cache-Control: max-age=0, no-cache, no-store, must-revalidate
Content-Type: text/html; charset=iso-8859-1
Origin-Trial: AmF3SS0NWoXo3HaojgmIVVXavukRnZH597u+xZNXRCiKWzSKzfNPHw9NC32GmblY12+
HXpkCEYeYGyvRBNkkJg0AAABbeyJvcmlnaW4iOiJodHRwczovL3BheXBhbC5jb206NDQzIiwiZmVhdHVyZSI6IlRwY2QiLCJleHBpcnkiOjE3MzUzNDM5OTksImlzU3ViZG9tYWluI
jp0cnVlfQ==
Paypal-Debug-Id: f6336301edf90
Set-Cookie: l7_az=ccg01.phx; Path=/; Domain=paypal.com; Expires=Mon, 07 Oct 2024 16:39:40 GMT; HttpOnly; Secure
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
```

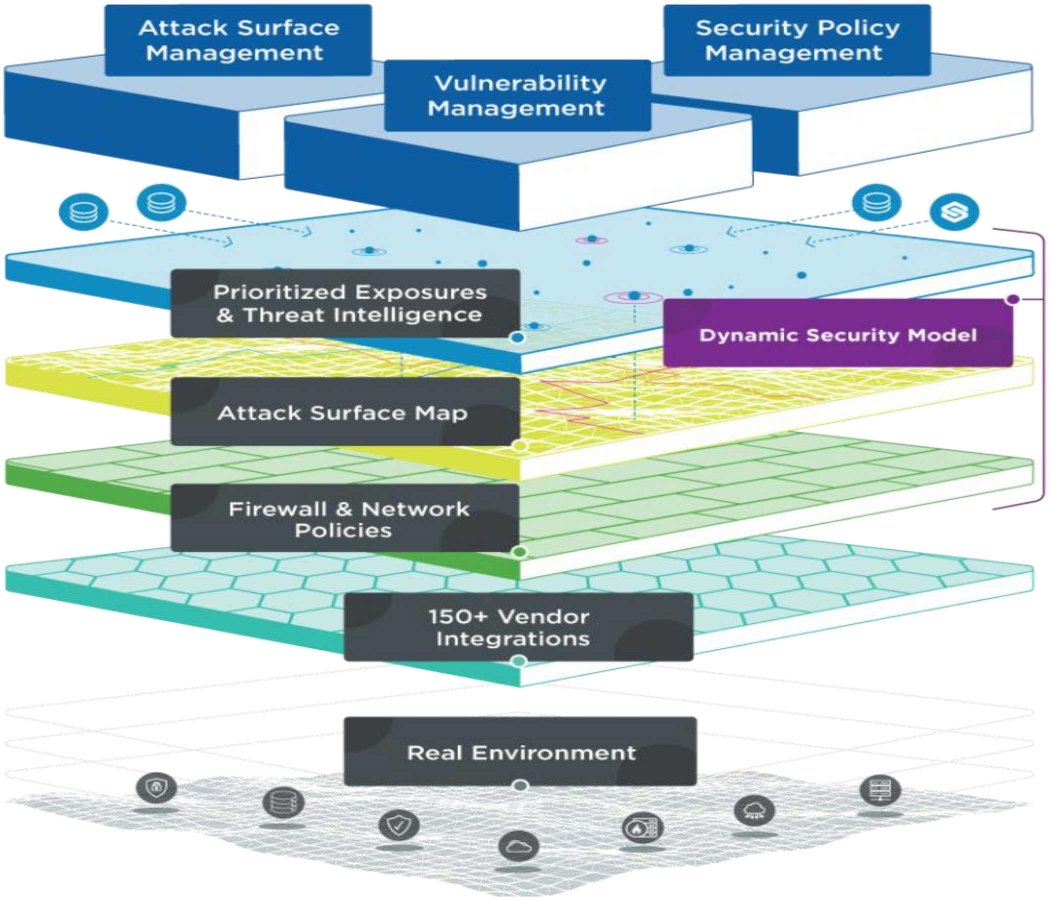
Alerts (31)
- PII Disclosure (57)
- Spring4Shell
- CSP: Wildcard Directive (324)
- CSP: script-src unsafe-eval (38)
- CSP: script-src unsafe-inline (2)
- CSP: style-src unsafe-inline (322)
- Content Security Policy (CSP) Header Not Set
- Cross-Domain Misconfiguration (73)
- Hidden File Found (12)
- Missing Anti-clickjacking Header (50)
- Vulnerable JS Library (2)
- CSP: Notices (320)
- Cookie No HttpOnly Flag (698)
- Cookie with SameSite Attribute None (141)

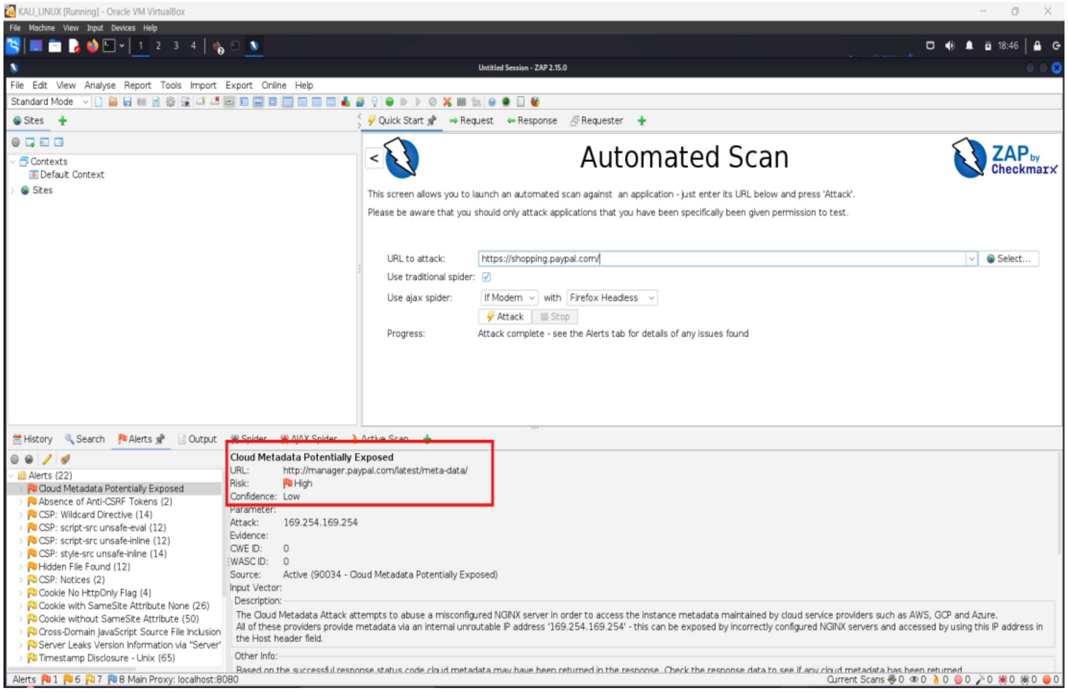
Spring4Shell

URL: http://history.paypal.com/sitemap.xml?class.module.classLoader.DefaultAssertionStatus=nonsense
Risk: High
Confidence: Medium
Parameter:
Attack: class.module.classLoader.DefaultAssertionStatus=nonsense
Evidence: HTTP/1.1 400 Bad Request
CWE ID: 78
WASC ID: 20
Source: Active (40045 - Spring4Shell)
Input Vector:
Description:
The application appears to be vulnerable to CVE-2022-22965 (otherwise known as Spring4Shell) - remote code execution (RCE) via data binding.
