



Sri Lanka Institute of Information Technology

Web Application Security Audit Assessment Report

IE2062 – Web Security

Submitted by:

Student Registration Number	Student Name
M. Thushitharan	IT19983370

Date of submission: 01/11/2021

Table of Contents

1. Declaration	3
2. Abstract	3
3. Acknowledgement.....	3
4. Assessment Objectives.....	4
5. Application Credentials and URL	4
6. Assessment Methodology	4
7. OWASP Top 10 Security Risks	5
8. Vulnerability Severity and Examples	6
9. Assessment Scope Details	7
9.1 Target in-scope	7
9.2 Out of scope.....	7
10. Information Gathering.....	8
10.1 Subdomain Enumeration	8
10.2 Gathering Website Archive Information	14
10.3 DNS Enumeration	15
10.4 File Structure Information Finding	16
10.5 Gathering Ports and Running services Information	17
10.6 Gather Firewall protection information.....	18
10.7 Publicly connected services Information	21
11. Vulnerability Analyzing and Reporting Phase	22
11.1 Netsparker Scanning and Review the Bugs	22
11.2 OWASP ZAP Report review.....	37
11.3 Nikto tool Scan	49
11.4 Skipfish tool Scan.....	50
12. Conclusion and Recommendations	52
13. References	52

Video Link:

[https://drive.google.com/file/d/1cfQF5scxL0_6LTa7BKm3mrbTEEwiTIad/view?
usp=sharing](https://drive.google.com/file/d/1cfQF5scxL0_6LTa7BKm3mrbTEEwiTIad/view?usp=sharing)

1. Declaration

I declare that this assignment Report has purely done by me and I mention that this report was not copied from anyone or any institute. And I have used internet to take reference notes, not for copying purpose.

2. Abstract

The project titled Web Application Security Audit is the most ideal approach to guarantee your application is secure before you discharge it and to forestall hacks, harm to notoriety and critical misfortunes to your main concern for application designers.

Find out a web application registered in Hackerone/Bugcrowd and analyzed their vulnerabilities by penetration testing tools. Security testing software' primary function is to perform functional testing of a web application under observance and find as many security issues as possible that could potentially lead to hacking. All of this is done without the need to access the source code. Here some security testing tools used to analyze the vulnerabilities such as Netsparker - web application security, Sublist3r - python tool designed to enumerate subdomains of websites using OSINT, OWASP ZAP - an open-source web application security scanner, Paros - Java based HTTP/HTTPS proxy for assessing web application vulnerability, Golismero - an open-source framework built in Python that can be used for security testing.

This project report will help to decrease the vulnerabilities and get out from the theft of sensitive data in this global technologized world. Many security tests have been carried on, and they are attached and discussed in this document.

3. Acknowledgement

The completion of this Project is not possible without the help and support of many people and entities. Their contributions are sincerely appreciated and gratefully acknowledged. However, the group would like to express their deep appreciation and indebtedness particularly to the following:

Our Lecturers Dr. Lakmal Rupasinghe, Ms. Chethana Liyanapathirana, Ms. Chathu Udagedara for their endless support and understanding spirit during the project development. Some professional bug bounty hunters Ben Sadeghipour ([@nahamsec](#)), STÖK ([@stokfredrik](#)), Vickie Li ([@vickiel7](#)) and Some YouTube Channel contents [HackerSploit](#), [John Hammond](#). I have got many useful ideas from these people and entities to finish this assignment.

4. Assessment Objectives

This web application security audit assignment for <http://netflix.com> is for second semester cyber security students. The objective of this assignment is to measure the capabilities to understand the security measurements of a web application in real world and moreover to understand how to write a vulnerability assessment and penetration testing report.

5. Application Credentials and URL

I have selected Bugcrowd platform to find the domain for this assignment. This assignment is based on root domain, even though I did some scanning to following subdomains

- Netflix.com (**root domain**)
- Help.netflix.com
- Media.netflix.com
- candidate.netflix.com
- brand.netflix.com
- meeclum.netflix.com
- openconnect.netflix.com
- devices.netflix.com
- dvd.netflix.com
- jobs.netflix.com
- partner.netflix.com

6. Assessment methodology



7. OWASP Top 10 Security Risks

OWASP is stands for **Open Web Application Security Project**. This is a non-profit organization which is launched to create a standard for web application security and improve it.

Risk	About it
Injections	These vulnerabilities allow hackers or attackers to inject malicious code in one or many systems. XPath Injection, SQL Injection, LDAP Injection, XML Injection, and Command Injection are some of the common injection vulnerabilities
Broken Authentication	This vulnerability is related to session management and authentication management. Using this vulnerability attackers can compromise passwords, session tokens and keys. Attackers can gain access through manual or automatic methods. This vulnerability allows to do credential stuffing and brute force attacks.
Sensitive Data Exposure	Customers and employees Passwords, Credit card numbers, tax IDs, medical information, and other personal information should be protected by the web application. If attackers gain the access of the web application, they can steal and compromise the data.
XML External Entity attacks (XXE)	If there is a vulnerable XML processor then Attackers can upload malicious xml or malicious content with XML file. Vulnerable code, Dependencies, integrations allows attackers to do XXE attacks.
Cross Site Scripting (XSS)	XSS attacks means injecting malicious client-side scripts into a web application and using the website as a propagation method. It allows the attacker to inject malicious script into a website and modify how it is displayed, forcing a end user browser to execute the code provided by the attacker while loading the page.
Security Misconfigurations	Unpatched flaws, Default configurations, Unused pages, Unprotected files and directories, and Unnecessary services leads to security misconfigurations
Broken Access Control	Access control implements strategy with the end goal that clients can't act outside of their planned authorizations. Disappointments normally lead to unapproved data divulgence, alteration or pulverization of all information, or playing out a business work outside of the restrictions of the client.

Insecure Deserialization	Controlled item is infused into the setting of the web application. In the event that the application is helpless, the item is deserialized and executed, which can bring about SQL Injection, Path Traversal, Application Denial of Service and Remote Code Execution
Using Components with known vulnerabilities	Parts, such as, libraries, structures, and other programming modules, quite often run with full benefits. In the event that a powerless part is misused, such an assault can encourage genuine information misfortune or worker takeover. Applications utilizing parts with realised weaknesses may subvert application safeguards furthermore, empower a scope of potential assaults and effects.
Insufficient logging and monitoring	Hackers depend on the absence of checking and opportune reaction to accomplish their objectives without being identified

Figure 1: Summary of Each OWASP top 10 Risks

8. Vulnerability Severity and Examples

Severity	Few Examples
High	<ul style="list-style-type: none"> • Server Security Misconfiguration • File Inclusion • SQL Injection • Command Injection • Authentication Bypass
Medium	<ul style="list-style-type: none"> • Insecure Direct Object Reference (IDOR) • Cross Site Scripting (XSS) • HTTP Response Manipulation • Content Spoofing
Low	<ul style="list-style-type: none"> • Clickjacking • CAPTCHA Bypass • Insecure SSL • Parameter Pollution • Session Expiration

9. Assessment Scope Details

9.1 Target In scope

api*.netflix.com	API Testing	HTTP
*.prod.ftl.netflix.com		
*.prod.cloud.netflix.com		
*.prod.dradis.netflix.com		
www.netflix.com	ReactJS	jQuery Lodash +10
secure.netflix.com		Website Testing
ichnaea.netflix.com		Website Testing
*.nflxvideo.net		Website Testing
*.nflxext.com		Website Testing
*.nflximg.net		Website Testing
*.nflxso.net		
help.netflix.com	Bootstrap	jQuery Backbone +1
dockhand.netflix.com		Website Testing
beacon.netflix.com		Website Testing
presentationtracking.netflix.com		Website Testing
nmtracking.netflix.com		Website Testing
customerevents.netflix.com		Website Testing
meechum.netflix.com		Website Testing

Figure 2: In scope sections listed in Bugcrowd Netflix page

9.2 Target Out of Scope

- Third Party websites hosted by non-Netflix domains
- Ir.netflix.com
- Netinvestor.com
- Netflix Client Application

10. Information Gathering (Recon) Phase

Gathering Information is an important part of penetration testing. It gives a huge amount of information as much as possible from the target web application. This phase also called as reconnaissance phase.

10.1 Finding Subdomains

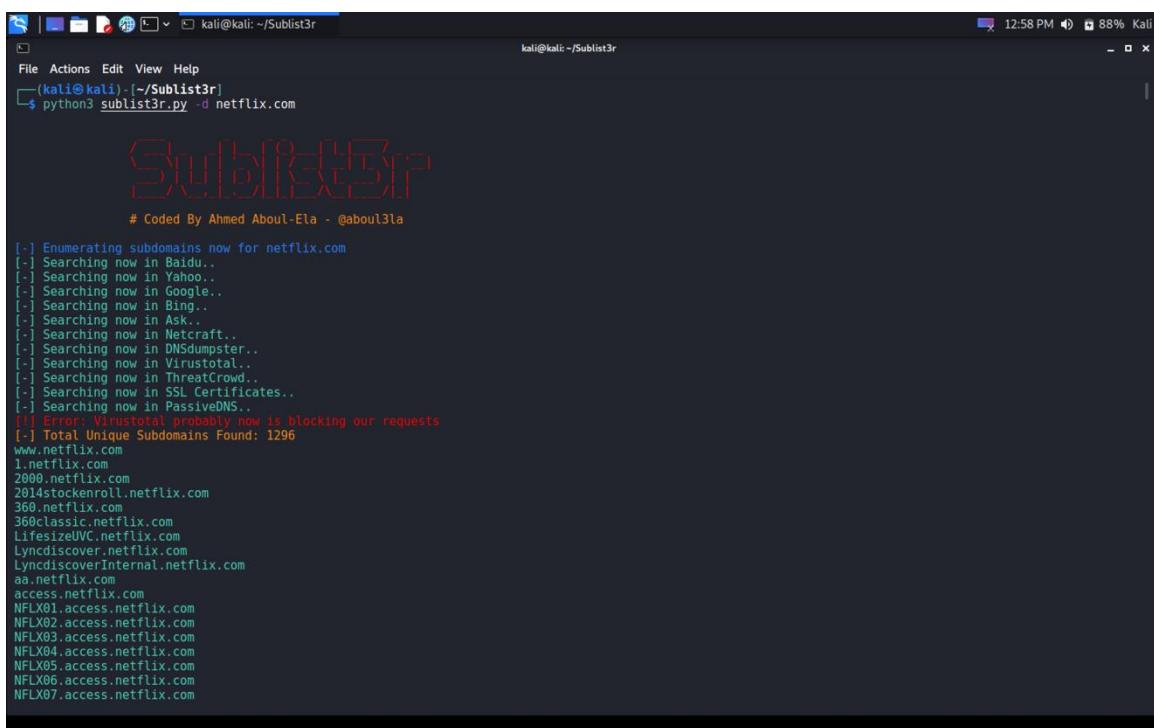
Sublist3r – Python Script

Github Repository Link: <https://github.com/aboul3la/Sublist3r>

Usage:

```
python3 sublist3r.py -d netflix.com
```

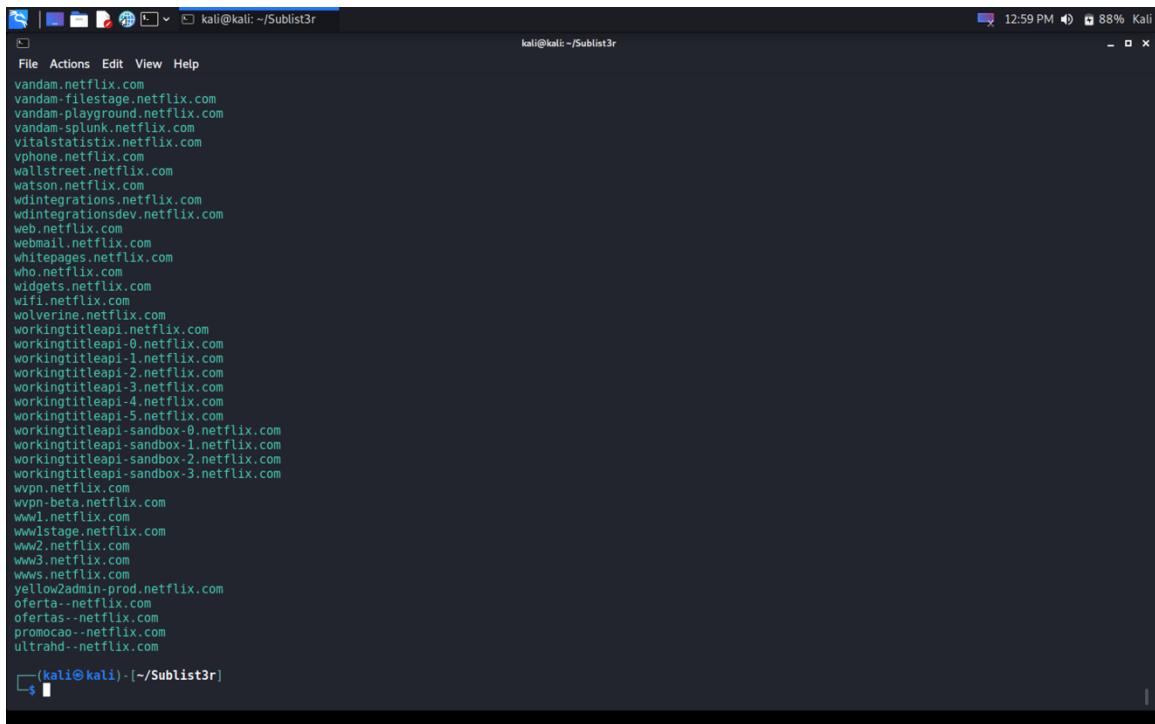
Scan Results: *This tool showed that 1296 sub domains found from this website.*



```
kali@kali: ~/Sublist3r
File Actions Edit View Help
(kali㉿kali)-~/Sublist3r
$ python3 sublist3r.py -d netflix.com

Sublist3r
# Coded By Ahmed Aboul-Ela - @aboul3la

[-] Enumerating subdomains now for netflix.com
[-] Searching now in Baidu..
[-] Searching now in Yahoo..
[-] Searching now in Google..
[-] Searching now in Bing..
[-] Searching now in Ask..
[-] Searching now in Netcraft..
[-] Searching now in DNSdumpster..
[-] Searching now in Virustotal..
[-] Searching now in ThreatCrowd..
[-] Searching now in SSL Certificates..
[-] Searching now in PassiveDNS..
[!] Error: Virustotal probably now is blocking our requests
[-] Total Unique Subdomains Found: 1296
www.netflix.com
1.netflix.com
2000.netflix.com
2014stockenroll.netflix.com
360.netflix.com
360classic.netflix.com
LifesizeUVC.netflix.com
Lyncdiscover.netflix.com
LyncdiscoverInternal.netflix.com
aa.netflix.com
access.netflix.com
NFLX01.access.netflix.com
NFLX02.access.netflix.com
NFLX03.access.netflix.com
NFLX04.access.netflix.com
NFLX05.access.netflix.com
NFLX06.access.netflix.com
NFLX07.access.netflix.com
```



