

***Sri Lanka Institute of Information Technology***  
***BSc Honors in Information Technology***  
***Specializing in Cyber Security***



**IE2062 - Web Security**

**Bug Bounty**  
**Journal Book**

<b>Student Register Number</b>	<b>Student Name</b>
IT 22083678	SAHAN H.P.T

## **ABSTRACT**

This journey provides a thorough record of my own bug bounty hunting journey. It goes into great detail about how to choose bug bounty programs, set up accounts on well-known sites like HackerOne, Bugcrowd and get through the difficult process of reporting vulnerabilities to participating companies. This journal attempts to give aspiring bug bounty hunters priceless insights and direction as they set out on their own odyssey in the exciting world of cybersecurity through open accounts of my experience, struggles and victories.

# Table of Contents

ABSTRACT .....	2
INTRODUCTION .....	4
Chapter 1: OWASP Top 10 Vulnerabilities .....	5
Chapter 2: Exploring Bug Bounty Programs.....	7
Chapter 3: Program Selection and Account Setup .....	8
Chapter 4: Selecting 10 Domains to make the report.....	9
Chapter 5: Reconnaissance on the Target.....	10
5.1 Sublist3r.....	10
5.2 Nuclei Scan.....	12
5.4 Dmitry Scan.....	13
5.5 SQLmap Scanner.....	14
5.6 Nmap .....	15
Chapter 6:Searching for vulnerabilities.....	17
6.1 Burp Suite .....	17
6.2 ZAP.....	19
6.3 Nikto .....	21
6.4 Netsparker.....	22
Challenges : .....	22
Reflections and takeaways: .....	23
Conclusion: .....	24
References: .....	25

# INTRODUCTION

Bug bounty programs are aa shining example of innovation in the rapidly changing field of cybersecurity. They are redefining old paradigms by utilizing the combined strength of talent from around the world to strengthen the digital defenses of businesses all over the world. These initiatives mark a paradigm shift by enabling people from all walks of life to actively participate in protecting digital ecosystems from new threats and by democratizing the process of vulnerability discovery and disclosure.

I could not avoid taking advantage of this exciting and dynamic field as a budding bug bounty hunter. I set out on a quest into the heart of bug bounty hunting with utmost commitment and limitless enthusiasm, driven by a passionate passion for cybersecurity and a perpetual hunger for knowledge. I was excited and felt a sense of purpose at the through of potentially solving hidden vulnerabilities, figuring out the workings of intricate system, and significantly improving the security of organizations.

This journal is proof of my steadfast dedication to the field of bug bounty hunting and my ever-ending quest for excellence in it. It documents my experience, ranging from the thrilling peaks of success vulnerability discoveries to the humiliating setbacks faced during the process. This journal is more than just a personal story, it serves as a source of wisdom and motivation for other enthusiasts stepping into the exciting world of bug bounty hunting. It provides prospective bug bounty hunters with a road map, compass, and guiding light to help them traverse the unknown waters of cybersecurity. This gives them the confidence, bravery, and conviction to set out on their own adventure.

# **Chapter 1: OWASP Top 10 Vulnerabilities**

The Open Web Application Security Project (OWASP) Top 10 vulnerabilities serve as a critical guide for understanding the most prevalent security risks faced by web applications today. In this section we delve into each vulnerability listed in the OWASP Top 10 providing insights into their nature impact and methods of exploitation.

## **1. Broken Access Control:**

Broken access control occurs when restrictions on what authenticated users are allowed to do are not properly enforced. This vulnerability can lead to unauthorized access to sensitive data privilege escalation and other security breaches. Common examples include insecure direct object references missing function level access controls and insufficiently protected APIs.

## **2. Cryptographic Failures:**

Cryptographic failures encompass weaknesses in the implementation of cryptographic mechanisms such as encryption hashing and digital signatures. These vulnerabilities can result in data exposure integrity violations and unauthorized tampering. Common cryptographic failures include the use of weak algorithms improper key management and insufficient entropy generation.

## **3. Injection:**

Injection vulnerabilities arise when untrusted data is inserted into a program or command leading to unintended execution of malicious code. The most common type of injection vulnerability is SQL injection where attackers manipulate SQL queries to gain unauthorized access to databases. Other types include command injection LDAP injection and XML injection.

## **4. Insecure Design:**

Insecure design vulnerabilities stem from flaws in the architectural and design decisions of web applications. These vulnerabilities can manifest in various forms including excessive trust in user input lack of proper security controls and failure to anticipate potential attack vectors.

## **5. Security Misconfiguration:**

Security misconfiguration occurs when security settings are not properly configured leaving systems vulnerable to exploitation. This vulnerability can result from default configurations unnecessary features enabled or inadequate hardening measures. Examples include open ports default passwords and unused services running on servers.

## **6. Vulnerable and Outdated Components:**

Vulnerabilities in third party libraries frameworks and components pose significant risks to web applications. Attackers can exploit known vulnerabilities in outdated or insecure components to compromise the security of the entire system. Regularly updating and patching components is essential to mitigate this risk.

## **7. Identification and Authentication Failures:**

Identification and authentication failures occur when mechanisms for verifying the identity of users are inadequate or improperly implemented. Weak authentication methods such as weak passwords or lack of multi factor authentication can lead to unauthorized access and account compromise.

## **8. Software and Data Integrity Failures:**

Software and data integrity failures occur when systems are unable to ensure the integrity and authenticity of data throughout its lifecycle. Without proper integrity controls attackers can modify delete or inject malicious data leading to data corruption fraud and other security incidents.

## **9. Security Logging and Monitoring Failures:**

Security logging and monitoring failures occur when systems fail to adequately log security events or lack effective monitoring capabilities. Without comprehensive logging and monitoring organizations may miss critical indicators of compromise making it difficult to detect and respond to security incidents in a timely manner.

## **10. Server-side Request Forgery (SSRF):**

Server side request forgery (SSRF) vulnerabilities allow attackers to manipulate server side requests initiated by the web application. By exploiting SSRF vulnerabilities attackers can access internal systems bypass access controls and perform various malicious actions. Common exploitation techniques include fetching sensitive data executing arbitrary commands and conducting port scanning.

## **Chapter 2: Exploring Bug Bounty Programs**

I devoted my second day of the bug bounty journey to researching the wide range of bug bounty programs offered on websites like HackerOne and Bugcrowd. This stage of research was essential because it helped me find programs that matched my interests, abilities, and areas of expertise, which set the groundwork for my bug bounty hunting activities.

I started navigation by becoming fully immersed in the rich ecosystem of bug bounty sites like HackerOne and Bugcrowd. These platforms function as focal points for enterprise aiming to interact with the worldwide security community for the purpose detecting and addressing security weaknesses in their digital assets.

As I browsed through HackerOne's program listings, I was astounded by the enormous variety of options available, from Fortune 500 companies to startups in sectors like technology, finance, healthcare, and more. Every program featured a distinct set of targets varying from mobile apps and IoT devices to web applications and APIs, offering a multitude of opportunities for bug hunters with different skill sets.

In a similar vein, I was pleased with Bugcrowd's extensive and comprehensive program offerings. The Bugcrowd platform featured a diverse range of bug bounty programs from well-established companies to recently established startups, spanning numerous industries and sectors.

As I discovered further into my research, I carefully examined the objectives, benefits, and guidelines for every bug bounty program. Knowing a program's scope is essential because it defines the features and weaknesses that can be tested, allowing bug hunters to concentrate their efforts on areas where they can have the biggest impact.

## Chapter 3: Program Selection and Account Setup

I did a lot of research on the various bug bounty platforms and programs before starting my bug bounty journey. Realizing that the caller and range of programs available would determine how successful my bug hunting endeavors would be, I took the time to investigate various platforms and assess what they had to offer. I conducted research by going on the websites of well-known bug bounty services like HackerOne and Bugcrowd. Within the cybersecurity community, these platforms are well-known for their strong program structures, varied clientele, and vibrant security researcher community.

I chose to focus my efforts on HackerOne due to their strong reputations, extensive program offerings, and active researcher communities.

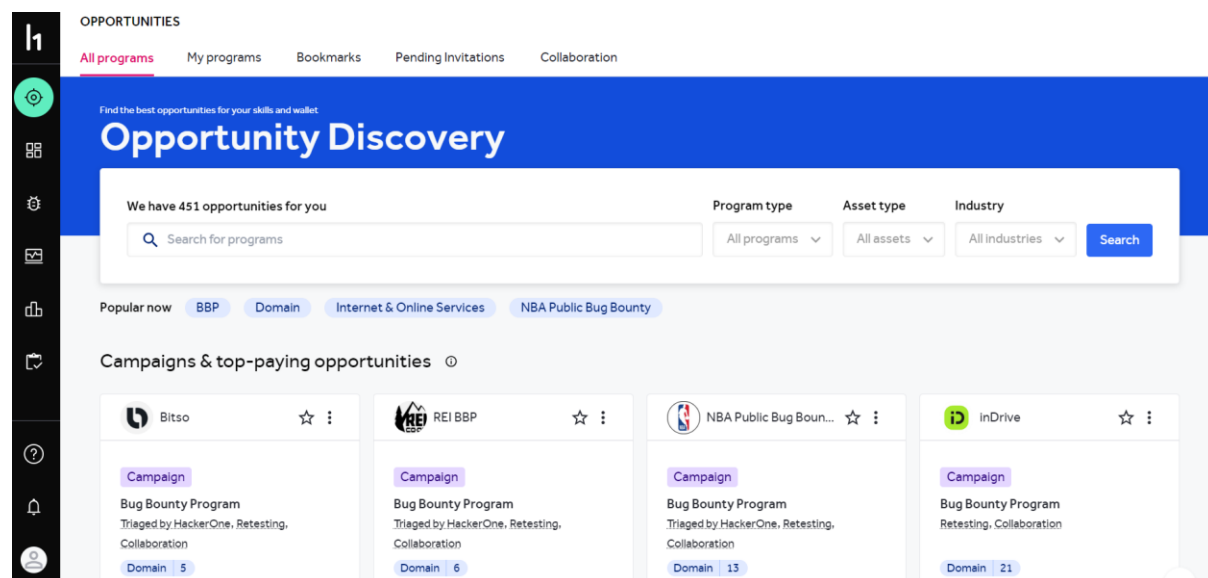


Figure 1: Hackerone Interface

After initially exploring bug bounty, my next step was to understand bug bounty programs and their guidelines.