Other Info:

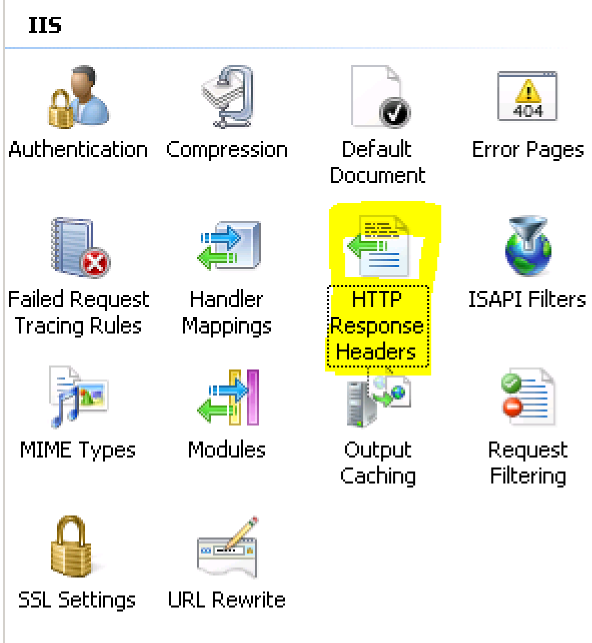
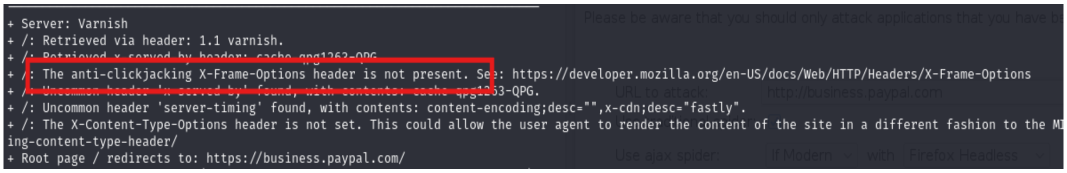
Alerts 2 9 11 9 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

7. Day 6

Date	2024/10/11
Summary Of the day’s activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 6 . I dedicated time to learning about data privacy and Cloud meta data potentially Exposed vulnerabilitiy.
Vulnerabilities discovered or explored	▪ Cloud meta data potentially Exposed.  Attack Surface Management Security Policy Management Vulnerability Management Prioritized Exposures & Threat Intelligence Dynamic Security Model Attack Surface Map Firewall & Network Policies 150+ Vendor Integrations Real Environment
Challenges faced and how they were overcome	▪ Limited time Resource - Prioritization of tasks. ▪ Access to resources and tools – Coloboration with Security Vendors.

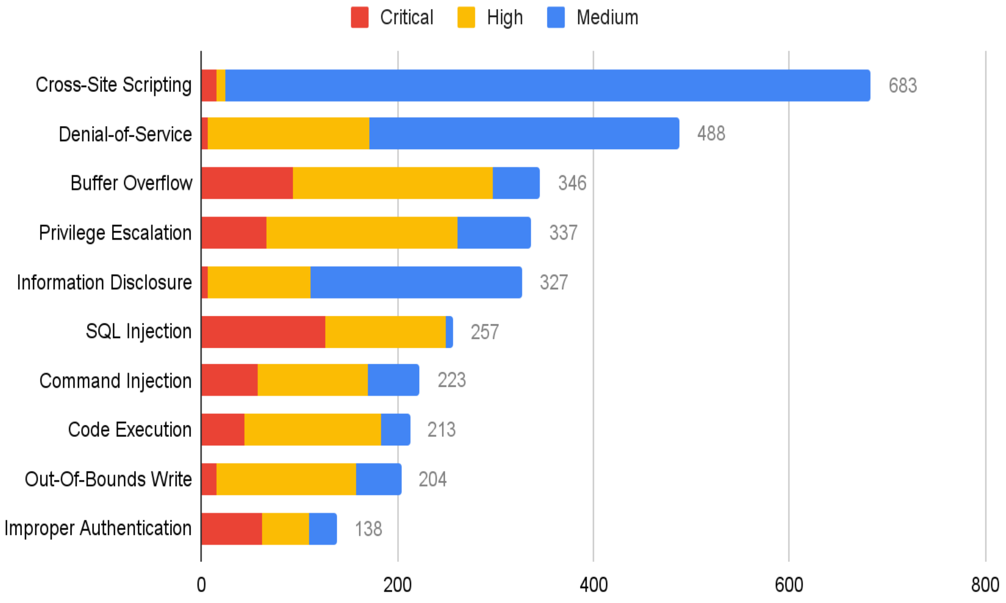
New tools,techniques or concepts learned	▪ OWASP ZAP  Untitled Session - ZAP 2.15.0 Automated Scan This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'. Please be aware that you should only attack applications that you have been specifically been given permission to test. URL to attack: https://shopping.paypal.com/ Use traditional spider: ☑ Use ajax spider: If Modern with Firefox Headless Attack Stop Progress: Attack complete - see the Alerts tab for details of any issues found Alerts (22) Cloud Metadata Potentially Exposed Absence of Anti-CSRF Tokens (2) CSP: Wildcard Directive (14) CSP: script-src unsafe-eval (12) CSP: script-src unsafe-inline (12) CSP: style-src unsafe-inline (14) Hidden File Found (12) CSP: Notices (2) Cookie No HttpOnly Flag (4) Cookie with SameSite Attribute None (26) Cookie without SameSite Attribute (50) Cross-Domain JavaScript Source File Inclusion Server Leaks Version Information via "Server" Timestamp Disclosure - Unix (65) Cloud Metadata Potentially Exposed URL: http://manager.paypal.com/latest/meta-data/ Risk: High Confidence: Low Parameter: Attack: 169.254.169.254 Evidence: CWE ID: 0 WASC ID: 0 Source: Active (90034 - Cloud Metadata Potentially Exposed) Input Vector: Description: The Cloud Metadata Attack attempts to abuse a misconfigured NGINX server in order to access the instance metadata maintained by cloud service providers such as AWS, GCP and Azure. All of these providers provide metadata via an internal unroutable IP address '169.254.169.254' - this can be exposed by incorrectly configured NGINX servers and accessed by using this IP address in the Host header field. Other Info: Based on the successful response status code cloud metadata may have been returned in the response. Check the response data to see if any cloud metadata has been returned. Alerts 1 6 7 8 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0
Reflections and takeaways	▪ The risk of security breaches increases dramatically with the unintentional exposure of cloud metadata, which often includes details on cloud resources, configurations, and user identities. One of the main critiques associated with the task is the understanding that a number of businesses do not pay considerable attention to whether any nd endpoints are secured or not and this is a weakness that can be easily exploited. This in turn could lead to the exposure of critical internal services to unauthorized parties, breaches resulting into loss of data as well as complete takeover of accounts and thus obsolete the rather limited oriented policies. The main point is that these services shall be well protected from every potential user except those that have been authorized to access the services of the cloud metadata management. It is advisable for the enterprises to put in place identity and access management (IAM) restrictions that limit access to their services as much as possible. So, effective monitoring and logging of the use of metadata endpoints will allow for detecting the signs of the occurrence of any breach and its trespassers’ real time tendencies.

