Sri Lanka Institute of Information Technology

IT21299902 (ZAKEY M.S.M.A)

Bug bounty journal.

Web security – IE2062

B.Sc. (Hons) in Information Technology Specialization in cyber security.

Declaration:

- I hereby certify that no part of this assignment has been copied from any other work or any other source. No part of this assignment has been written/produced for me by another person.

- I hold a copy of this assignment that I can produce if the original is lost or damaged.

Project Details:

Case Study	Bug bounty journal book
Date Of completion	18/05/2023

Table of contents.

Introduction.

Welcome to my bug bounty journal, where I document my exciting journey into the realm of bug bounty programs. Bug bounty programs offer a unique opportunity for ethical hackers like me to contribute to the security of various organizations by identifying and reporting vulnerabilities in their systems and applications.

In this journal, I will cover various aspects of my bug bounty journey. Firstly, I introduce the bug bounty methodology, outlining the systematic approach I followed in identifying, exploiting, and reporting vulnerabilities. This methodology served as the foundation for my efforts and ensured a structured and efficient approach to my bug bounty endeavors.

I then delve into the different phases I encountered during my bug bounty activities. From reconnaissance and vulnerability discovery to exploitation and reporting, each phase played a crucial role in my journey. I discuss the strategies and techniques employed in each phase, maximizing my chances of success.

To aid my bug bounty activities, I utilized a range of tools. These tools, including reconnaissance and information gathering tools, vulnerability scanners, and exploit frameworks, streamlined my workflow and helped me uncover potential security flaws effectively. I provide insights into the tools used and their impact on my bug bounty experience.

Throughout my bug bounty engagements, I explored various vulnerabilities. I detail these vulnerabilities, discussing their impact, exploitability, and potential consequences. I provide insights into the specific systems or applications affected, along with the steps taken to reproduce and validate these vulnerabilities.

Then I outline the obstacles I encountered during my bug bounty journey and share how I overcame them. I provide strategies, workarounds, and lessons learned from each challenge, showcasing my problem-solving abilities. Lastly, I reflect upon the overall experience gained from my bug bounty activities and share the key takeaways from my journey. I discuss the valuable insights acquired, the skills developed, and the personal growth experienced because of participating in bug bounty programs.

By documenting my bug bounty experiences and sharing them in this journal, I aim to inspire fellow cybersecurity enthusiasts, contribute to the bug bounty community, and continue my own growth as a proficient ethical hacker.

1. Introduction to Bug bounty methodology.

I. What is a bug bounty?

Bug bounty is a process where organizations invite independent security researchers to identify and report vulnerabilities or security issues in their software systems, websites, and applications [1]. This approach allows companies to identify and address security flaws before they can be exploited by malicious actors [2].

II. Phases of bug bounty?

Bug bounty programs typically have several phases that help to ensure a systematic and thorough review of the system's security posture [3].

1. Phase 01

The first phase involves scoping, where the organization defines the target systems, applications, and vulnerabilities that are in scope for the program. This is important because it helps to focus the efforts of security researchers and ensures that the organization can adequately address any vulnerabilities that are identified [4].

2. Phase 02

The second phase involves testing, where researchers use various tools and techniques to identify vulnerabilities in the target systems and applications. This phase can include manual testing, automated scanning, and other specialized tools designed to identify specific vulnerabilities. Some of the popular tools used in bug bounty programs include **Burp Suite, OWASP ZAP, Nmap, and Metasploit** [5].

3. Phase 03

Once vulnerabilities are identified, the third phase involves reporting, where the researcher provides a detailed report of the vulnerability to the organization. This report should include a description of the vulnerability, its potential impact, and any recommended steps for remediation. The organization then reviews the report and validates the vulnerability [6].

4. Phase 04

The fourth and final phase is remediation, where the organization takes steps to fix the vulnerability. Depending on the nature and severity of the vulnerability, this may involve simple configuration changes or more significant changes to the software code. Once the vulnerability is remediated, the organization can then confirm the fix and close out the bug report.

Overall, bug bounty programs are an important tool for organizations to improve their security posture and stay ahead of potential threats. By working with independent security researchers, companies can identify and address vulnerabilities before they can be exploited by malicious actors, thereby reducing the risk of a security breach [7].

2. Introduction to OWASP top 10 vulnerabilities.

I. Injection.

- This is the most common attack technique that is used by intruders these days to see whether injection vulnerability is present or not in targeted system or application. Injection means attacker simply sending untrusted data to system interpreter by in using URL bar, tools, intercepting the request.
- When the attacker sends that untrusted data to the interpreter it will allow the intruders to execute harmful commands. Then attackers might be able to retrieve the data that shouldn't be visible to users.
- Some of the most used injection types are SQL injection, OS command injection and LDAP injection [8].

i. SQL injection.

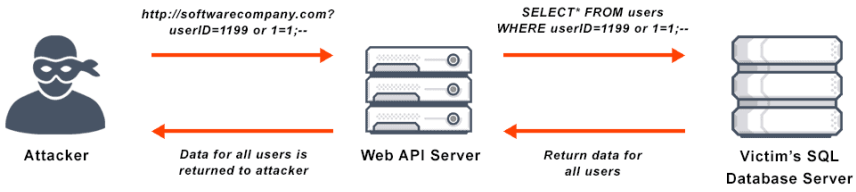


Figure 1: SQL injection example

ii. OS command injection.

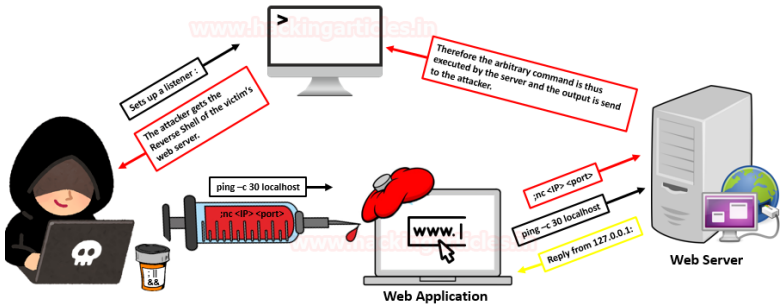


Figure 2:OS command injection

iii. LDAP injection.

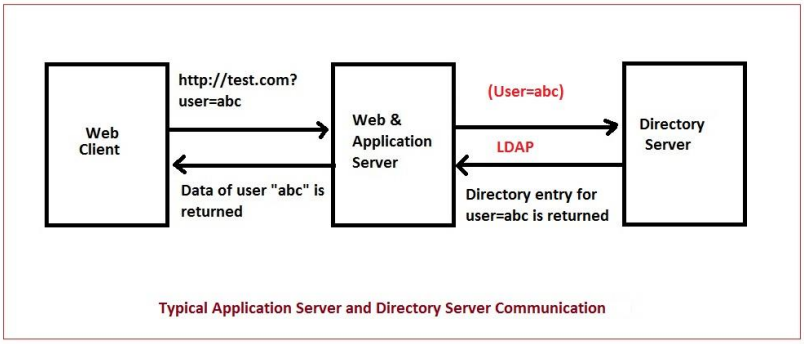


Figure 3: LDAP injection.

II. Broken Authentication.

- User authentication plays an important role in modern web applications. By this mechanism systems identify the specific user. Weakness in this system may lead to serious issues from the client perspective as well as form the system perspective.
- If a broken authentication vulnerability presents it the web application intruder may use that to impersonate user sessions, compromise session tokens or uses that to exploit other authentication flaws [9].

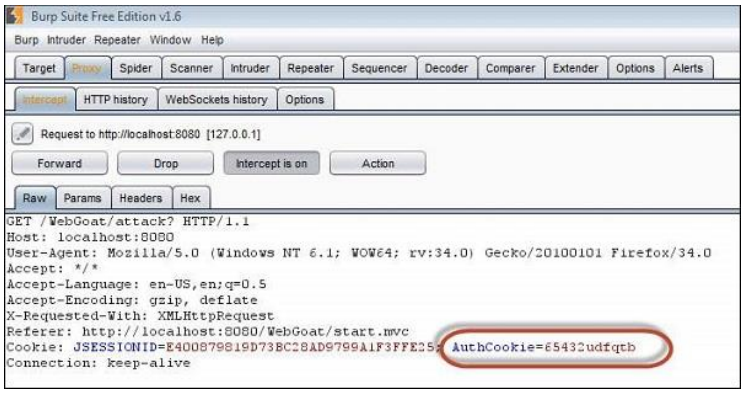
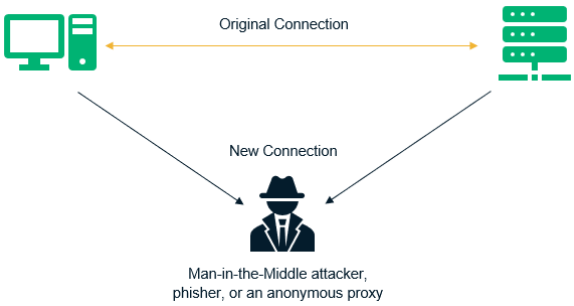


Figure 4: presence of broken authentication.

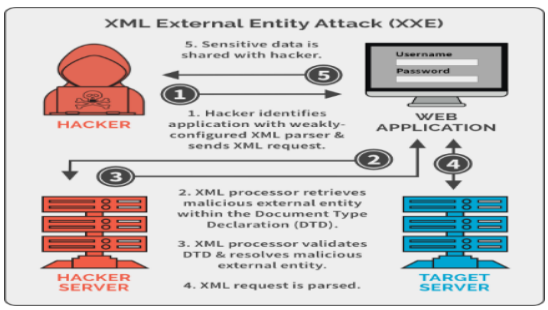
III. Sensitive Data Exposure.

- Web applications may not fully protect the data transmitting over the network or when storing sensitive information. Attacker will use that vulnerability to intercept the network and gather the sensitive information, modify that information, or steal that information and use it to exploit other system vulnerabilities.
- For example, if intruder able to get sensitive data he/she may be able to log in to the system by stealing authenticated user credentials, if he can log in as administrator attacker can modify the system configurations and he can disable security features to gain the full control of the web application [10].



IV. XML External Entities.

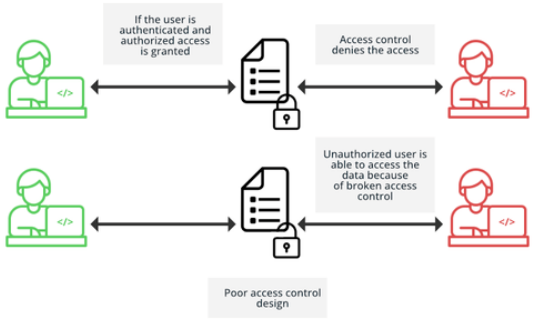
- An XXE attack makes use of XML translation and data features. An attacker will be able to interact with the backend or other external systems that the application can access, as well as read files on such systems.
- Additionally, XXE assaults can activate port scanning, enable remote code execution, and denial of service attacks. They can also lead to server-side request forgery attacks and denial of service attacks. XXE assaults come in **two flavors**:
    a. In-band XXE - The attacker receives a quick response to the XXE payload.
    b. Blind XXE (out-of-band XXE) - When there won't be a prompt response, the attacker may want to mirror the output of his payload to a different file or his own server [11].



V. Broken Access Control.

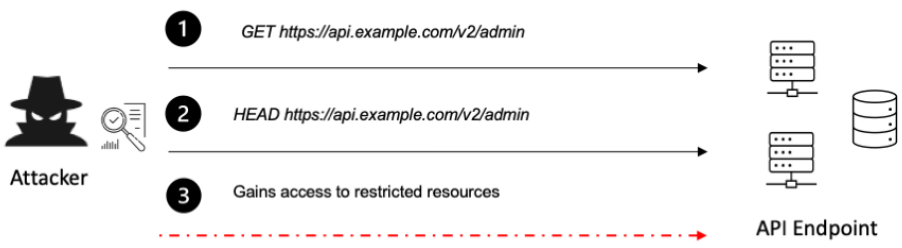
- If the web application fails to enforce access controls mechanisms properly, the attacker can use that as an entry point to do various harmful activities in the web application. For example, there are web pages that are hidden from the normal user. Those may only be available to authorized persons [12].
- If this vulnerability is percent normal users also be able to view that hidden web pages without proper authorization. This kind of attacks may lead,

    a. Disclosure of sensitive information.
    b. Gain access to unauthorized functionality.



VI. Security Misconfiguration.

- Security misconfigurations mean when the developer of the application may not properly implement the security measures to address all the possible attacks. That may lead to big problems like gaining unauthorized access to the system or database, stealing tokens and sessions [13].
- There are some most common security misconfigurations,
    a. Permission on cloud services is poorly configured such as S3 bucket.
    b. Unnecessary features are enabled such as services, pages, accounts, and privileges.
    c. HTTP security headers are not used or too much information is revealed in the server HTTP header.

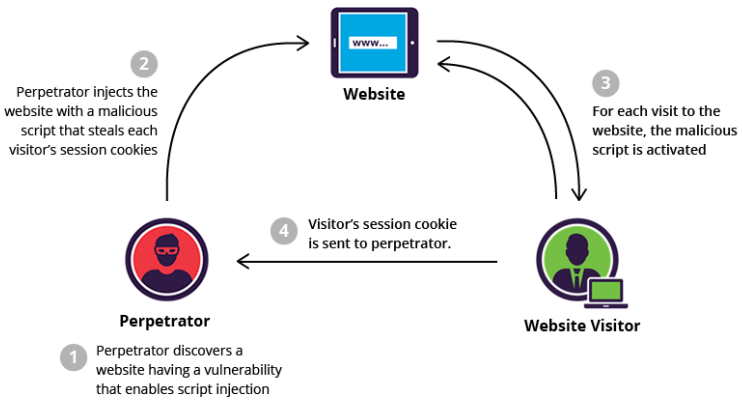


VII. Cross-Site-Scripting.

- XSS is also an injection attack.in this type of attack intruder simply injecting scripts into a vulnerable application. Usually, they execute those scripts on the victim's machine. Web development languages like CSS and Java script are more likely to vulnerable to this type of attack [14].
- To be successful in this attack intruder may need to social engineering to successfully send the script to the victim. With the successful execution of the script intruder can hijack the user session. There are mainly 3 types of XSS attacks.

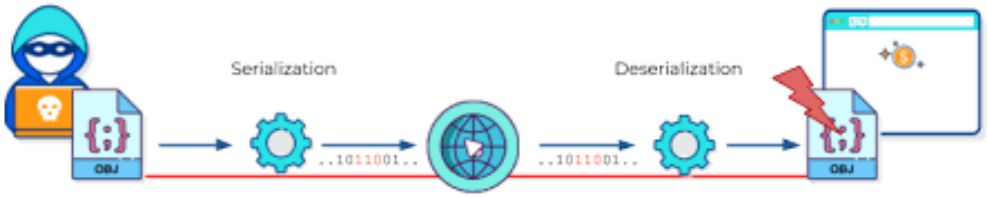
  Reflected XSS - An attacker creates a malicious payload which is a portion of the victims. request to the website, and this payload is included in response back to the user by the website. A user is tricked into clicking that specific URL by the attacker to execute the payload [15].

  a. **DOM-Based XSS** - Document Object Model is a programming interface for HTML and XML, and XSS changes the document structure, style, and content.
  b. **Stored XSS** - When the user input is not sanitized and inserted into the database, stored. XSS happens. As the malicious string originates from the website's database, this is the most dangerous type of XSS attack.
  c. **Reflected XSS** - An attacker creates a malicious payload which is a portion of the victim's request to the website, and this payload is included in response back to the user by the website. A user is tricked into clicking that specific URL by the attacker to execute the payload [16].



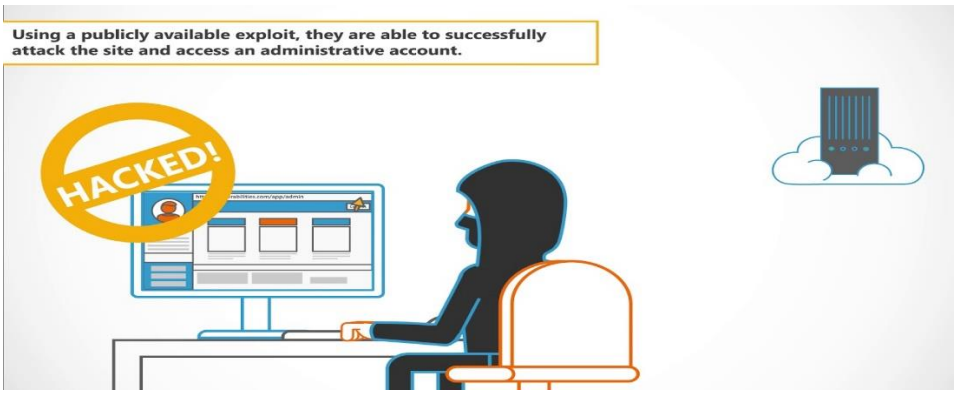
VIII. Insecure Deserialization.

- This happens when the logic of the application is replaced by the attacker with a malicious code. It will allow intruders to perform DOS with remote code execution, privilege escalation or other attacks. To these attacks applications like e-commers web applications, forums, APIS's and application runtimes like (Tomcat, Jenkins) are vulnerable to these kinds of attacks.
- By performing this kind of attack intruder might be able to gain full control of the application. So, then they can simply steal sensitive information and other details [17].



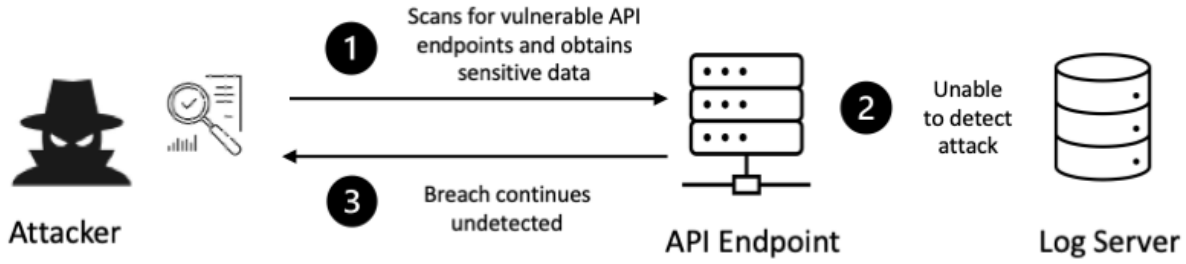
IX. Using Components with Known Vulnerabilities.

- If the underlying programs which are used on the website are outdated, there may be a high chance of finding a well-known vulnerability, which can be used to gain access to the system. So, as an attacker, one must do very simple work and that is why this vulnerability is rated low by OWASP.
- For example, let us assume that there is a website which is developed using WordPress as the content management system and the WordPress version is out-to-date, because developers have forgotten to update it.
- An attacker can easily detect the WordPress version at enumeration phase and look for a pre-defined specific vulnerability associated with that version and use it to gain access to the system. So, the attacker only must do some research on how to use the vulnerability, and it is simple as that [18].



X. Insufficient Logging and monitoring.

- This type of attack occurs when there is not enough logging and monitoring systems for the application. If so, the attacker can log in to an account that does not belong to him and can do various harmful activities on the application.
- To this problem there are some solutions like implementing intrusion detection systems, intrusion prevention systems, and logging. By implementing those system can overcome this issue on an extendable level [19].



3. Tools used in bug bounty program.

Throughout the finding vulnerabilities of web application, I have identified many web applications that had considerable amount of OWASP Top 10 vulnerabilities. I reported to the relevant party about the existence of that vulnerability on their site.

To this journal book I have used vulnerable web sites to show case the thing I learn, vulnerabilities I found, tools I used and to explain what I learn from finding those. The following vulnerable web applications are [20],

1. **MalwareByte.com**
2. **Inmobi.com**
3. **Merck.com**
4. **Curl.se**

To analyze vulnerabilities on those web applications I have used lots of techniques, theories, and various tools to figure out the vulnerability. As I mentioned before I have used lots of tools. To explain it I can categorize those tools into mainly 2 parts [21].

Note: for the ease of explanation of what kind of tools I used and how I used those tools I have only used a **1 domain** that I identified as a vulnerable web application.

a. **Reconnaissance tools. (Information gathering)**
b. **Vulnerability analysis & exploitation tools.**

Under each part there are lots of subcategories and tools that can be used to information gathering. Also, the same tool can be used to identify different types of information that we need to figure out the vulnerability.

This information will also be helpful when giving correct solutions to the identified vulnerability too.

a. Reconnaissance tools. (Information gathering)

Under the information gathering phase used many tools for different purposes. There are lots of automated tools as well as manual testing tools available. I have used some of the most recognized tools for information gathering phase under different categories.

i. **Target validation**
ii. **Subdomain Enumeration.**
iii. **Finding alive sub domains.**
iv. **Finding achieved information.**
v. **DNS Enumeration.**
vi. **Public Device Enumeration.**
vii. **Find the structure of the file system.**

Under each category I have used different types of software as well as plugins, framework's, automated bots, crawlers to gather information about the target web application.

i. Target validation.

This is the first step of vulnerability assessment. Before doing bug bounty we need to identify the vulnerable target by gathering information from all possible aspects. To target validation, I have used many inbuild and 3rd party software's.

A. **Nslookup.io**
B. **Censys**
C. **Wafw00f**
D. **Nmap**

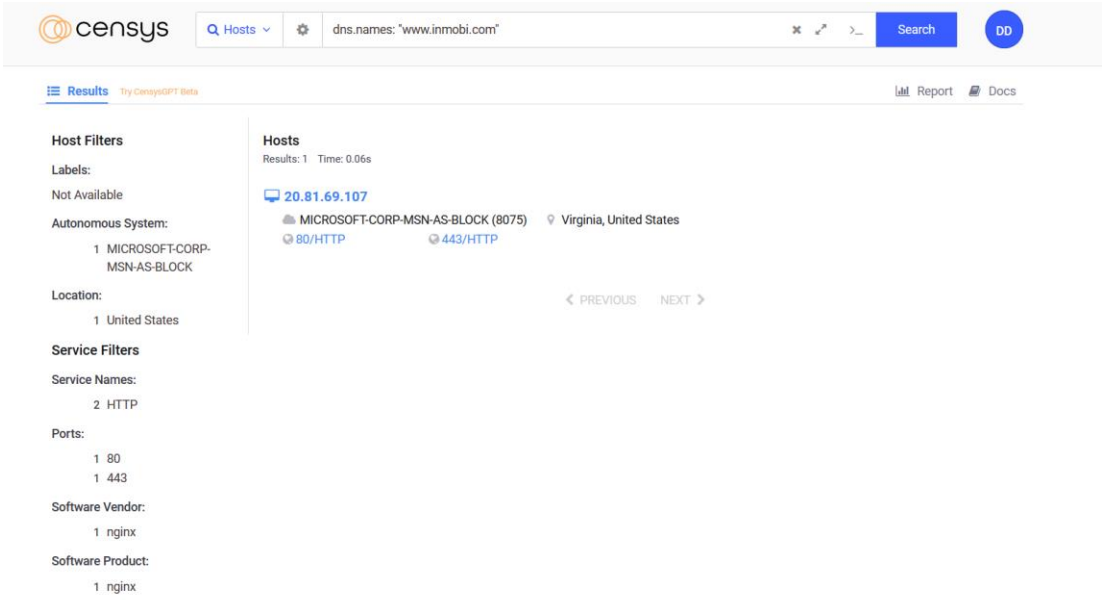
A. Nslookup.io

From this build in Linux tool, we can identify the Ip address or the domain name system records for the given host name [22].

```
┌──(root㉿error404)-[~]
└─# ping inmobi.com
PING inmobi.com (20.81.69.107) 56(84) bytes of data.
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=1 ttl=106 time=295 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=2 ttl=106 time=262 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=3 ttl=106 time=285 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=4 ttl=106 time=290 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=5 ttl=106 time=302 ms
^C
── inmobi.com ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 261.584/286.813/302.159/13.781 ms

┌──(root㉿error404)-[~]
└─# nslookup www.inmobi.com
Server:         192.168.8.1
Address:        192.168.8.1#53

Non-authoritative answer:
www.inmobi.com  canonical name = www.eastus.appgw.inmobi.com.
Name:   www.eastus.appgw.inmobi.com
Address: 20.81.69.107
```

B. Censys

Censys is a web application that has millions of hosts. We can simply type the Ip address host name in the search bar, and it will give all possible details about the host [23].

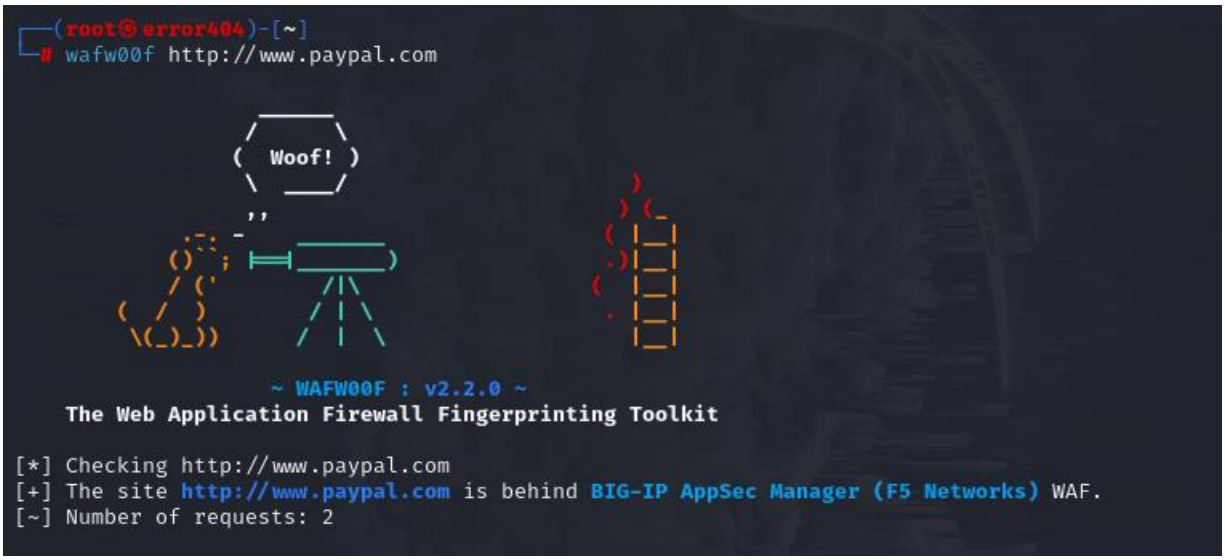


C. Wafw00f (Firewall detector)

This tool can be used to identify the firewall protection for the target web application. If the web application has a firewall protection (WAF enabled) it will show the type of WAF. Otherwise, don't show the result if it doesn't have WAF protection.

WAF protection is not enabled in our testing web application [24].

This site is protected by the firewall called BIG-IP AppSec Manager (F5 Networks) WAF.



D. Nmap

Nmap is a build in Linux too which can be used to identify network exploitation, host discovery and to do security auditing. It uses IP packets to identify the all the devices that are connected to the network & provide information about those devices and services that the system provides [25].

```
┌──(root㉿error404)-[~]
└─# ping inmobi.com
PING inmobi.com (20.81.69.107) 56(84) bytes of data.
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=1 ttl=106 time=283 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=2 ttl=106 time=262 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=3 ttl=106 time=265 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=4 ttl=106 time=265 ms
64 bytes from 20.81.69.107 (20.81.69.107): icmp_seq=5 ttl=106 time=262 ms
^C
── inmobi.com ping statistics ──
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 261.639/267.150/282.574/7.820 ms

┌──(root㉿error404)-[~]
└─# nmap -p80,443 -A -T4 20.81.69.107
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-06 22:16 +0530
Nmap scan report for 20.81.69.107
Host is up (0.034s latency).

PORT    STATE SERVICE   VERSION
80/tcp  open  http      Microsoft-Azure-Application-Gateway/v2
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 301 Moved Permanently
|     Server: Microsoft-Azure-Application-Gateway/v2
|     Date: Sat, 06 May 2023 16:46:36 GMT
|     Content-Type: text/html
|     Content-Length: 195
|     Connection: close
|     Location: http://www.inmobi.com/nice%20ports%2C/Tri%6Eity.txt%2ebak
|     <html>
|     <head><title>301 Moved Permanently</title></head>
|     <body>
|     <center><h1>301 Moved Permanently</h1></center>
|     <hr><center>Microsoft-Azure-Application-Gateway/v2</center>
|     </body>
|     </html>
|   GetRequest:
|     HTTP/1.1 301 Moved Permanently
|     Server: Microsoft-Azure-Application-Gateway/v2
|     Date: Sat, 06 May 2023 16:46:34 GMT
|     Content-Type: text/html
|     Content-Length: 195
|     Connection: close
|     Location: http://www.inmobi.com/
|     <html>
|     <head><title>301 Moved Permanently</title></head>
|     <body>
|     <center><h1>301 Moved Permanently</h1></center>
|     <hr><center>Microsoft-Azure-Application-Gateway/v2</center>
|     </body>
|     </html>
```

```
SF:eway/v2</center>\r\n</body>\r\n</html>\r\n");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge
Running: Oracle Virtualbox
OS CPE: cpe:/o:oracle:virtualbox
OS details: Oracle Virtualbox
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.39 ms 10.0.2.2
2   0.43 ms 20.81.69.107

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.95 seconds
```

```
┌──(root㉿error404)-[~]
└─# nmap inmobi.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-15 22:26 +0530
Nmap scan report for inmobi.com (20.81.69.107)
Host is up (0.032s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
25/tcp  open  smtp
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 25.21 seconds
```

ii. Subdomain Enumeration.

This is one of the most important parts in information gathering. In this step need to identify the sub domain that runs under the main root domain. So, by identifying those sub domains separately we can clearly go through each sub domain and can gather information's separately. To sub domain enumeration, I have used tools like,

A. Sublist3r.
B. Crt.sh.
C. Google dork.
D. Bug bounty Hunting Tool (BBHT).
E. Recon-ng

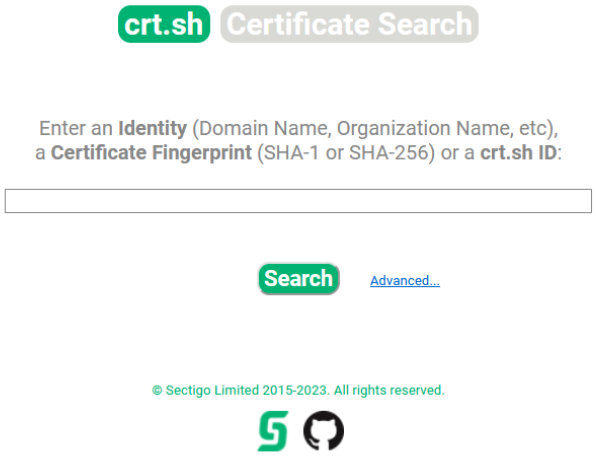
A. Sublist3r.

To show the workflow of these software I have used a web application that I used to find vulnerabilities. The site called https://www.inmobi.com/. this web application had a high risk of attacks.

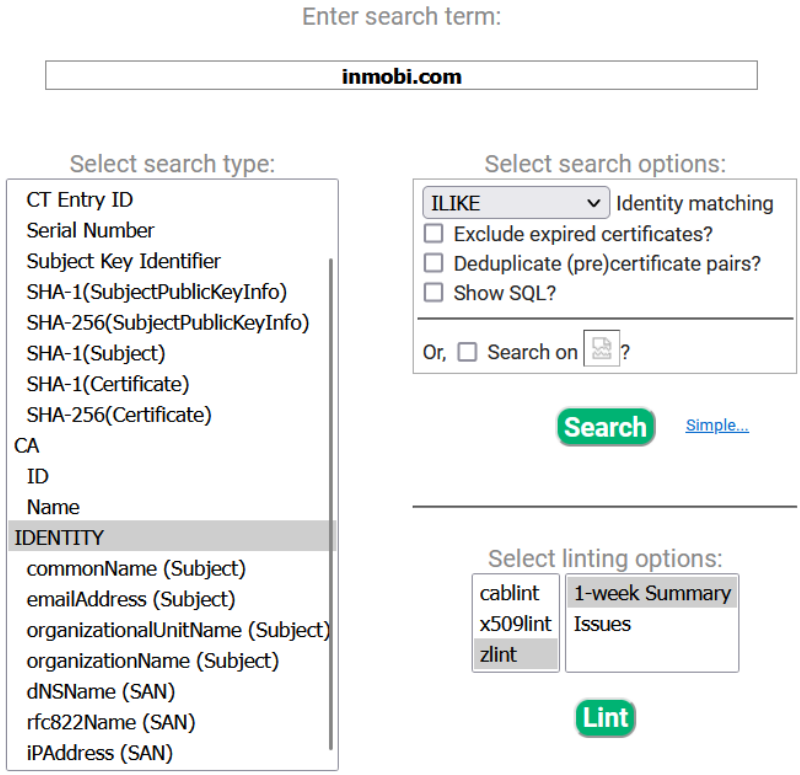
Up until sublis3t tools worked perfectly. From here onwards it started to give errors and not enumerating the sub domain anymore. So, this was a challenge. What I did is I skipped using this tool anymore and started to use other tools to enumerate sub domains.

B. Crt.sh

To find sub domain using this tool I used the same web applications domain and used a wild card to crawl more to find more sub domains.



Configuring the crawler.



Find sub domains. – some of sub domains are in the snapshot.

crt.sh Identity Search Group by Issuer

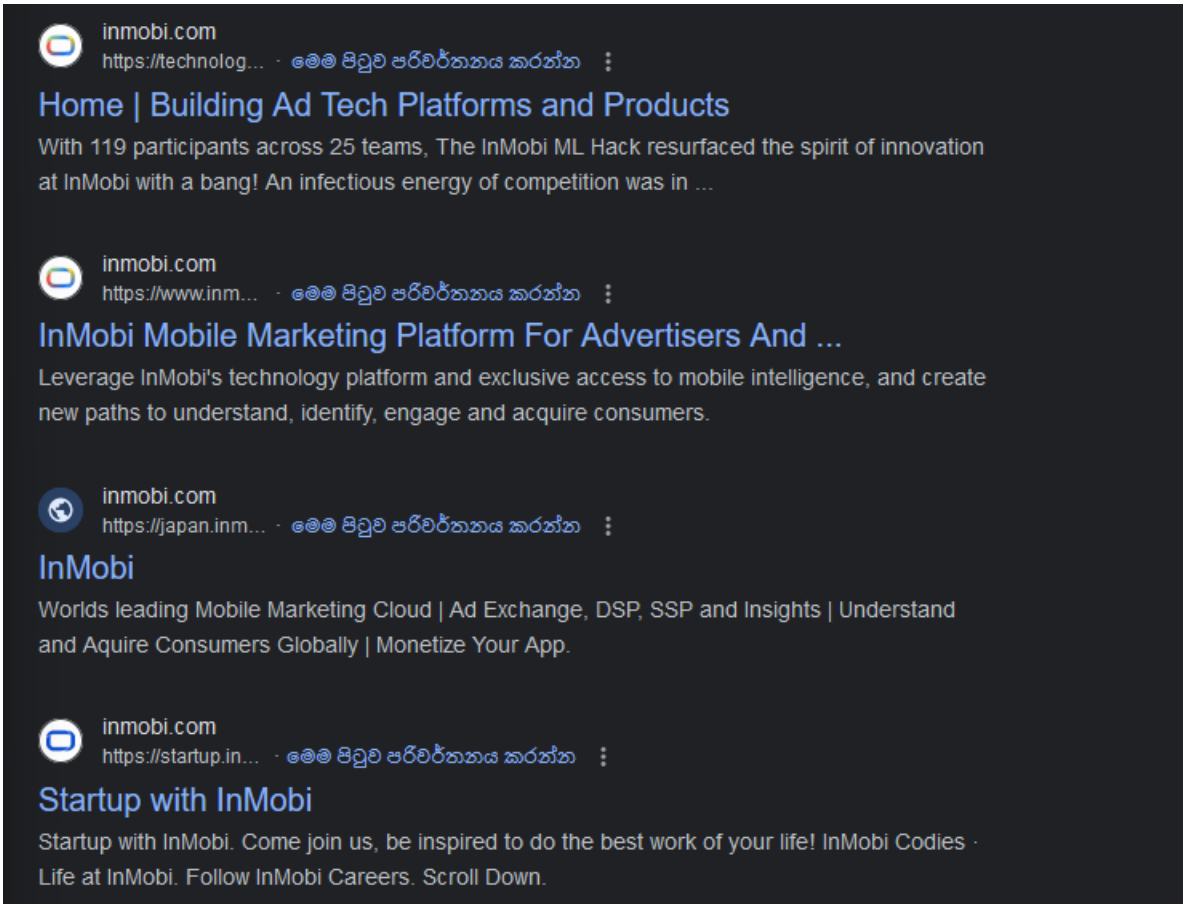
Criteria Type: Identity Match: ILIKE Search: 'inmobi.com'

Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	9313591578	2023-05-05	2023-05-05	2024-05-04	qa.geoservice.inmobi.com	qa.geoservice.inmobi.com www.qa.geoservice.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9313591725	2023-05-05	2023-05-05	2024-05-04	qa.geoservice.inmobi.com	qa.geoservice.inmobi.com www.qa.geoservice.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9305453355	2023-05-04	2023-05-04	2024-05-03	gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com	gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com www.gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo ECC Organization Validation Secure Server CA
	9305453180	2023-05-04	2023-05-04	2024-05-03	gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com	gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com www.gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo ECC Organization Validation Secure Server CA
	9285757441	2023-05-02	2023-05-02	2024-05-01	clarity.ssp.inmobi.com	clarity.ssp.inmobi.com www.clarity.ssp.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9285757216	2023-05-02	2023-05-02	2024-05-01	clarity.ssp.inmobi.com	clarity.ssp.inmobi.com www.clarity.ssp.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9278106307	2023-04-30	2023-04-30	2023-07-29	iff.inmobi.com	iff.inmobi.com post.iff.inmobi.com s.iff.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9269802477	2023-04-30	2023-04-30	2023-07-29	iff.inmobi.com	iff.inmobi.com post.iff.inmobi.com s.iff.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9278106728	2023-04-30	2023-04-30	2023-07-29	iff.inmobi.com	iff.inmobi.com post.iff.inmobi.com s.iff.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9269802512	2023-04-30	2023-04-30	2023-07-29	iff.inmobi.com	iff.inmobi.com post.iff.inmobi.com s.iff.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9275346449	2023-04-30	2023-04-29	2023-07-28	sso.glance.inmobi.com	sso.glance.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9266760058	2023-04-30	2023-04-29	2023-07-28	sso.glance.inmobi.com	sso.glance.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9274900437	2023-04-29	2023-04-29	2023-07-28	iam.auth.inmobi.com	iam.auth.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9265604205	2023-04-29	2023-04-29	2023-07-28	iam.auth.inmobi.com	iam.auth.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9252416496	2023-04-28	2023-04-28	2024-04-27	dashboard.dsp.inmobi.com	dashboard.dsp.inmobi.com www.dashboard.dsp.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo ECC Organization Validation Secure Server CA
	9252416561	2023-04-28	2023-04-28	2024-04-27	dashboard.dsp.inmobi.com	dashboard.dsp.inmobi.com www.dashboard.dsp.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo ECC Organization Validation Secure Server CA
	9261794708	2023-04-27	2023-04-27	2023-07-26	go.inmobi.com	go.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9247312258	2023-04-27	2023-04-27	2023-07-26	go.inmobi.com	go.inmobi.com	C=US, O=Let's Encrypt, CN=R3
	9244104576	2023-04-27	2023-04-27	2024-04-26	*.glance.inmobi.com	*.glance.inmobi.com glance.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9244104268	2023-04-27	2023-04-27	2024-04-26	*.glance.inmobi.com	*.glance.inmobi.com glance.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Organization Validation Secure Server CA
	9236078849	2023-04-26	2023-04-26	2024-04-25	astro.airflow-nonprod.inmobi.com	alertmanager.astro.airflow-nonprod.inmobi.com	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo ECC Organization

C. Google dork.

This is another tool that can be used to identify sub domains of a targeted domain. Also this can be called a simple technique that can use to identify sub domains using google search engine [26].