## Chapter 4: Selecting 10 Domains to make the report.

For the assignment, I had to perform bug bounty scans on 10 selected domains. The process of choosing the final domains was a bit challenging, as I went through numerous domains and subdomains before narrowing it down to the final 10. After hours and hours of scans I selected the my final 10 domains.

Domain	Sub Domain
<a href="https://www.transunion.com">https://www.transunion.com</a>	<a href="https://go.transunion.com">go.transunion.com</a>
<a href="https://www.moov.io">https://www.moov.io</a>	<a href="https://slack.moov.io">slack.moov.io</a>
<a href="https://www.grammarly.com">https://www.grammarly.com</a>	<a href="https://app.grammarly.com">app.grammarly.com</a>
<a href="https://www.vendasta.com">https://www.vendasta.com</a>	<a href="https://blog.vendasta.com">blog.vendasta.com.</a>
<a href="https://www.truist.com">https://www.truist.com</a>	<a href="https://developer.truist.com">developer.truist.com</a>
<a href="https://www.apnic.net">https://www.apnic.net</a>	<a href="https://apply.apnic.net">apply.apnic.net</a>
<a href="https://www.cbre.com">https://www.cbre.com</a>	<a href="https://brand.cbre.com">brand.cbre.com</a>
<a href="https://www.fastly.com">https://www.fastly.com</a>	<a href="https://cfg.fastly.com">cfg.fastly.com</a>
<a href="https://www.booking.com">https://www.booking.com</a>	<a href="https://blog.booking.com">blog.booking.com</a>
<a href="https://www.render.com">https://www.render.com</a>	<a href="https://onrender.com">onrender.com</a>

## Chapter 5: Reconnaissance on the Target

Over the period of the following three days, I took part in the thrilling process of targeted testing on the bug bounty programs that I had carefully chosen. I set out to find vulnerabilities hidden in the digital ecosystems of companies taking part in bug bounty programs, using a wide range of testing methods, both automated and manual.

I carefully examined the program scope and identified potential attack surfaces that were ready for investigation in order to approach each bug bounty program with utmost attention to detail. Equipped with this understanding, I created a strategic testing plan that was customized to meet the specific needs of every program, making use of both automated tools and manual testing methodologies to increase my productivity.

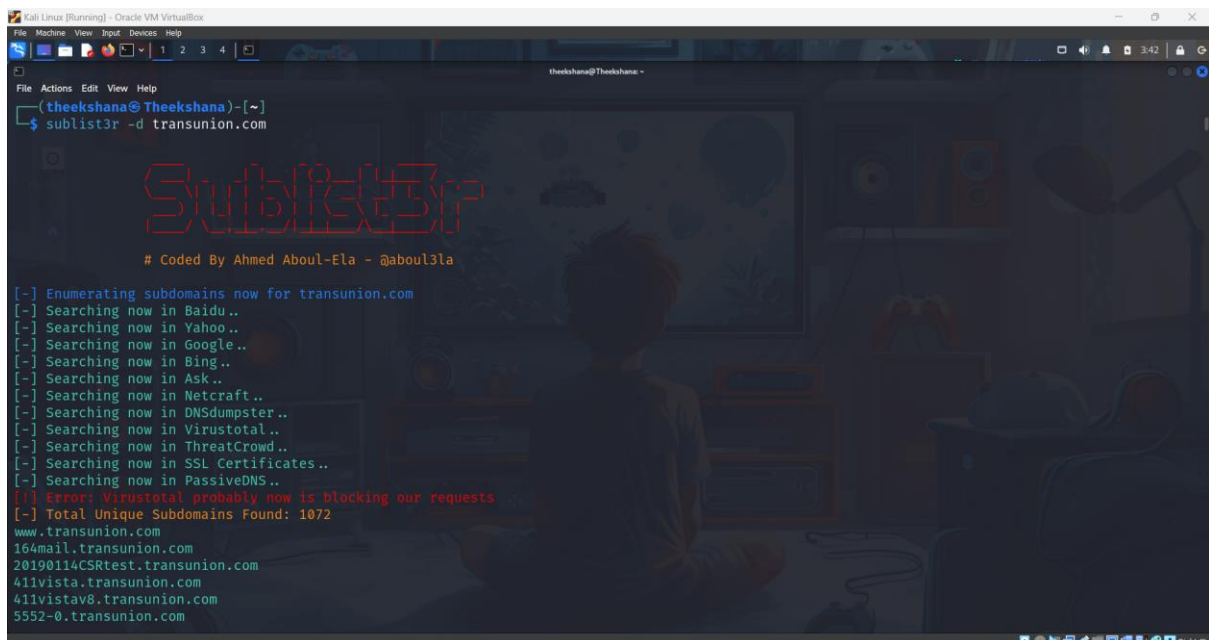
In one instance, I targeted a bug bounty program, focusing my efforts on its web application and API endpoints. To increase my testing coverage and quicken the discovery process, I wisely used automated tools to supplement my manual testing efforts. I performed thorough vulnerability scans using tool OWASP ZAP, and Nmap. These allowed me to find common security flaws like cross-site scripting (XSS).

### 5.1 Sublist3r

For subdomain enumeration, Sublist3r is a popular open-source tool. Its goal is to make it easier for penetration testers and security researchers to find subdomains connected to a given domain. Sublist3r finds subdomains by searching several search engines and other web resources. It is useful for scoping for potential routes of entry into a system or network.



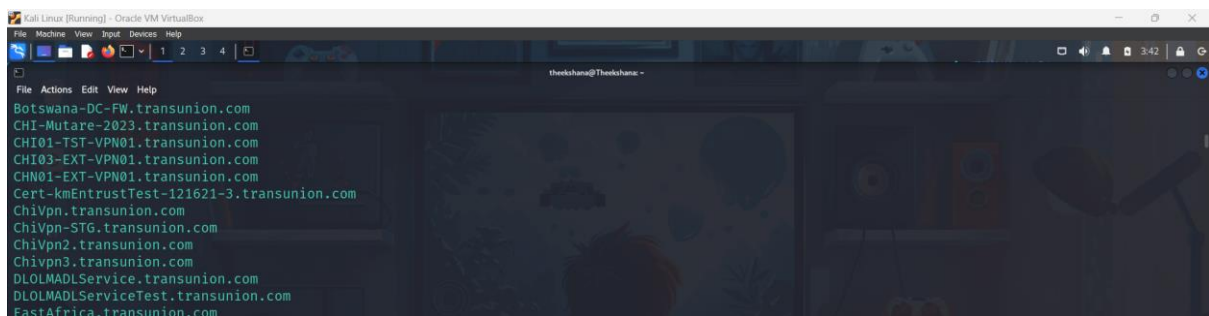
Using sublist3r tool scan the subdomains within the domain. I used Sublist3r which is a scanning tool to scan for subdomains within the domain [transunion.com](https://transunion.com) through Kali Linux. It scanned all the available subdomains Some of them are listed below.



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
theekshana@Theekshana: ~
(theekshana@Theekshana)-[~]
$ sublist3r -d transunion.com

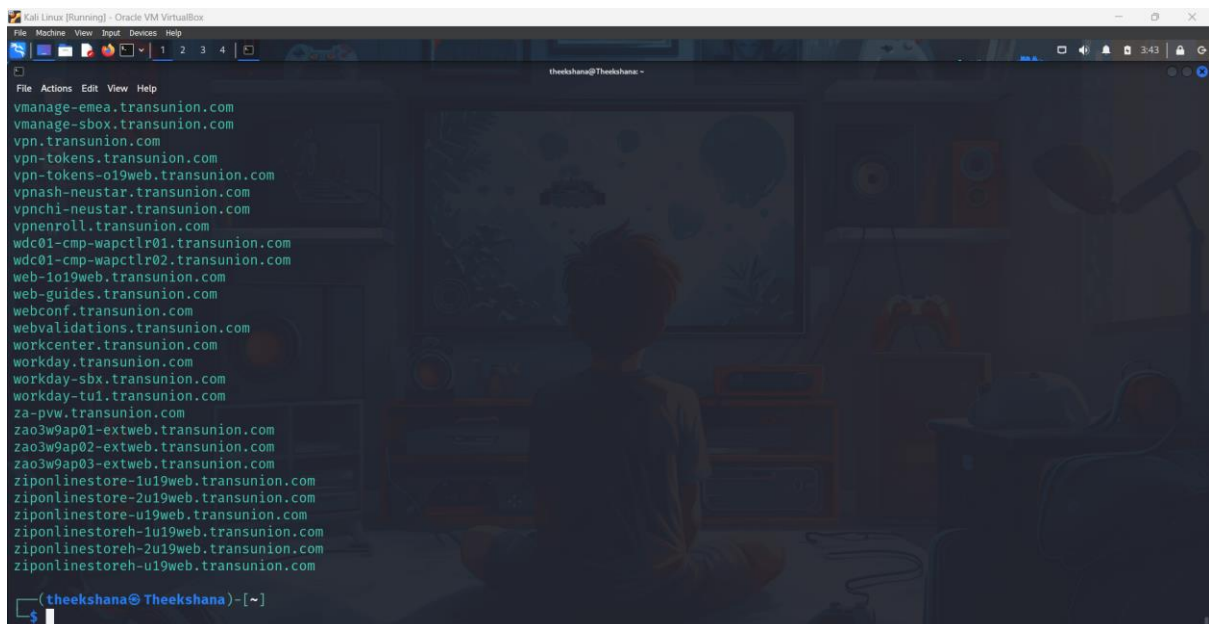
Sublist3r
# Coded By Ahmed Aboul-Ela - @aboul3la

