

BUG BOUNTY ASSIGNMENT



Domain : <https://www.jimdo.com/>
Student Name : P.A. Daham Thameera
Student Registration Number : IT20077624
Date of submission : 18/10/2021
Batch : Group 13.1

Table of Contents

Acknowledgment	2
Purpose	3
Introduction	4
Information Gathering	7
1. Passive information gathering tools	8
➤ Sublist3r	8
➤ Nslookup	10
➤ Whois	15
➤ Whatweb.....	16
➤ Dig.....	22
➤ Netcraft.....	24
➤ Whois Lookup	28
2. Active information gathering tools.....	29
➤ Nmap	29
➤ Dmitry	32
Planning and Analysis.....	36
Vulnerability Detection.....	38
➤ Legion	39
➤ Nikto.....	42
➤ Arachni	45
➤ Uniscan.....	52
➤ Netsparker	55
➤ Owasp ZAP	66
Penetration Testing	70
➤ Penetration testing according to the Arachni scan results	71
➤ Penetration testing according to the Netsparker scan results	73
Conclusion	76
References	77

Acknowledgment

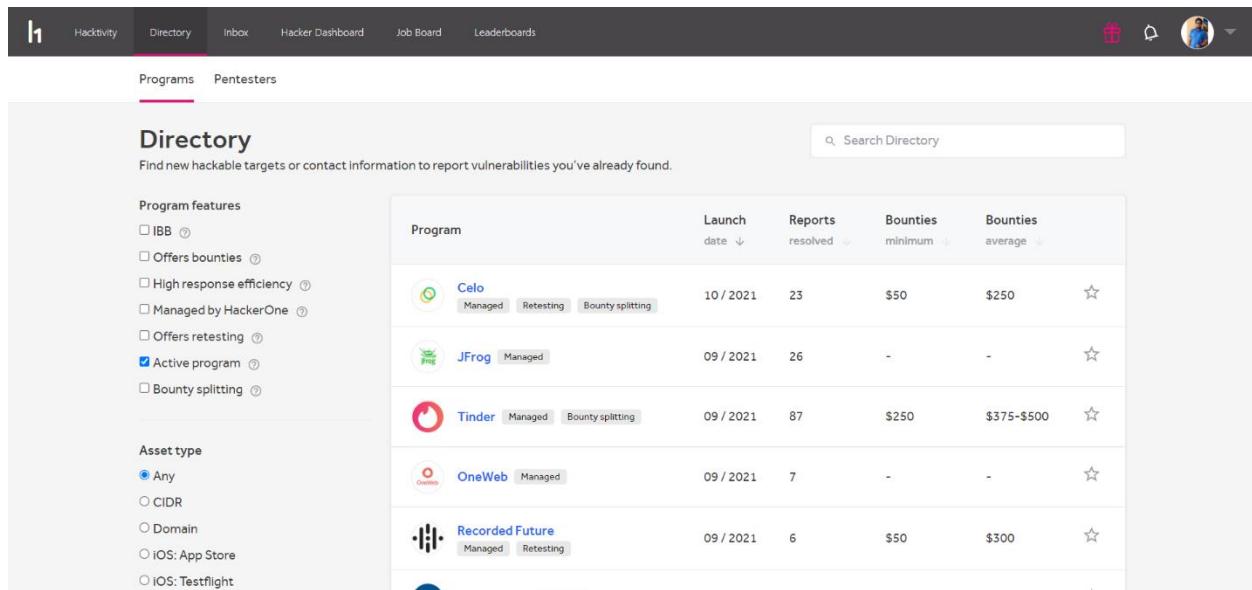
I would like to express my sincere gratitude to Dr. Lakmal Rupasinghe for his relentless effort in guiding us and advising us through difficult and unfamiliar phases of the project and helping us gain practical knowledge and skills in the subject.

It is with heartfelt appreciation that I thank Ms. Chetana Lyanapathirana, Ms. Lanisha Ruggahakotuwa, and Ms. Chathuri Udagedra for helping me throughout the semester to gain new knowledge and understand concepts of Web security and how to implement them in the practical world.

Purpose

The purpose of this assignment is to assess vulnerabilities of the web application. So, <https://www.hackerone.com/> (Fig.1) platform is used to find the websites and web applications for the Bug Bounty hunting. And there are a lot of Bug Bounty hunting platforms to improve our vulnerability assessing skills. As an example, <https://www.bugcrowd.com/> is one of the Bug Bounty hunting platforms. So, the purpose of using this HackerOne platform is because this website legally protects us to do Bug Bounty hunting for real-world web applications.

Using these websites benefits to get powerful knowledge about the penetration testing tool and how to use those tools. And these web audit reports are giving an excellent understanding of how to handle cybersecurity profession skills.



The screenshot shows the HackerOne platform's Directory page. At the top, there is a navigation bar with links for Hækktivity, Directory, inbox, Hacker Dashboard, Job Board, and Leaderboards. On the right side of the top bar are icons for a gift, a bell, and a user profile. Below the navigation bar, there are two tabs: Programs (selected) and Pentesters. The main content area is titled "Directory" and contains a search bar with the placeholder "Search Directory". A sub-header says "Find new hackable targets or contact information to report vulnerabilities you've already found." To the left of the main content, there are two sections: "Program features" and "Asset type". The "Program features" section includes checkboxes for IBB, Offers bounties, High response efficiency, Managed by HackerOne, Offers retesting, Active program (which is checked), and Bounty splitting. The "Asset type" section includes radio buttons for Any, CIDR, Domain, iOS: App Store, and iOS: Testflight, with "Any" selected. The main content area displays a table of programs with the following data:

Program	Launch date	Reports resolved	Bounties minimum	Bounties average	Star icon
Celo	10 / 2021	23	\$50	\$250	☆
JFrog	09 / 2021	26	-	-	☆
Tinder	09 / 2021	87	\$250	\$375-\$500	☆
OneWeb	09 / 2021	7	-	-	☆
Recorded Future	09 / 2021	6	\$50	\$300	☆

Fig. 1. <https://hackerone.com/>

Introduction

Web security is critical to web-based companies and businesses because cybercrime is increasing day by day. Every moment attackers are finding new paths for exploiting the web applications. And attackers develop their skills not only for fun they focus on money also. That is why ransomware attacks are most popular these days. Because of that protection is a must for web applications to defend against this type of cybercrime.

So, a lot of web-based companies and businesses are assigned to Bug Bounty programs to detect the vulnerabilities and fix those vulnerable domains before getting into attack. Hackerone(<https://www.hackerone.com/>) is one of the platforms that help web-based companies to fix vulnerabilities through Bug Bounty programs. And Hackerone platform and web-based companies are paying for penetration testing their web domains. So, I selected a web-based company called Jimdo(<https://www.jimdo.com>) for my Bug Bounty hunting program (Fig. 2). Jimdo platform is used by customers to create their websites (Fig. 3). This includes the ability to add custom JavaScript code to their website. And Jimdo offers a large number of subdomains to this Bug Bounty hunting program and also the little number of reports submitted because they did not pay for those reports and the service.

The screenshot shows the Jimdo GmbH profile on the Hackerone platform. At the top, there's a navigation bar with links for Hacktivity, Directory, Inbox, Hacker Dashboard, Job Board, and Leaderboards. On the right side of the header are icons for a gift, a bell, a user profile, and a dropdown menu. Below the header, the Jimdo logo and name are displayed, along with the URL <https://www.jimdo.com>. A 'Submit report' button is visible. To the right, there's a box for the 'Vulnerability Disclosure Program' which was launched on Nov 2020 and is managed by HackerOne. Below this are 'Bookmark' and 'Subscribe' buttons. In the center, there are two metrics: 'Reports resolved' (21) and 'Assets in scope' (30). At the bottom of the main section, there are links for Policy, Hacktivity, Thanks, and Updates (0). The bottom part of the screenshot shows two cards: 'Policy' and 'Response Efficiency'. The 'Policy' card contains a detailed introduction about maintaining website security and working with the security community. The 'Response Efficiency' card displays average response times: 2 days for average time to first response and 3 days for average time to triage. There is also a note about average time to resolution.

Fig. 2. Jimdo's Hackerone page

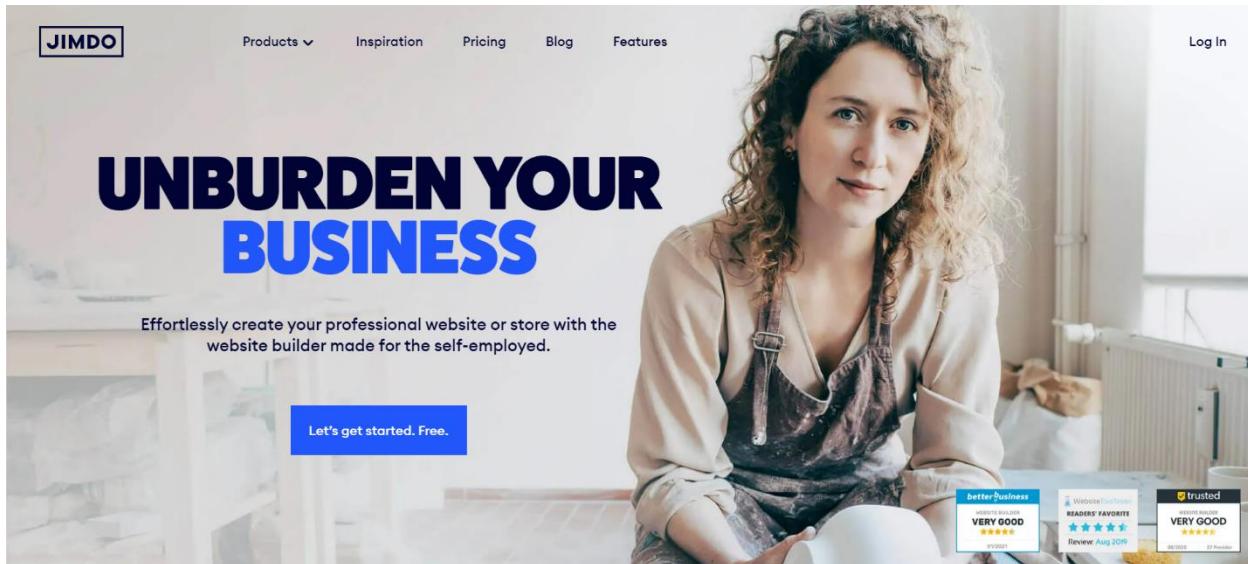


Fig. 3. <https://www.jimdo.com>

This Bug Bounty Assignment is used to be done according to the following web application security testing methodology (Fig. 4).

Web Application Security Testing Methodology



Fig. 4. web application security testing methodology

Before moving into the information gathering stage, we need to consider the top 10 web application's Security Risks and vulnerabilities in 2021. Because we can get an excellent idea for success in our information gathering stage. According to the Sucuri Guides, The Open Web Application Security Project (OWASP) is an online community that creates web application security papers, techniques, documentation, tools, and technologies. The OWASP Top 10 is a list of the top ten most frequent application Security Risks and vulnerabilities [1].

- Injection
- Broken authentication
- Sensitive data exposure
- XML external entities (XXE)
- Broken access control
- Security misconfigurations
- Cross-site scripting (XSS)
- Insecure deserialization
- Using components with known vulnerabilities
- Insufficient logging and monitoring

So, these are the top 10 vulnerabilities found by the OWASP in 2021. We need to focus on these types of vulnerabilities according to the scope and rules provided by Jimdo web-based company and the Hackerone platform.

The scope of the investigation was restricted to 30 domains of Jimdos's primary web application.

- www.jimdo.com
- cms.jimdo.com
- cms.e.jimdo.com
- jimdofree.com
- help.jimdo.com
- jimdo.design
- logo.e.jimdo.com
- account.e.jimdo.com

These are the domains selected to perform this Bug Bounty hunting program.

Information Gathering

Information gathering is the first step to building a strong foundation for this Bug Bounty hunting program. Because this step is about collecting the critical details of the targeted web application. If this step is not done well entire project can be a useless effort. So, more information means that we can capture more vulnerabilities from targeted domains. As an example, we have to find the targeted domain's IP addresses, details about open ports in the targeted domain, and what type of protection they use to protect their web application. According to the All About Testing (AAT), "The more useful information you have about a target, the more you can find vulnerabilities in the target and find more serious problems in the target by exploiting them." [2]. So, perfect information gathering is key to unlocking vulnerabilities from the target and it will help improve our vulnerability scanning process.

Information Gathering can be divided into two parts. They are,

1. Passive information gathering
 - Passive information gathering is collecting information from the targeted domain without invoking any kind of communication with the target systems.
2. Active information gathering
 - Active information gathering is collecting information from the targeted domain involves monitoring the target systems by building communication with the target. This method is detectable to the targeted system.

Considering the Passive and Active information gathering, there are many tools to gather information from the target domain using both methods. They are,

1. Passive information gathering tools
 - sublist3r
 - nslookup
 - whois
 - whatweb
 - dig
 - Netcraft (<https://sitereport.netcraft.com/>)
 - Whois Lookup (<https://whois.domaintools.com/>)

2. Active information gathering tools

- Nmap
- Dmitry

These are the information-gathering tools used to analyze the targeted web domain. And I give priority to Passive information gathering tools. Because Active information gathering is very noisy. But we need active information gathering to analyze information about what is open ports are in our targeted system.

1. Passive information gathering tools

➤ Sublist3r

Sublist3r is a subdomain enumeration tool. That means this is a tool to identify the unique subdomains associated with the target domain. Because of this tool, we can gather more information about subdomains. This tool is not built-in and comes with Kali Linux operating system and first, we need to install this tool in the Kali Linux operating system.

- Download the Sublist3r [3].

```
(daham㉿kali)-[~/Documents/Tools]
└─$ git clone https://github.com/aboul3la/Sublist3r.git

Cloning into 'Sublist3r'...
remote: Enumerating objects: 383, done.
remote: Total 383 (delta 0), reused 0 (delta 0), pack-reused 383
Receiving objects: 100% (383/383), 1.12 MiB | 1.42 MiB/s, done.
Resolving deltas: 100% (212/212), done.
```

Fig. 5. Download the Sublist3r using github link

- Check the downloaded location and go into the Sublist3r directory.

```
(daham㉿kali)-[~/Documents/Tools]
└─$ ls
Sublist3r

(daham㉿kali)-[~/Documents/Tools]
└─$ cd Sublist3r

(daham㉿kali)-[~/Documents/Tools/Sublist3r]
└─$ ls
LICENSE      README.md      setup.py  sublist3r.py
MANIFEST.in   requirements.txt  subbrute
```

Fig. 6. Go into Sublist3r directory

- Install Python3-pip in Kali Linux.

```
(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ sudo apt install python3-pip
```

Fig. 7. Install Python3-pip

- Install Dependencies in the Sublist3r directory.

```
(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ sudo pip install -r requirements.txt
Collecting argparse
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: dnspython in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (2.0.0)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt (line 3)) (2.25.1)
Installing collected packages: argparse
Successfully installed argparse-1.4.0
```

Fig. 8. Install Dependencies

- Install argparse module in the Sublist3r directory.

```
(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ sudo apt-get install python-argparse
```

Fig. 9. Install argparse module

- Check Sublist3r is ready to use and test the tool.

```
(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ ls
LICENSE MANIFEST.in README.md requirements.txt setup.py subbrute sublist3r.py

(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ python3 sublist3r.py -d jimdo.com
```

Fig. 10. Checking the tool

After installing Sublist3r next step is to do scan the main domain to capture subdomains in the targeted system (Jimdo.com).



```
(daham㉿kali)-[~/Documents/Tools/Sublist3r]
$ python3 sublist3r.py -d jimdo.com

[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 540
```

Fig. 11. Sublist3r scan result

After the scan, the Sublist3r tool found 540 unique subdomains related to the main domain (Jimdo.com).

➤ Nslookup

Nslookup is perfect DNS enumeration. That means this is a tool for gathering information about the Domain Name System (DNS) of the targeted system. Nslookup tool help to find out the information related to DNS record names, IP addresses of a target, DNS domain names, and the MX records for the domain or the NS servers of the domain. This tool is already built in the Kali Linux environment. So, I gather the information that

all selected domains to get a better understanding of DNS information related to the web application (Jimdo.com).

- Gather information about the IP address of the hostname.

```
[root@10 ~]# nslookup jimdo.com  
Server: [REDACTED]  
Address: [REDACTED]  
  
Non-authoritative answer:  
Name: jimdo.com  
Address: [REDACTED]  
Name: jimao.com  
Address: [REDACTED]  
Name: jimdo.com  
Address: [REDACTED]  
Name: jimdo.com  
Address: [REDACTED]
```

Fig. 12. IP address of the hostname (Jimdo.com)

- Gather information about the mail exchange (MX) records.

```
[root@10 ~]# nslookup -query=mx jimdo.com  
Server: [REDACTED]  
Address: [REDACTED]  
  
Non-authoritative answer:  
jimdo.com mail exchanger = [REDACTED].googlemail.com.  
jimdo.com mail exchanger = [REDACTED].google.com.  
jimdo.com mail exchanger = [REDACTED].aspmx.l.google.com.  
jimdo.com mail exchanger = [REDACTED].aspmx.l.google.com.  
jimdo.com mail exchanger = [REDACTED].googlemail.com.  
  
Authoritative answers can be found from:  
jimdo.com nameserver = [REDACTED].awsdns-01.com.  
jimdo.com nameserver = [REDACTED].awsdns-16.org.  
jimdo.com nameserver = [REDACTED].awsdns-04.net.  
jimdo.com nameserver = [REDACTED].awsdns-13.co.uk.  
[REDACTED].awsdns-01.com internet address = [REDACTED]  
[REDACTED].awsdns-16.org internet address = [REDACTED]  
awsdns-04.net internet address = [REDACTED]  
awsdns-13.co.uk internet address = [REDACTED]  
[REDACTED].google.com has AAAA address [REDACTED]  
[REDACTED].com has AAAA address [REDACTED]  
[REDACTED].net has AAAA address [REDACTED]  
[REDACTED].awsdns-16.org has AAAA address [REDACTED]
```

Fig. 13. MX records (-query=mx) of the Jimdo.com

- Gather information about the nameserver (NS) records.

```
(root@10)-[~/home/daham/Documents/WSTools]
# nslookup -query=ns jimdo.com
Server: [REDACTED]
Address: [REDACTED]

Non-authoritative answer:
jimdo.com      nameserver = [REDACTED].awsdns-01.com.
jimdo.com      nameserver = [REDACTED].awsdns-13.co.uk.
jimdo.com      nameserver = [REDACTED].awsdns-16.org.
jimdo.com      nameserver = [REDACTED].wsdns-04.net.

Authoritative answers can be found from:
[REDACTED].awsdns-01.com    internet address =
[REDACTED].awsdns-04.net     internet address =
[REDACTED].awsdns-16.org     internet address =
[REDACTED].awsdns-13.co.uk   internet address =
[REDACTED].awsdns-01.com     has AAAA address
[REDACTED].awsdns-04.net     has AAAA address
[REDACTED].awsdns-16.org     has AAAA address
[REDACTED].awsdns-13.co.uk   has AAAA address
```

Fig. 14. NS records (-query=ns) of the Jimdo.com

- Gather information about the “start of authority” (SOA) records. That means we can get details about the domain or region, like the administrator's email address, how long the server should wait between refreshes, and the very last time the domain was modified.

```
(root@10)-[~/home/daham/Documents/WSTools]
# nslookup -query=soa jimdo.com
Server: [REDACTED]
Address: [REDACTED]

Non-authoritative answer:
jimdo.com
  origin = [REDACTED].com
  mail addr = [REDACTED].com
  serial =
  refresh =
  retry =
  expire =
  minimum =

Authoritative answers can be found from:
```

Fig. 15. OSA records (-query=OSA) of the Jimdo.com

- “Any” keyword can use gather all the above information using only one command. So, I use that command to gather information on the in-scope domains.

```
(root@10-[ /home/daham/Documents/WSTools ]
# nslookup -query=any jimdo.com
Server: [REDACTED]
Address: [REDACTED]

Non-authoritative answer:
jimdo.com
    origin = [REDACTED].com
    mail addr = [REDACTED].com
    serial =
    refresh =
    retry =
    expire =
    minimum =
jimdo.com      mail exchanger = [REDACTED].google.com.
jimdo.com      mail exchanger = [REDACTED].google.com.
jimdo.com      mail exchanger = [REDACTED].google.com.
jimdo.com      mail exchanger = [REDACTED].googlemail.com.
jimdo.com      mail exchanger = [REDACTED].oglemail.com.
jimdo.com      nameserver = [REDACTED].awsdns-16.org.
jimdo.com      nameserver = [REDACTED].awsdns-13.co.uk.
jimdo.com      nameserver = [REDACTED].awsdns-01.com.
jimdo.com      nameserver = [REDACTED].awsdns-04.net.
Name:  jimdo.com
Address: [REDACTED]
Name:  jimdo.com
Address: [REDACTED]
Name:  jimdo.com
Address: [REDACTED]
Name:  jimdo.com
Address: [REDACTED]

Authoritative answers can be found from:
[REDACTED].google.com      internet address =
awsdns-01.com      internet address =
.awsdns-04.net      internet address =
awsdns-16.org      internet address =
[REDACTED].google.com has AAAA address
[REDACTED].google.com has AAAA address
```

```
(root@10-[ /home/daham ]
# nslookup -query=any cms.jimdo.com
Server: [REDACTED]
Address: [REDACTED]

Non-authoritative answer:
cms.jimdo.com  canonical name = dolphin-cms-auth-proxy-prod.jimdo-platform.net.

Authoritative answers can be found from:
jimdo.com      nameserver = [REDACTED].awsdns-04.net.
jimdo.com      nameserver = [REDACTED].awsdns-16.org.
jimdo.com      nameserver = [REDACTED].awsdns-01.com.
jimdo.com      nameserver = [REDACTED].awsdns-13.co.uk.
[REDACTED].awsdns-01.com      internet address =
[REDACTED].awsdns-01.com      has AAAA address
.awsdns-04.net      has AAAA address
awsdns-16.org      has AAAA address
awsdns-13.co.uk      internet address
awsdns-13.co.uk      has AAAA address
```

```
(root💀 10)-[~/home/daham]
# nslookup -query=any jimdofree.com
Server: [REDACTED]
Address: [REDACTED]

Non-authoritative answer:
jimdofree.com    nameserver = [REDACTED].awsdns-34.net.
jimdofree.com    nameserver = [REDACTED].awsdns-57.com.
jimdofree.com    nameserver = [REDACTED].awsdns-53.co.uk.
jimdofree.com    nameserver = [REDACTED].awsdns-34.org.

Authoritative answers can be found from:
jimdofree.com    nameserver = [REDACTED].awsdns-34.org.
jimdofree.com    nameserver = [REDACTED].awsdns-57.com.
jimdofree.com    nameserver = [REDACTED].awsdns-53.co.uk.
jimdofree.com    nameserver = [REDACTED].awsdns-34.net.
[REDACTED].awsdns-57.com    internet address =
[REDACTED].awsdns-34.net    internet address =
[REDACTED].awsdns-34.org    internet address =
awsdns-53.co.uk    internet address =
awsdns-57.com    has AAAA address
awsdns-34.net    has AAAA address
awsdns-34.org    has AAAA address
[REDACTED].awsdns-53.co.uk has AAAA address
```