Simply by typing site: domain name. can identify all the sub domains under it. We can use a crawler to filter all those sub domains for perfect sorting.



D. Bug bounty Hunting Tool (BBHT).

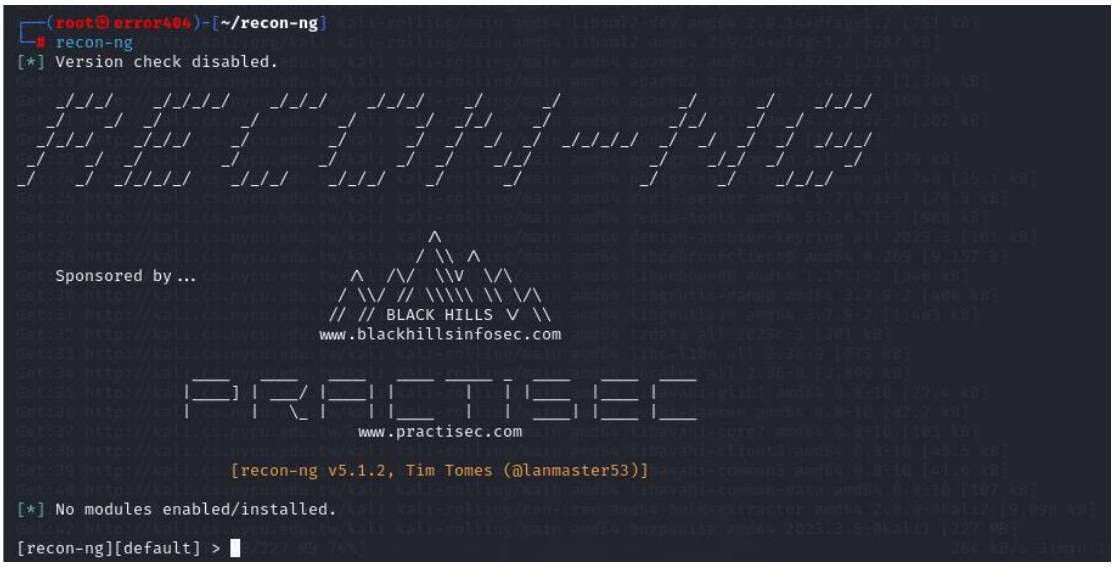
BBHT is a script that can be downloaded via various trusted web sites. It is an integrated tool that contains almost different types of tools that can be used in information gathering, identifying vulnerabilities, exploitation, and fixing.

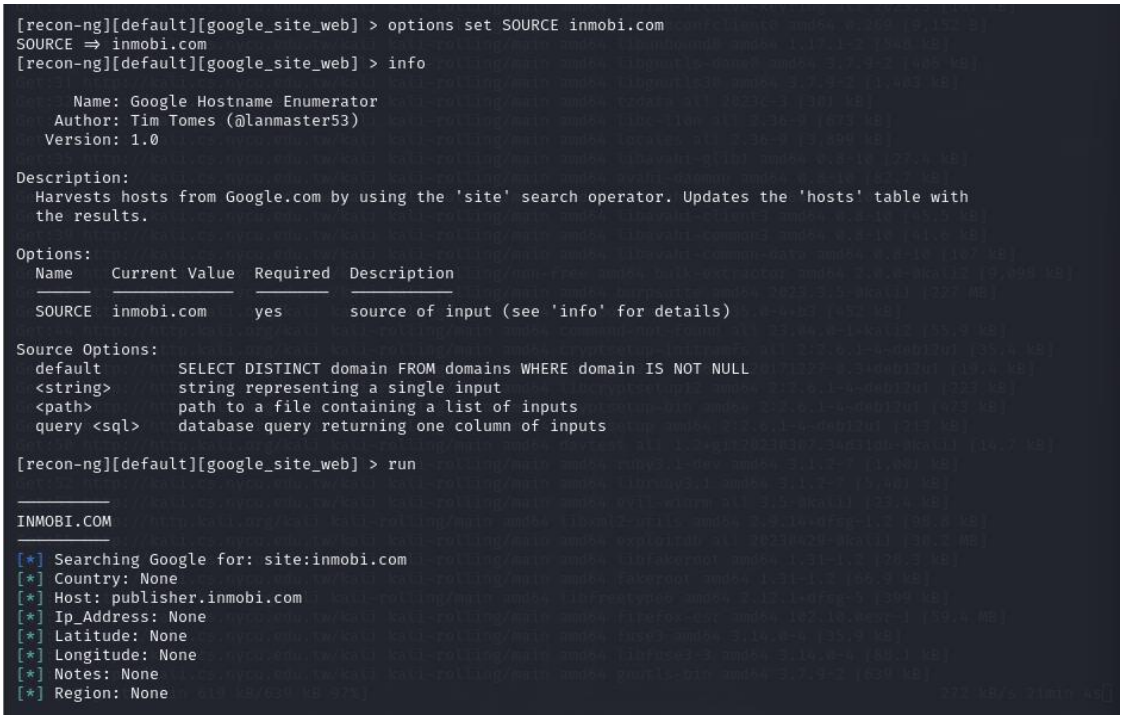
It has tools like SQL map- dev, sublist3r, virtual-host-directory, wpscan, webscreenshot, lazys3, JSParser, diresearch and so many [27].

```
┌──(root㉿error404)-[~/tools]
└─# ls
asnlookup  dirsearch  knock      lazys3   SecLists    Sublist3r          virtual-host-discovery
crtndstry  JSParser   lazyrecon  massdns  sqlmap-dev  teh_s3_bucketeers  wpscan
```

E. Recon-ng.

Recon-ng is also a well-known tool to identify sub domain of a target domain accurately. Also, it is very easy to use, and it will check sub domains in all well-known search engines.





Recon-ng found that there are total of 980 sub domains under the domain inmobi.com.

iii. Finding alive sub domain.

There can be thousands of sub domains under each domain. But in most case companies will change update their web application from time to time. When finding sub domain with automated tools it will go through each search engine and looks for unique sub domain that ends with *inmobi.com.

If the company updates their application, they might shutdown some of their sub domain as per their need. So, finding sub domain is not enough. After finding sub domain we need to gather the active sub domains. Because gathering information about inactive sub domains is not needed. Because they are no longer using sub domain anymore. To identify active sub domain, I have used a tool.

A. Httprobe.

A. Httprobe.

Htpprobe is a tool that can be used to identify alive sub domains. What is simply does is it send the request to all sub domains and if a sub domain gives a reply as success of communication establishment it will count that sub domain as active sub domain.

All sub domains.

```
┌──(root㉿error404)-[~]
└─# ./assetfinder -subs-only inmobi.com
qa.geoservice.inmobi.com
www.qa.geoservice.inmobi.com
gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com
www.gandalf-auth-grpc.staging-sea.glance-internal.inmobi.com
clarity.ssp.inmobi.com
www.clarity.ssp.inmobi.com
iff.inmobi.com
post.iff.inmobi.com
s.iff.inmobi.com
sso.glance.inmobi.com
iam.auth.inmobi.com
dashboard.dsp.inmobi.com
www.dashboard.dsp.inmobi.com
go.inmobi.com
glance.inmobi.com
glance.inmobi.com
alertmanager.astro.airflow-nonprod.inmobi.com
app.astro.airflow-nonprod.inmobi.com
astro.airflow-nonprod.inmobi.com
deployments.astro.airflow-nonprod.inmobi.com
grafana.astro.airflow-nonprod.inmobi.com
houston.astro.airflow-nonprod.inmobi.com
install.astro.airflow-nonprod.inmobi.com
kibana.astro.airflow-nonprod.inmobi.com
prometheus.astro.airflow-nonprod.inmobi.com
registry.astro.airflow-nonprod.inmobi.com
alertmanager.astro.astronomer-staging.inmobi.com
app.astro.astronomer-staging.inmobi.com
astro.astronomer-staging.inmobi.com
deployments.astro.astronomer-staging.inmobi.com
grafana.astro.astronomer-staging.inmobi.com
houston.astro.astronomer-staging.inmobi.com
install.astro.astronomer-staging.inmobi.com
kibana.astro.astronomer-staging.inmobi.com
prometheus.astro.astronomer-staging.inmobi.com
registry.astro.astronomer-staging.inmobi.com
i.w.eastus.trafficmanager.inmobi.com
et-wus.w.inmobi.com
mcg-ads.inmobi.com
www.mcg-ads.inmobi.com
litmus.inmobi.com
www.litmus.inmobi.com
idsp-test.cp.epsilon.tools.inmobi.com
www.idsp-test.cp.epsilon.tools.inmobi.com
staging.cp.epsilon.tools.inmobi.com
```

Active sub domains.

```
┌──(root㉿error404)-[~/httprobe]
└─# cat all.txt | httprobe >> alive.txt
```

```
┌──(root㉿error404)-[~/httprobe]
└─# cat alive.txt
https://s.iff.inmobi.com
https://post.iff.inmobi.com
https://iff.inmobi.com
https://iam.auth.inmobi.com
http://s.iff.inmobi.com
http://iam.auth.inmobi.com
http://post.iff.inmobi.com
http://iff.inmobi.com
https://dashboard.dsp.inmobi.com
https://go.inmobi.com
https://glance.inmobi.com
https://glance.inmobi.com
http://go.inmobi.com
http://glance.inmobi.com
http://glance.inmobi.com
http://dashboard.dsp.inmobi.com
https://sso.glance.inmobi.com
http://sso.glance.inmobi.com
https://i.w.eastus.trafficmanager.inmobi.com
https://mcg-ads.inmobi.com
http://i.w.eastus.trafficmanager.inmobi.com
http://mcg-ads.inmobi.com
https://et-wus.w.inmobi.com
https://idsp-test.cp.epsilon.tools.inmobi.com
https://dataexchange.inmobi.com
http://idsp-test.cp.epsilon.tools.inmobi.com
http://et-wus.w.inmobi.com
https://gtv.analytics.glance.inmobi.com
https://cloudengg-test-cdn.inmobi.com
https://analytics.glance.inmobi.com
https://curated.inmobi.com
http://gtv.analytics.glance.inmobi.com
https://w.inmobi.com
http://analytics.glance.inmobi.com
https://analytics.glance.inmobi.com
https://w.inmobi.com
https://analytics.glance.inmobi.com
http://analytics.glance.inmobi.com
http://w.inmobi.com
```

```
81 http://api.glance.inmobi.com
82 http://intelligence.inmobi.com
83 http://staging.iam.inmobi.com
84 https://unif-id.ssp.inmobi.com
85 http://eus.r.inmobi.com
86 http://unif-id.ssp.inmobi.com
```

Httprobe tool identified that there are **86 sub domains** currently active from a **total of 980 sub domains.**

iv. Finding achieved information.

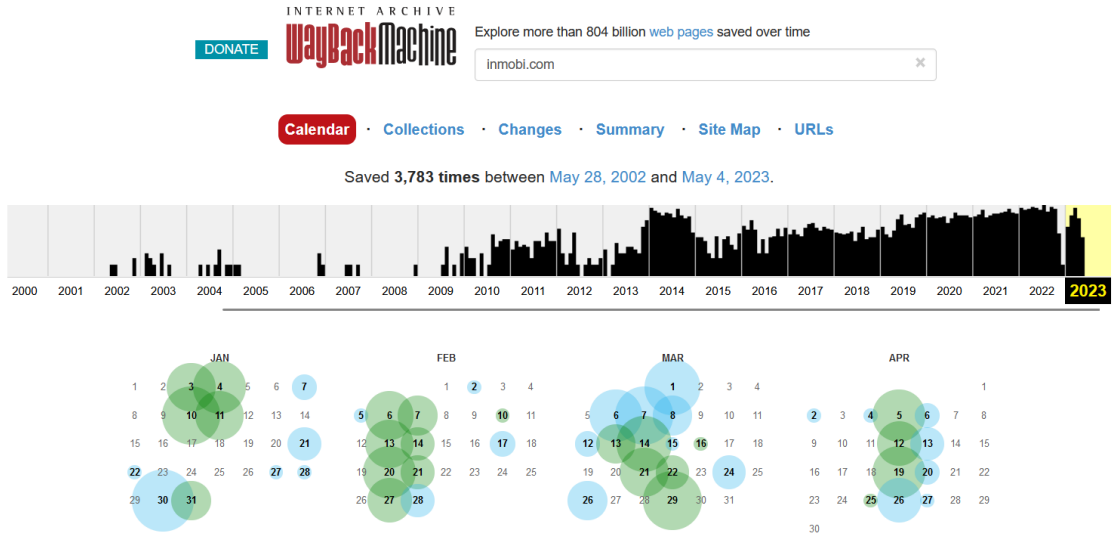
Finding older information about the application is an important part of information gathering. By finding those achieved information we can find what kind of vulnerabilities that had now what they did to fix those as so many like these.

To find those achieved information I have used an internet archive.

A. Way Back Machine.

A. Way back Machine.

This is a web tool that is simply an internet archive. It has a massive collection of snapshots of web applications, copies of its web pages, books, videos, audios, images, etc [28].



Calendar · Collections · Changes · Summary · Site Map · URLs

host inmobi.com

Indexed on January 21, 2023.

Saved **3,783 times** between May 28, 2002 and May 4, 2023.

MIME-types: Any MIME-types | Year Start: 2002 | Year End: 2023

All text image application video

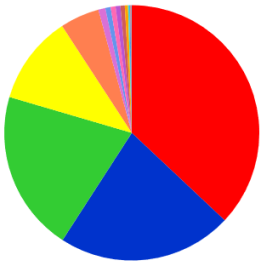
Summary on MIME-types Count

Quick search on MIME-types...

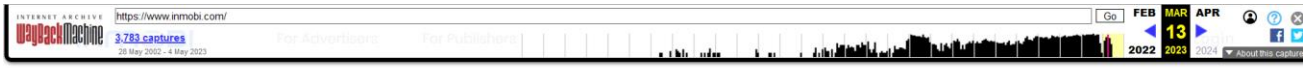
<< < 1 2 > >>

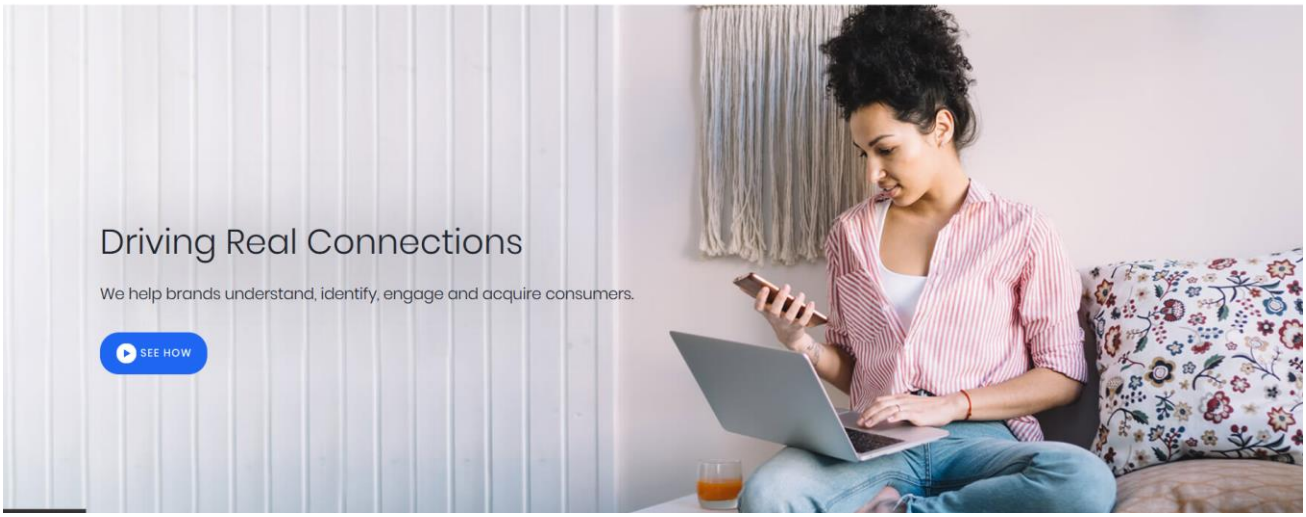
	Captures	URLs	New URLs
text/html	139,390	29,224	18,567
text/css	83,326	35,241	27,358
image/png	76,826	5,373	2,444
image/jpeg	42,422	3,679	1,968
application/javascript	18,757	1,047	891
image/gif	3,352	756	594
image/vnd.microsoft.icon	2,508	13	2

Captures



View of the inmobi.com web application in 2023 April.





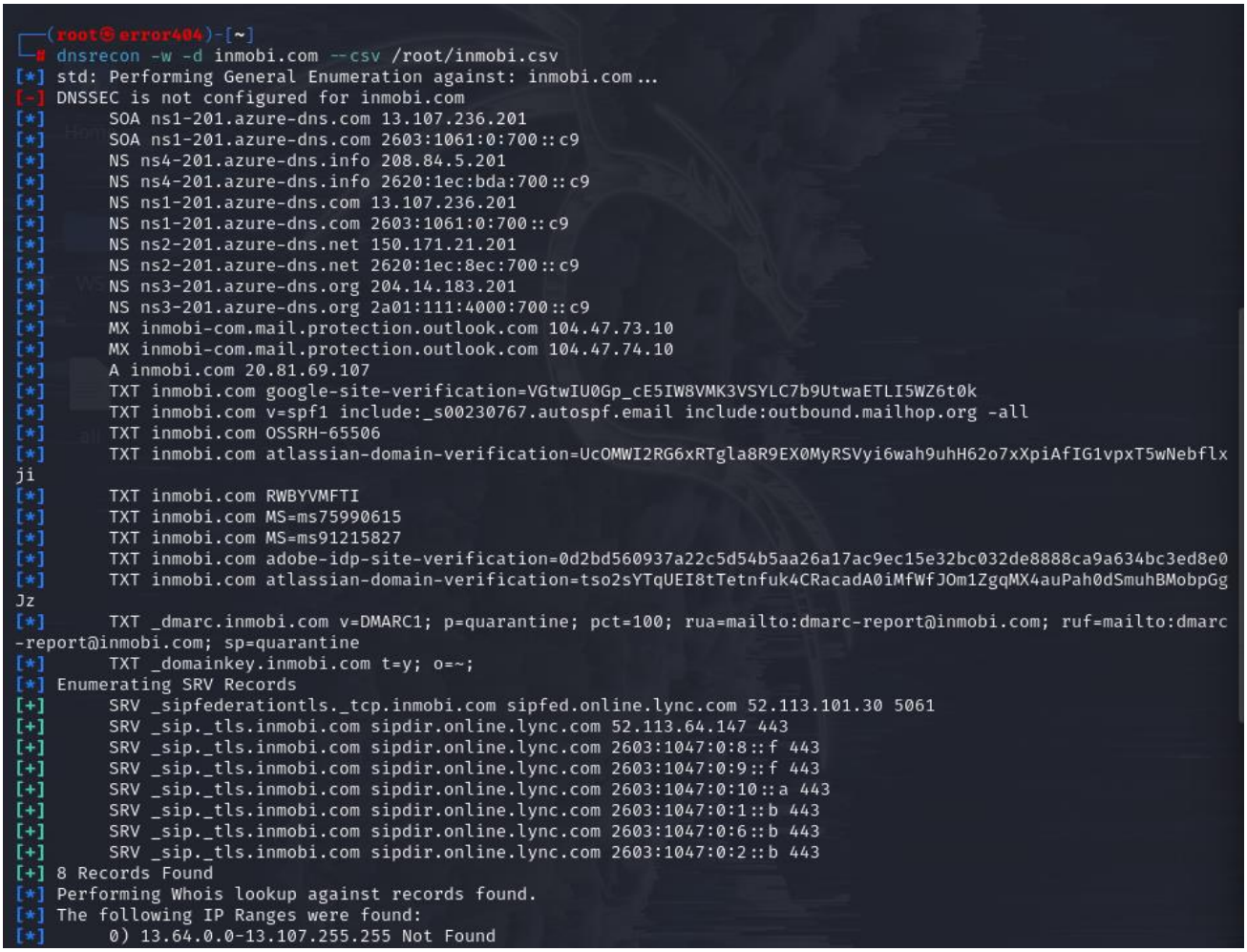
v. DNS enumeration.

In this phase what simply doing in identifying all possible DNS servers and their matching entries of the targeted organization or web application. By doing this we can reveal the size of the target application and can measure the size of the attack surface easily. To do this enumeration I have used many tools.

A. Dnsrecon
B. DNSEnum
C. Host Command

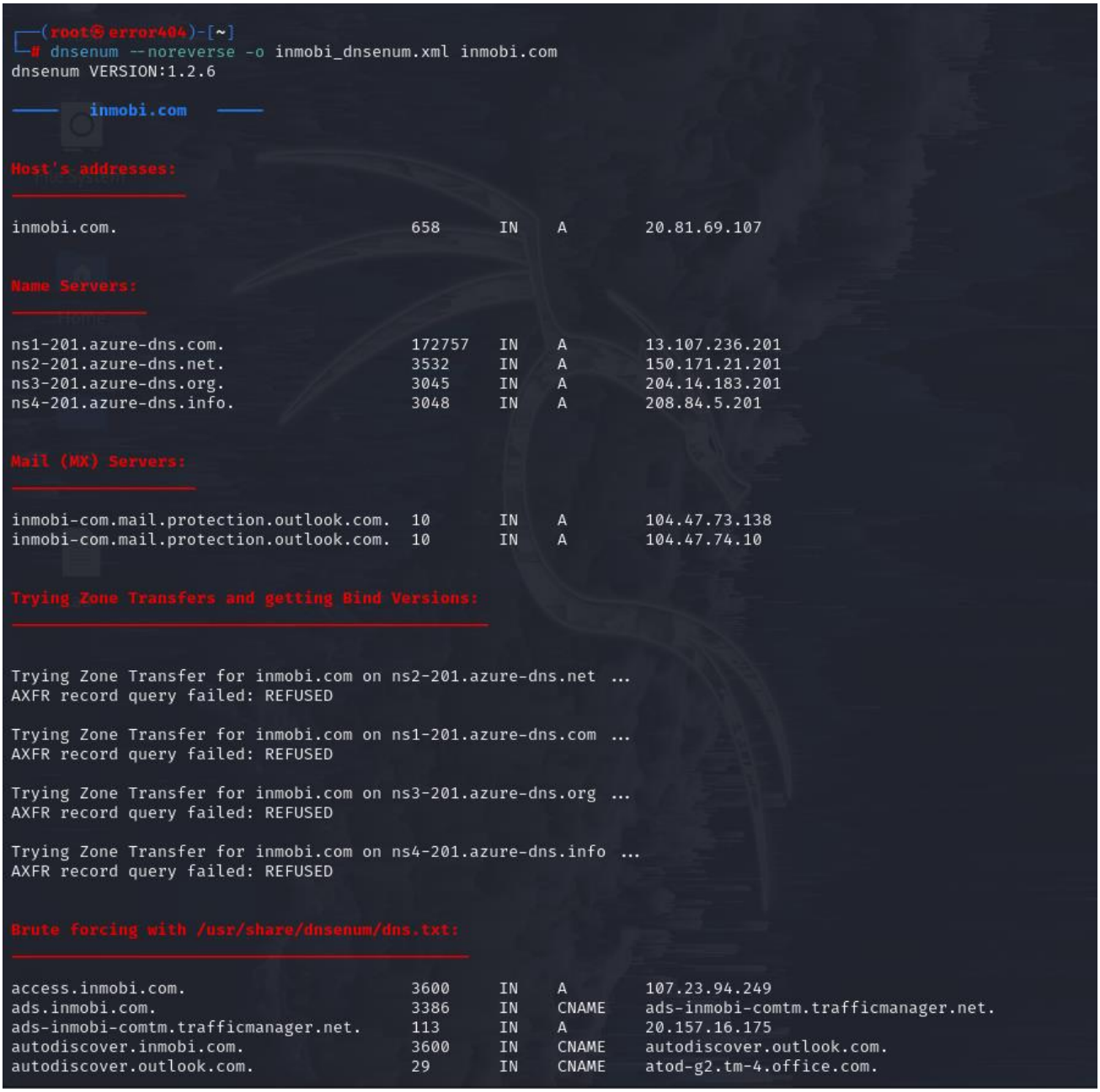
A. Dnsrecon.

Dnsrecon is a powerful tool that cam as a build in tool in kali Linux. By using this tool we can reverse look up an IP range, domain brute force enumeration, cache snooping against name servers, standard records enumeration, etc.



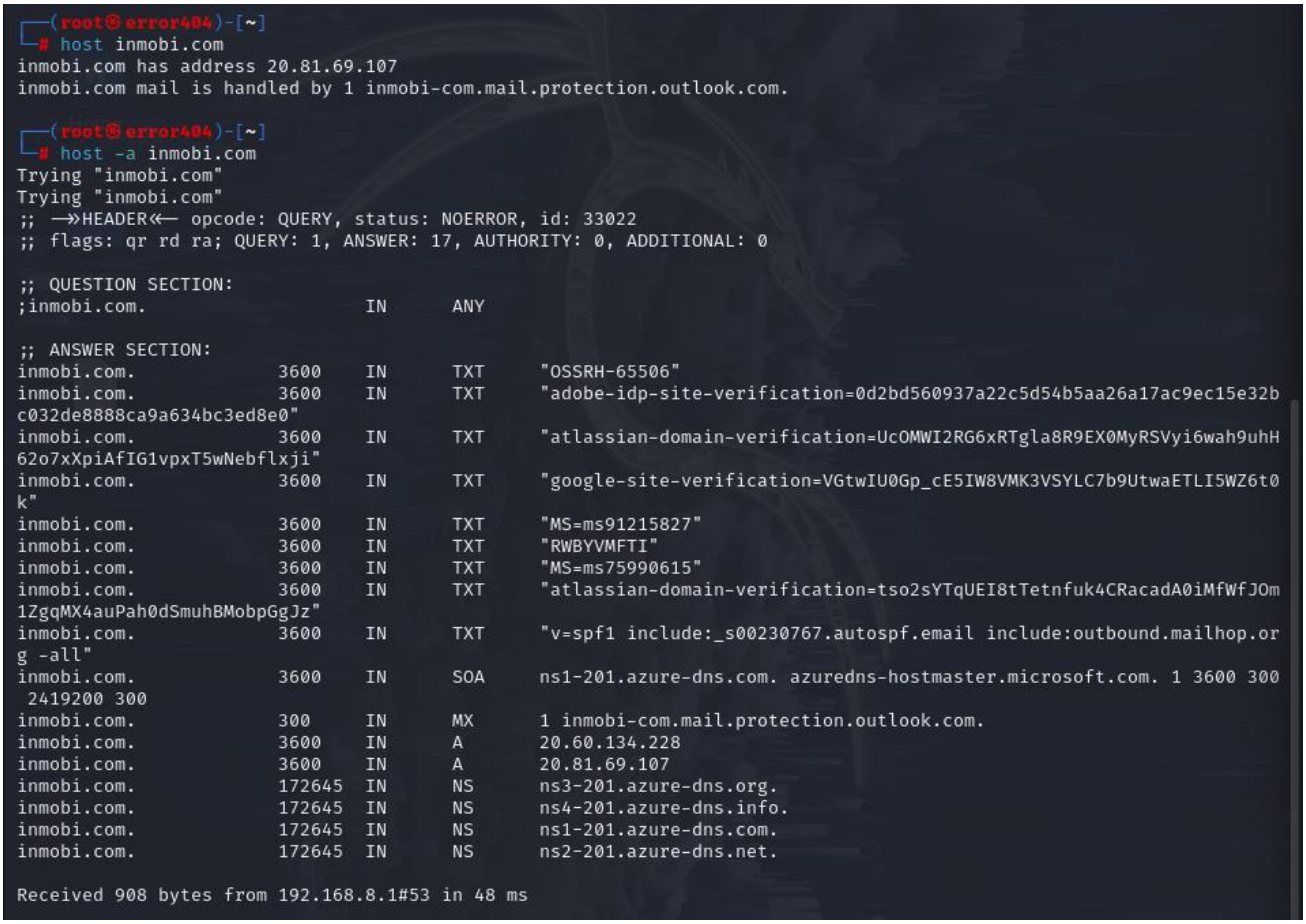
B. DNSEnum.

This also an inbuild Linux tool that can be used to identify DNS records such as MX, mail exchange servers, DNS servers and address records of a domain.



C. Host Command.

This is also a simple built-in feature which we can identify above-mentioned segments of the target host.



```
┌──(root㉿error404)-[~]
└─# host inmobi.com
inmobi.com has address 20.81.69.107
inmobi.com mail is handled by 1 inmobi-com.mail.protection.outlook.com.

┌──(root㉿error404)-[~]
└─# host -a inmobi.com
Trying "inmobi.com"
Trying "inmobi.com"
;; ⇾⇾HEADER⇽⇽ opcode: QUERY, status: NOERROR, id: 33022
;; flags: qr rd ra; QUERY: 1, ANSWER: 17, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;inmobi.com.                    IN      ANY

;; ANSWER SECTION:
inmobi.com.             3600    IN      TXT     "OSSRH-65506"
inmobi.com.             3600    IN      TXT     "adobe-idp-site-verification=0d2bd560937a22c5d54b5aa26a17ac9ec15e32b
c032de8888ca9a634bc3ed8e0"
inmobi.com.             3600    IN      TXT     "atlassian-domain-verification=UcOMWI2RG6xRTgla8R9EX0MyRSVyi6wah9uhH
62o7xXpiAfIG1vpxT5wNebflxji"
inmobi.com.             3600    IN      TXT     "google-site-verification=VGtwIU0Gp_cE5IW8VMK3VSYLC7b9UtwaETLI5WZ6t0
k"
inmobi.com.             3600    IN      TXT     "MS=ms91215827"
inmobi.com.             3600    IN      TXT     "RWBYVMFTI"
inmobi.com.             3600    IN      TXT     "MS=ms75990615"
inmobi.com.             3600    IN      TXT     "atlassian-domain-verification=tso2sYTqUEI8tTetnfuk4CRacadA0iMfWfJOm
1ZgqMX4auPah0dSmuhBMobpGgJz"
inmobi.com.             3600    IN      TXT     "v=spf1 include:_s00230767.autospf.email include:outbound.mailhop.or
g -all"
inmobi.com.             3600    IN      SOA     ns1-201.azure-dns.com. azuredns-hostmaster.microsoft.com. 1 3600 300
 2419200 300
inmobi.com.             300     IN      MX      1 inmobi-com.mail.protection.outlook.com.
inmobi.com.             3600    IN      A       20.60.134.228
inmobi.com.             3600    IN      A       20.81.69.107
inmobi.com.             172645  IN      NS      ns3-201.azure-dns.org.
inmobi.com.             172645  IN      NS      ns4-201.azure-dns.info.
inmobi.com.             172645  IN      NS      ns1-201.azure-dns.com.
inmobi.com.             172645  IN      NS      ns2-201.azure-dns.net.

Received 908 bytes from 192.168.8.1#53 in 48 ms
```

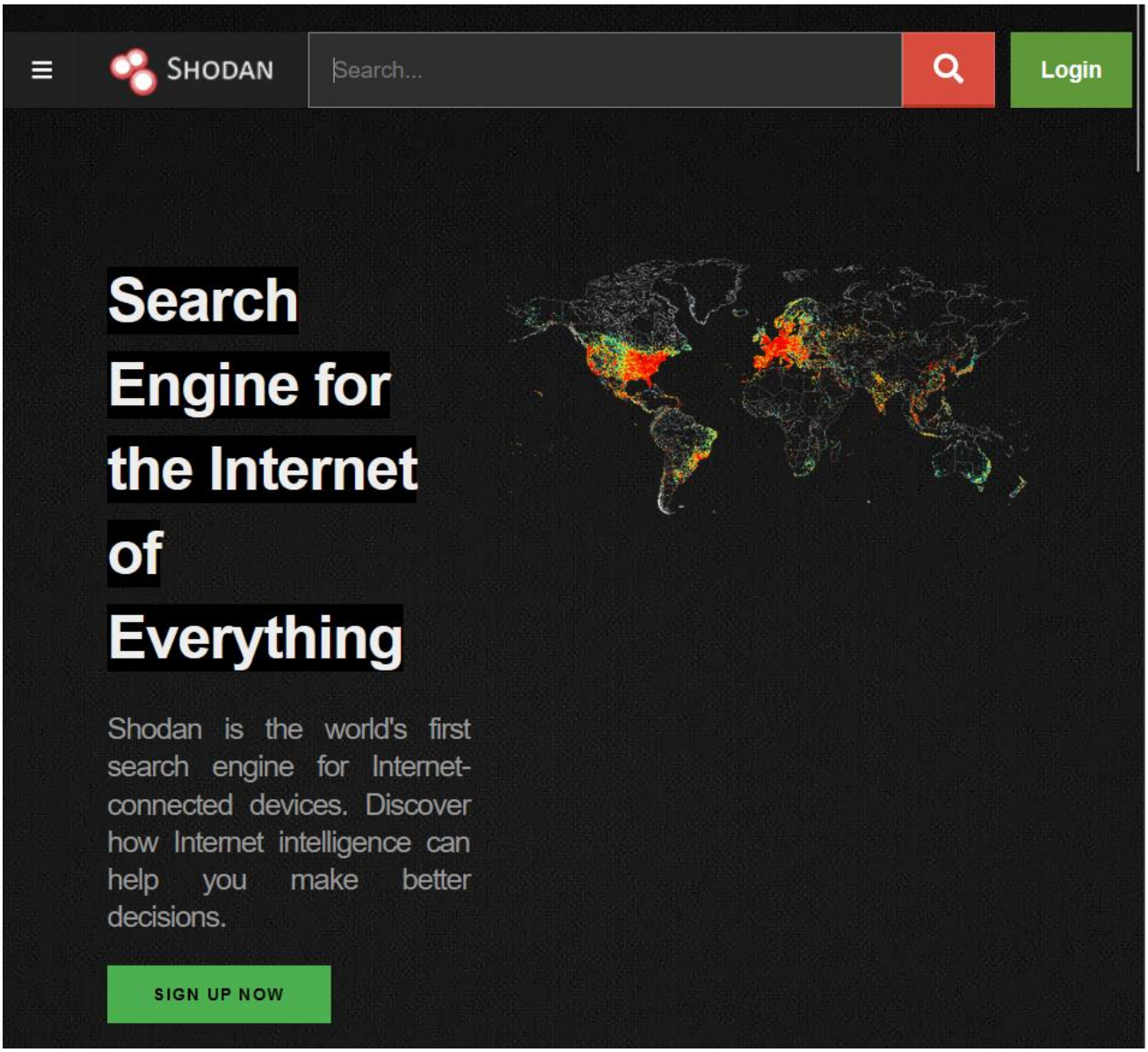
vi. Public Device Enumeration.