[~] Enumerating subdomains now for transunion.com
[~] Searching now in Baidu..
[~] Searching now in Yahoo..
[~] Searching now in Google..
[~] Searching now in Bing..
[~] Searching now in Ask..
[~] Searching now in Netcraft..
[~] Searching now in DNSdumpster..
[~] Searching now in Virustotal..
[~] Searching now in ThreatCrowd..
[~] Searching now in SSL Certificates..
[~] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[~] Total Unique Subdomains Found: 1072
www.transunion.com
164mail.transunion.com
20190114CSRtest.transunion.com
411vista.transunion.com
411vistav8.transunion.com
5552-0.transunion.com
```



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
theekshana@Theekshana: ~
(theekshana@Theekshana)-[~]
$ sublist3r -d transunion.com

Botswana-DC-FW.transunion.com
CHI-Mutare-2023.transunion.com
CHI01-TST-VPN01.transunion.com
CHI03-EXT-VPN01.transunion.com
CHN01-EXT-VPN01.transunion.com
Cert-kmEntrustTest-121621-3.transunion.com
ChiVpn.transunion.com
ChiVpn-STG.transunion.com
ChiVpn2.transunion.com
ChiVpn3.transunion.com
DL0LMADLService.transunion.com
DL0LMADLServiceTest.transunion.com
FastAfrica.transunion.com
```



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
theekshana@Theekshana: ~
(theekshana@Theekshana)-[~]
$ sublist3r -d transunion.com

vmanage-emea.transunion.com
vmanage-sbox.transunion.com
vpn.transunion.com
vpn-tokens.transunion.com
vpn-tokens-o19web.transunion.com
vpnnash-neustar.transunion.com
vpnnchi-neustar.transunion.com
vpnnroll.transunion.com
wdc01-cmp-wapctrl01.transunion.com
wdc01-cmp-wapctrl02.transunion.com
web-1o19web.transunion.com
web-guides.transunion.com
webconf.transunion.com
webvalidations.transunion.com
workcenter.transunion.com
workday.transunion.com
workday-sbx.transunion.com
workday-tu1.transunion.com
za-pvw.transunion.com
zao3w9ap01-extweb.transunion.com
zao3w9ap02-extweb.transunion.com
zao3w9ap03-extweb.transunion.com
ziponlinestore-1u19web.transunion.com
ziponlinestore-2u19web.transunion.com
ziponlinestore-u19web.transunion.com
ziponlinestoreh-1u19web.transunion.com
ziponlinestoreh-2u19web.transunion.com
ziponlinestoreh-u19web.transunion.com

(theekshana@Theekshana)-[~]
$
```

## 5.2 Nuclei Scan

A screenshot of a Kali Linux virtual machine running Oracle VM VirtualBox. The terminal window shows the user 'theekshana' at the prompt '~'. They have installed 'nuclei' and are now running it. The output displays the ASCII art logo for nuclei v3.3.4, the projectdiscovery.io website, and several informational messages: current version is v3.3.4 (outdated), current templates version is v10.0.2 (latest), scan results upload to cloud is disabled, new templates added in latest release are 68, templates loaded for current scan are 8654, and they are executing 8455 signed templates from projectdiscovery/nuclei-templates. A warning message states 'Loading 199 unsigned templates for scan. Use with caution.' Finally, it reports 'No results found. Better luck next time!'. The background of the terminal is a dark-themed desktop environment with various icons and a sidebar on the right containing 'Mouse integration ...' and 'Auto capture keyboard ...'.

```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

(theekshana@Theekshana)-[~]
$ nuclei

      _____
     /          \
    /             \
   /               \
  /                 \
 /                   \
/                     \
\                     /
 \                   /
  \                 /
   \               /
    \             /
     \          /
      \_____/

v3.3.4

projectdiscovery.io

[INF] Current nuclei version: v3.3.4 (outdated)
[INF] Current nuclei-templates version: v10.0.2 (latest)
[WRN] Scan results upload to cloud is disabled.
[INF] New templates added in latest release: 68
[INF] Templates loaded for current scan: 8654
[INF] Executing 8455 signed templates from projectdiscovery/nuclei-templates
[WRN] Loading 199 unsigned templates for scan. Use with caution.
[INF] No results found. Better luck next time!

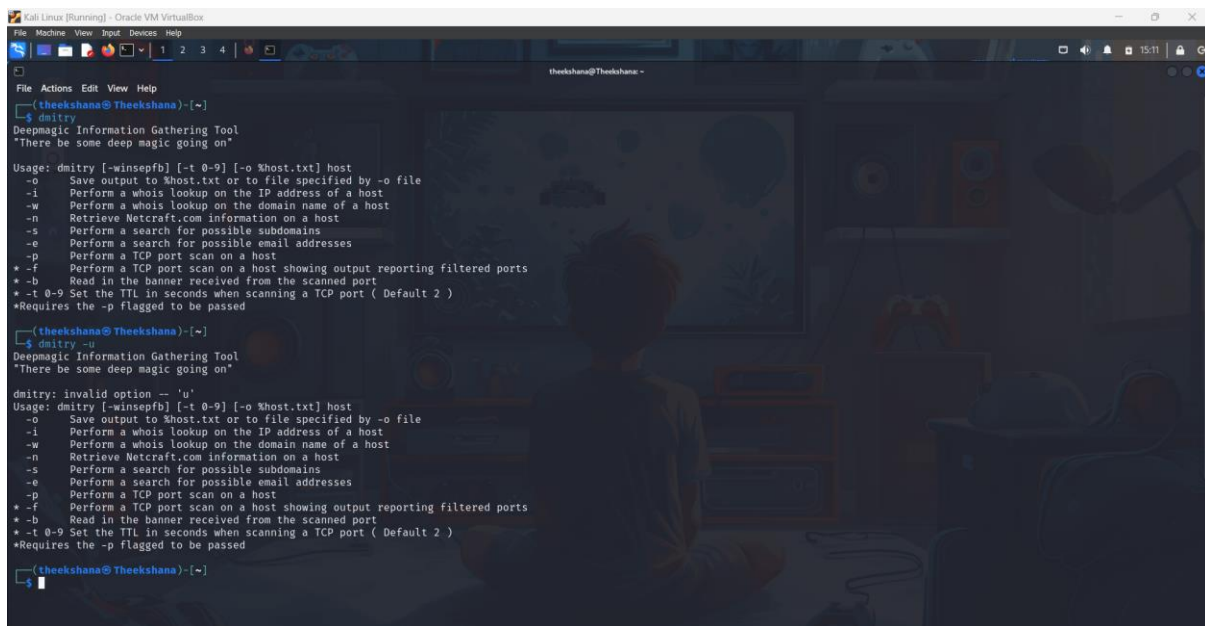
(theekshana@Theekshana)-[~]
$
```

Nuclei is an open-source, fast, and flexible vulnerability scanner developed by ProjectDiscovery. It is designed to automate the detection of security issues across various infrastructures, including web applications, APIs, cloud systems, and network environments. Nuclei uses customizable, YAML-based templates to identify vulnerabilities such as misconfigurations, outdated software, and known CVEs (Common Vulnerabilities and Exposures).

The key feature of Nuclei is its template-based scanning. Users can leverage thousands of pre-built templates maintained by the community, or they can create custom templates tailored to their unique security needs.

These templates cover a wide range of checks, including information disclosure, insecure server configurations, SSL/TLS weaknesses, outdated software versions, and critical vulnerabilities like SQL Injection, Cross-Site Scripting (XSS), and Remote Code Execution (RCE).

## 5.4 Dmitry Scan

A screenshot of a terminal window titled 'Kali Linux [Running] - Oracle VM VirtualBox'. The terminal shows the command 'dmitry' being executed. The output displays the tool's usage instructions, which include options for saving output to a file, performing whois lookups on IP addresses or domain names, retrieving Netcraft.com information, searching for subdomains and email addresses, and performing TCP port scans. The terminal also shows an error message 'dmitry: invalid option -- 'u'' when an invalid option is provided.