Fig. 16. Gather all information using “-query=any” examples

➤ Whois

Whois command gathers information related to targeted domain unknown and distant hosts, server information, network details, and many more details. This command also has a lot of filtering options and uses that “whois --help” command to grant filtering techniques (Fig. 17.).

```
(root@10)-[~/home/daham]
# whois --help
Usage: whois [OPTION] ... OBJECT ...

-h HOST, --host HOST      connect to server HOST
-p PORT, --port PORT      connect to PORT
-I                         query whois.iana.org and follow its referral
-H                         hide legal disclaimers
--verbose                  explain what is being done
--help                     display this help and exit
--version                  output version information and exit

These flags are supported by whois.ripe.net and some RIPE-like servers:
-l                         find the one level less specific match
-L                         find all levels less specific matches
-m                         find all one level more specific matches
-M                         find all levels of more specific matches
-c                         find the smallest match containing a mnt-irt attribute
-x                         exact match
-b                         return brief IP address ranges with abuse contact
-B                         turn off object filtering (show email addresses)
-G                         turn off grouping of associated objects
-d                         return DNS reverse delegation objects too
-i ATTR[,ATTR] ...          do an inverse look-up for specified ATTRibutes
-T TYPE[,TYPE] ...          only look for objects of TYPE
-K                         only primary keys are returned
-r                         turn off recursive look-ups for contact information
-R                         force to show local copy of the domain object even
                           if it contains referral
-a                         also search all the mirrored databases
-s SOURCE[,SOURCE] ...      search the database mirrored from SOURCE
-g SOURCE:FIRST-LAST        find updates from SOURCE from serial FIRST to LAST
-t TYPE                     request template for object of TYPE
-v TYPE                     request verbose template for object of TYPE
-q [version|sources|types]  query specified server info
```

Fig. 17. Whois --help

I did not want to filter the output because I need a full detailed report for my information gathering process. So, these are the sample output of this command (Fig. 18.).

```
(daham@10)-[~]
$ whois jimdo.com
Domain Name: JIMDO.COM
Registry Domain ID: [REDACTED]
Registrar WHOIS Server: whois.psi-usa.info
Registrar URL: http://www.psi-usa.info
Updated Date: 2021-07-05T15:37:42Z
Creation Date: 2005-12-01T17:26:05Z
Registry Expiry Date: 2021-12-01T17:26:05Z
Registrar: [REDACTED] a Domain Robot
Registrar IANA ID: 151
Registrar Abuse Contact Email: [REDACTED]
Registrar Abuse Contact Phone: [REDACTED]
Domain Status: clientTransferProhibited
Name Server: .AWSDNS-16.ORG
Name Server: .AWSDNS-01.COM
Name Server: .AWSDNS-13.CO.UK
Name Server: AWSDNS-04.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: [REDACTED]

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: jimdo.com
Registry Domain ID: [REDACTED]
Registrar WHOIS Server: whois.psi-usa.info
Registrar URL: [REDACTED].info
Updated Date: 2021-07-05T15:37:45Z
Creation Date: 2005-12-01T17:26:05Z
Registrar Registration Expiration Date: 2022-02-12T07:27:08Z
Registrar: [REDACTED]
Registrar IANA ID: 151
Registrar Abuse Contact Email: [REDACTED]
Registrar Abuse Contact Phone: [REDACTED]
Domain Status: clientTransferProhibited ntnp
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: [REDACTED]
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: [REDACTED]
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: [REDACTED]
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
Registrant Email: https://whoispro.domain-robot.org/whois/jimdo.com
```

Fig. 18. Whois Jimdo.com

➤ Whatweb

According to Kali Linux, “WhatWeb identifies websites. It recognizes web technologies including content management systems (CMS), blogging platforms, statistic/analytics packages, JavaScript libraries, web servers, and embedded devices.” [4]. This tool is very powerful because we can capture a lot of details using this Whatweb tool. Specially, we can gather information about what type of protection mechanism is used that the targeted domain to

protect their web application. But the output information is not sorted well. So, we can use filtering options to gather information in a sorted way.

Fig. 19. whatweb -h

But I did not filter the output because I need informative result about what kind of protection method that the targeted domain use to protect their web application and the filtering is a time-consuming process.

- Gather information related to www.jimdo.com

```
(root㉿kali)-[~/home/daham]
# whatweb www.jimdo.com
http://www.jimdo.com [Country] Country [HTTPServer], IP [IP], RedirectLocation[https://www.jimdo.com/], Strict-Transport-Security[max-age=31536000], UncommonHeaders[retry-after,x-served-by,x-cache-nits,x-timer], Varnish, Via-Proxy[1.1 varnish]
http://www.jimdo.com/ [Country] Country [HTML5, HTTPServer], If [If], MetaGenerator[Meta Generator], Open-Graph-Protocol, Script[application/javascript], Strict-Transport-Security[max-age=31536000], UncommonHeaders[content-security-policy,referrer-policy,x-content-type-options,x-served-by,x-cache-hits], Via-Proxy[1.1 varnish, 1.1 varnish], X-Frame-Options[SAMEORIGIN], X-UA-Compatible[ie=edge], X-XSS-Protection[1; mode=block], nginx[1.15.0]
```

Fig. 20. whatweb www.jimdo.com

- Gather information related to csm.jimdo.com

```
(root㉿kali)-[~/home/daham]
# whatweb cms.e.jimdo.com
http://cms.e.jimdo.com [Cookies[ClickAndChange], Country] Cookies[ClickAndChange], Country [HTTPServer], HttpOnly[ClickAndChange], IP [IP], RedirectLocation[https://account.e.jimdo.com/accounts/login], nginx [No matches found.]
https://account.e.jimdo.com/accounts/login [Country] Country [HTTPServer], IP [IP], RedirectLocation[/en/accounts/login], Strict-Transport-Security[max-age=15768000], UncommonHeaders[referrer-policy,x-content-type-options], X-XSS-Protection[1; mode=block], nginx
https://account.e.jimdo.com/en/accounts/login [OK] Content-Language[en], Cookies[csrftoken], Country [Django, HTML5, HTTPServer], HttpOnly[csrftoken], If [If], PasswordField[password], Script[application/json;text/javascript], Strict-Transport-Security[max-age=15768000], Title[Sign In], UncommonHeaders[content-security-policy,referrer-policy,x-content-type-options], X-Frame-Options[DENY], X-XSS-Protection[1; mode=block], nginx
```

Fig. 21. whatweb cms.jimdo.com

- Gather information related to csm.s.jimdo.com

```
(root㉿kali)-[~/home/daham]
# whatweb cms.e.jimdo.com
http://cms.e.jimdo.com [Cookies[ClickAndChange], Country] Cookies[ClickAndChange], Country [HTTPServer], HttpOnly[ClickAndChange], IP [IP], RedirectLocation[https://account.e.jimdo.com/accounts/login], nginx [No matches found.]
https://account.e.jimdo.com/accounts/login [Country(UNITED STATES)[US]] Country(UNITED STATES)[US] [HTTPServer], IP [IP], RedirectLocation[/en/accounts/login], Strict-Transport-Security[max-age=15768000], UncommonHeaders[referrer-policy,x-content-type-options], X-XSS-Protection[1; mode=block], nginx
https://account.e.jimdo.com/en/accounts/login [Content-language[en], Cookies[csrftoken], Country] Content-language[en], Cookies[csrftoken], Country [Django, HTML5, HTTPServer], HttpOnly[csrftoken], If [If], PasswordField[password], Script[application/json;text/javascript], Strict-Transport-Security[max-age=15768000], Title[Sign In], UncommonHeaders[content-security-policy,referrer-policy,x-content-type-options], X-Frame-Options[DENY], X-XSS-Protection[1; mode=block], nginx
```

Fig. 22. whatweb cms.e.jimdo.com

- Gather information related to jimdofree.com

```
(root㉿kali)-[~/home/daham]
# whatweb jimdofree.com
http://jimdofree.com [Country] Country [HTTPServer], IP [IP], RedirectLocation[https://www.jimdo.com/], UncommonHeaders[retry-after,x-served-by,x-cache-hits,x-timer], Varnish, Via-Proxy[1.1 varnish]
https://www.jimdo.com/ [Country] Country [HTML5, HTTPServer], If [If], MetaGenerator[Meta Generator], Open-Graph-Protocol, Script[application/ld+json,module], Strict-Transport-Security[max-age=31536000], UncommonHeaders[content-security-policy,referrer-policy,x-content-type-options,x-served-by,x-cache-hits], Via-Proxy[1.1 varnish, 1.1 varnish], X-Frame-Options[SAMEORIGIN], X-UA-Compatible[ie=edge], X-XSS-Protection[1; mode=block], nginx[1]
```

Fig. 23. whatweb jimdofree.com

- Gather information related to dash.e.jimdo.com

```
(root㉿kali)-[~/home/daham]
# whatweb dash.e.jimdo.com
ERROR Opening: http://dash.e.jimdo.com - Connection refused - connect(2) for "http://dash.e.jimdo.com" port [port]
```

Fig. 24. whatweb dash.e.jimdo.com

- Gather information related to help.jimdo.com

```
(root㉿kali)-[~/home/daham]
# whatweb help.jimdo.com
http://help.jimdo.com [Cookies[__cfruid], Country] Cookies[__cfruid], Country [HTTPServer], HttpOnly[__cfruid], IP [IP], RedirectLocation[https://help.jimdo.com/hc], UncommonHeaders[content-security-policy,zendesk-ep,x-zendesk-origin-server,x-request-id,x-zendesk-zorg,x-content-type-options,cf-cache-status,report-to,nel,cf-ray], X-XSS-Protection[1; mode=block]
https://help.jimdo.com/hc [Content-Language] Content-Language [Cookies[__cfruid], help_center_session], Country [Country], HTTPServer, If [If], __cfruid, HelpCenterSession, If [If], RedirectLocation[https://help.jimdo.com/hc/de], Strict-Transport-Security[max-age=259200], UncommonHeaders[x-zendesk-origin-server,x-request-id,protocol,x-zendesk-zorg,x-content-type-options,cf-cache-status,expect-ct,report-to,nel,cf-ray], X-Frame-Options[SAMEORIGIN], X-UA-Compatible[IE=edge], X-XSS-Protection[1; mode=block]
https://help.jimdo.com/hc/de [Content-Language[de], Cookies[__cfruid], Country] Content-Language[de], Cookies[__cfruid], Country [HTML5, HTTPServer], HttpOnly[__cfruid], If [If], JQuery, Ruby-on-Rails, Script[text/javascript], Strict-Transport-Security[max-age=259200], Title[Jimdo Creator Hilfe], UncommonHeaders[cf-ray,cf-cache-status,expect-ct,protocol,x-content-type-options,x-request-id,x-zendesk-origin-server,x-zendesk-zorg,report-to,nel], X-Frame-Options[SAMEORIGIN], X-UA-Compatible[IE=edge], X-XSS-Protection[1; mode=block]
```

Fig. 25. Whatweb help.jimdo.com

- Other gathered targeted subdomains.

```

[root@kali -] /home/daham
# whatweb jimdo.design
http://jimdo.design/ Country[██████████, HTTPServer ██████████, IP ██████████, RedirectLocation[https://jimdo.design/], Title[301 Moved Permanently], nginx[1.21.3]
https://jimdo.design/ Country[██████████, Google-Analytics[██████████], HTML5, HTTPServer ██████████, IP ██████████, Script[██████████
t, ██████████

[root@kali -] /home/daham
# whatweb logo.e.jimdo.com
ERROR Opening: http://logo.e.jimdo.com - Connection refused - connect(2) for "██████████" port ██████████

[root@kali -] /home/daham
# whatweb https://logo.e.jimdo.com/?seed=868403&suggestionsPage=1
[1] 5497

[root@kali -] /home/daham
# https://Logo.e.jimdo.com/?seed=868403 Cookies[anon_id], Country[UNITED STATES][US], HTML5, HTTPServer ██████████, IP ██████████, Script[text/plain], Title[Logo Creator - Jimdo], ██████████
[1] + done      whatweb https://Logo.e.jimdo.com/?seed=868403
[root@kali -] /home/daham
# 

```

Fig. 26. Other targeted domains

- Gather information about www.jimdo.com in a sorted way with filtering methods.

- ✓ Scan www.jimdo.com with verbose plugin descriptions (./whatweb -v www.jimdo.com) [4].
- ✓ An aggressive scan of www.jimdo.com detects the exact version of WordPress (./whatweb -a 3 www.jimdo.com) [4].

```

./whatweb -v -a 3 www.jimdo.com
whatWeb report for http://www.jimdo.com
State : 
Title : 
IP : 
Country : 

Summary : Via-Prox . Strict-Transport-Security[max-age=31536000], HTTPServer ██████████, UncommonHeaders[retry-after,x-served-by,x-cache-hits,x-timer], RedirectLocation[https://www.jimdo.com/], Varnish

Detected Plugins:
[ HTTPS-Verify ]
    HTTP server header string. This plugin also attempts to identify the operating system from the server header.
    String ██████████ (from server string)

[ Redirectlocation ]
    HTTP Server string location, used with http-status 301 and 302
    String : https://www.jimdo.com/ (from location)

[ Strict-Transport-Security ]
    Strict-Transport-Security is an HTTP header that restricts a web browser from accessing a website without the security of the HTTPS protocol.
    String : max-age=31536000

[ UncommonHeaders ]
    Uncommon HTTP server headers. The blacklist includes all the standard headers and many non standard but common ones. Interesting but fairly common headers should have their own plugins, eg. x-powered-by, server and x-aspmx-version.
    Info about headers can be found at www.http-stats.com
    String : retry-after,x-served-by,x-cache-hits,x-timer (from headers)

[ Varnish ]
    Varnish is an HTTP accelerator written in C designed for content-heavy dynamic web sites. In contrast to other HTTP accelerators such as Squid, which began life as a client-side cache, or Apache, which is primarily an origin server, Varnish was designed from the ground up as an HTTP accelerator.
    Website : http://www.varnish-cache.org

[ Via-Proxy ]
    This plugin extracts the proxy server details from the Via param of the HTTP header.
    String ██████████

```

```

HTTP Headers:
  HTTP/1.1
  Server: [REDACTED]
  Retry-After: [REDACTED]
  Location: https://www.jimdo.com/
  Content-Length: 6
  Accept-Ranges: bytes
  Date: Sat, 16 Oct 2021 05:07:04 GMT
  Via: 1.1 varnish
  Connection: close
  X-Served-By: cache-qpg1280-QPG
  X-Cache: HIT
  X-Cache-Hits: 0
  X-Cache-Miss: 5153600825.840362
  Strict-Transport-Security: max-age=31536000
  X-Frame-Options: SAMEORIGIN
  X-XSS-Protection: 1; mode=block
  X-UA-Compatible: ie=edge
  Content-Security-Policy: [REDACTED]
  Content-Type: application/json
  Content-Encoding: gzip
  Content-Language: en-US
  Content-Type-Options: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' data: https: wss: android-webview-video-poster; report-uri https://o378271.ingest.sentry.io/api/5281427/security/?sentry_key=b97b30df0244419fe1e6166ef0f19a
  Strict-Transport-Security: max-age=31536000

whatweb report for https://www.jimdo.com/
Status: 200 OK
Title: Jimdo - Create your website for free
IP: [REDACTED]
Country: [REDACTED]

Summary : Via-Proxy [REDACTED] Varnish, [REDACTED] Strict-Transport-Security[max-age=31536000], Open-Graph-Protocol, MetaGenerator[Gatsby 3.16.2], HTTPServer [REDACTED] X-Frame-Options[SAMEORIGIN], X-XSS-Protection[1; mode=block], X-UA-Compatible[ie=edge], [REDACTED] Script[application/ld+json,module], UncommonHeaders[content-security-policy,referrer-policy,x-content-type-options,x-served-by,x-cache-hits], HTML5
Detected Plugins:
[ HTML5 ] HTML version 5, detected by the doctype declaration

[ HTTPServer ] HTTP server header string. This plugin also attempts to identify the operating system from the server header.
String : nginx [from server string]

[ MetaGenerator ] This plugin identifies meta generator tags and extracts its value.
String : Gatsby

[ Open-Graph-Protocol ] The Open Graph protocol enables you to integrate your Web pages into the social graph. It is currently designed for Web pages containing profiles of real-world things - things like movies, sports teams, celebrities, and restaurants. Including Open Graph tags on your Web page, makes your page equivalent to a Facebook Page.

[ Script ] This plugin detects instances of script HTML elements and returns the script language/type.

String : application/ld+json,module

[ Strict-Transport-Security ] Strict-Transport-Security is an HTTP header that restricts a web browser from accessing a website without the security of the HTTPS protocol.

String : max-age=31536000

[ UncommonHeaders ] Uncommon HTTP server headers. The blacklist includes all the standard headers and many non standard but common ones. Interesting but fairly common headers should have their own plugins, eg. x-powered-by, server and x-aspmx-version. Info about headers can be found at www.http-stats.com

String : content-security-policy,referrer-policy,x-content-type-options,x-served-by,x-cache-hits (from headers)

[ Via-Proxy ] This plugin extracts the proxy server details from the via param of the HTTP header.

String : 1.1 varnish, 1.1 varnish

[ X-Frame-Options ] This plugin retrieves the X-Frame-Options value from the HTTP header. - More Info: http://msdn.microsoft.com/en-us/library/cc288472%28VS.85%29.aspx

String : SAMEORIGIN

[ X-UA-Compatible ] This plugin retrieves the X-UA-Compatible value from the HTTP header and meta http-equiv tag. - More Info: http://msdn.microsoft.com/en-us/library/cc817574.aspx

String : ie=edge

[ X-XSS-Protection ] This plugin retrieves the X-XSS-Protection value from the HTTP header. - More Info: http://msdn.microsoft.com/en-us/library/cc288472%28VS.85%29.aspx

String : 1; mode=block

[ nginx ] Nginx (Engine-X) is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server.

Version : [REDACTED]
Website : http://nginx.net/

HTTP Headers:
  HTTP/1.1
  Content-Type: text/html; charset=UTF-8
  Etag: "761509f22-81108"
  Last-Modified: Sat, 16 Oct 2021 05:18:26 GMT
  Referer-Policy: no-referrer-when-downgrade
  Server: nginx/1.15.8
  X-Content-Type-Options: nosniff
  X-Frame-Options: SAMEORIGIN
  X-XSS-Protection: 1; mode=block
  Via: 1.1 varnish, 1.1 varnish
  Accept-Ranges: bytes
  Date: Sat, 16 Oct 2021 05:07:06 GMT
  Age: 79915
  X-Served-By: cache-lcy19274-LCY, cache-qpg1269-QPG
  X-Cache: HIT
  X-Cache-Hits: 1
  Vary: Accept-Encoding
  Strict-Transport-Security: max-age=31536000

```

Fig. 27. whatweb -v -a 3 www.jimdo.com

Consider the information I gathered, these are my ideas related to targeted domains.

- ✓ X-Frame-Options HTTP Header is in the DENY use only two domains (Fig.21 & Fig.22). And those are the login pages of this web application. DENY is about the page must not be embedded into another page within an iframe [5]. So, using a frame to hijack the usernames and passwords using clickjacking attacks is protected.
- ✓ X-Frame-Options HTTP Header is in the SAMEORIGIN used by other web domains including the main domain. SAMEORIGIN is about the website can only be embedded in a site that's paired in terms of scheme, hostname, and port [5].
- ✓ X-XSS-Protection HTTP Header is in the 1; mode=block use by all domains. Using X-Frame-Options HTTP Header to detect the cross-site scripting attack. And using 1; mode=block to enable the filter and completely blocks the page [6].
- ✓ According to fig. 27, this domain uses Nginx to version [REDACTED] as the HTTP server software. This version is older, and it might be vulnerable to exploiting the target domain.
- ✓ Consider about HTTP Strict Transport Security (HSTS) according to the web application status code, it did not use much secure HSTS method to protect their web site. So, 301(Moved Permanently) and 302(Found) methods do vulnerable to attackers. Mainly that the SSL Stripping attacks can happen to this website.

So, the Whatweb tool give me a perfect understanding of targeted domains and the above details are the resent why I am mostly focused on this tool.

➤ Dig

Domain Information Groper (dig) is used for gathering information relevant to Domain Name System (DNS). This command is also the same as the nslookup command. But dig command present the information sorted way than the nslookup command.

```
(daham@10)-[~]
$ dig cms.e.jimdo.com

; <>> DiG 9.16.15-Debian <>> cms.e.jimdo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: [REDACTED]
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cms.e.jimdo.com.           IN      A

;; ANSWER SECTION:
cms.e.jimdo.com.      60      IN      CNAME   cms-prod.jimdo-platform.net.
cms-prod.jimdo-platform.net. 20 IN      CNAME   cms-prod-1462201866-1075462648.eu-west-1.elb.amazonaws.com.
cms-prod-1462201866-1075462648.eu-west-1.elb.amazonaws.com. 60 IN A
cms-prod-1462201866-1075462648.eu-west-1.elb.amazonaws.com. 60 IN A [REDACTED]

;; AUTHORITY SECTION:
eu-west-1.elb.amazonaws.com. 91 IN      NS      [REDACTED].awsdns-28.net.
eu-west-1.elb.amazonaws.com. 91 IN      NS      [REDACTED].awsdns-42.com.
eu-west-1.elb.amazonaws.com. 91 IN      NS      [REDACTED].awsdns-03.org.
eu-west-1.elb.amazonaws.com. 91 IN      NS      [REDACTED].awsdns-60.co.uk.

;; ADDITIONAL SECTION:
.awsdns-42.com.    89589   IN      A       [REDACTED]
.awsdns-42.com.    89589   IN      AAAA    [REDACTED]
.awsdns-28.net.    89589   IN      A       [REDACTED]
.awsdns-28.net.    89589   IN      AAAA    [REDACTED]
.awsdns-03.org.    89589   IN      A       [REDACTED]
.awsdns-03.org.    89589   IN      AAAA    [REDACTED]
.awsdns-60.co.uk.  91391   IN      A       [REDACTED]
.awsdns-60.co.uk.  91391   IN      AAAA    [REDACTED]

;; Query time: 244 msec
;; SERVER: [REDACTED] ( [REDACTED])
;; WHEN: Sat Oct 16 14:14:45 +0530 2021
;; MSG SIZE  rcvd: 496
```

Fig. 28. dig cms.e.jimdo.com

```
(daham@10)@[~]
$ dig help.jimdo.com

; <>> DiG 9.16.15-Debian <>> help.jimdo.com
;; global options: +cmd
;; Got answer:
;; →HEADER← opcode: QUERY, status: NOERROR, id: [REDACTED]
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1024
;; QUESTION SECTION:
;help.jimdo.com.           IN      A

;; ANSWER SECTION:
help.jimdo.com.        300     IN      CNAME   jimdo.zendesk.com.
jimdo.zendesk.com.    300     IN      A        [REDACTED]
jimdo.zendesk.com.    300     IN      A        [REDACTED]

;; AUTHORITY SECTION:
zendesk.com.          1083    IN      NS      [REDACTED].awsdns-24.org.
zendesk.com.          1083    IN      NS      [REDACTED].awsdns-59.co.uk.
zendesk.com.          1083    IN      NS      [REDACTED].awsdns-16.com.
zendesk.com.          1083    IN      NS      [REDACTED].awsdns-24.net.

;; ADDITIONAL SECTION:
.awsdns-16.com.       3234    IN      A       [REDACTED]
.awsdns-24.net.        89566   IN      A       [REDACTED]
.awsdns-24.org.        89534   IN      A       [REDACTED]
.awsdns-59.co.uk.      89545   IN      A       [REDACTED]
.awsdns-16.com.        89923   IN      AAAA    [REDACTED]
.awsdns-24.net.        89566   IN      AAAA    [REDACTED]
.awsdns-24.org.        89534   IN      AAAA    [REDACTED]
.awsdns-59.co.uk.      89545   IN      AAAA    [REDACTED]

;; Query time: 88 msec
;; SERVER: [REDACTED]
;; WHEN: Sat Oct 16 14:15:22 +0530 2021
;; MSG SIZE  rcvd: 416
```

Fig. 29. dig help.jimdo.com

```
(daham@10)-[~]
$ dig logo.e.jimdo.com

; <>> DiG 9.16.15-Debian <>> logo.e.jimdo.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: [REDACTED]
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;logo.e.jimdo.com.      IN      A

;; ANSWER SECTION:
logo.e.jimdo.com.    60      IN      CNAME   logo-prod.jimdo-platform.net.
logo-prod.jimdo-platform.net. 20 IN      CNAME   logo-authproxy-85bd43-1565355445-2033573164.eu-west-1.elb.amazonaws.com.
logo-authproxy-85bd43-1565355445-2033573164.eu-west-1.elb.amazonaws.com. 60 IN A
logo-authproxy-85bd43-1565355445-2033573164.eu-west-1.elb.amazonaws.com. 60 IN A

;; AUTHORITY SECTION:
eu-west-1.elb.amazonaws.com. 126 IN      NS      [REDACTED].awsdns-60.co.uk.
eu-west-1.elb.amazonaws.com. 126 IN      NS      [REDACTED].awsdns-03.org.
eu-west-1.elb.amazonaws.com. 126 IN      NS      [REDACTED].awsdns-42.com.
eu-west-1.elb.amazonaws.com. 126 IN      NS      [REDACTED].awsdns-28.net.

;; ADDITIONAL SECTION:
.awsdns-42.com.  90237  IN      A      [REDACTED]
.awsdns-28.net.  90237  IN      A      [REDACTED]
.awsdns-03.org.  90236  IN      A      [REDACTED]
.awsdns-60.co.uk. 90236  IN      A      [REDACTED]
.awsdns-42.com.  90237  IN      AAAA   [REDACTED]
.awsdns-28.net.  90237  IN      AAAA   [REDACTED]
.awsdns-03.org.  90236  IN      AAAA   [REDACTED]
.awsdns-60.co.uk. 90236  IN      AAAA   [REDACTED]

;; Query time: 232 msec
;; SERVER:
;; WHEN: Sat Oct 16 14:15:49 +0530 2021
;; MSG SIZE rcvd: 511
```

Fig. 30. dig logo.e.jimdo.com

➤ Netcraft

Netcraft (<https://sitereport.netcraft.com/>) is an online web tool used to gather information related to technologies utilized in web application development. This tool is helping to identify out-of-date software modules used to develop the web application. These outdated software modules can be vulnerable to exploitation.

