



IT19122410
W.P.T. Pamalka
Web Security
Web Audit Assignment

Content

1. Purpose & Scope	3
2. Information Gathering	4
3. Target Validation	5
4. Finding Subdomains	6
4.1 Sublist3r	7 - 8
5. Vulnerability scanning	9
5.1 Nikto	10 - 14
6. Fingerprinting	15
6.1 Nmaps	16 - 21
7. Vulnerabilities & CIA Triad	22
7.1 amazon.de	
7.2 amazon.co.uk	
7.3 amazon.com.mx	
7.4 amazon.co.jp	
7.5 amazon.fr	
7.6 amazon.in	
7.7 amazon.ca	
7.8 amazon.es	
8. Conclusion	61

Purpose

A web audit is done to assess vulnerabilities that may lead to security breaches and discover inconsistencies that are present in the site which may lead to google penalties, in a web application.

We were asked to perform a web audit on any domain of our choosing. The tests were conducted on 11th of October and concluded in 14th of October. This report shows how to conduct a primary web audit touching on information gathering topic.

Scope

The scope was limited to few subdomains of the main web application amazon.com.

In Scope

Domain	www.amazon.com	 Critical
Domain	www.amazon.co.uk	 Critical
Domain	www.amazon.co.jp	 Critical
Domain	www.amazon.de	 Critical
Domain	www.amazon.fr	 Critical
Domain	www.amazon.com.mx	 Critical
Domain	www.amazon.es	 Critical
Domain	www.amazon.in	 Critical
Domain	www.amazon.ca	 Critical

In this assignment, we have mainly elaborated the information gathering section.

Information gathering

This is the stage where the penetration tester gathers all publicly available information about the web application through various sources. It helps the penetration tester to understand the web application and the network system. This newfound information helps the tester to understand vulnerabilities and to identify the impact of those vulnerabilities.

There are two types of information gathering. They are;

1. Passive – gathering information without any contact with the target web application
2. Active – gathering information while being in contact with the target web application

In this assignment the main focus is on passive information gathering technique as it is contactless information gathering between myself and the target. In passive information gathering there are few steps that should be followed. They are: target validation, finding subdomains, vulnerability scanning, fingerprinting.

Target validation

Target validation is making sure that the target we are auditing is in scope and it is the one that client gave us. This purpose can be fulfilled by using several tools such as WHOIS, nslookup and dnsrecon.

The tool “WHOIS” is a query and response protocol that is used for querying databases that store the registered users of an Internet resource, such as a domain name, an IP address block or an autonomous system but is also used for a wider range of other information.

WHOIS LOOKUP

amazon.com is already registered*

Domain Name: AMAZON.COM
Registrar Domain ID: 291205.DOMAIN.COM-RSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2023-10-20T05:00:00Z
Creation Date: 1994-11-01T05:00:00Z
Registry Expiry Date: 2024-10-31T04:00:00Z
Registrar: NSI INC.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abuse@amazon.com
Registrar Abuse Contact Phone: +1 2028395740
Domain Status: clientDeleteProhibited https://icann.org/lepp/clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/lepp/clientTransferProhibited
Domain Status: serverDeleteProhibited https://icann.org/lepp/serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/lepp/serverTransferProhibited
Domain Status: updateProhibited https://icann.org/lepp/updateProhibited
Name Server: NS1.P3L.DYNECT.NET
Name Server: NS2.P3L.DYNECT.NET
Name Server: NS3.P3L.DYNECT.NET
Name Server: NS4.P3L.DYNECT.NET
Name Server: PONSA.ULTRADS.CO.UK
DNSSEC: unsigned
URL: https://www.icann.org/whois/ComplaintForm https://www.icann.org/lepp/
>>> Last update of whois database: 2023-10-13T07:18:29Z <<

For more information on Whois status codes, please visit <https://icann.org/lepp>

NOTICE: The expiration date displayed in this record is the date the registrant'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. The registrant is responsible for managing their registration and is liable for any actions that result in its termination before the date of expiration to view the registrant's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois database through the use of electronic processes that are high-volume and automated, except as reasonably required to register new domains or modify existing registrations; the Data in VeriSign Global Registry Services ("VeriSign") Whois database is provided by VeriSign for informational purposes only and is a reflection of the current status of domain names or related to a domain name registration record. VeriSign does not guarantee its accuracy. By submitting a Whois query, you agree to abide by the following terms of use. You agree that you may use this Data only

Popular
No Results Found

Professional
No Results Found

Filters

Popular

Arts and Culture

Audio and Video

Businesses

Colors

Computers and Internet

Descriptive

Educational and Academic

Financial and Banking

Food and Drink

Fun and Unique

Geographic

Health and Fitness

Lifestyles and Relationships

Marketing and Sales

Media and Music

Organizations

Personal

Products

Professional

Real Estate

Finding subdomains

Subdomains may have exploits or vulnerabilities within them which can lead to gaining access to the whole website. However, we do not have the slightest idea of what kind of vulnerabilities there are in the subdomains as they do not advertise them.

Most of the time subdomains are underdeveloped and have experimental features such as beta.facebook.com. As a result, there is a high chance of finding exploits in them.

Example: Researcher Anand Prakash was able to brute-force the restore password key for any Facebook user and was able to gain access to any Facebook user's account. This was only possible through **beta.Facebook.com** as Facebook used to check for a number of attempts or failed attempts, and they didn't implement that security feature in the beta version because they did not think anyone was going to go there.

Popular tools that are utilized to find subdomains are:

- Nmap (active scanning)
- Sublist3r
- crt.sh

For this project, I have used both Sublist3r and crt.sh.

Sublist3r

Sublist3r is a python-based tool that identify subdomains of a website. By using this tool, attackers and penetration testers collect subdomains of the targeted domain in order to run a penetration test or an attack.

Sublist3r search subdomains using search engines such as: Google, Bing, Yahoo, Baidu etc. and other services such as Netcraft, ThreatCrowd, DNSdumpster, ReverseDNS etc.

In order to install sublist3r in Kali Linux run

“git clone <https://github.com/aboul3la/Sublist3r.git>” in the terminal to download. After getting into sublist3r directory, run “python sublist3r.py -d domain name” to find subdomains of the main domain.

As to show subdomains that only have open ports 80 and 443 run command “python sublist3r.py -d domain name -p 80, 443” and to find subdomains from only specific search engines or services, run command “python sublist3r.py -e search engine/services -d domain name”

The screenshot shows a terminal window titled "root@kali: ~/Sublist3r". The terminal displays the following commands and output:

```
root@kali:~/Sublist3r# git clone https://github.com/aboul3la/Sublist3r.git
fatal: destination path 'Sublist3r' already exists and is not an empty directory.
root@kali:~/Sublist3r# ls
Desktop Documents Downloads Music Pictures Public Sublist3r Templates Videos
root@kali:~/Sublist3r# cd Sublist3r
root@kali:~/Sublist3r# No such file or directory
root@kali:~/Sublist3r# ls
LICENSE MANIFEST.in README.md requirements.txt setup.py subbrute sublist3r.py
root@kali:~/Sublist3r# python sublist3r.py
# Coded By Ahmed Aboul-Ela - @aboul3la
Usage: python sublist3r.py [Options] use -h for help
root@kali:~/Sublist3r# python sublist3r.py -d amazon.com
# Coded By Ahmed Aboul-Ela - @aboul3la
[+] Enumerating subdomains now for amazon.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 Shodan...
[-] Searching now in ThreatCrowd...
[-] Searching now in SSL Certificates...
[-] Searching now in PassiveDNS...
```

```
File Actions Edit View Help
root@kali: ~/Sublist3r
root@kali: ~/Sublist3r

polaren-us-west-2.amazon.com
policyengine.amazon.com
policyengine-eu.amazon.com
pong-us-east-1.amazon.com
pong-us-west-2.amazon.com
poweredby.amazon.com
prime.amazon.com
primenow.amazon.com
promotion-engine.amazon.com
promotions.amazon.com
provisioning-web.amazon.com
browse-query-editor.eu-dub.dub.proxy.amazon.com
dns-lookup-eu-1.iad.iad.proxy.amazon.com
gcs-legacy-tools-iad.iad.proxy.amazon.com
maxis-file-service-prod-pdx.pdx.proxy.amazon.com
ps-apps-us.amazon.com
pts-a2z-eu.amazon.com
pts-a2z-na.amazon.com
pts-a2z-syrah.amazon.com
pulse-eu.amazon.com
pulse-fe.amazon.com
pushoutgateway.amazon.com
pushoutintegration.amazon.com
pushup.amazon.com
push.amazon.com
pgv51-br-tra-p-1.amazon.com
pgv51-br-tra-p-12.amazon.com
pgv51-br-tra-p-13.amazon.com
pgv51-br-tra-p-24.amazon.com
pgws-apconfig-14.amazon.com
pgws-apconfig-15.amazon.com
quorusr.amazon.com
r.amazon.com
rainier-acms-media-upload-amazon.com
rainier-acms-media-upload-cn.amazon.com
rainier-mcms-media-upload-eu.amazon.com
rainier-mcms-media-upload-fe.amazon.com
rainier-rcm.amazon.com
rcm-images.amazon.com
rds-op-qe2.amazon.com
rds-op-use1.amazon.com
reading.amazon.com
reading.amazon.com
reading.reading.amazon.com
redfort.amazon.com
register-app.amazon.com
register-domain.amazon.com
whois.registrar.amazon.com
relay.amazon.com
reminds-polls-us.amazon.com
retail-integration.amazon.com
retail-setup-in.amazon.com
retail-setup-out.amazon.com
retail-test-in.amazon.com
retrocharge-tool-eu.amazon.com
retrocharge-tool-na.amazon.com
rewrite-eu.amazon.com
```