```
(theekshana@Theekshana)~$ dmitry
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o Xhost.txt] host
-o Save output to Xhost.txt or to file specified by -o file
-i Perform a whois lookup on the IP address of a host
-w Perform a whois lookup on the domain name of a host
-n Retrieve Netcraft.com information on a host
-s Perform a search for possible subdomains
-e Perform a search for possible email addresses
-p Perform a TCP port scan on a host
-f Perform a TCP port scan on a host showing output reporting filtered ports
-b Read in the banner received from the scanned port
-t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed

(theekshana@Theekshana)~$ dmitry -u
dmitry: invalid option -- 'u'
Usage: dmitry [-winsepfb] [-t 0-9] [-o Xhost.txt] host
-o Save output to Xhost.txt or to file specified by -o file
-i Perform a whois lookup on the IP address of a host
-w Perform a whois lookup on the domain name of a host
-n Retrieve Netcraft.com information on a host
-s Perform a search for possible subdomains
-e Perform a search for possible email addresses
-p Perform a TCP port scan on a host
-f Perform a TCP port scan on a host showing output reporting filtered ports
-b Read in the banner received from the scanned port
-t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed

(theekshana@Theekshana)~$
```

Dmitry (Deepmagic Information Gathering Tool) is an open-source command-line tool designed to assist security professionals and penetration testers in performing reconnaissance on a target host. Its primary focus is on gathering as much information as possible about the target during the initial phases of a security assessment, often referred to as the "footprinting" stage.

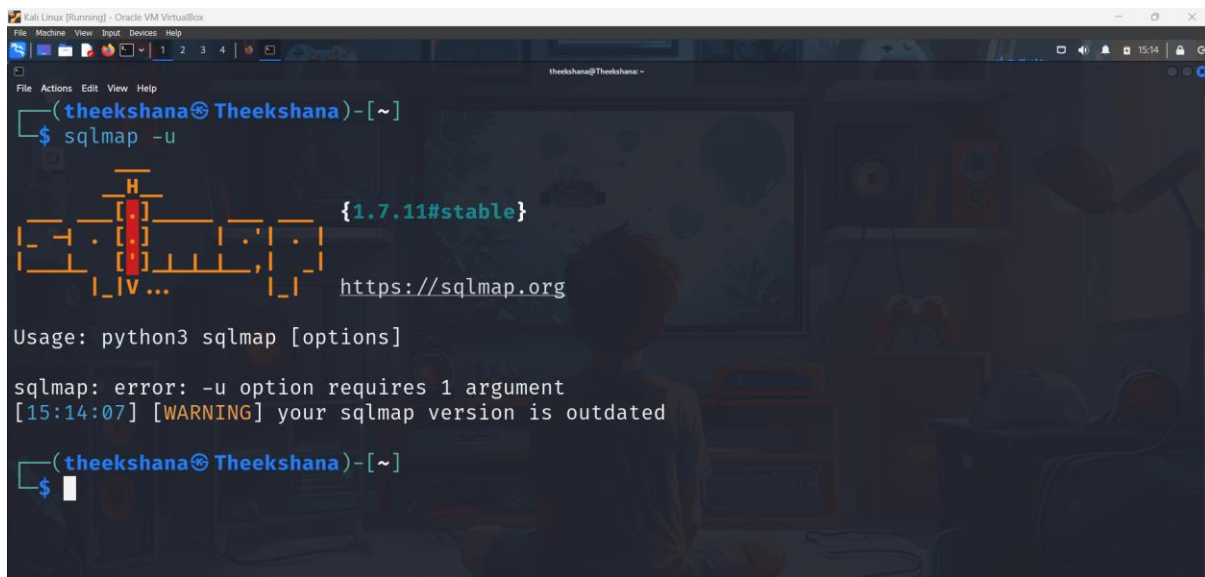
Dmitry operates by performing several different types of scans to collect various types of data about the target. It can gather basic details such as a host's domain name, IP address, and whois information.

### Dmitry include:

- **Domain Whois Lookups:** Dmitry can perform domain whois lookups to gather registration information about the target domain, which includes data like the domain owner, registration date, and expiration date.
- **IP Address Whois Lookups:** The tool can perform IP whois lookups to gather information about the IP addresses associated with the target domain.
- **Subdomain Detection:** Dmitry can attempt to discover subdomains that belong to the target, which can help expand the scope of the testing.
- **Email Address Harvesting:** The tool can pull email addresses associated with the target domain, useful for identifying points of contact or potential phishing vectors.
- **Port Scanning:** Dmitry can scan the target for open ports to identify which services are running and potentially vulnerable.



## 5.5 SQLmap Scanner



```
Kali Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
theekshana@Theekshana: ~
(theekshana@Theekshana)~$ sqlmap -u
{1.7.11#stable}
https://sqlmap.org
Usage: python3 sqlmap [options]
sqlmap: error: -u option requires 1 argument
[15:14:07] [WARNING] your sqlmap version is outdated
(theekshana@Theekshana)~$
```

SQLmap is an open-source penetration testing tool that automates the detection and exploitation of SQL injection vulnerabilities in web applications. SQL injection (SQLi) is a critical web vulnerability that allows attackers to interfere with a web application's queries to its database, potentially leading to unauthorized access, data exfiltration, or even full control over the database server.

SQLmap is widely regarded for its efficiency and versatility in discovering and exploiting SQL injection flaws.

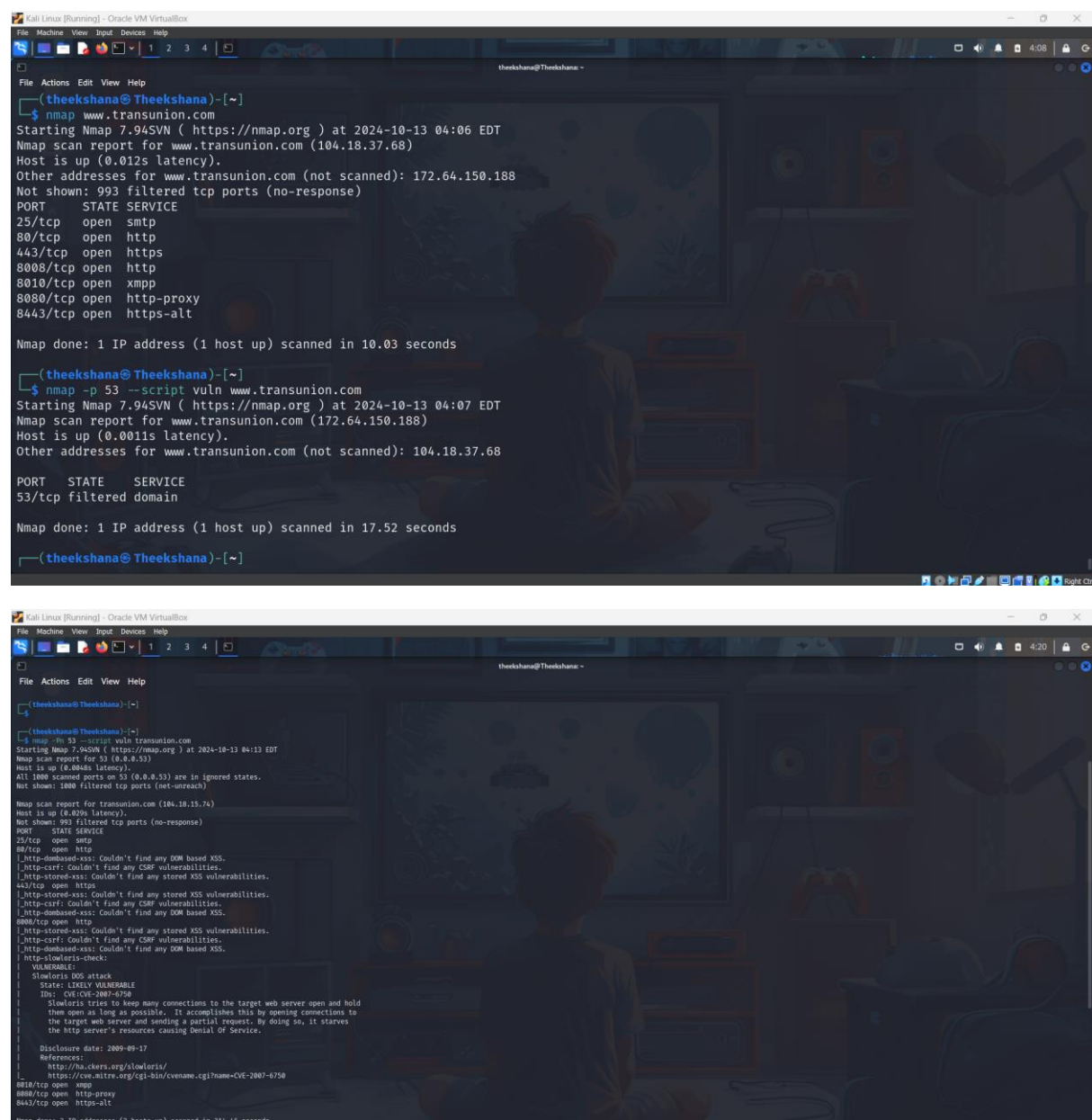
### Key Features of SQLmap:

1. **Automated SQL Injection Detection:** SQLmap can automatically detect different types of SQL injection vulnerabilities in a web application and propose appropriate methods to exploit them.
2. **Database Fingerprinting:** Once a vulnerability is detected, SQLmap can determine the type of database being used and its version, helping attackers or testers tailor their attack methods accordingly.
3. **Data Retrieval:** SQLmap allows the extraction of data from vulnerable databases. This includes dumping entire databases, tables, or specific columns.
4. **Advanced Exploitation:** The tool can be used to gain access to the underlying operating system by using SQL injection vulnerabilities to execute system commands, read/write files, or even establish a remote shell.
5. **Bypassing Protections:** SQLmap is capable of bypassing certain web application firewalls (WAFs) and input validation mechanisms, making it a powerful tool for penetration testers.

## 5.6 Nmap

Nmap, which means "Network Mapper," is a powerful open-source network scanning program that is used to map out a computer network by locating hosts and services on it. In addition to managing service upgrade schedules and identifying illegitimate devices or services, it lets users examine the network inventory.

Nmap uses raw IP packets to identify hosts that are accessible over the network, the services (name and version of the application) those hosts are providing, the operating systems (and versions) they are running, the kinds of firewalls or packet filters that are in place, and a host's many other features. It's a flexible tool that may be used for scheduling service upgrades, network inventory management, security audits, and network research. I used Nmap open port scanning tool to scan for open ports in [transunion.com](https://www.transunion.com) When scanning the results showed the following results.