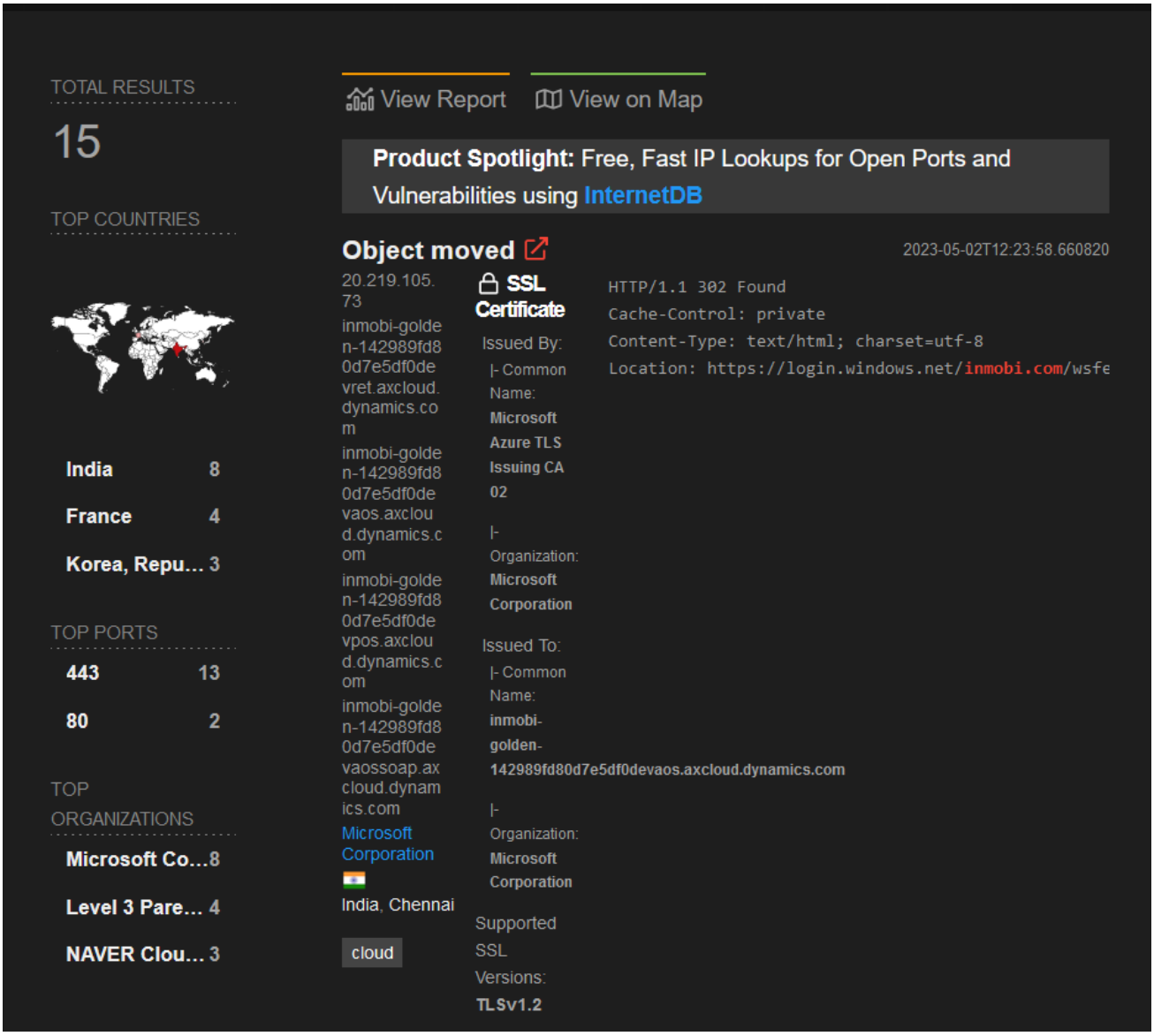
The purpose of this step is to identify all devices that are attached to the system or web application. To this step, I have used one tool. From this tool it will give all connected devices information in the targeted domain.

A. Shodan.io

A. Shodan.io

By this tool if there are any public IP addresses exposing a server on a certain port it will be available on this web site. It also shows details about web servers, banners, ISP, SSH, and FTP [29].




TOTAL RESULTS
15
TOP COUNTRIES
India 8
France 4
Korea, Repu... 3
TOP PORTS
443 13
80 2
TOP ORGANIZATIONS
Microsoft Co... 8
Level 3 Pare... 4
NAVER Clou... 3
View Report
View on Map
Product Spotlight: Free, Fast IP Lookups for Open Ports and Vulnerabilities using InternetDB
Object moved
2023-05-02T12:23:58.660820
20.219.105.73
inmobi-golden-142989fd80d7e5df0devret.axcloud.dynamics.com
inmobi-golden-142989fd80d7e5df0devaos.axcloud.dynamics.com
inmobi-golden-142989fd80d7e5df0devpos.axcloud.dynamics.com
inmobi-golden-142989fd80d7e5df0devaossoap.axcloud.dynamics.com
Microsoft Corporation
India, Chennai
cloud
SSL Certificate
Issued By:
|- Common Name: Microsoft Azure TLS Issuing CA 02
|- Organization: Microsoft Corporation
Issued To:
|- Common Name: inmobi-golden-142989fd80d7e5df0devaos.axcloud.dynamics.com
|- Organization: Microsoft Corporation
Supported SSL Versions: TLSv1.2
HTTP/1.1 302 Found
Cache-Control: private
Content-Type: text/html; charset=utf-8
Location: https://login.windows.net/inmobi.com/wsfe


Status 409
2023-05-02T04:46:28.694675
207.120.36.104
sni-missing-or-domain-unknown.help.section.io
www.sni-missing-or-domain-unknown.help.section.io
Level 3 Parent, LLC
France, Paris
SSL Certificate
Issued By:
|- Common Name: Sectigo RSA Domain Validation Secure Server CA
|- Organization: Sectigo Limited
Issued To:
|- Common Name: sni-missing-or-domain-unknown.help.section.io
Supported SSL Versions: TLSv1.2, TLSv1.3
HTTP/1.1 409 Conflict
Date: Tue, 02 May 2023 04:46:28 GMT
Content-Type: text/html
Content-Length: 26622
Connection: keep-alive
ETag: "644b4060-67fe"
section-ingress: not-configured
section-io-id: 5c932618e20a7f7c76786682532ddd8c

vii. Finding structure of the file system.

By using these tools, we can identify the structure of the file system. This is a very important step to take in the information gathering phase before moving into vulnerability scanning in the target domain. I have used many tools in this step-in order to find the correct structure of the file system.

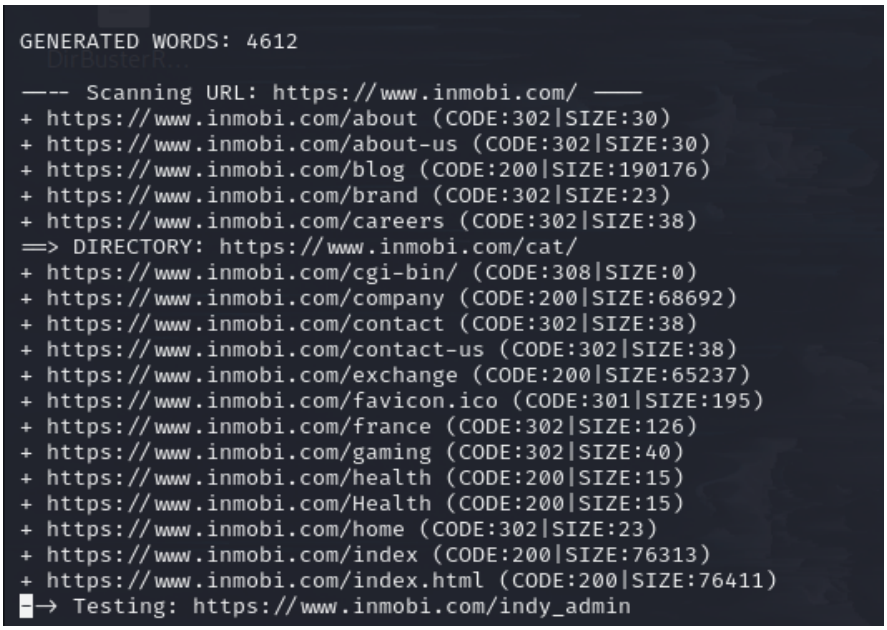
A. Dirsearch.
B. Dirb Tool.
C. OWASP dirbuster.

A. Diresearch.

This is also a very famous tool that is used to brute force directories and files on websites. I have used this tool to find hidden files of web application, this tool will find those hidden files and directories on a web server by sending HTTP requests to the server and then analyze the response.

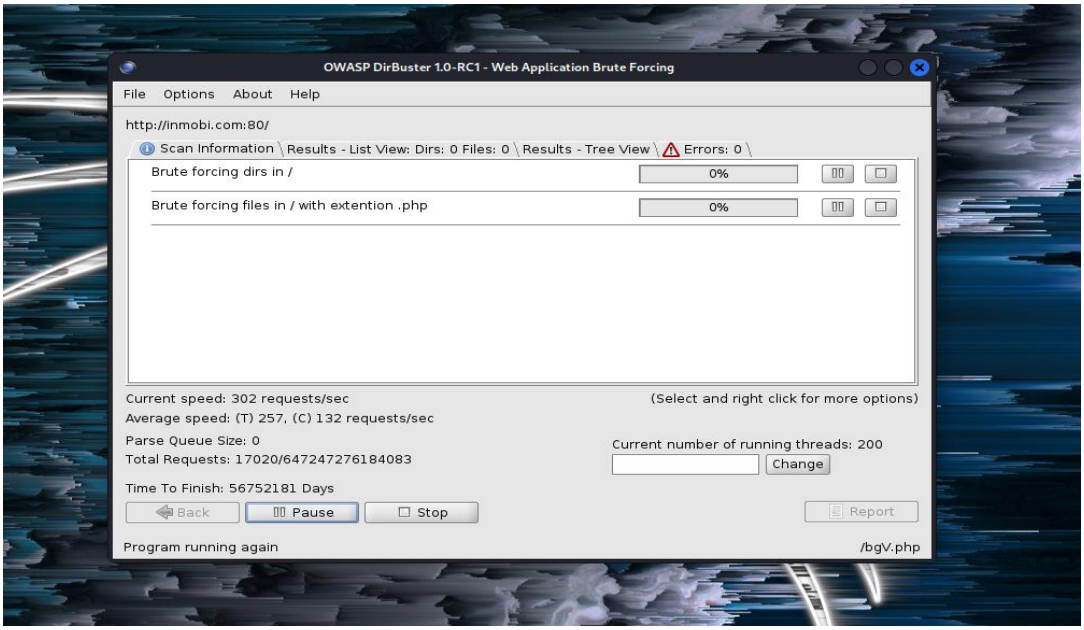
B. Dirb Tool.

A Dirb tool is a tool that can be used to identify the web contents of specific web sites. It will look for hidden or existing web objects of the web application. Dirb will identify those web content by launching directory attack against a web server and analyzing the requests and its response. Using this tool, we can simply identify all the contents that are available in the target domain.



C. OWASP dirbuster.

OWASP dirbuster is a another well know Linux built in tool that can be used to brute force directories and file names on the web application servers. Dirbuster comes with total of 9 different lists, this makes dirbuster extremely effective at finding those hidden files and directories of target domain.



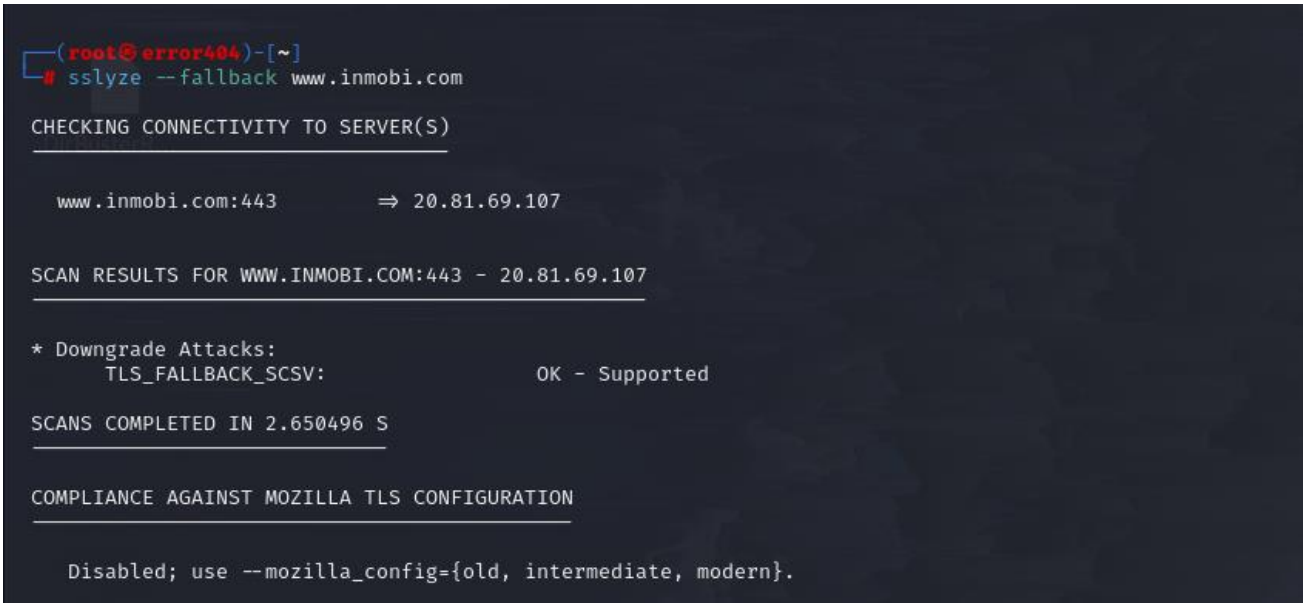
b. Vulnerability analyzing tools.

Under the vulnerability phase many tools are used to identify vulnerabilities and confirm is existence using various automated tools as well and using frameworks. There are lots of automated tools as well as manual testing tools available. I have used some of the most recognized tools for vulnerability gathering phase. For the vulnerability analysis I have used automated tools as well as manual testing tools for the confirmation of existence of vulnerabilities.

A. SSLyze.
B. Commix.
C. Crlf.
D. Corsy.
E. XSStrike.
F. Httpsmuggler.
G. Oralyzer.
H. Burp suite.
I. Nikto scan.
J. Net sparker.
K. OWASP zap.

A. SSLyze.

SSLyze is a tool which can be used to enumerate the SSL/TLS configuration of a particular server after connecting to it. Various issues such as bad certificates, weak cipher suites, Heartbleed, ROBOT can be identified with SSLyze very easily.



```
┌──(root㉿error404)-[~]
└─# sslyze --compression www.inmobi.com

CHECKING CONNECTIVITY TO SERVER(S)
──────────────────────────────────

  www.inmobi.com:443        ⇒ 20.81.69.107

SCAN RESULTS FOR WWW.INMOBI.COM:443 - 20.81.69.107
──────────────────────────────────────────────────

* Deflate Compression:
                                       OK - Compression disabled

SCANS COMPLETED IN 2.792443 S
─────────────────────────────

COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION
────────────────────────────────────────────

   Disabled; use --mozilla_config={old, intermediate, modern}.

┌──(root㉿error404)-[~]
└─#
```

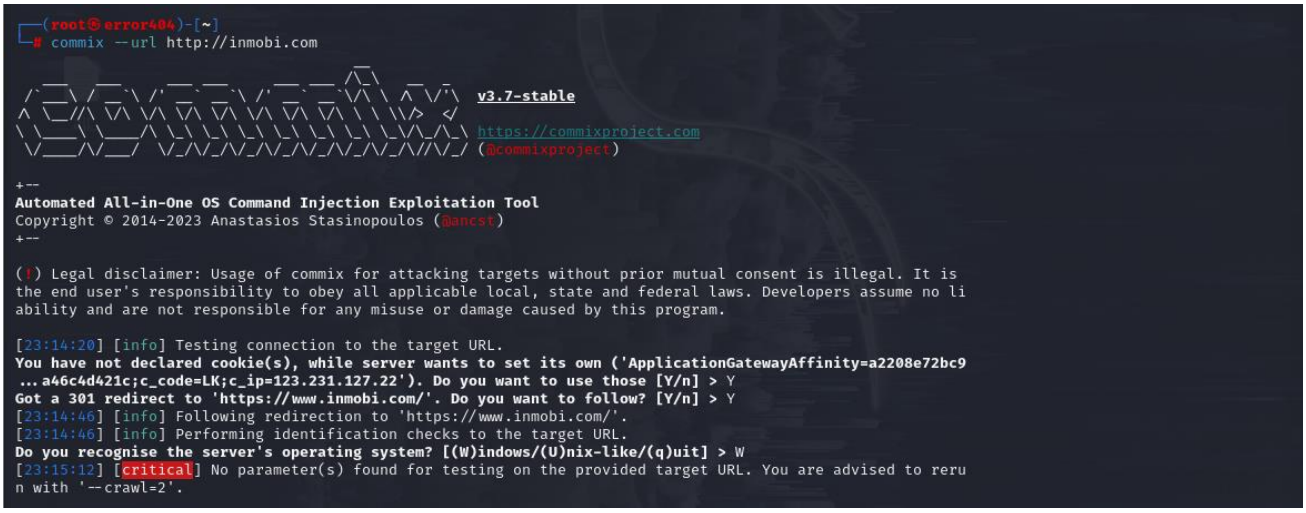
```
┌──(root㉿error404)-[~]
└─# sslyze --openssl_ccs www.inmobi.com

CHECKING CONNECTIVITY TO SERVER(S)
──────────────────────────────────

  www.inmobi.com:443        ⇒ 20.81.69.107

SCAN RESULTS FOR WWW.INMOBI.COM:443 - 20.81.69.107
──────────────────────────────────────────────────

* OpenSSL CCS Injection:
                                    OK - Not vulnerable to OpenSSL CCS injection

SCANS COMPLETED IN 2.562274 S
─────────────────────────────

COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION
────────────────────────────────────────────

   Disabled; use --mozilla_config={old, intermediate, modern}.
```

```
┌──(root㉿error404)-[~]
└─# sslyze --certinfo www.inmobi.com

 CHECKING CONNECTIVITY TO SERVER(S)
 ----------------------------------

   www.inmobi.com:443        ⇒ 20.81.69.107


 SCAN RESULTS FOR WWW.INMOBI.COM:443 - 20.81.69.107
 --------------------------------------------------

 * Certificates Information:
       Hostname sent for SNI:             www.inmobi.com
       Number of certificates detected:   1


     Certificate #0 ( _RSAPublicKey )
       SHA1 Fingerprint:                  578ea937dd581b7af71c8d9e59174756112d36cf
       Common Name:                       inmobi.com
       Issuer:                            Sectigo RSA Organization Validation Secure Server CA
       Serial Number:                     57877993682009893567261812994829022324
       Not Before:                        2022-11-03
       Not After:                         2023-11-03
       Public Key Algorithm:              _RSAPublicKey
       Signature Algorithm:               sha256
       Key Size:                          2048
       Exponent:                          65537
       SubjAltName - DNS Names:           ['inmobi.com', 'china.inmobi.com', 'indonesia.inmobi.com', 'japan.inmobi.c
om', 'korea.inmobi.com', 'www.inmobi.com']

     Certificate #0 - Trust
       Hostname Validation:               OK - Certificate matches server hostname
       Android CA Store (13.0.0_r9):      OK - Certificate is trusted
       Apple CA Store (iOS 16, iPadOS 16, macOS 13, tvOS 16, and watchOS 9):OK - Certificate is trusted
       Java CA Store (jdk-13.0.2):        OK - Certificate is trusted
       Mozilla CA Store (2022-12-11):     OK - Certificate is trusted
       Windows CA Store (2023-02-19):     OK - Certificate is trusted
       Symantec 2018 Deprecation:         OK - Not a Symantec-issued certificate
       Received Chain:                    inmobi.com ⟶ AAA Certificate Services ⟶ USERTrust RSA Certification Au
thority ⟶ Sectigo RSA Organization Validation Secure Server CA
       Verified Chain:                    inmobi.com ⟶ Sectigo RSA Organization Validation Secure Server CA ⟶ US
ERTrust RSA Certification Authority
       Received Chain Contains Anchor:    OK - Anchor certificate not sent
       Received Chain Order:              FAILED - Certificate chain out of order!
       Verified Chain contains SHA1:      OK - No SHA1-signed certificate in the verified certificate chain

     Certificate #0 - Extensions
       OCSP Must-Staple:                  NOT SUPPORTED - Extension not found
       Certificate Transparency:          OK - 3 SCTs included

     Certificate #0 - Extensions
       OCSP Must-Staple:                  NOT SUPPORTED - Extension not found
       Certificate Transparency:          OK - 3 SCTs included

     Certificate #0 - OCSP Stapling
       OCSP Response Status:              SUCCESSFUL
       Validation w/ Mozilla Store:       OK - Response is trusted
       Responder Key Hash:                b"\x17\xd9\xd6%'g\xf91\xc2IC\xd906D\x8cl\xa90\xeb"
       Cert Status:                       GOOD
       Cert Serial Number:                57877993682009893567261812994829022324
       This Update:                       2023-05-13
       Next Update:                       2023-05-20

 SCANS COMPLETED IN 3.506112 S
 -----------------------------

 COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION
 --------------------------------------------

    Disabled; use --mozilla_config={old, intermediate, modern}.
```

```
┌──(root㉿error404)-[~]
└─# sslyze --http_headers www.inmobi.com

 CHECKING CONNECTIVITY TO SERVER(S)
 ----------------------------------

   www.inmobi.com:443        ⇒ 20.81.69.107


 SCAN RESULTS FOR WWW.INMOBI.COM:443 - 20.81.69.107
 --------------------------------------------------

 * HTTP Security Headers:
     Strict-Transport-Security Header
       Max Age:                         15552000
       Include Subdomains:              True
       Preload:                         False

 SCANS COMPLETED IN 3.601933 S
 -----------------------------

 COMPLIANCE AGAINST MOZILLA TLS CONFIGURATION
 --------------------------------------------

    Disabled; use --mozilla_config={old, intermediate, modern}.
```

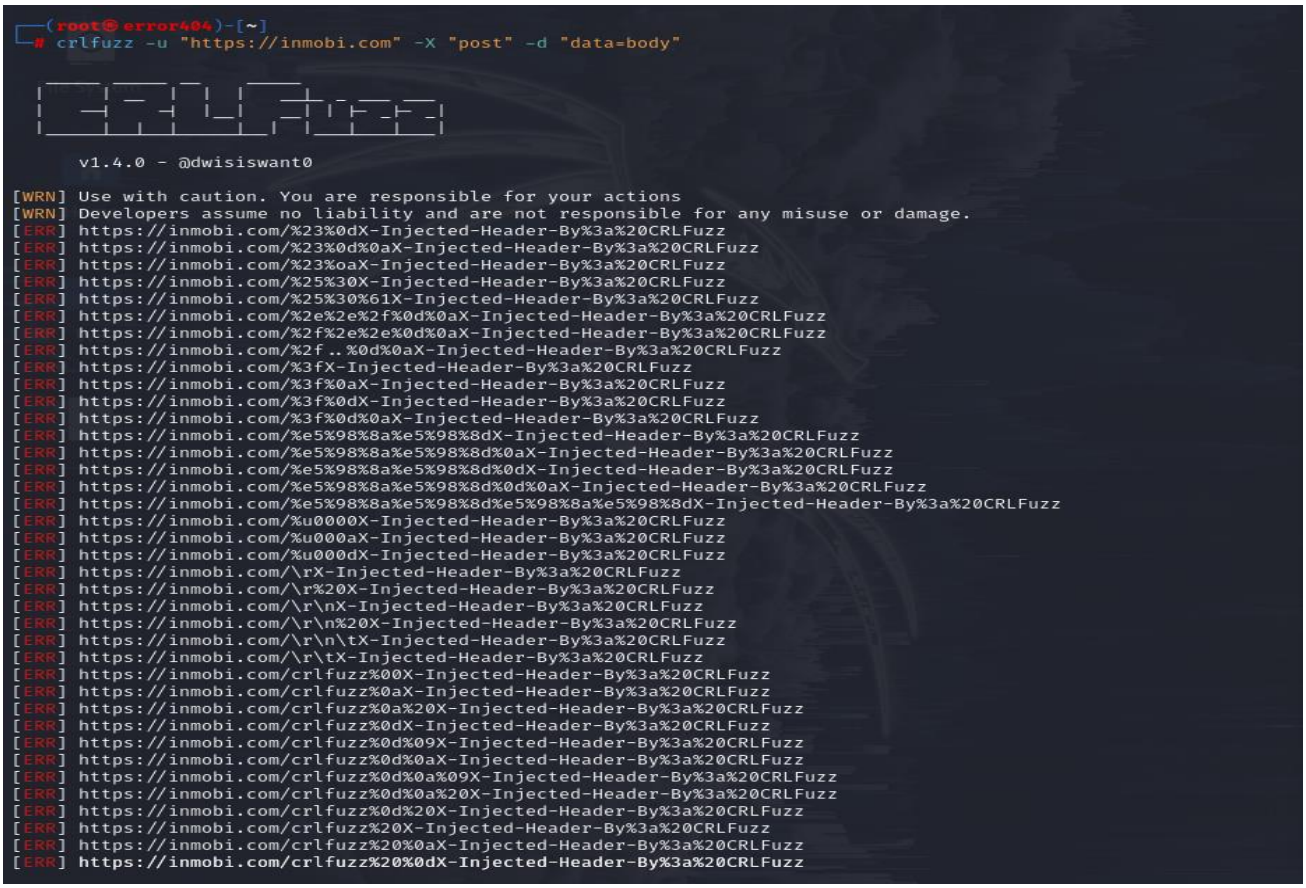
B. Commix.

As there is an underlying operating system which hosts the target website, we may able to run OS commands related to that OS through the web application. Commix is a tool which can be used to identify potential OS command injection vulnerabilities. Commix scan for our target can be performed as follows.



C. Crlf.

Carriage Return and Line Feed Injection (CRLF) vulnerability occurs when an attacker tries to inject carriage return characters to software applications where it is not expected. The python tool CRLF Injector Scanner can be used to test this vulnerability.



D. Corsy.

some HTTP headers that define trusted web origins and associated properties are used by the CORS protocol to allow access from other resources of the website such as subdomains and trusted third parties. Sensitive information may be disclosed to attackers by exploiting vulnerabilities related to CORS. Corsy is a python tool which can be used to detect CORS misconfigurations.

```
┌──(root㉿error404)-[~/Corsy]
└─# python3 corsy.py -u https://inmobi.com

    C O R S Y   {v1.0-beta}

- No misconfigurations found.
```

E. XSStrike.

This tool can be sued to identify the XSS vulnerability's if presents on the target web application. There are many types of XSS attacks. All these attack types can be tested using the python tool called XSStrike,

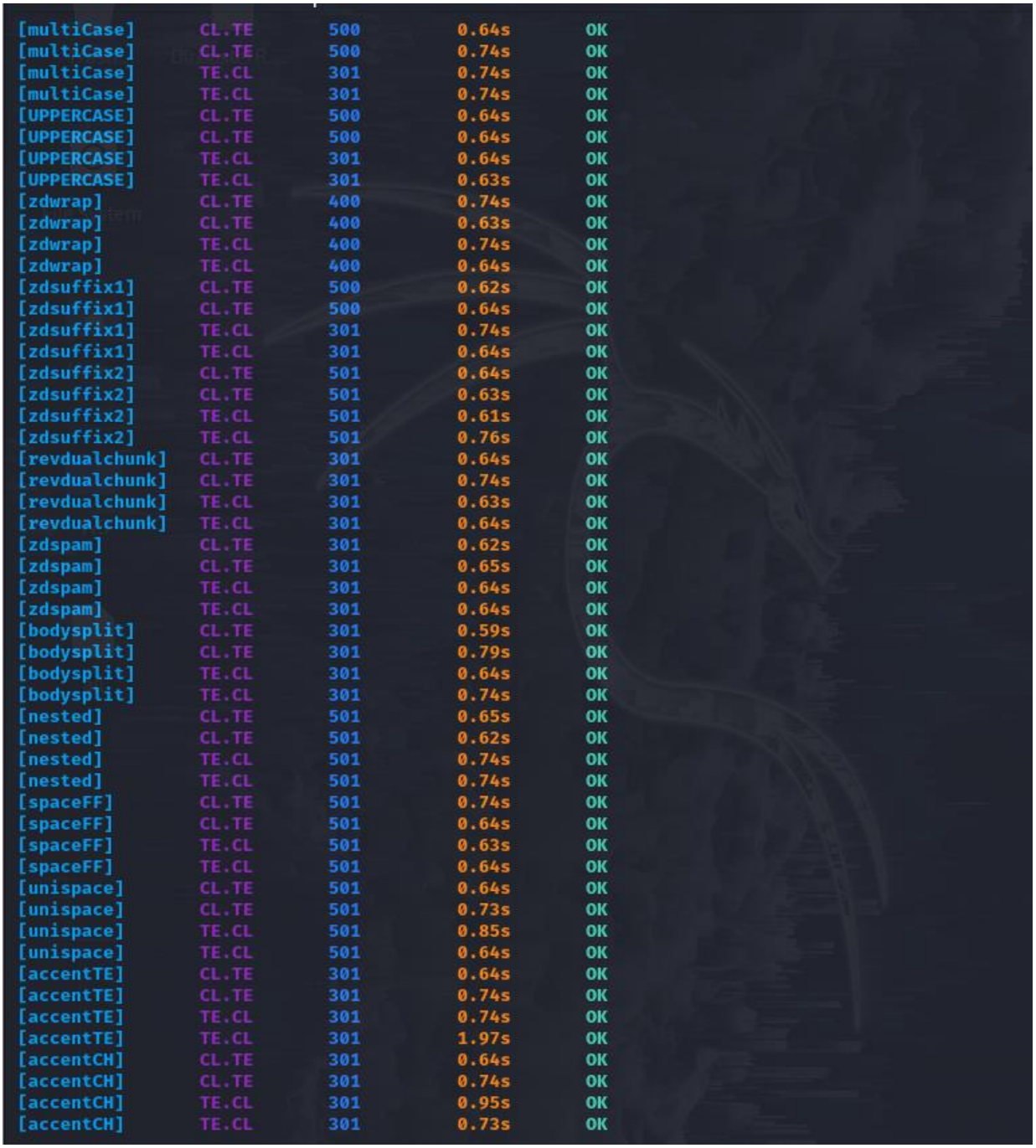
```
┌──(root㉿error404)-[~/XSStrike]
└─# python3 xsstrike.py -u "http://inmobi.com" --crawl

      XSStrike v3.1.5

[~] Crawling the target
[+] Potentially vulnerable objects found at http://inmobi.com
```

```
1   var sdkInstance="appInsightsSDK";window[sdkInstance]="appInsights";var aiName=window[sdkInstance],aisdk=window[ai
Name]||function(e){function n(e){t[e]=function(){var n=arguments;t.queue.push(function(){t[e].apply(t,n)})}}var t={co
nfig:e};t.initialize=!0;var i=document,a=window;setTimeout(function(){var n=i.createElement("script");n.src=e.url||"h
ttps://az416426.vo.msecnd.net/scripts/b/ai.2.min.js",i.getElementsByTagName("script")[0].parentNode.appendChild(n)});
try{t.cookie=i.cookie}catch(e){}t.queue=[],t.version=2;for(var r=["Event","PageView","Exception","Trace","DependencyD
ata","Metric","PageViewPerformance"];r.length;)n("track"+r.pop());n("startTrackPage"),n("stopTrackPage");var s="Track
"+r[0];if(n("start"+s),n("stop"+s),n("setAuthenticatedUserContext"),n("clearAuthenticatedUserContext"),n("flush"),!(!
0===e.disableExceptionTracking||e.extensionConfig&&e.extensionConfig.ApplicationInsightsAnalytics&&!0===e.extensionCo
nfig.ApplicationInsightsAnalytics.disableExceptionTracking)){n("_"+(r="onerror"));var o=a[r];a[r]=function(e,n,i,a,s)
{var c=o&&o(e,n,i,a,s);return!0!==c&&t["_"+r]({message:e,url:n,lineNumber:i,columnNumber:a,error:s}),c},e.autoExcepti
onInstrumented=!0}return t}(
14  document.cookie = "cookie-pref" + "=rejected;" + expiry + ";path=/";
28  var cookies = document.cookie.split(";");
29  for (var i = 0; i < cookies.length; i++) {
30  var cookie = cookies[i];
31  var eqPos = cookie.indexOf("=");
32  var name = eqPos > -1 ? cookie.substr(0, eqPos) : cookie;
33  if(name !== 'cookie-pref') {
34  document.cookie = name + "=;expires=Thu, 01 Jan 1970 00:00:00 GMT";
37  document.cookie = "cookie-pref" + "=rejected;" + expires + ";path=/";
38  const cookieEl = document.getElementsByClassName("accept-cookie")[0];
40  $(".accept-cookie").addClass("hide");
43  const name = "cookie-pref" + "=";
44  const ca = document.cookie.split(';');
50  if (c.indexOf(name) == 0) {
51  return c.substring(name.length, c.length);
61  var cookies = document.cookie.split(";");
62  for (var i = 0; i < cookies.length; i++) {
63  var cookie = cookies[i];
64  var eqPos = cookie.indexOf("=");
65  var name = eqPos > -1 ? cookie.substr(0, eqPos) : cookie;
66  document.cookie = name + "=;expires=Thu, 01 Jan 1970 00:00:00 GMT";
68  document.cookie = "cookie-pref" + "=rejected;" + expires + ";path=/";
69  const cookieEl = document.getElementsByClassName("accept-cookie")[0];
71  $(".accept-cookie").addClass("hide");
76  $(".accept-cookie").removeClass("hide");
80  const cookieEl = document.getElementsByClassName("accept-cookie")[0];
81  $(".accept-cookie").addClass("hide");
```

```
1   {"props":{"pageProps":{"allHomeData":[{"title":"/home","url_title":"home","home_hero_tile_desktop_image":[{"bg_im
age":"https://web.inmobicdn.net/website/website/6.0.1/uploads/misc/Home-Hero-Web-v1.jpg"},{"bg_image":"https://web.in
mobicdn.net/website/website/6.0.1/uploads/misc/Home-Hero-Web-v2_2.jpg"}],"home_hero_mobile_tile":[{"bg_image":"https:
//web.inmobicdn.net/website/website/6.0.1/uploads/misc/Home-Hero-Mobile-v1.webp"},{"bg_image":"https://web.inmobicdn.
net/website/website/6.0.1/uploads/misc/Home-Hero-Mobile-v2.webp"}],"home_promo_banner":[{"heading":"InMobi University
!","content":"New InMobi University Course available now.","caption":"InMobi University Banner 1","cta_link":"https:/
```

F. Http smuggler.

Interpretation of Content-length and/or Transfer-encoding headers may be inconsistent between HTTP proxy server chain and HTTP server implementations that allow an attacker to smuggle HTTP requests. We can test this using python tool called smuggling,



```
[multiCase]      CL.TE      500        0.64s      OK
[multiCase]      CL.TE      500        0.74s      OK
[multiCase]      TE.CL      301        0.74s      OK
[multiCase]      TE.CL      301        0.74s      OK
[UPPERCASE]      CL.TE      500        0.64s      OK
[UPPERCASE]      CL.TE      500        0.64s      OK
[UPPERCASE]      TE.CL      301        0.64s      OK
[UPPERCASE]      TE.CL      301        0.63s      OK
[zdwrap]         CL.TE      400        0.74s      OK
[zdwrap]         CL.TE      400        0.63s      OK
[zdwrap]         TE.CL      400        0.74s      OK
[zdwrap]         TE.CL      400        0.64s      OK
[zdsuffix1]      CL.TE      500        0.62s      OK
[zdsuffix1]      CL.TE      500        0.64s      OK
[zdsuffix1]      TE.CL      301        0.74s      OK
[zdsuffix1]      TE.CL      301        0.64s      OK
[zdsuffix2]      CL.TE      501        0.64s      OK
[zdsuffix2]      CL.TE      501        0.63s      OK
[zdsuffix2]      TE.CL      501        0.61s      OK
[zdsuffix2]      TE.CL      501        0.76s      OK
[revdualchunk]   CL.TE      301        0.64s      OK
[revdualchunk]   CL.TE      301        0.74s      OK
[revdualchunk]   TE.CL      301        0.63s      OK
[revdualchunk]   TE.CL      301        0.64s      OK
[zdspam]         CL.TE      301        0.62s      OK
[zdspam]         CL.TE      301        0.65s      OK
[zdspam]         TE.CL      301        0.64s      OK
[zdspam]         TE.CL      301        0.64s      OK
[bodysplit]      CL.TE      301        0.59s      OK
[bodysplit]      CL.TE      301        0.79s      OK
[bodysplit]      TE.CL      301        0.64s      OK
[bodysplit]      TE.CL      301        0.74s      OK
[nested]         CL.TE      501        0.65s      OK
[nested]         CL.TE      501        0.62s      OK
[nested]         TE.CL      501        0.74s      OK
[nested]         TE.CL      501        0.74s      OK
[spaceFF]        CL.TE      501        0.74s      OK
[spaceFF]        CL.TE      501        0.64s      OK
[spaceFF]        TE.CL      501        0.63s      OK
[spaceFF]        TE.CL      501        0.64s      OK
[unispace]       CL.TE      501        0.64s      OK
[unispace]       CL.TE      501        0.73s      OK
[unispace]       TE.CL      501        0.85s      OK
[unispace]       TE.CL      501        0.64s      OK
[accentTE]       CL.TE      301        0.64s      OK
[accentTE]       CL.TE      301        0.74s      OK
[accentTE]       TE.CL      301        0.74s      OK
[accentTE]       TE.CL      301        1.97s      OK
[accentCH]       CL.TE      301        0.64s      OK
[accentCH]       CL.TE      301        0.74s      OK
[accentCH]       TE.CL      301        0.95s      OK
[accentCH]       TE.CL      301        0.73s      OK
```

G. Oralyzer.

If the attacker can control where the target website is redirected, that is where open redirection vulnerability occurs, because he/she may be able to redirect the victim to a malicious website of his/her own. We can test this vulnerability using a python tool called Oralyzer.