Certificates	crt.sh ID	Logged At	Not Before	Not After	Common Name	Matching Identities	Issuer Name
	2381082454	2020-01-27	2014-06-25	2014-07-12	chat.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2381078306	2020-01-27	2014-06-25	2014-07-12	chat.amazon.cn	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380841887	2020-01-26	2013-06-20	2014-06-21	crowd-staging.labcollab.net	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380745941	2020-01-26	2013-08-14	2014-08-15	wiki.labcollab.net	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380357667	2020-01-26	2011-09-01	2013-03-31	sea-salmon.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380350099	2020-01-26	2011-01-28	2013-01-27	acsvc.us-east-1.amazonaws.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380241244	2020-01-26	2010-11-08	2013-11-07	atv-ps.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380319489	2020-01-26	2010-10-28	2013-10-27	adserver.www.endless.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380158160	2020-01-26	2010-10-08	2013-10-07	target-preview.amazonpmi.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380159044	2020-01-26	2010-10-08	2013-10-07	targeteconosale.mac10131963.amazonaws.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380154381	2020-01-26	2010-09-15	2013-09-14	digital-delivery-jp.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G3
	2380154203	2020-01-26	2010-10-08	2013-10-07	digital-delivery-eu-preprod.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380154074	2020-01-26	2010-08-30	2013-08-29	www.amazon.it	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380152266	2020-01-26	2011-02-22	2012-02-22	cde-la-g7.amazon.co.uk	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380152142	2020-01-26	2010-09-15	2013-10-14	digital-delivery-jp-preprod.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380150969	2020-01-26	2010-10-08	2013-10-07	xm1-target.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380150027	2020-01-26	2010-10-08	2013-10-07	shippingpolicy-na.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380148793	2020-01-26	2010-10-10	2013-10-07	digital-delivery-na-preprod.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380147695	2020-01-26	2010-10-08	2013-10-07	cde-la-g7-preprod.amazon.co.uk	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380146447	2020-01-26	2010-07-08	2011-07-08	merchant-api.amazon.de	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380146363	2020-01-26	2010-06-23	2011-06-23	s3.amazonaws.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380146067	2020-01-26	2010-05-26	2011-05-26	website-g7.amazon.com	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2
	2380146174	2020-01-26	2010-06-26	2011-06-26	website-g7.amazon.co.uk	Amazon.com Inc.	C=US, O=VeriSign Inc., OU=VeriSign Trust Network, OU=Terms of use at https://www.verisign.com/pa/c10/CN=VeriSign Class 3 Secure Server CA -G2

<https://crt.sh/?q=%25.amazon.com>

vulnerability scanning

vulnerability is a weakness within a system that may leads to the system being attacked, if exploited successfully, therefore vulnerability management is crucial. However, in order to manage vulnerabilities first it must be discovered. It can be only achieved through an extensive vulnerability scanning process.

Vulnerability scanner is a tool that identifies and creates an inventory of all systems. Then it cross examines each of these systems against well-known vulnerabilities and scan whether such vulnerabilities are present in the system.

Finally, the scanner displays a list of systems and highlights any vulnerabilities that may need attention. A few popular open source scanners are:

- Nikto
- Wireshark
- OpenVAS
- Retina

Nikto

Nikto is a vulnerability scanning tool that scan dangerous files and scripts and other issues of the target web application. Nikto runs a large number of tests for security vulnerabilities and it also scan for out of date and unpatched software as well as dangerous files on the web server. Nikto is capable of identifying a wide range of issues and also search for configuration issues.

```
root@kali:~# nikto -h https://www.amazon.com
- Nikto v2.1.6

+ Target IP: 54.230.151.86
+ Target Hostname: www.amazon.com
+ Target Port: 443

+ SSL Info:
  Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=www.amazon.com
  Ciphers: TLS_AES_128_GCM_SHA256
  ISSUED BY: /C=US/ST=California/O=DigiCert Global CA G2
  Start Time: 2020-10-11 15:55:08 (GMT+0)

+ Server: Server
+ Server header: 1.1/07fb7f22e92d3c5c571c1cf046093a.cloudfront.net (CloudFront)
+ Uncommon header 'x-cache' found, with contents: Miss from cloudfront
+ Uncommon header 'accept-ch-lifetime' found, with contents: 86400
+ Uncommon header 'x-amzn-trace-id' found, with contents: 00000000000000000000000000000000
+ Uncommon header 'x-amz-cf-pop' found, with contents: SIN2-C
+ Uncommon header 'x-amz-rid' found, with contents: 0025K27RSC6C6BTBK6
+ The site uses SSL and expect-CT header is not present.
+ SSL certificate is valid
+ Cookie session-id created without the httponly flag
+ Cookie session-id created without the secure flag
+ Cookie session-time created without the secure flag
+ Cookie session-time created without the httponly flag
+ Cookie i18n-prefs created without the secure flag
+ Cookie i18n-prefs created without the httponly flag
+ Cookie skin created without the secure flag
+ Cookie skin created without the httponly flag
+ Server proxy modes as a trace header found with file /MDEQREQ.pwd, inode: 687, size: 5ae85a47b9840, mtime: gzip
+ Uncommon header 'x-sdch-encode' found, with contents: 0
+ No CGI Directories found (use --all to force check all possible flags)
+ Uncommon header 'x-amz-id-1' found, with contents: AVEF2E4A340B859
+ Uncommon header 'x-amz-1' found, with contents: REPKXRC4q40SPFXTCH
+ Entry '/exec/obidos/account-access-login/' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/exec/obidos/change-style/' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/exec/obidos/choose-language/' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/exec/obidos/handle-buy-box/' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/exec/obidos/tg/cm/member/' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/gp/aw/privacy/policy' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/aw/privacy/terms' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/ichip/syltguides/create/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ Entry '/gp/yourstore/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ Entry '/gp/yourstore/categories/categories.html' in robots.txt returned a non-forbidden or redirect HTTP code (301)
+ Entry '/gp/vote/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/voting/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/music/wma-pop-up/' in robots.txt returned a non-forbidden or redirect HTTP code (1.29 seconds)
+ Entry '/gp/music/customize-reviews/comdu...' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/yourstore/categories/categories.html' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ Entry '/gp/item-dispatch/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/amusic/order/handle-buy-box.html' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ Entry '/gp/yourstore/categories/categories.html' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/product-availability/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/wishlist/vendor-button/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/wishlist/universal/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/wishlist/empty/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/review/common/du/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/gp/erc/mwl/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Server banner has changed from 'Server' to 'AkamaiHost' which may suggest a WAF, load balancer or proxy is in place
```

“Nikto -host” is used to specify the target host where the target is the website to scan. Here as example, the domain amazon is set as the target. Therefore “nikto -h <http://www.amazon.com>” is typed to run a basic scan against the target.

"-p" command could also be used here to specify which ports to examine. Nikto is able to scan single, multiple or range of ports anywhere between 1 - 1000. If port is not specified, it will only scan port 80 as default.

“nikto -h <http://www.amazon.com> -p 1-1000” command scans for a range of ports where as “nikto -h <http://www.amazon.com> -p80,443” scans multiple ports at once.

Amazon.de

```
root@kali:~# nikto -h https://amazon.de
Nikto v2.1.6
+ Target IP:      54.239.39.102
+ Target Hostname: amazon.de
+ Target Port:    443
+ SSL Info:       Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=*.peg.az2.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=DigiCert Inc/OU=www.DigiCert Global CA G2
+ Message:        Multiple IP addresses found: 54.239.39.102, 176.32.100.105, 52.95.120.34
+ Start Time:     2020-10-14 14:17:29 (GMT-4)
+ Server: Server
+ 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 site uses SSL and the Strict-Transport-Security header is not defined.
+ The site uses SSL and the 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
+ Root page / redirects to: https://www.amazon.de/
+ No CGI Directories Found (use -c all' to force check all possible dirs)
+ Server is using a wildcard certificate: *.peg.az2.com
+ Hostname 'amazon.de' does not match certificate's names: *.peg.az2.com
+ 7997 requests: 5 error(s) and 7 item(s) reported on remote host
+ End Time:       2020-10-14 17:30:02 (GMT-4) (11553 seconds)

+ 1 hour(s) tested
root@kali:~#
```

Amazon.co.uk

```
root@kali:~/Sublist3r# nikto -h https://amazon.co.uk
Nikto v2.1.6
+ Target IP:      54.239.34.171
+ Target Hostname: amazon.co.uk
+ Target Port:    443
+ SSL Info:       Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=*.peg.az2.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=DigiCert Inc/OU=www.DigiCert Global CA G2
+ Message:        Multiple IP addresses found: 54.239.34.171, 178.236.7.220, 54.239.33.58
+ Start Time:     2020-10-14 13:14:05 (GMT-4)
+ Server: Server
+ 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 site uses SSL and the Strict-Transport-Security header is not defined.
+ The site uses SSL and the 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
+ Root page / redirects to: https://www.amazon.co.uk/
+ No CGI Directories Found (use -c all' to force check all possible dirs)
+ Server is using a wildcard certificate: *.peg.az2.com
+ Hostname 'amazon.co.uk' does not match certificate's names: *.peg.az2.com
+ 7869 requests: 3 error(s) and 7 item(s) reported on remote host
+ End Time:       2020-10-14 16:36:52 (GMT-4) (12167 seconds)

+ 1 hour(s) tested
root@kali:~#
```

Amazon.com.mx

```
root@kali:~# nikto -h https://amazon.com.mx
Nikto v2.1.6
+ Target IP: 52.94.225.241 Information gathering
+ Target Hostname: amazon.com.mx
+ Target Port: 443
+ SSL Info: Subject: /CN=*.da.peg.a2z.com
  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
  Issuer: /C=US/O=Amazon/OU=Server CA 1B/Ch=Amazon
+ Message: Multiple IP addresses found: 52.94.225.241, 54.239.18.174, 52.94.239.35
+ Start Time: 2020-10-14 15:37:00 (GMT-4)
+ Server: Server
+ 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 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
+ Root page / redirect to: https://www.amazon.com.mx/
+ No CGI Directories Found (use -C all to force check all possible dirs)
+ Server is using a wildcard certificate: *.da.peg.a2z.com
+ Hostname 'amazon.com.mx' does not match certificate's names: *.da.peg.a2z.com
[...]
```