```
(theekshana@Theekshana)~$ nmap www.transunion.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 04:06 EDT
Nmap scan report for www.transunion.com (104.18.37.68)
Host is up (0.012s latency).
Other addresses for www.transunion.com (not scanned): 172.64.150.188
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https
8008/tcp   open  http
8010/tcp   open  xmpp
8080/tcp   open  http-proxy
8443/tcp   open  https-alt

Nmap done: 1 IP address (1 host up) scanned in 10.03 seconds

(theekshana@Theekshana)~$ nmap -p 53 --script vuln www.transunion.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 04:07 EDT
Nmap scan report for www.transunion.com (172.64.150.188)
Host is up (0.0011s latency).
Other addresses for www.transunion.com (not scanned): 104.18.37.68
PORT      STATE SERVICE
53/tcp    filtered domain

Nmap done: 1 IP address (1 host up) scanned in 17.52 seconds

(theekshana@Theekshana)~$ nmap -p 53 --script vuln transunion.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 04:13 EDT
Nmap scan report for 53 (0.0.0.53)
Host is up (0.0046s latency).
All 1000 scanned ports on 53 (0.0.0.53) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for transunion.com (104.18.15.74)
Host is up (0.029s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
443/tcp   open  https
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
8008/tcp   open  http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-slowloris-check:
VULNERABLE:
Slowloris DOS attack
State: LITELY VULNERABLE
Id: CVE-2007-6750
Slowloris tries to keep many connections to the target web server open and hold
them open as long as possible. It accomplishes this by opening connections to
the target web server and sending a partial request. By doing so, it starves
the http server's resources causing Denial Of Service.
Disclosure date: 2009-09-17
References:
http://ha.cfers.org/slowloris/
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
8010/tcp   open  xmpp
8080/tcp   open  http-proxy
8443/tcp   open  https-alt

Nmap done: 2 IP addresses (2 hosts up) scanned in 314.46 seconds
```

The results shown above have carried successful scanning. Next, I have run the vulnerable port scan using Nmap on the open ports to find out if there's any vulnerable ports. The scan has resulted that no vulnerabilities were found.

```

theekshana@theekshana:~$ nmap -Pn 80 --script vuln transunion.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-13 04:21 EDT
Nmap scan report for 80 (0.0.0.80)
Host is up (0.0015s latency).
All 1000 scanned ports on 80 (0.0.0.80) are in ignored states.
Not shown: 1000 filtered tcp ports (net-unreach)

Nmap scan report for transunion.com (104.18.15.74)
Host is up (0.013s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
443/tcp   open  https
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
8008/tcp  open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
8010/tcp  open  xmpp
8080/tcp  open  http-proxy
|_http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves

```