8. Day 7

Date	2024/10/12
Summary Of the day's activities	• Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 7. I dedicated time to learning about data privacy and Anti clickjacking x – frame-option header not found vulnerabilitiy.
Vulnerabilities discovered or explored	▪ Anti clickjacking x – frame-option header not found.  IIS Authentication Compression Default Document Error Pages 404 Failed Request Tracing Rules Handler Mappings HTTP Response Headers ISAPI Filters MIME Types Modules Output Caching Request Filtering SSL Settings URL Rewrite
Challenges faced and how they were overcome	
New tools,techniques or concepts learned	▪ NIKTO  + Server: Varnish + /: Retrieved via header: 1.1 varnish. + /: Retrieved x-served-by header: cache-qpg1263-QPG. + /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options + /: Uncommon header 'x-served-by' found, with contents: cache-qpg1263-QPG. + /: Uncommon header 'server-timing' found, with contents: content-encoding;desc="",x-cdn;desc="fastly". + /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MI ing-content-type-header/ + Root page / redirects to: https://business.paypal.com/

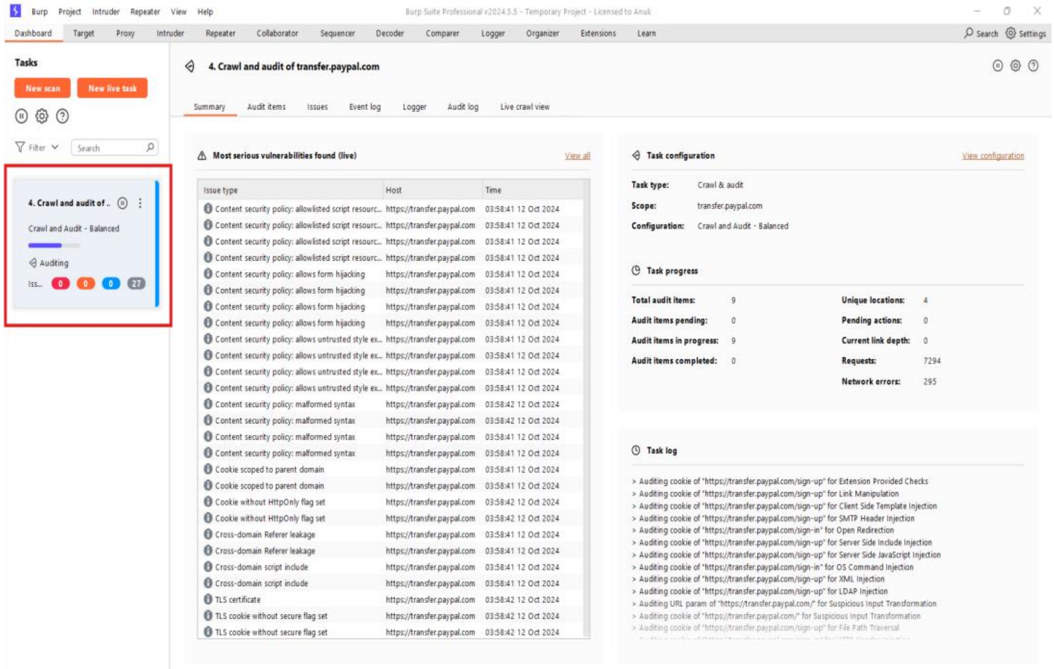
Reflections and takeaways	▪ The risk of clickjacking attacks is high in systems or applications where the X-Frame-Options security feature is not made available. Clickjacking refers to social engineering attacks where web users are tricked into clicking on objects on a webpage without their knowledge and effecting some action without their permission. One of the reflections is the realization that appropriate use of this security header is crucial such that web applications are not exposed to such threats. The X-Frame-Options header covers the scope of the clickjacking the ability to click on an embedded window that, although appears to be the top-level window, is not. ▪ There is an essential reminder that developers and security teams should always be in the contingency planning mode by making sure the X-Frame-Options header is provided for in the application security policies. Companies must require that all web applications feature this argument, along with suitable directives, such as DENY or SAMEORIGIN where applicable. Security audits and vulnerability scans overtime can assist in addressing security headers that are missing and the compliance to all best practices.

9. Day 8

Date	2024/10/13
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 8. I dedicated time to learning about data privacy and Input returned in response (Reflected Cross-Site Scripting (XSS))vulnerability.
Vulnerabilities discovered or explored	▪ Input returned in response (Reflected Cross-Site Scripting (XSS)) 
Challenges faced and how they were overcome	▪ Time Consuming – Use Automation tools.