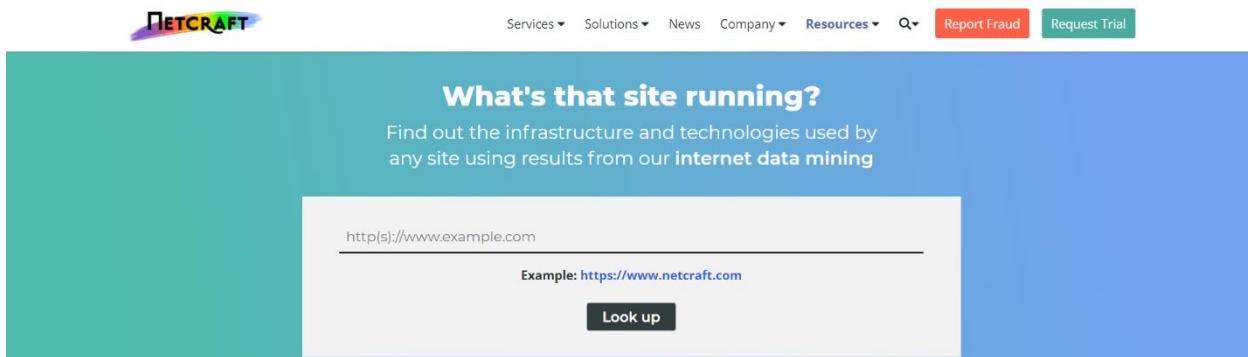


Fig. 31. https://sitereport.netcraft.com/

This is the main interface of the Netcraft tool. We have to enter the domain name to get the details from this tool.

- Gather details about the Network and Background of the targeted domain.

The screenshot shows the Netcraft tool's main interface with the following sections:

Background

Site title	Bring Your Business Online Websites and More - Jimdo	Date first seen	February 2006
Site rank	[REDACTED]	Netcraft Risk Rating	0/10 [REDACTED]
Description	Not Present	Primary language	English

Network

Site	http://www.jimdo.com	Domain	jimdo.com
Netblock Owner	Fastly	Nameserver	ns-15.awsdns-01.com
Hosting company	Fastly	Domain registrar	psi-usa.info
Hosting country	US	Nameserver organisation	whois.markmonitor.com
IPv4 address	[REDACTED]	Organisation	jimdo GmbH [REDACTED]
IPv4 autonomous systems	AS54113	DNS admin	awsdns-hostmaster@amazon.com
IPv6 address	Not Present	Top Level Domain	Commercial entities (.com)
IPv6 autonomous systems	Not Present	DNS Security Extensions	unknown
Reverse DNS	unknown		

Fig. 32. Details about Network and Background

- Gather information regarded to IP Delegation of the targeted domain.

The screenshot shows the Netcraft tool's interface for IP delegation with the following table:

IP delegation

IPv4 address	Country	Name	Description
[REDACTED]	N/A	IANA-BLK	The whole IPv4 address space
[REDACTED]	United States	NET199	American Registry for Internet Numbers
[REDACTED]	United States	SKYCA-3	Fastly
[REDACTED]	United States	SKYCA-3	Fastly

Fig. 33. information regarded to IP Delegation

- Gather the information about Hosting History, Sender Policy Framework, and DMARC of the targeted domain. And we can find the same older HTTP server founded using the whatweb command (Nginx version 1.15.0).

Hosting History

Netblock owner	IP address	OS	Web server	Last seen
Fastly PO Box 78266 San Francisco CA 94120 USA	151.101.62.2	unknown	nginx	15-Oct-2021
Fastly PO Box 78266 San Francisco CA 94120 USA	151.101.62.2	Linux	nginx	21-Jun-2020
Fastly PO Box 78266 San Francisco CA 94120 USA	151.101.62.2	Linux	nginx	8-Dec-2017

Sender Policy Framework

A host's Sender Policy Framework (SPF) describes who can send mail on its behalf. This is done by publishing an SPF record containing a series of rules. Each rule consists of a **qualifier** followed by a specification of which domains to apply this qualifier to. For more information please see [open-spf.org](#).

Warning: It appears that this host does not have an SPF record. There may be an SPF record on jimdo.com: Check the [site report](#).

Setting up an SPF record helps prevent the delivery of forged emails from your domain. Please note that an SPF record will only protect the domain it is added to and not any [mail-enabled subdomains](#). It is recommended to add an SPF record to any subdomain with an MX record.

DMARC

DMARC (Domain-based Message Authentication, Reporting and Conformance) is a mechanism for domain owners to indicate how mail purporting to originate from their domain should be authenticated. It builds on SPF and DKIM, providing a method to set policy and to give reporting of failures. For more information please see [dmarc.org](#).

This host does not have a DMARC record. There may be a DMARC record on the site report for jimdo.com: Check the [site report](#).

Fig. 34. Gather the information about Hosting History, Sender Policy Framework, and DMARC

- Gather the information about Site Technology.

Site Technology (fetched 13 days ago)

HTTP Accelerator

A web accelerator is a proxy server that reduces web site access times.

Technology	Description	Popular sites using this technology
Varnish	An HTTP accelerator for web applications	www.amazon.fr , www.amazon.co.uk , www.gov.uk

Server-Side

Includes all the main technologies that Netcraft detects as running on the server such as PHP.

Technology	Description	Popular sites using this technology
SSL	A cryptographic protocol providing communication security over the Internet	

Client-Side

Includes all the main technologies that run on the browser (such as JavaScript and Adobe Flash).

Technology	Description	Popular sites using this technology
Asynchronous Javascript	No description	www.bloomberg.com , www.roblox.com , www.microsoft.com
JavaScript	Widely-supported programming language commonly used to power client-side dynamic content on websites	

Content Delivery Network

A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers in the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance.

Technology	Description	Popular sites using this technology
Fastly CDN	Content Delivery Network	

RSS Feed

RSS Rich Site Summary is a family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format.

Technology	Description	Popular sites using this technology
RSS	Standardized web feed format used to publish frequently updated works	www.cnblogs.com , www.heise.de , www.tagesschau.de

Character Encoding

A character encoding system consists of a code that pairs each character from a given repertoire with something else such as a bit pattern, sequence of natural numbers, octets, or electrical pulses in order to facilitate the transmission of data (generally numbers or text) through telecommunication networks or for data storage.

Technology	Description	Popular sites using this technology
UTF8	UCS Transformation Format 8 bit	



Services ▾ Solutions ▾ News Company ▾ Resources ▾ Report Fraud Request Trial

HTTP Compression

HTTP compression is a capability that can be built into web servers and web clients to make better use of available bandwidth, and provide greater transmission speeds between both.

Technology	Description	Popular sites using this technology
Gzip Content Encoding	Gzip HTTP Compression protocol	www.sozcu.com.tr , www.newsit.gr , www.varzesh3.com

Web Browser Targeting

Web browser targeting enables software applications to make use of specific functions of the browser as well as optimizing the application for specific browser versions.

Technology	Description	Popular sites using this technology
Document Compatibility Mode	A meta-tag used in Internet Explorer 8 to enable compatibility mode	www.msn.com , www.amazon.com , www.instagram.com
X-Content-Type-Options	Browser MIME type sniffing is disabled	www.linkedin.com , accounts.google.com , mail.yahoo.com
Strict Transport Security	Web security policy mechanism whereby a web server declares that complying user agents are to interact with it using only secure HTTP connections	twitter.com , outlook.office.com , web.whatsapp.com
X-Frame-Options Same Origin	Do not allow this site to be rendered within an iframe	www.google.com , www.baidu.com , www.twitch.tv
Referrer Policy	Restrict referrer information included in subsequent requests	www.canva.com , www.qwant.com , www.bbc.co.uk
Content Security Policy Report	Detect, mitigate and report attacks in the browser	www.startpage.com , teams.microsoft.com , yandex.ru
X-XSS-Protection Block	Block pages on which cross-site scripting is detected	www.paypal.com , docs.microsoft.com , mail.google.com
Content Security Policy	Detect and mitigate attacks in the browser	

Doctype

A Document Type Declaration, or DOCTYPE, is an instruction that associates a particular SGML or XML document (for example, a webpage) with a Document Type Definition (DTD).

Technology	Description	Popular sites using this technology
HTML5	Latest revision of the HTML standard, the main markup language on the web	

HTML 5

HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard.

Technology	Description	Popular sites using this technology
Video Tag	Native browser video playback	www.apple.com , www.pexels.com , www.notion.so
Viewport meta tag	HTML5 tag usually used for mobile optimization	coinmarketcap.com , login.live.com

Fig. 35. Information about Site Technology.

➤ Whois Lookup

Whois Lookup (<https://whois.domaintools.com/>) is an online web tool used to gather information about the hosted company, owner of a target, Server Type, and location of servers.



Fig. 36. <https://whois.domaintools.com/>

- Gather the IP information using the targeted domain IP address.

IP Information for [REDACTED]

Quick Stats	
IP Location	[REDACTED] San Francisco Fastly
ASN	[REDACTED] AS54113 FASTLY, US (registered Oct 04, 2011)
Whois Server	whois.arin.net
IP Address	[REDACTED]
NetRange:	[REDACTED]
CIDR:	[REDACTED]
NetName:	[REDACTED]
NetHandle:	NET-199-232-0-0-1
Parent:	NET199 (NET-199-0-0-0-0)
NetType:	Direct Assignment
OriginAS:	[REDACTED]
Organization:	Fastly (SKYCA-3)
RegDate:	2016-04-14
Updated:	2016-04-14
Ref:	https://rdap.arin.net/registry/ip/199.232.0.0
OrgName:	Fastly
OrgId:	SKYCA-3
Address:	P0 Box 78266
City:	San Francisco
StateProv:	CA
PostalCode:	94107
Country:	US
RegDate:	2011-09-16
Updated:	2021-09-20
Ref:	https://rdap.arin.net/registry/entity/SKYCA-3

```
OrgAbuseHandle: ABUSE4771-ARIN
OrgAbuseName: Abuse Account
OrgAbusePhone: [REDACTED]
OrgAbuseEmail: [REDACTED]
OrgAbuseRef: https://[REDACTED]/registry/entity/ABUSE4771-ARIN

OrgNOCHandle: FNO19-ARIN
OrgNOCName: Fastly Network Operations
OrgNOCPhone: [REDACTED]
OrgNOCEmail: [REDACTED]
OrgNOCRef: https://[REDACTED]/registry/entity/FNO19-ARIN

OrgTechHandle: FRA19-ARIN
OrgTechName: Fastly RIR Administrator
OrgTechPhone: [REDACTED]
OrgTechEmail: [REDACTED]
OrgTechRef: https://[REDACTED]/registry/entity/FRA19-ARIN
```

Fig. 37. Gather the IP information

2. Active information gathering tools

➤ Nmap

Nmap is a tool used to recognize the state of ports, the host is up and running or not, and much other useful information can gather using this tool. Nmap tool also can be used to scan vulnerabilities inside the targeted domain. But now I use this tool only to gather information about the open ports or those ports are filtered, closed, or unfiltered. So, using the Nmap tool to execute SYN scan to gather the details about the open port of the targeted domains.

- Gather open port information about the www.jimdo.com web domain.

```
└$ sudo nmap -sS www.jimdo.com -iL NmapforInputDomain.txt -oN Jimdo.txt
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-14 14:10 +0530
Nmap scan report for www.jimdo.com [REDACTED]
Host is up (0.044s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
[REDACTED]  tcp  open  smtp
[REDACTED]  tcp  open  http
[REDACTED]  tcp  open  https
```

Fig. 38. Open ports of the www.jimdo.com

- Gather open port information about the cms.jimdo.com web domain.

```
Nmap scan report for cms.jimdo.com [REDACTED]
Host is up (0.15s latency).
Other addresses for cms.jimdo.com (not scanned):
rDNS record for [REDACTED] ec2-52-215-158-41.eu-west-1.compute.amazonaws.com
Not shown: 997 closed ports
PORT      STATE SERVICE
/tcp      open  smtp
/tcp      open  http
/tcp      open  https
```

Fig. 39. Open ports of the cms.jimdo.com

- Gather open port information about the cms.e.jimdo.com web domain.

```
Nmap scan report for cms.e.jimdo.com [REDACTED]
Host is up (0.19s latency).
Other addresses for cms.e.jimdo.com (not scanned):
rDNS record for [REDACTED]: ec2-34-251-23-218.eu-west-1.compute.amazonaws.com
Not shown: 997 closed ports
PORT      STATE SERVICE
/tcp      open  smtp
/tcp      open  http
/tcp      open  https
```

Fig. 40. Open ports of the cms.e.jimdo.com

- Gather open port information about the jimdofree.com web domain.

```
Nmap scan report for jimdofree.com [REDACTED]
Host is up (0.0098s latency).
Other addresses for jimdofree.com (not scanned):
Not shown: 998 filtered ports
PORT      STATE SERVICE
[REDACTED]  tcp  open  http
[REDACTED]  tcp  open  https
```

Fig. 41. Open ports of the jimdofree.com

- Gather open port information about the help.jimdo.com web domain. There are some additional open ports.

✓ [REDACTED]/tcp open http-proxy
✓ [REDACTED]/tcp open https-alt

```
Nmap scan report for help.jimdo.com [REDACTED]
Host is up (0.0068s latency).
Other addresses for help.jimdo.com (not scanned):
Not shown: 995 filtered ports
PORT      STATE SERVICE
[REDACTED] /tcp  open  smtp
[REDACTED] /tcp  open  http
[REDACTED] /tcp  open  https
[REDACTED] /tcp  open  http-proxy
[REDACTED] /tcp  open  https-alt
```

Fig. 42. Open ports of the hlpe.jimdo.com

- Gather open port information about the jimdo.design web domain.

```
Nmap scan report for jimdo.design [REDACTED]
Host is up (0.16s latency).
Other addresses for jimdo.design (not scanned)
rDNS record for [REDACTED]: ec2-63-33-112-0.eu-west-1.compute.amazonaws.com
Not shown: 997 closed ports
PORT      STATE SERVICE
[REDACTED] /tcp  open  smtp
[REDACTED] /tcp  open  http
[REDACTED] /tcp  open  https
```

Fig. 43. Open ports of the jimdo.design

- Gather open port information about the logo.e.jimdo.com web domain.

```
Nmap scan report for logo.e.jimdo.com
Host is up (0.16s latency).
Other addresses for logo.e.jimdo.com (not scanned):
rDNS record for [REDACTED] ec2-52-18-108-7.eu-west-1.compute.amazonaws.com
Not shown: 998 closed ports
PORT      STATE SERVICE
/tcp      open  smtp
/tcp      open  https
```

Fig. 44. Open ports of the logo.e.jimdo.com

➤ Dmitry

Dmitry is a collection of information-gathering tools. Because of that, this tool is a combination or package of tools. Using this tool, we can gather details related to Whois lookup web tool information, Netcraft information, and open port details. Because this tool gathers information about open ports, Dmitry is an Active information gathering tool.

```
> Executing "dmitry"
Deepmagic Information Gathering Tool
"There be some deep magic going on"

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

Fig. 45. Dmitry filtering commands

- Gathering Information related to Inet-whois according to Jimdo domain IP address.

```

└$ dmitry jimdo.com
Deepmagic Information Gathering Tool
"There be some deep magic going on"

HostIP: [REDACTED]
HostName:jimdo.com

Gathered Inet-whois information for [REDACTED]

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

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

% This query was served by the RIPE Database Query Service version 1.101 (ANGUS)

```

```
Gathered Inic-whois information for jimdo.com

Domain Name: JIMDO.COM
Registry Domain ID: [REDACTED]
Registrar WHOIS Server: whois.nsi-usa.info
Registrar URL: http://[REDACTED]
Updated Date: 2021-07-05T15:37:42Z
Creation Date: 2005-12-01T17:26:05Z
Registry Expiry Date: 2021-12-01T17:26:05Z
Registrar: PSI-USA, Inc. dba Domain Robot
Registrar IANA ID: 151
Registrar Abuse Contact Email: [REDACTED]
Registrar Abuse Contact Phone: [REDACTED]
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Name Server: .AWSDNS-16.ORG
Name Server: AWSDNS-01.COM
Name Server: .AWSDNS-13.CO.UK
Name Server: AWSDNS-04.NET
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
```

Fig. 46. Inet-whois information

- Gathering Information related to Netcraft according to Jimdo domain.

```
Retrieving Netcraft.com information for jimdo.com
Netcraft.com Information gathered

Gathered Subdomain information for jimdo.com

Searching Google.com:80 ...
HostName:www.jimdo.com
HostIP [REDACTED]
HostName:account.e.jimdo.com
HostIP [REDACTED]
HostName:dash.e.jimdo.com
HostIP [REDACTED]
HostName:cms.jimdo.com
HostIP [REDACTED]
HostName:logo.e.jimdo.com
HostIP [REDACTED]
HostName:nein.jimdo.com
HostIP [REDACTED]
HostName:cms.e.jimdo.com
HostIP [REDACTED]
HostName:webmail.jimdo.com
HostIP [REDACTED]
HostName:vncaccounting.jimdo.com
HostIP [REDACTED]
HostName:register.jimdo.com
HostIP [REDACTED]
HostName:aicarevista.jimdo.com
HostIP [REDACTED]
HostName:x3www.jimdo.com
HostIP [REDACTED]
Searching Altavista.com:80 ...
Found 12 possible subdomain(s) for host jimdo.com, Searched 0 pages containing 0 results
```

Fig. 47. Netcraft information

- Gathering Information related to E-mail and state of TCP port according to Jimdo domain.

```
Gathered E-Mail information for jimdo.com
_____
Searching Google.com:80 ...
Searching Altavista.com:80 ...
Found 0 E-Mail(s) for host jimdo.com, Searched 0 pages containing 0 results