```

Host is up (0.013s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
443/tcp   open  https
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
8008/tcp  open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
8010/tcp  open  xmpp
8080/tcp  open  http-proxy
|_http-slowloris-check:
| VULNERABLE:
| Slowloris DOS attack
| State: LIKELY VULNERABLE
| IDs: CVE:CVE-2007-6750
| Slowloris tries to keep many connections to the target web server open and hold
| them open as long as possible. It accomplishes this by opening connections to
| the target web server and sending a partial request. By doing so, it starves
| the http server's resources causing Denial Of Service.
|
| Disclosure date: 2009-09-17
| References:
| http://ha.ckers.org/slowloris/
| https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
8443/tcp  open  https-alt
Nmap done: 2 IP addresses (2 hosts up) scanned in 307.63 seconds

```

By scanning [transunion.com](https://transunion.com) with Nmap I found out these information.

Port	State	Service
25/tcp	Open	SMTP
80/tcp	Open	HTTP

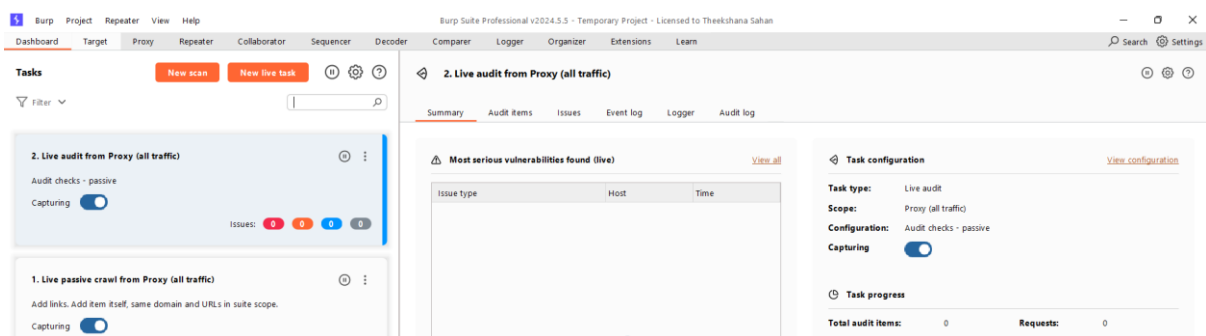


## Chapter 6: Searching for vulnerabilities

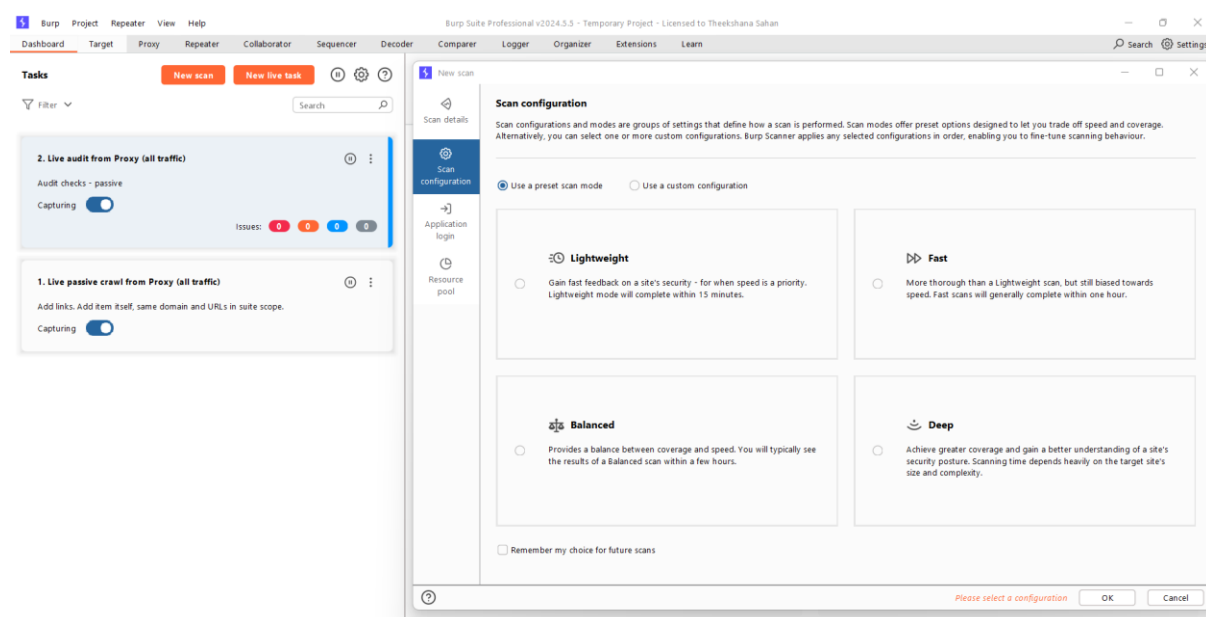
This chapter focuses on the vulnerability scanning phase of my bug bounty journey. To begin, I needed to determine which tools would be suitable for this phase. Through my research, I discovered that there are numerous tools available for this purpose, but a few stood out to me.

### 6.1 Burp Suite

Burp Suite is a software that developed by the company PortSwigger. it is a security application used for penetration testing of web applications. Both a free and a paid version of the software are available. Burp Suite is a powerful tool when it comes to scanning for vulnerabilities.



There are 4 available preset scan modes in burp suite.



We can select one of those 4 options and just add a url and Burp will do the rest..i preferred the fast mode.

The top screenshot shows the Burp Suite Professional v2024.5.5 interface. The 'Tasks' panel on the left has three options: '3. Crawl and audit of www.transunion.com', '2. Live audit from Proxy (all traffic)', and '1. Live passive crawl from Proxy (all traffic)'. The '3. Crawl and audit' task is selected and shows a progress bar. The main panel displays 'Most serious vulnerabilities found (live)' with a list of issues. The bottom screenshot shows the same interface but with the 'Issues' tab selected, displaying a detailed list of vulnerabilities including 'Cross-origin resource sharing (CORS)', 'User agent-dependent response', 'Cacheable HTTP response', and 'Strict transport security not enforced'.

With the help of burp suite I could find some of the critical vulnerabilities like

- Cross-origin resource sharing (CORS)
- Cross site request forgery (CSRF)
- Cacheable HTTP response
- User Agent-dependent response
- Strict transport security

## 6.2 ZAP

OWASP ZAP, also known as OWASP Zed Attack Proxy, is a popular open-source web application security testing tool. It is intended to assist developers and security experts in identifying security flaws in web applications as they are being developed and tested. Numerous features are available in ZAP for both automatic and manual security testing, such as:

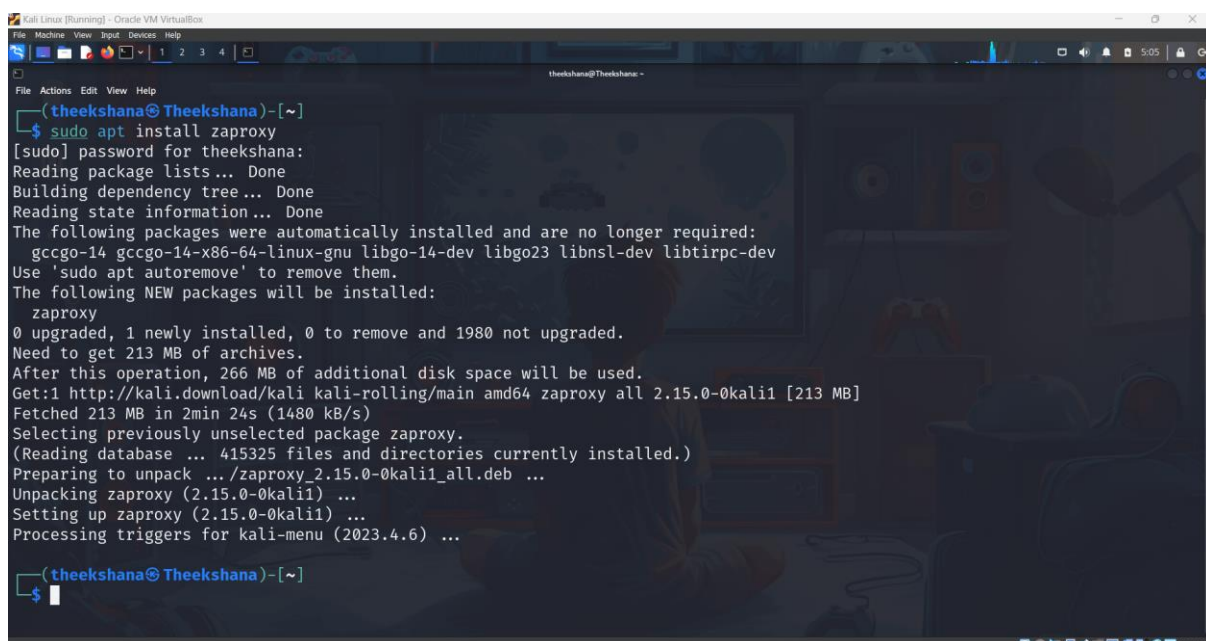
**1. Automated scanning:** ZAP can automatically check web applications for common security flaws including insecure direct object references (IDOR), SQL injection, and cross-site scripting (XSS).

**2. human testing:** By enabling users to intercept and alter HTTP requests and answers, human testing for security flaws that automatic scans might miss is possible.

**3. Active and passive scanning:** ZAP is capable of carrying out both passive and active scans, which involve passively analyzing traffic to find possible vulnerabilities and sending specially constructed queries to the application to find flaws.

**4. Reporting:** ZAP produces thorough reports that list all vulnerabilities found and offer suggestions for fixing them. In order to assure the security of web applications, developers, security experts, and quality assurance teams frequently utilize OWASP ZAP, which is an effective tool for evaluating the security posture of web applications.

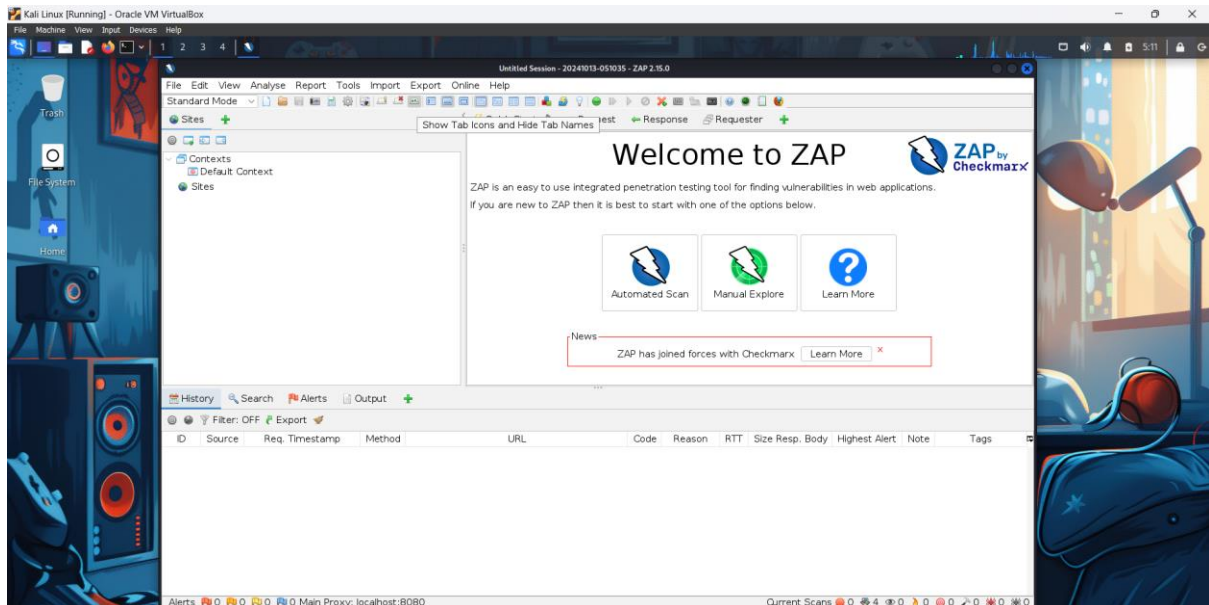
Zap(Zed Attack Proxy), previously known as OWASP Zap is a widely used free and open source powerful automated tool for vulnerability scanning Since zap was not available in my kali Linux environment, I had to install it first.



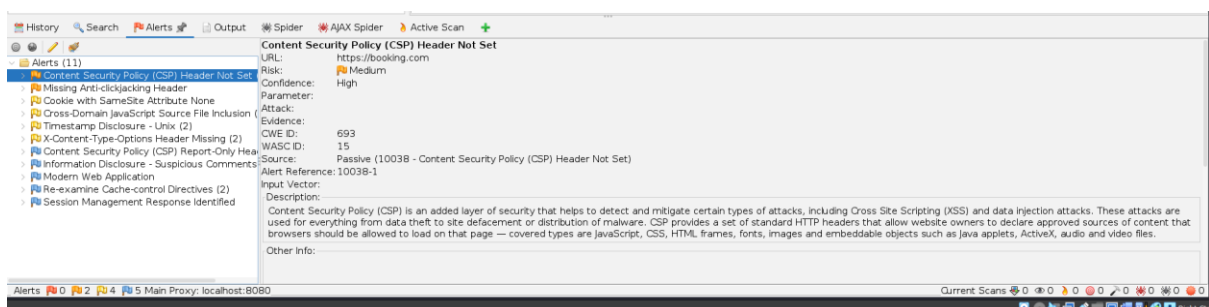
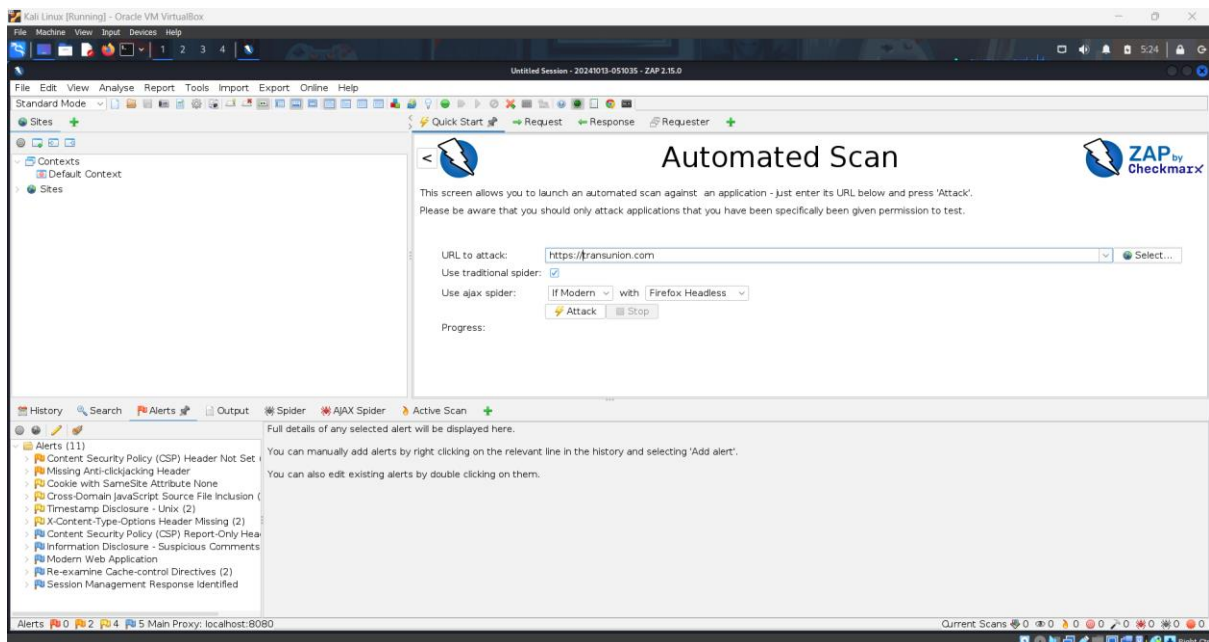
```
(theekshana@Theekshana)-[~]
$ sudo apt install zaproxy
[sudo] password for theekshana:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gccgo-14 gccgo-14-x86-64-linux-gnu libgo-14-dev libgo23 libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  zaproxy
0 upgraded, 1 newly installed, 0 to remove and 1980 not upgraded.
Need to get 213 MB of archives.
After this operation, 266 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 zaproxy all 2.15.0-0kali1 [213 MB]
Fetched 213 MB in 2min 24s (1480 kB/s)
Selecting previously unselected package zaproxy.
(Reading database ... 415325 files and directories currently installed.)
Preparing to unpack .../zaproxy_2.15.0-0kali1_all.deb ...
Unpacking zaproxy (2.15.0-0kali1) ...
Setting up zaproxy (2.15.0-0kali1) ...
Processing triggers for kali-menu (2023.4.6) ...