A screenshot of a terminal window titled "kali@kali: ~/Sublist3r". The window shows a list of domain names, likely generated by the Sublist3r tool. The domains listed include various Netflix-related subdomains such as vandam.netflix.com, wdintegrations.netflix.com, web.netflix.com, webmail.netflix.com, whitepages.netflix.com, who.netflix.com, widgets.netflix.com, wifi.netflix.com, wolverine.netflix.com, workingtitleapi.netflix.com, workingtitleapi-0.netflix.com, workingtitleapi-1.netflix.com, workingtitleapi-2.netflix.com, workingtitleapi-3.netflix.com, workingtitleapi-4.netflix.com, workingtitleapi-5.netflix.com, workingtitleapi-sandbox-0.netflix.com, workingtitleapi-sandbox-1.netflix.com, workingtitleapi-sandbox-2.netflix.com, workingtitleapi-sandbox-3.netflix.com, vpn.netflix.com, vpn-beta.netflix.com, www1.netflix.com, www1stage.netflix.com, www2.netflix.com, www3.netflix.com, www.netflix.com, yellow2admin-prod.netflix.com, ofertas--netflix.com, ofertas-.netflix.com, promocao--netflix.com, ultrahd--netflix.com. The terminal window has a dark background and light-colored text. The title bar and status bar at the top right indicate the session is running on a Kali Linux system at 12:59 PM with 88% battery.

```
vandam.netflix.com
vandam-filestage.netflix.com
vandam-playground.netflix.com
vandam-splunk.netflix.com
vitalstatistix.netflix.com
vphone.netflix.com
wallstreet.netflix.com
watson.netflix.com
wdintegrations.netflix.com
web.netflix.com
webmail.netflix.com
whitepages.netflix.com
who.netflix.com
widgets.netflix.com
wifi.netflix.com
wolverine.netflix.com
workingtitleapi.netflix.com
workingtitleapi-0.netflix.com
workingtitleapi-1.netflix.com
workingtitleapi-2.netflix.com
workingtitleapi-3.netflix.com
workingtitleapi-4.netflix.com
workingtitleapi-5.netflix.com
workingtitleapi-sandbox-0.netflix.com
workingtitleapi-sandbox-1.netflix.com
workingtitleapi-sandbox-2.netflix.com
workingtitleapi-sandbox-3.netflix.com
vpn.netflix.com
vpn-beta.netflix.com
www1.netflix.com
www1stage.netflix.com
www2.netflix.com
www3.netflix.com
www.netflix.com
yellow2admin-prod.netflix.com
ofertas--netflix.com
ofertas-.netflix.com
promocao--netflix.com
ultrahd--netflix.com
```

Knock – Python Script

Github Repository Link: <https://github.com/guelfoweb/knock>

Usage:

```
python3 knockpy.py netflix.com
```

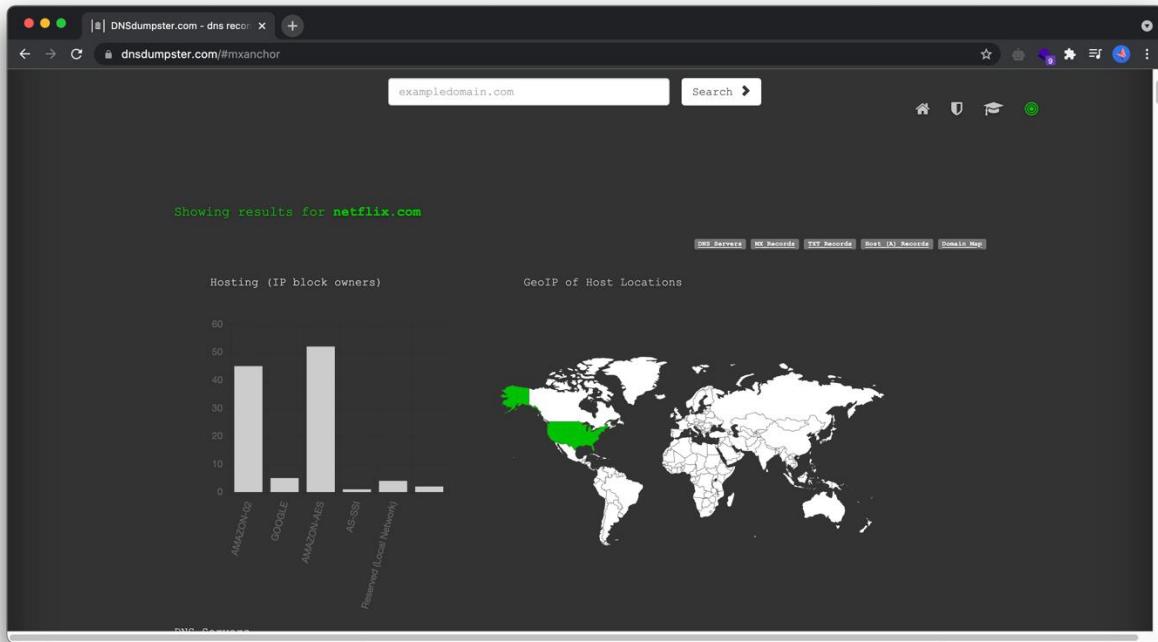
Scan Results: *This subdomain finder shows active subdomains and its IP address, status code, hosted server, and real hostname of its.*

Ip address	Code	Subdomain	Server	Real hostname
34.205.97.36	200	about.netflix.com	Apache	pubpresssite-2140919448.us-east-1.elb.amazonaws.com
52.38.7.83	200	account.netflix.com	nq_website_nonmember-prod-r	dualstack.apiproxy-website-nlb-prod-1-bcf28d21f4bbcf2c.elb.us-west-2.amazonaws.com
52.42.41.159	404	api.netflix.com	api-prod-1-06e79382098793b5	api.us-west-2.origin.prodaa.netflix.com
52.39.7.83	200	app.netflix.com	nq_website_nonmember-prod-r	dualstack.apiproxy-website-nlb-prod-1-bcf28d21f4bbcf2c.elb.us-west-2.amazonaws.com
52.31.48.193		athena.netflix.com		detour.prod.netflix.net
207.45.73.144		b2b.netflix.com		
52.31.48.193	404	blog.netflix.com	Apache	detour.prod.netflix.net
44.236.157.89	200	brand.netflix.com	nginx	brand-prod-external-1837059155.us-west-2.elb.amazonaws.com
104.18.119.155	200	cache.netflix.com	cloudflare	netflix-sites.cloudcannon.com
172.217.194.121	200	calendar.netflix.com		ghs.googlehosted.com
52.39.128.98		catalog.netflix.com		dualstack.apiproxy-catalog-vpc0-1424560578.us-west-2.elb.amazonaws.com
104.75.84.11	404	cdn.netflix.com	AkamaiNetStorage	a1386.g.akamai.net
204.236.236.127	200	contact.netflix.com	poe-secure-1-092f8055e87dc	api-prod-1-093f9c45dd744989
44.234.232.238	404	css.netflix.com	api-prod-1-093f9c45dd744989	dualstack.apiproxy-website-nlb-prod-2-e98cb8cf33ff3581.elb.us-west-2.amazonaws.com
207.210.238.73		delta.netflix.com		
34.252.74.1	200	developer.netflix.com	nq_website_nonmember-prod-r	detour.prod.netflix.net
142.251.10.121	200	drive.netflix.com		ghs.googlehosted.com
207.45.72.201	200	dvd.netflix.com		
74.125.200.121	200	email.netflix.com		ghs.googlehosted.com
142.251.10.121	200	employees.netflix.com	GSE	ghs.googlehosted.com
18.232.2.164	200	g.netflix.com	GSE	go2-meechum.prod.netflix.net
10.45.64.1		gateway.netflix.com		
18.232.2.164	200	go.netflix.com	GSE	go2-meechum.prod.netflix.net
142.251.10.121	200	groups.netflix.com	GSE	ghs.googlehosted.com
44.226.110.19	200	help.netflix.com	poe-secure-1-049eca6259ff23	dualstack.padme-uswest-vpc0-735221756.us-west-2.elb.amazonaws.com
104.75.84.10	404	image.netflix.com	AkamaiNetStorage	a743.g.akamai.net
104.75.84.9	404	images.netflix.com	AkamaiNetStorage	a743.g.akamai.net
44.240.158.19	404	info.netflix.com	api-prod-1-07bc7762eb0fc710	dualstack.apiproxy-website-nlb-prod-1-bcf28d21f4bbcf2c.elb.us-west-2.amazonaws.com
46.137.171.215	200	ir.netflix.com	cloudflare	detour.prod.netflix.net
100.81.15.115		jira.netflix.com		corpiraprod.mgmt.netflix.net
54.149.198.120	200	jobs.netflix.com	atsjobsiteui-448e70f1-e013	api-proxy-lambda-atsjobsiteui-v2-392483922.us-west-2.elb.amazonaws.com
100.127.235.114		mail.netflix.com		internal-vpc0-mailrelay-26877350.us-west-2.elb.amazonaws.com
172.26.1.8		manage.netflix.com		
54.156.127.29	200	media.netflix.com	Apache	pubmediacenter-site-64305187.us-east-1.elb.amazonaws.com
46.137.171.215	200	movies.netflix.com	nq_website_nonmember-prod-r	detour.prod.netflix.net
52.3.77.81		partner.netflix.com		
172.26.2.250		pix.netflix.com		
46.137.171.215	200	pr.netflix.com	Apache	detour.prod.netflix.net
52.297.189.14	200	research.netflix.com	researchml-1e64ea7a-b4f6-40	apiproxy-research-767688609.us-east-1.elb.amazonaws.com
45.57.90.1	403	secure.netflix.com	nginx	sec-oc.netflix.com
91.235.133.193	400	secured.netflix.com	Apache	h-netflix.online-metrinet.net
44.240.158.19	404	service.netflix.com	api-prod-1-0b6bb59b77e253f1	dualstack.apiproxy-website-nlb-prod-1-bcf28d21f4bbcf2c.elb.us-west-2.amazonaws.com
34.252.74.1	200	shop.netflix.com	cloudflare	detour.prod.netflix.net
52.41.95.74	200	signup.netflix.com	nq_website_nonmember-prod-r	dualstack.ecweb-prod-vpc0-2058625167.us-west-2.elb.amazonaws.com
69.53.236.55		static.netflix.com		
52.31.48.193	200	tv.netflix.com	nq_website_nonmember-prod-r	detour.prod.netflix.net
104.18.118.155	409	updates.netflix.com	cloudflare	netflix-sites.cloudcannon.com
216.35.131.141		vpn.netflix.com		
44.242.13.161	200	www.netflix.com	nq_website_nonmember-prod-r	dualstack.apiproxy-website-nlb-prod-1-bcf28d21f4bbcf2c.elb.us-west-2.amazonaws.com
34.194.141.126		www1.netflix.com		
34.213.69.2		www2.netflix.com		
34.216.233.138		www3.netflix.com		
106.82.14.115		yellow.netflix.com		

DNSdumpster – Online Tool

Link: <https://dnsdumpster.com/>

Scan Results: *This tool categorized each subdomain in to MX Records, Host (A) Records and DNS Servers. Moreover, it generated a domain map which is showing interconnected subdomains and dns servers.*



The figure shows the DNSdumpster interface for the domain netflix.com, displaying detailed results. The 'DNS Servers' section lists several nameservers with their IP addresses and locations: ns-1372.awsdns-43.org (205.251.197.92, AMAZON-02, United States), ns-1984.awsdns-56.co.uk (205.251.199.192, AMAZON-02, United States), ns-659.awsdns-18.net (205.251.194.147, AMAZON-02, United States), and ns-81.awsdns-10.com (205.251.192.81, AMAZON-02, United States). The 'MX Records' section lists email delivery points: aspmx1.google.com (142.250.123.27, GOOGLE, United States), aspmx2.googlemail.com (108.177.12.26, GOOGLE, United States), aspmx3.googlemail.com (64.233.186.26, GOOGLE, United States), alt1.aspmx1.google.com (108.177.12.27, GOOGLE, United States), and alt2.aspmx1.google.com (64.233.186.27, GOOGLE, United States). The 'TXT Records' section shows SPF configurations: "8cd468d7d5994ecc93d350683a8cb07a1" and "docsigmae249396f-e8150-48#2-8bd2-705be6e03826".