Gathered TCP Port information for [REDACTED]
_____

Port          State
[REDACTED] /tcp      open
[REDACTED] /tcp      open

Portscan Finished: Scanned 150 ports, 0 ports were in state closed

All scans completed, exiting
```

Fig. 48. E-mail and TCP port information

These are the Passive and Active tools I use to gather information about the www.jimdo.com domain.

Planning and Analysis

After the information gathering stage, we need to analyze those details to plan what we focused on next stages. The planning stage is very essential because vulnerability detection is a time-consuming process and with the plan, we can do vulnerability detection in a targeted way. So, we can save our time and vulnerability detection also can be done in a very efficient manner.

So, after the information gathering process that the collected data can be sorted down according to the technical details such as, Web server details, Application server details, and Database server details. And also, that the state of the ports and the HTTP protection methods are the details focused on to execute the vulnerability scan.

➤ Technical Details

○ Web server

- ✓ HTTPS server is Nginx
 - www.jimdo.com (version : 1. [REDACTED])
 - cms.jimdo.com (version not defined)
 - cms.e.jimdo.com (version not defined)
 - jimdofree.com (version : 1. [REDACTED])
 - jimdo.design (version : 1. [REDACTED])
- ✓ HTTP server is Nginx
 - cms.e.jimdo.com (version not defined)
 - jimdo.design (version : 1. [REDACTED])
- ✓ HTTP server is Varnish
 - www.jimdo.com (version not defined)
 - jimdofree.com (version not defined)
- ✓ HTTP/HTTPS server is Cloudflare
 - help.jimdo.com (version not defined)

○ Application server

- ✓ Python
 - cms.jimdo.com
 - account.e.jimdo.com
- ✓ PHP
 - jimdofree.com

- Database server
 - ✓ PostgresSQL
 - cms.jimdo.com
 - ✓ MySQL
 - cms.e.jimdo.com
 - jimdofree.com
 - account.e.jimdo.com
- Open ports details are in the Nmap scan report done in the information gathering stage.
- HTTP security details are in the Wahtweb scan report done in the information gathering stage.

After that select vulnerability scanning tools according to the gathered information and plan the vulnerability scanning according to the information analysis details.

Vulnerability Detection

Vulnerability Detection is a very important stage in Bug Bounty assessment. Because before moving to the penetration testing stage we need to identify vulnerabilities in the particular system. According to Balbix, “Vulnerability scanning is the process of identifying security weaknesses and flaws in the system.” [7].

There are two vulnerability detection methods. They are the automated scanning method and the manual scanning method. I use both of these methods to detect vulnerabilities in the targeted system. Most of the tools can scan vulnerabilities in the system for both of these two methods. Manual scanning is something like a filtered way of scanning and automated scanning is go through all subdomains in the system and scans all vulnerabilities in the system. The automated scanning method is very easy, but it is a time-consuming method. Because that manual scanning is an efficient way of the vulnerability detection method.

So, detecting those vulnerabilities can be done using the Vulnerability Detection tools. There is a lot of open source and paid tools. They are,

- Legion
- Nikto
- Nmap
- Arachni
- Uniscan
- Netsparker
- Nessus
- Owasp Zap

So, I choose that the most suitable vulnerability detection tool according to the gathered information and the usability of those tools. Because some of those tools are not freeware. So, Legion, Nikto, Arachni, Uniscan, Netsparker, Owasp Zap are the tool chosen for use in this Bug Bounty assessment.

➤ Legion

Legion is an open-source network vulnerability detection tool to discover online devices in a network, obtain useful information about targeted systems, and expose targeted system exploits. This tool is a combination of vulnerability detecting tools. Such as Nmap, Whatweb, sslyzer, vulners, SMBenum, and Shodan tools are used in the Legion tool. So, do not need to use Nmap and other tools to detect vulnerabilities in the targeted system.

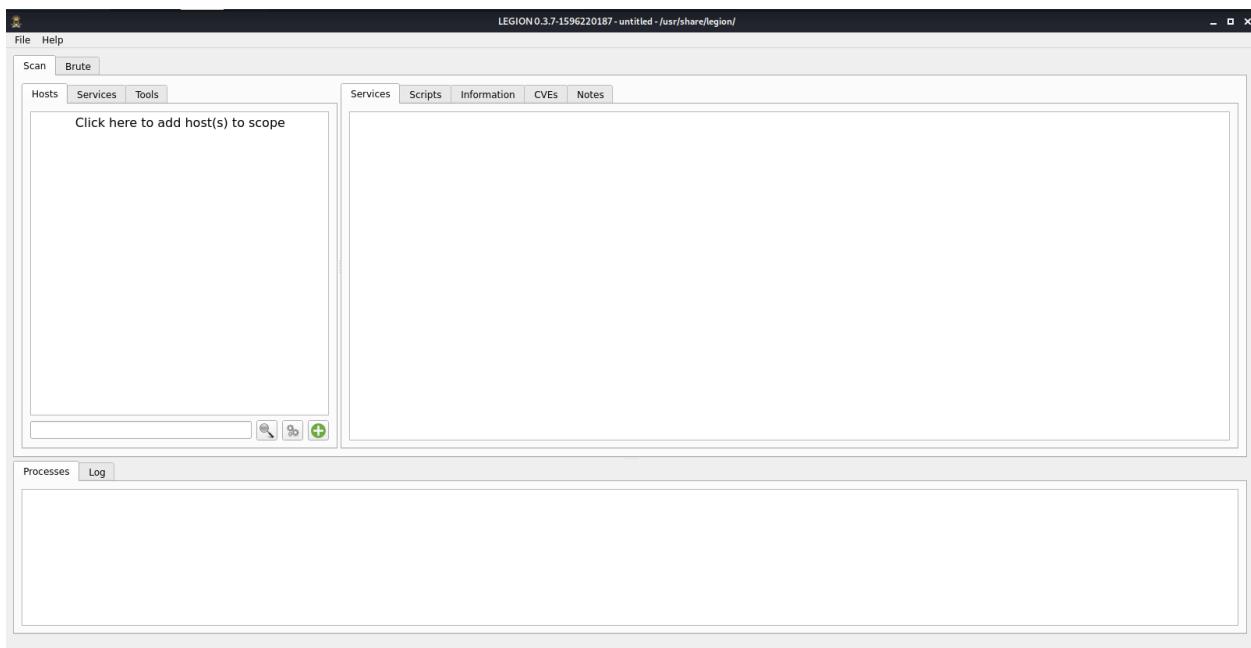


Fig. 49. Dashboard of the Legion Tool

This is the dashboard of the Legion tool. Using the green plus button we can do any type of customization to scan vulnerabilities and provide relevant subdomain links to this tool.

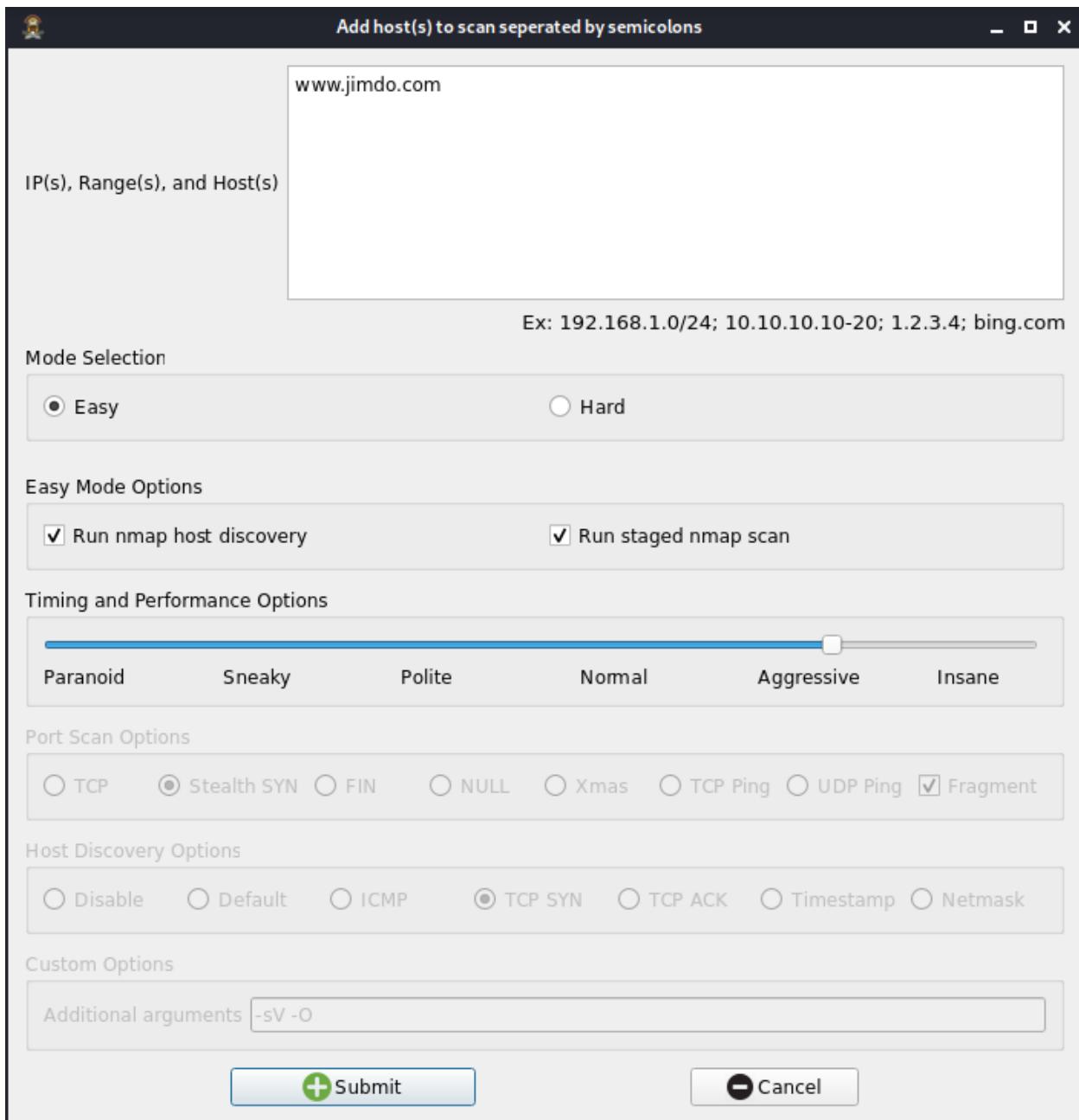


Fig. 50. Customization to scan vulnerabilities

So, I choose automated scan because I need a full scan report of targeted domains and this tool did not take much time to scan. Targeted domain IP address or hostname can use to identify the targeted system and even automated scan this tool provides some Nmap customization methods. After that the customization process is done, we need to submit to get the scanning result from this tool.

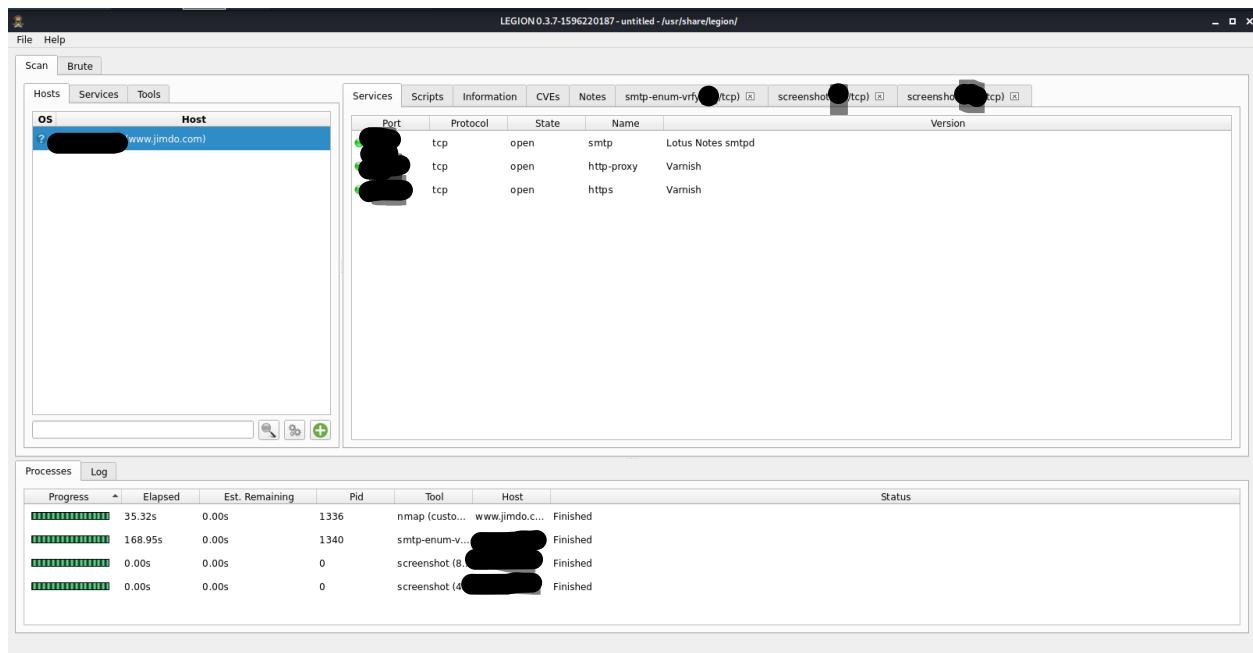


Fig. 51. Port scanning result

Port scanning also gives the same result given through the Nmap scanning done in the information gathering stage. Because that is the same tool used in this scan. [REDACTED] port(HTTP), [REDACTED] port(HTTPS), [REDACTED] port(SMTP) are the open port in the targeted domain. Port 80 can use to exploit vulnerabilities. Because that port is not a protected HTTP port.

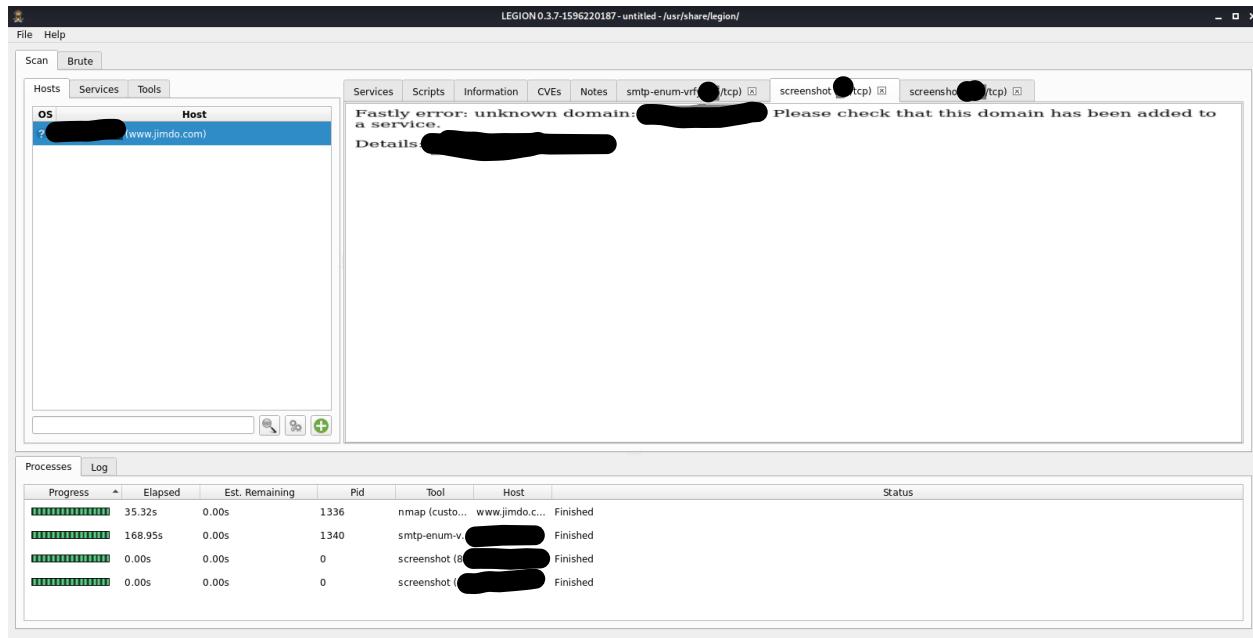


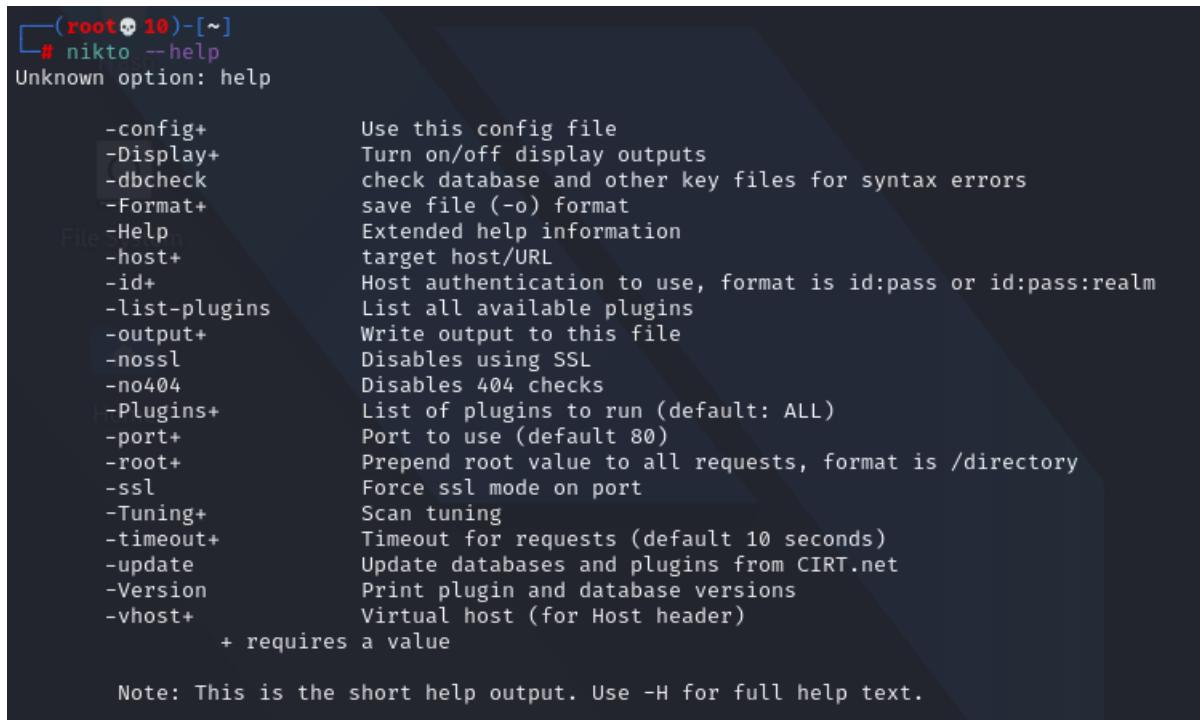
Fig. 52. Screenshot of the port 80

Other domains also give almost identical to the main domain result. And port 80 is an open port that is vulnerable to exploitation.

➤ Nikto

Nikto is a web vulnerability scanner that use to detect vulnerabilities on the targeted domain server. This tool actually detects that the server misconfiguration done by the developers. So, the Nikto tool can find misconfiguring ports in the targeted subdomain and output what type of vulnerabilities have in those subdomains.

- To get a better idea about the Nikto tool we can use the “nikto –help” command.



```
(root💀 10)-[~]
# nikto --help
Unknown option: help

      -config+          Use this config file
      -Display+         Turn on/off display outputs
      -dbcheck          check database and other key files for syntax errors
      -Format+          save file (-o) format
      -Help              Extended help information
      -host+            target host/URL
      -id+              Host authentication to use, format is id:pass or id:pass:realm
      -list-plugins     List all available plugins
      -output+          Write output to this file
      -noSSL            Disables using SSL
      -no404            Disables 404 checks
      -Plugins+         List of plugins to run (default: ALL)
      -port+             Port to use (default 80)
      -root+            Prepend root value to all requests, format is /directory
      -ssl               Force ssl mode on port
      -Tuning+           Scan tuning
      -timeout+          Timeout for requests (default 10 seconds)
      -update            Update databases and plugins from CIRT.net
      -Version           Print plugin and database versions
      -vhost+            Virtual host (for Host header)
                        + requires a value

Note: This is the short help output. Use -H for full help text.
```

Fig. 53. Nikto --help

So, now we need open ports scan details that were collected during the information gathering stage using the Nmap tool to get the scan result of the Nikto tool. According to the Nmap scan results, I scan all open ports use in all the targeted subdomains. So, we can use to input the hostname to the Nikto tool “-h” command and input the port address “-p” command.

- Scan result of the www.jimdo.com using open port 80

```
(root@10:[~]# nikto -h www.jimdo.com -p [REDACTED]
- Nikto v2.1.6

+ Target IP: [REDACTED]
+ Target Hostname: www.jimdo.com
+ Target Port: 80
+ Start Time: 2021-10-24 11:15:00 (GMT5.5)

+ Server: Varnish
+ Retrieved via header: [REDACTED] varnish
+ Retrieved x-served-by header: cache-ppg1271-QPG
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-served-by' found, with contents: cache-ppg1271-QPG
+ Uncommon header 'x-timer' found, with contents: S1635054300.304006,V50,VE0
+ Uncommon header 'x-cache' found, with contents: HIT
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://www.jimdo.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
www.jimdo.com[Sun Oct 24 11:17:38 2021 'Request Hash' = {
    'whisker' => {
        'ssl' => 0,
        'version' => [REDACTED],
        'require_newline_after_headers' => 0,
        'force_bodysnatch' => 0,
        'uri_param_sep' => '?',
        'http_eol' => "\r\n",
        'uri_prefix' => '',
        'method' => 'GET',
        'force_close' => 0,
        'data' => '<!--#include virtual="/index.jsp"-->',
        'force_open' => 0,
        'host' => 'www.jimdo.com',
        'normalize_incoming_headers' => 1,
        'port' => 80,
        'ssl_certfile' => undef,
        'max_size' => 750000,
        'keep-alive' => 1,
        'retry' => 0,
        'ssl_save_info' => 1,
        'uri_postfix' => '',
        'invalid_protocol_return_value' => 1,
        'include_host_in_uri' => 0,
        'timeout' => 10,
        'lowercase_incoming_headers' => 1,
        'MAGIC' => 31339,
        'uri' => '/3rdparty/phpMyAdmin/db_details_importdocs.php?submit_show=true&do=import&docpath=../',
        'protocol' => 'HTTP',
        'http_space2' => ' ',
        'ssl_rsacerfile' => undef,
        'ignore_duplicate_headers' => 0,
        'http_space1' => ' ',
        'trailing_slurp' => 0
    },
    'Content-Length' => 36,
    'Content-Type' => 'application/x-www-form-urlencoded',
    'Host' => 'www.jimdo.com',
    'Connection' => 'Keep-Alive',
    'User-Agent' => 'Mozilla/5.00 (Nikto/2.1.6) (Evasions:None) (Test:000408)'
};
```

Fig. 54. nikto -h www.Jimdo.com -p [REDACTED]

- ✓ X-XSS-Protection is not defined. So, this protection is a must to have, and this website can be vulnerable to the Cross-Site Scripting (XSS) attack.
- ✓ And X-Content-Type-Option header also is not set. According to MDN Web Docs, “The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should be followed and not be changed.” [8] . This also could have the risk of a Cross-Site Scripting (XSS) attack.

- Scan result of the cms.jimdo.com using open port 80. This Jimdo website login web page and the result of the scan are identical to the main domain results. And also, some vulnerabilities can find in all the selected subdomains.

```
[root@IP ~]# nikto -h cms.jimdo.com -o
- Nikto v2.1.6

Target IP: [REDACTED]
Target Hostname: [REDACTED]
Target Port: 80
Message: Multiple IP addresses found
Start Time: 2021-10-24 11:53:48 (GMT5.5)

Server: No banner retrieved
The anti-clickjacking X-Frame-Options header is not present.
The x-XSS-Protection header is not present. This header can hint to the user agent to protect against some forms of XSS
The Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
Cookie SameSite was present without the httpOnly flag
Root page / redirected to https://[REDACTED].e.jimdo.com/openid/authorize?client_id=mc-auth-proxy&redirect_uri=https%3A%2F%2Fcms.jimdo.com%2Foidc%2Fcallback&response_type=code&scope=openid+email+profile+social_profiles&state=eYjyZWpcmVjdC1QWlVlI9
No CGI Directories found (use -C dir to force check all possible dirs)
Server header was changed from 'nginx' which may suggest a man-in-the-middle, load balancer or proxy is in place
OpenID Connect endpoint allowed origin header
```

Fig. 55. nikto -h cms.jimdo.com -p

- Scan result of the cms.jimdo.com using open port 443. Extra protection is there because of the HTTPS use here. But same vulnerabilities can find also in this port.

```
[+] nikto -h cms.jimdo.com
[Nikto v2.1.6]

+ Target IP: 192.168.1.119
+ Target Hostname: cms.jimdo.com
+ Target Port: 80

+ SSL Info:   Subject: /CN=*.jimdo.com
              Ciphers: ECDHE-RSA-AES128-GCM-SHA256
              Issuer: /C=US/O=DigiCert Inc/CN=DigiCert Global TLS DV RSA Mixed SHA256 2020 CA-1
+ Message:    Multiple IP addresses found
+ Start Time: 2021-10-24 22:08:03 (GMT+5)

Server: No banner retrieved
The x-priority header is present
The x-ms-rfc2822-from header is not defined. This header can hint to the user agent to protect against some forms of XSS
The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
The site uses SSL and Expect-CT header is not present.
The x-content-type-options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
The page / redirectes to https://account.jimdo.com/openid/authorize?client_id=cms-auth-proxy&redirect_uri=https%3A%2F%2Fcms.jimdo.com%2Foidc_callback&response_type=code&scope=openid+email+profile+social_profiles&state=eyJzWRpvcWjdC10WVl1l19
No CGI Directories found (use -C <dir> to force check all possible dirs)
The server is using a wildcard certificate (*.jimdo.com)
Server banner has changed from '' to 'nginx' which may suggest a WAF, load balancer or proxy is in place
```

Fig. 56. nikto -h cms.jimdo.com -p [REDACTED]

- Using below commands can use to get the scanning report to a text file.