(theekshana@Theekshana)-[~]
$
```

This is how zap looks like , Numerous features are available in ZAP for both automatic and manual security testing .



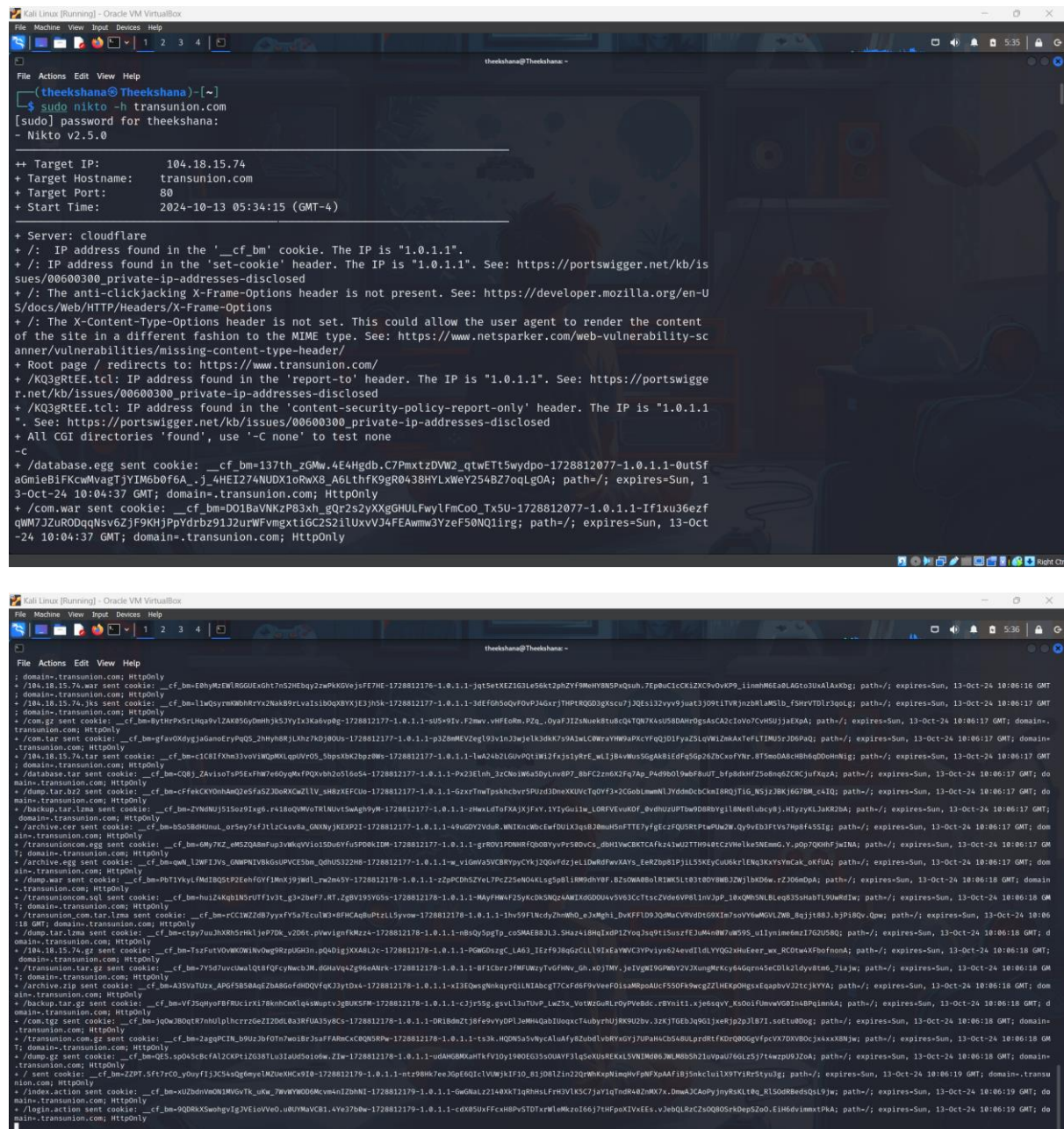
For me the preferred scan mode is Automated scan.there we have to just enter a url and scan for it.





## 6.3 Nikto

Nikto is a free web server scanner used for detecting vulnerabilities in web servers and applications. Nikto checks for potentially dangerous files, outdated server versions, and configuration issues. As of my understanding, Nikto is not as powerful as both Burp Suite and ZAP, but it is a comprehensive tool when it comes to vulnerability scanning.



```
(theekshana@Theekshana)-[~]
$ sudo nikto -h transunion.com
[sudo] password for theekshana:
- Nikto v2.5.0

+-- Target IP: 104.18.15.74
+ Target Hostname: transunion.com
+ Target Port: 80
+ Start Time: 2024-10-13 05:34:15 (GMT-4)

+ Server: cloudflare
+ /: IP address found in the '__cf_bm' cookie. The IP is "1.0.1.1".
+ /: IP address found in the 'set-cookie' header. The IP is "1.0.1.1". See: https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: 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 MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://www.transunion.com/
+ /KQ3gRtEE.tcl: IP address found in the 'report-to' header. The IP is "1.0.1.1". See: https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed
+ /KQ3gRtEE.tcl: IP address found in the 'content-security-policy-report-only' header. The IP is "1.0.1.1". See: https://portswigger.net/kb/issues/00600300_private-ip-addresses-disclosed
+ All CGI directories 'found', use "-C none" to test none
-C
+ /database.egg sent cookie: __cf_bm=137th_zGMw.4E4Hgdb.C7PmxtzDVW2_qtWtT5wydpo-1728812077-1.0.1.1-0utSfagmleBIFKcWtMvagtJYIM6b0f6A..j_4HEI274NUDX1oRwX8_A6LthfK9gR0438HYLXWey254BZ7oqJGOA; path=/; expires-Sun, 13-Oct-24 10:04:37 GMT; domain=.transunion.com; HttpOnly
+ /com.war sent cookie: __cf_bm=D01BaVnKzP83xh_gQr2s2yXgGhULFwlyFmCoO_Tx5U-1728812077-1.0.1.1-If1xu36ezfQWM7J2uR0DqqNs6VzJ9KHJPPyRbZ91J2uRWFvmgxt1GCS21lUxvV34FEAmw3YzeF50NQ1irg; path=/; expires-Sun, 13-Oct-24 10:04:37 GMT; domain=.transunion.com; HttpOnly