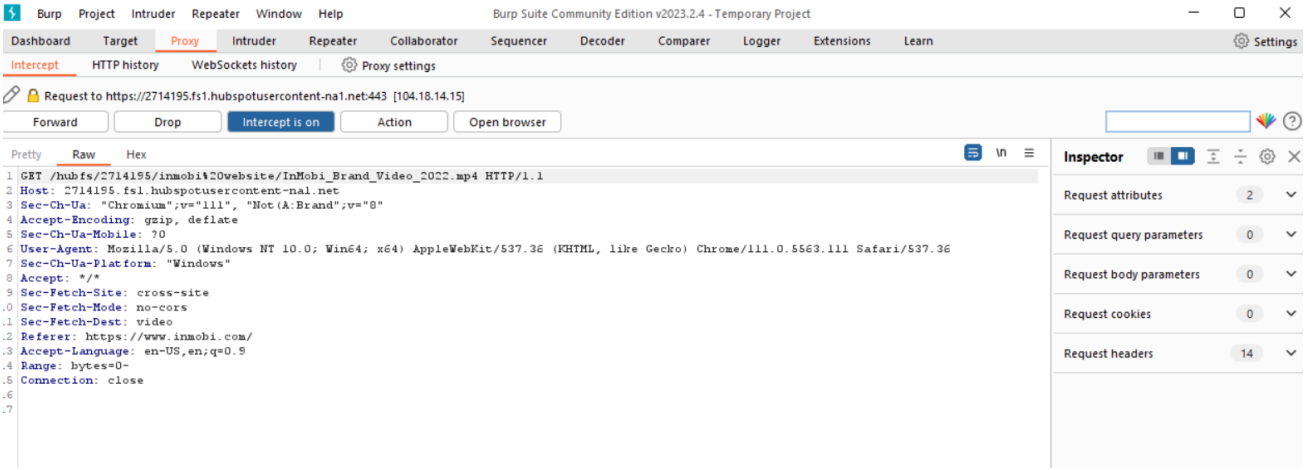
```
┌──(root㉿error404)-[~/oralyzer]
└─# ./oralyzer.py -u https://inmobi.com

      Oralyzer

[!] Appending payloads just after the URL
[!] Infusing payloads
[+] Header Based Redirection : https://inmobi.com/http://www.google.com → https://www.inmobi.com/http://www.google.
com
[+] Header Based Redirection : https://inmobi.com/http%3A%2F%2Fwww.google.com → https://www.inmobi.com/http%3A%2F%2
Fwww.google.com
[+] Header Based Redirection : https://inmobi.com/https%3A%2F%2Fwww.google.com → https://www.inmobi.com/https%3A%2F
%2Fwww.google.com
[+] Header Based Redirection : https://inmobi.com///www.google.com → https://www.inmobi.com///www.google.com
[+] Header Based Redirection : https://inmobi.com/https:www.google.com → https://www.inmobi.com/https:www.google.co
m
[+] Header Based Redirection : https://inmobi.com/google.com → https://www.inmobi.com/google.com
[+] Header Based Redirection : https://inmobi.com//%5C/%5Cgoogle.com → https://www.inmobi.com//%5C/%5Cgoogle.com
[+] Header Based Redirection : https://inmobi.com//%5C/google.com → https://www.inmobi.com//%5C/google.com
[+] Header Based Redirection : https://inmobi.com/////google.com → https://www.inmobi.com/////google.com
[+] Header Based Redirection : https://inmobi.com/HtTP://google.com → https://www.inmobi.com/HtTP://google.com
[+] Header Based Redirection : https://inmobi.com/HTTP://google.com → https://www.inmobi.com/HTTP://google.com
[+] Header Based Redirection : https://inmobi.com/hTTp://google.com → https://www.inmobi.com/hTTp://google.com
[+] Header Based Redirection : https://inmobi.com/HtTPs://google.com → https://www.inmobi.com/HtTPs://google.com
[+] Header Based Redirection : https://inmobi.com/hthttp://tp://google.com → https://www.inmobi.com/hthttp://tp://g
oogle.com
[+] Header Based Redirection : https://inmobi.com/x00http://google.com → https://www.inmobi.com/x00http://google.co
m
[+] Header Based Redirection : https://inmobi.com/%5Cx20http://google.com → https://www.inmobi.com/%5Cx20http://goo
gle.com
[+] Header Based Redirection : https://inmobi.com/216.58.214.206 → https://www.inmobi.com/216.58.214.206
[+] Header Based Redirection : https://inmobi.com/172.217.167.46 → https://www.inmobi.com/172.217.167.46
[+] Header Based Redirection : https://inmobi.com//216.58.214.206 → https://www.inmobi.com//216.58.214.206
[+] Header Based Redirection : https://inmobi.com/////216.58.214.206 → https://www.inmobi.com/////216.58.214.206
[+] Header Based Redirection : https://inmobi.com//%5C216.58.214.206 → https://www.inmobi.com//%5C216.58.214.206
[+] Header Based Redirection : https://inmobi.com///216.58.214.206 → https://www.inmobi.com///216.58.214.206
[+] Header Based Redirection : https://inmobi.com////216.58.214.206 → https://www.inmobi.com////216.58.214.206
[+] Header Based Redirection : https://inmobi.com///google%E3%80%82com → https://www.inmobi.com///google%E3%80%82co
m
[+] Header Based Redirection : https://inmobi.com///google%E3%80%82com → https://www.inmobi.com///google%E3%80%82co
m
[+] Header Based Redirection : https://inmobi.com/http%5Cx3A%5Cx2F%5Cx2Fgoogle.com → https://www.inmobi.com/http%5C
x3A%5Cx2F%5Cx2Fgoogle.com
[+] Header Based Redirection : https://inmobi.com/////google.com/.. → https://www.inmobi.com/////google.com/..
[+] Header Based Redirection : https://inmobi.com////google.com/.. → https://www.inmobi.com////google.com/..
[+] Header Based Redirection : https://inmobi.com///google.com/.. → https://www.inmobi.com///google.com/..
[+] Header Based Redirection : https://inmobi.com//google.com/.. → https://www.inmobi.com//google.com/..
[+] Header Based Redirection : https://inmobi.com///google.com/.. → https://www.inmobi.com///google.com/..
[+] Header Based Redirection : https://inmobi.com/////google.com/..%2F → https://www.inmobi.com/////google.com/..%2
F
[+] Header Based Redirection : https://inmobi.com////google.com/..%2F → https://www.inmobi.com////google.com/..%2F
[+] Header Based Redirection : https://inmobi.com///google.com/..%2F → https://www.inmobi.com///google.com/..%2F
[+] Header Based Redirection : https://inmobi.com/////google.com/%2F.. → https://www.inmobi.com/////google.com/%2F.
.
```

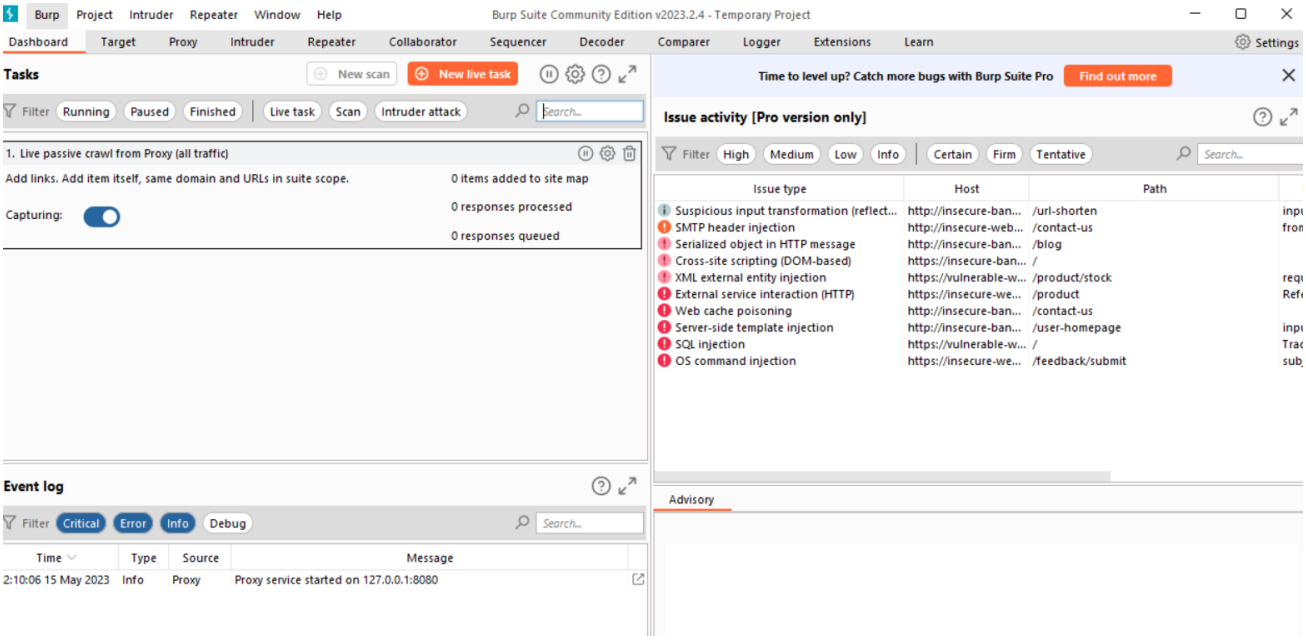
```
[+] Header Based Redirection : https://inmobi.com/http://0×d83ad6ce/inmobi.com → https://www.inmobi.com/http://0×d8
3ad6ce/inmobi.com
[+] Header Based Redirection : https://inmobi.com/http:0×d83ad6ce.inmobi.com → https://www.inmobi.com/http:0×d83ad6
ce.inmobi.com
[+] Header Based Redirection : https://inmobi.com/http:0×d83ad6ce/inmobi.com → https://www.inmobi.com/http:0×d83ad6
ce/inmobi.com
[+] Header Based Redirection : https://inmobi.com/http://%5B::216.58.214.206%5D.inmobi.com → https://www.inmobi.com
/http://%5B::216.58.214.206%5D.inmobi.com
[+] Header Based Redirection : https://inmobi.com/http://%5B::216.58.214.206%5D/inmobi.com → https://www.inmobi.com
/http://%5B::216.58.214.206%5D/inmobi.com
[+] Header Based Redirection : https://inmobi.com/http:%5B::216.58.214.206%5D.inmobi.com → https://www.inmobi.com/h
ttp:%5B::216.58.214.206%5D.inmobi.com
[+] Header Based Redirection : https://inmobi.com/http:%5B::216.58.214.206%5D/inmobi.com → https://www.inmobi.com/h
ttp:%5B::216.58.214.206%5D/inmobi.com
```

H. Burp suite.

Burp Suite is a popular integrated platform for performing web application security testing. Burp Suite provides a range of tools and features to help security professionals and penetration testers identify and exploit security vulnerabilities in web applications. Some of the key features of Burp Suite include:

- A web proxy that allows users to intercept and modify web traffic in real-time.
- A web vulnerability scanner that automatically identifies security vulnerabilities in web applications.
- An application-aware spider that can discover hidden content and functionality in web applications.
- A repeater tool that allows users to modify and re-send requests to web applications.
- A sequencer tool that can test the randomness and strength of tokens and session identifiers.
- A collaborator tool that can be used to identify blind vulnerabilities in web applications.

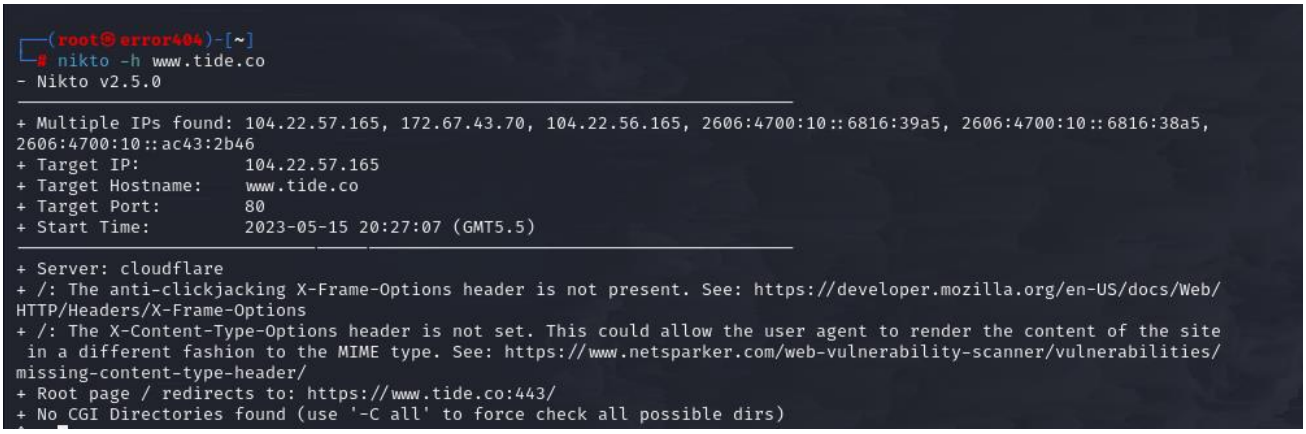




I. Nikto scan.

Nikto is an open-source web server scanner that is used to identify potential security vulnerabilities in web servers and applications. It is a command-line tool that can be used to scan web servers for common vulnerabilities and misconfigurations, including outdated software, insecure file permissions, and known exploits.

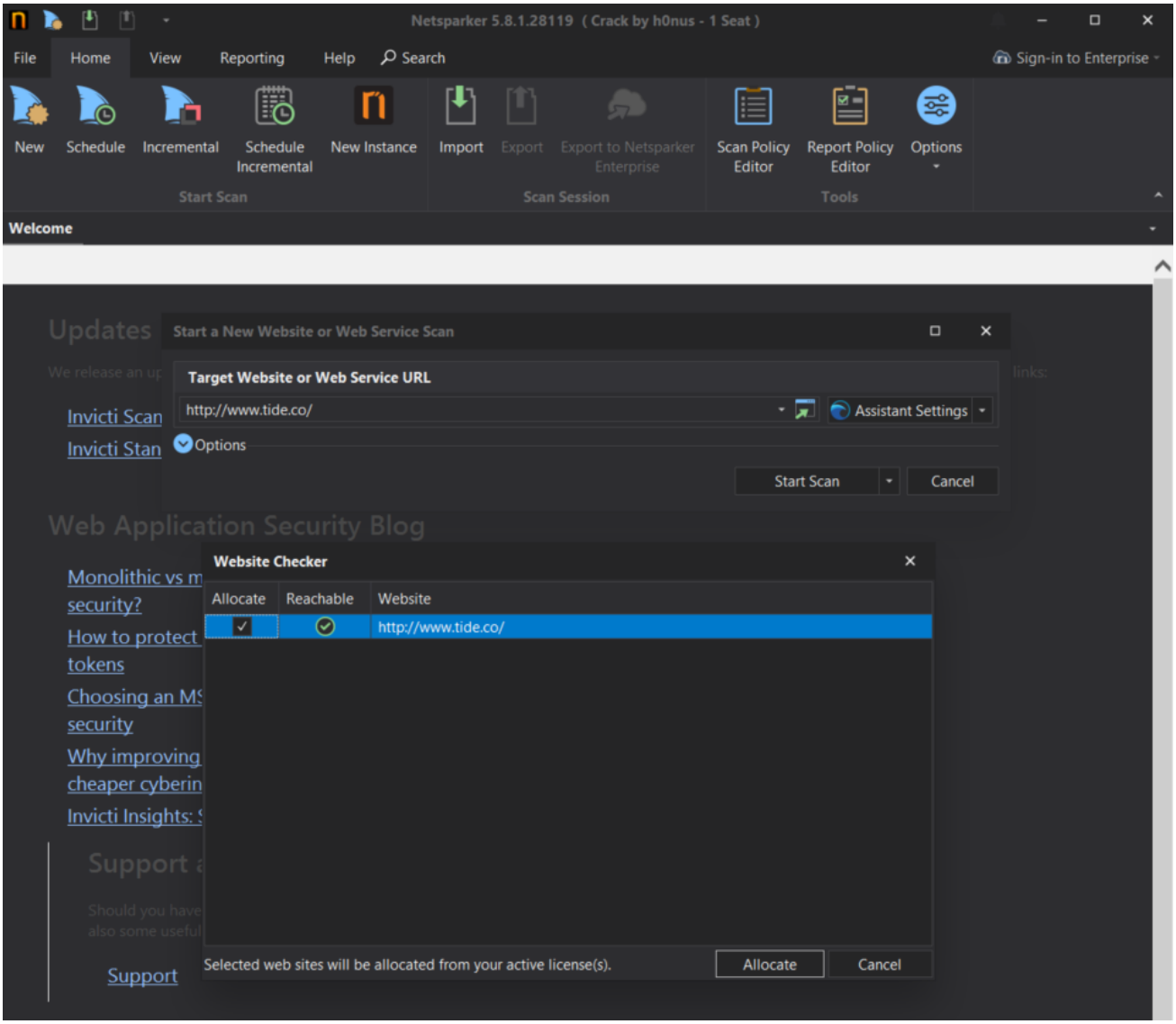
- Support for multiple protocols, including HTTP and HTTPS.
- Integration with other security tools and services, including the National Vulnerability Database (NVD) and the Common Vulnerability Scoring System (CVSS).
- Customizable scanning options that allow users to tailor their scans to specific systems or software configurations.
- Automated scanning capabilities that reduce the need for manual intervention and make it easy to scan large and complex web servers.

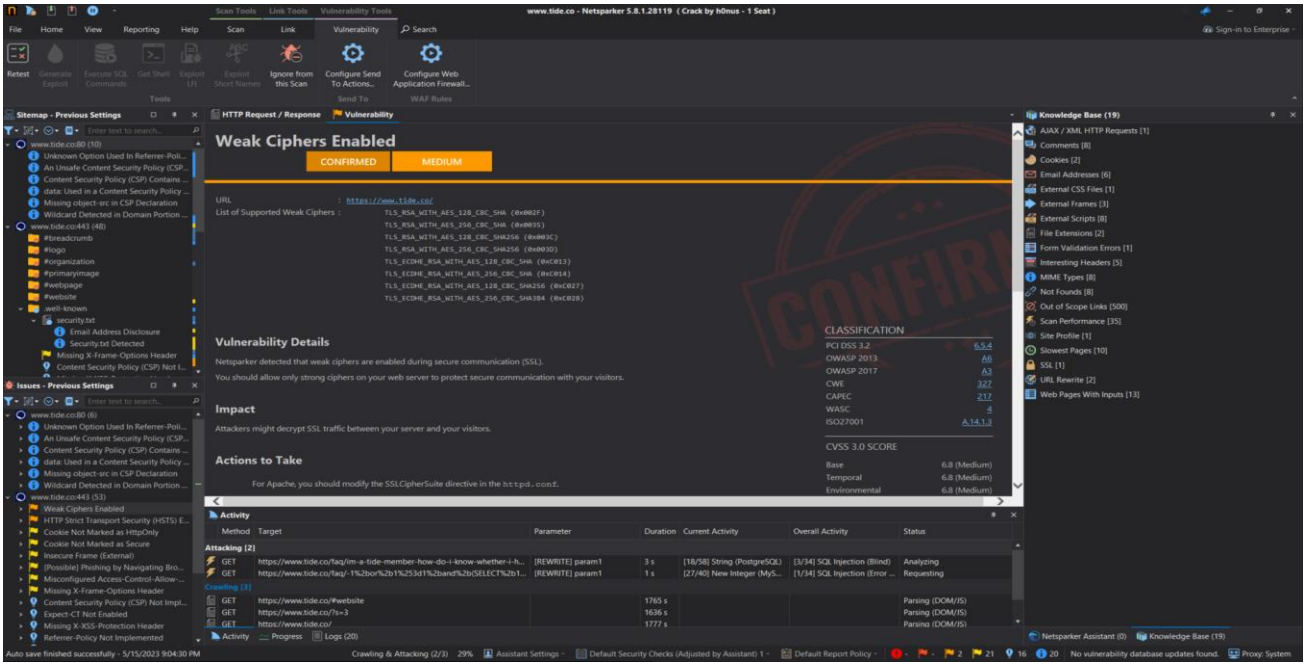


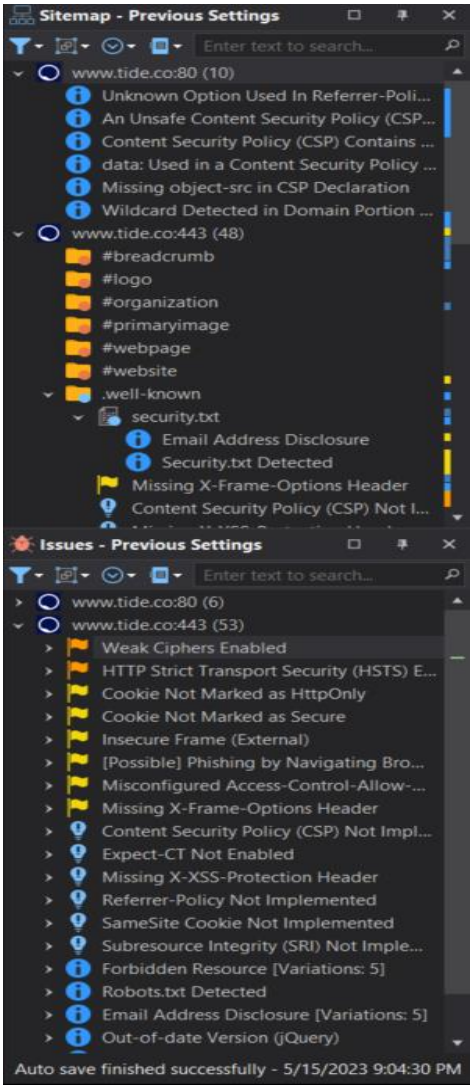
J. Netsparker.

Netsparker is a powerful web application security scanner that is used to identify and report security vulnerabilities in web applications. It is designed to be highly automated and easy to use, making it an ideal tool for security professionals, web developers, and system administrators.

- Automated scanning capabilities that reduce the need for manual intervention and make it easy to scan large and complex web applications.
- Customizable scanning options that allow users to tailor their scans to specific systems or software configurations.
- Integration with other security tools and services, including the OWASP Top 10 list of web application vulnerabilities.


Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)
File
Home
View
Reporting
Help
Search
Sign-in to Enterprise
New
Schedule
Incremental
Schedule Incremental
New Instance
Import
Export
Export to Netsparker Enterprise
Scan Policy Editor
Report Policy Editor
Options
Start Scan
Scan Session
Tools
Welcome
Updates
Start a New Website or Web Service Scan
Target Website or Web Service URL
http://www.tide.co/
Assistant Settings
Options
Start Scan
Cancel
Invicti Scan
Invicti Stan
Web Application Security Blog
Monolithic vs m
security?
How to protect
tokens
Choosing an MS
security
Why improving
cheaper cyberin
Invicti Insights:
Support
Website Checker
Allocate
Reachable
Website
http://www.tide.co/
Selected web sites will be allocated from your active license(s).
Allocate
Cancel
Support

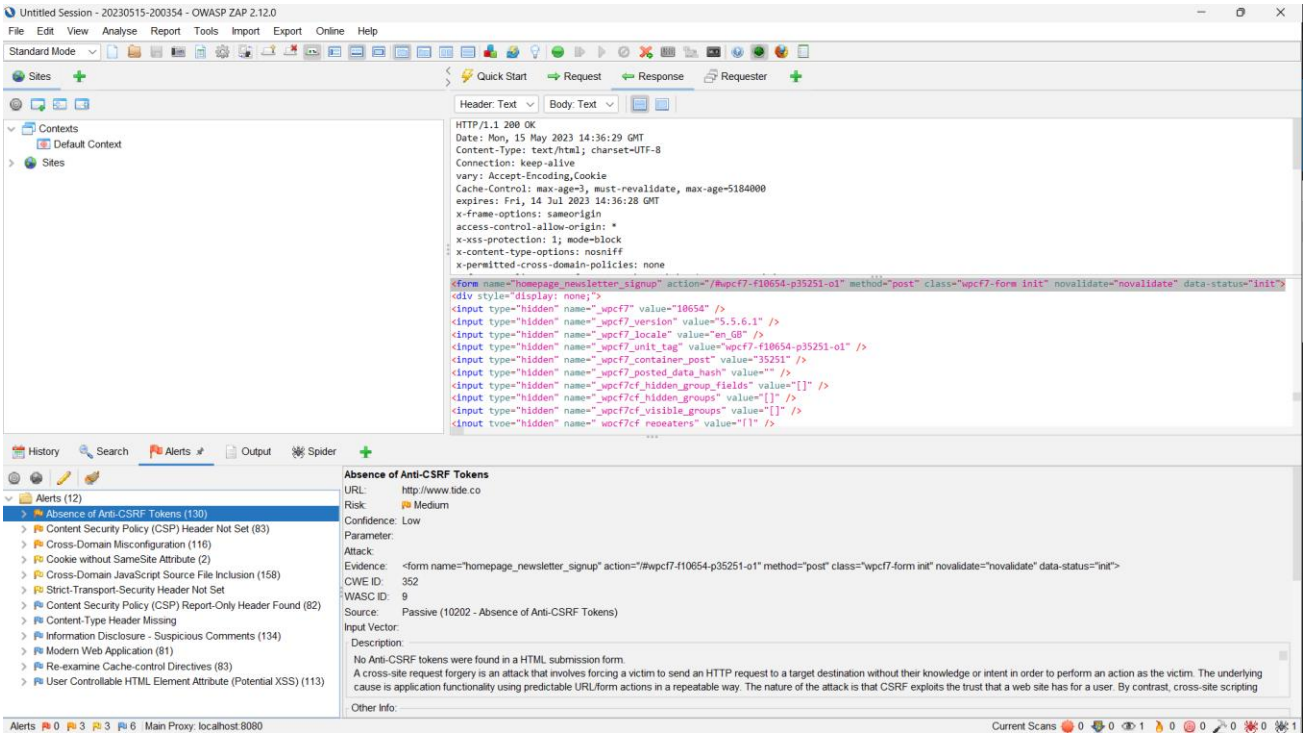

Weak Ciphers Enabled
CONFIRMED
MEDIUM
Vulnerability Details
Impact
Actions to Take
CLASSIFICATION
CVSS 3.0 SCORE


Sitemap - Previous Settings
Enter text to search...
www.tide.co:80 (10)
Unknown Option Used In Referrer-Poli...
An Unsafe Content Security Policy (CSP...
Content Security Policy (CSP) Contains ...
data: Used in a Content Security Policy ...
Missing object-src in CSP Declaration
Wildcard Detected in Domain Portion ...
www.tide.co:443 (48)
#breadcrumb
#logo
#organization
#primaryimage
#webpage
#website
.well-known
security.txt
Email Address Disclosure
Security.txt Detected
Missing X-Frame-Options Header
Content Security Policy (CSP) Not I...
Issues - Previous Settings
Enter text to search...
www.tide.co:80 (6)
www.tide.co:443 (53)
Weak Ciphers Enabled
HTTP Strict Transport Security (HSTS) E...
Cookie Not Marked as HttpOnly
Cookie Not Marked as Secure
Insecure Frame (External)
[Possible] Phishing by Navigating Bro...
Misconfigured Access-Control-Allow-...
Missing X-Frame-Options Header
Content Security Policy (CSP) Not Impl...
Expect-CT Not Enabled
Missing X-XSS-Protection Header
Referrer-Policy Not Implemented
SameSite Cookie Not Implemented
Subresource Integrity (SRI) Not Imple...
Forbidden Resource [Variations: 5]
Robots.txt Detected
Email Address Disclosure [Variations: 5]
Out-of-date Version (jQuery)
Auto save finished successfully - 5/15/2023 9:04:30 PM

K. OWASP ZAP

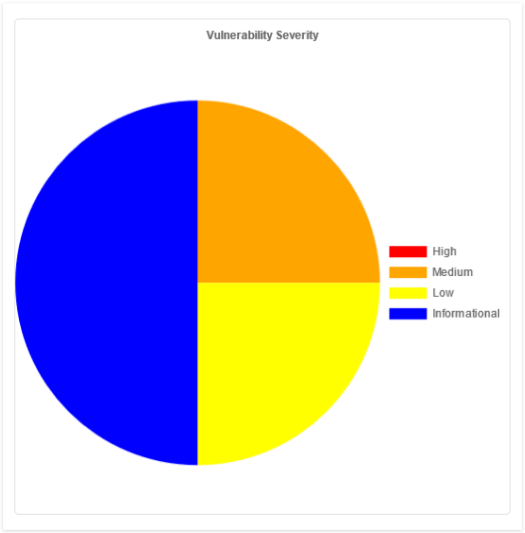
OWASP ZAP (Zed Attack Proxy) is a free and open-source web application security scanner that is designed to help security professionals and system administrators identify and address security vulnerabilities in web applications. It is a powerful tool that can be used to perform comprehensive security assessments of web applications, including identifying vulnerabilities such as cross-site scripting (XSS), SQL injection, and other types of security flaws.

- Support for a wide range of web application technologies, including HTML5, Ajax, and REST.
- Built-in proxy functionality that allows for easy interception and modification of web traffic.
- An intuitive user interface that makes it easy to configure and manage scans.
- Integration with other security tools and services, including the National Vulnerability Database (NVD) and the Common Vulnerability Scoring System (CVSS).
- Customizable reporting capabilities, including the ability to generate detailed reports in various formats, such as HTML, CSV, and JSON.



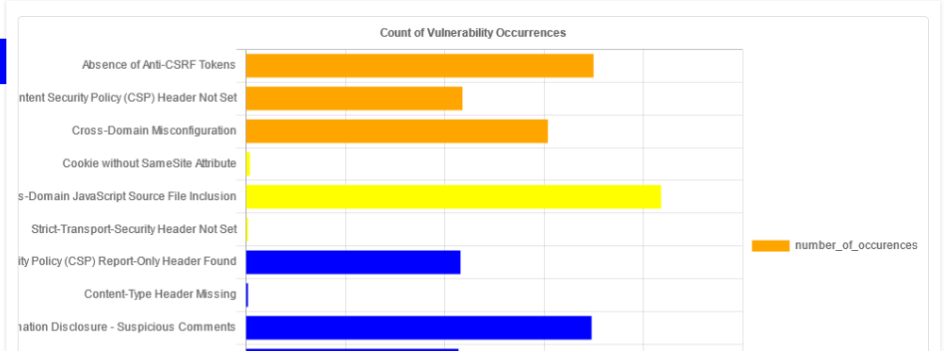
Most Severe Alert

Medium



Most Common Bug

User Controllable HTML Element Attribute (Potential XSS) (218)



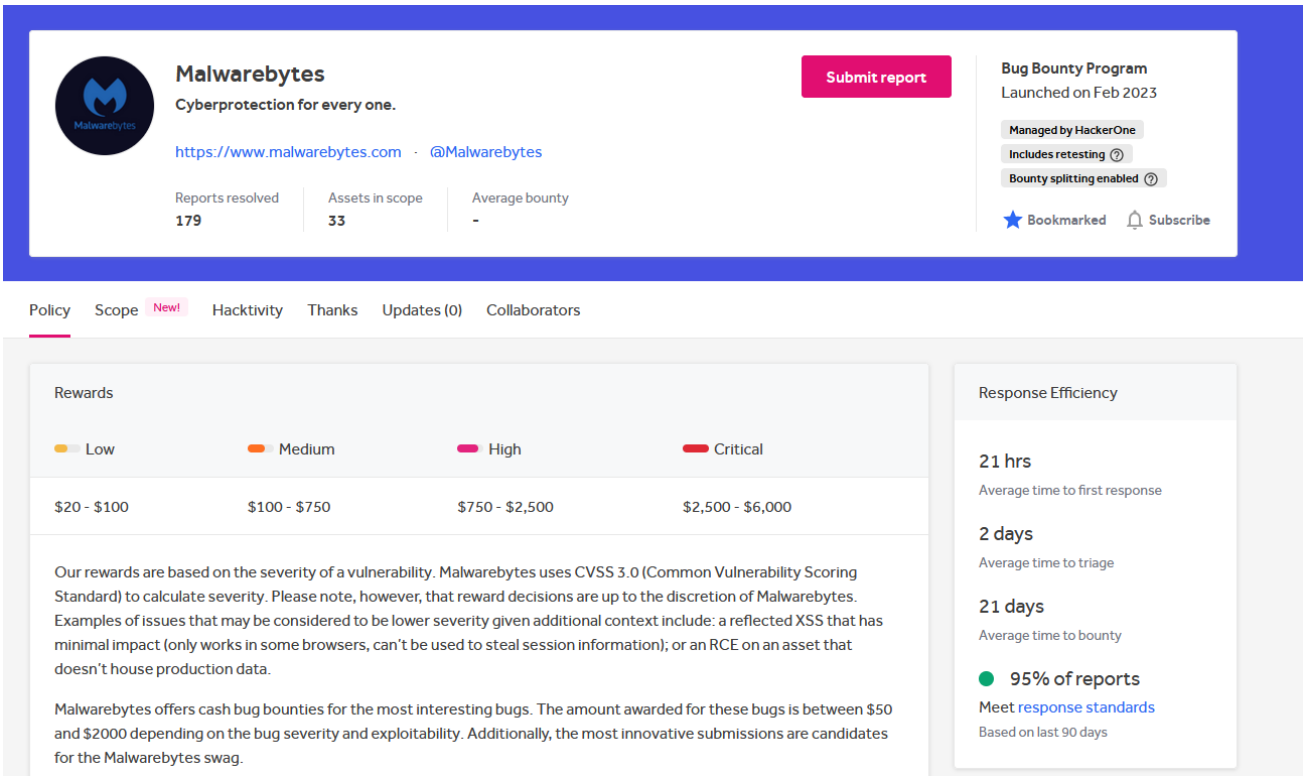
5. Identified vulnerabilities.

In this bug bounty program, I have tested many web applications against various of vulnerabilities.to that I have used lots of tools under each phase of vulnerability assessment. For the ease of identification of vulnerabilities, I have used fully automated tools like OWASP ZAP, Net sparker. In case of further confirmation needed about existence of that identified vulnerability I have used Linux build in tools and frameworks for the double confirmations.