DNS Servers	IP Address	Location
ns-1372.awsdns-43.org.	205.251.197.92	AMAZON-02 United States
ns-1984.awsdns-56.co.uk.	205.251.199.192	AMAZON-02 United States
ns-659.awsdns-18.net.	205.251.194.147	AMAZON-02 United States
ns-81.awsdns-10.com.	205.251.192.81	AMAZON-02 United States

MX Records ** This is where email for the domain goes...		
1 aspmx1.google.com.	142.250.123.27	GOOGLE United States
10 aspmx2.googlemail.com.	108.177.12.26	GOOGLE United States
10 aspmx3.googlemail.com.	64.233.186.26	GOOGLE United States
5 alt1.aspmx1.google.com.	108.177.12.27	GOOGLE United States
5 alt2.aspmx1.google.com.	64.233.186.27	GOOGLE United States

TXT Records ** Find more hosts in Sender Policy Framework (SPF) configurations		
"8cd468d7d5994ecc93d350683a8cb07a1"		
"docsigmae249396f-e8150-48#2-8bd2-705be6e03826"		

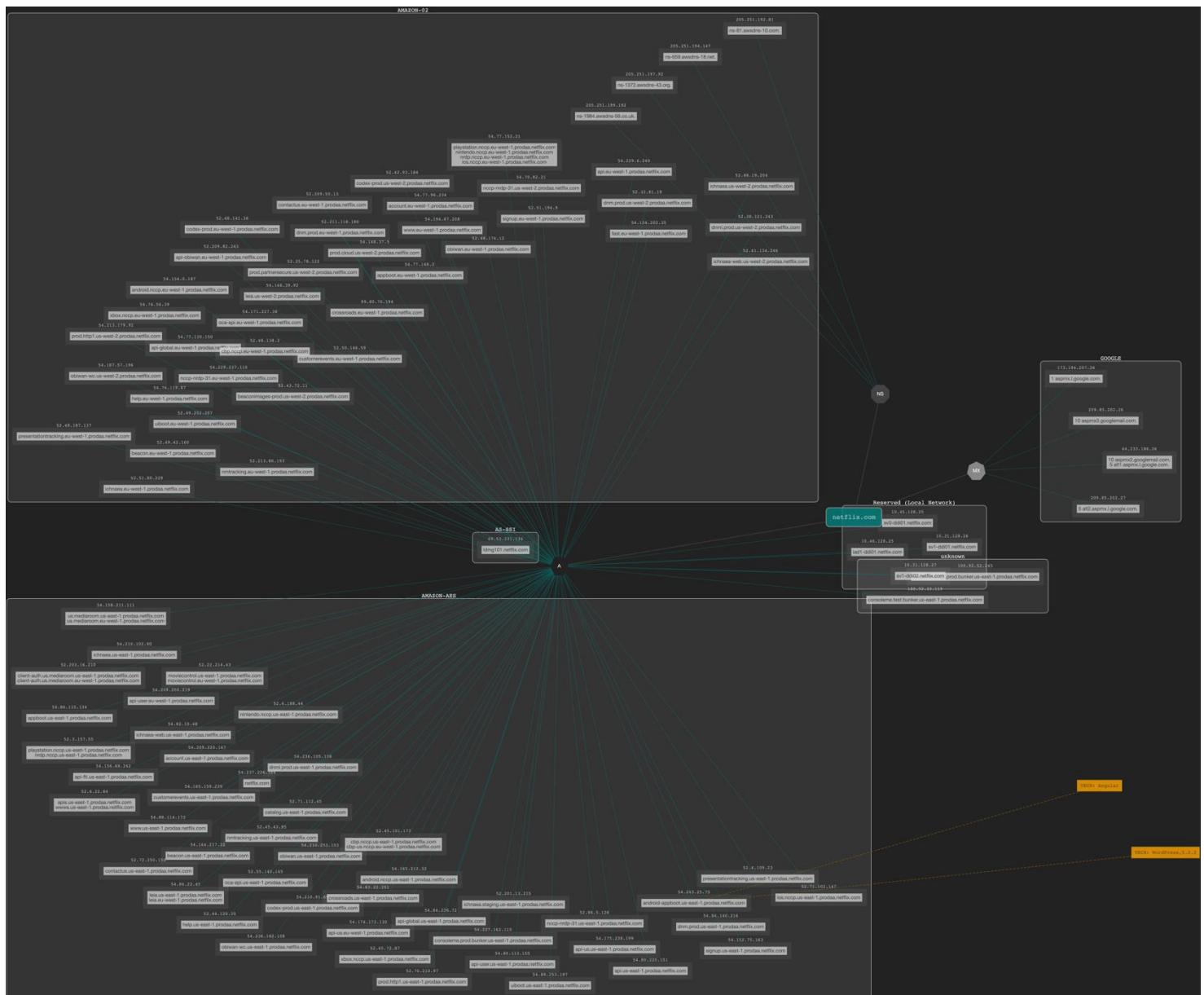


Figure 2: In scope sections listed in Bugcrowd Netflix page

Crt.sh – Online Tool

Link: <https://crt.sh/>

Scan Results:

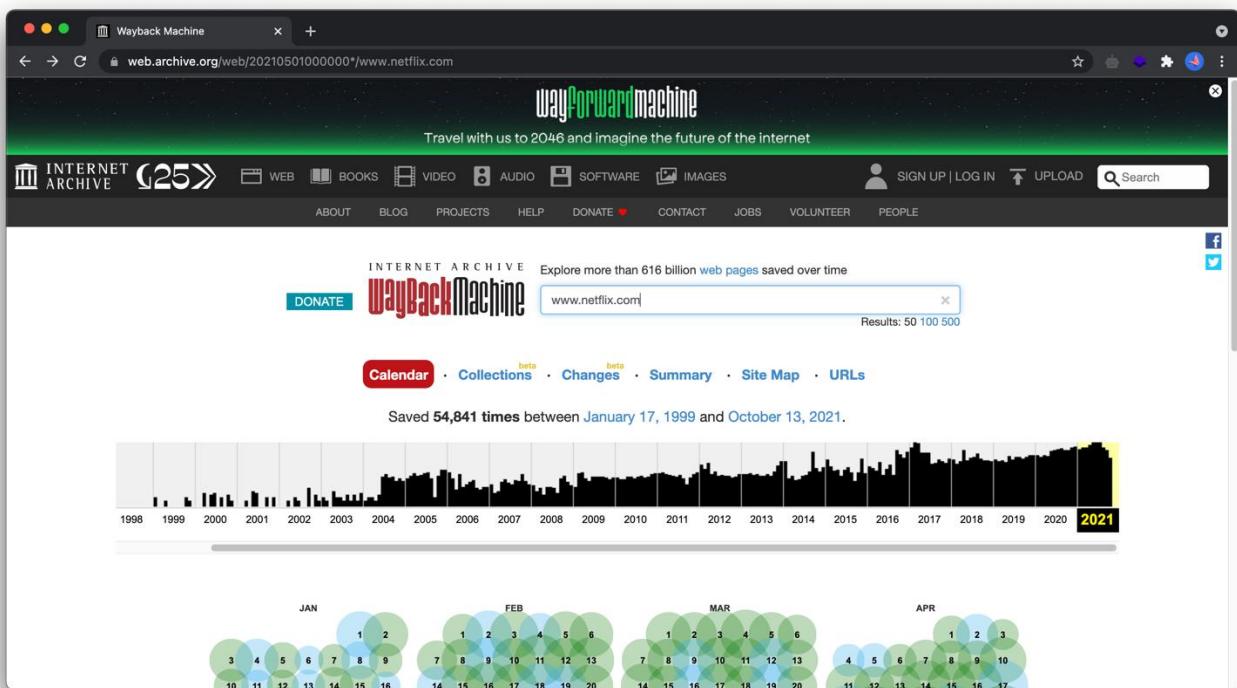
Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	2382744470	2020-01-27	2017-01-31	2017-03-02	*.nfixso.net	ftl.netflix.com	C=US,O=Symantec Corporation,OU=Symantec Trust Network,CN=Symantec Class 3 Secure Server CA - G4
	2382730258	2020-01-27	2017-01-03	2017-02-02	*.nfixso.net	ftl.netflix.com	C=US,O=Symantec Corporation,OU=Symantec Trust Network,CN=Symantec Class 3 Secure Server CA - G4
	2380425321	2020-01-26	2010-11-16	2011-11-19	nccp-ironman.netflix.com	nccp-ironman.netflix.com	C=US,GeoTrust Inc.,OU=Domain Validated SSL,CN=GeoTrust DV SSL CA
	2380107839	2020-01-26	2009-09-01	2010-09-11	widgets.netflix.com	Netflix.com widgets.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA - G2
	2380067278	2020-01-26	2009-07-13	2010-08-09	airmail.netflix.com	airmail.netflix.com Netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA - G2
	2380017538	2020-01-26	2009-06-09	2010-06-15	ftp.netflix.com	ftp.netflix.com Netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA - G2
	2380004422	2020-01-26	2009-07-11	2010-07-17	usairways.netflix.com	Netflix.com usairways.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA - G2
	2379979803	2020-01-26	2009-09-01	2010-09-14	wvpn.netflix.com	Netflix.com wvpn.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA - G2
	2374202746	2020-01-25	2010-02-01	2011-02-03	nrd.netflix.com	nrd.netflix.com	C=US,O=Equifax,OU=Equifax Secure Certificate Authority
	2372644572	2020-01-24	2009-03-25	2010-03-27	research.netflix.com	Netflix.com research.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)09,CN=VeriSign Class 3 Secure Server CA
	2372626369	2020-01-24	2008-12-01	2009-12-01	pqr.netflix.com	Netflix.com pqr.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)05,CN=VeriSign Class 3 Secure Server CA
	2372625850	2020-01-24	2008-01-18	2010-01-22	delta.netflix.com	delta.netflix.com Netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)05,CN=VeriSign Class 3 Secure Server CA
	2372587502	2020-01-24	2009-02-04	2010-03-06	image.netflix.com	image.netflix.com Netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)05,CN=VeriSign Class 3 Secure Server CA
	2372588008	2020-01-24	2009-02-20	2010-03-05	www.netflix.com	Netflix.com www.netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)05,CN=VeriSign Class 3 Secure Server CA
	2372574817	2020-01-24	2008-08-29	2009-09-14	wvpn.netflix.com	Netflix.com	C=US,O=VeriSign,Inc.,OU=VeriSign Trust Network,OU=Terms of use at https://www.verisign.com/pa(c)05,CN=VeriSign Class 3 Secure Server CA

10.2 Website Archive Information

Wayback Machine

Link: <https://archive.org/web/>

Search Result: *This tool is used to view the website archive information from the beginning state. Here the Netflix.com website archive data information has been displayed from 1999 to 2021.*