Amazon.co.jp

```
root@kali:~# nikto -h https://amazon.co.jp
Nikto v2.1.6
+ Target IP: 52.119.168.48 Information gathering
+ Target Hostname: amazon.co.jp
+ Target Port: 443
+ SSL Info: Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=*.peg.a2z.com
  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
  Issuer: /C=US/O=DigiCert Inc/CN=DigiCert Global CA 62
+ Message: Multiple IP addresses found: 52.119.168.48, 52.119.161.5, 52.119.164.21
+ Start Time: 2020-10-14 14:18:57 (GMT-4)
+ Server: Server
+ 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 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
+ Root page / redirect to: https://www.amazon.co.jp/
+ No CGI Directories Found (use -C all to force check all possible dirs)
+ Server is using a wildcard certificate: *.peg.a2z.com
+ Hostname 'amazon.co.jp' does not match certificate's names: *.peg.a2z.com
+ 7873 requests: 4 error(s) and 7 item(s) reported on remote host
+ End Time: 2020-10-14 17:27:38 (GMT-4) (11801 seconds)
+ 1 host(s) tested
root@kali:~#
```

Amazon.fr

```
root@kali:~# nikto -h https://amazon.fr
Nikto v2.1.6
+ Target IP:      52.95.116.113
+ Target Hostname: amazon.fr
+ Target Port:    443
+ SSL Info:       Subject: /CN=*.cz.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.95.116.113, 54.239.33.91, 52.95.126.39
+ Start Time:    2020-10-14 14:18:12 (GMT-4)
+ End Time:      2020-10-14 17:31:30 (GMT-4)
+ Server: Server
+ 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 site uses SSL and the Strict-Transport-Security 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
+ No CGI Directories found (use -C all' to force check all possible dirs)
+ Server is using a wildcard certificate: *.cz.peg.a2z.com
+ Host header and certificate not matching. Certificate's names: *.cz.peg.a2z.com
+ EOROR Error limit (20) reached for host, giving up. Last error: opening stream: can't connect: SSL negotiation failed: at /var/lib/nikto/plugins/LW2.pm line 5157.
; Interrupted system call at /var/lib/nikto/plugins/LW2.pm line 5157.
; Interrupted system call at /var/lib/nikto/plugins/LW2.pm line 5157.
; Skipped 1 item(s)
+ Scan completed: 13 Arrois(s) and 7 item(s) reported on remote host
+ End Time:      2020-10-14 17:31:30 (GMT-4) (11598 seconds)
+ 1 host(s) tested
root@kali:~#
```

Amazon.in

```
root@kali:~# nikto -h https://amazon.in
Nikto v2.1.6
+ Target IP:      54.239.33.92
+ Target Hostname: amazon.in
+ Target Port:    443
+ SSL Info:       Subject: /CN=*.cy.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 54.239.33.92, 52.95.120.67, 52.95.116.1
+ Start Time:    2020-10-14 15:37:49 (GMT-4)
+ End Time:      2020-10-14 18:35:56 (GMT-4)
+ Server: Server
+ 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 site uses SSL and the Strict-Transport-Security 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
+ No CGI Directories found (use -C all' to force check all possible dirs)
+ Server is using a wildcard certificate: *.cy.peg.a2z.com
+ Host header and certificate not matching. Certificate's names: *.cy.peg.a2z.com
+ EOROR Error limit (20) reached for host, giving up. Last error: opening stream: can't connect: SSL negotiation failed: at /var/lib/nikto/plugins/LW2.pm line 5157.
; Interrupted system call at /var/lib/nikto/plugins/LW2.pm line 5157.
; Interrupted system call at /var/lib/nikto/plugins/LW2.pm line 5157.
; Skipped 1 item(s)
+ Scan completed: 13 Arrois(s) and 7 item(s) reported on remote host
+ End Time:      2020-10-14 18:35:56 (GMT-4) (10687 seconds)
+ 1 host(s) tested
root@kali:~#
```

Amazon.ca

```
root@kali:~# nikto -h https://amazon.ca
Nikto v2.1.6
+ Target IP: 52.94.225.242
+ Target Hostname: amazon.ca
+ Target Port: 443
+ SSL Info: Subject: /CN=*.bx.peg.a2z.com
  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
  Issuer: /C=US/O=Amazon/OU=Server CA 1B/Ch=Amazon
+ Message: Multiple IP addresses found: 52.94.225.242, 54.239.19.238, 54.239.18.172
+ Start Time: 2020-10-14 15:38:01 (GMT-4)
+ End Time: 2020-10-14 19:18:54 (GMT-4)

+ Server: Server
+ 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 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 Content-Type 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.amazon.ca/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Script engines: None
+ Hostname 'amazon.ca' does not match certificate's names: *.bx.peg.a2z.com
+ 7888 requests: 13 error(s) and 7 item(s) reported on remote host
+ End Time: 2020-10-14 19:18:54 (GMT-4) (12773 seconds)

+ 1 host(s) tested
root@kali:~#
```

Amazon.es

```
root@kali:~# nikto -h https://amazon.es
Nikto v2.1.6
+ Target IP: 52.95.116.112
+ Target Hostname: amazon.es
+ Target Port: 443
+ SSL Info: Subject: /CN=*.bw.peg.a2z.com
  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
  Issuer: /C=US/O=Amazon/OU=Server CA 1B/Ch=Amazon
+ Message: Multiple IP addresses found: 52.95.116.112, 54.239.33.98, 52.95.120.38
+ Start Time: 2020-10-14 15:37:39 (GMT-4)
+ End Time: 2020-10-14 19:01:54 (GMT-4)

+ Server: Server
+ 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 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 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.amazon.es/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server is using a wildcard certificate: *.bw.peg.a2z.com
+ Hostname 'amazon.es' does not match certificate's name: *.bw.peg.a2z.com
+ 7888 requests: 13 error(s) and 7 item(s) reported on remote host
+ End Time: 2020-10-14 19:01:54 (GMT-4) (12255 seconds)

+ 1 host(s) tested
root@kali:~#
```

Fingerprinting

Fingerprinting also known as foot printing's purpose is to accumulate as much information as possible regarding a target web application. It is a set of information that can identify network protocols, operating systems, hardware and software etc.

This process is deployed as a security measure to authenticate users however attackers may exploit this to spot vulnerabilities within the target web application. This provide information such as operating system types and versions, domain names, network blocks, VPN point etc.

In order to gather information a custom packet set is launched and when they receive a response from the target, information regarding operating systems, protocols and other are withdrawn.

There are 2 main types of fingerprinting, they are passive and active. Active fingerprinting consists of sending packets to the target and wait for it to reply then analyzing the response. Passive fingerprinting consists of monitoring the target's network traffic without any direct involvements.

Tools that could be used to do fingerprinting are;

- Nmap
- Ettercap
- PacketFence
- Netcat

Nmap

Nmap, also known as Network Mapper, is a comprehensive active stack fingerprinting tool. It identifies what devices are running on the targeted system, discover host that are available, find open ports, detect security risks and more.

Nmap can be used to scan singular host as well as a network that has many devices and subnets. Nmap gathers information by sending raw data packets to system ports and determines whether the posts are open or closed or filtered by a firewall or other.

“nmap <http://www.amazon.com>” (domain name/ IP address) is one of the basic commands of nmap that scans a single IP address whereas “nmap <http://www.amazon.com> -p 20-3000” scans a range of ports in nmap.

In this project, “nmap -p80,443 -A -T4 (IP address)” command has been used. The “-A” command orders to perform an OS check and a version check and “-T4” is a speed template that tells the nmap how to perform the scan faster. This method can be seen has very aggressive and obtrusive than usual nmap commands. It also scans multiple ports (port 80 and 443) at once.

Amazon.com

```
root@kali:~# msf -p80,443 -A -Tq 178.32.98.166
Starting port scan on https://178.32.98.166
Scan report for 178.32.98.166
Host is up (0.33s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http   Server
| fingerprint-strings:
|_ Found0FourRequests:
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Fri, 16 Oct 2020 12:16:15 GMT
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Location: https://nicewebs%20ports%2C/Tri%20City.txt%2ebak
<html>
<head><title>301 Moved Permanently</title></head>
<body bcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<br><br><center>Server</center>
</body>
</html>
GetRequest:
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Fri, 16 Oct 2020 12:16:08 GMT
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body bcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<br><br><center>Server</center>
</body>
</html>
HTTPOptions:
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Fri, 16 Oct 2020 12:16:09 GMT
Content-Type: text/html
Content-Language: en
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body bcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<br><br><center>Server</center>
</body>
</html>
SIPOptions:
HTTP/1.1 400 Bad Request
Server: Server
Date: Fri, 16 Oct 2020 12:16:49 GMT
Content-Type: text/html
Content-Length: 167
```

Amazon.de

Amazon.co.uk

Amazon.com.mx

```
[root@kali: ~]# nmap -p80,443 -A -T4 52.94.225.241
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 18:03 EDT
Nmap scan report for 52.94.225.241
Host is up (0.31s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Server
          Fingerprint string:
          FingerprintRequest:
HTTP/1.1 301 Moved Permanently
Server: Apache/2.4.38 (Ubuntu)
Date: Wed, 14 Oct 2020 12:33:50 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
Location: https://nice20ports2C/TriMEtTy.txt2ebak
.html
<head><title>301 Moved Permanently</title></head>
<body>bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>Server</center>
</body>
</html>
GetRequest:
HTTP/1.1 301 Moved Permanently
Server: Apache/2.4.38 (Ubuntu)
Date: Wed, 14 Oct 2020 12:33:42 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body>bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>Server</center>
</body>
</html>
HTTPOptions:
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Wed, 14 Oct 2020 12:33:43 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body>bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>Server</center>
</body>
</html>
SIPOptions:
HTTP/1.1 400 Bad Request
Server: Server
Date: Wed, 14 Oct 2020 12:34:25 GMT
Content-Type: text/html
Content-Length: 167
```