; domain=.transunion.com; HttpOnly
; /Bk-18-15-74-war sent cookie: __cf_bm=EhhyRzEWLROGueDh7d2HEby2wPkwK9vjeJ7F7HE-1728812176-1.0.1.1-jqt5eXEtZIG3L56kzt2pZYFMehYBM5Pqush.7Epbc1CkZCX9vovKP9_l1mhM6eA0eAG103uAlAxbg; path=/; expires-Sun, 13-Oct-24 10:06:16 GMT
; domain=.transunion.com; HttpOnly
; /Bk-18-15-74-jks sent cookie: __cf_bm=1WQyymwMDhRrY2NA8B9rva1sBoqXVXJEJhSk-1728812177-1.0.1.1-3dEfgS0QvFOVP34grJTHPRQD3gKscu7JQEs132vyy9Jut3J09LTVWjnzBRLAMSLB_fShrvTDI3qql; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT
; domain=.transunion.com; HttpOnly
; /com.g2 sent cookie: __cf_bm=HYRPSLHq9vLZAK850dmmhJk5Jy1x3Kavpog-1728812177-1.0.1.1-uS9v91v.F2mw.vHfCohm.PZg_OyafJ3ZsNuek8tubcQ4TQNKAsUS0dAHr0gAsCA2C1ov07CvH9UjjaEPA; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /com.tar sent cookie: __cf_bm=fav0XdyJ1Gd9eFvPqD5_2HhY8RjLKhZ7KdJ80uJ-1728812177-1.0.1.1-p12BMMEZeg193v1nJ3jle13dKk750A1aCmWzYmWpAPKvYqJ01FyZ1sQW1ZMAxTeFTIMJ5rJ08PaQ; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /Bk-18-15-74-war sent cookie: __cf_bm=C1C81Fhm33v0V1WQMLqpUvR05_5bpb8K2bzp8W-1728812177-1.0.1.1-1W4242LQvUqP1lW12f3jYrE_wL1B4vWu5S6gAB1EdfG56pZ8CxfYR_BT5mdA8cHm6Qd0WHnig; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /database.tar sent cookie: __cf_bm=C8J_Zav1stP5eXfFW7e0yQvWfQXv8h2o516o54-1728812177-1.0.1.1-Px23ELnh_3cNo1W6s5Dy1uW8P7B8FC2z6K2Fg7Ap_P49b019wbF8u0T_bfp8dKHfZ50bq6ZCRJufKqZA; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /dump.tar.bz2 sent cookie: __cf_bm=CfKcKcyOhnQm2eSfA52D8wKCa21LV_sH8XEFUCu-1728812177-1.0.1.1-GzxrTmTpskhcbvP5Pd3D8KXUCvQvF3x2GobLmmN1YddmcbCk18RQJ71G_W5j2BK1667BM_c41Q; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /backcup.tar.lzma sent cookie: __cf_bm=ZVYHNUJ515o2Iq8..r418Q9Wv8TR1UUV1SwAg9yM-1728812177-1.0.1.1-2HwLdToFAJXJfYr_V1YGu1w_L0RFvEuX0F_BvduHJUPThwD8R9Yg1lB8e1ubcy8J.H1zyK1J3KRR2BA; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /archive.tar sent cookie: __cf_bm=5B05dHUmU_0r5ey7f21lcCv8a_GNNyJKEXP21-1728812177-1.0.1.1-4rUd0Y2VdR.W1K1K8mEwDuCfWJX3qB2mH5HfTTE7yfgtcZFQ5URPtwUwZw.QyVE3F3vY7Huf64551g; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /transunion.com.egg sent cookie: __cf_bm=6MyTKZ_eMSZQARFup3jWvQV1o150uYfUSP0KIDM-1728812177-1.0.1.1-gr0V1PDH1RfQ0BYvYp5R0Vcs_d0H1VwCBKtCAfz41wJ2TH940CZVHe1K5NEmG.Y.p0p7QKHfjW1NA; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /archive.egg sent cookie: __cf_bm=qWm_L2WFIJVs_QWMP1V8K5UPVCE5bm_QDHUS32H8-1728812177-1.0.1.1-w_vlgWva5VCBRyPyCYk12QovFdzJcLldWdPwXAYs_EerZbp81P3j1L55K5yCu06krLEnQ3KvYsYac_k0kFUA; path=/; expires-Sun, 13-Oct-24 10:06:17 GMT; domain=.transunion.com; HttpOnly
; /dump.war sent cookie: __cf_bm=PbTVkYLM1BQ5P2EhGfV1mXj9JwL_rW2m45V-1728812176-1.0.1.1-z2pPCDSZvL7PcZ5e04KSLsg5p8l18Wd8vBYF.B50WA8801K1KSL103180YBWB2JW1bX0w.rZJ060Pa; path=/; expires-Sun, 13-Oct-24 10:06:16 GMT; domain=.transunion.com; HttpOnly
; /transunion.com.egg sent cookie: __cf_bm=hu124KqJLNSrUf1Vt_g3zbeF7_RT.ZgV93YGS5-1728812178-1.0.1.1-M4yFWmF25yKdSMQz4W1E6G00UvSV53CtScZv6d6Vp11vJp3p10XQMS1BLq8355HbTL9Wd8dW; path=/; expires-Sun, 13-Oct-24 10:06:16 GMT; domain=.transunion.com; HttpOnly
; /transunion.com.tar.lzma sent cookie: __cf_bm=RCC1WZ287yxF5a7Ecu1W3BfCAQbPtL3YvWw-1728812178-1.0.1.1-1h59F1NcdyZWhMh0_eJMHg1_DvKFF1D9JQdMcVRyD66XIm7soVYwMGLW2B_Bqj1883.b3P18Qv.Qw; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /dump.tar.lzma sent cookie: __cf_bm=ctyp7umJhX8H5hK1j9P7DM_v20et.pwvignfKz2a-1728812178-1.0.1.1-n85QySpTg_c0SMAE88JL3.Sh4z418HqLndP12Yq3sq9t1SuzrFEJ3uM4hW7uW9S_u1ynimem2I7G2US8Q; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /Bk-18-15-74-g2 sent cookie: __cf_bm=7sZfU7vOWK0M1W0vG8Rz2pUg3n.pQ401gJX0A8L2c-1728812178-1.0.1.1-PGMS0zgc_LA63_1EzF938qGcL11914EAYWVC3YViy624evd11dLYQ2XHuEer_w_RCDtW4XfBofn0NA; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /transunion.tar.gz sent cookie: __cf_bm=7Y5v0dwa1Q18fQcyWcb3k_8dW4vZ2g66ANr-1728812178-1.0.1.1-Bf1CRRJfMUUyVtGvHn_Gh_x0JTWy.je1VgW19W9B2V2JXWgRkCY64Gqns5eCD1k21dyv8t6_71ajw; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /archive.zip sent cookie: __cf_bm=A35vATuz_ApGf58BA8eZBA86fndQvFq3Jy1dX-1728812178-1.0.1.1-x13EQmgWnkqyQ1LNTAbcQ7Cxf68FVvveF01saMRp0AUCF50FA9wczZLHEK9Dh8EgqpbV321cjkYVA; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /backcup.tar.gz sent cookie: __cf_bm=Vf35qHofFRUC1rX178hnhCmX1qskmvtv3gUK5FW-1728812178-1.0.1.1-c3jY55g.vsu13L3uTUP_Lw25x_VotW2GdLr0yPvEddc.rBvnt1.xj6sqvY_KsoiFumwVGD1nABPqmkA; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /com.tgz sent cookie: __cf_bm=jQwJ80qT87rHdUplhczr2eZ1D08x3RfUA35y8C-1728812178-1.0.1.1-0R1BmZ3j8f6v9yVdP136MhQz1U0qxT4uByrUJ8K9U2bv.3ZKJ7G6Jq8q1JaeRj3p21B71.s0tU8XG; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /transunion.com.gz sent cookie: __cf_bm=zagUC1R_BU0z70f0Tn7w81r3fAFmKcCRQ5RPw-1728812178-1.0.1.1-t3k1.HQMS3v5hYCa1uafY82ubdlv8YKv9Y7UPAHMc3AS4SLpr0rTfXDR08GvfpCYX7DXV8Dc3xxxX8Jw; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; /dump.gz sent cookie: __cf_bm=QES.sp045cbfAl2CPK12038T1u3u0s1o1ow.Z1w-1728812178-1.0.1.1-uAHG0MKXhTKfVoy190E0G350uVYF31q5eXUSREKSLVNIIM06.NLM85h21uvpaU76Lz5j7twpz9J2oa; path=/; expires-Sun, 13-Oct-24 10:06:18 GMT; domain=.transunion.com; HttpOnly
; / sent cookie: __cf_bm=Z2PT.5f7rC0_yohyF1J354Q8mY1MZuWKhC9I8-1728812179-1.0.1.1-ntr98K7ee3p6EQ1C1VWgJf0_8J0B1Z1n22QrwhKp1mghvFpFpAAAF1B3ncklu1lX9TY1R5tyu3g; path=/; expires-Sun, 13-Oct-24 10:06:19 GMT; domain=.transunion.com; HttpOnly
; /index.action sent cookie: __cf_bm=xLzBdvM0V1MVGvT8_uKw_WWY0W0G6CvM4nIZ8NI-1728812179-1.0.1.1-Gw0Aul22140Xt1gRthHsLf1gK1V3K37jYt1qTnR4Z2M0X7..DmaJ2CaPyJnyRKLt8_R1S0d8BedsQ19jw; path=/; expires-Sun, 13-Oct-24 10:06:19 GMT; domain=.transunion.com; HttpOnly
; /login.action sent cookie: __cf_bm=QD6K8SwghvJ3V1oVvo.u0UWAvCB1.4Yv37bW-1728812179-1.0.1.1-cDR50XfFCkRbPvSTDTxWmXo16677hFp0pIVXEs.vJ6bQLRzCz0Q805F8Dep52oI.H1H6v1mmtPKA; path=/; expires-Sun, 13-Oct-24 10:06:19 GMT; domain=.transunion.com; HttpOnly
```

With Nikto scans I was able to identify some of the vulnerabilities as following.

- The anti-clickjacking X-Frame-Options header is not present
- The X-Content-Type-Options header is not set
- Uncommon header 'accept-ch' found

## 6.4 Netspaker

### Challenges :

My bug bounty journey was full of challenges and obstacles. In the first few days I had to overcome things like how to select the tools I want to use, what are the domains I can check for vulnerabilities.as I doing some search online it was clear to me what tools I could use to do the process.and with some more search I was able to identify the 10 domains I wanted to test for my bug bounty assignment.

Furthermore, I had to face to a problem where the targets have limited scope in the eligible criteria. With the limited scope there was not much to test on some websites.in addition of that some of the findings was a bit unclear to me at first. I had to do some research to find out what some things were.

Regarding Burp Suite, Burp has two edition.Free and Professional.we can not do vulnerability scans with the Free version.So I had to find a cracked version of the Burp Suite Professional and install it in my pc.Since a one copy of Burp suite professional version costs \$449 per year.And another problem I had to face was some of the tools I used didn't work properly sometimes.once when in a middle of a scan Zap just frozed.sometimes subfinder didn't work properly.and also nikto had some issues.to resolve these issues I had to do things like restarts,reinstall the tools and update them.

Another challenge I had to overcome was the limited time frame. Especially when searching for the targets, I had to do find subdomains of each site. And I had to go through those one by one carefully.That was time consuming.and also I had to manage time between this assignment and other academic work which was kind of a challenge in some situations.

## **Reflections and takeaways:**

This assignment was capable of pushing me to my limits. I was able to get hand on experience when it comes to testing actual websites for vulnerabilities which can help me when I enter the industry in the future.

Furthermore,

- How to properly write a bug bounty report
- How to understand the scope of the domain
- How to use tools effectively
- Manual and automated testing
- Research about fixes
- Understanding vulnerabilities and their behavior
- Web application behavior
- How to summarize daily progress in to a journal

Above mentioned are some of the many key skills I learned throughout the period of doing this assignment.

## **Conclusion:**

My bug bounty adventure has been an eye-opening voyage filled with priceless lessons, thrilling discoveries, and significant personal development. Taking stock of the many experiences that have molded my career, I am extremely appreciative of the chances, difficulties, and victories that have shaped my path in the everevolving field of vulnerability disclosure and cybersecurity.

My bug bounty adventure has been an eye-opening voyage filled with priceless lessons, thrilling discoveries, and significant personal development. Taking stock of the many experiences that have molded my career, I am extremely appreciative of the chances, difficulties, and victories that have shaped my path in the everevolving field of vulnerability disclosure and cybersecurity.

My bug bounty journey was started with careful program selection, which led me to programs that matched my interests, skills, and areas of expertise. I found chances to capitalize on my abilities and support the security of companies in a variety of sectors and industries by doing thorough research and using sound judgment.



## References:

1. <https://www.cvedetails.com/>
2. <https://owasp.org/www-project-top-ten/>
3. <https://hackerone.com/elastic?type=team>
4. [https://hackerone.com/poloniex/policy\\_scopes](https://hackerone.com/poloniex/policy_scopes)
5. <https://www.youtube.com/watch?v=dAV3z2O7ghY>
6. <https://www.youtube.com/watch?v=3aKA4RkAg78>
7. <https://www.youtube.com/watch?v=M6N7gEZ-IUQ>
8. <https://github.com/vavkamil/awesome-bugbounty-tools>
9. <https://docs.bugcrowd.com/researchers/reporting-managing-submissions/reporting-a-bug/>
10. <https://dewcode.medium.com/how-to-write-a-good-bug-bounty-report-in-just-10-minutes980a0a1b1b18>

Student Register Number	Student Name
IT 22083678	SAHAN H.P.T