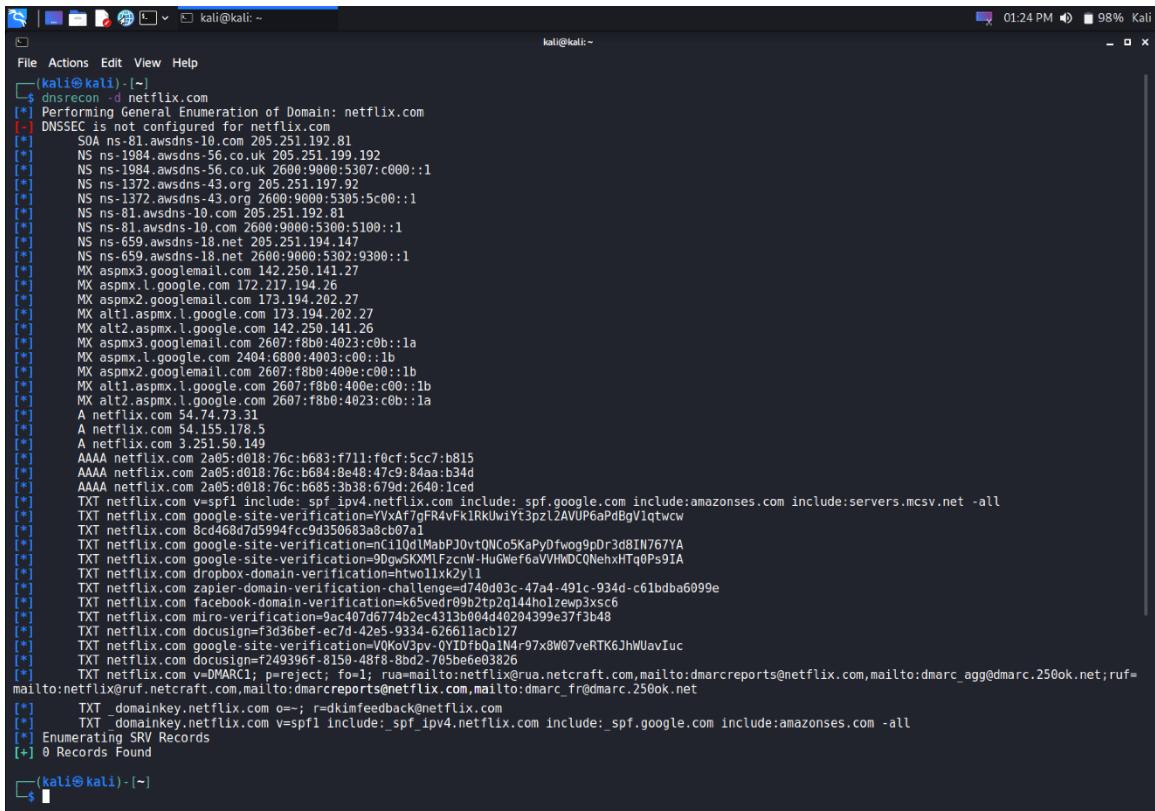
10.3 DNS Enumeration

dnsrecon – Kali Linux inbuilt Tool

Usage:

```
dnsrecon -d netflix.com
```

Scan Results: *Enumerate all DNS records of a web application.*



```
(kali㉿kali)-[~] $ dnsrecon -d netflix.com
[*] Performing General Enumeration of Domain: netflix.com
[+] DNSSEC is not configured for netflix.com
[*] SOA ns-81.awsdns-10.com 205.251.192.81
[*] NS ns-1984.awsdns-56.co.uk 205.251.199.192
[*] NS ns-1984.awsdns-56.co.uk 2600:9000:5307:c000::1
[*] NS ns-1372.awsdns-43.org 205.251.197.92
[*] NS ns-1372.awsdns-43.org 2600:9000:5305:5c00::1
[*] NS ns-81.awsdns-10.com 205.251.192.81
[*] NS ns-81.awsdns-10.com 2600:9000:5300:5100::1
[*] NS ns-659.awsdns-18.net 205.251.194.147
[*] NS ns-659.awsdns-18.net 2600:9000:5302:9300::1
[*] MX aspmx3.googlemail.com 142.250.141.27
[*] MX aspmx.l.google.com 172.21.194.26
[*] MX aspmx2.googlemail.com 173.194.202.27
[*] MX alt1.aspmx.l.google.com 173.194.202.27
[*] MX alt2.aspmx.l.google.com 142.250.141.26
[*] MX aspmx3.googlemail.com 2607:f8b0:4023:c0b::1a
[*] MX aspmx.l.google.com 2404:6800:4003:c00::1b
[*] MX aspmx2.googlemail.com 2607:f8b0:400e:c00::1b
[*] MX alt1.aspmx.l.google.com 2607:f8b0:400e:c00::1b
[*] MX alt2.aspmx.l.google.com 2607:f8b0:4023:c0b::1a
[*] A netflix.com 54.74.73.31
[*] A netflix.com 54.155.178.5
[*] A netflix.com 3.251.50.149
[*] AAAA netflix.com 2a05:d018:76c:b683:f711:f0cf:5cc7:b815
[*] AAAA netflix.com 2a05:d018:76c:b684:8e48:47c9:84aa:b34d
[*] AAAA netflix.com 2a05:d018:76c:b685:3b38:679d:2640:1ced
[*] TXT netflix.com v=spf1 include:_spf4.netflix.com include:_spf.google.com include:amazonse.com include:servers.mcsv.net -all
[*] TXT netflix.com google-site-verification=YXAf7qFR4vFk1RkUwiYt3pzl2AVUP6aPdBgV1qtWcw
[*] TXT netflix.com 8cd468d7d5994fc9d35063a38cb07a1
[*] TXT netflix.com google-site-verification=nci0q1MapbJ0vTNCo5KaPyfwog9pDr3d81N767YA
[*] TXT netflix.com google-site-verification=99qy5KXMUfzcnW-HuGWeF6aVHMDCQNehxHtg0Ps97IA
[*] TXT netflix.com dropbox-domain-verification=htwolk2y1l
[*] TXT netflix.com zapiier-domain-verification=d740d03c-47a4-491c-934d-c61bdba6099e
[*] TXT netflix.com facebook-domain-verification=k65ver89b2tp2q144holzep3xsc6
[*] TXT netflix.com miro-verification=9ac407d6774b2ec431b804d40204399e37f7b48
[*] TXT netflix.com docusign=f3d3b6nef-ec7d-42e5-9334-62661acbd127
[*] TXT netflix.com google-site-verification=VKoV3pv-QYIDfbQaIN4r97x8W07veRTK6JhWUavIuc
[*] TXT netflix.com docusign=f249396f-8150-48f8-8bd2-795be6e83826
[*] TXT netflix.com v=DMARC; p=reject; fo=1; rua=mailto:netflix@ua.netcraft.com.mailto:dmarcreports@netflix.com.mailto:dmarc_ag@dmarc.250ok.net;ruf=
mailto:netflix@uf.netcraft.com.mailto:dmarcreports@netflix.com.mailto:dmarc_fr@dmarc.250ok.net
[*]     TXT _domainkey.netflix.com o=-; r=dkimfeedback@netflix.com
[*]     TXT _domainkey.netflix.com v=spf1 include:_spf4.netflix.com include:_spf.google.com include:amazonse.com -all
[*] Enumerating SRV Records
[+] 0 Records Found
```

10.4 File structure Enumeration

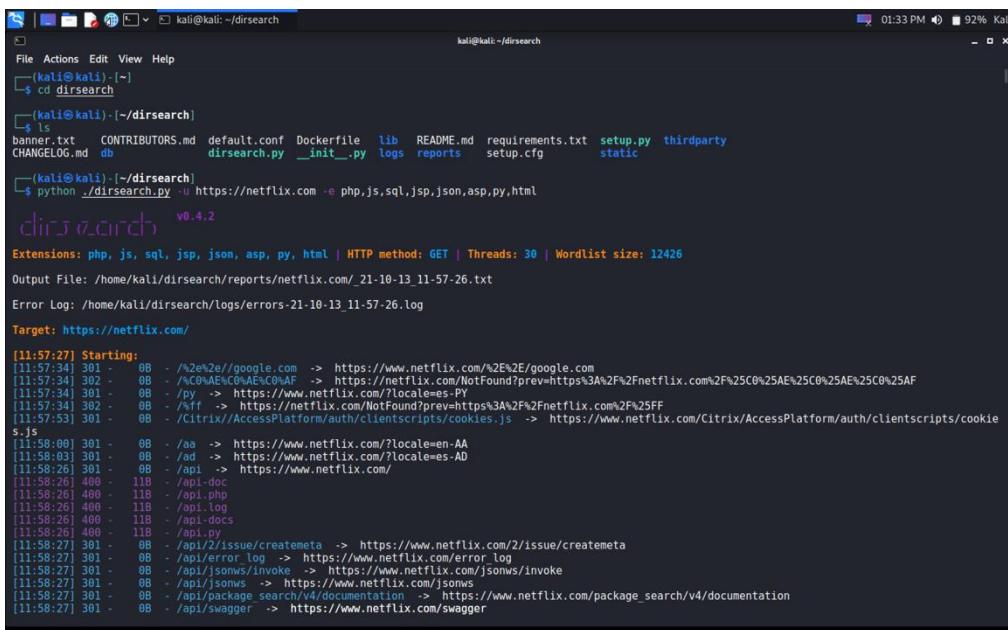
Dirsearch - Python Script

Github Repository Link: <https://github.com/maurosoria/dirsearch>

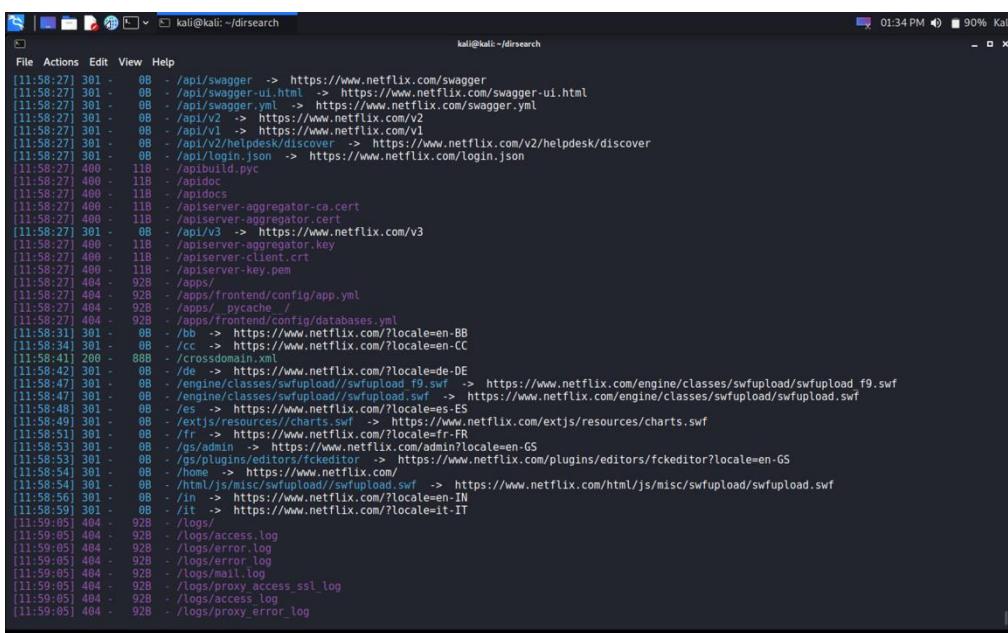
Usage:

```
python dirsearch.py -u https://netflix.com -e php,js,sql,jsp,json,asp,py,html
```

Scan Result: *This is used to brute force directories and files in webserver.*



```
kali@kali:~/dirsearch
File Actions Edit View Help
(kali㉿kali)-[~]
$ cd dirsearch
(kali㉿kali)-[~/dirsearch]
└─$ python ./dirsearch.py -u https://netflix.com -e php,js,sql,jsp,json,asp,py,html
v0.4.2
Extensions: php, js, sql, jsp, json, asp, py, html | HTTP method: GET | Threads: 30 | Wordlist size: 12426
Output File: /home/kali/dirsearch/reports/netflix.com_21-10-13_11-57-26.log
Error Log: /home/kali/dirsearch/logs/errors-21-10-13_11-57-26.log
Target: https://netflix.com/
[11:57:27] Starting:
[[11:57:34] 301 - 0B - /%2E/google.com -> https://www.netflix.com/%2E/google.com
[[11:57:34] 301 - 0B - /C%AE%0A%F%0A%F -> https://www.netflix.com/NotFound?prev=https%3A%2F%2Fnetflix.com%2F%25C0%25AE%25C0%25AE%25C0%25AF
[[11:57:34] 301 - 0B - /%2F -> https://www.netflix.com/NotFound?prev=https%3A%2F%2Fnetflix.com%2F%25FF
[[11:57:34] 302 - 0B - /%2F -> https://www.netflix.com/NotFound?prev=https%3A%2F%2Fnetflix.com%2F%25FF
[[11:57:33] 301 - 0B - /Citrix//AccessPlatform/auth/clientscripts/cookies.js -> https://www.netflix.com/Citrix//AccessPlatform/auth/clientscripts/cookie
s.js
[[11:58:00] 301 - 0B - ./aa -> https://www.netflix.com/?locale=en_AA
[[11:58:03] 301 - 0B - ./ad -> https://www.netflix.com/?locale=es_AD
[[11:58:26] 301 - 0B - ./api -> https://www.netflix.com/
[[11:58:26] 408 - 11B - ./api-doc
[[11:58:27] 301 - 0B - ./api/2/issue/createmeta -> https://www.netflix.com/2/issue/createmeta
[[11:58:27] 301 - 0B - ./api/error.log -> https://www.netflix.com/error.log
[[11:58:27] 301 - 0B - ./api/jsonws/invoke -> https://www.netflix.com/jsonws/invoke
[[11:58:27] 301 - 0B - ./api/jsonws -> https://www.netflix.com/jsonws
[[11:58:27] 301 - 0B - ./api/package_search/v4/documentation -> https://www.netflix.com/package_search/v4/documentation
[[11:58:27] 301 - 0B - ./api/swagger -> https://www.netflix.com/swagger
```



```
kali@kali:~/dirsearch
File Actions Edit View Help
[[11:58:27] 301 - 0B - ./api/swagger -> https://www.netflix.com/swagger
[[11:58:27] 301 - 0B - ./api/swagger-ui.html -> https://www.netflix.com/swagger-ui.html
[[11:58:27] 301 - 0B - ./api/swagger.yml -> https://www.netflix.com/swagger.yml
[[11:58:27] 301 - 0B - ./api/v2 -> https://www.netflix.com/v2
[[11:58:27] 301 - 0B - ./api/v1 -> https://www.netflix.com/v1
[[11:58:27] 301 - 0B - ./api/v2/helpdesk/discover -> https://www.netflix.com/v2/helpdesk/discover
[[11:58:27] 301 - 0B - ./api/login.json -> https://www.netflix.com/login.json
[[11:58:27] 408 - 11B - ./api/build0.pyc
[[11:58:27] 408 - 11B - ./api/docs
[[11:58:27] 408 - 11B - ./apiserver-aggregator-ca.cert
[[11:58:27] 408 - 11B - ./apiserver-aggregator.cert
[[11:58:27] 301 - 0B - ./api/v3 -> https://www.netflix.com/v3
[[11:58:27] 400 - 11B - ./apiserver-aggregator.key
[[11:58:27] 400 - 11B - ./apiserver-client.crt
[[11:58:27] 400 - 11B - ./apiserver-key.pem
[[11:58:27] 404 - 92B - ./apps/
[[11:58:27] 404 - 92B - ./apps/frontend/config/app.yml
[[11:58:27] 404 - 92B - ./apps/frontend/config/databases.yml
[[11:58:31] 301 - 0B - ./bb -> https://www.netflix.com/?locale=en_BB
[[11:58:34] 301 - 0B - ./cc -> https://www.netflix.com/?locale=en_CC
[[11:58:41] 200 - 88B - ./crossdomain.xml
[[11:58:42] 301 - 0B - ./de -> https://www.netflix.com/?locale=de_DE
[[11:58:47] 301 - 0B - ./engine/classes/swfupload/swfupload.f9.swf -> https://www.netflix.com/engine/classes/swfupload/swfupload.f9.swf
[[11:58:47] 301 - 0B - ./engine/classes/swfupload/swfupload.swf -> https://www.netflix.com/engine/classes/swfupload/swfupload.swf
[[11:58:48] 301 - 0B - ./es -> https://www.netflix.com/?locale=es_ES
[[11:58:50] 301 - 0B - ./extjs/resources/charts.swf -> https://www.netflix.com/extjs/resources/charts.swf
[[11:59:51] 301 - 0B - ./fr -> https://www.netflix.com/?locale=fr_FR
[[11:59:53] 301 - 0B - ./gs/admin -> https://www.netflix.com/admin?locale=en_GS
[[11:59:53] 301 - 0B - ./gs/plugins/editors/fckeditor -> https://www.netflix.com/plugins/editors/fckeditor?locale=en_GS
[[11:59:54] 301 - 0B - ./home -> https://www.netflix.com/
[[11:59:56] 301 - 0B - ./in -> https://www.netflix.com/?locale=en_IN
[[11:59:59] 301 - 0B - ./it -> https://www.netflix.com/?locale=it_IT
[[11:59:05] 404 - 92B - ./logs/
[[11:59:05] 404 - 92B - ./logs/access.log
[[11:59:05] 404 - 92B - ./logs/error.log
[[11:59:05] 404 - 92B - ./logs/error_log
[[11:59:05] 404 - 92B - ./logs/mail.log
[[11:59:05] 404 - 92B - ./logs/proxy_access_ssl_log
[[11:59:05] 404 - 92B - ./logs/access.log
[[11:59:05] 404 - 92B - ./logs/proxy_error_log
```