Amazon.co.jp

```
root@kali:~ # nmap -p80,443 -A -T4 52.19.168.48
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 14:25 EDT
Nmap scan report for 52.19.168.48
Host is up (0.24s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Server
fingerprint-string:
  fourOhFourRequest:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:55:52 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://nice%20ports%2C/Tri%6Eity.txt%2ebak
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  GetRequest:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:55:45 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body bgcolor="white">
    <center><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  HTTPOptions:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:55:46 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body bgcolor="white">
    <center><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  SHTTP:
    HTTP/1.1 400 Bad Request
    Server: Server
    Date: Wed, 14 Oct 2020 08:56:25 GMT
    Content-Type: text/html
    Content-Length: 167
```

Amazon.fr

```
root@kali:~ # nmap -p80,443 -A -T4 52.95.116.113
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 14:26 EDT
Nmap scan report for 52.95.116.113
Host is up (0.28s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Server
fingerprint-string:
  fourOhFourRequest:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:56:41 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://nice%20ports%2C/Tri%6Eity.txt%2ebak
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body bgcolor="white">
    <center><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  GetRequest:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:56:34 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body bgcolor="white">
    <center><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  HTTPOptions:
    HTTP/1.1 301 Moved Permanently
    Server: Server
    Date: Wed, 14 Oct 2020 08:56:35 GMT
    Content-Type: text/html
    Content-Length: 179
    Connection: keep-alive
    Location: https://
    <html>
    <head><title>301 Moved Permanently</title></head>
    <body bgcolor="white">
    <center><h1>Moved Permanently</h1></center>
    <br><center>Server</center>
    </body>
    </html>
  SHTTP:
    HTTP/1.1 400 Bad Request
    Server: Server
    Date: Wed, 14 Oct 2020 08:57:14 GMT
    Content-Type: text/html
    Content-Length: 167
```

Amazon.in

```
root@kali: ~ nmap -p80,443 -A -T4 54.239.33.92
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 19:00 EDT
Nmap scan report for 54.239.33.92
Host is up (0.27s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Server
          fingerprint-string:
FourOhFourRequest:
  HTTP/1.1 301 Moved Permanently
  Server: Apache
  Date: Wed, 14 Oct 2020 13:30:42 GMT
  Content-Type: text/html
  Content-Length: 179
  Connection: keep-alive
  Location: https://nice20ports2c/TriMe6Eity.txt%2ebak
  <html>
  <head><title>301 Moved Permanently</title></head>
  <body><h1>Moved Permanently</h1></center>
  <br><center>Server</center>
  </body>
  </html>
GetRequest, HTTPOptions:
  HTTP/1.1 301 Moved Permanently
  Server: Server
  Date: Wed, 14 Oct 2020 13:30:35 GMT
  Content-Type: text/html
  Content-Length: 179
  Connection: keep-alive
  Location: https://
  <html>
  <head><title>301 Moved Permanently</title></head>
  <body>bgcolor="white">
  <body><h1>Moved Permanently</h1></center>
  <br><center>Server</center>
  </body>
  </html>
SIPOptions:
  HTTP/1.1 400 Bad Request
  Server: Server
  Date: Wed, 14 Oct 2020 13:31:13 GMT
  Content-Type: text/html
  Content-Length: 167
  Connection: close
  <head><title>400 Bad Request</title></head>
  <body>bgcolor="white">
  <center><h1>400 Bad Request</h1></center>
  <br><center>Server</center>
  </body>
  </html>
HTTP-server-header: Server
HTTP-title: Did not follow redirect to https://54.239.33.92/
Https-redir: ERROR: Script execution failed (use -d to debug)
443/tcp open  ssl/http  Server
fingerprint-string:
FourOhFourRequest:
  HTTP/1.1 400 Bad Request
```

Amazon.ca

```
root@kali: ~ nmap -p80,443 -A -T4 52.94.225.242
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 19:15 EDT
Nmap scan report for 52.94.225.242
Host is up (0.32s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Server
          fingerprint-string:
FourOhFourRequest:
  HTTP/1.1 301 Moved Permanently
  Server: Server
  Date: Wed, 14 Oct 2020 13:46:00 GMT
  Content-Type: text/html
  Content-Length: 179
  Connection: keep-alive
  Location: https://nice20ports2c/TriMe6Eity.txt%2ebak
  <html>
  <head><title>301 Moved Permanently</title></head>
  <body>bgcolor="white">
  <body><h1>Moved Permanently</h1></center>
  <br><center>Server</center>
  </body>
  </html>
GetRequest, HTTPOptions:
  HTTP/1.1 301 Moved Permanently
  Server: Server
  Date: Wed, 14 Oct 2020 13:45:53 GMT
  Content-Type: text/html
  Content-Length: 179
  Connection: keep-alive
  Location: https://
  <html>
  <head><title>301 Moved Permanently</title></head>
  <body>bgcolor="white">
  <body><h1>Moved Permanently</h1></center>
  <br><center>Server</center>
  </body>
  </html>
SIPOptions:
  HTTP/1.1 400 Bad Request
  Server: Server
  Date: Wed, 14 Oct 2020 13:46:33 GMT
  Content-Type: text/html
  Content-Length: 167
  Connection: close
  <head><title>400 Bad Request</title></head>
  <body>bgcolor="white">
  <center><h1>400 Bad Request</h1></center>
  <br><center>Server</center>
  </body>
  </html>
HTTP-server-header: Server
HTTP-title: Did not follow redirect to https://52.94.225.242/
Https-redir: ERROR: Script execution failed (use -d to debug)
443/tcp open  ssl/http  Server
fingerprint-string:
FourOhFourRequest:
  HTTP/1.1 400 Bad Request
```

Amazon.es

```
root@kali: ~ # nmap -p80,443 -A -T4 52.95.116.112
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 23:17 EDT
Segmentation fault
root@kali: ~ # nmap -p80,443 -A -T4 52.95.116.112
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 23:23 EDT
Nmap scan report for 52.95.116.112
Host is up (0.27s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http   Server
          fin-ack-syn-ack
          fin-ack-request:
          HTTP/1.1 301 Moved Permanently
          Server: Server
          Date: Wed, 14 Oct 2020 17:54:04 GMT
          Content-Type: text/html
          Content-Length: 179
          Connection: keep-alive
          Location: https://nice20ports2C/TrixE6ity.txt%2ebak
          <html>
          <head><title>301 Moved Permanently</title></head>
          <body bgcolor="white">
          <center><h1>301 Moved Permanently</h1></center>
          <br><br><center>Server</center>
          </body>
          </html>
          <!-->
          <!-->
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Wed, 14 Oct 2020 17:53:55 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<br><br><center>Server</center>
</body>
</html>
HTTP/1.1 301 Moved Permanently
Server: Server
Date: Wed, 14 Oct 2020 17:53:57 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
Location: https://
<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<br><br><center>Server</center>
</body>
</html>
<!-->
<!-->
HTTP/1.1 400 Bad Request
Server: Server
          <!-->
```

Vulnerabilities

Vulnerability is a weakness within a system that can be exploited during a cyber attack to perform unauthorized actions or to gain access. When such weakness is exploited, the system loses its security which damages the CIA triad.

The CIA triad includes:

- Confidentiality – Privileged information is presented to only those who have clearance
- Integrity – Protects data from being unauthorized modified
- Availability – ensuring that data is available to customer when they need it

In order to violate the security of one's system, the attackers must launch an attack that damages the exterior of the CIA triad. Such attacks are phishing or wiretapping to damage confidentiality, DoS/DDoS attacks to damage availability or man in the middle attack to damage integrity feature of the system.

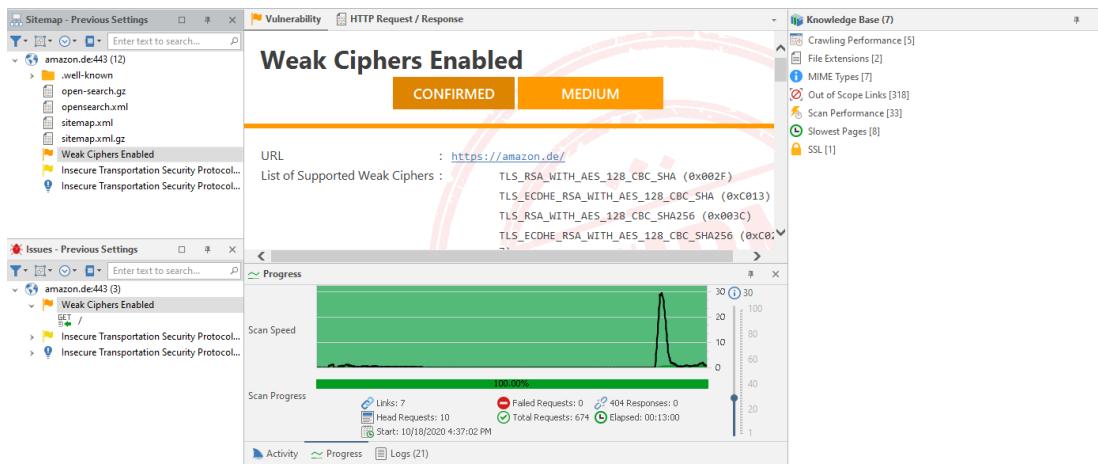
Such attacks are launched using a vulnerability that is hidden within the system itself. Therefore, such vulnerabilities must be found and patched in a routine checkup.

Here, using Netsparker and nikto all chosen subdomains were scanned for vulnerabilities within them.

Amazon.de

Weak Ciphers Enabled

Severity: Medium



Weak cipher is an encryption-decryption algorithm that uses a key of insufficient length. This opens up the probability of the encryption scheme being broken. The impact of using weak ciphers is that the attackers may be able to decrypt SSL traffic between the server and the visitor.