New tools,techniques or concepts learned

- Burpsuite



4. Crawl and audit of transfer.paypal.com

Most serious vulnerabilities found (live)

Issue type	Host	Time
Content security policy: allowlisted script resourc...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allowlisted script resourc...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allowlisted script resourc...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allowlisted script resourc...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows form hijacking	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows form hijacking	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows form hijacking	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows form hijacking	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows untrusted style ex...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows untrusted style ex...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows untrusted style ex...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: allows untrusted style ex...	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: malformed syntax	https://transfer.paypal.com	03:58:42 12 Oct 2024
Content security policy: malformed syntax	https://transfer.paypal.com	03:58:42 12 Oct 2024
Content security policy: malformed syntax	https://transfer.paypal.com	03:58:41 12 Oct 2024
Content security policy: malformed syntax	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cookie scoped to parent domain	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cookie scoped to parent domain	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cookie without HttpOnly flag set	https://transfer.paypal.com	03:58:42 12 Oct 2024
Cookie without HttpOnly flag set	https://transfer.paypal.com	03:58:42 12 Oct 2024
Cross-domain Referer leakage	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cross-domain Referer leakage	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cross-domain script include	https://transfer.paypal.com	03:58:41 12 Oct 2024
Cross-domain script include	https://transfer.paypal.com	03:58:41 12 Oct 2024
TLS certificate	https://transfer.paypal.com	03:58:42 12 Oct 2024
TLS cookie without secure flag set	https://transfer.paypal.com	03:58:42 12 Oct 2024
TLS cookie without secure flag set	https://transfer.paypal.com	03:58:42 12 Oct 2024

Task configuration

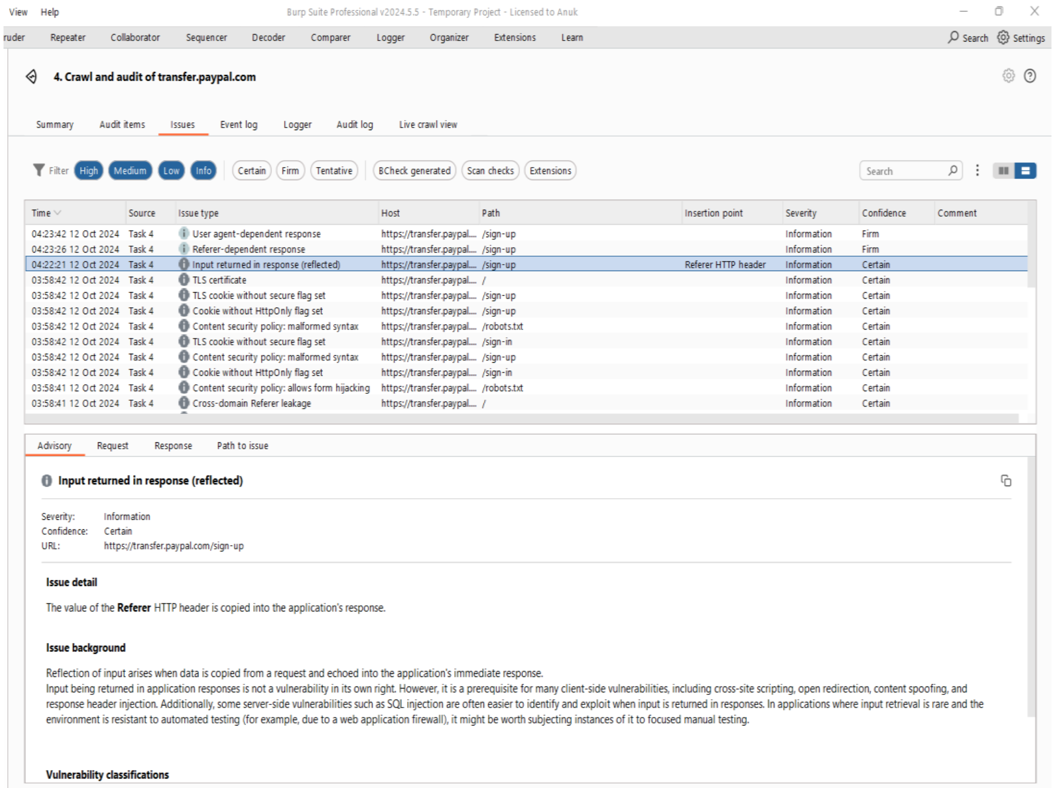
Task type: Crawl & audit
Scope: transfer.paypal.com
Configuration: Crawl and Audit - Balanced

Task progress

Total audit items:	9	**Unique locations:**	4
Audit items pending:	0	**Pending actions:**	0
Audit items in progress:	9	**Current link depth:**	0
Audit items completed:	0	**Requests:**	7294
		Network errors:	295

Task log

```
> Auditing cookie of "https://transfer.paypal.com/sign-up" for Extension Provided Checks
> Auditing cookie of "https://transfer.paypal.com/sign-up" for Link Manipulation
> Auditing cookie of "https://transfer.paypal.com/sign-up" for Client Side Template Injection
> Auditing cookie of "https://transfer.paypal.com/sign-up" for SMTP Header Injection
> Auditing cookie of "https://transfer.paypal.com/sign-in" for Open Redirection
> Auditing cookie of "https://transfer.paypal.com/sign-up" for Server Side Include Injection
> Auditing cookie of "https://transfer.paypal.com/sign-up" for Server Side JavaScript Injection
> Auditing cookie of "https://transfer.paypal.com/sign-in" for OS Command Injection
> Auditing cookie of "https://transfer.paypal.com/sign-up" for XML Injection
> Auditing cookie of "https://transfer.paypal.com/sign-up" for LDAP Injection
> Auditing URL param of "https://transfer.paypal.com/" for Suspicious Input Transformation
> Auditing cookie of "https://transfer.paypal.com/" for Suspicious Input Transformation
> Auditing cookie of "https://transfer.paypal.com/sign-up" for File Path Traversal
```



4. Crawl and audit of transfer.paypal.com

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment
04:23:42 12 Oct 2024	Task 4	User agent-dependent response	https://transfer.paypal....	/sign-up		Information	Firm	
04:23:26 12 Oct 2024	Task 4	Referer-dependent response	https://transfer.paypal....	/sign-up		Information	Firm	
04:22:21 12 Oct 2024	Task 4	Input returned in response (reflected)	https://transfer.paypal....	/sign-up	Referer HTTP header	Information	Certain	
03:58:42 12 Oct 2024	Task 4	TLS certificate	https://transfer.paypal....	/		Information	Certain	
03:58:42 12 Oct 2024	Task 4	TLS cookie without secure flag set	https://transfer.paypal....	/sign-up		Information	Certain	
03:58:42 12 Oct 2024	Task 4	Cookie without HttpOnly flag set	https://transfer.paypal....	/sign-up		Information	Certain	
03:58:42 12 Oct 2024	Task 4	Content security policy: malformed syntax	https://transfer.paypal....	/robots.txt		Information	Certain	
03:58:42 12 Oct 2024	Task 4	TLS cookie without secure flag set	https://transfer.paypal....	/sign-in		Information	Certain	
03:58:42 12 Oct 2024	Task 4	Content security policy: malformed syntax	https://transfer.paypal....	/sign-up		Information	Certain	
03:58:42 12 Oct 2024	Task 4	Cookie without HttpOnly flag set	https://transfer.paypal....	/sign-in		Information	Certain	
03:58:41 12 Oct 2024	Task 4	Content security policy: allows form hijacking	https://transfer.paypal....	/robots.txt		Information	Certain	
03:58:41 12 Oct 2024	Task 4	Cross-domain Referer leakage	https://transfer.paypal....	/		Information	Certain	

Input returned in response (reflected)

Severity: Information
Confidence: Certain
URL: https://transfer.paypal.com/sign-up

Issue detail

The value of the **Referer** HTTP header is copied into the application's response.

Issue background

Reflection of input arises when data is copied from a request and echoed into the application's immediate response.