```
[root@10 ~]# /home/daham/Documents/nikto -h www.jimdo.com -p 80 -o nikto_jimdo_main -F txt
- Nikto v2.1.6
+ Target IP: [REDACTED]
+ Target Hostname: www.jimdo.com
+ Target Port: 80
+ Start Time: 2021-10-24 13:05:46 (GMT5.5)

+ Server: Varnish
+ Retrieved via header: vary value == 1
+ Retrieved x-served-by header: cache-ppg1270-QPG
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ Uncommon header 'x-timer' found, with contents: S1635060946.742360,V$0,VE0
+ Uncommon header 'x-cache' found, with contents: HIT
+ Uncommon header 'x-served-by' found, with contents: cache-ppg1270-QPG
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Root page / redirects to: https://www.jimdo.com/
```

Fig. 57. nikto -h www.jimdo.com -p 80 -o nikto_jimdo_main -F txt

➤ Arachni

Arachni tool is an open-source web application vulnerability detection tool, and it is a high-performance scanner as well as this is fully automated scanning tool. And the disadvantage is that this tool cannot do any customized way of scanning. Because of that scanning is a very time-consuming process.

Fuzzing, taint-analysis, differential analysis, and timing/delay attacks are the combination of techniques that the Arachni tool use to scan the vulnerabilities in the targeted system.

First, that the Arachni tool needs to install in our operating system. So, we can simply extract the files into our operating system.

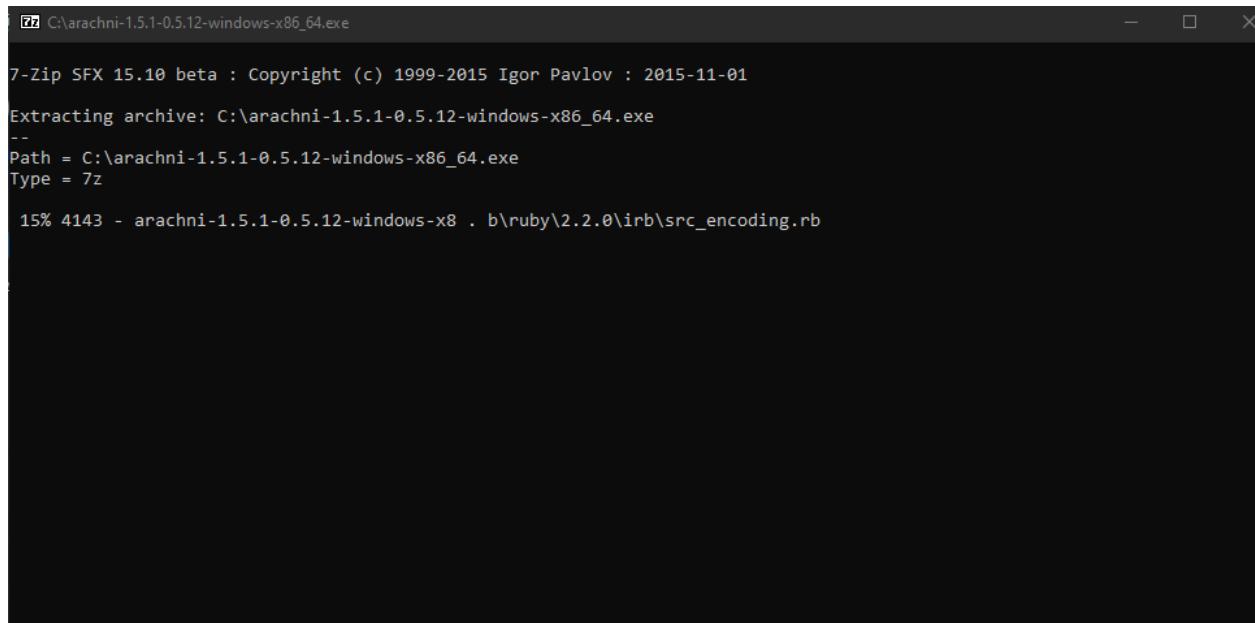
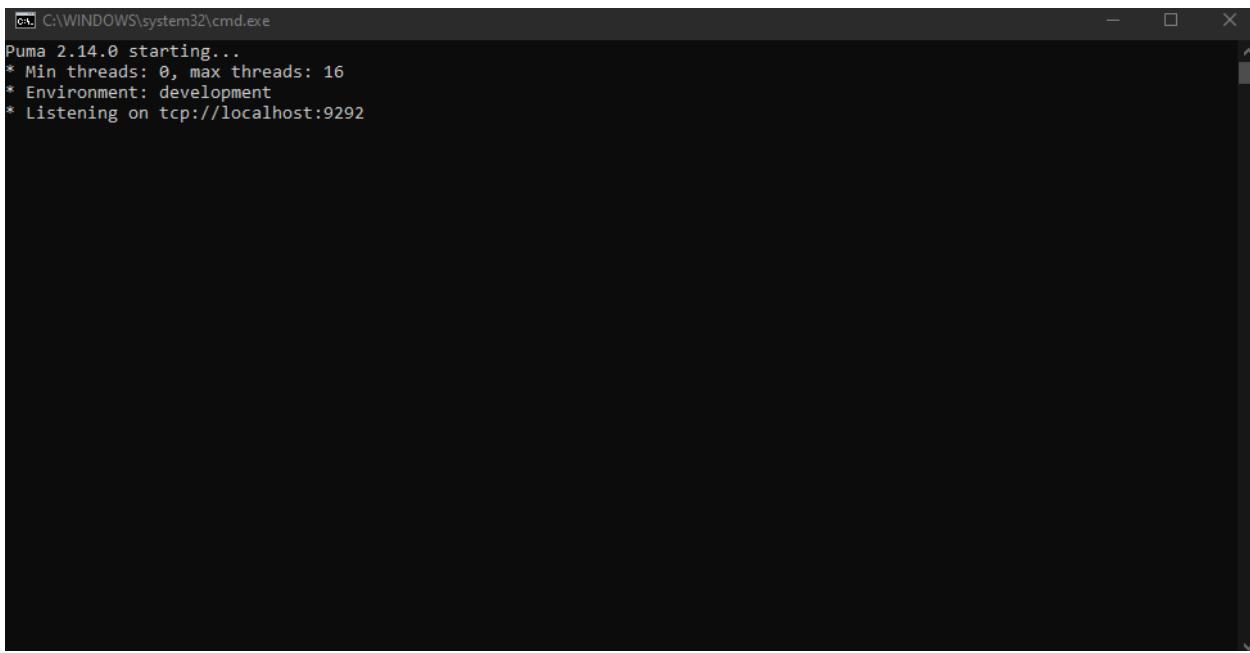


Fig. 58. Extract the files

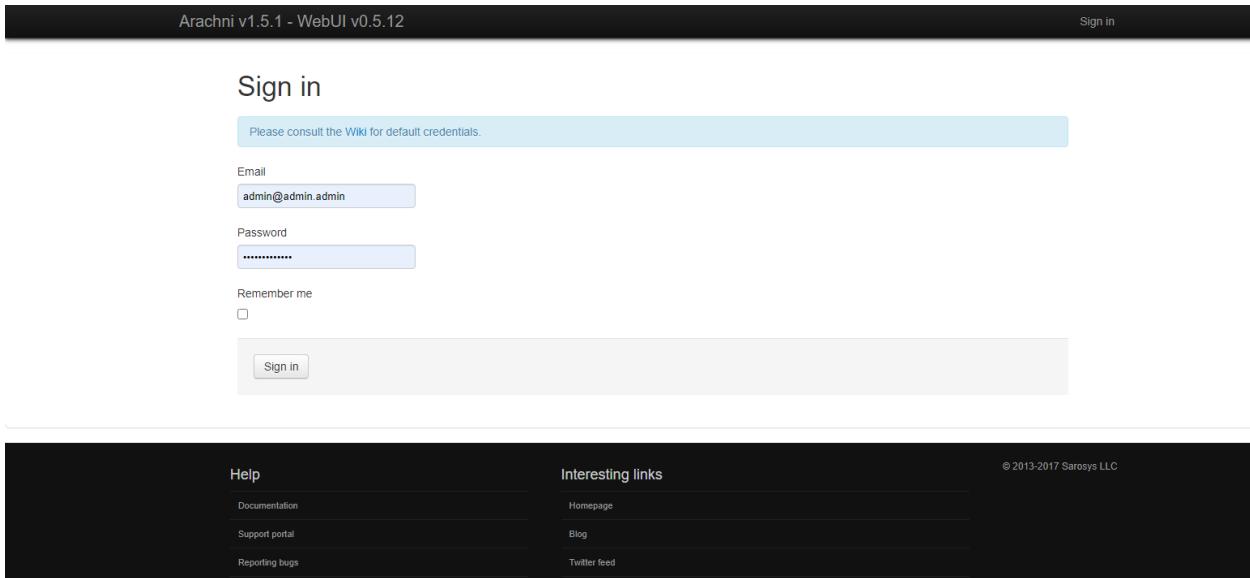
And we can run this tool by going to the extract as well as move into the bin folder and after that, we need to open “arachni_web.bat” to access the web application link(//localhost:9292) of this tool and run this tool.



```
C:\WINDOWS\system32\cmd.exe
Puma 2.14.0 starting...
* Min threads: 0, max threads: 16
* Environment: development
* Listening on tcp://localhost:9292
```

Fig. 59. Access the web application link(//localhost:9292)

After that log in to the Arachni tool by using the provided email(admin@admin.admin) and password(administrator).



The screenshot shows the Arachni v1.5.1 - WebUI v0.5.12 login interface. At the top, there's a dark header bar with the text "Arachni v1.5.1 - WebUI v0.5.12" on the left and "Sign in" on the right. Below the header is a light blue banner with the text "Please consult the [Wiki](#) for default credentials.". The main form has fields for "Email" (containing "admin@admin.admin") and "Password" (containing "*****"). There's a "Remember me" checkbox followed by a small square input field. A "Sign in" button is at the bottom of the form. At the very bottom of the page, there's a footer bar with links to "Help" (Documentation, Support portal, Reporting bugs), "Interesting links" (Homepage, Blog, Twitter feed), and the copyright notice "© 2013-2017 Sarosys LLC".

Fig. 60. Login to the Arachni tool

This is the dashboard of the Arachni tool, and they provide details about this tool.

Arachni v1.5.1 - WebUI v0.5.12 Scans ▾ Profiles ▾ Dispatchers ▾ Users ▾ **Administrator**

Welcome to Arachni's web interface,
please skim through this page to get the information you need before you start wandering about.

Choosing the right database

By default, this interface uses an SQLite3 database and that allows for a configuration-free out-of-the-box experience. However, this setup is unsuitable for larger workloads and may cause stability issues or crashes.

If you plan on only having a few scans running at any given time then there shouldn't be any issue, if, however, you plan on running and managing multiple active scans and Dispatchers then it is recommended that you use PostgreSQL.

Unfortunately, you can't move your data between databases; thus, if you plan on evaluating Arachni using the default database, before moving to PostgreSQL, please refrain from performing operations whose data you're not willing to lose.

Getting help

Don't hesitate to ask for assistance if you run into issues or have any questions, but do use the right way to get in touch:

- Documentation**
Cut and dry documentation, goes through all the available features and explains what everything does.
- Reporting bugs**
If something exploded or you're seeing a scary error please do take some time to tell us about it so that we can fix it and have you back on your way.
- Asking for help**
The support portal allows you to easily ask for assistance or clarifications. If you are unsure about how to do something, this is the right way to find out.

You can quickly access the above resources from any page of the interface via the links in the footer.

Fig. 61. Dashboard of Arachni tool

Using that the scans option in the status bar and start a new scan. We can provide the relevant domain URL to start the scan.

Arachni v1.5.1 - WebUI v0.5.12 Scans ▾ Profiles ▾ Dispatchers ▾ Users ▾ **Administrator**

Start a scan

The only thing you need to do is provide some basic information and make a simple choice about the type of scan you want to perform.

https://www.jimdo.com/ Default (Global)

Full URL of the targeted web application (must include the appropriate protocol, http or https). Configuration profile to use.

Description: Share with: Regular User

You can use Markdown for text formatting.

Advanced options

Distribution **Scheduling**

Instance count: 1

How many Instances to utilise for the scan.

Multi-Instance scans can achieve high efficiency levels which will result in significantly diminished scan times and better utilization of multi-core/multi-CPU systems.

Multi-Instance scans cannot be suspended.

Fig. 62. Scan Dashboard

- Scan the result of the www.jimdo.com domain.

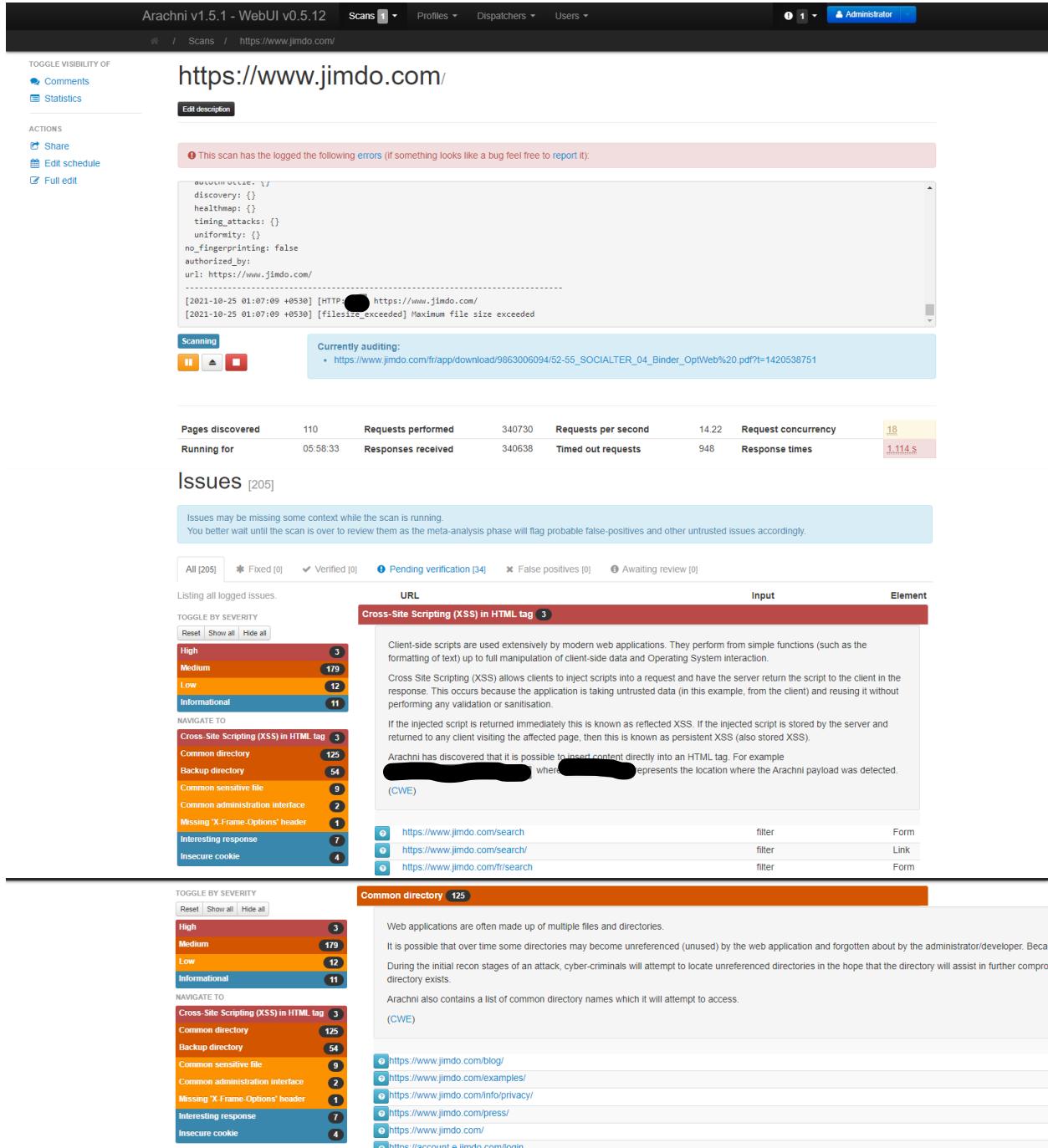


Fig. 63. Scan result of the www.jimdo.com

- ✓ As mentioned before in the Nikto scan result there are High stage vulnerabilities related to Cross-Site Scripting (XSS) attacks. According to the description targeted domain HTML injection type of Cross-Site Scripting (XSS) attack.

So, these are the subdomains we can use to exploit this vulnerability,

- https://www.jimdo[REDACTED]
- https://www.jimdo[REDACTED]1/
- https://www.jimdo[REDACTED]

- Scan result of the cms.jimdo.com domain.

The screenshot shows the Arachni WebUI interface. At the top, it displays "Arachni v1.5.1 - WebUI v0.5.12" and the URL "https://account.e.jimdo.com/". The main content area shows the scan results for the URL "https://account.e.jimdo.com/". A message box states: "This scan has logged the following errors (if something looks like a bug feel free to report it):". Below this, a code block shows the log output:

```
autoDiscover: '/',
discover: {},
healthmap: {},
timing_attacks: {},
uniformity: {},
no_fingerprinting: false,
authorized_by: '',
url: 'https://account.e.jimdo.com/'-----[2021-10-24 17:07:30 +0530] [HTTP: 200] https://account.e.jimdo.com/_next/static/chunks/392-073474e0244d97d3949b.js
[2021-10-24 17:07:30 +0530] [filesize_exceeded] Maximum file size exceeded
```

A note below the log says: "The scan was aborted after 00:47:02."

Below the log, there's a section titled "Issues [32]" with a sub-section "Missing 'Strict-Transport-Security' header". It includes a table with columns: URL, Input, and Element. The URL is "Missing 'Strict-Transport-Security' header". The input is "The HTTP protocol by itself is clear text, meaning that any data that is transmitted via HTTP can be captured and the contents viewed. To keep data private and prevent it from being intercepted, HTTP is often tunneled through either Secure Sockets Layer (SSL) or Transport Layer Security (TLS). When either of these encryption standards are used, it is referred to as HTTPS. HTTP Strict Transport Security (HSTS) is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. This will be enforced by the browser even if the user requests a HTTP resource on the same server." The element is "(CWE)".

There are also other issue sections visible, such as "Common directory" which notes that web applications are often made up of multiple files and directories, and "Allowed HTTP methods".

Fig. 64. Scan result of the cms.jimdo.com

- ✓ As previously mentioned in the Whatweb scan when happening in the information gathering stage. According to the status code, there is no Strict-Transport-Security in those domains, and this can be vulnerable to SSL Stripping attack. So, this is the login subdomain we can use to exploit this vulnerability,
- <https://account.e.jimdo.com/en/signup>

- Scan result of the jimdo.design domain.

The screenshot shows the Arachni WebUI interface for a scan of the <https://jimdo.design> domain. The main content area displays a red box containing an error message: "This scan has the logged the following errors (if something looks like a bug feel free to report it):". Below this is a code snippet showing log entries from 2021-10-25 at 00:27:42 +0530, including one for a maximum file size exceeded issue. A green bar at the bottom indicates the scan completed in 00:19:36.

Severity	Issue Description	Input	Element
Medium	Missing 'Strict-Transport-Security' header	https://jimdo.design/	Server
Low	Missing 'X-Frame-Options' header		
Informational	Common sensitive file		
Informational	Interesting response		
Informational	Insecure cookie		

Issues [5]

All [5] * Fixed [0] ✓ Verified [0] ⚡ Pending verification [0] ✘ False positives [0] ⓘ Awaiting review [0]

Listing all logged issues.

TOGGLE BY SEVERITY

Reset Show all Hide all

Medium 1

Low 2

Informational 2

NAVIGATE TO

Missing 'Strict-Transport-Security' header 1

Missing 'X-Frame-Options' header 1

Common sensitive file 1

Interesting response 1

Insecure cookie 1

Missing 'Strict-Transport-Security' header 1

The HTTP protocol by itself is clear text, meaning that any data that is transmitted via HTTP can be captured and the contents viewed. To keep data private and prevent it from being intercepted, HTTP is often tunneled through either Secure Sockets Layer (SSL) or Transport Layer Security (TLS). When either of these encryption standards are used, it is referred to as HTTPS. HTTP Strict Transport Security (HSTS) is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. This will be enforced by the browser even if the user requests a HTTP resource on the same server.

Cyber-criminals will often attempt to compromise sensitive information passed from the client to the server using HTTP. This can be conducted via various Man-in-The-Middle (MitM) attacks or through network packet captures.

Arachni discovered that the affected application is using HTTPS however does not use the HSTS header.

(CWE)

Missing 'X-Frame-Options' header 1

Clickjacking (User Interface Redress Attack, UI Redress Attack, UI Redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

The server didn't return an `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

The `X-Frame-Options` HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks by ensuring that their content is not embedded.

Fig. 65. Scan result of the jimdo.design

- ✓ This domain and other selected subdomains output the almost same result. There are also X-Frame options in this domain. And that is low profile attack.

- This Arachni tool also provides finalized report to gather information about those vulnerabilities.

<https://www.jimdo.com/> Generated on 2021-10-25 07:12:10 +0530

Summary

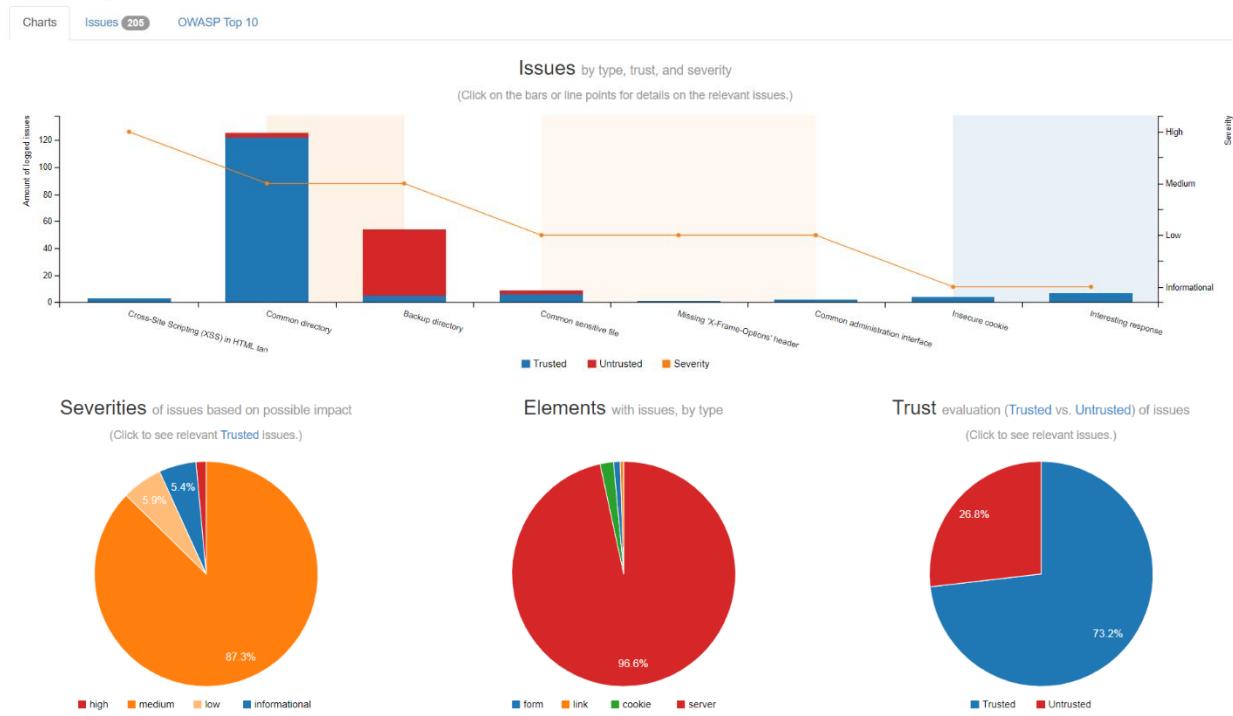


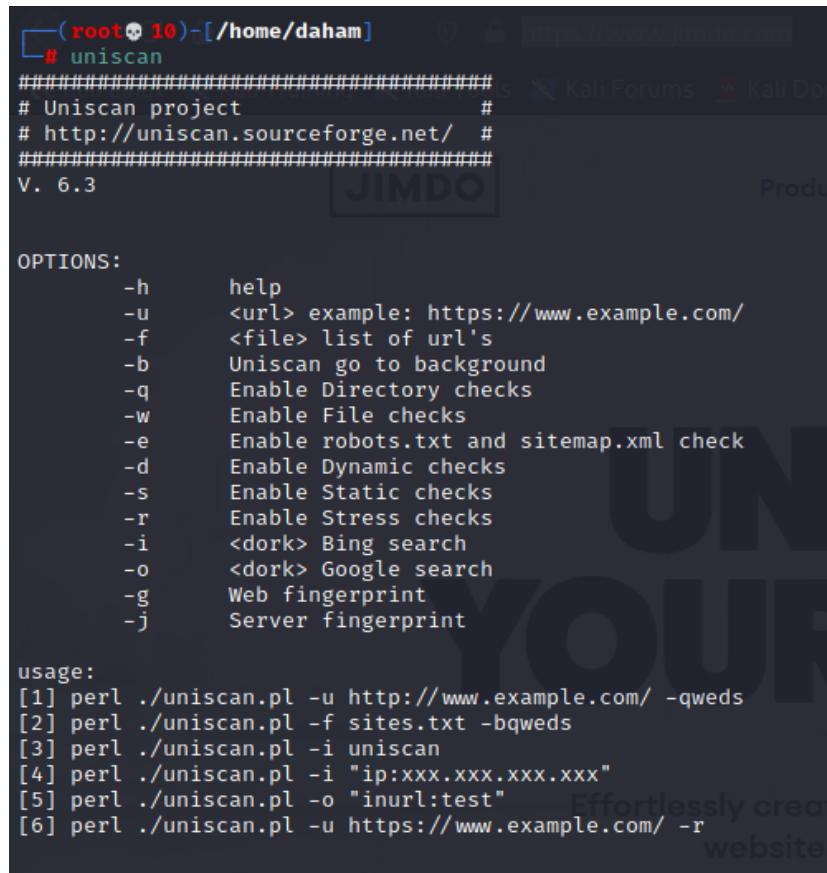
Fig. 66. Scan report of www.jimdo.com

So, according to this chart, we can analyze vulnerabilities perfectly. And we can get a good understanding of how much this domain is vulnerable to attackers.

➤ Uniscan

Uniscan is an open-source vulnerability detection tool that can be used to scan vulnerabilities in the targeted web application, such as, cross-site scripting(XSS), remote file inclusion, web shell vulnerabilities, SQL injection, blind SQL injection, and hidden backdoors. Also, the Uniscan tool is capable to do a Bing and Google search for finding domains on shared IP addresses.

So, this tool is inbuilt in the Kali Linux operating system, and we need to give root permission to access this tool. Uniscan tool can be manually configurable. So, this tool is suitable for the filtered way of scanning.



The screenshot shows a terminal window with the following content:

```
(root💀10)-[~/home/daham] # uniscan
#####
# Uniscan project      #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3
JIMDO
Productivity

OPTIONS:
-h      help
-u      <url> example: https://www.example.com/
-f      <file> list of url's
-b      Uniscan go to background
-q      Enable Directory checks
-w      Enable File checks
-e      Enable robots.txt and sitemap.xml check
-d      Enable Dynamic checks
-s      Enable Static checks
-r      Enable Stress checks
-i      <dork> Bing search
-o      <dork> Google search
-g      Web fingerprint
-j      Server fingerprint

usage:
[1] perl ./uniscan.pl -u http://www.example.com/ -qweds
[2] perl ./uniscan.pl -f sites.txt -bweds
[3] perl ./uniscan.pl -i uniscan
[4] perl ./uniscan.pl -i "ip:xxx.xxx.xxx.xxx"
[5] perl ./uniscan.pl -o "inurl:test"
[6] perl ./uniscan.pl -u https://www.example.com/ -r
```

Fig. 67. Options that can use to filtered way of scanning

These are the commands used to filter the scanning results. Because of that the manual configuration efficiently gets the scan result.

- Scan the result of the www.jimdo.com domain.

```
(root@10)-[~/home/daham]
# uniscan -u https://www.jimdo.com/ -bwedsgj
#####
# Uniscan project
# http://uniscan.sourceforge.net/
#####
V. 6.3

Going to background with pid: [4968]
Scan date: 19-10-2021 17:20:17

[root@10]-[~/home/daham]
#
Domain: https://www.jimdo.com/
Server: nginx/1
IP: [REDACTED]

TimThumb vulnerability: a big number of WordPress plugins and themes are affected

p class="pl">Payment methods from PayPal to SEPA direct debit
a href="https://account.e.jimdo.com/accounts/signup/" class="pm" data-tracking="nav_mobile_signup">Sign up
a href="https://account.e.jimdo.com/en/accounts/login/" class="pm" data-tracking="nav_mobile_login">Log In
a href="https://account.e.jimdo.com/en/accounts/logout/" class="pm" data-tracking="nav_mobile_logout">Logout
script data-react-helmet="true" type="application/ld+json">{ "@context": "http://schema.org", "@type": "Organization", "name": "Jimdo GmbH", "url": "https://www.jimdo.com", "sameAs": ["https://www.facebook.com/Jimdo", "https://twitter.com/jimdo"] }
}
data-tracking="Footer.visit_twitters" href="https://twitter.com/jimdo" aria-label="Visit Jimdo's Twitter page" rel="noopener noreferrer" target="_blank">
data-tracking="Footer.visit_facebook" href="https://www.facebook.com/Jimdo" aria-label="Visit Jimdo's Facebook page" rel="noopener noreferrer" target="_blank">
div class="index-module__login-R_q2P index-module__title-1Ww1">
nav class="index-module__loginState-3MwU">
script data-react-helmet="true" type="application/ld+json">{ "@context": "http://schema.org", "@type": "WebSite", "url": "https://www.jimdo.com", "name": "Jimdo", "alternateName": "Jimdo" }
}
script type="application/ld+json">{ "@context": "http://schema.org", "@type": "BreadcrumbList", "itemListElement": [{"@type": "ListItem", "position": 1, "name": "Home", "item": {"@id": "https://www.jimdo.com"} } ] }

WHOIS
No match for "WWW.JIMDO.COM".
NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
learn more about the registrant of this domain.
Learn More

BANNER GRABBING:
X-Meta-Generator: Gatsby
X-XSS-Protection: 1; mode=block

PING www.jimdo.com (199.232.46.2) 56(84) bytes of data.
64 bytes from 199.232.46.2 (199.232.46.2): icmp_seq=1 ttl=54 time=87.6 ms
64 bytes from 199.232.46.2 (199.232.46.2): icmp_seq=2 ttl=54 time=87.1 ms
64 bytes from 199.232.46.2 (199.232.46.2): icmp_seq=3 ttl=54 time=84.4 ms
64 bytes from 199.232.46.2 (199.232.46.2): icmp_seq=4 ttl=54 time=108 ms
--- www.jimdo.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 84.429/91.761/107.931/9.413 ms

TRACEROUTE
traceroute to www.jimdo.com (199.232.46.2), 30 hops max, 60 byte packets
1 [REDACTED] 0.288 ms 0.243 ms 0.231 ms
2 [REDACTED] 10.949 ms 10.937 ms 10.927 ms

NSLOOKUP
Server: 192.168.1.1
Address: 192.168.1.1#53

Non-authoritative answer:
www.jimdo.com canonical name = f2.shared.global.fastly.net.
Authoritative answers can be found from:
fastly.net
    origin = ns1.fastly.net
    mail addr = hostmaster.fastly.com
    serial = 2017052201
    refresh = 3600
    retry = 600
    expire = 604800
    minimum = 30
Name: www.jimdo.com
Address: 199.232.46.2
;; connection timed out; no servers could be reached

Acunetix | August 4, 2011

Recently a new high risk vulnerability was discovered in the highly popular TimThumb small PHP script for cropping, zooming and resizing web images (jpg, png, gif). Perfect for us applications.

TimThumb is included in a lot of WordPress plugins and themes (free and paid). Exploit attacker can upload and execute a PHP file of his choice on a vulnerable website. He
```

```

NMAP acunetix.com
Failed to resolve "www.jimdo.com". No targets were specified, so 0 hosts scanned.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-19 17:22 +0530
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
NSE: Script Post-scanning.
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
Initiating NSE at 17:22
Completed NSE at 17:22, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 0 IP addresses (0 hosts up) scanned in 10.45 seconds
Raw packets sent: 0 (0B) | Rcvd: 0 (0B)

Dynamic tests:
Plugin name: Learning New Directories v.1.2 Loaded.
Plugin name: FCKeditor tests v.1.1 Loaded.
Plugin name: Timthumb < 1.32 vulnerability v.1 Loaded.
Plugin name: Find Backup Files v.1.2 Loaded.
Plugin name: Blind SQL-injection tests v.1.3 Loaded.
Plugin name: Local File Include tests v.1.1 Loaded.
Plugin name: PHP CGI Argument Injection v.1.1 Loaded.
Plugin name: Remote Command Execution tests v.1.1 Loaded.
Plugin name: Remote File Include tests v.1.2 Loaded.
Plugin name: SQL-injection tests v.1.2 Loaded.
Plugin name: Cross-Site Scripting tests v.1.2 Loaded.
Plugin name: Web Shell Finder v.1.3 Loaded.
[+] 0 New directories added

FCKeditor tests:
Skipped because https://www.jimdo.com/testing123 did not return the code 404

Timthumb < 1.33 vulnerability:

Backup Files:
Skipped because https://www.jimdo.com/testing123 did not return the code 404