Weak ciphers are encryption decryption algorithms that uses less than 128 bits key sizes as their value

List of Supported Weak Ciphers :

TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)

The risk level of this vulnerability could be identified as medium. Configuring the web server to disallow using weak ciphers can be recognized as the remedy. In order to be stronger, the cipher key value should be larger.

Discovering and removing weak ciphers

one method of finding weak ciphers is to utilize a scanner such as netsparker. Another way is to type command “`openssl s_client – connect`

<hostname:port> – cipher" where the interface will contact the host/port and discover the lowest security cipher supported. The returned ciphers must be removed from the system as soon as possible.

Weak ciphers are supported by most web server applications even in this day and age therefore old web-based client software need to be up to date in order to be secured. Programs that can exploit weak ciphers can be created or acquired very easily.

Actions to Take

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

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

Lighttpd:

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

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 `regedit32` 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
```

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low

The screenshot shows the Netsparker interface with the following details:

- Vulnerability:** Insecure Transportation Security Protocol Supported (TLS 1.0)
Status: CONFIRMED | LOW
- URL:** <https://amazon.de/>
- Classification:**
 - PCI DSS 3.2: 6.5.4
 - OWASP 2013: A6
 - OWASP 2017: A3
 - CWE: 326
 - CAPEC: 217
 - WASC: 4
- Progress:** Scan Speed: 100.00%, Scan Progress: 100.00%
- Tools:** Retest, Generate Exploit, Execute SQL Commands, Get Shell, Exploit LFI, Exploit Short Names, Ignore from this Scan, Configure Send To Actions..., Configure Web Application Firewall..., WAF Rules.
- Knowledge Base:** Crawling Performance [5], File Extensions [2], MIME Types [7], Out of Scope Links [318], Scan Performance [33], Slowest Pages [8], SSL [1].

TSL or Transport Layer Security protocol provide privacy and data security for communication over the internet by primarily encrypting the communication between web applications a server. TSL can also be used to encrypt emails, messaging and voice over IP.

However, it was deemed that TSL 1.0 will be no longer be used for secure communication as it was vulnerable for many attacks including man-in-the-middle-attacks which risk the integrity and authentication of data sent between the website and the browser.

Man-in-the-middle attack

An attacker is placed between a conversation of user and application in order to collect data that would normally would be encrypted.

How to fix this issue

Web server operators should disable TSL 1.0 and replace it with TSL 1.2 or above version.

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.

Click on Start and then Run, type regedit32 or regedit, and then click OK.

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\
```

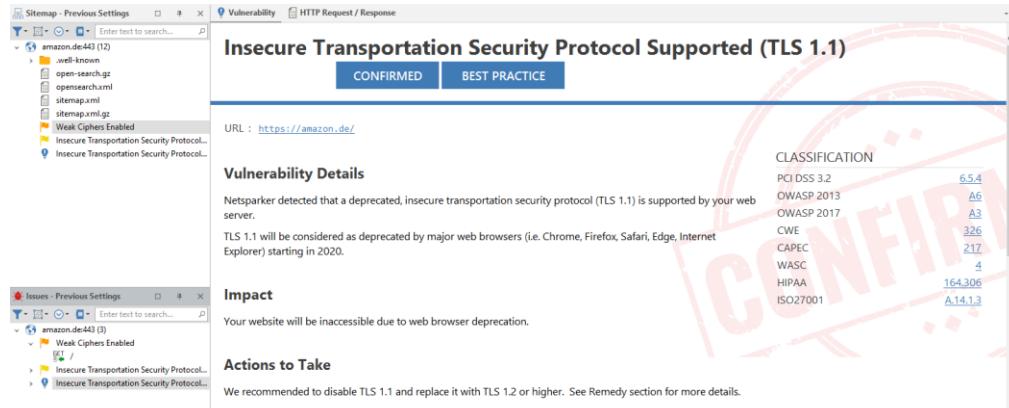
Locate a key named Server or create if it doesn't exist.

Under the Server key, 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"
```

Why it's TSL 1.2 or above and not TSL 1.1



The screenshot shows the Netsparker Security Scanner interface. The main window displays a 'Vulnerability' report for 'Insecure Transportation Security Protocol Supported (TLS 1.1)'. The status is 'CONFIRMED'. The URL is https://amazon.de/. The report details that TLS 1.1 is supported and will be deprecated by major web browsers starting in 2020. It also notes that the website will be inaccessible due to browser deprecation. The 'Actions to Take' section recommends disabling TLS 1.1 and replacing it with TLS 1.2 or higher. On the left, there are two side-by-side 'Issues' panes showing findings for 'amazon.de:443' and 'amazon.de:443'. A large red 'CONFIRMED' stamp is overlaid on the right side of the main report area.

TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

Amazon.co.uk

Missing X-XXS protection header

Severity: Best Practice

```
root@kali:~# nikto -h https://amazon.co.uk
- Nikto v2.1.6
-----
+ Target IP:      54.239.34.171
+ Target Hostname: amazon.co.uk
+ Target Port:    443
-----
+ SSL Info:       Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=*.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=DigiCert Inc/CN=DigiCert Global CA G2
+ Message:        Multiple IP addresses found: 54.239.34.171, 178.236.7.220, 54.239.33.58
+ Start Time:     2020-10-14 13:14:05 (GMT-4)
-----
+ Server: Server
+ The X-XSS-Protection 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 site uses SSL and Expect-CT header is not present.
```

Security headers are fundamentally necessary for website security to protect itself from many kinds of attacks such as XXS, code injections and more.

X-XXS protection header is set as a built-in web browser cross-site scripting filter meaning the header stops from loading when detecting reflected cross-site scripting attacks. When this header is missing from a website it is vulnerable for XXS attacks.

X-XSS-Protection: 0

X-XSS-Protection: 1

X-XSS-Protection: 1; mode=block

X-XSS-Protection: 1; report=<reporting-uri>

In XXS attacks, the attacker injects malicious scripts into a vulnerable website. There are two types of XXS attacks that could occur. They are: Server XXS and Client XXS

Impact of XXS attacks can be vary from attack to attack but most common ones are theft of private data such as cookies or session information and more.

How to fix this issue

Add X-XXS protection header with value “1; mode=block”

Solution:

Configure this header for the web application ensuring correct values are set.

X-XSS-Protection: 1; mode=block

PHP

```
header("X-XSS-Protection: 1; mode=block");
```

Apache (.htaccess)

```
<IfModule mod_headers.c>
```

```
Header set X-XSS-Protection "1; mode=block"
```

```
</IfModule>
```

Nginx

```
add_header "X-XSS-Protection" "1; mode=block";
```

The anti-clickjacking X-frame-option header is not present

Severity: Low

```
root@kali:~# nikto -h https://amazon.co.uk
- Nikto v2.1.6

-----
+ Target IP:      54.239.34.171
+ Target Hostname: amazon.co.uk
+ Target Port:    443

-----
+ SSL Info:       Subject: /C=US/ST=Washington/L=Seattle/O=Amazon.com, Inc./CN=*.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=DigiCert Inc/CN=DigiCert Global CA G2
+ Message:        Multiple IP addresses found: 54.239.34.171, 178.236.7.220, 54.239.33.58
+ Start Time:     2020-10-14 13:14:05 (GMT-4)

-----
+ Server: Server
+ 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 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-frame-option header indicates whether or not the browser should allow to render a page within <frame>, <iframe>, <embed>, <object>. This header protects the website from clickjacking attacks.

Clickjacking Attacks

Clickjacking attacks is a malicious way of tricking the user into clicking on item that is disguised as item user believes to click into. The hidden page/item can cause users to visit malicious web pages or download malware or a legitimate page that user did not intended to visit.

As the user believes they are on the site they intended to visit, user can provide sensitive information to those sites, transfer money or purchase products etc. Likejacking and cursorjacking are few well known clickjacking attacks.

How to fix this issue

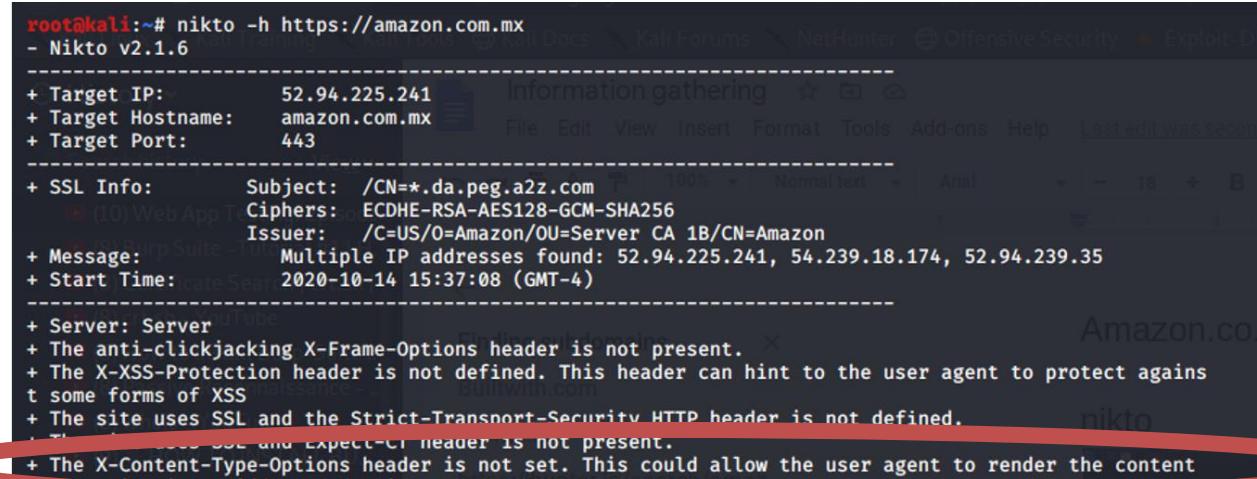
There are three values for the x-frame option header that instruct the browser to allow/disallow framing from other domains:

- DENY – does not allow any domain to display the page
- SAMEORIGIN – allows the page to display on another page but within the current domain
- ALLOW-FROM URI – allows the page to display in a frame but only in a specific frame

Amazon.com.mx

The X-content-type-options header is not set

Severity: Low



root@kali:~# nikto -h https://amazon.com.mx
- Nikto v2.1.6

+ Target IP: 52.94.225.241 Information gathering ☆ 📁 🌐
+ Target Hostname: amazon.com.mx File Edit View Insert Format Tools Add-ons Help Last edit was second ago
+ Target Port: 443

+ SSL Info: Subject: /CN=*.da.peg.a2z.com
Ciphers: ECDHE-RSA-AES128-GCM-SHA256
Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message: Multiple IP addresses found: 52.94.225.241, 54.239.18.174, 52.94.239.35
+ Start Time: 2020-10-14 15:37:08 (GMT-4)

+ Server: Server Amazon.co
+ The anti-clickjacking X-Frame-Options header is not present. nikto
+ The X-XSS-Protection 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 Content-Security-Policy 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

This header is used by servers to indicate that the MIME types that are presented in content type headers should not be changed and followed. This prevent the browser from MIME type sniffing attacks.