Input being returned in application responses is not a vulnerability in its own right. However, it is a prerequisite for many client-side vulnerabilities, including cross-site scripting, open redirection, content spoofing, and response header injection. Additionally, some server-side vulnerabilities such as SQL injection are often easier to identify and exploit when input is returned in responses. In applications where input retrieval is rare and the environment is resistant to automated testing (for example, due to a web application firewall), it might be worth subjecting instances of it to focused manual testing.

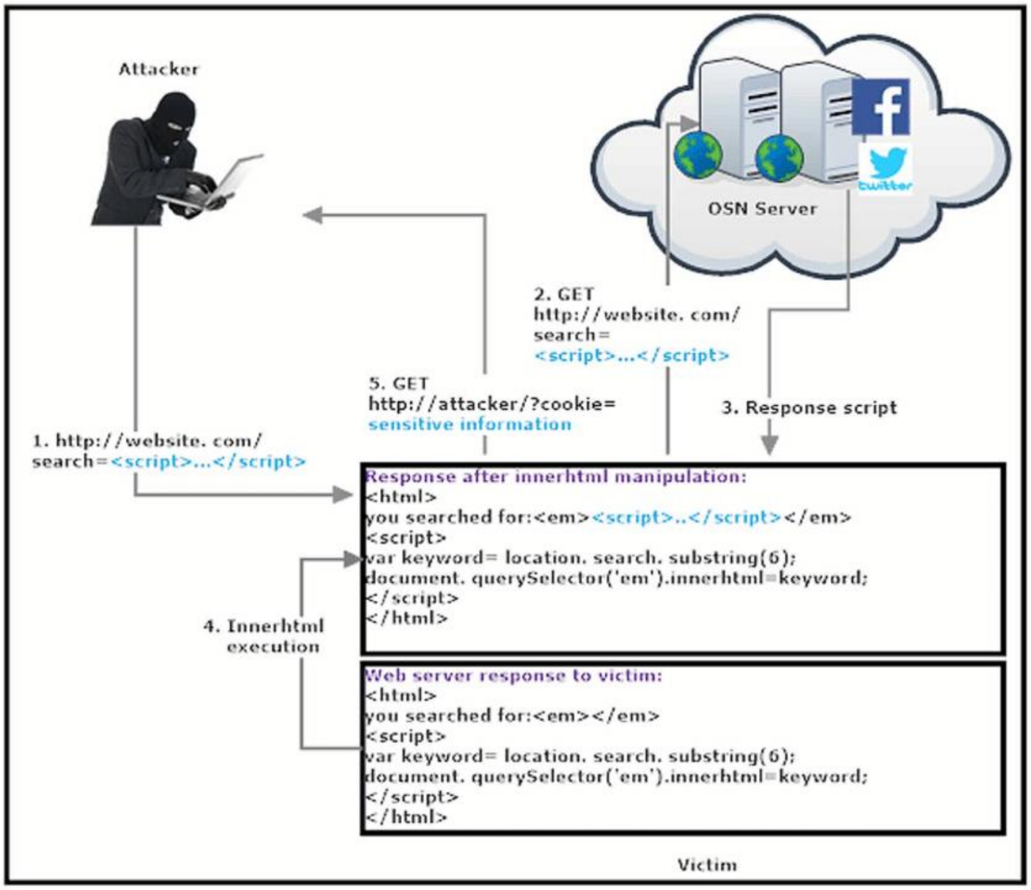
Vulnerability classifications

SLIIT
Discover Your Future

Reflections and takeaways	▪ Reflected Cross-Site Scripting (XSS) is a general web application security vulnerability that is caused when a web application returns any untrusted data in the response without validating or escaping it. One important reflection on this issue is to consider the fact that any user input no matter how trivial it is; provided that it is incorporated in the application, such as query parameters or form submissions, could be used to run a script in context of the user's session. This makes it imperative to implement proper input validation and output encoding techniques in order to prevent such attacks.

10. Day 9

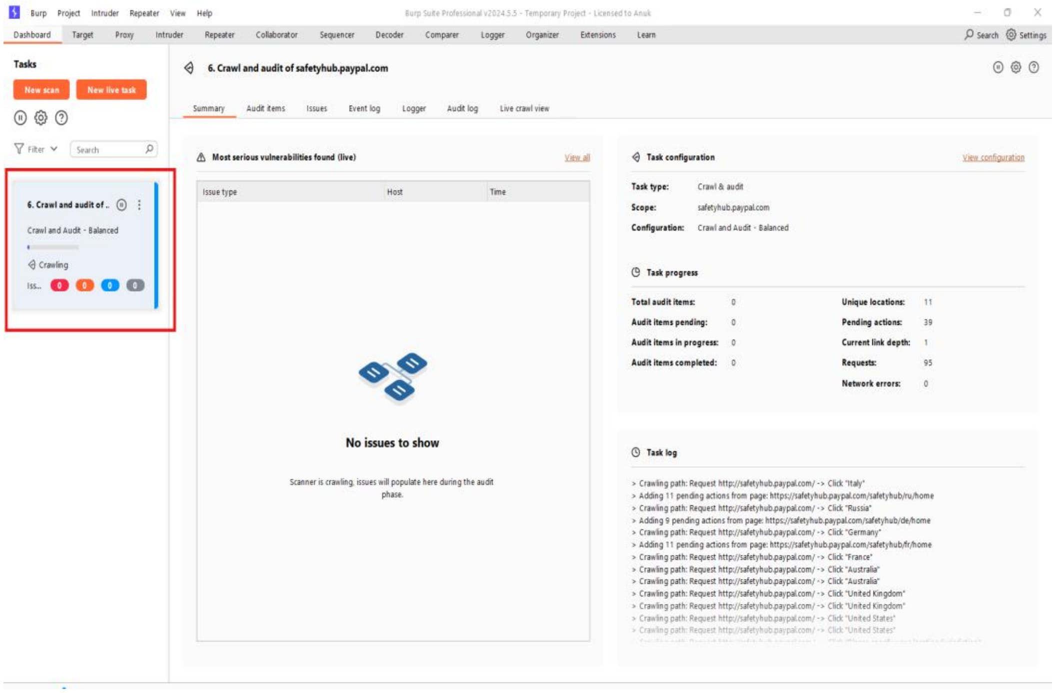
Date	2024/10/14
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 9. I dedicated time to learning about data privacy and vulnerability.

Vulnerabilities discovered or explored	▪ HTML 5 Storage Manipulation (DOM-based).
	
	 Attacker
	OSN Server
	twitter
	1. http://website. com/ search=<script>...</script>
	2. GET http://website. com/ search= <script>...</script>
	3. Response script
	4. Innerhtml execution
	5. GET http://attacker/?cookie= sensitive information
	Victim
	Response after innerhtml manipulation:
	<html>
	you searched for:<em><script>..</script></em>
	<script>
	var keyword= location. search. substring(6);
	document. querySelector('em').innerhtml=keyword;
	</script>
	</html>
	Web server response to victim:
	<html>
	you searched for:<em></em>
	<script>
	var keyword= location. search. substring(6);
	document. querySelector('em').innerhtml=keyword;
	</script>
	</html>

Challenges faced and how they were overcome	▪ It takes a long time to scan.