In this journal book I have mentioned all the vulnerabilities I found in each web application and for the reporting purposes of the existence of the vulnerability I have only used most crucial vulnerabilities that I have found in each domain.

A. Domain: Malwarebytes.com

Proof of bug bounty program:



I have identified many vulnerabilities in under domain **www.malwarebytes.com/.** I figured out it has vulnerabilities that listed by OWASP Top 10. And the overall website risk level is **medium**. I used net sparker to identify vulnerabilities in the following domain. And used some of the previously mentioned tools in section. I identified the following OWASP Top vulnerabilities in following domain.

1. **Security misconfiguration**
2. **Sensitive data exposure**
3. **Using components with known vulnerabilities.**

netsparker

5/15/2023 7:12:47 PM (UTC+05:30)

OWASP Top Ten 2013 Report

https://www.malwarebytes.com/

Scan Time	: 5/15/2023 3:44:33 PM (UTC+05:30)
Scan Duration	: 00:03:06:45
Total Requests	: 39,690
Average Speed	: 3.5r/s

Risk Level:
MEDIUM

Explanation

This report is generated based on OWASP Top Ten 2013 classification.
There are 11 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.

15 IDENTIFIED	3 CONFIRMED	0 CRITICAL
0 HIGH	4 MEDIUM	5 LOW
	1 BEST PRACTICE	5 INFORMATION

Identified Vulnerabilities

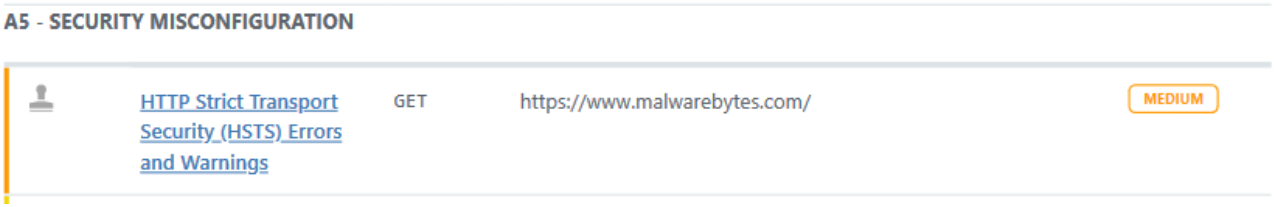
Critical	0
High	0
Medium	4
Low	5
Best Practice	1
Information	5
TOTAL	**15**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	2
Best Practice	0
Information	0
TOTAL	**3**

1. Security misconfiguration. -Reported

i. HSTS security errors and warnings



Under this OWASP category I have identified many vulnerabilities that can occur due to bad implementation of security under domain of **https://www.malwarebytes.com/.**

- **Impacts of HSTS errors and warnings.**

1. HSTS faults or warnings indicate that an HTTPS connection is not being properly enforced by the web application. Insecure connections. As a result, users may access the application over insecure HTTP connections, increasing the risk of attackers listening in, meddling with, or intercepting their communications.

2. HSTS failures or warnings can make the web application susceptible to MITM attacks. Attackers can force the browser to communicate over insecure HTTP rather than HTTPS by intercepting the initial HTTP request and altering the response to delete or change the HSTS headers.

3. The website may be unable to create secure connections with compliant browsers due to HSTS issues. Users may get browser errors or warnings as a result, indicating that the connection might not be secure. Users can feel deterred from using the application or think it's less reliable.

4. HSTS faults or warnings may make SSL/TLS's (Secure Sockets Layer/Transport Layer Security) defense against SSL/TLS-stripping attacks less effective. These attacks seek to switch the connection from HTTPS to HTTP, which would enable attackers to eavesdrop on or alter user-app communication.

Because of this vulnerability marked as MEDIUM. It’s a must to address it with a good solution.

- **Steps to reproduce this vulnerability.**

1. Clear the cache in your web browser first before attempting to access your online application. By doing this, you may be sure that all previous HSTS settings have been eliminated and can witness the behavior in its purest form.

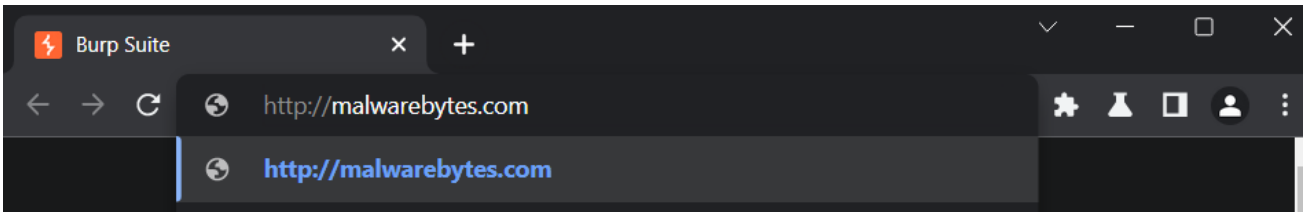
2. Access your web application through HTTP: Enter the URL using the HTTP protocol (for example, "http://www.malwarebytes.com") rather than HTTPS to access your web application. This should open a connection that is not secure.

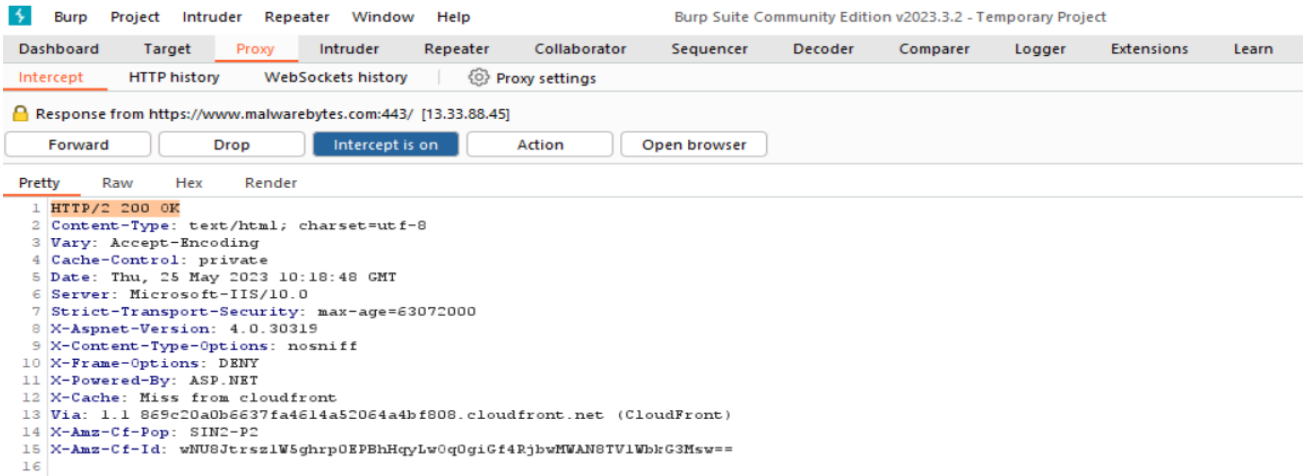
3. After utilizing HTTP to visit the web application, watch your web browser's actions. The browser ought to immediately switch the connection to HTTPS, enforcing a secure connection, if HSTS is properly implemented. However, the browser may show an error message, a warning, or fail to create a secure connection if there are HSTS issues or warnings.

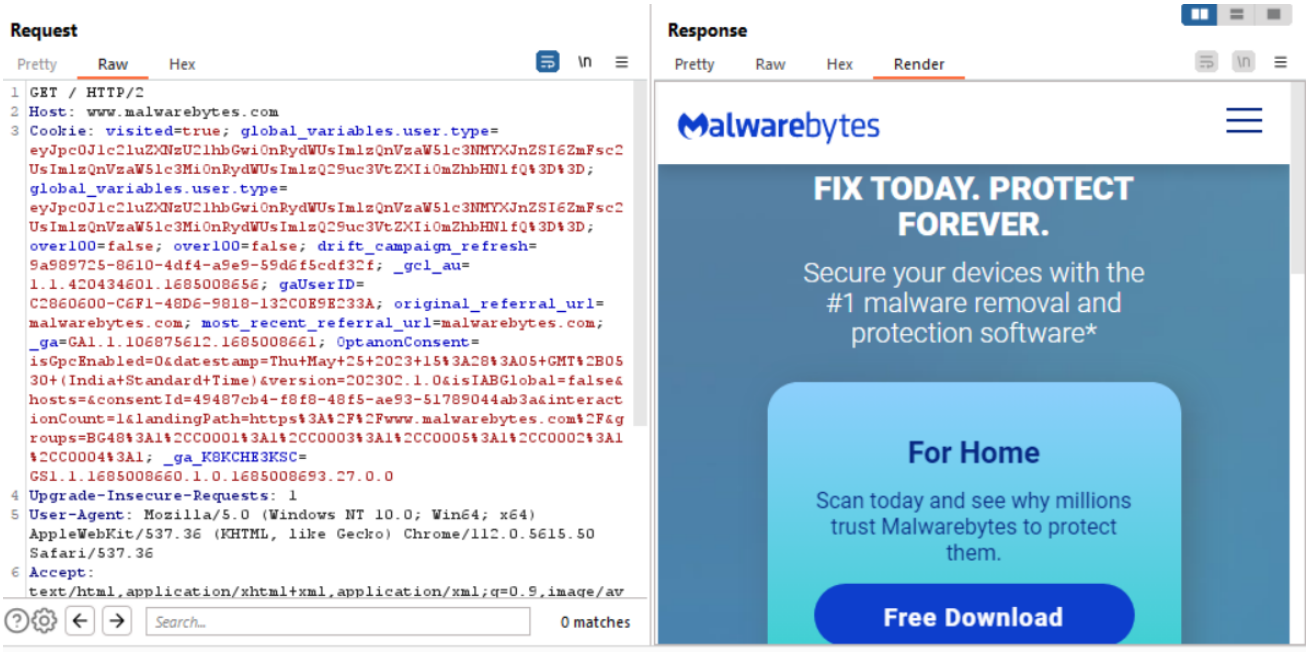
4. Examine the developer tools in the browser by opening them. To access the developer tools, right-click on the webpage and select "Inspect" or "Inspect Element" in most browsers. Look for any HSTS-related error messages, warnings, or console logs.

- **Proof of Concept.**

1. **Access web application through HTTP.**







2. HSTS scan report by SSLLabs

SSL Report: malwarebytes.com

Assessed on: Thu, 25 May 2023 10:17:27 UTC | Hide | Clear cache

Scan Another >>

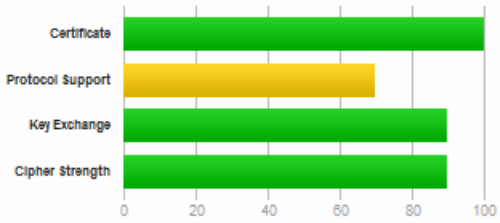
	Server	Test time	Grade
1	99.84.238.97 server-99-84-238-97.sfo5.r.cloudfront.net Ready	Thu, 25 May 2023 09:59:47 UTC Duration: 262.355 sec	B
2	99.84.238.177 server-99-84-238-177.sfo5.r.cloudfront.net Ready	Thu, 25 May 2023 10:04:09 UTC Duration: 267.374 sec	B
3	99.84.238.194 server-99-84-238-194.sfo5.r.cloudfront.net Ready	Thu, 25 May 2023 10:08:36 UTC Duration: 266.71 sec	B
4	99.84.238.172 server-99-84-238-172.sfo5.r.cloudfront.net Ready	Thu, 25 May 2023 10:13:02 UTC Duration: 264.250 sec	B

SSL Report v2.1.10

Summary

Overall Rating

B



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server supports TLS 1.0 and TLS 1.1. Grade capped to B. MORE INFO »

This site works only in browsers with SNI support.

This server supports TLS 1.3.

HTTP Requests

1 https://malwarebytes.com/ (HTTP/1.1 301 Moved Permanently)

	Content-Length	0
	Connection	close
	Server	CloudFront
	Date	Wed, 24 May 2023 18:15:19 GMT
	Location	https://www.malwarebytes.com/
1	Cache-Control	max-age=86400
	X-Cache	Hit from cloudfront
	Via	1.1 9e2f847ffc5e44974bd7f01a7603f72c.cloudfront.net (CloudFront)
	X-Amz-Cf-Pop	SFO5-C3
	X-Amz-Cf-Id	RcCTtlLM1TAPh8kJQwXlH8WTOJ9N3YW-couqBQdXApDJ6ptgTo_s5g==
	Age	56674

- **Proof of existence of vulnerability.**

Request

```
GET / HTTP/1.1
Host: www.malwarebytes.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 394.9114 Total Bytes Received : 99524 Body Length : 98932 Is Compressed : No

```
HTTP/1.1 200 OK
X-Cache: Miss from cloudfront
Cache-Control: private
Strict-Transport-Security: max-age=63072000
Transfer-Encoding: chunked
X-Powered-By: ASP.NET
Server: Microsoft-IIS/10.0
X-Amz-Cf-Id: atIflY-iXNub1rOSkvyNvvPxhzhZrm-We6nwj5RTD_8IBClGuho51A==
X-Content-Type-Options: nosniff
X-AspNet-Version: 4.0.30319
Connection: keep-alive
X-Frame-Options: DENY
Vary: Accept-Encoding
X-Amz-Cf-Pop: SIN2-C1
Via: 1.1 c8c43b7bd0e92cbb9fbe171dc985f060.cloudfront.net (CloudFront)
Content-Type: text/html; charset=utf-8
Date: Mon, 15 May 2023 10:15:59 GMT
Content-Encoding:
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">

<title>Cyber Security Software & Anti-Malware | Malwarebytes</title>
<meta name="description" content="Protect your home and business PCs, Macs, iOS and Android devices fro
m the latest cyber threats and malware, including ransomware."/>
<link rel="canonical" href="https://www.malwarebytes.com/"/></link>

<link rel="alternate" hreflang="x-default" href="https://www.malwarebytes.com"/>
<link rel="alternate" href="https://www.malwarebytes.com" hreflang="en-us"/>
<link rel="alternate" href="https://es.malwarebytes.com" hreflang="es-es"/>
<link rel="alternate" href="https://de.malwarebytes.com" hreflang="de-de"/>
<link rel="alternate" href="https://it.malwarebytes.com" hreflang="it-it"/>
<link rel="alternate" href="https://fr.malwarebytes.com" hreflang="fr-fr"/>
<link rel="alternate" href="https://nl.malwarebytes.com" hreflang="nl-nl"/>
<link rel="alternate" href="https://br.malwarebytes.com" hreflang="pt-br"/>
<link rel="alternate" href="https://pt.malwarebytes.com" hreflang="pt-pt"/>
<link rel="alternate" href="https://pl.malwarebytes.com" hreflang="pl-pl"/>
<link rel="alternate" href="https://ru.malwarebytes.com" hreflang="ru-ru"/>
<link rel="alternate" href="https://www.malwarebytes.com/jp" hreflang="ja-jp"/>
<link rel="alternate" href="https://www.malwarebytes.com/se" hreflang="sv-se"/>
<script type=
…
```

I have attached the request and response. You can see that it responded to me with a HTTP.it seems preload directive is missing or not enabled.

- **Solutions to the above-mentioned vulnerability (Remedy).**

The first step is to identify the error codes and resolve those problems. The domain must be added to the HSTS preload list after the issues with the HSTS header have been resolved. Browsers automatically connect to the website via HTTPS when the domain is added to the HSTS preload list, preventing users from making HTTPS requests to the server.

The web application needs to be set in accordance with the following requirements before being added to the preload list of the browser.

1. Present a legitimate certificate.
2. If you are listening on port 80, switch all HTTP domains on the same host to HTTPS. Serve every subdomain using HTTPS.
   - If a DNS record for the www subdomain exists, you must support HTTPS for that subdomain.
3. Provide a HSTS header for HTTPS queries on the base domain:
   - The maximum age requirement is 31536000 seconds (one year).
   - It is necessary to specify the included Subdomains directive.
   - Preload directive needs to be mentioned.
   - The HSTS header must be included on the additional redirection that is being served from your HTTPS site, not on the destination page.

2. Sensitive data exposure. – reported

i. Weak ciphers enabled.

Under this OWASP category I have identified many vulnerabilities that can occur due to bad implementation of security implementations under domain of **https://www.malwarebytes.com/**

- **Impacts of HSTS errors and warnings.**

1. So, the effect of allowing weak ciphers during SSL communication is attackers might decrypt the SSL traffic between client and server if they used good mechanism.

2. The online application is more vulnerable to numerous cryptographic attacks with weak ciphers, such as brute force, cipher-text-only, and chosen-plaintext. Attackers can use these flaws to decrypt private information sent between the client and the server.

3. Security requirements and laws may not be followed if weak ciphers are enabled. Strong cryptographic protocols and ciphers are necessary to secure sensitive data according to many security standards, including the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS). If these criteria are not met, there may be legal and regulatory repercussions.

4. In comparison to stronger ciphers, some weaker ciphers could have a higher computational cost. The server's processing time and resource use may rise if such ciphers are enabled, which can affect the web application's performance and response times.

Because of this vulnerability marked as MEDIUM. It’s a must to address it with a good solution.

- **Steps to reproduce this vulnerability.**

1. Must first find the list of ciphers that your web application supports. The SSL/TLS settings on the server are normally setup with this data.

2. There are several internet resources that can scan the SSL/TLS settings of your web application and reveal the ciphers that are enabled. The SSL Server Test from SSL Labs is one such tool **(https://www.ssllabs.com/ssltest/)**. The tool will conduct a thorough examination of your SSL/TLS configuration, including the supported ciphers, after you enter the URL of your web application**. (http://www.malwarebytes.com)**

3. Review the scan results when the program has finished its study to find any ciphers that are enabled in your web application that are weak or out-of-date. The program will often give the SSL/TLS configuration a grade or score and point out any flaws or vulnerabilities it discovers.

4. Test cipher negotiation: Run a cipher negotiation test against your web application using a network testing tool like OpenSSL or Nmap. In this test, you'll establish a connection to your web application and watch as the negotiated cipher suite is established during the handshake.

5. Examine the cipher suite that was agreed upon: See if any weak ciphers were chosen for the connection by looking at the results of the encryption negotiation test. Search for any ciphers that are regarded as being brittle, exposed, or dated.

6. Test on various clients: To watch the SSL/TLS handshake, access your web application using various web browsers and client devices. Verify whether weak ciphers are consistently chosen and whether the negotiated cipher suite varies between clients.

7. Validate with vulnerability scanning tools: Conduct a thorough security analysis of your web application using vulnerability scanning tools like Nessus or OpenVAS. These programs may check for SSL/TLS vulnerabilities and offer thorough findings on ciphers with weak security or possible attacks.

- **Proof of concept.**

1. SSL/TLS scan report by SSLLabs

Certificate #1: RSA 2048 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	malwarebytes.com Fingerprint SHA256: 75cdc223af69e2c49a9a8fd11e6040960513c71e77883413f21285d7a43a8b1b Pin SHA256: FYc3OTCEQQcxLPEyAcMNES7dNwMHZLrDfCXsfkoWbQ=
Common names	malwarebytes.com
Alternative names	malwarebytes.com *.beta.malwarebytes.com *.cloud.malwarebytes.com *.mbamupdates.com *.api.cloud.malwarebytes.com *.malwarebytes.com *.api-stage.cloud.malwarebytes.com *.sre.malwarebytes.com *.malwarebytes.org *.data.service.malwarebytes.org *.mwbsys.com *.eng-prod.mb-internal.com *.mb-cosmos.com
Serial Number	0a018051b91c334887cb4179d526bef1
Valid from	Thu, 23 Feb 2023 00:00:00 UTC
Valid until	Fri, 03 Nov 2023 23:59:59 UTC (expires in 5 months and 9 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	Amazon RSA 2048 M01 AIA: http://crt.r2m01.amazontrust.com/r2m01.cer
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl.r2m01.amazontrust.com/r2m01.crl OCSP: http://ocsp.r2m01.amazontrust.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

Configuration

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Used protocols.

Protocols	
TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Supported cipher suites in each protocol.

1. TLS 1.3

# TLS 1.3 (suites in server-preferred order)	
TLS_AES_128_GCM_SHA256 (0x1301) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_AES_256_GCM_SHA384 (0x1302) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) ECDH x25519 (eq. 3072 bits RSA) FS	256

2. TLS 1.2

# TLS 1.2 (suites in server-preferred order)	
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128

3. TLS 1.1

# TLS 1.1 (suites in server-preferred order)	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128

4. TLS 1.0

# TLS 1.0 (suites in server-preferred order)	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH x25519 (eq. 3072 bits RSA) FS WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128

SSL/TLS handshake simulation in different clients.

1. Android 4.0.4

Protocols

TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) WEAK	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) WEAK	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) WEAK	256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) WEAK	256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f) WEAK	256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) WEAK	112
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008) WEAK	112
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16) WEAK	112
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d) WEAK	112
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003) WEAK	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) WEAK	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) WEAK	128
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) WEAK	128
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e) WEAK	128
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011) INSECURE	128
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007) INSECURE	128
TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c) INSECURE	128
TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0xff)	-

(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	Yes INSECURE
Session tickets	Yes
OCSP stapling	No
Signature algorithms	-
Named Groups	sect163k1, sect163r1, sect163r2, sect193r1, sect193r2, sect233k1, sect233r1, sect239k1, sect283k1, sect283r1, sect409k1, sect409r1, sect571k1, sect571r1, secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, secp224k1, secp224r1, secp256k1, secp256r1, secp384r1, secp521r1
Next Protocol Negotiation	Yes
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

2. Android 4.3

Protocols

TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) WEAK	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) WEAK	256
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA (0xc022) WEAK	256
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA (0xc021) WEAK	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) WEAK	256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x38) WEAK	256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f) WEAK	256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005) WEAK	256
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012) WEAK	112
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008) WEAK	112
TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA (0xc01c) WEAK	112
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA (0xc01b) WEAK	112
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x16) WEAK	112
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d) WEAK	112
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003) WEAK	112
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) WEAK	128
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA (0xc01f) WEAK	128
TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA (0xc01e) WEAK	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) WEAK	128
TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x32) WEAK	128
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e) WEAK	128
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004) WEAK	128
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011) INSECURE	128
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007) INSECURE	128
TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c) INSECURE	128
TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0xff)	-

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	No
Signature algorithms	-
Named Groups	sect571r1, sect571k1, secp521r1, sect409k1, sect409r1, secp384r1, sect283k1, sect283r1, secp256k1, secp256r1, sect239k1, sect233k1, sect233r1, secp224k1, secp224r1, sect193r1, sect193r2, secp192k1, secp192r1, sect163k1, sect163r1, sect163r2, secp160k1, secp160r1, secp160r2
Next Protocol Negotiation	Yes
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

3. Chorme 80 /windows 10

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	Yes
TLS 1.0	Yes
SSL 3	No
SSL 2	No

Cipher Suites (in order of preference)

TLS_GREASE_4A (0x4a4a)	-
TLS_AES_128_GCM_SHA256 (0x1301) Forward Secrecy	128
TLS_AES_256_GCM_SHA384 (0x1302) Forward Secrecy	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) WEAK	128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) WEAK	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) WEAK	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) WEAK	256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) WEAK	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) WEAK	256
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112

Protocol Details

Server Name Indication (SNI)	Yes
Secure Renegotiation	Yes
TLS compression	No
Session tickets	Yes
OCSP stapling	Yes
Signature algorithms	SHA256/ECDSA, RSA_PSS_SHA256, SHA256/RSA, SHA384/ECDSA, RSA_PSS_SHA384, SHA384/RSA, RSA_PSS_SHA512, SHA512/RSA, SHA1/RSA
Named Groups	tls_grease_0a0a, x25519, secp256r1, secp384r1
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	Yes h2 http/1.1
SSL 2 handshake compatibility	No

4. IE 8 / XP

Protocols

TLS 1.3	No
TLS 1.2	No
TLS 1.1	No
TLS 1.0	Yes
SSL 3 INSECURE	Yes
SSL 2	No

Cipher Suites (in order of preference)

TLS_RSA_WITH_RC4_128_MD5 (0x4) INSECURE	128
TLS_RSA_WITH_RC4_128_SHA (0x5) INSECURE	128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa) WEAK	112
TLS_RSA_WITH_DES_CBC_SHA (0x9) INSECURE	56
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x64) INSECURE	56
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x62) INSECURE	56
TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x3) INSECURE	40
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x6) INSECURE	40
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x13) WEAK	112
TLS_DHE_DSS_WITH_DES_CBC_SHA (0x12) INSECURE	56
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x63) INSECURE	56

Protocol Details

Server Name Indication (SNI)	No
Secure Renegotiation	Yes
TLS compression	No
Session tickets	No
OCSP stapling	No
Signature algorithms	-
Named Groups	-
Next Protocol Negotiation	No
Application Layer Protocol Negotiation	No
SSL 2 handshake compatibility	No

Testing vulnerabilities related to weak cipher suits.

Protocol Details

DROWN	Unable to perform this test due to an internal error. (1) For a better understanding of this test, please read this longer explanation (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete INTERNAL ERROR: connect timed out INTERNAL ERROR: connect timed out
Secure Renegotiation	Supported
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Not mitigated server-side (more info) TLS 1.0: 0xc013
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0xc027
GOLDENDOODLE	No (more info) TLS 1.2 : 0xc027
OpenSSL 0-Length	No (more info) TLS 1.2 : 0xc027
Sleeping POODLE	No (more info) TLS 1.2 : 0xc027

Downgrade attack prevention	Yes, TLS_FALLBACK_SCSV supported (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	With modern browsers (more info)
ALPN	Yes h2 http/1.1
NPN	No
Session resumption (caching)	No (IDs assigned but not accepted)
Session resumption (tickets)	Yes
OCSP stapling	Yes
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	Unknown
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No, DHE suites not supported
DH public server param (Ys) reuse	No, DHE suites not supported
ECDH public server param reuse	No
Supported Named Groups	x25519, secp256r1, secp384r1 (server preferred order)
SSL 2 handshake compatibility	No
0-RTT enabled	No

- **Proof of existence of vulnerability.**

Request

```
[NETSPARKER] SSL Connection
```

Response

Response Time (ms) : 1 Total Bytes Received : 27 Body Length : 0 Is Compressed : No

```
[NETSPARKER] SSL Connection
```

This domain supports following weak ciphers:

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xC028)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027

- **Solutions to the above-mentioned vulnerability (Remedy).**

Solution may differ mechanism used to implement the cipher suite. Can suggest some solutions for Apache HTTP server, Nginx, Microsoft IIS, and node js.

1. **Apache**

Change the SSLCipherSuite directive in the Apache configuration file (such as httpd.conf or ssl.conf) to only include strong ciphers. Take out of the list any ciphers that are old or weak.

```
SSLCipherSuite HIGH:!aNULL:!MD5:!3DES
```

2. **Nginx**

Update the ssl_ciphers directive in the Nginx configuration file (such as nginx.conf or ssl.conf) to only contain secure ciphers. Eliminate any ciphers that are weak or exposed**.**

```
ssl_ciphers 'HIGH:!aNULL:!MD5:!3DES';
```

3. **Node js**

Your Node.js application's SSL/TLS configuration must be updated. Only safe ciphers should be included in the cipher's parameter of the https. create Server () function.

```
const https = require('https');
const fs = require('fs');

const options = {
  key: fs.readFileSync('path/to/private-key.pem'),
  cert: fs.readFileSync('path/to/certificate.pem'),
  ciphers: 'HIGH:!aNULL:!MD5:!3DES',
};

const server = https.createServer(options, (req, res) => {
  // Server logic
});

server.listen(443);
```

4. **Microsoft IIS**

For SSL/TLS configuration, use the IIS Manager. Go to your website's "SSL Settings" and, under "Ciphers," only choose strong ciphers. Cross out any weak ciphers from the list.

3. Using components with known vulnerabilities. – reported

i. Possible breach attack -reported.

Under this OWASP category I have identified a vulnerability called BREACH attack is possible in this web application. It is under the domain of **https://www.malwarebytes.com/business/pricing?quantity=%2527.**

- **Impact of BREACH attack on the web application.**

The result of this vulnerability is that an attacker may be able to use this online application to undertake a BREACH (Browser Reconnaissance & Exfiltration through Adaptive Compression of Hypertext) attack.

It is still possible to carry out this type of attack and to breach the data from the online application due to factors that make the BREACH attack conceivable on this web site, even if it is secured with SSL/TLS protected communication.

1. Information leakage: If a BREACH attack is successful, encrypted HTTPS answers may reveal private data. This includes any private data transmitted within the compressed response body, such as CSRF tokens, session IDs, authentication credentials, and others.

2. Compression feature of HTTP exploited: BREACH makes use of the compression capability of HTTP to lower the size of the response body. An attacker can determine whether certain information is present in the answer by altering the content and tracking changes in the compressed response size.

3. Side-channel attack vector: The BREACH attack uses numerous queries to track changes in the size of compressed answers as a side-channel attack vector. An attacker can obtain sensitive information by monitoring the variations in the compressed response sizes and conducting controlled queries while evaluating these changes.

Because of this vulnerability marked as MEDIUM. It's a must to address it with a good solution.

- **Steps to reproduce BREACH attack vulnerability.**

Reproduction of this vulnerability must be done in a controlled environment and at your own risk of anything happening.

1. First need to Setup network traffic intercepting tool like burp suite to capture the request and response to such an attack successfully.

2. Determine which pages that send sensitive data, such as CSRF tokens, session IDs, or other private information. In this case I have identified the page **pricing** sends sensitive data in unencrypted manner. (https://www.malwarebytes.com/business/pricing)

3. Use a tool or write a script to send carefully constructed requests to the target pages. Change the requests to include patterns or secrets you think might be hidden in the compressed answers.

4. Take note of the compressed response size for each well-constructed request. Keep an eye out for any variations in response size when the request's content or secrets are changed.

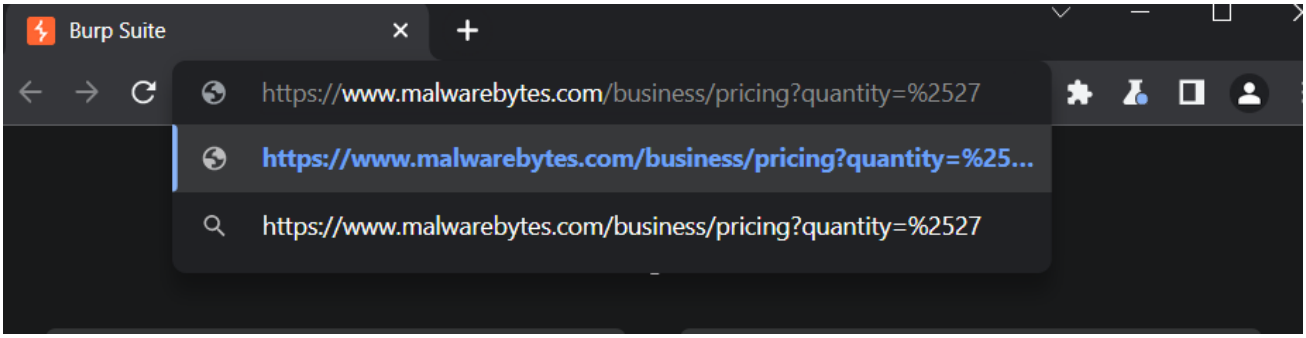
5. Compare the response times for various queries with different secret or content variants. Observe any patterns or noteworthy modifications in the compressed answer sizes.

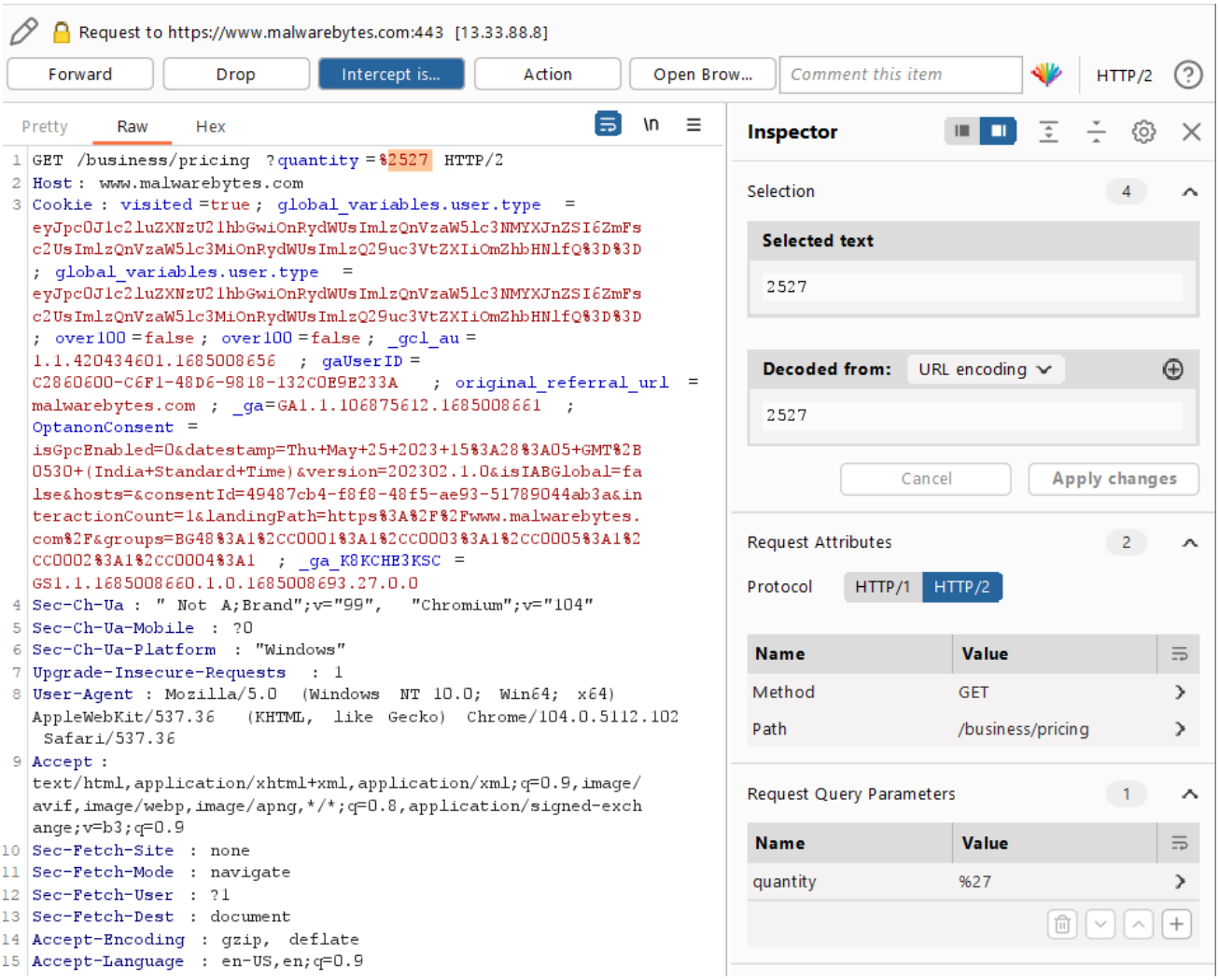
6. Compare the response times for various queries with different secret or content variants. Observe any patterns or noteworthy modifications in the compressed answer sizes.

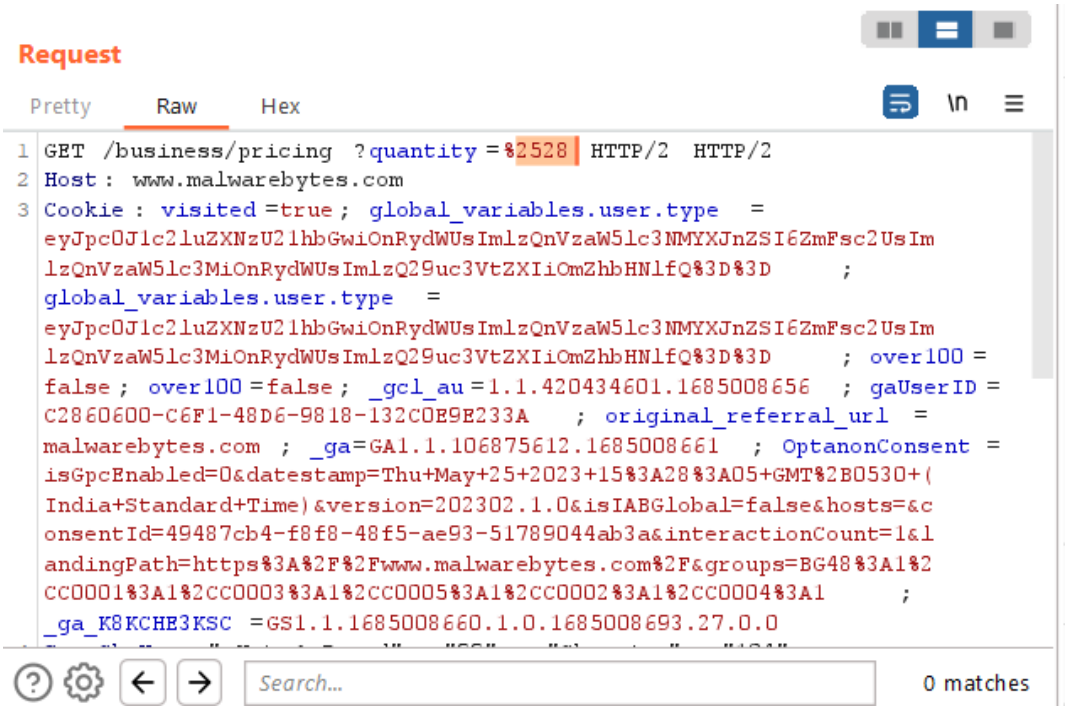
7. Repeat the procedure to further validate the vulnerability by adjusting the created requests or adding new variations. Analyze the variations in response size and check for consistency in your observations.

8. Verify the correlation between changes in response size and the presence or absence of sensitive information. Check to see if the changes in compressed response size can be reliably seen.

- **Proof of concept.**




Request to https://www.malwarebytes.com:443 [13.33.88.8]
Forward
Drop
Intercept is...
Action
Open Brow...
Comment this item
HTTP/2
Pretty
Raw
Hex
1 GET /business/pricing ?quantity =%2527 HTTP/2
2 Host : www.malwarebytes.com
3 Cookie : visited =true ; global_variables.user.type =
eyJpcOJ1c2luZXNzU2lhbGwiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFs
c2UsImlzQnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D
; global_variables.user.type =
eyJpcOJ1c2luZXNzU2lhbGwiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFs
c2UsImlzQnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D
; over100 =false ; over100 =false ; _gcl_au =
1.1.420434601.1685008656 ; gaUserID =
C286D6DD-C6F1-48D6-9818-132C0E9E233A ; original_referral_url =
malwarebytes.com ; _ga=GA1.1.106875612.1685008661 ;
OptanonConsent =
isGpcEnabled=0&datestamp=Thu+May+25+2023+15%3A28%3A05+GMT%2B
0530+(India+Standard+Time)&version=202302.1.0&isIABGlobal=fa
lse&hosts=&consentId=49487cb4-f8f8-48f5-ae93-51789044ab3a&in
teractionCount=1&landingPath=https%3A%2F%2Fwww.malwarebytes.
com%2F&groups=BG48%3A1%2CC0001%3A1%2CC0003%3A1%2CC0005%3A1%2
CC0002%3A1%2CC0004%3A1 ; _ga_K8KCHE3KSC =
GS1.1.1685008660.1.0.1685008693.27.0.0
4 Sec-Ch-Ua : " Not A;Brand";v="99", "Chromium";v="104"
5 Sec-Ch-Ua-Mobile : ?0
6 Sec-Ch-Ua-Platform : "Windows"
7 Upgrade-Insecure-Requests : 1
8 User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102
Safari/537.36
9 Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
ange;v=b3;q=0.9
10 Sec-Fetch-Site : none
11 Sec-Fetch-Mode : navigate
12 Sec-Fetch-User : ?1
13 Sec-Fetch-Dest : document
14 Accept-Encoding : gzip, deflate
15 Accept-Language : en-US,en;q=0.9
Inspector
Selection
4
Selected text
2527
Decoded from:
URL encoding
2527
Cancel
Apply changes
Request Attributes
2
Protocol
HTTP/1
HTTP/2
Name
Value
Method
GET
Path
/business/pricing
Request Query Parameters
1
Name
Value
quantity
%27
763 bytes | 951 millis



- **Proof of existence of vulnerability.**

Reflected Parameter(s)

- quantity

Sensitive Keyword(s)

- token,nonce

Request