If the server does not return a correct x-content-type-options header, the website would be at risk of XXS attacks.

MIME type sniffing

MIME type sniffing is a standard functionality of browsers to find a suitable way of render data, when the HTTP headers sent by the servers are inconclusive or missing. This allows the web browser to perform a MIME sniffing response causing the response to interpret and display as a different type of content.

If the website allows users to upload content which then publish on the web server and an attacker carry out a XXS attack to manipulate the content

and to render the content as the HTML browser, it is possible to inject malicious code and make the victim execute it.

How to fix this issue.

Remediation

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:

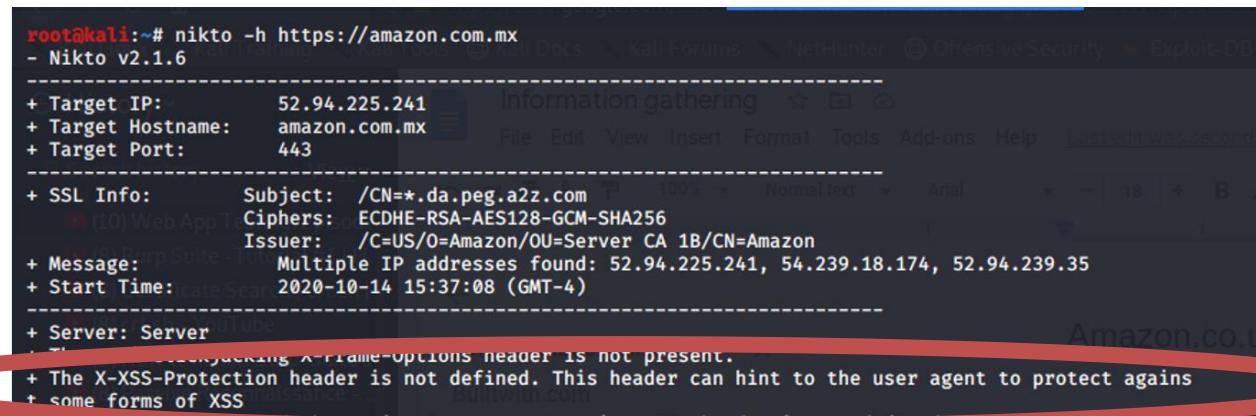
```
Content-Type: text/html
```

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
```

Missing X-XXS protection header

Severity: Best Practice



The screenshot shows the terminal output of the Nikto web scanner against the Amazon.com.mx website. A red oval highlights the final two lines of the output:

```
root@kali:~# nikto -h https://amazon.com.mx
- Nikto v2.1.6
-----
+ Target IP:      52.94.225.241
+ Target Hostname: amazon.com.mx
+ Target Port:    443
-----
+ SSL Info:       Subject: /CN=*.da.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.94.225.241, 54.239.18.174, 52.94.239.35
+ Start Time:     2020-10-14 15:37:08 (GMT-4)
-----
+ Server: Server
+ The X-XSS-Protection header is not present.
+ The 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
```

X-XXS protection header is set as a built-in web browser cross-site scripting filter meaning the header stops from loading when detecting reflected cross-site scripting attacks. When this header is missing from a website it is vulnerable for XSS attacks.

X-XSS-Protection: 0
X-XSS-Protection: 1
X-XSS-Protection: 1; mode=block
X-XSS-Protection: 1; report=<reporting-uri>

Cross-Site Scripting (XXS)

The attacker injects malicious scripts into a vulnerable website. Impact of XSS attacks can be vary from attack to attack but most common ones are theft of private data such as cookies or session information and more.

How to fix this issue

Add X-XXS protection header with value “1; mode=block”

Solution:

Configure this header for the web application ensuring correct values are set.

X-XSS-Protection: 1; mode=block

PHP

```
header("X-XSS-Protection: 1; mode=block");
```

Apache (.htaccess)

```
<IfModule mod_headers.c>
```

```
Header set X-XSS-Protection "1; mode=block"
```

```
</IfModule>
```

Nginx

```
add_header "X-XSS-Protection" "1; mode=block";
```

Expect – CT header is not defined

Severity: Best Practice

```
root@kali:~# nikto -h https://amazon.com.mx
- Nikto v2.1.6
-----
+ Target IP:      52.94.225.241
+ Target Hostname: amazon.com.mx
+ Target Port:    443
-----  
Information gathering
File Edit View Insert Format Tools Add-ons Help Last edit was seconds ago
+ SSL Info:       Subject: /CN=*.da.peg.a2z.com
+ SSL Info:       Ciphers: ECDHE-RSA-AES128-GCM-SHA256
+ SSL Info:       Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.94.225.241, 54.239.18.174, 52.94.239.35
+ Start Time:     2020-10-14 15:37:08 (GMT-4)
-----  
+ Server: Server
+ 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 Content-Security-Policy header is not defined.
+ The Strict-Transport-Security header is not defined.
+ The site uses SSL and Expect-CT header is not present.
```

Certificate transparency, shortly known as CT is a technology that makes it very difficult for a certificate authority to issue an SSL certificate for a domain without the certificate being visible to the owner of the domain.

In 2018 Google announced that if it finds a website with a certificate that is not in the CT log it will consider the certificate invalid and reject the connection.

How to fix this issue

Remediation

Configure your web server to respond with Expect-CT header.

Expect-CT: enforce, max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"

Note: We strongly suggest you to use Expect-CT header in **report-only mode** first. If everything goes well and your certificate is ready, go with the Expect-CT enforce mode. To use **report-only mode** first, omit **enforce** flag and see the browser's behavior with your deployed certificate.

Expect-CT: max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"

Amazon.co.jp

Weak Ciphers Enabled

Severity: Medium

Weak Ciphers Enabled

CONFIRMED MEDIUM

URL : <https://amazon.co.jp/>

List of Supported Weak Ciphers :

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC813)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC827)

Vulnerability Details

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.

CLASSIFICATION

PCI DSS 3.2	6.54
OWASP 2013	A6
OWASP 2017	A3
CWE	327
CAPEC	217
WASC	4
ISO27001	A14.1.3

CVSS 3.0 SCORE

Actions to Take

Progress

Scan Speed

Weak cipher is an encryption-decryption algorithm that uses a key of insufficient length. This opens up the probability of the encryption scheme being broken. The impact of using weak ciphers is that the attackers may be able to decrypt SSL traffic between the server and the visitor.

Weak ciphers are encryption decryption algorithms that uses less than 128 bits key sizes as their value.

How to fix this issue

Configuring the web server to disallow using weak ciphers can be recognized as the remedy. In order to be stronger, the cipher key value should be larger.

Actions to Take

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

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

Lighttpd:

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

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 `regedit32` 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
```

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low

The screenshot shows the Netsparker web application security scanner interface. A prominent red watermark with the word "CONFIRMED" is overlaid on the page. The main content area displays a "Vulnerability" card for "Insecure Transportation Security Protocol (TLS 1.0)". The status is "CONFIRMED" and the severity is "LOW". The URL is listed as <https://amazon.co.jp/>. The "Vulnerability Details" section states: "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." The "Impact" section notes: "Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors." The "Actions to Take" section recommends: "We recommend to disable TLS 1.0 and replace it with TLS 1.2 or higher. See Remedy section for more details." On the right side, there is a "CLASSIFICATION" table comparing the vulnerability across various frameworks and standards:

	PCI DSS 3.2	OWASP 2013	OWASP 2017	CWE	CAPEC	WASC	HIPAA	ISO27001
	6.5.4	A6	A3	326	217	4	164.306	A.14.1.3

TSL can also be used to encrypt emails, messaging and voice over IP. However, it was deemed that TSL 1.0 will be no longer be used for secure communication as it was vulnerable for many attacks including man-in-the-middle-attacks which risk the integrity and authentication of data sent between the website and the browser.

Man-in-the-middle-attacks are where the attacker is placed between a conversation of user and application in order to collect data that would normally would be encrypted.

How to fix this issue

Web server operators should disable TSL 1.0 and replace it with TSL 1.2 or above version.

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.

Click on Start and then Run, type regedit32 or regedit, and then click OK.