New tools,techniques or concepts learned

- Burpsuite


Burp Project Intruder Repeater View Help
Burp Suite Professional v2024.5.5 - Temporary Project - Licensed to Anuk
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Search Settings
Tasks
New scan
New live task
Filter
Search
6. Crawl and audit of ..
Crawl and Audit - Balanced
Crawling
Iss... 0 0 0 0
Summary Audit items Issues Event log Logger Audit log Live crawl view
Most serious vulnerabilities found (live)
View all
Issue type
Host
Time

6. Crawl and audit of safetyhub.paypal.com

No issues to show

Scanner is crawling, issues will populate here during the audit phase.

Task configuration

View configuration

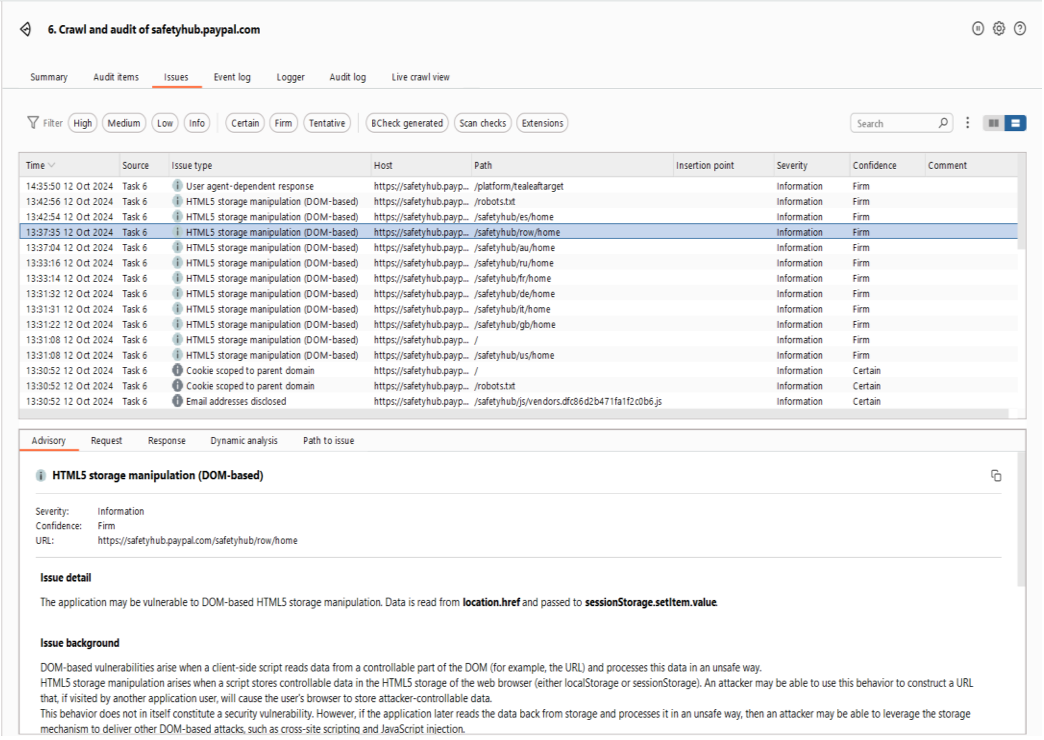
Task type: Crawl & audit
Scope: safetyhub.paypal.com
Configuration: Crawl and Audit - Balanced

Task progress

Total audit items:	0	Unique locations:	11
Audit items pending:	0	Pending actions:	39
Audit items in progress:	0	Current link depth:	1
Audit items completed:	0	Requests:	95
		Network errors:	0

Task log

```
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Italy"
> Adding 11 pending actions from page: https://safetyhub.paypal.com/safetyhub/ru/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Russia"
> Adding 9 pending actions from page: https://safetyhub.paypal.com/safetyhub/de/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Germany"
> Adding 11 pending actions from page: https://safetyhub.paypal.com/safetyhub/fr/home
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "France"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Australia"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "Australia"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United Kingdom"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United Kingdom"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United States"
> Crawling path: Request http://safetyhub.paypal.com/ -> Click "United States"
```


Summary Audit items Issues Event log Logger Audit log Live crawl view
Filter High Medium Low Info Certain Firm Tentative BCheck generated Scan checks Extensions
Search
Advisory Request Response Dynamic analysis Path to issue

6. Crawl and audit of safetyhub.paypal.com

Time	Source	Issue type	Host	Path	Insertion point	Severity	Confidence	Comment
14:35:50 12 Oct 2024	Task 6	User agent-dependent response	https://safetyhub.payp...	/platform/tealeaftarget		Information	Firm	
13:42:56 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/robots.txt		Information	Firm	
13:42:54 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/es/home		Information	Firm	
13:37:35 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/row/home		Information	Firm	
13:37:04 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/au/home		Information	Firm	
13:33:16 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/ru/home		Information	Firm	
13:33:14 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/fr/home		Information	Firm	
13:31:32 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/de/home		Information	Firm	
13:31:31 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/it/home		Information	Firm	
13:31:22 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/gb/home		Information	Firm	
13:31:08 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/		Information	Firm	
13:31:08 12 Oct 2024	Task 6	HTML5 storage manipulation (DOM-based)	https://safetyhub.payp...	/safetyhub/us/home		Information	Firm	
13:30:52 12 Oct 2024	Task 6	Cookie scoped to parent domain	https://safetyhub.payp...	/		Information	Certain	
13:30:52 12 Oct 2024	Task 6	Cookie scoped to parent domain	https://safetyhub.payp...	/robots.txt		Information	Certain	
13:30:52 12 Oct 2024	Task 6	Email addresses disclosed	https://safetyhub.payp...	/safetyhub/js/vendors.dfc86d2b471fa1f2c0b6.js		Information	Certain	

HTML5 storage manipulation (DOM-based)

Severity: Information
Confidence: Firm
URL: https://safetyhub.paypal.com/safetyhub/row/home

Issue detail

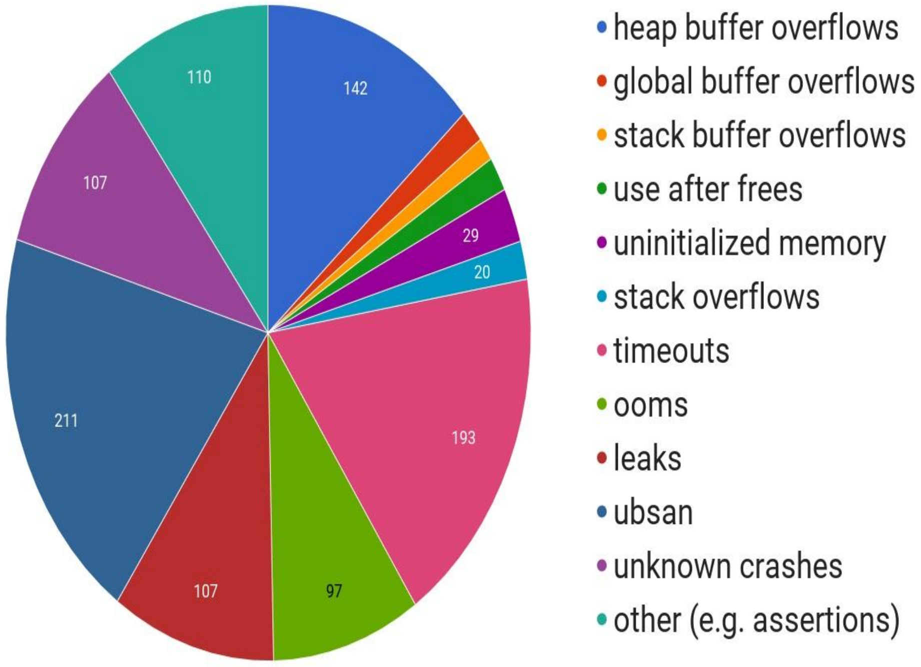
The application may be vulnerable to DOM-based HTML5 storage manipulation. Data is read from **location.href** and passed to **sessionStorage.setItem.value**.