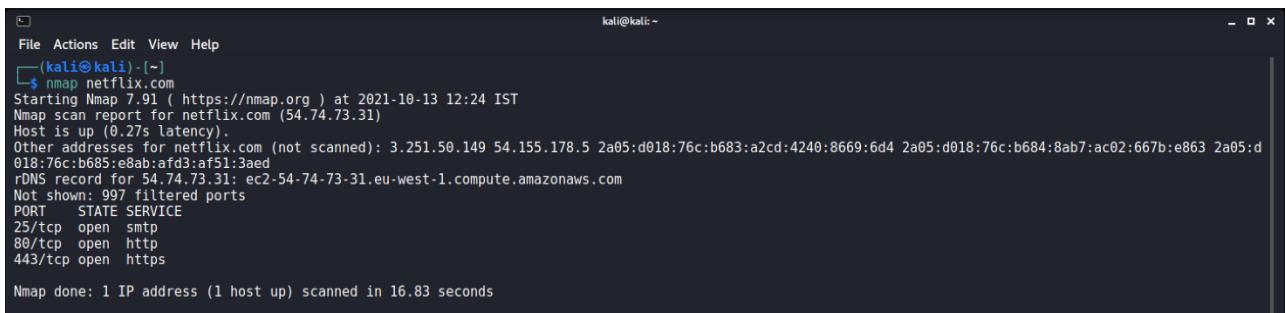
10.5 Gather ports and running services information

Nmap

Usage:

```
nmap netflix.com
```

Scan Result: *This scan returned port and running services information of the website.*



A screenshot of a terminal window titled "kali@kali: ~". The window shows the command \$ nmap netflix.com and its output. The output includes the start time, host status, other addresses, rDNS record, filtered ports, and open ports (25/tcp, 80/tcp, 443/tcp) with their respective services (smtp, http, https). The scan took 16.83 seconds.

```
File Actions Edit View Help
[kali㉿kali:~]
$ nmap netflix.com
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-13 12:24 IST
Nmap scan report for netflix.com (54.74.73.31)
Host is up (0.27s latency).
Other addresses for netflix.com (not scanned): 3.251.50.149 54.155.178.5 2a05:d018:76c:b683:a2cd:4240:8669:6d4 2a05:d018:76c:b684:8ab7:ac02:667b:e863 2a05:d018:76c:b685:e8ab:af3:a51:3aed
rDNS record for 54.74.73.31: ec2-54-74-73-31.eu-west-1.compute.amazonaws.com
Not shown: 997 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 16.83 seconds
```

10.6 Gather Firewall protection Information

Wafwoof

Github Repository Link: <https://github.com/EnableSecurity/wafw00f>

Usage:

```
wafw00f https://netflix.com
```

Scan Results: *This tool returned firewall protection information of the domain. So, I did scan for the selected subdomains.*

Netflix.com firewall protection details

```
File Actions Edit View Help
[kali㉿kali:~] $ wafw00f https://netflix.com


404 Hack Not Found
405 Not Allowed
403 Forbidden
502 Bad Gateway
500 Internal Error

~ WAFW00F : v2.1.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://netflix.com
[+] Generic Detection results:
[*] The site https://netflix.com seems to be behind a WAF or some sort of security solution
[-] Reason: The server header is different when an attack is detected.
The server header for a normal response is "nd_website_nnonmember-prod-release 7ad0f63c-841f-4459-a47c-461b62b2bd8e", while the server header a response to a
n attack is "nd_website_nnonmember-prod-release f003d252-270f-405f-bffc-55dfa6a7c872",
[-] Number of requests: 7
```

Partnet.netflix.com firewall protection details

```
File Actions Edit View Help
[kali㉿kali:~] $ wafw00f https://partner.netflix.com


404 Hack Not Found
405 Not Allowed
403 Forbidden
502 Bad Gateway
500 Internal Error

~ WAFW00F : v2.1.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://partner.netflix.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[-] Number of requests: 7
```

Jobs.netflix.com firewall protection details

```
(kali㉿kali)-[~]
└─$ wafw00f https://jobs.netflix.com


~ WAFW00F : v2.1.0 -
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://jobs.netflix.com
[+] Generic Detection results:
[*] The site https://jobs.netflix.com seems to be behind a WAF or some sort of security solution
[-] Reason: The server header is different when an attack is detected.
The server header for a normal response is "atsjobsiteui 48d32be1-fc02-48ad-a199-7a2862b067a8", while the server header a response to an attack is "atsjobsi
teui 448c70f1-e013-4767-9226-987246f165c2",
[-] Number of requests: 7
```

dvd.netflix.com firewall protection details

```
(kali㉿kali)-[~]
└─$ wafw00f https://dvd.netflix.com


404 Hack Not Found
405 Not Allowed
403 Forbidden
500 Internal Error
502 Bad Gateway

~ WAFW00F : v2.1.0 -
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://dvd.netflix.com
[+] The site https://dvd.netflix.com is behind NetScaler AppFirewall (Citrix Systems) WAF.
[-] Number of requests: 2
```

Devices.netflix.com firewall protection details

```
(kali㉿kali)-[~]
└─$ wafw00f https://devices.netflix.com


404 Hack Not Found
405 Not Allowed
403 Forbidden
500 Internal Error
502 Bad Gateway

~ WAFW00F : v2.1.0 -
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://devices.netflix.com
[+] The site https://devices.netflix.com is behind Cloudflare (Cloudflare Inc.) WAF.
[-] Number of requests: 2
```

Meechum.netflix.com firewall protection details

```
(kali㉿kali)-[~]
└─$ wafw00f https://meechum.netflix.com


~ WAFW00F : v2.1.0 -
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://meechum.netflix.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[-] Number of requests: 7
```

Brand.netflix.com firewall protection details

Candidate.netflix.com firewall protection details

Media.netflix.com firewall protection details

```
(kali㉿kali)-[~]
$ wafw00f https://media.netflix.com

 404 Hack Not Found





~ WAFWOOF : v2.1.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://media.netflix.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[-] Number of requests: 7
```

***Help.netflix.com* firewall protection details**

10.7 Publicly Connected Service Details Enumeration

Shodan.io – Online Tool

Link: <https://www.shodan.io/>

Scan Results:

TOTAL RESULTS
11,670

TOP COUNTRIES

Country	Count
United States	7,226
Ireland	3,461
Australia	122
United Kingdom	3
Germany	2

TOP PORTS

Port	Count
7002	6,281
443	2,925
80	2,448

HTTP Status 404 “Not Found”

52.41.138.8
HTTP/1.1 404 Not Found
Content-Language: en
Content-Type: text/html; charset=utf-8
Date: Wed, 13 Oct 2021 09:18:36 GMT
Server: api-prod l-0b44le67ee6850a7e
Set-Cookie: nfvid=8QFmAAEBEBGJUUEwB8bdh9w4X84xhAQWML-CPQ6e8bmDukgb7QzWn1cuBkUy#3LjfhcoImEfnBn7UeJ_Uerdm2YNN1D6Bw91; cloud

HTTP Status 404 “Not Found”

52.144.140
HTTP/1.1 404 Not Found
Content-Language: en
Content-Type: text/html; charset=utf-8
Date: Wed, 13 Oct 2021 09:18:05 GMT
Server: api-prod l-0ef4b372d9fa2eed0
Set-Cookie: nfvid=BQFmAAEBEBG-1N140xRRMbfxaomsPYtAmXh_bJP8-iQodhWkc9_mltvJ5yyPtgNUxi9EUm_BjBUfzGipx-exRrStKzit0o03vh60; cloud

403 Forbidden

34.212.13.15
HTTP/1.1 403 Forbidden
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 13 Oct 2021 09:15:09 GMT
Server: nocp-modern 344d8512-8efc-4ec4-a031-e2403595e9e5
Set-Cookie: memcid=66de8679-c29f-4740-94a2-c27671370ad3; Max-Age=31536000; Expires=Thu, 13 Oct 2022 09:15:09 GMT; Path=/; i

Censys – Online Tool

Link: <https://censys.io/domain>

Scan Results:

Search 2.0 is now public! See our launch announcement and try it out! Warning: Search 1.0 is deprecated, see FAQ for more details.

Censys

Q Websites Expand Register Sign In

Results Report Docs

Quick Filters
For all fields, see [Data Definitions](#)

Protocol:
51 80/http
49 443/https
48 80/http_www
46 443/https_www
19 25/smtp

Tag:
51 http
49 https
19 smtp

Websites
Page: 1/3 Results: 53 Time: 155ms

Website	Details
netflix.com (54.170.196.176)	20 25/smtp, 443/https, 443/https_www, 80/http, 80/http_www Netflix - Watch TV Shows Online, Watch Movies Online www.netflix.com , account.netflix.com , ca.netflix.com
domain.netflix.com	
fast.com (23.198.103.141)	1,230 443/https, 443/https_www, 80/http, 80/http_www Internet Speed Test Fast.com fast.com , www.fast.com
isitdownrightnow.com (34.231.60.151)	9,236 25/smtp, 443/https, 443/https_www, 80/http, 80/http_www Is It Down Right Now? Website Down or Not? ip-172-30-0-91 , localhost, localhost.localdomain
80.http.www.get.body:<div class="status" style="margin-top:3px;">Netflix.com	
stuffgate.com (172.67.180.241)	12,396 443/https, 80/http, 80/http_www Free Online Website Analyzer with Alexa rank, Whois information, DNS records, traffic analysis, site... sni.cloudflaressl.com , stuffgate.com , *.stuffgate.com
443.https.get.body:</td> </tr><tr> <td style='width:180px;'><img src='/api/screenshot.file.php?url=netflix	