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\
```

Locate a key named Server or create if it doesn't exist.

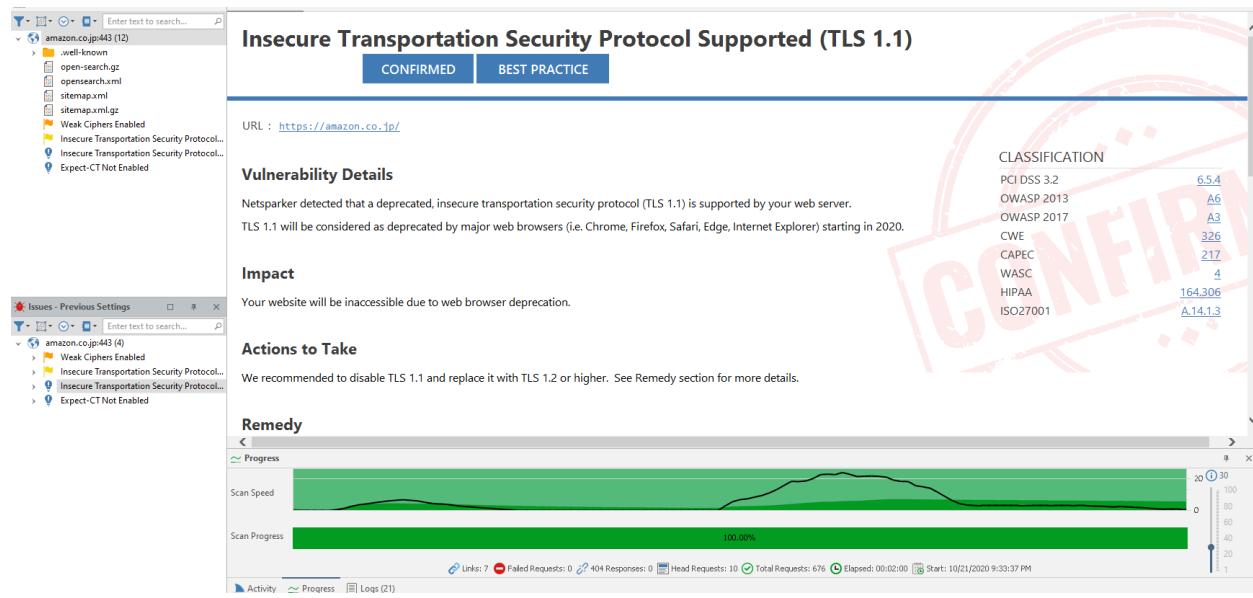
Under the Server key, locate a DWORD value named Enabled or create if it doesn't exist and set its value to "0".

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"
```

Insecure Transport Security Protocol Supported (TSL 1.1)

Severity: Best Practice



TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

How to fix this issue

Discontinue using TSL 1.1 and replace it with TSL 1.2 or higher protocol

Amazon.fr

Weak Ciphers Enabled

Severity: Medium

The screenshot shows the Netsparker interface with the following details:

- Vulnerability Details:** Weak Ciphers Enabled (CONFIRMED, MEDIUM)
- URL:** <https://amazon.fr/>
- List of Supported Weak Ciphers:**
 - TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
 - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
 - TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
 - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)
- Vulnerability Details:** 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.
- Classification:**
 - PCI DSS 3.2: 6.5.4
 - OWASP 2013: A6
 - OWASP 2017: A3
 - CWE: 327
 - CAPEC: 217
 - WASC: 4
 - ISO27001: A.14.1.3
- CVSS 3.0 SCORE:** 6.5.4

Weak cipher is an encryption-decryption algorithm that uses a key of insufficient length. This opens up the probability of the encryption scheme being broken. The impact of using weak ciphers is that the attackers may be able to decrypt SSL traffic between the server and the visitor.

Weak ciphers are encryption decryption algorithms that uses less than 128 bits key sizes as their value.

How to fix this issue

Configuring the web server to disallow using weak ciphers can be recognized as the remedy. In order to be stronger, the cipher key value should be larger.

Actions to Take

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

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

Lighttpd:

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

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 `regedit32` 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
```

Strict transport security HTTP header not defined

Severity: Medium

```
root@kali:~# nikto -h https://amazon.fr
- Nikto v2.1.6
-----
+ Target IP:      52.95.116.113
+ Target Hostname: amazon.fr
+ Target Port:    443
-----
+ SSL Info:       Subject: /CN=*.cz.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.95.116.113, 54.239.33.91, 52.95.120.39
+ Start Time:     2020-10-14 14:18:12 (GMT-4)
-----
+ Server: Server
+ 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 site uses SSL and the Strict-Transport-Security HTTP header is not defined. (This is a critical issue)
+ The site uses SSL and the Expect-CT header is not present.
+ The X-Content-Type-Options header is not present.
```

HSTS, otherwise known as HTTP Strict Transport Security is a web security policy where the server declares that complying user agents such as web browser are to interact with it only using secure HTTPS connections. The HSTS policy communicated to the user is via the strict transport security header. This policy specifies a time period where the user can access the server only in a secure fashion.

This policy automatically turns any insecure HTTP links to secure links and if the security of a connection could not be ensured, an error message is displayed and user is not allowed to access the web application.

How to fix this issue

Remediation

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
    ServerAlias *
    RewriteEngine On
    RewriteRule ^(.*)$ https:// %{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
    # Use HTTP Strict Transport Security to force client to use secure connections only
    Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

    # Further Configuration goes here
    [...]
</VirtualHost>
```

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low

Vulnerability Details

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.

Actions to Take

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

Classification

Classification	Score
PCI DSS 3.2	65.4
OWASP 2013	A6
OWASP 2017	A3
CWE	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A14.1.3

TSL can also be used to encrypt emails, messaging and voice over IP. However, it was deemed that TSL 1.0 will be no longer be used for secure communication as it was vulnerable for many attacks including man-in-the-middle-attacks which risk the integrity and authentication of data sent between the website and the browser.

Man-in-the-middle-attacks are where the attacker is placed between a conversation of user and application in order to collect data that would normally would be encrypted.

How to fix this issue

Web server operators should disable TSL 1.0 and replace it with TSL 1.2 or above version.

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.

Click on Start and then Run, type regedit32 or regedit, and then click OK.

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\
```

Locate a key named Server or create if it doesn't exist.

Under the Server key, 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"
```

Insecure Transport Security Protocol Supported (TSL 1.1)

Severity: Best Practice

The screenshot shows the Netsparker interface with two tabs: 'Sitemap - Previous Settings' and 'Vulnerability'. The 'Vulnerability' tab is active, displaying the following details:

Vulnerability Details
Insecure Transportation Security Protocol Supported (TLS 1.1)
CONFIRMED **BEST PRACTICE**
URL : <https://amazon.fr/>

Classification

Standard	Score
PCI DSS 3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
CWE	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A.14.1.3

Impact
Your website will be inaccessible due to web browser deprecation.

Actions to Take
We recommend to disable TLS 1.1 and replace it with TLS 1.2 or higher. See Remedy section for more details.

Remedy

TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

How to fix this issue

Discontinue using TSL 1.1 and replace it with TSL 1.2 or higher protocol

Amazon.in

The X-content-type-options header is not set

Severity: Low

```
root@kali:~# nikto -h https://amazon.in
- Nikto v2.1.6
-----
+ Target IP:      54.239.33.92
+ Target Hostname: amazon.in
+ Target Port:     443
-----
+ SSL Info:       Subject: /CN=*.cy.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 54.239.33.92, 52.95.120.67, 52.95.116.139
15 Start Time:   2020-10-14 15:18:32 (GMT-4)
+ Start Time:     2020-10-14 15:37:49 (GMT-4)
-----
+ Server: Server jacking X-Frame-Options header is not present.
+ The anti-clickjacking X-Frame-Options header is not present. hint to the user agent to protect against
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to pr
otection against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site does not expect or header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render
```

This header is used by servers to indicate that the MIME types that are presented in content type headers should not be changed and followed. This prevent the browser from MIME type sniffing attacks.

If the server does not return a correct x-content-type-options header, the website would be at risk of XXS attacks.

MIME type sniffing

MIME type sniffing is a standard functionality of browsers to find a suitable way of render data, when the HTTP headers sent by the servers are inconclusive or missing. This allows the web browser to perform a MIME sniffing response causing the response to interpret and display as a different type of content.

If the website allows users to upload content which then publish on the web server and an attacker carry out a XXS attack to manipulate the content

and to render the content as the HTML browser, it is possible to inject malicious code and make the victim execute it.

How to fix this issue.

Remediation

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
```

Expect – CT header is not defined

Severity: Best Practice

CWE : 16
WASC : 15
ISO27001 :
A.14.1.2

Certificate transparency, shortly known as CT is a technology that makes very difficult for a certificate authority to issue an SSL certificate for a domain without the certificate being visible to the owner of the domain.

In 2018 Google announce that if it finds a website with a certificate that is not in the CT log it will consider the certificate is invalid and reject the connection.

How to fix this issue

Expect-CT: enforce, max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"

Expect-CT: max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"

Note: We strongly suggest you to use Expect-CT header in **report-only mode** first. If everything goes well and your certificate is ready, go with the Expect-CT enforce mode. To use **report-only mode** first, omit **enforce** flag and see the browser's behavior with your deployed certificate.

Weak Ciphers Enabled

Severity: Medium

Weak Ciphers Enabled

CONFIRMED MEDIUM

URL : <https://amazon.in/>

List of Supported Weak Ciphers :

- TLS_RSA_WITH_AES_128_CBC_SHA (0x002F)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xC013)
- TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003C)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xC027)

Vulnerability Details

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.

Actions to Take

CVSS 3.0 SCORE

CLASSIFICATION	SCORE
PCI DSS 3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
CWE	327
CAPEC	217
WASC	4
ISO27001	A.14.1.3

Weak cipher is an encryption-decryption algorithm that uses a key of insufficient length. This opens up the probability of the encryption scheme being broken. The impact of using weak ciphers is that the attackers may be able to decrypt SSL traffic between the server and the visitor.

Weak ciphers are encryption decryption algorithms that uses less than 128 bits key sizes as their value.

How to fix this issue

Configuring the web server to disallow using weak ciphers can be recognized as the remedy. In order to be stronger, the cipher key value should be larger.

Actions to Take

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

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

Lighttpd:

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

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 `regedit32` or type `regedit`, and then click OK.
- b. In Registry Editor, locate the following registry key:
`HKEY\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\MDS
```

Amazon.ca

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low

The screenshot shows the Netsparker interface with the following details:

- Vulnerability Details:** Insecure Transportation Security Protocol (TSL 1.0) is supported by the web server.
- Impact:** Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.
- Actions to Take:** We recommend to disable TSL 1.0 and replace it with TSL 1.2 or higher. See Remedy section for more details.
- Classification:** A large red stamp in the background reads "CONFIRMED".
- PCI DSS 3.2:** 6.5.4
- OWASP 2013:** A6
- OWASP 2017:** A3
- CWE:** 326
- CAPEC:** 217
- WASC:** 4
- HIPAA:** 164.306
- ISO27001:** A.14.1.3

TSL can also be used to encrypt emails, messaging and voice over IP. However, it was deemed that TSL 1.0 will be no longer be used for secure communication as it was vulnerable for many attacks including man-in-the-middle-attacks which risk the integrity and authentication of data sent between the website and the browser.