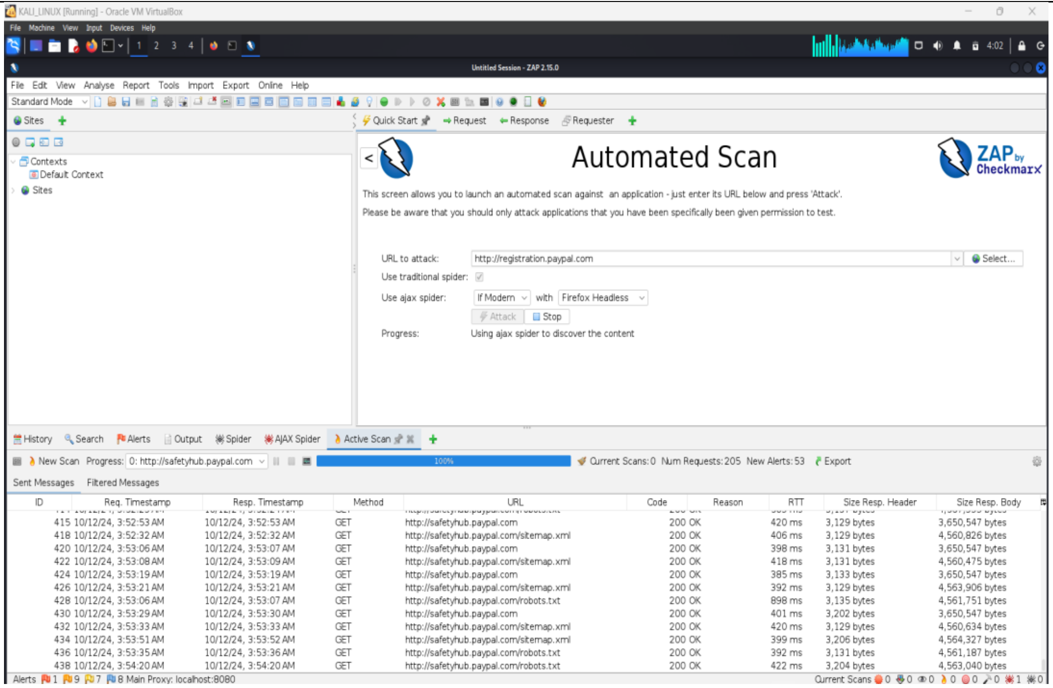
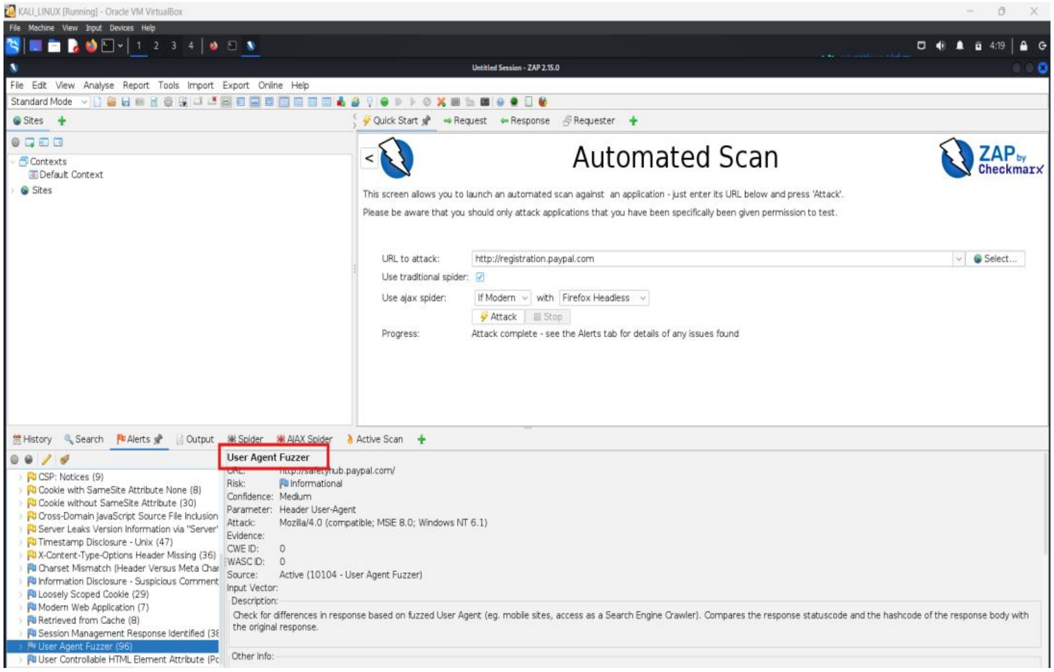
Issue background

DOM-based vulnerabilities arise when a client-side script reads data from a controllable part of the DOM (for example, the URL) and processes this data in an unsafe way.
HTML5 storage manipulation arises when a script stores controllable data in the HTML5 storage of the web browser (either localStorage or sessionStorage). An attacker may be able to use this behavior to construct a URL that, if visited by another application user, will cause the user's browser to store attacker-controllable data.
This behavior does not in itself constitute a security vulnerability. However, if the application later reads the data back from storage and processes it in an unsafe way, then an attacker may be able to leverage the storage mechanism to deliver other DOM-based attacks, such as cross-site scripting and JavaScript injection.