Blind SQL Injection:
Local File Include:
Recently a new high risk vulnerability was discovered in the highly popular TimThumb plugin. The vulnerability allows an attacker to upload a small php script for cropping, zooming and resizing web images (jpg, png, gif). Perfect for malicious applications."
PHP CGI Argument Injection:
Remote Command Execution:
TimThumb is included in a lot of WordPress plugins and themes (free and paid). An attacker can upload and execute a PHP file of his choice on a vulnerable website.
Remote File Include:
SQL Injection:
Cross-Site Scripting (XSS):
// check allowed sizes (if required)
if ($allow_resizeimage) {
    $allowresizeimage = true;
}
Web Shell Finder:

```

Fig. 68. Full scan report of www.jimdo.com
 this is the scan result we can get from this tool. So, there are no vulnerabilities captured by this tool. But Nmap and other scan results are important to find vulnerabilities in the targeted system.

➤ Netsparker

Netsparker is a powerful vulnerability detection tool. According to [Netsparker.com](https://www.netsparker.com), “Netsparker is an automated, yet fully configurable, web application security scanner that enables you to scan websites, web applications, and web services, and identify security flaws.” [9]. This scanner is covered every unsecured area in the web application and consistently outputs every bit of weakness that the targeted domain has. But this process takes a lot of time and processing power to output the result.

Why because this tool goes through a lot of vulnerabilities in the targeted domain. Such as Remote Code Evaluation, SQL injection, Blind SQL injection, Boolean SQL injection, Remote File Inclusion (RFI), Command injection, Server-side Template Injection, Blind Command Injection, Injection via Local File Inclusion, and Local File Inclusion (LFI). Not only that Netsparker tool provides a full report of the exploitation method of those found vulnerabilities and the prevention method.

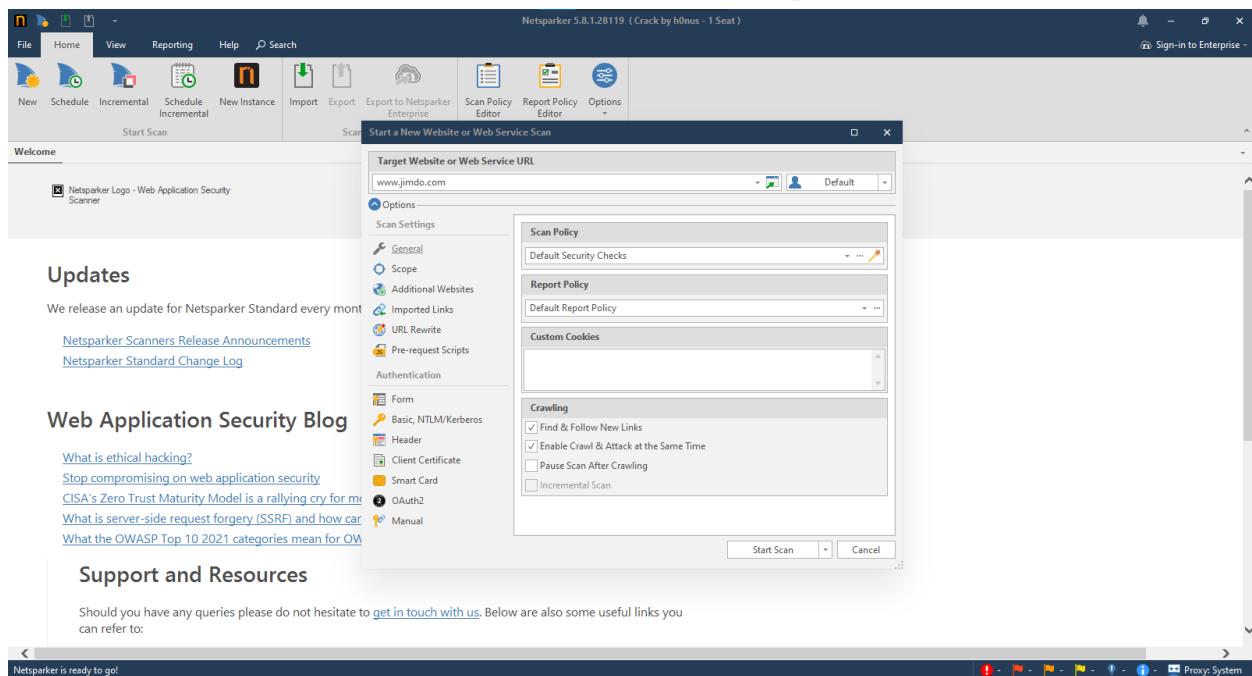


Fig. 69. Dashboard of the Netsparker Tool

So, this tool is automated, but we can customize technical details of the targeted domain and reduce the time taken to scan the target website.

- Customize the Web Server details of the targeted domain.

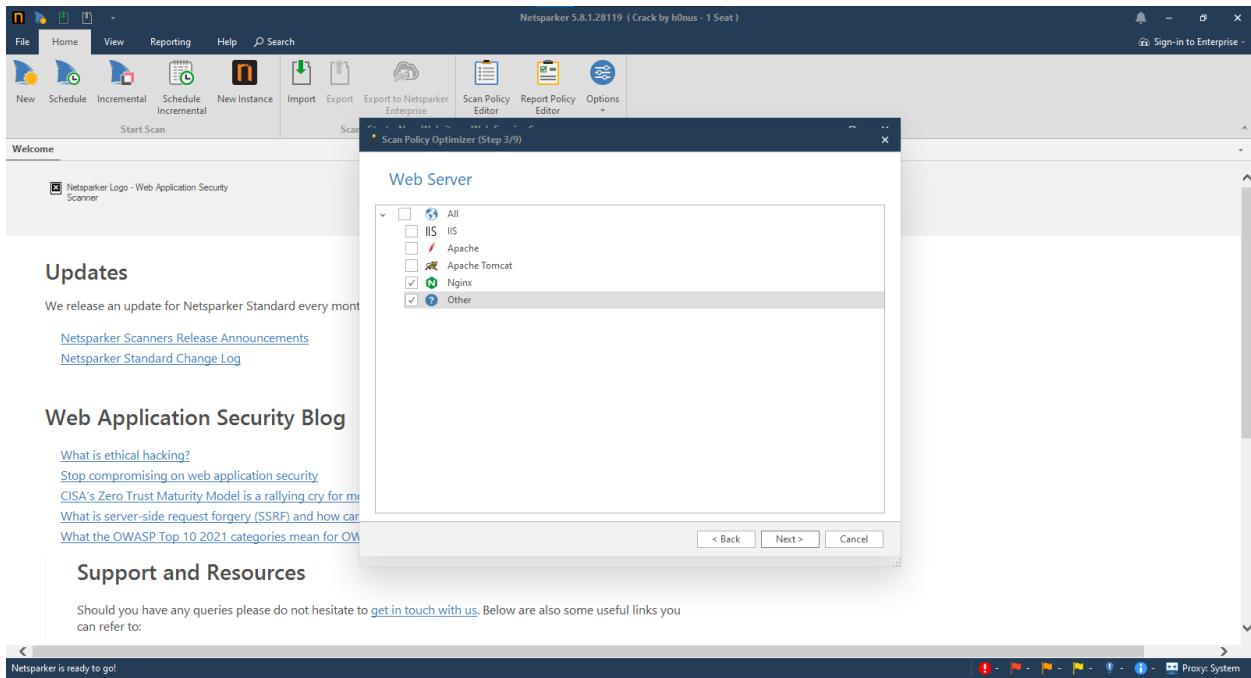


Fig. 70. Customize the Web Server details

- Customize the Application Server details of the targeted domain.

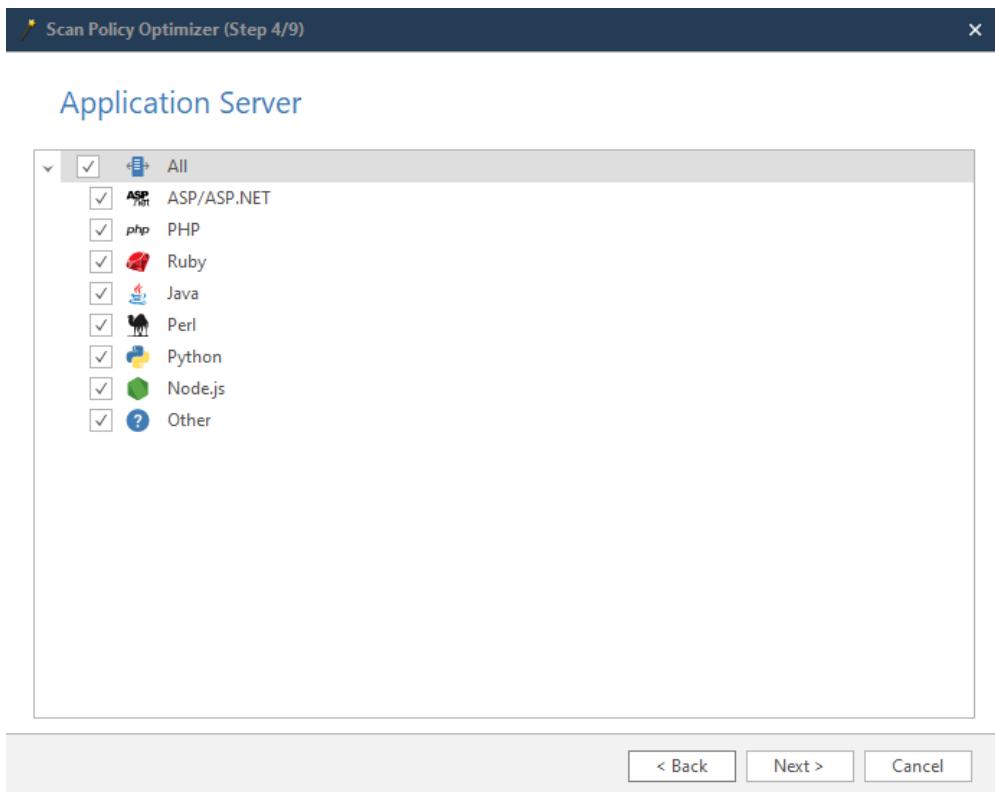


Fig. 71. Customize the Application Server details

- Customize the Database Server details of the targeted domain.

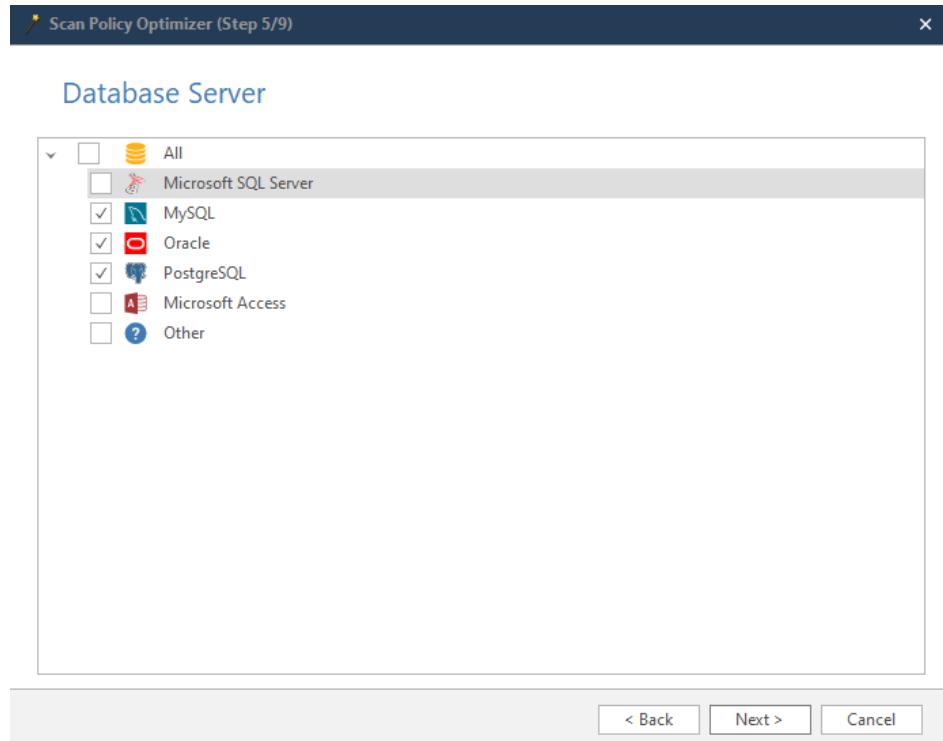


Fig. 72. Customize the Database Server details

- Targeted domain scanning process.

The screenshot shows the Netsparker application interface. The top bar includes 'File', 'Home', 'View', 'Reporting', 'Help', 'Scan' (selected), and 'Search'. The URL 'www.jimdo.com - Netsparker 5.8.1.28119 (Crack by h0nus - 1 Seat)' is displayed. The interface has several panels: 'Sitemap' (listing URLs like www.jimdo.com:80 and www.jimdo.com:443), 'Updates' (noting a security.txt detection), 'Web Application Security Blog' (linking to release announcements and change logs), 'Attacking [10]' (showing 10 active attacks with columns for Method, Target, Parameter, Duration, Current Activity, Overall Activity, and Status), and 'Netsparker Assistant (3)*' (displaying three notifications: 'Skip Threshold Reached', 'DOM Simulation Timeout Exceeded', and 'Maximum Signature Exceeded'). The bottom status bar shows 'Crawl and Attack phase started.', 'Crawling & Attacking (2/3) 2%', and other system information.

Fig. 73. Scanning process

After scanning the domain Netsparker tool output the vulnerability scanning report as a PDF file. So, these are the scanning results according to each domain.

- Scan report of www.jimdo.com.

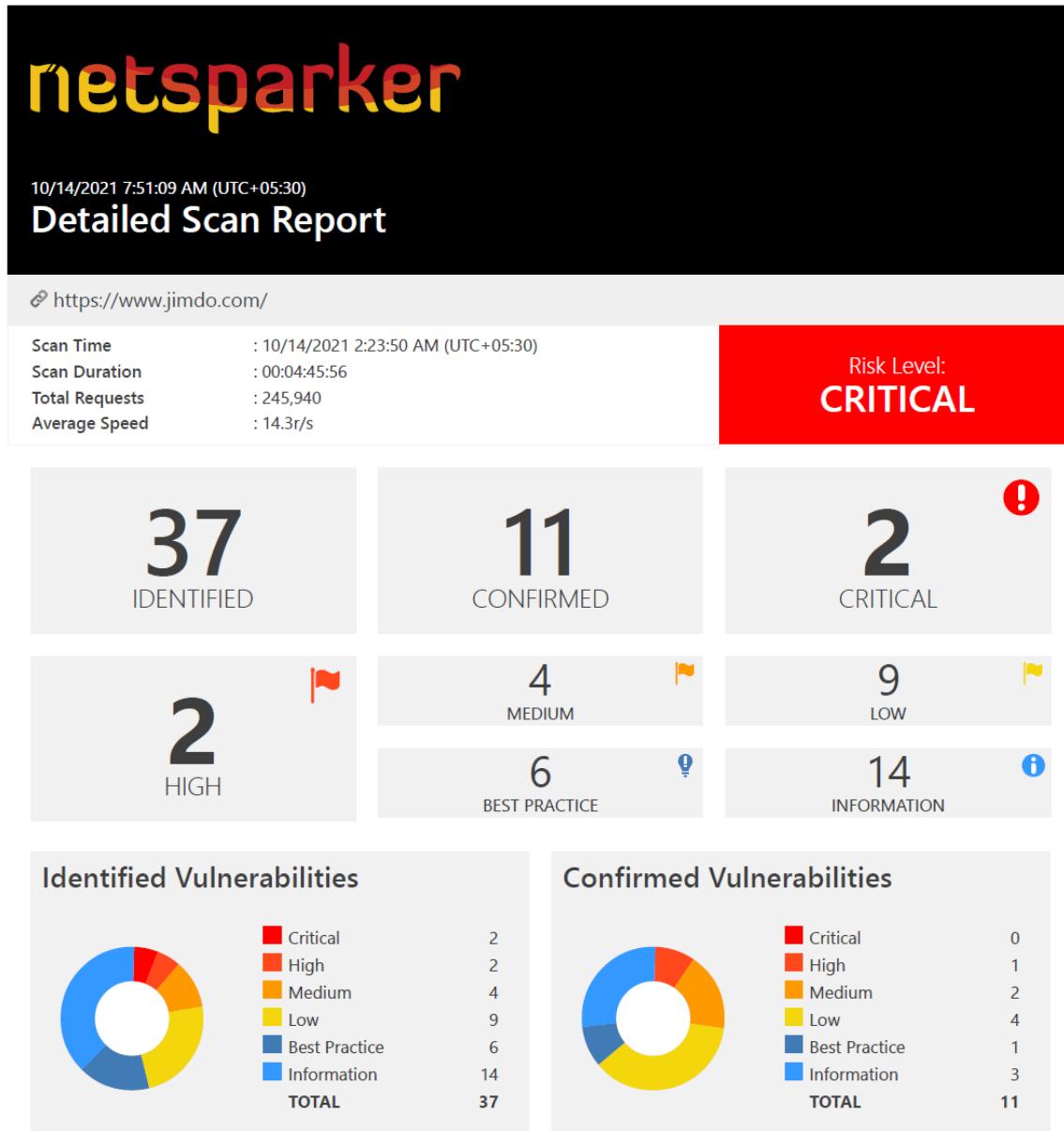


Fig. 74. Summary of the scan report

This is the report summary of the scanned vulnerabilities of the main domain. Netsparker tool identified two Critical and two High damageable vulnerabilities in the main domain.

Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
!	Out-of-date Version (Lodash)	GET	https://www.jimdo.com/	
!	Out-of-date Version (Nginx)	GET	https://www.jimdo.com/	
!	Out-of-date Version (Modernizr)	GET	https://www.jimdo.com/	
!	Session Cookie Not Marked as Secure	GET	https://www.jimdo.com/	
!	HTTP Strict Transport Security (HSTS) Errors and Warnings	GET	https://www.jimdo.com/	
!	Out-of-date Version (jQuery)	GET	https://www.jimdo.com/	
!	Active Mixed Content over HTTPS	GET	https://www.jimdo.com/about-jimdo	
!	Weak Ciphers Enabled	GET	https://www.jimdo.com/	

Fig. 75. Vulnerability Summary

These are the vulnerabilities that scan by the Netsparker tool and there are three confirmed vulnerabilities. One of these is High-level damageable vulnerability.

Jimdo's website uses out-of-date version servers and other services. Also, that the confirmed vulnerability is about the security issue of Session and Cookies. So, compared with other vulnerability detection tools Netsparker is a very efficient and more reliable tool.

Consider the first vulnerability captured by this tool, it is about the Lodash server-side software used in this website is the Out-of-date version. In the information gathering stage, we found older version software used to develop this targeted domain.

1. Out-of-date Version (Lodash)

CRITICAL  1

Netsparker identified that the target web site is using Lodash and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

Lodash Improper Neutralization of Special Elements used in a Command ('Command Injection') Vulnerability

** DISPUTED ** A command injection vulnerability in Lodash 4.17.21 allows attackers to achieve arbitrary code execution via the template function. This is a different parameter, method, and version than CVE-2021-23337. NOTE: the vendor's position is that it's the developer's responsibility to ensure that a template does not evaluate code that originates from untrusted input.

Affected Versions

4.17.21

External References

- [CVE-2021-41720](#)

Vulnerabilities

1.1. <https://www.jimdo.co> [REDACTED]

Identified Version

- 4.17.21

Latest Version

- 4.17.21 (in this branch)

Vulnerability Database

- Result is based on 10/13/2021 20:30:00 vulnerability database content.

Certainty



Fig. 76. Out-of-date Version (Lodash)

This older version of the software is vulnerable to the attacker because of the bugs have in the older versions. So, this software needs to be updated to a newer version or switch to the low risky another newer version of the software which the same work done in Lodash software.

According to the Educative website, “Lodash is a JavaScript library that provides utility functions for common programming tasks using a functional programming paradigm; it builds upon the older underscore.js library.” [10]. So, this software also does not

updatable because this software is in the latest version. Instead of Lodash software, a newer version of the software can be used.

According to Netsparker, there is a command injection vulnerability that allows attackers to achieve arbitrary code execution via the template function and the targeted subdomain also mention in this report (<https://www.jimdo.com> [REDACTED]).

Other Critical vulnerability is also the Out-of-date Version vulnerability of the Nginx software.

2. Out-of-date Version (Nginx)

CRITICAL  1

Netsparker identified you are using an out-of-date version of Nginx.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

Nginx Allocation of Resources Without Limits or Throttling Vulnerability

Some HTTP/2 implementations are vulnerable to a header leak, potentially leading to a denial of service. The attacker sends a stream of headers with a 0-length header name and 0-length header value, optionally Huffman encoded into 1-byte or greater headers. Some implementations allocate memory for these headers and keep the allocation alive until the session dies. This can consume excess memory.

Affected Versions

1.9.5 to 1.16.0

External References

- [CVE-2019-9516](#)

Nginx Other Vulnerability

Some HTTP/2 implementations are vulnerable to resource loops, potentially leading to a denial of service. The attacker creates multiple request streams and continually shuffles the priority of the streams in a way that causes substantial churn to the priority tree. This can consume excess CPU.

Affected Versions

1.9.5 to 1.16.0

External References

- [CVE-2019-9513](#)

⚠ Nginx Allocation of Resources Without Limits or Throttling Vulnerability

Some HTTP/2 implementations are vulnerable to window size manipulation and stream prioritization manipulation, potentially leading to a denial of service. The attacker requests a large amount of data from a specified resource over multiple streams. They manipulate window size and stream priority to force the server to queue the data in 1-byte chunks. Depending on how efficiently this data is queued, this can consume excess CPU, memory, or both.

Affected Versions

1.9.5 to 1.16.0

External References

- [CVE-2019-9511](#)

❗ Nginx Off-by-one Error Vulnerability

A security issue in nginx resolver was identified, which might allow an attacker who is able to forge UDP packets from the DNS server to cause 1-byte memory overwrite, resulting in worker process crash or potential other impact.

Affected Versions

1.7.4 to 1.20.0

External References

- [CVE-2021-23017](#)

Fig 77. Out-of-date Version (Nginx)

According to the Nginx, “NGINX is open-source software for web serving, reverse proxying, caching, load balancing, media streaming, and more.” [11]. Because of that, this is critical software used in the targeted domain. That is why need to all the time update critical functioning software like Nginx.

So, this vulnerability is identified also in the information gathering stage. When using Netcraft and Whatweb tools it mentions that this domain uses an older version of Nginx software. So, that assumes as vulnerable to attack and it is confirmed by the Netsparker tool.

According to the Netsparker tool, this is HTTP/2 implementations are vulnerable to a header leak, resource loops, window size manipulation, and stream prioritization manipulation, potentially leading to denial of service attacks. And there is vulnerable to an attacker who can falsify DNS server UDP packets to trigger a 1-byte memory overwrite, resulting in worker process

crash or other possible effects. These attacks can be done directly to the main domain (<https://www.jimdo.com/>).

Other vulnerabilities are High and Medium level vulnerabilities. First, consider that there is a High level of confirmed vulnerability regarding Session and Cookies.

4. Session Cookie Not Marked as Secure

HIGH  1 CONFIRMED  1

Netsparker identified a session cookie not marked as secure, and transmitted over HTTPS.

This means the cookie could potentially be stolen by an attacker who can successfully intercept the traffic, following a successful man-in-the-middle attack.

It is important to note that Netsparker inferred from its name that the cookie in question is session related.

Impact

This cookie will be transmitted over a HTTP connection, therefore an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he/she can force the victim to make an HTTP request to your website in order to steal the cookie.

Vulnerabilities

4.1. <https://www.jimdo.co> [REDACTED]

CONFIRMED

Identified Cookie(s)

- PHPSESSID

Cookie Source

- HTTP Header

Fig. 78. Session and Cookie Not Marked as Secure

This vulnerability is that the Session and Cookies are not protected using protection methods. Consider the Session and Cookies are used to maintain the state of the web application. So, these types of vulnerabilities are fair enough to leak your account details to the attacker. Because of the man-in-the-middle attack possible to happen with those vulnerabilities in the targeted domain. And also targeted domain is mentioned and confirmed in the report (<https://www.jimdo.com> [REDACTED]).

Next, consider the Medium level much important vulnerability that we need aware of. This vulnerability is related to HTTP Strict Transport Security (HSTS).

6. HTTP Strict Transport Security (HSTS) Errors and Warnings

MEDIUM



1

Netsparker detected errors during parsing of Strict-Transport-Security header.

Impact

The HSTS Warning and Error may allow attackers to bypass HSTS, effectively allowing them to read and modify your communication with the website.

Vulnerabilities

6.1. <https://www.jimdo.com/>

Error

Resolution

preload directive not present

Submit domain for inclusion in browsers' HTTP Strict Transport Security (HSTS) preload list.

Certainty

Fig. 79. HTTP Strict Transport Security (HSTS) Errors and Warnings

In the information gathering stage that the Whatweb tool used found the HTTP Strict Transport Security (HSTS) is not in this web application according to the status code used in targeted domains. Because of that, this website can be vulnerable to the SSL Stripping attack. Even this is a Medium level of vulnerability that the impact is in the critical stage. Not only this vulnerable domain is also the main domain of this web application (<https://www.jimdo.com/>).

- Scan report of jimdo.design

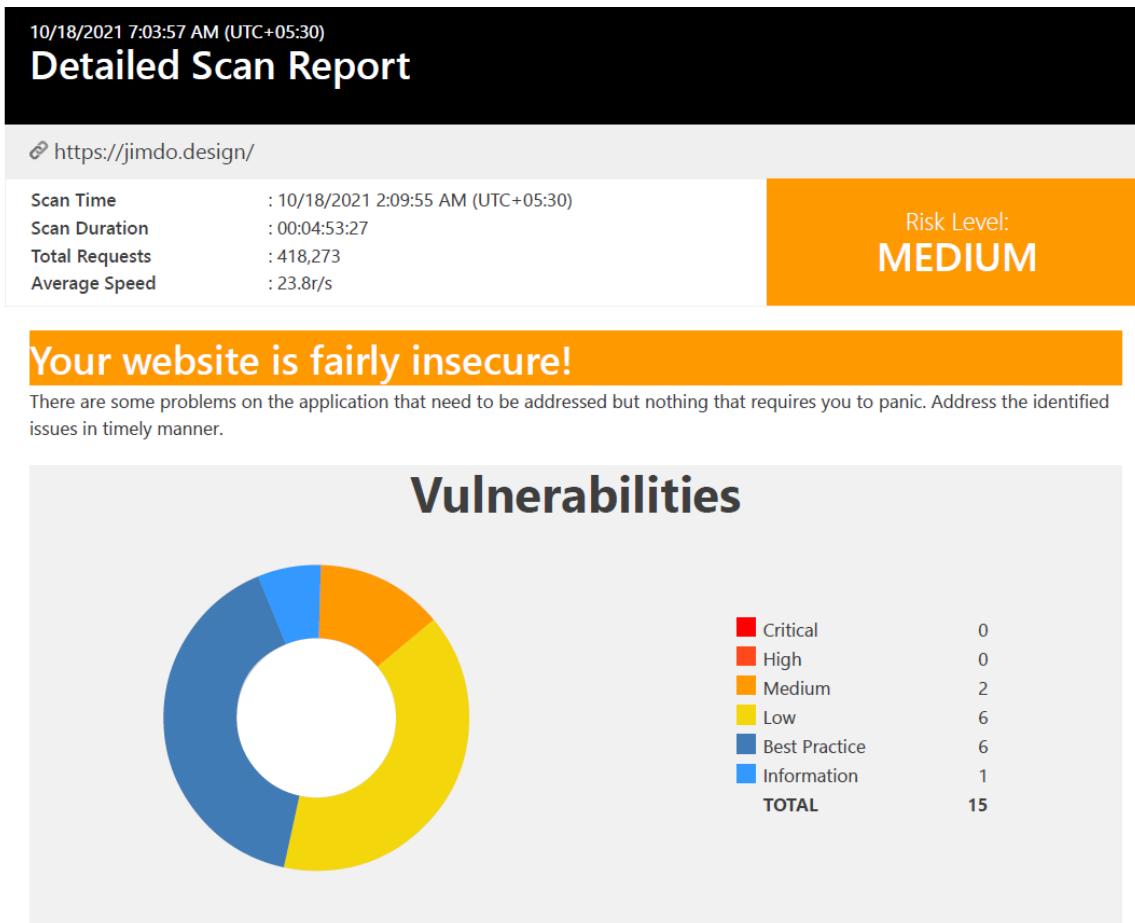


Fig. 80. Report of jimdo.design

- Scan report of halp.jimdo.com

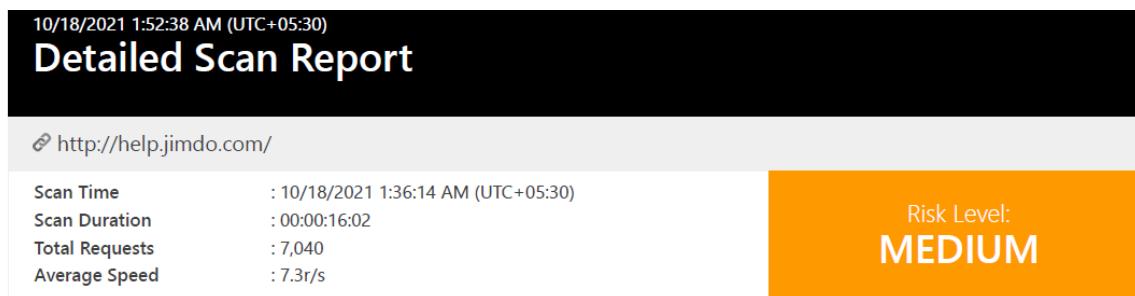


Fig. 81. Report of help.jimdo.com

Other scanned reports also do not capture any Critical or High level vulnerabilities.

➤ Owasp ZAP

The Open Web Application Security Project Zed Attack Proxy (OWASP ZAP) is an open-source web application vulnerability detection tool. This is one of the best vulnerability detection tool and efficient than compared with most other tools. Owasp ZAP is also can be used as web application professional penetration testers. This tool work according to the OWASP top 10 security threats. Such as Cross-site scripting (XSS), Broken access control, SQL injection, Broken authentication and session management, Security misconfiguration and other security threats.

Consider that how does Owasp ZAP work, according to Srijan's Framework and Libraries, "ZAP creates a proxy server and makes your website traffic pass through that server. It comprises of auto scanners that help you intercept the vulnerabilities in your website." [12]. There is an automated or manual scanning option and for this assignment choose that the automated scan method because the automated method filter and scan only the in-scope subdomains. The automated scan is also customizable and if it is customized well, we can reduce that time taken for scanning the targeted subdomain.

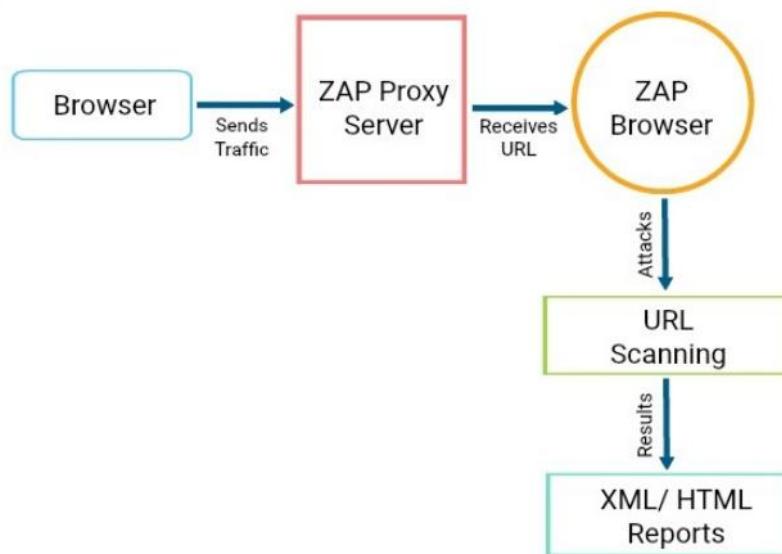


Fig. 82. How does Owasp ZAP work

- Customize the scan policy and add a new policy according to the targeted subdomain.

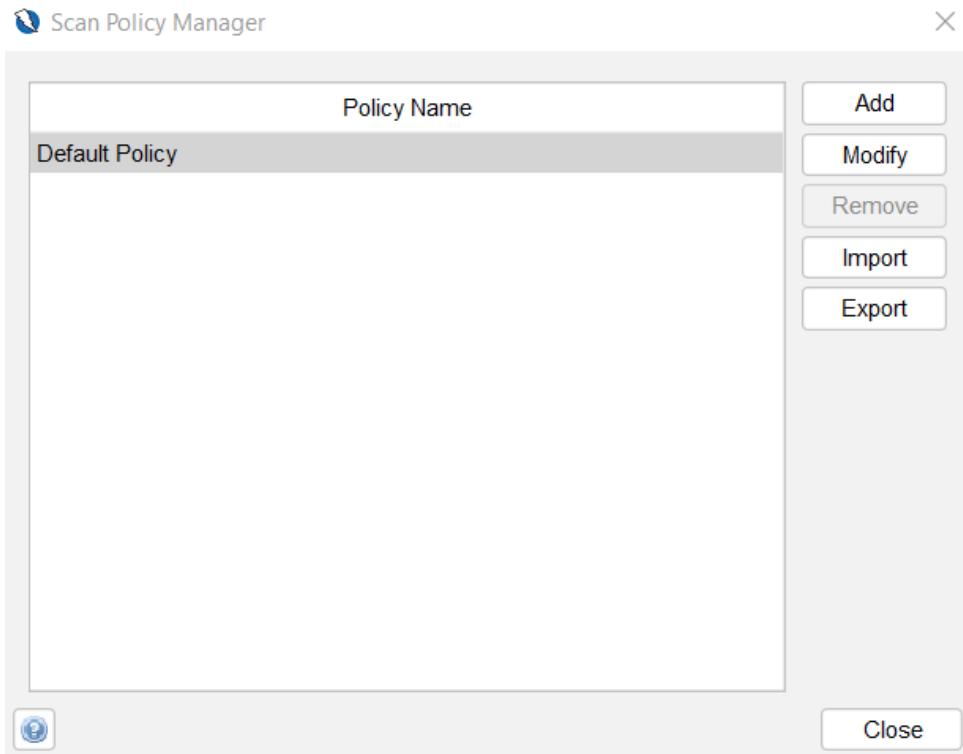


Fig. 83. Scan Policy Manager

- Change the scan policies according to the targeted domain.

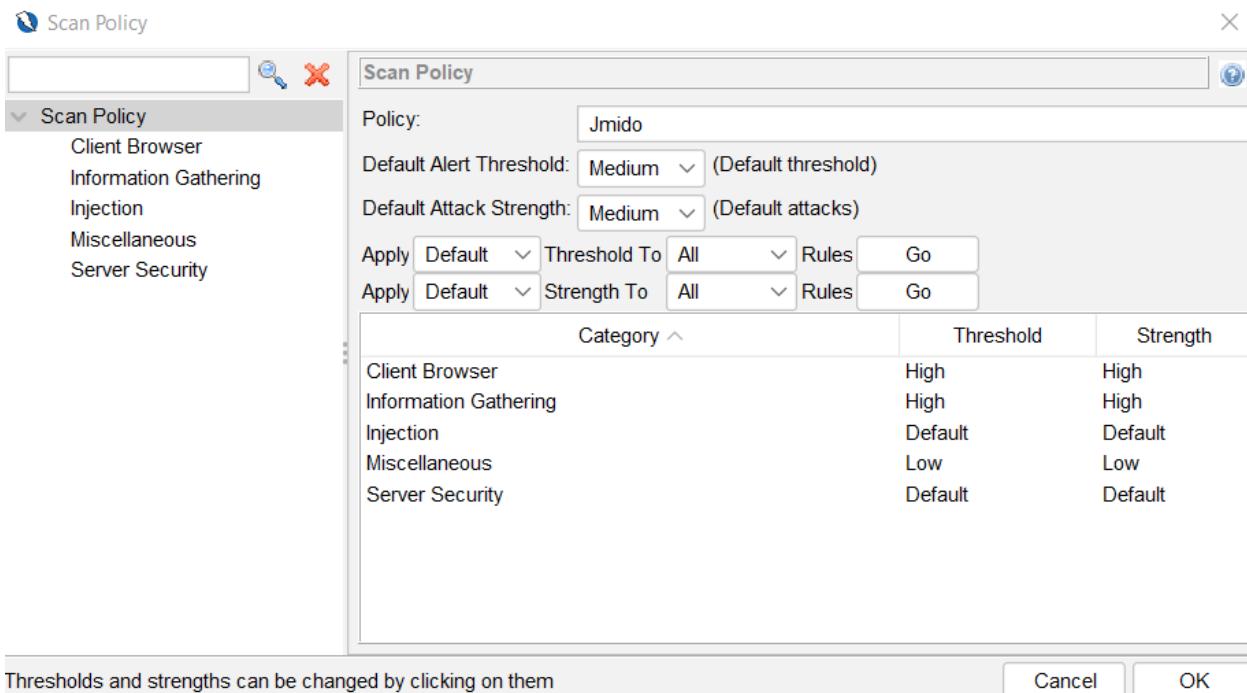


Fig. 84. Customize the new scan policy profile

- Scanning process of www.jimdo.com

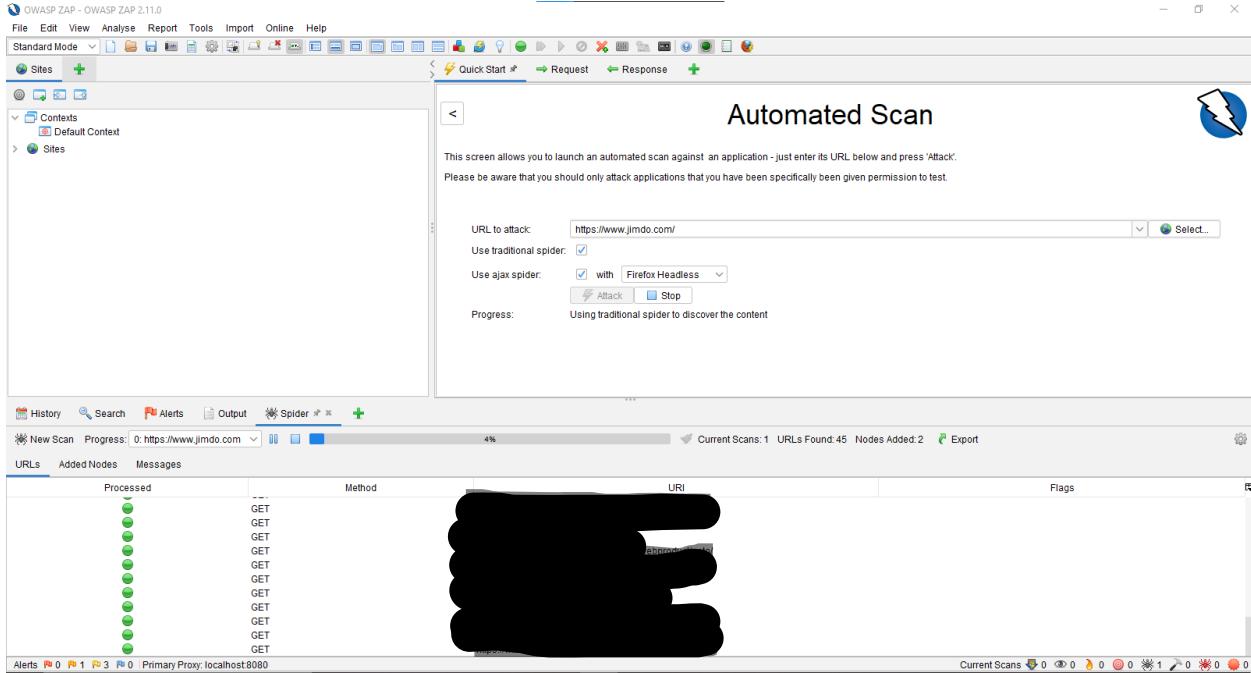
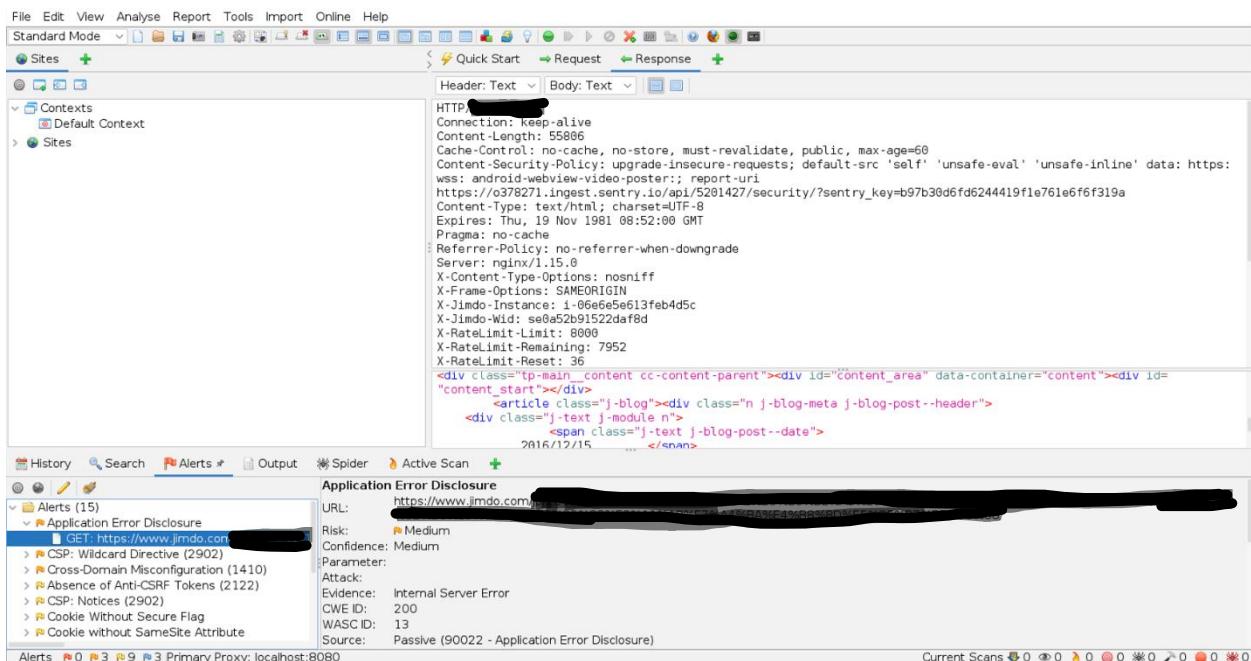


Fig. 85. Automated Scanning Process

- Scan results of www.jimdo.com