```
GET /business/pricing?quantity=%2527 HTTP/1.1
Host: try.malwarebytes.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Connection: Keep-Alive
Cookie: global_variables.user.type=eyJpc0J1c2luZXNzU21hbGwiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFsc2UsImlz
QnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D; global_variables.user.type=eyJpc0J1c2luZXNzU21hbG
wiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFsc2UsImlzQnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D; ove
r100=false; over100=false; visited=true
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 2148.3101 Total Bytes Received : 131583 Body Length : 130663 Is Compressed : No

```
HTTP/1.1 200 OK
set-cookie: ubvs=296d5623-e43a-4a0b-ab9e-6d28a5104fa1; Max-Age=15552000; Path=/; SameSite=Lax
set-cookie: ubvt=v2%7C296d5623-e43a-4a0b-ab9e-6d28a5104fa1%7Cfcbe530a-2a05-4c25-8124-080cc3626596%3Ab%3
Asingle; Max-Age=259200; Domain=malwarebytes.com; Path=/; SameSite=Lax
set-cookie: ubpv=b%2Cfcbe530a-2a05-4c25-8124-080cc3626596; Max-Age=15897600; Path=/business/pricing/10-
99-devices/; SameSite=Lax
x-unbounce-visitorid: 296d5623-e43a-4a0b-ab9e-6d28a5104fa1
link: <https://try.malwarebytes.com/business/pricing/10-99-devices/>; rel="canonical"
x-unbounce-pageid: fcbe530a-2a05-4c25-8124-080cc3626596
x-proxy-backend: page-server
x-unbounce-variant: b
content-length: 17394
content-location: https://try.malwarebytes.com/business/pricing/10-99-devices/
content-type: text/html; charset=utf-8
content-encoding:
date: Mon, 15 May 2023 10:56:25 GMT
etag: "b:296d5623e43a4a0bab9e6d28a5104fa1"

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-stric
t.dtd"><html xmlns="http://www.w3.org/1999/xhtml"><head><META http-equiv="Content-Type" content="text/h
tml; charset=UTF-8" >
<!--fcbe530a-2a05-4c25-8124-080cc3626596 b-->

<title>Pricing: 10-99 Device Support Malwarebytes</title>
<meta name="keywords" content="">
<meta name="description" content="Find out pricing options for cyber security protection for businesses
 with 10-99 endpoints.">
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

1. If possible, disable HTTP level compression.

2. Separate sensitive information from user input

3. Protect vulnerable pages with CSRF token. The Same Site Cookie attribute will mitigate this issue, because to exploit this issue an attacker forces the victim to visit a target website using invisible frames. With the Same Site cookie attribute added, cookies that belong to the target won't be sent with a request that does not include top level navigation.

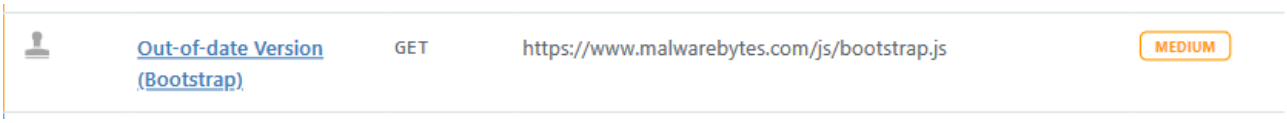
4. Hide the length of the traffic by adding a random number of bytes to the responses.

5. Add in a rate limit, so that the page maximum is reached five times per minute.
6. Implement per-session, randomized CSRF tokens to stop attackers from successfully launching CSRF attacks. To defend against CSRF attacks, think about employing double-submit cookies or other secure techniques.

7. To defend against additional attack routes that could indirectly expose data, like SQL injection and Cross-Site Scripting (XSS), provide appropriate input validation and output encoding.

8. To identify any unusual behavior or anomalies relating to compression, response times, or data leakage, implement monitoring and intrusion detection systems.

ii. Outdated version of bootstrap (XSS) -reported.



Under this OWASP category I have identified this vulnerability too. identified that this version of bootstrap program is vulnerable to **cross site scripting attacks**. Also, there is considerable amount of record to prove that this version is vulnerable to this XSS attacks. Found this vulnerability under the domain of **https://www.malwarebytes.com/js/bootstrap.js.**

- **Impact of using outdated bootstrap version.**

1. Older versions of Bootstrap can include security flaws that have been found and patched in more recent versions. Attackers may take advantage of these flaws to jeopardize the security of your application. To reduce the danger of such vulnerabilities, it is crucial to maintain updated frameworks and libraries.

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 4.1.2, XSS is possible in the collapse data-parent attribute.

Affected Versions

1.0.0 to 3.3.7

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 4.1.2, XSS is possible in the data-container property of tooltip.

Affected Versions

1.0.0 to 3.3.7

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap 3.x before 3.4.0 and 4.x-beta before 4.0.0-beta.2, XSS is possible in the data-target attribute, a different vulnerability than CVE-2018-14041.

Bootstrap Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In Bootstrap before 3.4.0, XSS is possible in the tooltip data-viewport attribute.

2. Updates for Bootstrap frequently include new features, improvements, and additions. You lose out on these upgrades if you continue to use an outdated version, which restricts the functionality and potential of your web application. The most recent responsive design features, component updates, bug fixes, and other advancements made in more recent versions might not be available to you.

3. Performance improvements in Bootstrap's latest iterations frequently include improved code organization, smaller file sizes, and faster rendering. You can lose out on these improvements if you're using an outdated version, which will make the user experience slower and less effective.

Because of this vulnerability marked as MEDIUM It’s a must to address it with a good solution.

- **Steps to reproduce this vulnerability.**

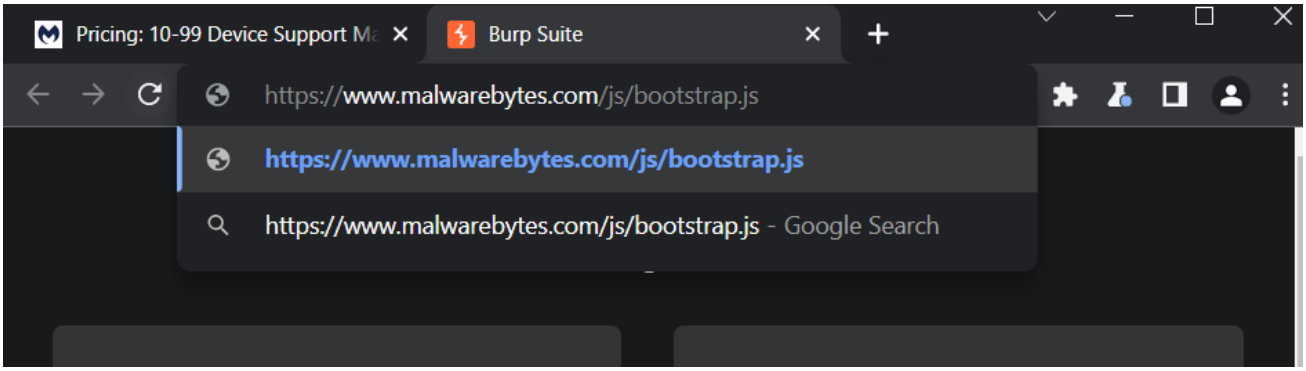
1. Examine the source code of the page.
    - Launch a web browser and access the web application.
    - Open the browser's developer tools by selecting "Inspect" or "Inspect Element" from the context menu when you right-click on the web page.
    - Search the page source code for references to Bootstrap files.

2. Verify Bootstrap's version:
    - Look for the Bootstrap-related CSS and JavaScript files in the page's source code.
    - Search for files with "bootstrap" or "bootstrap.min" and a ".css" or ".js" ending.
    - Determine whether the filenames or contents contain the version number.

3. Referencing the documentation for Bootstrap
    - Go to the documentation page on the official Bootstrap website (getbootstrap.com).
    - Search for the documentation for the version that you noted in the prior step.
    - Compare the code used in your web application with the features, elements, and syntax stated in the documentation.

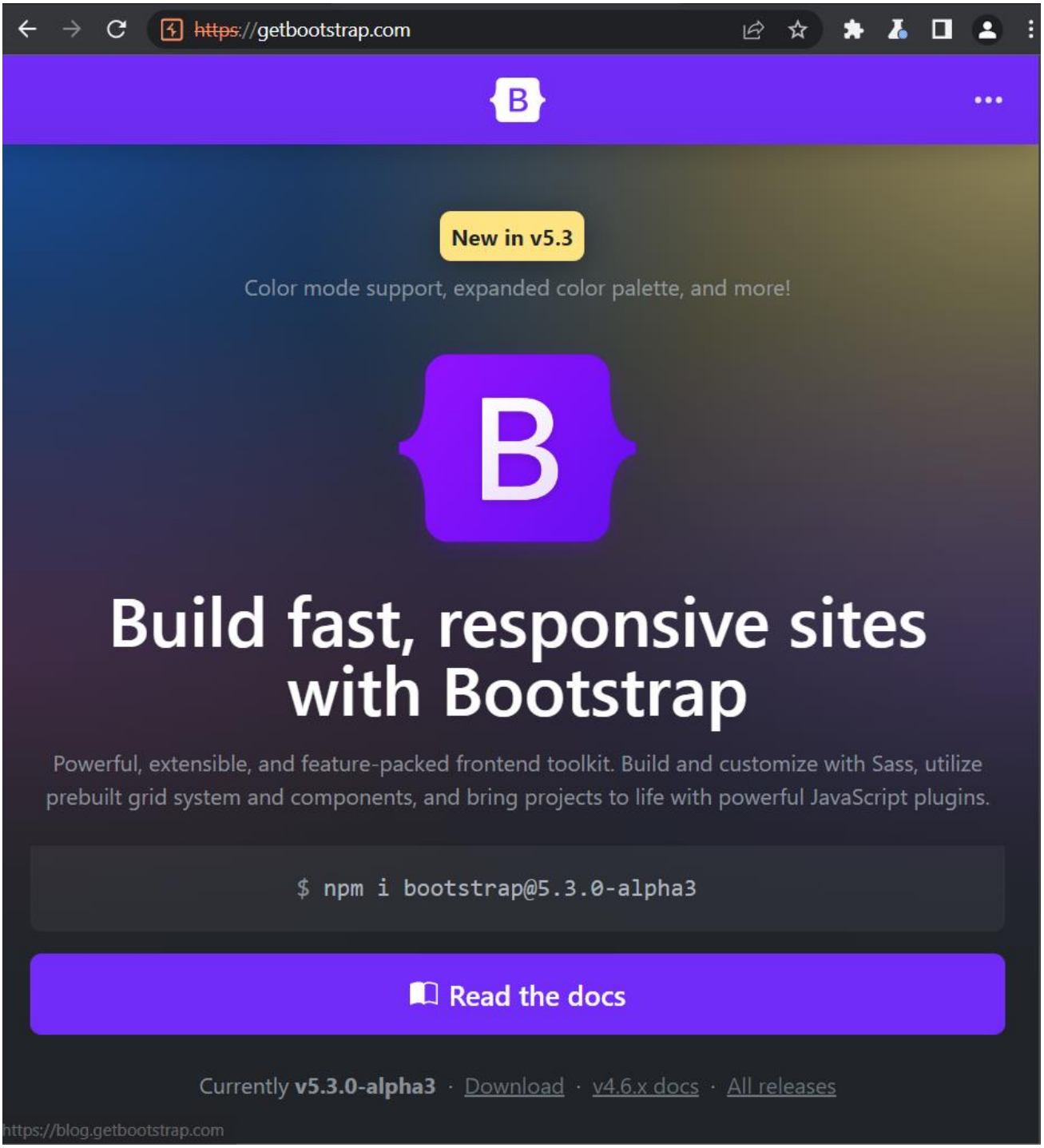
4. Also, the versions of the libraries used in a web application, including Bootstrap, can be determined via automated methods. Outdated library versions can be found with the help of tools like Retire.js, Dependency Check, or built-in security scanners in web development frameworks.

- **Proof of Concept.**



```
/*!
 * Bootstrap v3.3.5 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under the MIT license
 */

if (typeof jQuery === 'undefined') {
    throw new Error('Bootstrap\'s JavaScript requires jQuery')
}

+

function($) {
    'use strict';
    var version = $.fn.jquery.split(' ')[0].split('.')
    if ((version[0] < 2 && version[1] < 9) || (version[0] == 1 && version[1] == 9 && version[2] < 1))
{
        throw new Error('Bootstrap\'s JavaScript requires jQuery version 1.9.1 or higher')
    }
}(jQuery);

/* ========================================================================
 * Bootstrap: transition.js v3.3.5
 * http://getbootstrap.com/javascript/#transitions
 * ========================================================================
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * ======================================================================== */


+
```

```
/*!
 * Bootstrap v3.3.5 (http://getbootstrap.com)
 * Copyright 2011-2015 Twitter, Inc.
 * Licensed under the MIT license
 */
```


https://getbootstrap.com
New in v5.3
Color mode support, expanded color palette, and more!
Build fast, responsive sites with Bootstrap
Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.
$ npm i bootstrap@5.3.0-alpha3
Read the docs
Currently v5.3.0-alpha3 · Download · v4.6.x docs · All releases
https://blog.getbootstrap.com

- **Proof of existence of vulnerability.**

Identified Version

- 3.3.5

Latest Version

- 3.4.1 (in this branch)

Vulnerability Database

- Result is based on 05/09/2023 20:30:00 vulnerability database content.

Request

```
GET /js/bootstrap.js HTTP/1.1
Host: www.malwarebytes.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: global_variables.user.type=eyJpc0J1c2luZXNzU21hbGwiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFsc2UsImlz
QnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D; global_variables.user.type=eyJpc0J1c2luZXNzU21hbG
wiOnRydWUsImlzQnVzaW5lc3NMYXJnZSI6ZmFsc2UsImlzQnVzaW5lc3MiOnRydWUsImlzQ29uc3VtZXIiOmZhbHNlfQ%3D%3D; ove
r100=false; over100=false; visited=true
Referer: https://www.malwarebytes.com/se
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 416.5949 Total Bytes Received : 75915 Body Length : 75275 Is Compressed : No

```
/*!
* Bootstrap v3.3.5(http://getbootstrap.com)
* Copyright 2011-2015 Twitter, Inc.
* Licensed under the MIT license
*/

if (typeof jQuery === 'undefined') {
throw new Error('Bootstrap\'s JavaScript requires jQuery
…
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

1. Find the most recent stable Bootstrap version, then update your web application to use it. New features, performance enhancements, security updates, and bug fixes are often included in the most recent versions. To ensure a seamless move to the new version, follow the migration guides and documentation for Bootstrap.

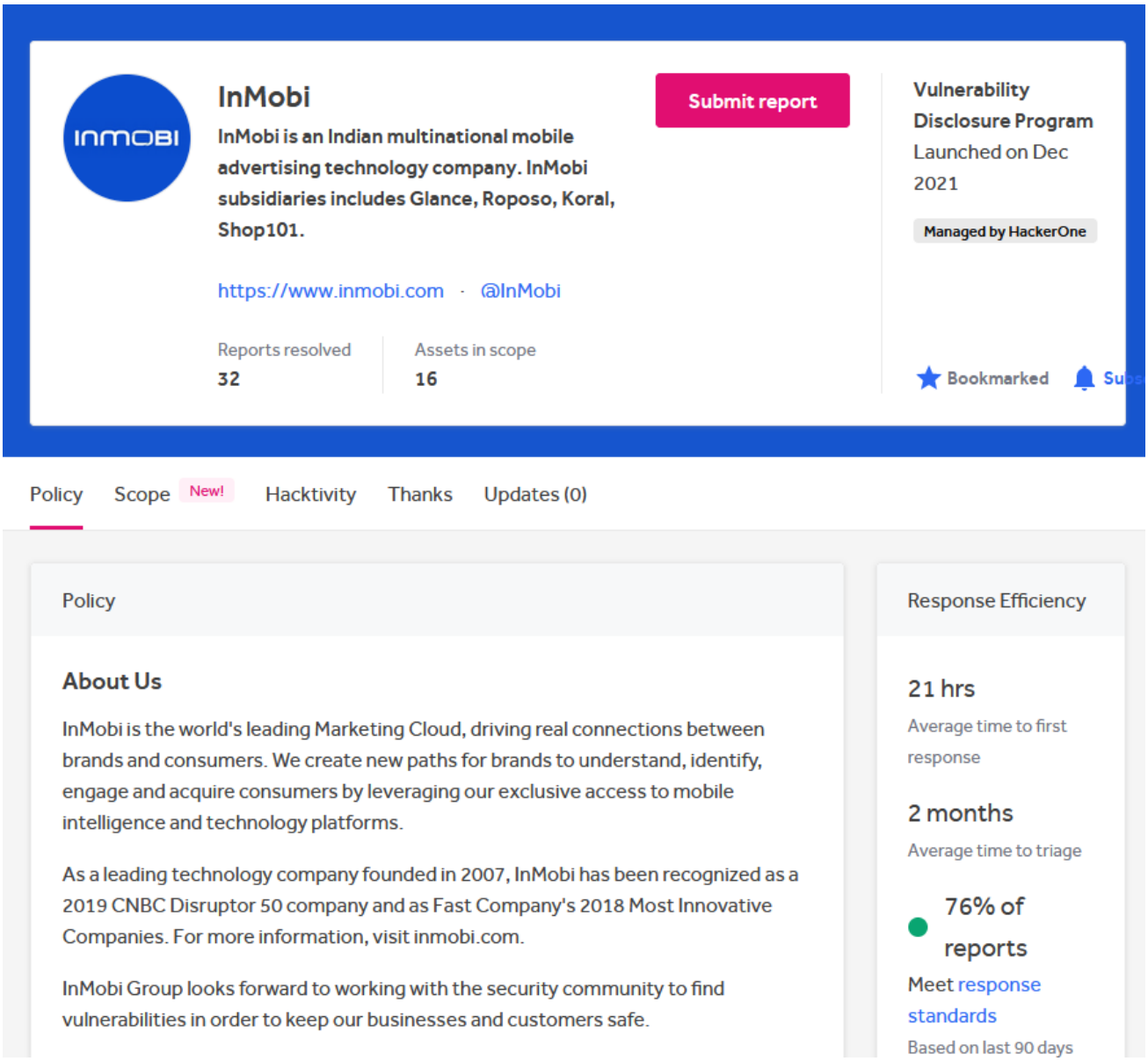
2. If updating to the most recent version is not immediately possible, look for any security updates or Bootstrap-specific changes. Patches for known vulnerabilities may have been released by Bootstrap or the development community. To fix the security flaws in your current version, apply these updates.

3. Update all your dependencies and libraries, including Bootstrap. This makes it possible to guarantee that your web application makes use of the most recent security updates and features. Check for updates frequently, and plan maintenance windows to update libraries as needed.

4. Keep up with any Bootstrap-related security warnings or vulnerability announcements. To get updates on new vulnerabilities and patches, join security newsletters or follow trustworthy sources. Apply any security updates to your web application as soon as possible.

B. Domain: Inmobi.com

Proof of bug bounty program:



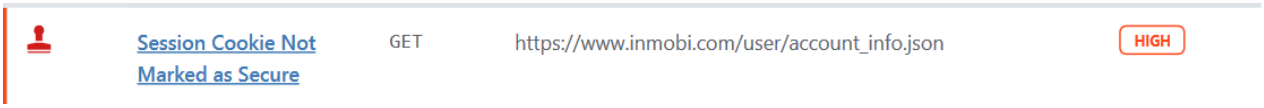
I have identified many vulnerabilities in under domain www.inmboi.com/. I figured out it has vulnerabilities that listed by OWASP Top 10. And the overall website risk level is **high**. I used net sparker to identify vulnerabilities in the following domain. And used some of the previously mentioned tools in section. I identified the following OWASP Top vulnerabilities in following domain.

1. **Sensitive Data exposure**
2. **Using components with known vulnerabilities.**


netsparker
5/16/2023 7:26:48 PM (UTC+05:30)
OWASP Top Ten 2017 Report
https://www.inmobi.com/
Scan Time : 5/5/2023 8:23:16 PM (UTC+05:30)
Scan Duration : 00:00:36:13
Total Requests : 6,231
Average Speed : 2.9r/s
Risk Level:
HIGH
Explanation
This report is generated based on OWASP Top Ten 2017 classification.
There are 11 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.
18
IDENTIFIED
8
CONFIRMED
0
CRITICAL
1
HIGH
6
MEDIUM
7
LOW
2
BEST PRACTICE
2
INFORMATION
Identified Vulnerabilities
Critical 0
High 1
Medium 6
Low 7
Best Practice 2
Information 2
TOTAL 18
Confirmed Vulnerabilities
Critical 0
High 1
Medium 1
Low 4
Best Practice 1
Information 1
TOTAL 8

1. Sensitive data exposure. -Reported

i. Session cookie not marked as secure.



Under this OWASP category I have identified many vulnerabilities like sensitive data exposure that can occur due to bad implementation of security implementations under domain of **https://www.inmobi.com/user/account_info.json**

I have identified while scanning the target domain it is sending the session cookie over a HTTPS enabled connection without marking session cookie as secure. Because of this cookie can be stolen by the attacker by intercepting traffic successfully as **MIMT (Man in the middle attack).** Because of this attacker can still steal the session cookie and can hijack the session of the victim successfully.

If attacker carries and man in the middle attack, he can force the victim to make HTTP request to the web application to steal the Cookie.

- **Impact of not marking session cookie as secure.**

1. The session cookie can be transmitted through unencrypted HTTP connections rather than HTTPS connections if the "secure" tag is not set. This makes it possible for attackers using MitM attacks to intercept the cookie. Attackers may listen in on the conversation and take the session cookie to access the user's session without authorization.

2. An attacker can obtain the user's session identifier using an unsecured session cookie by taking advantage of flaws like session sniffing or session high jacking. Once the attacker has the session identifier, they can use it to assume the user's identity, log into their account, and take actions on their behalf.

3. Insecure session cookies may not meet regulatory compliance standards like the General Data Protection Regulation (GDPR) or Payment Card Industry Data Security Standard (PCI DSS). The absence of the "secure" characteristic may be flagged as non-compliant by security audits as a vulnerability.

4. Failure to declare the session cookie as secure can reduce user confidence in the security of the online application. Users might be reluctant to conduct sensitive acts or transactions on the website, which could harm the app's credibility and reputation.

Because of this vulnerability marked as High It's a must to address it with a good solution.

- **Steps to reproduce this vulnerability.**

1. Start by configuring two unique browsers, Chrome and Firefox, or browser profiles, or by using two distinct devices.

2. Make sure you are logged into your user account in one of the browsers or devices before opening the web application in both.

3. To record the network traffic between the browser and the web server, use a network analysis program like Wireshark.

4. Use a proxy tool like Burp Suite to intercept the login request in the second browser or device when you are not logged in.

5. alter the obtained login request such that the session cookie is obtained. You can accomplish this by copying the session ID value from the "Set-Cookie" header.

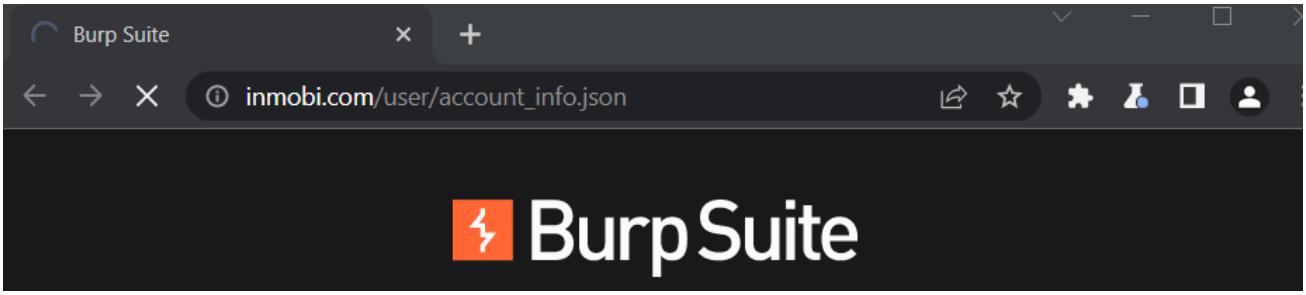
6. To set the value of the session cookie that was captured, open the developer tools on the second browser or device and run the JavaScript script.

7. Use the second browser or device to reload the page or move to a different page inside the online application.

8. If the session hijacking is successful, the web application might acknowledge the session cookie as legitimate and provide you access to the user account without requesting a login.

- **Proof of Concept.**

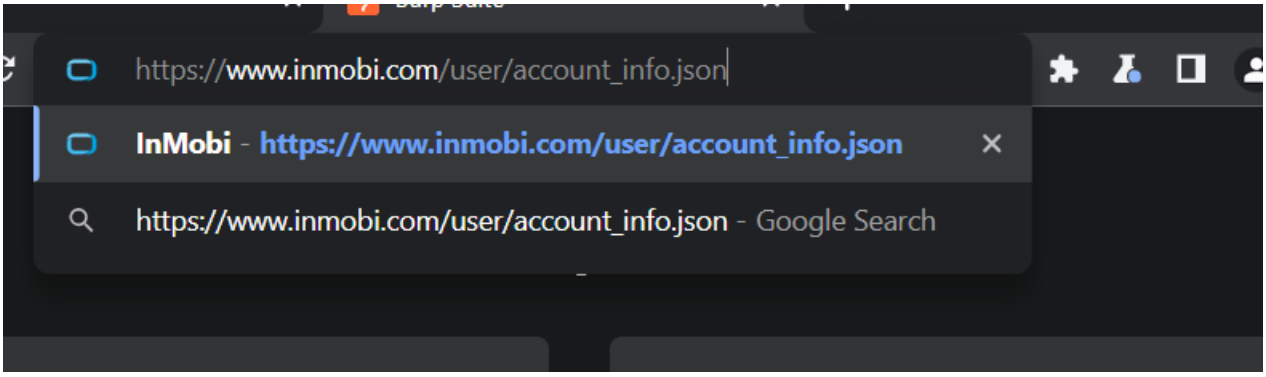
Loging to the web application using legitimate credentials.



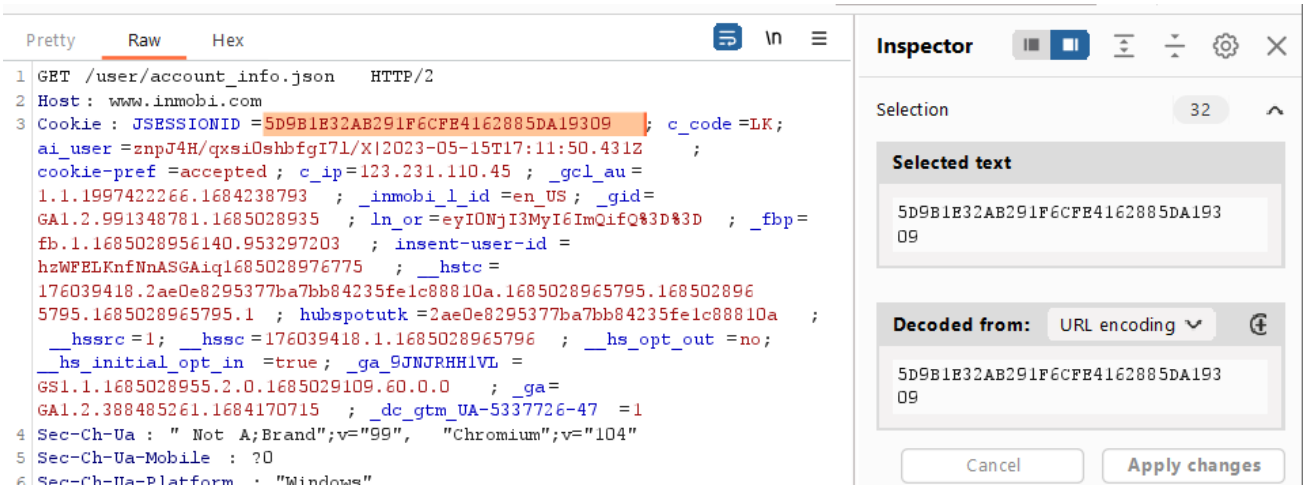
Getting the session cookie to hi jack the session for legitimate user.

```
Pretty  Raw  Hex                                                    \n  ≡
1 GET /user/login.html HTTP/1.1
2 Host : www.inmobi.com
3 Upgrade-Insecure-Requests : 1
4 User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102
  Safari/537.36
5 Accept :
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
  mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
  0.9
6 Accept-Encoding : gzip, deflate
7 Accept-Language : en-US,en;q=0.9
8 Cookie : JSESSIONID =4D5CD7D88E9904FFB9C180C156EEC80A ; cookie-pref =
  accepted ; _gcl_au =1.1.1997422266.1684238793 ; _ga=
  GA1.1.388485261.1684170715 ; _ga_9JNJRHH1VL =
  GS1.1.1684238800.1.1.1684239019.44.0.0
9 Connection : close
0
1
```

Opening the login page using another tab with stolen session cookie.



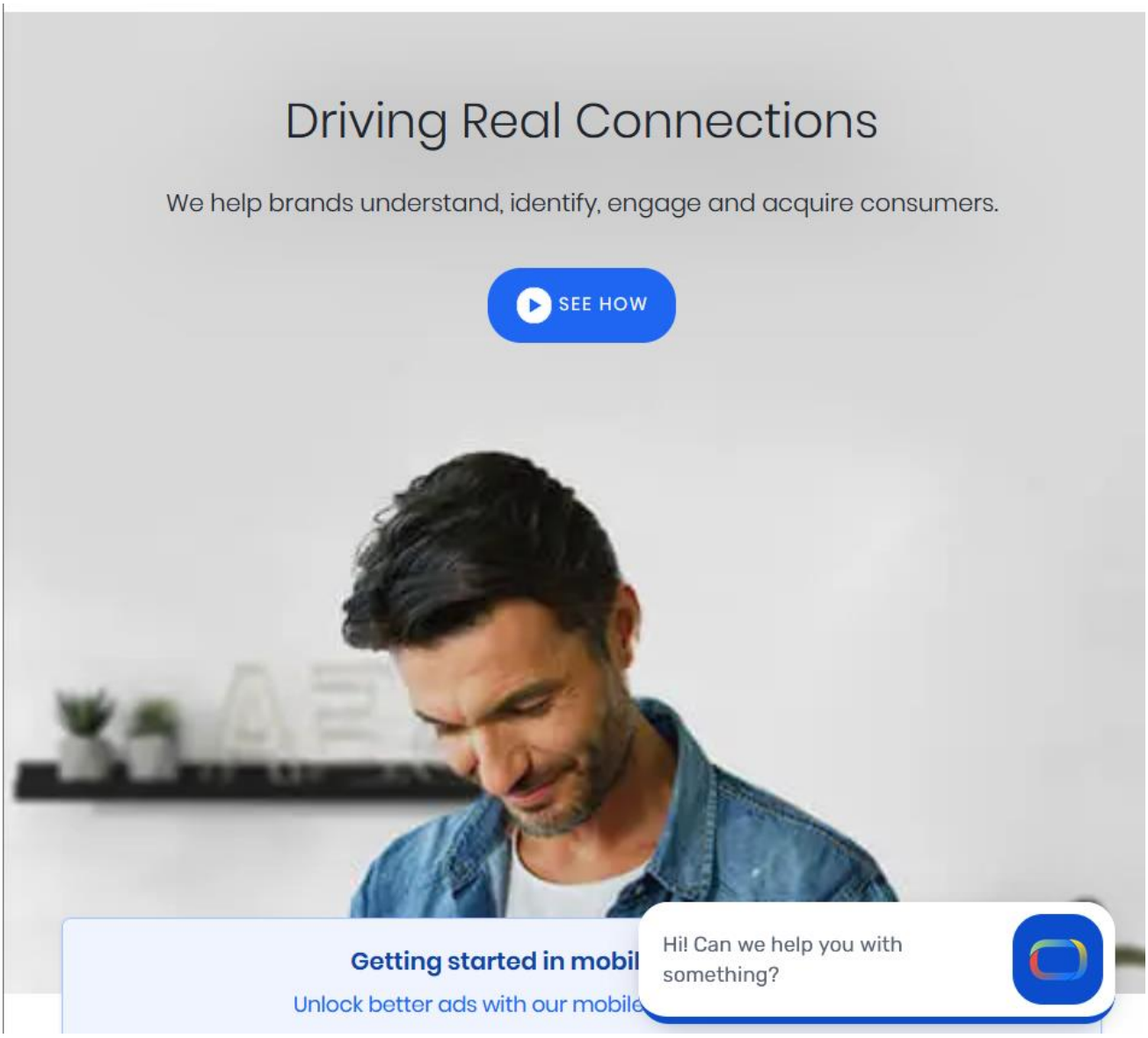
Replacing session cookie with solen one.