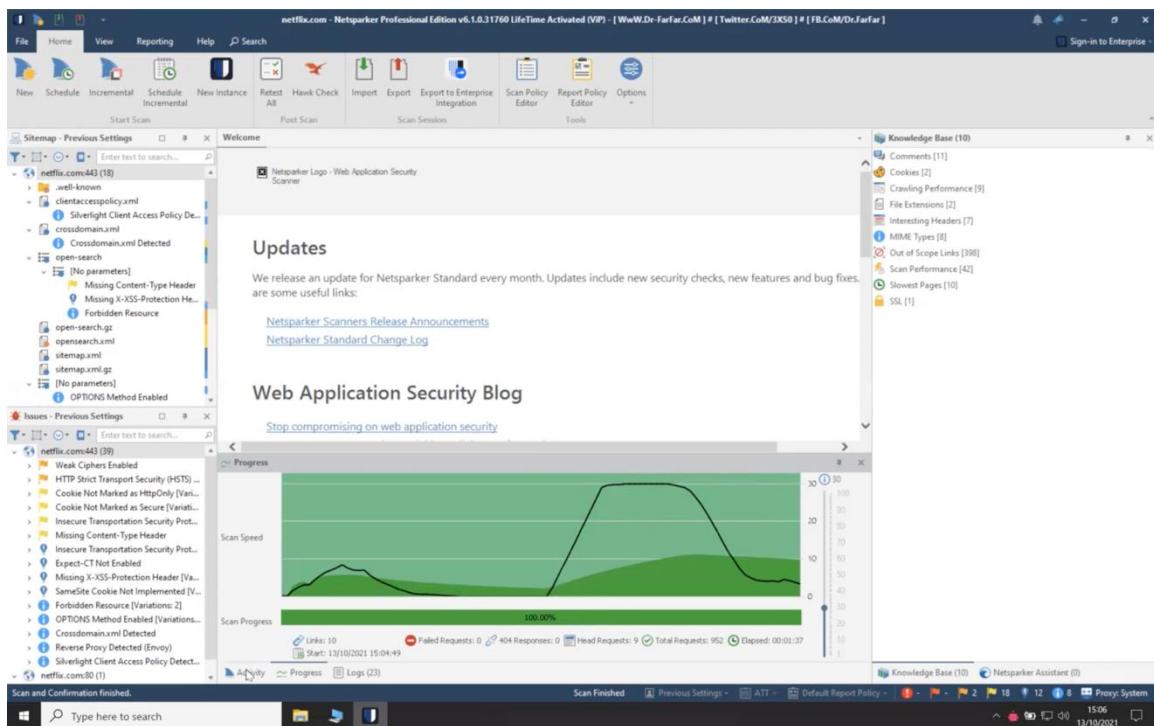
11. Vulnerability Analyzing and Reporting Phase

Tools I used for Vulnerability scanning

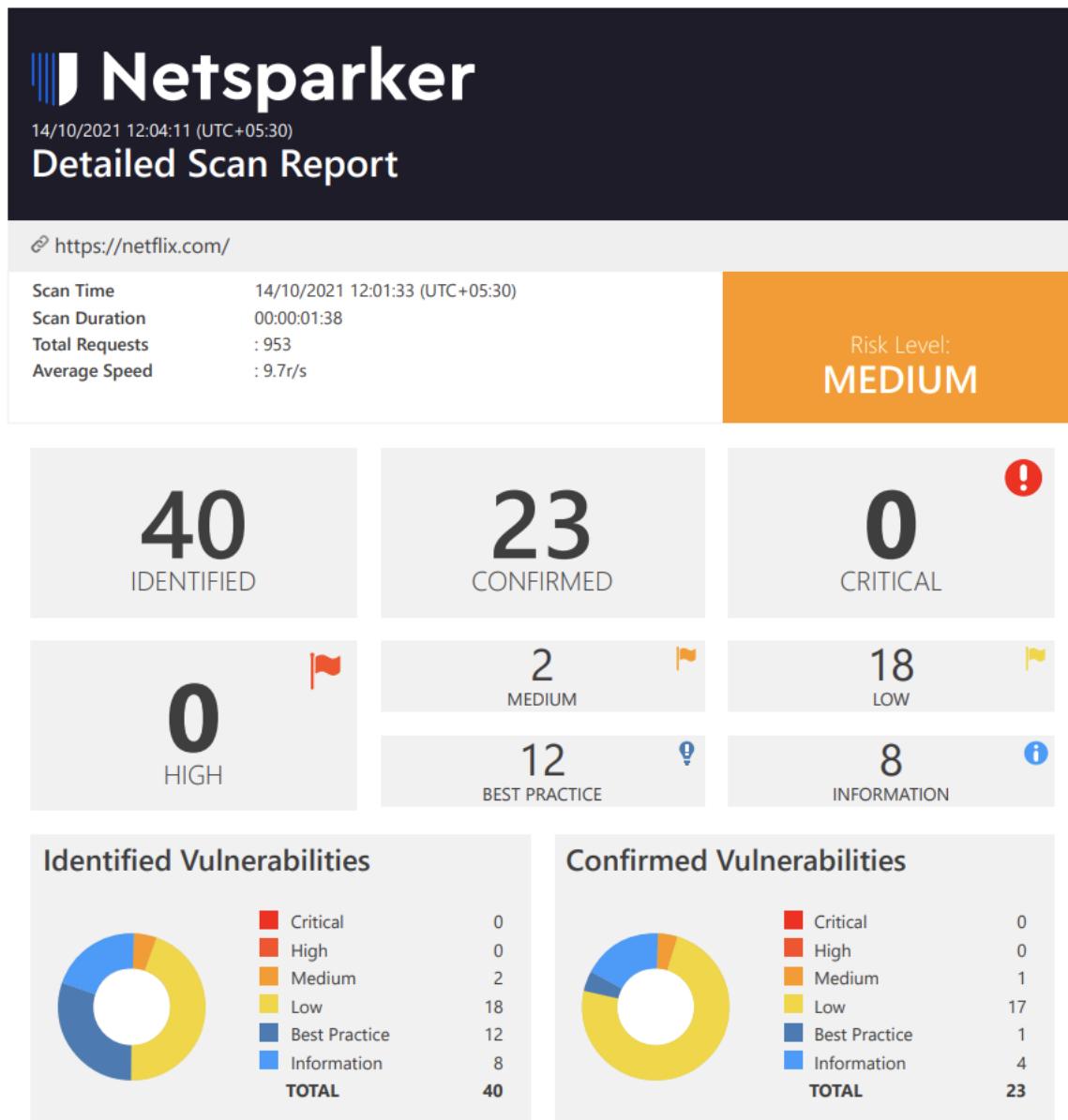
- Netsparker
- Nikto
- OWASP Zap Proxy

11.1 Netsparker – Automated Vulnerability Scanner

Netsparker is an automated vulnerability scanner which is used to find and analyze common vulnerabilities in web applications. Also, this scanner gives us the suggestion of remedy those found vulnerabilities. I did the scan process for root domain of Netflix.



Through this scanning I could able to generate Full-fledged vulnerability report. From this report I got **2 Medium risk level** bugs, and **18 Low level Bugs**, and some informative bugs.



This summary is cover page of the report which is graphically explains the whole report in a single view.

Following chart shows the name, risk level, method and URL of every vulnerability.

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
!	HTTP Strict Transport Security.(HSTS) Errors and Warnings	GET	https://netflix.com/	
!	Weak Ciphers Enabled	GET	https://netflix.com/	
!	Missing Content-Type Header	HEAD	https://netflix.com/opensearch	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/.well-known/	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/clientaccesspolicy.xml	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/crossdomain.xml	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/open-search.gz	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/opensearch.xml	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/sitemap.xml	
!	Cookie Not Marked as HttpOnly	GET	https://netflix.com/sitemap.xml.gz	
!	Cookie Not Marked as Secure	GET	https://netflix.com/	
!	Cookie Not Marked as Secure	GET	https://netflix.com/.well-known/	
!	Cookie Not Marked as Secure	GET	https://netflix.com/clientaccesspolicy.xml	
!	Cookie Not Marked as Secure	GET	https://netflix.com/crossdomain.xml	
!	Cookie Not Marked as Secure	GET	https://netflix.com/open-search.gz	

		Cookie Not Marked as Secure	GET	https://netflix.com/opensearch.xml
		Cookie Not Marked as Secure	GET	https://netflix.com/sitemap.xml
		Cookie Not Marked as Secure	GET	https://netflix.com/sitemap.xml.gz
		Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://netflix.com/
		Expect-CT Not Enabled	GET	https://netflix.com/
		Missing X-XSS-Protection Header	HEAD	https://netflix.com/opensearch
		Missing X-XSS-Protection Header	GET	https://netflix.com/swagger.json
		SameSite Cookie Not Implemented	GET	https://netflix.com/
		SameSite Cookie Not Implemented	GET	https://netflix.com/.well-known/
		SameSite Cookie Not Implemented	GET	https://netflix.com/clientaccesspolicy.xml
		SameSite Cookie Not Implemented	GET	https://netflix.com/crossdomain.xml
		SameSite Cookie Not Implemented	GET	https://netflix.com/open-search.gz
		SameSite Cookie Not Implemented	GET	https://netflix.com/opensearch.xml
		SameSite Cookie Not Implemented	GET	https://netflix.com/sitemap.xml
		SameSite Cookie Not Implemented	GET	https://netflix.com/sitemap.xml.gz
		Insecure Transportation Security Protocol Supported (TLS 1.1)	GET	https://netflix.com/
		Crossdomain.xml Detected	GET	https://netflix.com/crossdomain.xml

	HTTP Strict Transport Security (HSTS) via HTTP	GET	http://netflix.com/
	Reverse Proxy Detected (Envoy)	GET	https://netflix.com/
	Silverlight Client Access Policy Detected	GET	https://netflix.com/clientaccesspolicy.xml
	Forbidden Resource	HEAD	https://netflix.com/opensearch
	Forbidden Resource	GET	https://netflix.com/swagger.json
	OPTIONS Method Enabled	OPTIONS	https://netflix.com/
	OPTIONS Method Enabled	OPTIONS	https://netflix.com/.well-known/

Let's Discuss about some important vulnerabilities from this scan result report.

1. 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

1.1. <https://netflix.com/>

Error	Resolution
preload directive not present	Submit domain for inclusion in browsers' HTTP Strict Transport Security (HSTS) preload list.

Solution

Ideally, after fixing the errors and warnings, you should consider adding your domain to the HSTS preload list. This will ensure that browsers automatically connect your website by using HTTPS, actively preventing users from visiting your site using HTTP. Since this list is hardcoded in users' browsers, it will enable HSTS even before they visit your page for the first time, eliminating the need for Trust on First Use (TOFU) with its associated risks and disadvantages. Unless you fix the errors and warnings your website won't meet the conditions required to enter the browser's preload list.

Browser vendors declared:

- Serve a valid certificate.
- If you are listening on port 80, redirect all domains from HTTP to HTTPS on the same host. Serve all subdomains over HTTPS:
 - In particular, you must support HTTPS for the www subdomain if a DNS record for that subdomain exists
- Serve an HSTS header on the base domain for HTTPS requests:
 - The max-age must be at least 31536000 seconds (1 year)
 - ***includeSubDomains*** directive must be specified
 - The preload directive must be specified
 - If you are serving an additional redirect from your HTTPS site, that redirect must have the HSTS header (rather than the page it redirects to)

2. Weak Ciphers Enabled

MEDIUM  | 1

CONFIRMED  | 1

Netsparker detected that weak ciphers are enabled during secure communication (SSL).

You should allow only strong ciphers on your web server to protect secure communication with your visitors.

Impact

Attackers might decrypt SSL traffic between your server and your visitors.

Vulnerabilities

2.1. <https://netflix.com/>

CONFIRMED

List of Supported Weak Ciphers

- TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000A)
- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xC014)

Action to Take:

1. For Apache, you should modify the SSLCipherSuite directive in the `httpd.conf`.

```
SSLCipherSuite HIGH:MEDIUM:!MD5:!RC4
```

2. Lighttpd:

```
ssl.honor-cipher-order = "enable"  
ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM"
```

3. For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.

b. In Registry Editor, locate the following registry key: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`

c. Set "Enabled" DWORD to "0x0" for the following registry keys:

```
SCHANNEL\Ciphers\DES 56/56
SCHANNEL\Ciphers\RC4 64/128
SCHANNEL\Ciphers\RC4 40/128
SCHANNEL\Ciphers\RC2 56/128
SCHANNEL\Ciphers\RC2 40/128
SCHANNEL\Ciphers\NULL
SCHANNEL\Hashes\MD5
```

Solution

Configure your web server to disallow using weak ciphers.

3. Cookie Not Marked as HttpOnly

LOW  8

CONFIRMED  8

Netsparker identified a cookie not marked as HTTPOnly.

HTTPOnly cookies cannot be read by client-side scripts, therefore marking a cookie as HTTPOnly can provide an additional layer of protection against cross-site scripting attacks.

Impact

During a cross-site scripting attack, an attacker might easily access cookies and hijack the victim's session.

Vulnerabilities

3.1. <https://netflix.com/>

CONFIRMED

Identified Cookie(s)

- nfvdid
- memclid

Cookie Source

- HTTP Header

3.2. <https://netflix.com/.well-known/>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.3. <https://netflix.com/clientaccesspolicy.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.4. <https://netflix.com/crossdomain.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.5. <https://netflix.com/open-search.gz>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.6. <https://netflix.com/opensearch.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.7. <https://netflix.com/sitemap.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

3.8. <https://netflix.com/sitemap.xml.gz>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

Solution

Mark the cookie as `HTTPOnly`. This will be an extra layer of defense against XSS. However, this is not a silver bullet and will not protect the system against cross-site scripting attacks. An attacker can use a tool such as XSS Tunnel to bypass `HTTPOnly` protection.

4. Cookie Not Marked as Secure

LOW  8

CONFIRMED  8

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

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

Impact

This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (*such as a session cookie*), 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 steal the cookie.