```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 528848
Cache-Control: public, max-age=60
Content-Security-Policy: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' data: https://wss: android-webview-video-poster;; report-uri https://o378271.ingest.sentry.io/api/5201427/security/?sentry_key=b97b30d6fd6244419f1e761e6f6f319a
Content-Type: text/html; charset=UTF-8
ETag: "61690f22-811d0"
Last-Modified: Fri, 15 Oct 2021 05:18:26 GMT
Referer-Policy: no-referrer-when-downgrade
Server: nginx/1.15.0
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Via: 1.1 varnish, 1.1 varnish
Accept-Ranges: bytes
Date: Fri, 15 Oct 2021 14:16:59 GMT
Age: 26511

Very large response body (528,848 bytes) - switch views (using the pulldown currently showing Body: Large Response above) to display.
Be aware that this message may take some time to load.
You can change the minimum message size used for the Large Response view via Options / Display.

CSP: Wildcard Directive
URL: https://www.[REDACTED]
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' data: https://wss: android-webview-video-poster;; report-uri http://[REDACTED]
CWE ID: 693
WASC ID: 15
Source: Passive (10055 - CSP)

CSP: Wildcard Directive
URL: https://www.[REDACTED]
Risk: Medium
Confidence: Medium
Parameter:
Attack:
Evidence: default-src 'none'; frame-src 'self' www.facebook.com staticxx.facebook.com cms.e.jimdo.com checkout.jimdo.com dash.e.jimdo.com stats.e.jimdo.com register.jimdo.com cms.jimdo.com *.hotjar.com a.jimdo.com *.fls.doubleclick.net td.jimdo.com *.googlesyndication.com domainsfrontend-prod.jimdo-platform.net; connect-src 'self' *.hotjar.com *.bugsraq.com t.jimdo-platform.net *.abtasty.com *.fullstory.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.tiktok.com *.dcmn.io img-sr 'self' www.facebook.com t.jimdo-platform.net *.bboxbox.co td.jimdo.com www.jimdo.com *.peaksandpies.io data: b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.pinimg.com *.pinterest.com *.taboola.com tagmanager.google.com www.googletagmanager.com *.googlesyndication.com *.tiktok.com *.dcmn.io www.google.com www.google.de www.google.at www.google.ch www.google.co.jp ssl.gstatic.com www.gstatic.com cx.admt.com *.jimstatic.com font-src 'self' data: jindo.github.io fonts.gstatic.com *.jimstatic.com style-src 'self' 'unsafe-inline' jimdo.github.io fonts.googleapis.com *.jimstatic.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.pinimg.com *.pinterest.com *.taboola.com tagmanager.google.com www.googletagmanager.com *.googlesyndication.com *.tiktok.com *.dcmn.io script-src 'self' 'unsafe-inline' 'unsafe-eval' connect.facebook.net a.jimdo.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.pinimg.com *.

CSP: Wildcard Directive
URL: https://www.[REDACTED]
Risk: Medium
Confidence: Medium
Parameter:
Attack:
default-src 'none'; frame-src 'self' www.facebook.com staticxx.facebook.com cms.e.jimdo.com checkout.jimdo.com dash.e.jimdo.com stats.e.jimdo.com register.jimdo.com cms.jimdo.com *.hotjar.com a.jimdo.com *.fls.doubleclick.net td.jimdo.com *.googlesyndication.com domainsfrontend-prod.jimdo-platform.net; connect-src 'self' *.hotjar.com *.bugsraq.com t.jimdo-platform.net *.abtasty.com *.fullstory.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.tiktok.com *.dcmn.io img-sr 'self' www.google.de www.google.at www.google.ch www.google.co.jp ssl.gstatic.com www.gstatic.com cx.admt.com *.jimstatic.com font-src 'self' data: jindo.github.io fonts.gstatic.com *.jimstatic.com style-src 'self' 'unsafe-inline' jimdo.github.io fonts.googleapis.com *.jimstatic.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.pinimg.com *.pinterest.com *.taboola.com tagmanager.google.com www.googletagmanager.com *.googlesyndication.com *.tiktok.com *.dcmn.io script-src 'self' 'unsafe-inline' 'unsafe-eval' connect.facebook.net a.jimdo.com b97.yahoo.co.jp *.doubleclick.net *.tvssquared.com *.outbrain.com *.bing.com *.ytag.jp *.google-analytics.com www.googleleadservices.com *.pinimg.com *.

```

Fig. 86. Scan Results

Even this is a customized scan this tool consumes a lot of time to process the scan. Because of that, I had to abort the scan and get the result. So, these results are not the finalized result, and I already identified the vulnerabilities in the targeted system using the Netsparker and other tools.

Penetration Testing

Penetration Testing is a really important stage in Bug Bounty Assessment. Because in this stage test the scanned vulnerabilities found in the targeted subdomain. So, we can find out that vulnerabilities are actually exploitable or not. According to the National Institute of Standards and Technology (NIST), “Penetration Testing is a method of testing where testers target individual binary components or the application as a whole to determine whether intra or intercomponent vulnerabilities can be exploited to compromise the application, its data, or its environment resources.” [13]. Also, this process is named ethical hacking or pen-testing. So, this process can help to confirm the vulnerabilities in the targeted system.

Penetration testing is a crucial aspect in the confirmation of data security in every aspect of the data is used today. The necessity of it is highlighted due to the benefits it gives. Penetration tests allow us to identify new bugs and loopholes in existing software, test new software for existing bugs, and whether the implemented security controls are sufficient to handle the latest security threats. It enables us or our company to be able to stay up to standard with recognized international standards like General Data Protection Regulation (EU GDPR), Data Protection Act (DPA), Payment Card Industry Data Security Standard (PSI DSS), fix the identified bugs and loopholes in security controls that have already been implemented to assure our clients and stakeholders that their data is secure.

After confirming those vulnerabilities, we need to report these vulnerabilities and the protection methods to the relevant company belong the targeted system. And this process needs to be done before attackers exploit the system.

In the vulnerability detection stage, there are identified Critical level and High level vulnerabilities. In the penetration testing stage check, these identified vulnerabilities are impacting the targeted domains and suggest that the protection techniques secure the target domains. So, mainly most successful scanning results are produced by the Arachni tool and the Netsparker tool.

➤ Penetration testing according to the Arachni scan results

The Arachni tool found a High-level vulnerability and three subdomains are impacts for this same Cross-Site Scripting (XSS) vulnerability in HTML tag.

- ✓ https://www.jimdo.com/[REDACTED]
- ✓ https://www.jimdo.com/[REDACTED]
- ✓ https://www.jimdo.com/[REDACTED]

These are the three subdomains vulnerable to this attack. According to the Jimdo company policies, As this type of behavior is intended, each and every report which asks a user to place a Stored XSS payload on their own page to exploit anonymous users visiting the website, is taken as Out of Scope and will be closed as such [14]. So, we can not test those vulnerabilities as anonymous users and exploit those subdomains. Because of that, I do not attempt to exploit and see the results. But according to Arachni's scan report, I will suggest the prevention methods, what type of Cross-Site Scripting (XSS) vulnerability, and how it happens.

Modern-day web applications make extensive use of client-side scripting. Client-side scripts handle anything from basic text formatting to comprehensive data processing and operating system interactions on the client-side. So, clients can inject scripts into a request and have the server respond with scripts. This is known as cross-site scripting (XSS) and this occurs as a result of web applications acquiring untrustworthy data and reusing it without carrying out any validation or sanitization. There are three main types of cross-site scripting attacks,

- Reflected cross-site scripting
 - ✓ where the malicious script comes from the current HTTP request [15].
- Stored cross-site scripting
 - ✓ where the malicious script comes from the website's database [15].
- DOM-based cross-site scripting
 - ✓ where the vulnerability exists in client-side code rather than server-side code [15].

So, the Reflected cross-site scripting and the Stored cross-site scripting attacks are mostly can be done with this HTML injection vulnerability. It has been discovered by Arachni that it is almost possible to directly insert content into an HTML tag. For example <INJECTION_HERE href=.....etc> where INJECTION_HERE shows the place where the payload was detected by Arachni.

Because many browsers attempt to implement XSS protection, any manual verification of this finding should be conducted using multiple different browsers and browser versions.

The figure consists of three vertically stacked screenshots from the Arachni web application, each showing a different URL path and its corresponding XSS findings:

- Top Screenshot:** URL: https://www.jimdo.com/search/. It shows an injected seed of "<arachni_xss_in_tag=cd341739586d6da364a5f54aa05eb535& blah=>" and a proof of "<arachni_xss_in_tag=cd341739586d6da364a5f54aa05eb535& blah=>". Below these are sections for Vector information, Affected page (https://www.jimdo.com/search/), and Referring page (https://www.jimdo.com/search).
- Middle Screenshot:** URL: https://www.jimdo.com/search/?q=1&filter=0&module=1. It shows the same injected seed and proof as the top screenshot. Below these are sections for Vector information, Affected page (https://www.jimdo.com/search/?q=1&filter=0&module=1), and Referring page (https://www.jimdo.com/search).
- Bottom Screenshot:** URL: https://www.jimdo.com/fr/search/. It shows the same injected seed and proof as the other screenshots. Below these are sections for Vector information, Affected page (https://www.jimdo.com/fr/search/), and Referring page (https://www.jimdo.com/fr/search).

Fig. 87. Proof the Cross-Site Scripting (XSS) vulnerability in HTML tag

Consider how to protect from this vulnerability, It is critical to avoid using untrusted or unfiltered data in HTML page code. Untrusted data might come from a variety of sources, including the client, a third party, a previously submitted file, and so on. And also, Converting special characters to their HTML entity encoded counterparts is a common method of filtering untrustworthy material. As an example converting ">" to ">". Despite the fact that unsafe input may be filtered, there are five locations on an HTML page where it should never be placed, and Each of these locations has its own method of escape and filtration. as an example, these are the location unsafe inputs can happen, inside an HTML comment, directly in a script, in

a tag name, in an attribute name, and so on. So, these are the prevention methods that can be used to protect against HTML injection attacks.

➤ Penetration testing according to the Netsparker scan results

The Netsparkrt tool found several Critical level vulnerabilities. So, in this assignment consider that penetration testing the two Critical vulnerabilities that can be a threat to the targeted system.

1. Out-of-date Version (Lodash) Critical vulnerability

The current version that the Lodash use in Jimdo's system is an outdated version 4.█████ and it is vulnerable to the command injection attack. The vendor's position is that it's the developer's responsibility to ensure that a template does not evaluate code that originates from untrusted input. So, these types of vulnerabilities allow attackers to use the template function to execute arbitrary code.

✓ <https://www.jimdo.com/████████>

This is the subdomain vulnerable to this attack. Command injection vulnerability can impact the targeted domain when the system receives data from an untrustworthy source, the data is part of a string that the system interprets as a command, and the application grants an attacker, a privilege or capability that they would not otherwise have by performing the command. According to the, “If a malicious user injects a character (such as a semi-colon) that delimits the end of one command and the beginning of another, it may be possible to then insert an entirely new and unrelated command that was not intended to be executed.” [16].

So, the only way to protect from this command injection attack is to upgrade the installation of Lodash to the latest stable version.

Request

```
GET /info/terms-of-service/ HTTP/1.1
Host: www.jimdo.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: shd=5d1776a2-4242-4622-9963-4a7b40b28b5a; ckies_functional=deny; ckies_marketing=deny; ckies_necessary=allow; ckies_performance=deny; jLang=en; _dd_s=logs=1&id=89733f59-ce15-41a2-ab2f-b809b2b30caa&created=1634158440124&expire=1634159340129&rum=1
Referer: https://www.jimdo.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) : 288.0005 Total Bytes Received : 492805 Body Length : 491888 Is Compressed : No
```

```
HTTP/1.1 200 OK
X-Cache: HIT, HIT
Age: 33197
Cache-Control: public, max-age=60
ETag: W/"6166be3b-78170"
Strict-Transport-Security: max-age=31536000
Server: nginx/1.15.0
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Connection: keep-alive
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
X-Cache-Hits: 3, 1
Content-Length: 90354
Vary: Accept-Encoding
Via: 1.1 varnish, 1.1 varnish
Last-Modified: Wed, 13 Oct 2021 11:08:43 GMT
Content-Type: text/html; charset=UTF-8
X-Served-By: cache-lcy19256-LCY, cache-qpg1227-QPG
Content-Security-Policy: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' data: https: wss: android-webview-video-poster:; report-uri https://o378271.ingest.sentry.io/api/5201427/security/?sentry_key=b97b30d6fd6244419f1e761e6f6f319a
Date: Wed, 13 Oct 2021 20:54:22 GMT
Content-Encoding:
```

Fig. 88. Request and Response of the vulnerable domain

2. Out-of-date Version (Nginx) Critical vulnerability

In the information gathering and the vulnerability detection stages already concerned about this vulnerability and the prevention methods. <https://www.████████> is infected for this vulnerability. As the protection method suggests upgrading the installation of Nginx to the latest stable version.

Vulnerabilities	
2.1. https://[REDACTED]	
Identified Version	
• 1.15.0	
Latest Version	
• 1.21.3 (in this branch)	
Fig. 89. Latest Version	
So, in previous stages provide a lot of details about this vulnerability and I am not going to describe those things again and again here.	
<pre>Content-Security-Policy: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' d ta: https: wss: android-webview-video-poster:; report-uri https://o378271.ingest.sentry.io/api/5201427 security/?sentry_key=b97b30d6fd6244419f1e761e6f6f319a Date: Wed, 13 Oct 2021 20:53:51 GMT ConHTTP/1.1 200 OK X-Cache: HIT, HIT Age: 33170 Cache-Control: public, max-age=60 ETag: W/"6166be3b-80500" Strict-Transport-Security: max-age=31536000 Server: nginx/1.[REDACTED] X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Connection: keep-alive Referrer-Policy: no-referrer-when-downgrade X-Frame-Options: SAMEORIGIN Accept-Ranges: bytes X-Cache-Hits</pre>	

Fig. 89. Latest Version

So, in previous stages provide a lot of details about this vulnerability and I am not going to describe those things again and again here.

Content-Security-Policy: upgrade-insecure-requests; default-src 'self' 'unsafe-eval' 'unsafe-inline' data: https: wss: android-webview-video-poster:; report-uri https://o378271.ingest.sentry.io/api/5201427/security/?sentry_key=b97b30d6fd6244419f1e761e6f6f319a
Date: Wed, 13 Oct 2021 20:53:51 GMT
ConHTTP/1.1 200 OK
X-Cache: HIT, HIT
Age: 33170
Cache-Control: public, max-age=60
ETag: W/"6166be3b-80500"
Strict-Transport-Security: max-age=31536000
Server: nginx/1.17.8
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Connection: keep-alive
Referrer-Policy: no-referrer-when-downgrade
X-Frame-Options: SAMEORIGIN
Accept-Ranges: bytes
X-Cache-Hits

Fig. 90. Proof of this vulnerability

Conclusion

The purpose of this assignment was to assess the vulnerabilities of the web application. While engaging in this assignment I was able to identify a lot of Bug Bounty hunting platforms which helped to improve vulnerability assessing skills and knowledge about the penetration testing tool and how to use those tools. I decided to use the Hackerone platform because this website legally protects us to do Bug Bounty hunting for real-world web applications. The web audit reports give an excellent understanding of how to handle cybersecurity professional skills.

References

- [1] Sucuri Guides' Coporate Auther, "Sucuri," 2021. [Online]. Available: <https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2021/>. [Accessed 11 10 2021].
- [2] AAT TEAM, "All About Teating(AAT)," 21 9 2021. [Online]. Available: <https://allabouttesting.org/information-gathering-techniques-for-penetration-testing/>. [Accessed 13 10 2021].
- [3] Github, "Sublist3r," Github, 29 6 2020. [Online]. Available: <https://github.com/aboul3la/Sublist3r>. [Accessed 13 10 2021].
- [4] Kali linux, "WhatWeb," Kali Linux, [Online]. Available: <https://www.kali.org/tools/whatweb/>. [Accessed 14 10 2021].
- [5] OWASP, "OWASP Cheat Sheet Series," OWASP, 2021. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html#defending-with-x-frame-options-response-headers. [Accessed 15 10 2021].
- [6] MDN contributors, "MDN Web Docs," Mozilla, 13 7 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>. [Accessed 15 10 2021].
- [7] Balbix Corporation, "Balbix," Balbix Corporation, 2020. [Online]. Available: <https://www.balbix.com/insights/what-is-vulnerability-scanning/>. [Accessed 17 10 2021].
- [8] MDN contributors, "MDN Web Docs," Mozilla, 4 10 2021. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>. [Accessed 24 10 2021].
- [9] Netsparker Corporate, "What is Netsparker?," Netsparker, 2021. [Online]. Available: <https://www.netsparker.com/support/what-is-netsparker/>. [Accessed 27 10 2021].
- [10] Edpresso Team, "What is Lodash?," Eduative, [Online]. Available: <https://www.educative.io/edpresso/what-is-lodash>. [Accessed 25 10 2021].
- [11] NGINX Corporation, "What is NGINX?," NGINX, [Online]. Available: <https://www.nginx.com/resources/glossary/nginx/>. [Accessed 26 10 2021].
- [12] S. Gokte, "An intro to OWASP Zed Attack Proxy," Srijan, 15 12 2017. [Online]. Available: <https://www.srijan.net/resources/blog/intro-owasp-zed-attack-proxy#gsekhbrk>. [Accessed 25 10 2021].
- [13] COMPUTER SECURITY RESOURCE CENTER, "Information Technology Laboratory," National Institute of Standards and Technology (NIST), [Online]. Available: https://csrc.nist.gov/glossary/term/penetration_testing. [Accessed 29 10 2021].
- [14] Jimdo's corporation, "A Note About Stored XSS," Hackerone, 11 2020. [Online]. Available: https://hackerone.com/jimdo?type=team&view_policy=true. [Accessed 29 10 2021].
- [15] PortSwigger, "Web Security Academy," PortSwigger, [Online]. Available: <https://portswigger.net/web-security/cross-site-scripting>. [Accessed 29 10 2021].
- [16] Common Weaknesses Enumeration (CWE), "CWE-77: Improper Neutralization of Special

Elements used in a Command ('Command Injection')," Common Weaknesses Enumeration (CWE), 21 7 2021. [Online]. Available: <http://cwe.mitre.org/data/definitions/77.html>. [Accessed 30 10 2021].