Reflections and takeaways	▪ HTML5 Storage Manipulation (DOM-based) vulnerabilities happen due to lack of adequate security measures in web applications that make use of web storage APIs like localStorage and sessionStorage to store user data. One prominent insight concerning the relayed concern is the perception that although HTML5 storage helps in storing, managing, and retrieving data efficiently on the client side, it can prove to be more harmful than beneficial when used without proper precautions. Users may be able to exploit such a flaw and change other data that is stored in the application that may lead to some undesired consequences such as retrieval of previously protected data.

11. Day 10

Date	2024/10/15
Summary Of the day's activities	▪ Today was a another productive day focused on enhancing my skills in cybersecurity and Bug Bounty Hunting day 10. I dedicated time to learning about data privacy and vulnerability.
Vulnerabilities discovered or explored	▪ User Agent Fuzzer. 
Challenges faced and how they were overcome	
New tools,techniques	▪ OWASP ZAP

or concepts learned	 

KALI_LINUX [Running] - Oracle VM VirtualBox

Untitled Session - ZAP 2.15.0

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://registration.paypal.com
Use traditional spider: ☑
Use ajax spider: If Modern with Firefox Headless
Progress: Using ajax spider to discover the content

New Scan Progress: 0: http://safetyhub.paypal.com 100% Current Scans: 0 Num Requests: 205 New Alerts: 53 Export

```
 ID  Req. Timestamp         Resp. Timestamp        Method  URL                                     Code  Reason  RTT     Size Resp. Header  Size Resp. Body
 415 10/12/24, 3:52:53 AM   10/12/24, 3:52:53 AM   GET     http://safetyhub.paypal.com             200   OK      420 ms  3,129 bytes        3,650,547 bytes
 418 10/12/24, 3:52:32 AM   10/12/24, 3:52:32 AM   GET     http://safetyhub.paypal.com/sitemap.xml 200   OK      406 ms  3,129 bytes        4,560,826 bytes
 420 10/12/24, 3:53:06 AM   10/12/24, 3:53:07 AM   GET     http://safetyhub.paypal.com             200   OK      398 ms  3,131 bytes        3,650,547 bytes
 422 10/12/24, 3:53:08 AM   10/12/24, 3:53:09 AM   GET     http://safetyhub.paypal.com/sitemap.xml 200   OK      418 ms  3,131 bytes        4,560,475 bytes
 424 10/12/24, 3:53:19 AM   10/12/24, 3:53:19 AM   GET     http://safetyhub.paypal.com             200   OK      385 ms  3,133 bytes        3,650,547 bytes
 426 10/12/24, 3:53:21 AM   10/12/24, 3:53:21 AM   GET     http://safetyhub.paypal.com/sitemap.xml 200   OK      392 ms  3,129 bytes        4,563,906 bytes
 428 10/12/24, 3:53:06 AM   10/12/24, 3:53:07 AM   GET     http://safetyhub.paypal.com/robots.txt  200   OK      898 ms  3,135 bytes        4,561,751 bytes
 430 10/12/24, 3:53:29 AM   10/12/24, 3:53:30 AM   GET     http://safetyhub.paypal.com             200   OK      401 ms  3,202 bytes        3,650,547 bytes
 432 10/12/24, 3:53:33 AM   10/12/24, 3:53:33 AM   GET     http://safetyhub.paypal.com/sitemap.xml 200   OK      420 ms  3,129 bytes        4,560,634 bytes
 434 10/12/24, 3:53:51 AM   10/12/24, 3:53:52 AM   GET     http://safetyhub.paypal.com/sitemap.xml 200   OK      399 ms  3,206 bytes        4,564,327 bytes
 436 10/12/24, 3:53:35 AM   10/12/24, 3:53:36 AM   GET     http://safetyhub.paypal.com/robots.txt  200   OK      392 ms  3,131 bytes        4,561,187 bytes
 438 10/12/24, 3:54:20 AM   10/12/24, 3:54:20 AM   GET     http://safetyhub.paypal.com/robots.txt  200   OK      422 ms  3,204 bytes        4,563,040 bytes
```

Alerts 1 9 7 8 Main Proxy: localhost:8080

KALI_LINUX [Running] - Oracle VM VirtualBox

Untitled Session - ZAP 2.15.0

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://registration.paypal.com
Use traditional spider: ☑
Use ajax spider: If Modern with Firefox Headless
Progress: Attack complete - see the Alerts tab for details of any issues found

- CSP: Notices (9)
- Cookie with SameSite Attribute None (8)
- Cookie without SameSite Attribute (30)
- Cross-Domain JavaScript Source File Inclusion
- Server Leaks Version Information via "Server"
- Timestamp Disclosure - Unix (47)
- X-Content-Type-Options Header Missing (36)
- Charset Mismatch (Header Versus Meta Char
- Information Disclosure - Suspicious Comment
- Loosely Scoped Cookie (29)
- Modern Web Application (7)
- Retrieved from Cache (8)
- Session Management Response Identified (38
- User Agent Fuzzer (96)
- User Controllable HTML Element Attribute (Pc

User Agent Fuzzer

URL: http://safetyhub.paypal.com/
Risk: Informational
Confidence: Medium
Parameter: Header User-Agent
Attack: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1)
Evidence:
CWE ID: 0
WASC ID: 0
Source: Active (10104 - User Agent Fuzzer)
Input Vector:
Description:
Check for differences in response based on fuzzed User Agent (eg. mobile sites, access as a Search Engine Crawler). Compares the response statuscode and the hashcode of the response body with the original response.
Other Info:

Reflections and takeaways	▪ The User Agent Fuzzer refers to a software testing tool for web applications' security, which allows for changing the user agent string of a browser. Implicit in the use of the User Agent Fuzzer, however, is that applications are capable of handling all sorts of user agents, as well as validating the input of users. Security testers doing this can detect issues like mishandling of requests, both of which are security concerns and may even cause the application to behave improperly. ▪ The important lesson here, is that user agent strings are useful tools not just in testing but can be employed for the security of the application. In this sense, there are ways in which the user agents can be used against the application and thus, the input validation mechanisms can be improved and application controls can be tightened. Furthermore, it highlights the need to understand user agent strings as a vector of attack for purposes that include but are not limited to impersonation and getting around security measures.

12. CONCLUSION

- The ten days of intensive learning and vulnerability exploration have significantly enhanced my understanding of web security. This journey underscored the importance of a proactive approach in identifying, analyzing, and mitigating security threats. By utilizing a combination of automated tools, collaboration, and strategic planning, I developed a comprehensive perspective on securing web applications. The insights gained have reinforced the need for a thorough, multifaceted approach to cybersecurity, one that blends technical expertise with a commitment to vigilance and adaptability. As I continue this path, I am equipped with a stronger foundation and a renewed resolve to tackle the challenges in the ever-evolving landscape of cybersecurity.