Vulnerabilities

4.1. <https://netflix.com/>

CONFIRMED

Identified Cookie(s)

- nfvdid
- memclid

Cookie Source

- HTTP Header

4.2. <https://netflix.com/.well-known/>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

4.3. <https://netflix.com/clientaccesspolicy.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

4.4. <https://netflix.com/crossdomain.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

4.5. <https://netflix.com/open-search.gz>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

4.6. <https://netflix.com/opensearch.xml>

CONFIRMED

Identified Cookie(s)

28/75

- memclid

Cookie Source

- HTTP Header

4.7. <https://netflix.com/sitemap.xml>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

4.8. <https://netflix.com/sitemap.xml.gz>

CONFIRMED

Identified Cookie(s)

- memclid

Cookie Source

- HTTP Header

Solution

Mark all cookies used within the application as secure.

5. Insecure Transportation Security Protocol Supported (TLS 1.0)

LOW  1

CONFIRMED  1

Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.

TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).

Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018.

Impact

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Vulnerabilities

5.1. <https://netflix.com/>

CONFIRMED

Actions to Take

We recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the `SSLProtocol` directive provided by the `mod_ssl` module. This directive can be set either at the server level or in a virtual host configuration.

```
SSLProtocol +TLSv1.2
```

- For Nginx, locate any use of the directive `ssl_protocols` in the `nginx.conf` file and remove `TLSv1`.

```
ssl_protocols TLSv1.2;
```

- For Microsoft IIS, you should make some changes on the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

1. Click on Start and then Run, type `regedit32` or `regedit`, and then click OK.
2. In Registry Editor, locate the following registry key or create if it does not exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\tls 1.0\
```

3. Locate a key named Serverkey create if it doesn't exist.
 4. Under the Serverkey, locate a DWORD value named Enabled or create if it doesn't exist and set its value to "0".
- For lighttpd, put the following lines in your configuration file:

```
ssl.use-sslv2 = "disable"  
ssl.use-sslv3 = "disable"  
ssl.openssl.ssl-conf-cmd = ("Protocol" => "-TLSv1.1, -TLSv1, -SSLv3") # v1.4.48 or up  
ssl.ec-curve = "secp384r1"
```

6. Missing Content-Type Header

LOW 

| 1

Netsparker detected a missing Content-Type header which means that this website could be at risk of a MIME-sniffing attacks.

Impact

MIME type sniffing is a standard functionality in browsers to find an appropriate way to render data where the HTTP headers sent by the server are either inconclusive or missing.

This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type.

The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.

Vulnerabilities

6.1. <https://netflix.com/opensearch>

Remedy

1. When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served. For example, if you are serving an HTML page, you should send the HTTP header:

```
Content-Type: text/html
```

2. Add the X-Content-Type-Options header with a value of "nosniff" to inform the browser to trust what the site has sent is the appropriate content-type, and to not attempt "sniffing" the real content-type.

```
X-Content-Type-Options: nosniff
```

11.2 OWASP ZAP (Zed Attack Proxy) – OWASP Top 10 Vulnerabilities scanner

This is a learner friendly and pentester friendly free software which is developed and maintained by international volunteers to find web application vulnerabilities. This software works in Windows, Mac OS, Linux OS platforms. This software can generate easily understandable scanning report.

Active scanning

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
24	2021-10-13, 2:36:54 p.m.	2021-10-13, 2:36:54 p.m.	GET	http://netflix.com/7666061141933719834	200	OK	465 ms	2,012 bytes	16,902 bytes
25	2021-10-13, 2:36:58 p.m.	2021-10-13, 2:36:58 p.m.	GET	http://netflix.com/476549683597438865	200	OK	201 ms	2,012 bytes	16,902 bytes
27	2021-10-13, 2:37:00 p.m.	2021-10-13, 2:37:00 p.m.	GET	http://netflix.com/WEB-INF/web.xml	301	Moved Perma...	311 ms	833 bytes	0 bytes
28	2021-10-13, 2:37:00 p.m.	2021-10-13, 2:37:01 p.m.	GET	http://netflix.com/WEB-INF/applicationContext...	301	Moved Perma...	297 ms	862 bytes	0 bytes
29	2021-10-13, 2:37:02 p.m.	2021-10-13, 2:37:02 p.m.	GET	http://netflix.com/	301	Moved Perma...	222 ms	808 bytes	0 bytes
30	2021-10-13, 2:37:02 p.m.	2021-10-13, 2:37:02 p.m.	GET	http://netflix.com/	301	Moved Perma...	414 ms	809 bytes	0 bytes
31	2021-10-13, 2:37:02 p.m.	2021-10-13, 2:37:02 p.m.	GET	http://netflix.com/robots.txt	301	Moved Perma...	214 ms	823 bytes	0 bytes
32	2021-10-13, 2:37:02 p.m.	2021-10-13, 2:37:02 p.m.	GET	http://netflix.com/sitemap.xml	301	Moved Perma...	223 ms	633 bytes	0 bytes
33	2021-10-13, 2:37:04 p.m.	2021-10-13, 2:37:04 p.m.	GET	http://netflix.com/	200	OK	207 ms	2,148 bytes	301,537 bytes
34	2021-10-13, 2:37:07 p.m.	2021-10-13, 2:37:07 p.m.	GET	http://netflix.com/robots.txt/	200	OK	186 ms	1,043 bytes	3,565 bytes
35	2021-10-13, 2:37:04 p.m.	2021-10-13, 2:37:04 p.m.	GET	http://netflix.com/	200	OK	206 ms	2,148 bytes	301,869 bytes
36	2021-10-13, 2:37:07 p.m.	2021-10-13, 2:37:08 p.m.	GET	http://netflix.com/sitemap.xml/	404	Not Found	300 ms	833 bytes	0 bytes
37	2021-10-13, 2:37:09 p.m.	2021-10-13, 2:37:10 p.m.	GET	http://netflix.com/elmah.ashx	301	Moved Perma...	277 ms	821 bytes	0 bytes

Found Vulnerabilities from ZAP

Alerts (7)
> Absence of Anti-CSRF Tokens (2) > Cookie No HttpOnly Flag (12) > Cookie Without Secure Flag (3) > Cookie with SameSite Attribute (12) > Cross-Domain JavaScript Source File Inclusion (3) > Timestamp Disclosure - Unix (18) > Information Disclosure - Suspicious Comments (2)

Alerts (7)

Absence of Anti-CSRF Tokens (2)

Cookie No HttpOnly Flag (12)

Cookie Without Secure Flag (3)

Cookie with SameSite Attribute (12)

Cross-Domain JavaScript Source File Inclusion (3)

Timestamp Disclosure - Unix (18)

Information Disclosure - Suspicious Comments (2)

URL: http://netflix.com
Risk: Low
Confidence: Medium
Parameter:
Evidence: <form class="cta-form email-form" data-uia="email-form" method="GET">
CWE ID: 352
WASC ID: 9
Source: Passive (10202 – Absence of Anti-CSRF Tokens)
Description:
No Anti-CSRF tokens were found in a HTML submission form.
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way.
Other Info:
No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anonscsrf, csrf_token, csrf, csrfSecret, __csrf_magic, CSRF, _token, __csrf_token] was found in the following HTML form: [Form 1: "id_email_hero_fui"].

Zap proxy scan result report for [Netflix.com](#)

Summary of Alerts

Generated on Thu, 14 Oct 2021 22:41:06

Risk Level	Number of Alerts
High	0
Medium	0
Low	7
Informational	3

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Low	2
Cookie No HttpOnly Flag	Low	11
Cookie Without SameSite Attribute	Low	11
Cookie Without Secure Flag	Low	3
Cross-Domain JavaScript Source File Inclusion	Low	3
Information Disclosure - Suspicious Comments	Informational	1
Timestamp Disclosure - Unix	Informational	17

Low (Medium)	Absence of Anti-CSRF Tokens
	No Anti-CSRF tokens were found in a HTML submission form.
	A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.
Description	CSRF attacks are effective in a number of situations, including: <ul style="list-style-type: none">* The victim has an active session on the target site.* The victim is authenticated via HTTP auth on the target site.* The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
URL	http://netflix.com
Method	GET
Evidence	<form class="cta-form email-form" data-uia="email-form" method="GET">
URL	http://netflix.com
Method	GET
Evidence	<form class="cta-form email-form" data-uia="email-form" method="GET">
Instances	2
Solution	<p>Phase: Architecture and Design</p> <p>Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.</p> <p>For example, use anti-CSRF packages such as the OWASP CSRGGuard.</p> <p>Phase: Implementation</p> <p>Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.</p> <p>Phase: Architecture and Design</p>

Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

Note that this can be bypassed using XSS.

Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS.

Use the ESAPI Session Management control.

This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

Phase: Implementation

Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Other information

No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF] was found in the following HTML form: [Form 2: "id_email_faq"].

Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Source ID	3

Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	https://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
Instances	3
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://netflix.com/
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com/
Method	GET
Parameter	clSharedContext
Evidence	Set-Cookie: clSharedContext
URL	http://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com/
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
Instances	8
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	https://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
Instances	3
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie No HttpOnly Flag
Description	A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then session hijacking may be possible.
URL	http://netflix.com/
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com
Method	GET
Parameter	clSharedContext
Evidence	Set-Cookie: clSharedContext
URL	http://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
Instances	8
Solution	Ensure that the HttpOnly flag is set for all cookies.
Reference	https://owasp.org/www-community/HttpOnly
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie Without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	https://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
Instances	3
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie Without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.
URL	http://netflix.com
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	http://netflix.com/robots.txt
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com/
Method	GET
Parameter	nfvdid
Evidence	Set-Cookie: nfvdid
URL	http://netflix.com
Method	GET
Parameter	clSharedContext
Evidence	Set-Cookie: clSharedContext
Instances	8
Solution	Ensure that the SameSite attribute is set to either 'lax' or ideally 'strict' for all cookies.
Reference	https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site
CWE Id	16
WASC Id	13
Source ID	3

Low (Medium)	Cookie Without Secure Flag
Description	A cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.
URL	https://netflix.com/robots.txt
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/sitemap.xml
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
URL	https://netflix.com/
Method	GET
Parameter	memclid
Evidence	Set-Cookie: memclid
Instances	3
Solution	Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
Reference	https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html
CWE Id	614
WASC Id	13
Source ID	3

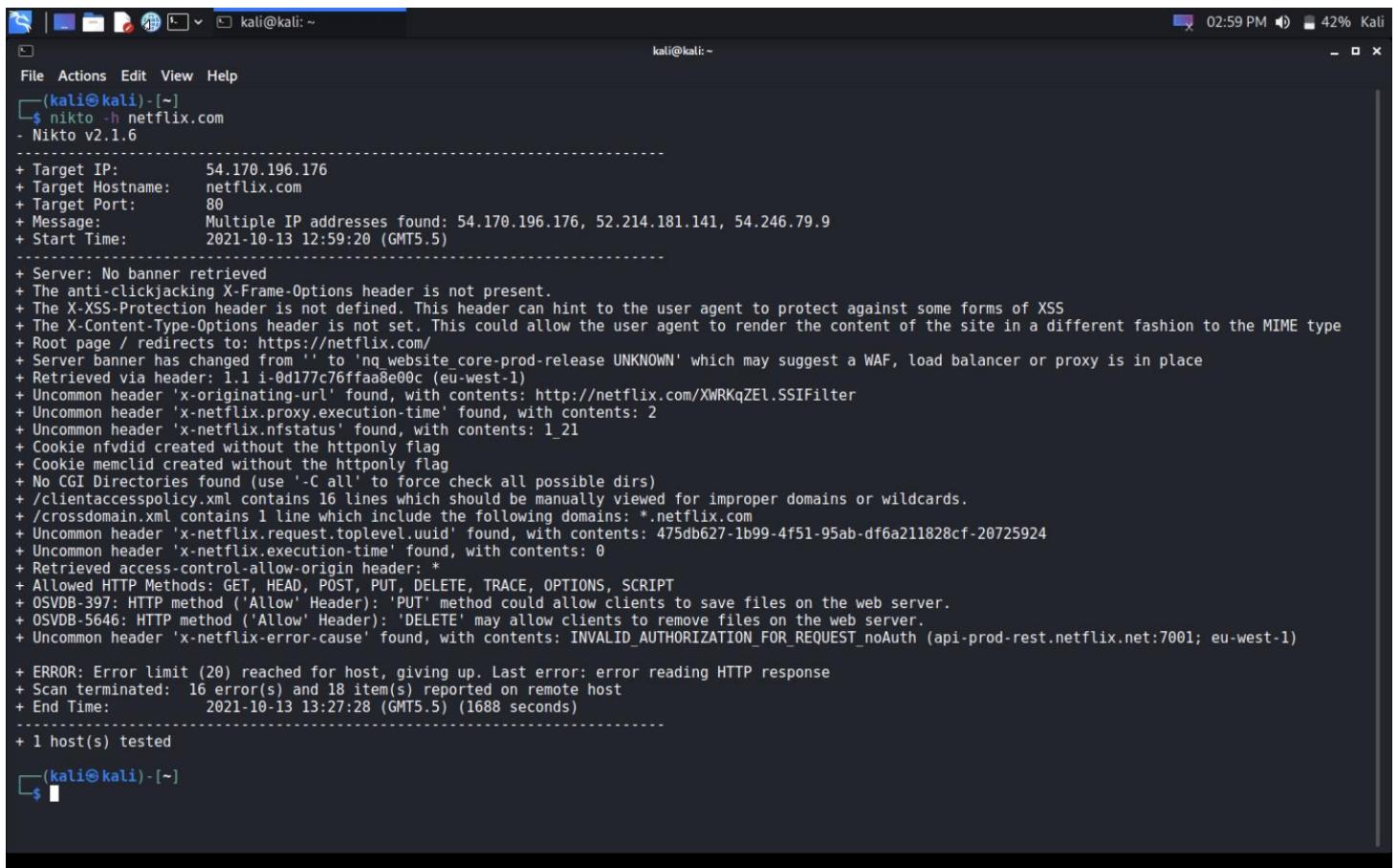
Low (Medium)	Cross-Domain JavaScript Source File Inclusion
Description	The page includes one or more script files from a third-party domain.
URL	http://netflix.com
Method	GET
Parameter	https://codex.netflixext.com/%5E3.0.0/truthBundle/webui/1.22.5-shakti-js-vad47cefd/js/js/bootstrap.js,common%7Cbootstrap.js/2/0b3d022S2L2U052I2Y3c070l003e2X382_2V372M2Z302N2F01390N/bck/true/none
Evidence	<script src=" https://codex.netflixext.com/%5E3.0.0/truthBundle/webui/1.22.5-shakti-js-vad47cefd/js/js/bootstrap.js,common%7Cbootstrap.js/2/0b3d022S2L2U052I2Y3c070l003e2X382_2V372M2Z302N2F01390N/bck/true/none "></script>
URL	http://netflix.com
Method	GET
Parameter	https://codex.netflixext.com/%5E3.0.0/truthBundle/webui/1.22.5-shakti-js-vad47cefd/js/js/signup%7Cnmhp%7CnmhpFrameworkClient.js/2/0b3d022S2L2U052I2Y3c070l003e2X382_2V372M2Z302N2F01390N/l/true/none
Evidence	<script src=" https://codex.netflixext.com/%5E3.0.0/truthBundle/webui/1.22.5-shakti-js-vad47cefd/js/js/signup%7Cnmhp%7CnmhpFrameworkClient.js/2/0b3d022S2L2U052I2Y3c070l003e2X382_2V372M2Z302N2F01390N/l/true/none "></script>
URL	http://netflix.com
Method	GET
Parameter	https://cdn.cookielaw.org/scripttemplates/otSDKStub.js
Evidence	<script type="text/javascript" charSet="UTF-8" data-domain-script="87b6a5c0-0104-4e96-a291-092c11350111" src=" https://cdn.cookielaw.org/scripttemplates/otSDKStub.js "></script>
Instances	3
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.
Reference	
CWE Id	829
WASC Id	15
Source ID	3

11.3 Nikto Scanning

Usage:

```
nikto -h netflix.com
```

Scan Result: *Nikto is a kali Linux inbuilt tool which is used to scan webservers for malicious files or dangerous files, expired server data information.*



```
(kali㉿kali)-[~]
$ nikto -h netflix.com
- Nikto v2.1.6
-----
+ Target IP:      54.170.196.176
+ Target Hostname: netflix.com
+ Target Port:    80
+ Message:        Multiple IP addresses found: 54.170.196.176, 52.214.181.141, 54.246.79.9
+ Start Time:     2021-10-13 12:59:20 (GMT5.5)
-----
+ Server: No banner retrieved
+ 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
+ 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://netflix.com/
+ Server banner has changed from '' to 'nq website.core-prod-release UNKNOWN' which may suggest a WAF, load balancer or proxy is in place
+ Retrieved via header: 1.1 i-0d177c76fffaa8e00c (eu-west-1)
+ Uncommon header 'x-originating-url' found, with contents: http://netflix.com/XWRKqZE1.SSIFilter
+ Uncommon header 'x-netflix.proxy.execution-time' found, with contents: 2
+ Uncommon header 'x-netflix.nfstatus' found, with contents: 1_21
+ Cookie nfvid created without the httponly flag
+ Cookie memclid created without the httponly flag
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /clientaccesspolicy.xml contains 16 lines which should be manually viewed for improper domains or wildcards.
+ /crossdomain.xml contains 1 line which include the following domains: *.netflix.com
+ Uncommon header 'x-netflix.request.toplevel.uuid' found, with contents: 475db627-1b99-4f51-95ab-df6a211828cf-20725924
+ Uncommon header 'x-netflix.execution-time' found, with contents: 0
+ Retrieved access-control-allow-origin header: *
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, SCRIPT
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ Uncommon header 'x-netflix-error-cause' found, with contents: INVALID_AUTHORIZATION_FOR_REQUEST_noAuth (api-prod-rest.netflix.net:7001; eu-west-1)

+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 16 error(s) and 18 item(s) reported on remote host
+ End Time:       2021-10-13 13:27:28 (GMT5.5) (1688 seconds)
-----
+ 1 host(s) tested

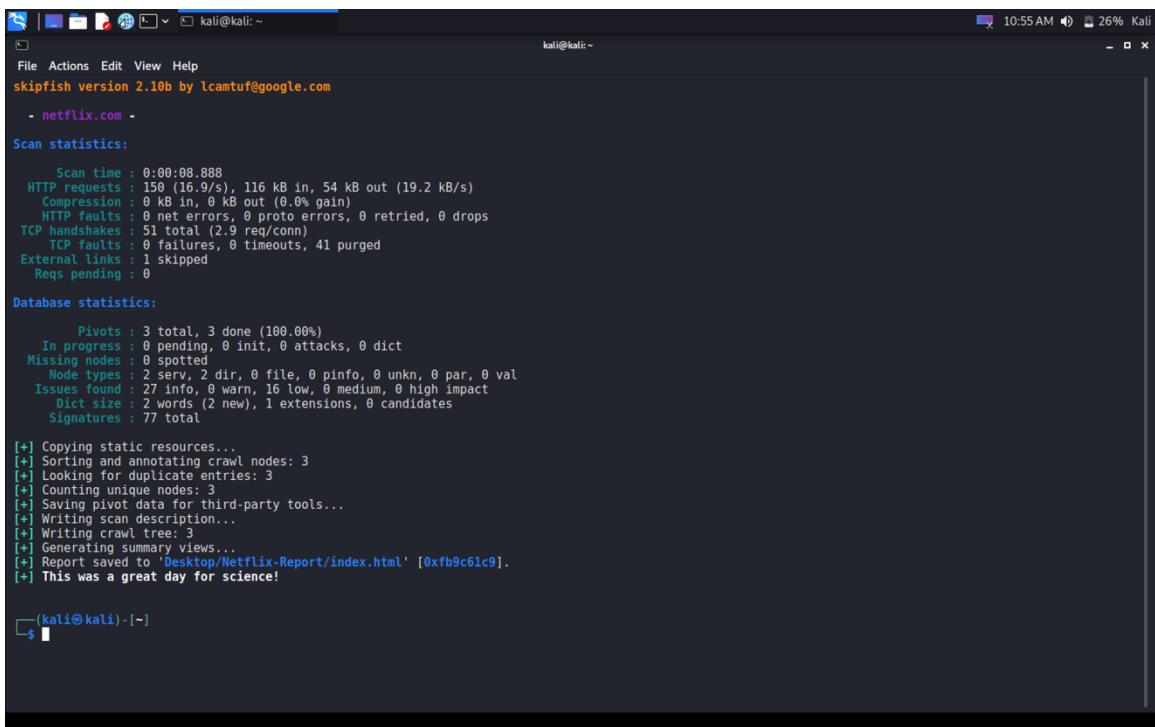
(kali㉿kali)-[~]
$
```

11.4 Skipfish Scanning

Usage:

```
skipfish -o Desktop/Netflix-Report http://netflix.com
```

Scan Result:



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar indicates it's running on Kali at 10:55 AM with 26% battery. The terminal content displays the results of a skipfish scan against the Netflix website. The output includes detailed statistics about the scan time, network traffic, and database processing, followed by a series of informational messages indicating the progress of post-processing steps like copying static resources and generating reports.

```
kali@kali: ~
File Actions Edit View Help
skipfish version 2.10b by lcamtuf@google.com
- netflix.com -
Scan statistics:
  Scan time : 0:00:08.888
  HTTP requests : 150 (16.9/s), 116 kB in, 54 kB out (19.2 kB/s)
  Compression : 0 kB in, 0 kB out (0.0% gain)
  HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
  TCP handshakes : 51 total (2.9 req/conn)
  TCP faults : 0 failures, 0 timeouts, 41 purged
  External links : 1 skipped
  Reqs pending : 0

Database statistics:
  Pivots : 3 total, 3 done (100.00%)
  In progress : 0 pending, 0 init, 0 attacks, 0 dict
  Missing nodes : 0 spotted
  Node types : 2 serv, 2 dir, 0 file, 0 pinfo, 0 unkn, 0 par, 0 val
  Issues found : 27 info, 0 warn, 16 low, 0 medium, 0 high impact
  Dict size : 2 words (2 new), 1 extensions, 0 candidates
  Signatures : 77 total

[+] Copying static resources...
[+] Sorting and annotating crawl nodes: 3
[+] Looking for duplicate entries: 3
[+] Counting unique nodes: 3
[+] Saving pivot data for third-party tools...
[+] Writing scan description...
[+] Writing crawl tree: 3
[+] Generating summary views...
[+] Report saved to 'Desktop/Netflix-Report/index.html' [0xfb9c61c9].
[+] This was a great day for science!
```

Generated Report:

Skipfish - scan results browser - Mozilla Firefox

file:///home/kali/Desktop/Netflix-Report/index.html

Scanner version: 2.10b Scan date: Fri Oct 15 09:25:06 2021
Random seed: 0xb9c61c9 Total time: 0 hr 0 min 8 sec 989 ms
Problems with this scan? Click here for advice.

Crawl results - click to expand:

- + http://netflix.com/ 11
Code: 301, length: 0, declared: [none], charset: [none] | show trace +
- + https://netflix.com/ 16 16
Code: 301, length: 0, declared: [none], charset: [none] | show trace +

Document type overview - click to expand:

- text/xml (1)

Issue type overview - click to expand:

- SSL certificate host name mismatch (16)
- Incorrect or missing charset (low risk) (2)
- New 404 signature seen (2)
- New 'X-*' header value seen (15)
- New 'Via' header value seen (2)
- New 'Server' header value seen (2)
- New HTTP cookie added (3)
- SSL certificate issuer information (1)

NOTE: 100 samples maximum per issue or document type.

Skipfish - scan results browser - Mozilla Firefox

file:///home/kali/Desktop/Netflix-Report/index.html

Scanner version: 2.10b Scan date: Fri Oct 15 09:25:06 2021
Random seed: 0xb9c61c9 Total time: 0 hr 0 min 8 sec 989 ms
Problems with this scan? Click here for advice.

Crawl results - click to expand:

- http://netflix.com/ 11
Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Incorrect or missing charset (low risk)
 - 1. Code: 400, length: 94, declared: application/xml, detected: text/xml, charset: [none] | show trace +
 - New 404 signature seen
 - 1. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - New 'X-*' header value seen
 - 1. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: X-Xss-Protection
 - 2. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: X-Content-Type-Options
 - 3. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: X-Frame-Options
 - 4. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: X-NetInfo-rftstatus
 - 5. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: X-Netflix-proxy-execution-time
 - New 'Via' header value seen
 - 1. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: 1.1+0x40cef95e6831fa (eu-west-1)
 - New 'Server' header value seen
 - 1. Code: 301, length: 0, declared: [none], charset: [none] | show trace +
 - Memo: rq_website_nmember-prod-release UNKNOWN
 - New HTTP cookie added
 - 1. Code: 301, length: 18, declared: [none], charset: [none] | show trace +
 - Memo: nvfid
 - 2. Code: 301, length: 18, declared: [none], charset: [none] | show trace +
 - Memo: memcid
 - + https://netflix.com/ 16 16
Code: 301, length: 0, declared: [none], charset: [none] | show trace +

12. Conclusion and Recommendations

This report clearly explained about why information gathering is important in web application penetration testing and how the gathered informations are helpful in vulnerability assessment. Vulnerabilities are categorized by their severity as high, medium, low and informational. Informational bugs do not affect the business, but Critical, High, medium level bugs are major part of the security and they should have cleared.

From this assessment I learnt how to use tools efficiently and how to write a proper audit report. This assignment will be helpful to me in future when I am working and doing researches. Web security is one of the most important part of information security. If we don't secure applications, it will not only affect the website's confidentiality, availability, integrity. Also, it will affect the business related through the web applications. As a Cyber Security student, it is our responsibility to spread awareness about security threads in digital world. Be Secured.

13. References

- [1] " OWASP Top Ten". Available: <https://owasp.org/www-project-top-ten/>. [Accessed: 15- October- 2021].
- [2] " Netflix's Bug Bounty Program". Available: <https://bugcrowd.com/netflix>. [Accessed: 15- October- 2021].
- [3] " How to Write a Better Vulnerability Report". Available: <https://medium.com/swlh/how-to-write-a-better-vulnerability-report-20163ab913fb> [Accessed: 15- October- 2021].
- [4] " Sublist3r". <https://github.com/aboul3la/Sublist3r>
- [5] " Knockpy". <https://github.com/guelfoweb/knock>
- [6] " Dirsearch". <https://github.com/maurosoria/dirsearch>
- [7] " Netsparker Reporting Overview video walkthroughs". <https://www.youtube.com/watch?v=wpFo6uPAgJY>
- [8] " ZAP Deep Dive: Report Generation". <https://www.youtube.com/watch?v=kD540gUWJ3I>

Thank You