```
Pretty  Raw  Hex                                                    \n  ≡
1 GET /user/account_info.json HTTP/2
2 Host : www.inmobi.com
3 Cookie : JSESSIONID =5D9B1E32AB291F6CFE4162885DA19309 ; c_code =LK;
  ai_user =znpJ4H/qxsiOshbfgI7l/X|2023-05-15T17:11:50.431Z ;
  cookie-pref =accepted ; c_ip=123.231.110.45 ; _gcl_au =
  1.1.1997422266.1684238793 ; _inmobi_l_id =en_US ; _gid=
  GA1.2.991348781.1685028935 ; ln_or =eyIONjI3MyI6ImQifQ%3D%3D ; _fbp=
  fb.1.1685028956140.953297203 ; insent-user-id =
  hzWFELKnfNnASGAiq1685028976775 ; __hstc =
  176039418.2ae0e8295377ba7bb84235fe1c88810a.1685028965795.168502896
  5795.1685028965795.1 ; hubspotutk =2ae0e8295377ba7bb84235fe1c88810a ;
  __hssrc =1; __hssc =176039418.1.1685028965796 ; __hs_opt_out =no;
  __hs_initial_opt_in =true; _ga_9JNJRHH1VL =
  GS1.1.1685028955.2.0.1685029109.60.0.0 ; _ga=
  GA1.2.388485261.1684170715 ; _dc_gtm_UA-5337726-47 =1
4 Sec-Ch-Ua : " Not A;Brand";v="99", "Chromium";v="104"
5 Sec-Ch-Ua-Mobile : ?0
6 Sec-Ch-Ua-Platform : "Windows"
```

```
Inspector
Selection 32
Selected text
5D9B1E32AB291F6CFE4162885DA193
09
Decoded from: URL encoding
5D9B1E32AB291F6CFE4162885DA193
09
Cancel  Apply changes
```

Successfully logged in.



- **Proof of existence of vulnerability.**

Identified Cookie(s)

- JSESSIONID

Cookie Source

- HTTP Header

Request

```
GET /user/account_info.json HTTP/1.1
Host: www.inmobi.com
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5,en-US,en;q=0.9
Cache-Control: no-cache
Cookie: c_code=LK; c_ip=116.206.245.109; ai_user=XJhwp5xxtTsAdHzSWDbpg5|2023-05-05T14:53:28.137Z; _csrf
=SS0tMFb0Ge2Rk2a03ojO-51k; exp_csrf_token=8e0a53e214129f87818cafdac2a6c1e3b0057cfd; exp_last_activity=1
683298777; exp_last_visit=1367938472; exp_tracker=%7B%220%22%3A%22rss%2Fblog%22%2C%221%22%3A%22company%
2Fpress%2Fidentity-resolution-and-contextual-targeting-rank-highest-priority-in-newly-released-publishe
r-study-from-inmobi-publisher-insight-survey%2FNetsparkerb84e46f444c64937a4d089da6ebc083b%22%2C%222%22%
3A%22company%2Fpress%2Fidentity-resolution-and-contextual-targeting-rank-highest-priority-in-newly-rele
ased-publisher-study-from-inmobi-publisher-insight-survey%22%2C%223%22%3A%22company%2Fpress%2Finmobi-ap
points-susannah-llewellyn-as-vp-of-agency-partnerships-for-asia-pacific%2FNetsparker7f2afd0a07494901bb7
06ecbeece344e%22%2C%224%22%3A%22company%2Fpress%2Finmobi-appoints-susannah-llewellyn-as-vp-of-agency-pa
rtnerships-for-asia-pacific%2FNetsparkerb71aed894afb457f8c49214b418556b7%22%2C%22token%22%3A%22564c8b59
d0ae47678980b445bab2fe7a372af375f7ffd14604615384d31688c7608ed11265da3b49361c2fda14c5f98e%22%7D; Applica
tionGatewayAffinity=a2208e72bc92267e732a00a46c4d421c; ai_session=R3UgM74Rq7bY17UmmXo+Ak|1683298408159|1
683298807246
```

```
Referer: https://www.inmobi.com/blog/how-are-in-app-advertising-rates-calculated
Request-Id: |fc5db84abed94efb85219f2264d3b71b.2678168f57754290
traceparent: 00-fc5db84abed94efb85219f2264d3b71b-2678168f57754290-01
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Requested-With: XMLHttpRequest
X-Scanner: Netsparker
```

Response

Response Time (ms) : 307.7932 Total Bytes Received : 440 Body Length : 58 Is Compressed : No

```
HTTP/1.1 200 OK
Set-Cookie: _inmobi_l_id=en_US; Domain=.inmobi.com; Path=/; Secure
Set-Cookie: JSESSIONID=F90BE30111BE085F93B39603D54365BC; Path=/user/; HttpOnly

Server: nginx
X-Content-Type-Options: nosniff
Connection: keep-alive
Content-Length: 58
X-Frame-Options: DENY
Content-Type: text/html;charset=UTF-8
Date: Fri, 05 May 2023 15:00:07 GMT
isAuthenticated: false

{"isLogout":true,"url":"/user/logout.html","status":true}
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

1. Make that the "Secure" attribute is set on the session cookie. This attribute tells the browser to send cookies exclusively over HTTPS-encrypted connections. By setting this attribute, you can reduce the chance of an attacker intercepting the session cookie by preventing its transmission via insecure channels.

```
session_set_cookie_params([
    'secure' => true, // Ensure cookie is only sent over HTTPS
    'httponly' => true, // Restrict cookie access to HTTP requests
    'samesite' => 'Lax', // Enforce same-site policy
]);
```

2. Make sure that HTTPS is being used to deliver your complete web application. This guarantees that all communication, including the transmission of session cookies, is encrypted between the client and server. To enforce secure connections, configure your web server to forward all HTTP requests to HTTPS.

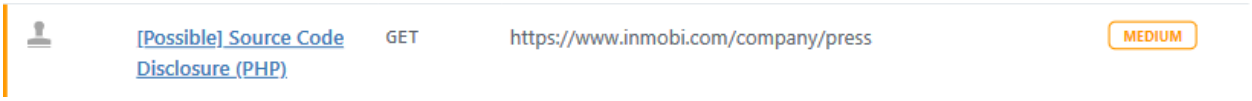
3. Use HSTS to tell the browser to always communicate with your web application via HTTPS. This helps lower the danger of session cookie interception by preventing users from visiting your site using unsecured HTTP connections.

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

4. If a significant event occurs, such as a successful login, password change, or privilege elevation, implement a way to regenerate the session identification. Even if an attacker succeeds in stealing a session cookie, this helps to lessen the effects of session hijacking.

5. Implement a technique to regularly re-authenticate users by asking them to do so after a specific period of inactivity or at predetermined intervals. Even if an attacker obtains access to a legitimate session cookie, this can lessen the possibility of session hijacking.

ii. Source code disclosure PHP.



Under this OWASP category I have identified many vulnerabilities like sensitive data exposure that can occur due to bad implementation of security implementations under domain of **https://www.inmobi.com/company/press**

- **Impact of source code disclosure.**

I discovered a PHP code leak because of this scan. Depending on the sort of source code that was leaked, the impact may include the exposure of the internal workings and business logic of the application, as well as database connection strings, usernames, and passwords.

1. the database or other data sources can be accessed. It may be able to read, update, or delete arbitrary data from the database depending on the privileges of the account determined from the source code.

2. To fully operate the application, acquire access to password-protected administrative tools including dashboards, management consoles, and admin panels.

3. Investigate the source code for input validation issues and logic flaws to develop additional attacks.

4. The exposed source code can be altered by attackers to add backdoors, logic bombs, or other malicious code. Because of this, they may be able to obtain unauthorized access, alter data, take unlawful action, or even launch an attack on users or other systems using your web application.

5. Leaked source code can reveal sensitive data, including API keys, database logins, encryption keys, and other private information. Attackers may use this information to launch additional assaults, obtain illegal access to databases, or jeopardize the security of other systems linked to your web application.

Because of this vulnerability marked as medium It's a must to address it with a good solution.

- **Step to reproduce this vulnerability.**

1. Find the specific PHP file or exposed PHP code by using code analysis tools, code reviews, or log analysis.

2. To find the main reason for the source code leak, employ security scanning tools or a manual inquiry. Misconfigured server settings, unsafe file permissions, an application vulnerability, or other elements could be to blame. OWASP ZAP, Burp Suite, and Nikto are a few examples of tools that might assist in spotting potential vulnerabilities.

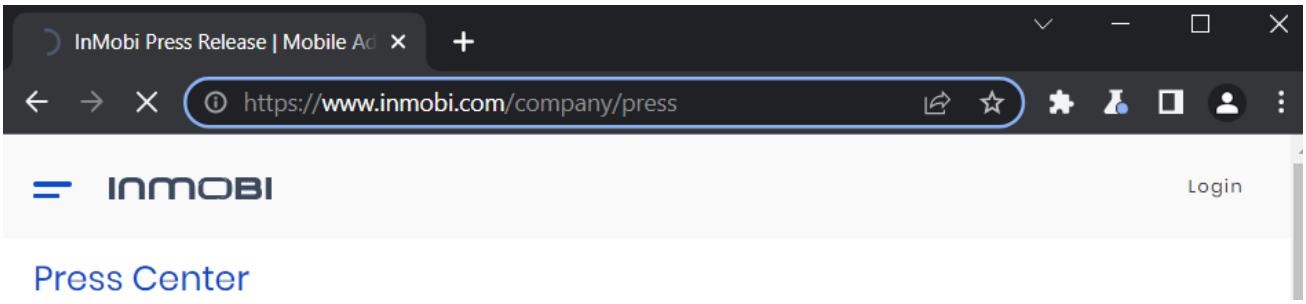
3. Utilize applications like VirtualBox, VMware, or Docker to create a local or isolated testing environment. You can use this to replicate and examine the vulnerability without having an adverse effect on the production system.

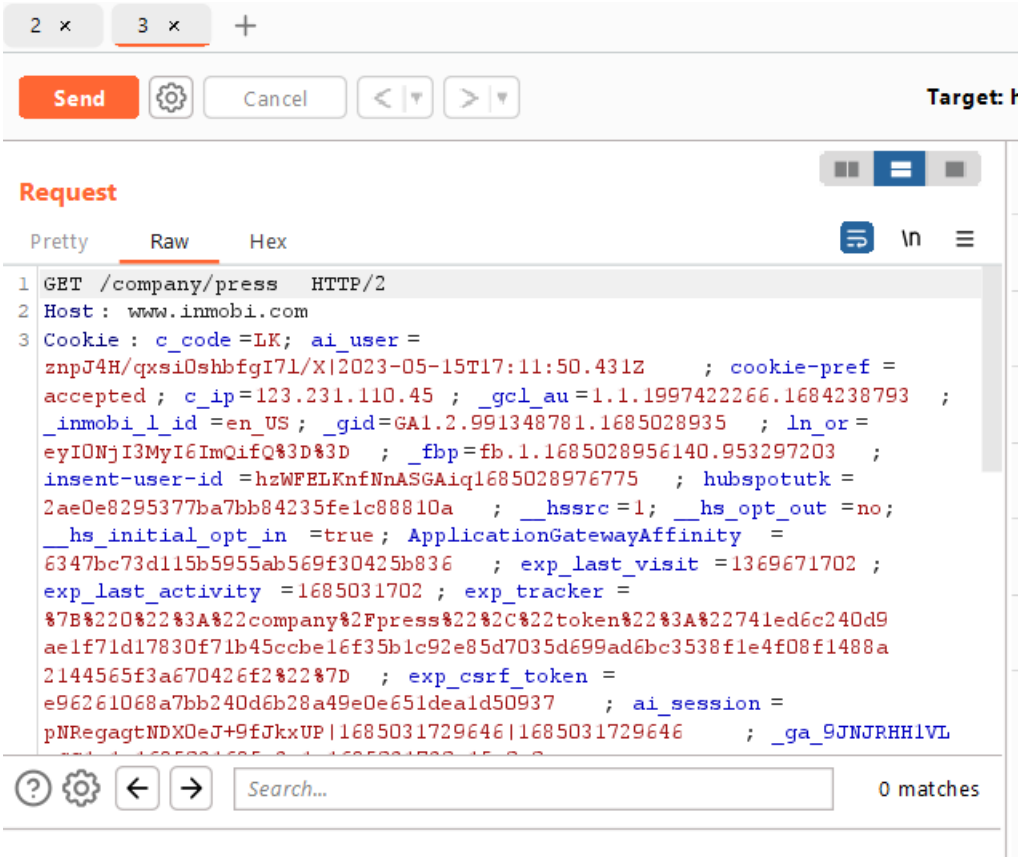
4. To duplicate the situation in which the source code fragment is exposed, modify the server settings, or take advantage of the vulnerability. To exploit vulnerabilities or replicate misconfigurations, you can utilize programs like Metasploit, OWASP ZAP, or your own scripts.

5. Access the weak PHP file or exposed fragment using a web browser or an HTTP client program like cURL or Postman. In the address bar or as part of the request, type the URL or path to the PHP file.

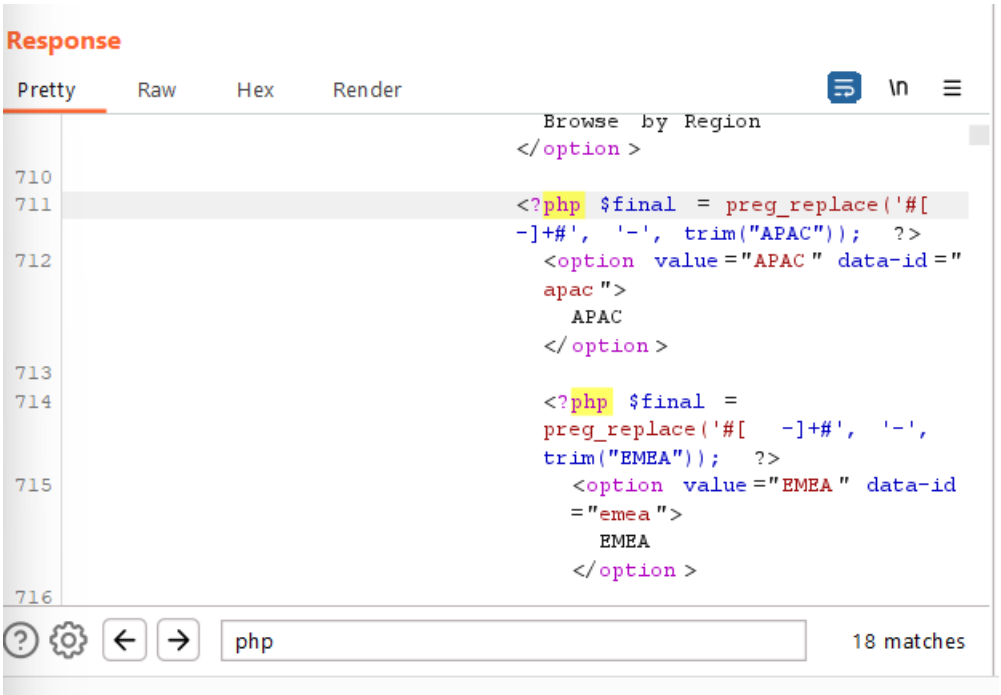
6. Verify that the PHP source code snippet is available and leaked. Examine the response in the browser or use programs like Burp Suite, ZAP, or Wireshark to analyze the server response. Verify that the code is being exposed as expected by looking at the content the server has returned.

- **Proof of concept.**





Leaky PHP code



- **Proof of existence of vulnerability.**

Identified Source Code

```
<?php $final = preg_replace('#[ -]+#', '-', trim("APAC")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("EMEA")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("Europe")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("Global")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("India")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("LATAM")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("MENA")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("North America")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("SEA")); ?>
…
<?php $final = preg_replace('#[ -]+#', '-', trim("APAC")); ?>
…
<?p...
```

Request

```
GET /company/press HTTP/1.1
Host: www.inmobi.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: c_code=LK; c_ip=116.206.245.109; ai_user=XJhwp5xxtTsAdHzSWDbpg5|2023-05-05T14:53:28.137Z; ai_se
ssion=R3UgM74Rq7bY17UmmXo+Ak|1683298408159|1683298484523; ApplicationGatewayAffinity=a2208e72bc92267e73
2a00a46c4d421c; exp_csrf_token=8e0a53e214129f87818cafdac2a6c1e3b0057cfd; exp_last_activity=1683298518;
 exp_last_visit=1367938472; exp_tracker=%7B%220%22%3A%22rss%2Finsights%22%2C%221%22%3A%22rss%2Fpress%2
2%2C%222%22%3A%22rss%2Fpress%2FN3TSP4RKE2%22%2C%223%22%3A%22rss%2Fpress%2Fnxtspxrkex%22%2C%22token%22%3
A%22ee7a1a0b00535968ed6c9fb381e573c2e12402ac5f83c7b0771d8f96644dd5c6ddb6fb470accc9f458fa8e22784b0f59%2
2%7D
Referer: https://www.inmobi.com/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 6494.2685 Total Bytes Received : 560662 Body Length : 559432 Is Compressed : No

```
HTTP/1.1 200 OK
Set-Cookie: ApplicationGatewayAffinity=c4e775a69ade163c9ae3e4cb423ea79b; Path=/
Set-Cookie: exp_last_visit=1367938472; expires=Sat, 04-May-2024 14:55:19 GMT; Max-Age=31536000; path=/;
 HttpOnly; SameSite=Lax
Set-Cookie: exp_last_activity=1683298519; expires=Sat, 04-May-2024 14:55:19 GMT; Max-Age=31536000; path
=/; HttpOnly; SameSite=Lax
Set-Cookie: exp_tracker=%7B%220%22%3A%22company%2Fpress%22%2C%221%22%3A%22rss%2Finsights%22%2C%222%22%3
A%22rss%2Fpress%22%2C%223%22%3A%22rss%2Fpress%2FN3TSP4RKE2%22%2C%224%22%3A%22rss%2Fpress%2Fnxtspxrkex%2
2%2C%22token%22%3A%22fe4eb58f929b17238d28614bb9ef2117542855655c705adf001ec8bed45641c7de6bd4f7ddf00903ab
f7cbdeb62c757e%22%7D; path=/; HttpOnly; SameSite=Lax
Set-Cookie: exp_csrf_token=8e0a53e214129f87818cafdac2a6c1e3b0057cfd; expires=Fri, 05-May-2023 16:55:19
 GMT; Max-Age=7200; path=/; HttpOnly; SameSite=Lax
Server: nginx
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Last-Modified: Fri, 05 May 2023 14:55:22 GMT
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Content-Encoding:
X-Forwarded-For: 116.206.
…
```

```
<select class="select-dropdown">
<option value="Browse by Region" data-id="browse-by-regions">Browse by Region</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("APAC")); ?>
<option value="APAC" data-id="apac">APAC</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("EMEA")); ?>
<option value="EMEA" data-id="emea">EMEA</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("Europe")); ?>
<option value="Europe" data-id="europe">Europe</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("Global")); ?>
<option value="Global" data-id="global">Global</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("India")); ?>
<option value="India" data-id="india">India</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("LATAM")); ?>
<option value="LATAM" data-id="latam">LATAM</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("MENA")); ?>
<option value="MENA" data-id="mena">MENA</option>
```

```
<?php $final = preg_replace('#[ -]+#', '-', trim("North America")); ?>
<option value="North America" data-id="north-america">North America</option>

<?php $final = preg_replace('#[ -]+#', '-', trim("SEA")); ?>
<option value="SEA" data-id="sea">SEA</option>

</select>
</div>
</div>

<!-- Mobile Filter -->
<div cl
…
</div>
<div class="filter-content">
<h4>Region</h4>
<ul class="filter-list" id="reg-filter">

<?php $final = preg_replace('#[ -]+#', '-', trim("APAC")); ?>
<li data-id="apac"><span>APAC</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("EMEA")); ?>
<li data-id="emea"><span>EMEA</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("Europe")); ?>
<li data-id="europe"><span>Europe</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("Global")); ?>
<li data-id="global"><span>Global</span></li>
```

```
<?php $final = preg_replace('#[ -]+#', '-', trim("India")); ?>
<li data-id="india"><span>India</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("LATAM")); ?>
<li data-id="latam"><span>LATAM</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("MENA")); ?>
<li data-id="mena"><span>MENA</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("North America")); ?>
<li data-id="north-america"><span>North America</span></li>

<?php $final = preg_replace('#[ -]+#', '-', trim("SEA")); ?>
<li data-id="sea"><span>SEA</span></li>

</ul>
</div>
</div>
<div class="filter-submit"><i
…
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

As solutions there are lots of ways that can achieve the expected security about source code disclosure. Among them there are some methods that can be mentioned as good remedy methods.

1. Verify precisely which portions of the source code are really leaked; in certain cases, this may not be feasible because of the restrictions of this sort of vulnerability. Verify that this feature was not intended.

2. Remove any sensitive data from the source code files, including API keys, database logins, and other private information. Use the necessary techniques (e.g., environment variables) to incorporate sensitive information in the PHP files by storing it in a secure configuration file that isn't in the web root directory.

3. If it's a file that the application needs, modify its permissions to bar anyone from accessing it. Remove it from the web server if it isn't.

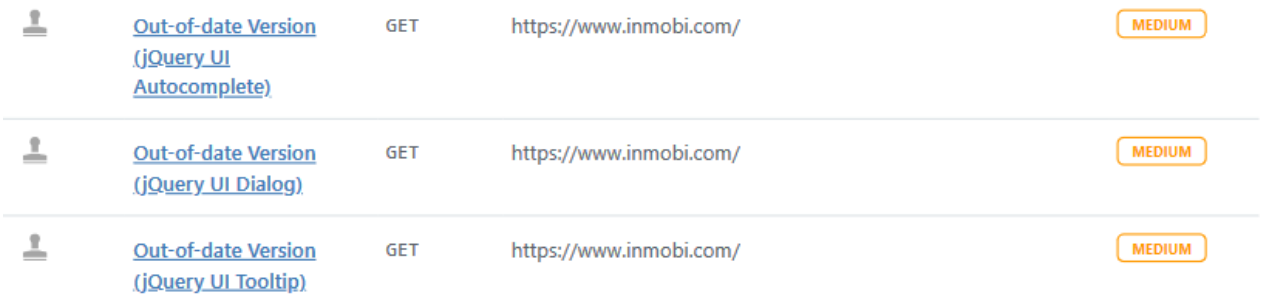
4. Verify that all recent security fixes have been installed on the server.

5. Delete all backup and temporary files from the web server.

6. Use secure code deployment techniques to reduce the possibility of source code leakage. Using secure file transfer protocols (such as SFTP and SCP) and adhering to secure coding standards, such as input validation, output encoding, and defense against code injection attacks, are examples of how to do this.

2. Using components with known vulnerabilities.

i. Outdated jQuery version (UI Autocomplete/UI Dialog/UI Tool tip.



	Out-of-date Version (jQuery UI Autocomplete)	GET	https://www.inmobi.com/	MEDIUM
	Out-of-date Version (jQuery UI Dialog)	GET	https://www.inmobi.com/	MEDIUM
	Out-of-date Version (jQuery UI Tooltip)	GET	https://www.inmobi.com/	MEDIUM

I have identified the vulnerability that falls under OWASP top 10 list as Using components with known vulnerabilities. I found it under domain of **https://www.inmobi.com/.**

- **Impact of using outdated jQuery version.**

1. Since this is an older version of the software series there are some records that mentioned vulnerabilities in the following version vulnerabilities. Says that this version is vulnerable to **cross site scripting attacks (XSS)**.

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `altField` option of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `altField` option is now treated as a CSS selector. A workaround is to not accept the value of the `altField` option from untrusted sources.

Affected Versions
1.12.0 to 1.12.1

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `of` option of the `.position()` util from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `of` option is now treated as a CSS selector. A workaround is to not accept the value of the `of` option from untrusted sources.

Affected Versions
1.12.0 to 1.12.1

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of various `*Text` options of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. The values passed to various `*Text` options are now always treated as pure text, not HTML. A workaround is to not accept the value of the `*Text` options from untrusted sources.

2. In older variations of jQuery, performance optimizations and efficiency enhancements may not be present. This may result in sluggish page loads, longer response times, or wasteful use of system resources. The user experience may suffer as a result, particularly when working with huge datasets or intricate UI interactions.

3. Older jQuery releases might not have important security improvements added since then. Improved output encoding, input sanitization, and defenses against typical security risks are a few examples of these improvements. You could unintentionally impair your application's security safeguards and increase its vulnerability to attacks if you use an outdated version.

4. Your application's attack surface is increased if you're still using an old version of jQuery. Since attackers are aware of the flaws in certain versions, they can target them specifically to undermine the security of your application. By failing to update to the most recent version of jQuery, you give attackers a known point of entry for exploitation.

Because of this vulnerability marked as medium It's a must to address it with a good solution.

- **Step to reproduce this vulnerability.**

1. Identify the feature that uses the jQuery version that is vulnerable. Concentrate on locations that receive user input, including input fields, form submissions, or AJAX calls that communicate with the exposed jQuery functions. Can use burp suite, chrome dev tools or code editor.

2. To create a malicious payload including JavaScript code, use Burp Suite, OWASP ZAP, or online XSS payload generators. For example, `<script>alert('XSS') </script>`

3. To access the correct page of your web application, use a web browser (such as Google Chrome, Firefox, or Microsoft Edge). Place the carefully constructed payload in input forms, query parameters, or any other areas that take user input. To change HTML or JavaScript code dynamically, utilize the browser developer tools.

4. Once the payload has been injected, observe the browser's actions. Check to see if the payload is performed and if any unexpected activity, such as an alert box, takes place. To verify that the payload was successfully executed, you can analyze network requests and the DOM with the aid of browser development tools. To inject can use burp suite or OWASP ZAP or any.

5. To make sure that the vulnerability is consistent, run the test again with various payload changes and more scenarios. Check to see if the XSS vulnerability can be continuously reproduced and if it compromises the security of your application.

- **Proof of existence of vulnerability.**

Identified Version

- 1.12.1

Latest Version

- 1.12.1 (in this branch)

Vulnerability Database

- Result is based on 05/03/2023 20:30:00 vulnerability database content.

Request

```
GET / HTTP/1.1
Host: www.inmobi.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1579.7632 Total Bytes Received : 77124 Body Length : 76298 Is Compressed : No

```
HTTP/1.1 200 OK
X-DNS-Prefetch-Control: off
ETag: "12a0a-OY/LHtYibZf+jJG2s4tEf1LSzSM"
Set-Cookie: ApplicationGatewayAffinity=a2208e72bc92267e732a00a46c4d421c; Path=/
Set-Cookie: c_code=LK; Max-Age=31556952; Path=/; Expires=Sat, 04 May 2024 20:42:29 GMT; HttpOnly; Secur
e
Set-Cookie: c_ip=116.206.245.109; Max-Age=31556952; Path=/; Expires=Sat, 04 May 2024 20:42:29 GMT; Http
Only; Secure
Strict-Transport-Security: max-age=15552000; includeSubDomains
Transfer-Encoding: chunked
X-Powered-By: Next.js
Server: nginx
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Connection: keep-alive
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
Vary: Accept-Encoding
Content-Type: text/html; charset=utf-8
X-Forwarded-For: 116.206.245.109
Date: Fri, 05 May 2023 14:53:17 GMT
Content-Encoding:
```

<!DOCTYPE html><html lang="en"><head><meta charSet="utf-8"/><title>InMobi Mobile Marketing Platform For Advertisers And Publishers</title><meta property="og:title" content="InMobi Mobile Marketing Platform For Advertisers And Publishers"/><meta name="description" content="InMobi Mobile Marketing Platform to grow your business. Take leverage of InMobi's intelligence to identify, engage and acquire your best customers for your business."/><meta http-equiv="Content-Type" content="text/html; charset=utf-8"/><meta name="viewport" content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0"/><link rel="canonical" href="https://www.inmobi.com/"/><meta name="robots" content="index, follow, noodp, noydir"/><meta property="og:locale" content="en_us"/><meta property="fb:admins" content="585110623"/><meta property="og:url" content="https://www.inmobi.com/"/><meta property="og:site_name" content="InMobi"/><meta property="og:type" content="website"/><meta property="og:image" content="https://web.inmobicdn.net/website/website/6.0.1/ui/uploads/misc/InMobi_Group_logo_color.png"/><meta name="google-site-verification" content="7nispBWYRSS2ALAFEj
…

```
┌──(root㉿error404)-[~]
└─# nikto -h http://www.inmobi.com
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          20.81.69.107
+ Target Hostname:    www.inmobi.com
+ Target Port:        80
+ Start Time:         2023-05-16 18:43:56 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Microsoft-Azure-Application-Gateway/v2
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
ing-content-type-header/
+ Root page / redirects to: https://www.inmobi.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ 8074 requests: 0 error(s) and 2 item(s) reported on remote host
+ End Time:           2023-05-16 19:26:08 (GMT5.5) (2532 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

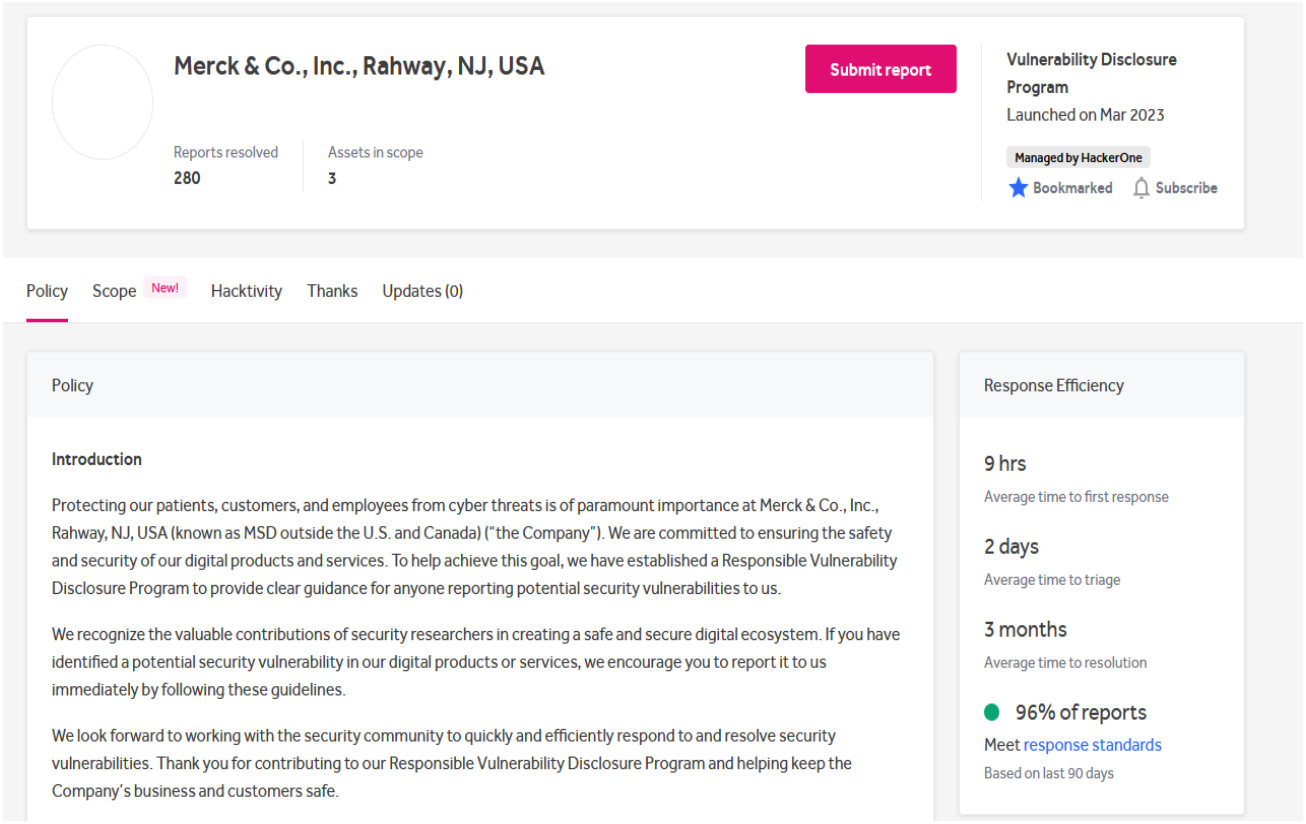
1. Upgrade to a more recent and secure version of jQuery. Update your application to comply with the most recent stable version by visiting the jQuery website or GitHub repository. You can gain access to bug fixes, security patches, and better code quality by updating to a newer version.

2. To reduce the danger of cross-site scripting (XSS) attacks, use a content security policy. You can set a policy with CSP that limits the kinds of content that can be loaded and run on your web pages. Malicious scripts that are injected using weak components can be stopped from running by properly specifying CSP directives.

3. Conduct routine security audits and code reviews of the front-end code of your web application, paying particular attention to the jQuery usage and the custom code for the dialog, autocomplete, and tooltip functionalities. Keep an eye out for any potential security holes and take aggressive measures to fix them.

C. Domain: merck.com

Proof of bug bounty program:



I have identified many vulnerabilities in under domain **www.merck.com/**. I figured out it has vulnerabilities that listed by OWASP Top 10. And the overall website risk level is **high**. I used net sparker to identify vulnerabilities in the following domain. And used some of the previously mentioned tools in section. I identified the following OWASP Top vulnerabilities in following domain.

1. **Using components with known vulnerabilities.**

netsparker

5/17/2023 12:59:26 AM (UTC+05:30)

OWASP Top Ten 2017 Report

https://www.merck.com/

Scan Time	: 5/5/2023 9:19:37 PM (UTC+05:30)
Scan Duration	: 00:00:07:14
Total Requests	: 6,821
Average Speed	: 15.7r/s

Risk Level:
HIGH

Explanation

This report is generated based on OWASP Top Ten 2017 classification.
There are 12 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.

8 IDENTIFIED	3 CONFIRMED	0 CRITICAL
1 HIGH	2 MEDIUM	3 LOW
	1 BEST PRACTICE	1 INFORMATION

Identified Vulnerabilities

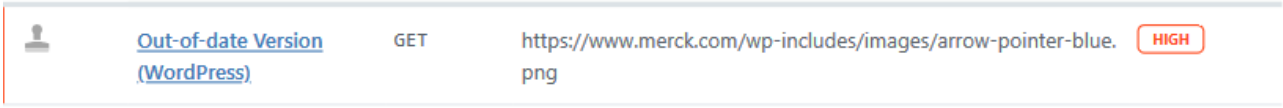
Critical	0
High	1
Medium	2
Low	3
Best Practice	1
Information	1
TOTAL	**8**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	1
Best Practice	0
Information	1
TOTAL	**3**

1. Using components with known vulnerabilities.

i. Outdated WordPress (possible SSRF).



Under this OWASP category I have identified vulnerabilities like Using components with known vulnerabilities that can occur due to bad implementation of security implementations under domain of **https://www.merck.com/wp-includes/images/arrow-pointer-blue.png**

Detected that web application using outdated word press. Word press is a free and open-source content management system (CMS) based on PHP and MySQL.

- **Impact of using an outdated WordPress**

1. WordPress versions that are out of date are more likely to include security flaws. Attackers may use these flaws to modify data, execute arbitrary commands, introduce malicious code, or obtain unauthorized access. The longer a version is out-of-date, the more likely it is that automated scripts and attackers searching for known vulnerabilities will target it.

WordPress Time-of-check Time-of-use (TOCTOU) Race Condition Vulnerability

WordPress is affected by an unauthenticated blind SSRF in the pingback feature. Because of a TOCTOU race condition between the validation checks and the HTTP request, attackers can reach internal hosts that are explicitly forbidden.

Affected Versions
6.0.3 to 6.1.1

WordPress Uncontrolled Resource Consumption Vulnerability

WordPress through 6.1.1 depends on unpredictable client visits to cause wp-cron.php execution and the resulting security updates, and the source code describes "the scenario where a site may not receive enough visits to execute scheduled tasks in a timely manner," but neither the installation guide nor the security guide mentions this default behavior, or alerts the user about security risks on installations with very few visits.

Affected Versions
6.0.3 to 6.1.1

2. You cannot access the most recent security fixes and upgrades if your WordPress installation is out of date. These upgrades fix known security flaws and aid in shielding your website from prospective attackers. You expose your website to known security dangers by not updating your WordPress version.

3. It's possible that performance optimizations added to current WordPress versions are absent in older versions. It's possible that outdated versions lack performance advancements like caching techniques, database improvements, or code optimizations. As a result, the functionality of your

website can be compromised, which might result in slower page loads, a worse user experience, and even lower search engine results.

4. Official support for obsolete WordPress versions gradually decreases over time. This implies that it can be difficult to get support from the WordPress community or developers if you run into any problems or need help. If you continue using an outdated version, you can lose access to vital support tools.

Because of this vulnerability marked as High It's a must to address it with a good solution.

- **Step to reproduce this vulnerability.**

1. Choose the precise WordPress version web application uses and Keep track of the precise version that is vulnerable to uncontrolled resource use.

2. On your website, produce a heavy load or carry out actions that use plenty of resources. This can be done in several ways, including:
    - using tools for load testing, such as Apache JMeter, Siege, or Locust, to simulate a high volume of concurrent user requests. running resource-intensive plugins or themes, or creating posts, uploading files, and performing several simultaneous tasks.
    - sending several queries to vulnerable endpoints or features in a targeted manner.

3. While simulating a heavy load, keep an eye on how your testing environment is using its resources. Monitor database connections and queries, as well as the server's CPU, memory, and disk consumption.

4. Look for evidence that the system resources are not being correctly managed or controlled, such as performance degradation, sluggish response times, server crashes, excessive memory utilization, or other signals.

5. To find any resource-related issues, such as memory leaks, database connection restrictions, or ineffective resource utilization, analyze server logs, performance metrics, and error reports.

- **Proof of existence of vulnerability.**

Identified Versions

- 6.0.3, 6.0.2

Latest Version

- 6.0.3 (in this branch)

Vulnerability Database

- Result is based on 05/03/2023 20:30:00 vulnerability database content.

Request

```
GET /wp-includes/images/arrow-pointer-blue.png HTTP/1.1
Host: www.merck.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 286.9638 Total Bytes Received : 1917 Body Length : 1569 Is Compressed : No

```
Binary response detected, response has not saved.
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

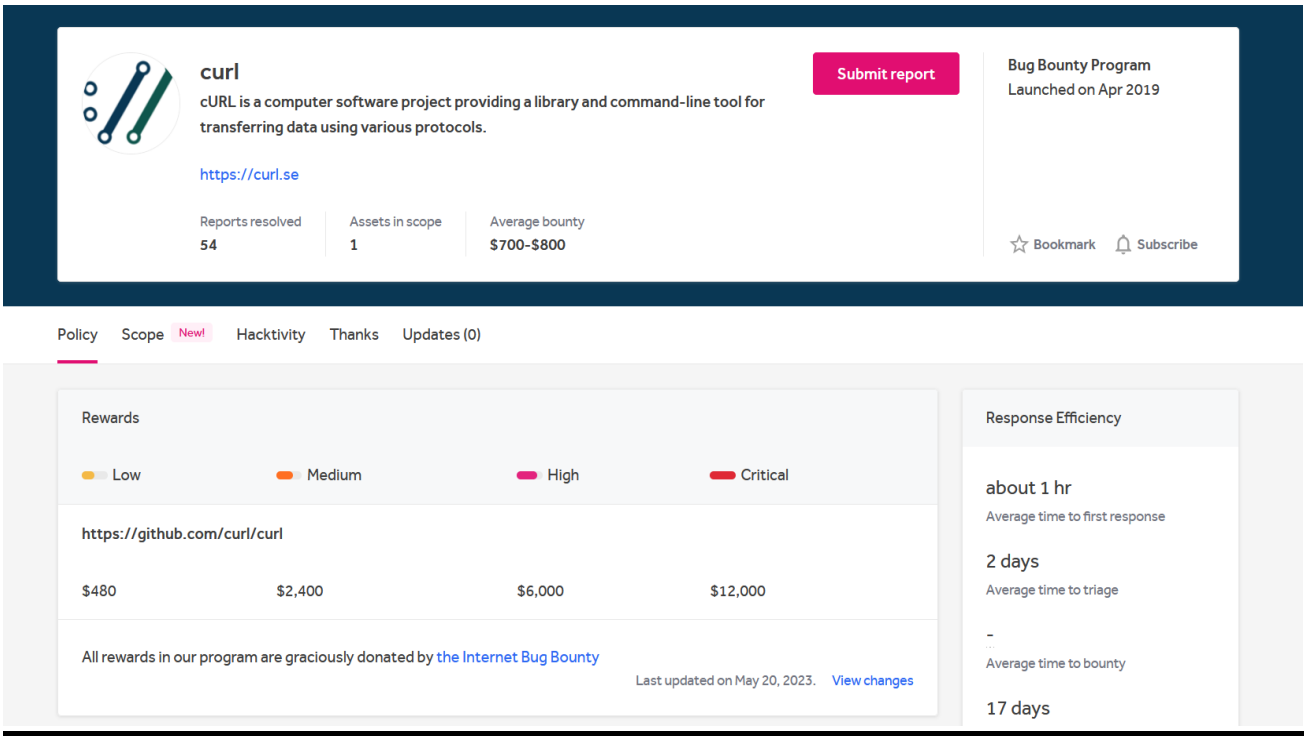
1. Updating your WordPress installation to the most recent stable version is the best solution. Security patches, bug fixes, and performance enhancements are frequently included in updates.

2. Check the code of your WordPress theme and plugins for any possible security flaws. usage secure coding techniques to reduce the risk of common vulnerabilities like XSS (Cross-Site Scripting) and SQL injection. These techniques include input validation, output sanitization, and effective usage of WordPress APIs.

3. To add an extra layer of security to your WordPress website, install and set up reliable security plugins. Strong password requirements, the implementation of firewall rules, the detection and mitigation of common vulnerabilities, and routine security scans can all be assisted by these plugins.
4. To find any potential vulnerabilities in your WordPress installation, themes, or plugins, do vulnerability checks periodically using specialist tools like WPScan or security plugins. As soon as vulnerabilities are found, update or replace the impacted components.

D. Domain: Curl.se

Proof of bug bounty program:



I have identified many vulnerabilities in under domain **www.curl.se/**. I figured out it has vulnerabilities that listed by OWASP Top 10. And the overall website risk level is **Critical**. I used net sparker to identify vulnerabilities in the following domain. And used some of the previously mentioned tools in section. I identified the following OWASP Top vulnerabilities in following domain.

1. **Using components with known vulnerabilities.**
2. **Sensitive data exposure.**

netsparker

5/22/2023 6:26:19 PM (UTC+05:30)

OWASP Top Ten 2017 Report

https://curl.se/

Scan Time	: 5/22/2023 4:48:19 PM (UTC+05:30)
Scan Duration	: 00:01:37:44
Total Requests	: 66,387
Average Speed	: 11.3r/s

Risk Level:
CRITICAL

Explanation

This report is generated based on OWASP Top Ten 2017 classification.
There are 17 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.

137 IDENTIFIED	8 CONFIRMED	1 CRITICAL
0 HIGH	117 MEDIUM	11 LOW
	1 BEST PRACTICE	7 INFORMATION

Identified Vulnerabilities

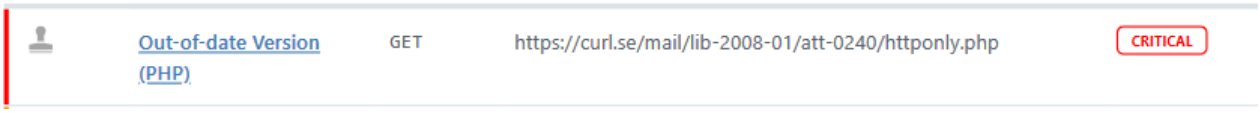
Critical	1
High	0
Medium	117
Low	11
Best Practice	1
Information	7
TOTAL	**137**

Confirmed Vulnerabilities

Critical	0
High	0
Medium	1
Low	5
Best Practice	0
Information	2
TOTAL	**8**

1. **Using components with known vulnerabilities. -Reported**

i. Out of date version PHP (7.4.10).



Under this OWASP category I have identified many vulnerabilities like using components with known vulnerabilities and sensitive data exposure that can occur due to bad implementation security under domain of **https://curl.se/mail/lib-2008-01/att-0240/httponly.php**

Also detected that this web application is using an outdated, also vulnerable PHP version to full fill their back-end processes. Figured out that this version of PHP is vulnerable to following attacks. Also, many features are depreciated in this version of PHP.

- **Impacts of using outdated PHP.**

1. Known weaknesses in outdated PHP versions may exist and be used by attackers. Unauthorized access, remote code execution, SQL injection, cross-site scripting (XSS), and other sorts of attacks could be made possible by these vulnerabilities.

PHP Out-of-bounds Write Vulnerability

In PHP versions 7.3.x below 7.3.29, 7.4.x below 7.4.21 and 8.0.x below 8.0.8, when using Firebird PDO driver extension, a malicious database server could cause crashes in various database functions, such as getAttribute(), execute(), fetch() and others by returning invalid response data that is not parsed correctly by the driver. This can result in crashes, denial of service or potentially memory corruption.

PHP CVE-2022-31629 Vulnerability

In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the vulnerability enables network and same-site attackers to set a standard insecure cookie in the victim's browser which is treated as a `__Host-` or `__Secure-` cookie by PHP applications.

PHP Use After Free Vulnerability

In PHP versions 7.4.x below 7.4.28, 8.0.x below 8.0.16, and 8.1.x below 8.1.3, when using filter functions with FILTER_VALIDATE_FLOAT filter and min/max limits, if the filter fails, there is a possibility to trigger use of allocated memory after free, which can result it crashes, and potentially in overwrite of other memory chunks and RCE. This issue affects: code that uses FILTER_VALIDATE_FLOAT with min/max limits.

PHP Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') Vulnerability

In PHP versions 7.4.x below 7.4.30, 8.0.x below 8.0.20, and 8.1.x below 8.1.7, when pdo_mysql extension with mysqlnd driver, if the third party is allowed to supply host to connect to and the password for the connection, password of excessive length can trigger a buffer overflow in PHP, which can lead to a remote code execution vulnerability.

PHP Release of Invalid Pointer or Reference Vulnerability

In PHP versions 7.4.x below 7.4.30, 8.0.x below 8.0.20, and 8.1.x below 8.1.7, when using Postgres database extension, supplying invalid parameters to the parametrized query may lead to PHP attempting to free memory using uninitialized data as pointers. This could lead to RCE vulnerability or denial of service.

PHP Integer Overflow or Wraparound Vulnerability

The Keccak XKCP SHA-3 reference implementation before fdc6fef has an integer overflow and resultant buffer overflow that allows attackers to execute arbitrary code or eliminate expected cryptographic properties. This occurs in the sponge function interface.

PHP Out-of-bounds Read Vulnerability

In PHP versions prior to 7.4.33, 8.0.25 and 8.2.12, when using imageloadfont() function in gd extension, it is possible to supply a specially crafted font file, such as if the loaded font is used with imagechar() function, the read outside allocated buffer will be used. This can lead to crashes or disclosure of confidential information.

PHP Uncontrolled Recursion Vulnerability

In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the phar uncompressor code would recursively uncompress "quines" gzip files, resulting in an infinite loop.

PHP Out-of-bounds Write Vulnerability

In PHP versions 7.3.x up to and including 7.3.31, 7.4.x below 7.4.25 and 8.0.x below 8.0.12, when running PHP FPM SAPI with main FPM daemon process running as root and child worker processes running as lower-privileged users, it is possible for the child processes to access memory shared with the main process and write to it, modifying it in a way that would cause the root process to conduct invalid memory reads and writes, which can be used to escalate privileges from local unprivileged user to the root user.

PHP Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') Vulnerability

In PHP versions 7.3.x below 7.3.31, 7.4.x below 7.4.24 and 8.0.x below 8.0.11, in Microsoft Windows environment, ZipArchive::extractTo may be tricked into writing a file outside target directory when extracting a ZIP file, thus potentially causing files to be created or overwritten, subject to OS permissions.

PHP Improper Input Validation Vulnerability

In PHP versions 7.3.x below 7.3.29, 7.4.x below 7.4.21 and 8.0.x below 8.0.8, when using URL validation functionality via filter_var() function with FILTER_VALIDATE_URL parameter, an URL with invalid password field can be accepted as valid. This can lead to the code incorrectly parsing the URL and potentially leading to other security implications - like contacting a wrong server or making a wrong access decision.

PHP Other Vulnerability

In PHP versions 7.3.x below 7.3.33, 7.4.x below 7.4.26 and 8.0.x below 8.0.13, certain XML parsing functions, like simplexml_load_file(), URL-decode the filename passed to them. If that filename contains URL-encoded NUL character, this may cause the function to interpret this as the end of the filename, thus interpreting the filename differently from what the user intended, which may lead it to reading a different file than intended.

PHP Improper Input Validation Vulnerability

In PHP versions 7.3.x below 7.3.26, 7.4.x below 7.4.14 and 8.0.0, when validating URL with functions like filter_var($url, FILTER_VALIDATE_URL), PHP will accept an URL with invalid password as valid URL. This may lead to functions that rely on URL being valid to mis-parse the URL and produce wrong data as components of the URL.

PHP NULL Pointer Dereference Vulnerability

In PHP versions 7.3.x below 7.3.27, 7.4.x below 7.4.15 and 8.0.x below 8.0.2, when using SOAP extension to connect to a SOAP server, a malicious SOAP server could return malformed XML data as a response that would cause PHP to access a null pointer and thus cause a crash.

2. The PHP development team no longer provides security updates and patches for out-of-date PHP versions. This implies that any newly found flaws or problems won't be fixed, making your application more vulnerable to assaults.

3. Certain features or functions are designated as deprecated in PHP versions and then finally removed in newer versions. Utilizing an out-of-date PHP version puts your codebase at risk of future support and maintenance issues because it depends on deprecated features.

Because of this vulnerability marked as Critical It's a must to address it with a good solution.

- **Step to reproduce this vulnerability.**

1. Determine the specific outdated PHP version you want to test. Note the exact version number.

2. Make a test script that includes the code patterns that are relevant to the found vulnerabilities. To achieve this, code fragments that cause integer overflow, wraparound, or use-after-free behavior may need to be created.

3. To create inputs that result in integer overflow or wraparound scenarios, use fuzzing frameworks that you have created or tools like the AFL (American Fuzzy Lop). To examine the code and identify flaws, these tools produce test inputs automatically.

4. To find use-after-free vulnerabilities, use programs like AddressSanitizer (ASan) or Valgrind (Memcheck). These tools can be used to detect erroneous memory accesses brought on using released objects.

5. To exploit the vulnerabilities found, run the test script or use the fuzzing tool's generated inputs.

6. Watch how the PHP interpreter behaves while the test is running. Keep an eye out for errors, crashes, or other strange output that can point to a vulnerability.

7. Examine how the vulnerabilities affect the PHP interpreter and the system. Examining crash logs, memory dumps, or error messages may be necessary to gauge the seriousness and consequences of the vulnerabilities.

- **Proof of existence of vulnerability.**

Identified Version

- 7.4.10

Latest Version

- 7.4.33 (in this branch)

Vulnerability Database

- Result is based on 05/16/2023 20:30:00 vulnerability database content.

Certainty

Request

```
GET /mail/lib-2008-01/att-0240/httponly.php HTTP/1.1
Host: curl.se
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: GoogleAdServingTest=Good; GoogleAdServingTest=Good
Referer: https://curl.se/mail/lib-2008-01/attachment.html
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1884.4088 Total Bytes Received : 1444 Body Length : 341 Is Compressed : No

```
HTTP/1.1 200 OK
X-Cache: MISS, MISS
X-Timer: S1684757787.157918,VS0,VE530
Cache-Control: max-age=60
Set-Cookie: volatile_cookie=1
Set-Cookie: nonvolatile_cookie=1; expires=Monday, 09-Nov-09 23:12:40 GMT
Set-Cookie: httponly_volatile_cookie=1;     HttponLy
Set-Cookie: httponly_nonvolatile_cookie=1; expires=Monday, 09-Nov-09 23:12:40 GMT; httPOnly
Strict-Transport-Security: max-age=31536000
X-Powered-By: PHP/7.4.10
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

1. Regular PHP upgrades to the most recent stable release or a supported version that receives security updates and fixes are essential to reducing these risks. This makes sure that your web application takes advantage of the most recent PHP development team features, performance upgrades, and security patches.

2. Applying patches created by the PHP community to address the specific vulnerabilities you've found can be an option if updating to the most recent version of PHP is not possible right away. The known security flaws in the out-of-date version may be fixed by these patches.

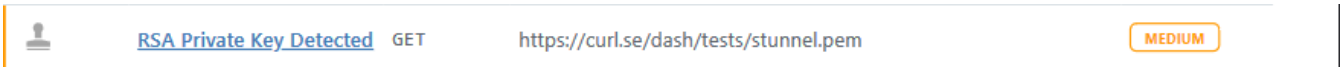
3. You can further defend your application against potential vulnerabilities by adhering to best practices in web application security, such as input validation, safe coding techniques, and frequent security audits.

4. To enforce security settings, review and alter the php.ini configuration file. Disable, for instance, elements of your program that are known to be weak points or superfluous. To assist in locating potential security concerns, enable proper error reporting.

5. To add an additional layer of defense against typical web vulnerabilities, use a web application firewall. WAFs can aid in the detection and mitigation of attacks utilizing known exploits that target PHP vulnerabilities.

2. Sensitive data exposure.

i. RSA private key compromised.



Under this OWASP category I have identified vulnerabilities sensitive data exposure that can occur due to bad implementation security under domain of **https://curl.se/dash/tests/stunnel.pem**

When attempting to connect onto a secure server, the client application uses a digital signature to demonstrate that you own the private key. The server then verifies that the signature is valid and that the public key is approved for your username. If everything is okay, access is allowed to you. If this key is compromised, then there is no point in using the secure mechanism. Anyone can access your secure server with this key.

- **Impacts of using compromised RSA key to server authentication.**

When the private key is not password-protected, anyone who obtains it can access all your accounts.

1. Even if it is password-protected, an attacker with average computational power can test a vast variety of password combinations. Your passphrase can probably be cracked in a matter of seconds if it contains a dictionary word.

2. The attacker may be able to impersonate authorized users, obtain access to user accounts without authorization, or fake digital signatures if the leaked RSA key is used for user authentication or digital signatures. As a result, users may perform unlawful activities on their behalf, such as takeover of accounts, data manipulation, or fraudulent transactions.

3. In SSL/TLS certificates, RSA keys are frequently used to provide secure communication between clients and your web server. An attacker may be able to decrypt and intercept sensitive data sent via HTTPS connections if the RSA key is compromised. As a result, user conversations are no longer secure or private, leaving critical information vulnerable to interceptions or man-in-the-middle assaults.

4. An attacker may edit the application code, introduce malicious code, or disseminate modified copies of your application if the compromised RSA key is utilized for code signing or application integrity checks. This jeopardizes the reliability and integrity of your program and may cause the spread of malware or unauthorized changes.

Because of this vulnerability marked as medium It's a must to address it with a good solution.

- **Step to reproduce this vulnerability.**

1. Find the web application endpoint or feature that is causing the RSA secret key to appear in the response body.

2. To intercept and inspect HTTP requests and replies, set up a web proxy tool like Burp Suite or OWASP ZAP.

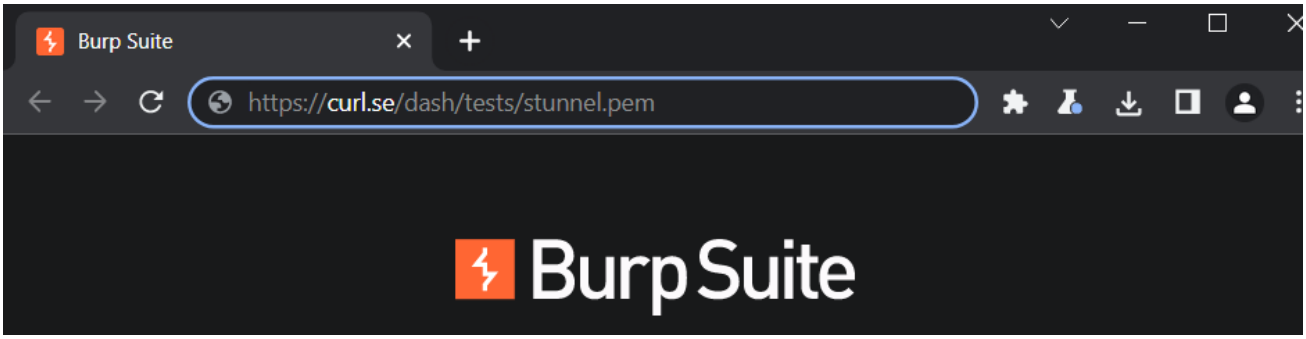
3. Set up your web browser to use the web proxy. The traffic between your browser and the web application can then be recorded and analyzed thanks to this.

4. Use the proxy tool to start recording HTTP traffic, then send a request that results in a response that contains the RSA secret key. This could entail submitting a form, using an API, or carrying out a certain task inside of your program.

5. Examine the response in the proxy tool after the request has been captured. Check the response body for the RSA secret key or any other private information that shouldn't be shared.

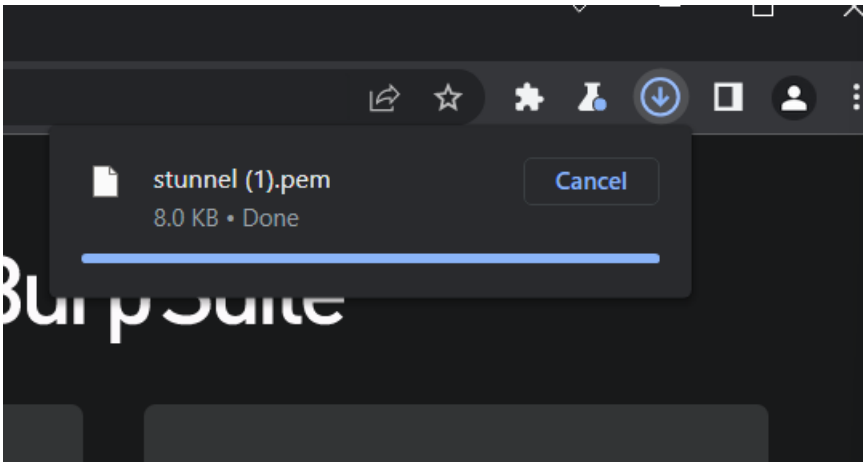
6. To ensure that the RSA secret key is consistently disclosed in the response body, test various scenarios or inputs.

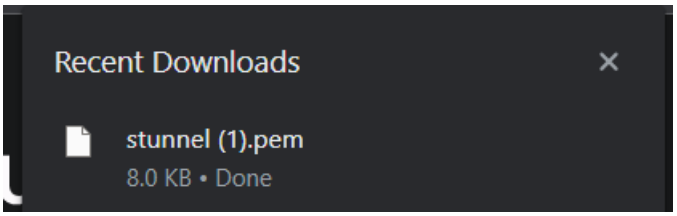
- **Proof of concept.**



Pretty Raw Hex \n

```
GET /dash/tests/stunnel.pem HTTP/2
Host : curl.se
Sec-Ch-Ua : " Not A;Brand";v="99", "Chromium";v="104"
Sec-Ch-Ua-Mobile : ?0
Sec-Ch-Ua-Platform : "Windows"
Upgrade-Insecure-Requests : 1
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.5112.102
Safari/537.36
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.9
Sec-Fetch-Site : none
Sec-Fetch-Mode : navigate
Sec-Fetch-User : ?1
Sec-Fetch-Dest : document
Accept-Encoding : gzip, deflate
Accept-Language : en-US,en;q=0.9
```





```
[something]
# The key
# the certificate
# some dhparam
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCrCrAD0Hb+Xs4V
3mHV45FvfNa7yiaOeL4mNdGmWfHVPFU+CSzsoNSvDjxaorWweFGVYoCAcchOn1lZ
k0ASsqnOss0Xi58n8+PPI3gG0gYjX5sg7EJ3Zq2kXoK0TZRy6hNkcvzLgyzXoYv1
LkzTwYiyyJgZX++Y/GKAs2fMHyP8XzjNgm4tltk1k/4pomllwN9Fqz+sFxgAgEq3
ybq4Xym7xKwWl8xXNBDJNmVsPtiJRcilQoR8Xs0a6PE+VbMhD9A2E/LEL7lzQfqH
qtxE1mSW5FpQ+Uqf4KLnafStWs86IOWnCeLP6BmhAK6ouyICNFyzz7UkTHa/renx
uNOGun2TAgMBAAECggEAH0BsKb5Ax7h90jwYRzL141d9isFkaxq/r46c2FbN24bT
EmstxKycP8ILoAnjxbMuQOvHC/D+RvNRqY7Aocn4Qdakp50wvuWOpc3Ww/RC/9qb
pxfUCyn9Jy/HlPcp3RdM5MknzG2S13Fid7F2gyh0+CmztMs1JZBT1S0ylXbJJfbY
1pdlHcf9oEbYo36vGd9rtJHAFzsFfwua0idl76XYuOnR3bpOkHl1B5cJ8jpOliPv
VTmzn0cIgAmk7IByHHqGQ0u30PFiElI9kEbkKWoxAM1hq1pFU58jQhvp0ZkjVENL
bSFB2B4DbyosxPlbUgvJCN4B7nclqzYqBdrrk6/ZLQKBgQC1lDrPSGIGXLwvkZYS
xc0wtaCC7u6m7zV8rzh5HGcEoVvtmya/VyoZR8KGIpSor8COIkZqFtan6C77C3MH
wClbu2Kn3FkGb76D5U2Xwl38zepzjn8Z5qXc3bZfccrsDY1gXPicgsmcKUY9xV5/
T0RjESDKB+xxkJpCjia6klm2NQKBgQDxJNuqB6frDYKaj7mW/rvyHqkeT94J6eDY
```

- **Proof of existence of vulnerability.**

Request

```
GET /dash/tests/stunnel.pem HTTP/1.1
Host: curl.se
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Referer: https://curl.se/dash/tests/
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.
77 Safari/537.36
X-Scanner: Netsparker
```

Response

Response Time (ms) : 1640.3241 Total Bytes Received : 9016 Body Length : 8150 Is Compressed : No

```
HTTP/1.1 200 OK
X-Cache: MISS, MISS
X-Timer: S1684758746.943941,VS0,VE214
Age: 0
Cache-Control: max-age=1800
ETag: "1fd6-5f0c77b87bbec"
Strict-Transport-Security: max-age=31536000
Server: nginx/1.21.1
X-Content-Type-Options: nosniff
Connection: keep-alive
Expires: Mon, 22 May 2023 13:02:26 GMT
X-Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
X-Cache-Hits: 0, 0
Content-Length: 8150
Via: 1.1 varnish, 1.1 varnish
alt-svc: h3=":443";ma=86400,h3-29=":443";ma=86400,h3-27=":443";ma=86400
Last-Modified: Tue, 27 Dec 2022 04:14:05 GMT
Content-Type: application/x-pem-file
X-Served-By: cache-bma1655-BMA, cache-qpg1246-QPG
Content-Security-Policy: default-src 'self' curl.haxx.se www.curl.se curl.se www.fastly-insights.com fa
stly-insights.com; style-src 'unsafe-inline' 'self' curl.haxx.se www.curl.se curl.se
Date: Mon, 22 May 20
```

```
_value     = Edel Curl Arctic Illudium Research Cloud
commonName                  = "Common Name"
commonName_value            = localhost
```

```
[something]
# The key
# the certificate
# some dhparam
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCrCrAD0Hb+Xs4V
3mHV45FvfNa7yiaOeL4mNdGmWfHVPFU+CSzsoNSvDjxaorWweFGVYoCAcchOn1lZ
k0ASsqnOss0Xi58n8+PPI3gG0gYjX5sg7EJ3Zq2kXoK0TZRy6hNkcvzLgyzXoYv1
LkzTwYiyyJgZX++Y/GKAs2fMHyP8XzjNgm4tltk1k/4pomllwN9Fqz+sFxgAgEq3
ybq4Xym7xKwWl8xXNBDJNmVsPtiJRcilQoR8Xs0a6PE+VbMhD9A2E/LEL7lzQfqH
qtxE1mSW5FpQ+Uqf4KLnafStWs86IOWnCeLP6BmhAK6ouyICNFyzz7UkTHa/renx
uNOGun2TAgMBAAECggEAH0BsKb5Ax7h90jwYRzL141d9isFkaxq/r46c2FbN24bT
EmstxKycP8ILoAnjxbMuQOvHC/D+RvNRqY7Aocn4Qdakp50wvuWOpc3Ww/RC/9qb
pxfUCyn9Jy/HlPcp3RdM5MknzG2S13Fid7F2gyh0+CmztMs1JZBT1S0ylXbJJfbY
1pdlHcf9oEbYo36vGd9rtJHAFzsFfwua0idl76XYuOnR3bpOkHl1B5cJ8jpOliPv
VTmzn0cIgAmk7IByHHqGQ0u30PFiElI9kEbkKWoxAM1hq1pFU58jQhvp0ZkjVENL
bSFB2B4DbyosxPlbUgvJCN4B7nclqzYqBdrrk6/ZLQKBgQC1lDrPSGIGXLwvkZYS
xc0wtaCC7u6m7zV8rzh5HGcEoVvtmya/VyoZR8KGIpSor8COIkZqFtan6C77C3MH
wClbu2Kn3FkGb76D5U2Xwl38zepzjn8Z5qXc3bZfccrsDY1gXPicgsmcKUY9xV5/
```

```
T0RjESDKB+xxkJpCjia6klm2NQKBgQDxJNuqB6frDYKaj7mW/rvyHqkeT94J6e...
Certificate:
Data:
Version: 3 (0x2)
Serial Number: 16717980999635 (0xf3475519fd3)
Signature Algorithm: sha256WithRSAEncryption
Issuer:
countryName
…
```

- **Solutions to the above-mentioned vulnerability (Remedy).**

1. The first and most important step is to delete the RSA private key from the web application's response. Check that no server response, including HTTP response bodies, headers, or any other network transmission, contains the private key.

2. Instead of including the private key in the response from the web application, securely store it on the server or in a separate key management system. To ensure that only authorized individuals can access and handle the private key, implement the necessary access controls.

3. Use secure communication channels, Make that the web application encrypts data while it is being transmitted between the server and the client via a secure communication channel, such as HTTPS/TLS. This provides defense against listening devices and man-in-the-middle attacks.

4. Follow standard practices for key management, such as routinely updating and rotating the private key, limiting access to only authorized users, and implementing safe backup and recovery methods. To safely manage and store the private key, think about utilizing a hardware security module (HSM).

5. Challenges faced & how did I overcome them.

Challenge 1: Identifying Targets and Scope

- Initially, it was challenging to select the right targets and understand the scope of the bug bounty program.
- To overcome this, I extensively researched and studied the program guidelines, scoping documents, and any available information about the target organization.
- I also utilized tools like recon-ng, subdomain enumeration, and Google dorking to identify potential targets within the program's scope.

Challenge 2: Prioritizing Vulnerabilities

- With numerous potential vulnerabilities to explore, it was difficult to prioritize and focus on the most critical ones.
- To overcome this challenge, I adopted a risk-based approach. I analyzed the potential impact and exploitability of each vulnerability, considering factors like data sensitivity, user privacy, and the organization's overall security posture.
- I prioritized vulnerabilities with high impact, easy exploitability, and a larger attack surface.

Challenge 3: Evading Security Controls

- Many organizations implement security controls like WAFs, rate limiting, and IP blocking, which can hinder the discovery of vulnerabilities.
- To overcome this challenge, I employed various techniques such as bypassing WAFs using different encodings or payloads, evading rate limits by rotating IPs or using multi-threading and employing anti-detection techniques to bypass IP blocking.
- Continuous monitoring and analyzing the target's response allowed me to adapt my techniques and avoid detection.

Challenge 4: Dealing with False Positives

- Identifying false positives was a common challenge, as not every reported vulnerability turned out to be genuine.
- To address this, I ensured thorough testing and validation of potential vulnerabilities before reporting them. I followed a systematic methodology, attempting to reproduce the issue multiple times to rule out false positives.
- By gathering sufficient evidence and providing a detailed explanation in my reports, I minimized the chances of false positives and increased the chances of successful vulnerability identification.

Challenge 5: Collaborating with Program Owners

- Communication and collaboration with program owners can sometimes be challenging, especially when clarifying the impact or receiving feedback on reported vulnerabilities.
- To overcome this, I maintained clear and concise communication with the program owners. I provided detailed reports, including proof-of-concepts, screenshots, and steps to reproduce the vulnerabilities.
- If there were any communication barriers or delays, I remained patient and persistent, following up with program owners until we were on the same page.

Challenge 6: Staying Motivated

- Bug bounty programs can be mentally challenging, particularly when encountering long periods without discovering significant vulnerabilities or facing repeated rejections.
- To stay motivated, I set realistic goals and milestones, celebrating smaller successes along the way. I actively engaged in bug bounty communities, seeking advice, and learning from others' experiences.
- I also maintained a growth mindset, recognizing that each attempt was an opportunity to learn and improve my skills, regardless of the outcome.

6. Reflections & takeaways.

1. Real-world Exposure:

- Participating in a bug bounty program provided me with valuable real-world exposure to various technologies, systems, and applications, allowing me to understand their security vulnerabilities in practical scenarios.

2. Hands-on Experience:

- Through the bug bounty program, I gained hands-on experience in identifying, exploiting, and reporting vulnerabilities, strengthening my technical skills and knowledge in cybersecurity.

3. Expanded Knowledge:

- The bug bounty program expanded my knowledge across different domains of cybersecurity, including web application security, network security, mobile security, and more. I gained insights into new attack vectors, vulnerabilities, and mitigation techniques.

4. Improved Analytical Skills:

- Engaging in bug bounty programs enhanced my analytical skills, as I learned to dissect complex systems, trace attack paths, and identify potential security weaknesses. This analytical mindset carries over to other areas of my work.

5. Effective Communication:

- Participating in the bug bounty program improved my communication skills, as i learned to document and explain vulnerabilities clearly and concisely in my reports. This skill is crucial for effective collaboration with program owners and other security professionals.

6. Collaboration and Networking:

- Bug bounty programs fostered collaboration and networking opportunities with fellow security researchers and program owners. Through engagement in bug bounty communities, I have learned from others' experiences, shared knowledge, and built professional connections.

Conclusion.

In conclusion, my bug bounty journey has been an incredibly enriching experience that has allowed me to dive deep into the world of cybersecurity. Through this report, I have shared my findings, challenges, and reflections, showcasing the growth and knowledge I have gained throughout my bug bounty activities [30].

Participating in bug bounty programs has not only honed my technical skills but also fostered critical thinking, problem-solving, and effective communication. I have encountered various vulnerabilities, ranging from common misconfigurations to complex logic flaws, and have successfully reported them to program owners, contributing to the overall security of their systems.

Looking back at my bug bounty journey, I am filled with a sense of pride and satisfaction. The experience has expanded my knowledge, broadened my network within the cybersecurity community, and provided a platform for continuous learning and improvement. I am grateful for the opportunities bug bounty programs have presented and for the individuals who have supported me along the way [31].

As I conclude this bug bounty report, I am excited to continue my journey as a cybersecurity student, further refining my skills, staying updated with emerging threats, and making a positive impact in securing digital systems. Bug bounty programs have not only served as a means of personal growth but have also reinforced my passion for protecting organizations and individuals from malicious actors.

References.

[1] [Online]. Available: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjzperG2YH_AhUCcWwGHY2JAWsQFnoECEoQAQ&url=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FBug_bounty_program&usg=AOvVaw1u9BEoeY0jIXO8lHDYzgh9.

[2] [Online]. Available: https://www.hackerone.com/vulnerability-management/what-are-bug-bounties-how-do-they-work-examples.

[3] [Online]. Available: https://medium.com/@Land2Cyber/14-recon-phases-for-mastering-bug-bounty-hunting-448672522968.

[4] [Online]. Available: https://infosecwriteups.com/bug-bounty-hunting-methodology-toolkit-tips-tricks-blogs-ef6542301c65?gi=f3d2c9de48dc.

[5] [Online]. Available: https://infosecwriteups.com/bug-bounty-hunting-methodology-toolkit-tips-tricks-blogs-ef6542301c65?gi=f3d2c9de48dc.

[6] [Online]. Available: https://infosecwriteups.com/bug-bounty-hunting-methodology-toolkit-tips-tricks-blogs-ef6542301c65?gi=f3d2c9de48dc.

[7] [Online]. Available: https://infosecwriteups.com/bug-bounty-hunting-methodology-toolkit-tips-tricks-blogs-ef6542301c65?gi=f3d2c9de48dc.

[8] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[9] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[10] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[11] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[12] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[13] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[14] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[15] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[16] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[17] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[18] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[19] [Online]. Available: https://www.veracode.com/security/owasp-top-10.

[20] [Online]. Available: https://www.infosectrain.com/blog/top-tools-needed-to-become-a-bug-bounty-hunter/.

[21] [Online]. Available: https://www.infosectrain.com/blog/top-tools-needed-to-become-a-bug-bounty-hunter/.

[22] [Online]. Available: https://www.kali.org/tools/.

[23] [Online]. Available: https://www.kali.org/tools/.

[24] [Online]. Available: https://www.kali.org/tools/.

[25] [Online]. Available: https://www.kali.org/tools/.

[26] [Online]. Available: https://www.kali.org/tools/.

[27] [Online]. Available: https://www.kali.org/tools/.

[28] [Online]. Available: https://www.kali.org/tools/.

[29] [Online]. Available: https://www.kali.org/tools/.

[30] [Online]. Available: https://www.openbugbounty.org/.

[31] [Online]. Available: https://www.openbugbounty.org/.