Man-in-the-middle-attacks are where the attacker is placed between a conversation of user and application in order to collect data that would normally would be encrypted.

How to fix this issue

Web server operators should disable TSL 1.0 and replace it with TSL 1.2 or above version.

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.**

Click on Start and then Run, type regedit32 or regedit, and then click OK.

In Registry Editor, locate the following registry key or create it if it does not exist:

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

Locate a key named Server or create if it doesn't exist.

Under the Server key, 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"
```

Insecure Transport Security Protocol Supported (TSL 1.1) Severity: Best Practice

The screenshot shows a software interface for vulnerability scanning. At the top, there are tabs for 'Sitemap - Previous Settings', 'Vulnerability' (which is selected), and 'HTTP Request / Response'. Below the tabs, the main title is 'Insecure Transportation Security Protocol Supported (TSL 1.1)'. There are two buttons: 'CONFIRMED' (highlighted in blue) and 'BEST PRACTICE'. A large red stamp with the word 'CONFIRMED' is overlaid on the right side of the screen. The URL listed is 'https://amazon.ca/'. Under 'Vulnerability Details', it states: 'Netsparker detected that a deprecated, insecure transportation security protocol (TSL 1.1) is supported by your web server. TSL 1.1 will be considered as deprecated by major web browsers (i.e. Chrome, Firefox, Safari, Edge, Internet Explorer) starting in 2020.' In the 'Impact' section, it says: 'Your website will be inaccessible due to web browser deprecation.' The 'Actions to Take' section recommends: 'We recommend to disable TSL 1.1 and replace it with TSL 1.2 or higher. See Remedy section for more details.' The 'Remedy' section is partially visible at the bottom. On the right, there is a 'CLASSIFICATION' table with various industry scores:

Classification	Score
PCI DSS 3.2	6.5.4
OWASP 2013	A6
OWASP 2017	A3
CWE	326
CAPEC	217
WASC	4
HIPAA	164.306
ISO27001	A14.1.3

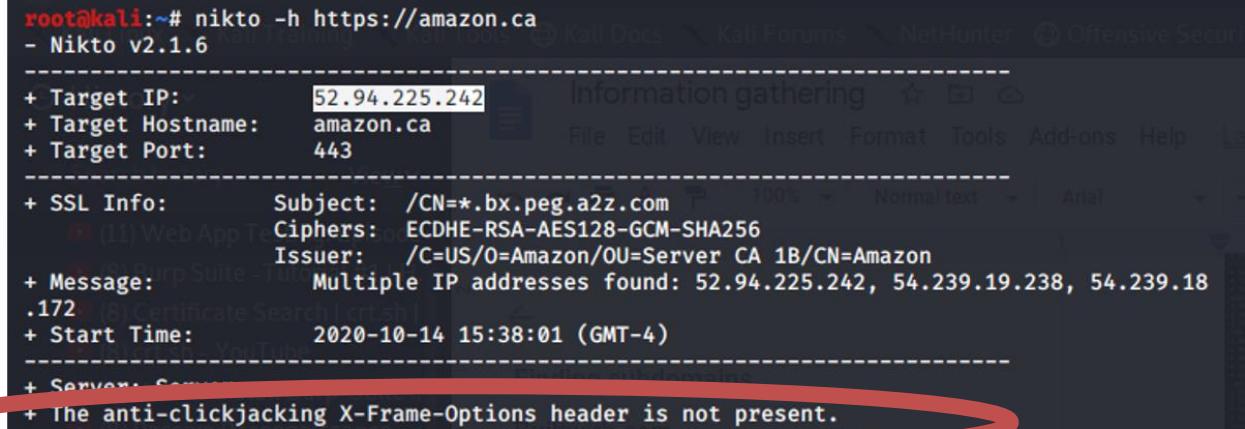
TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

How to fix this issue

Discontinue using TSL 1.1 and replace it with TSL 1.2 or higher protocol

The anti-clickjacking X-frame-option header is not present

Severity: Low



```
root@kali:~# nikto -h https://amazon.ca
- Nikto v2.1.6
-----
+ Target IP:      52.94.225.242
+ Target Hostname: amazon.ca
+ Target Port:    443
-----
+ SSL Info:       Subject: /CN=*.bx.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.94.225.242, 54.239.19.238, 54.239.18
.172
+ Start Time:    2020-10-14 15:38:01 (GMT-4)
-----
+ Server:        Amazon
+ The anti-clickjacking X-Frame-Options header is not present.
```

The X-frame-option header indicates whether or not the browser should allow to render a page within <frame>, <iframe>, <embed>, <object>. This header protects the website from clickjacking attacks.

Clickjacking Attacks

Clickjacking attacks is a malicious way of tricking the user into clicking on item that is disguised as item user believes to click into. The hidden page/item can cause users to visit malicious web pages or download malware or a legitimate page that user did not intended to visit.

As the user believes they are on the site they intended to visit, user can provide sensitive information to those sites, transfer money or purchase products etc. Likejacking and cursorjacking are few well known clickjacking attacks.

How to fix this issue

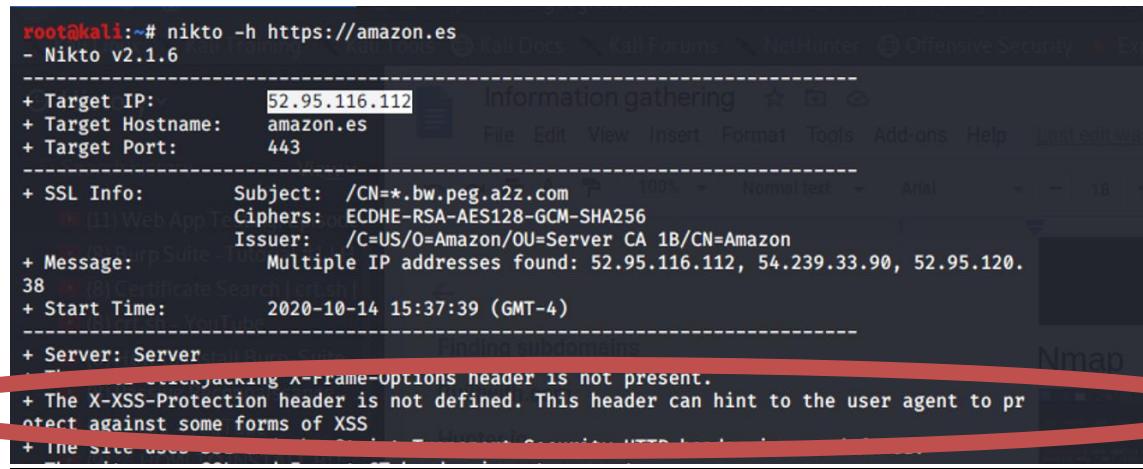
There are three values for the x-frame option header that instruct the browser to allow/disallow framing from other domains:

- DENY – does not allow any domain to display the page
- SAMEORIGIN – allows the page to display on another page but within the current domain
- ALLOW-FROM URI – allows the page to display in a frame but only in a specific frame

Amazon.es

Missing X-XXS protection header

Severity: Best Practice



The screenshot shows the terminal output of the Nikto web scanner against the URL `https://amazon.es`. The output includes details about the target IP (52.95.116.112), hostname (amazon.es), port (443), SSL info (Subject: /CN=*.bw.peg.a2z.com, Ciphers: ECDHE-RSA-AES128-GCM-SHA256, Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon), and a message indicating multiple IP addresses found. A red circle highlights the section of the output where it states: "The Clickjacking X-Frame-Options header is not present." and "The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS".

```
root@kali:~# nikto -h https://amazon.es
- Nikto v2.1.6
-----
+ Target IP:      52.95.116.112
+ Target Hostname:    amazon.es
+ Target Port:     443
-----
+ SSL Info:       Subject: /CN=*.bw.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer:  /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.95.116.112, 54.239.33.90, 52.95.120.
38
+ Start Time:    2020-10-14 15:37:39 (GMT-4)
-----
+ Server:        Server
+ OS:             Microsoft Windows Server 2012 R2 Standard
+ 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 Site uses SSL/TLS encryption (HTTP/2.0) and supports HTTP/1.1
-----
```

X-XXS protection header is set as a built-in web browser cross-site scripting filter meaning the header stops from loading when detecting reflected cross-site scripting attacks. When this header is missing from a website it is vulnerable for XSS attacks.

```
X-XSS-Protection: 0
X-XSS-Protection: 1
X-XSS-Protection: 1; mode=block
X-XSS-Protection: 1; report=<reporting-uri>
```

How to fix this issue

Add X-XXS protection header with value “1; mode=block”

Solution:

Configure this header for the web application ensuring correct values are set.

X-XSS-Protection: 1; mode=block

PHP

```
header("X-XSS-Protection: 1; mode=block");
```

Apache (.htaccess)

```
<IfModule mod_headers.c>
```

```
Header set X-XSS-Protection "1; mode=block"
```

```
</IfModule>
```

Nginx

```
add_header "X-XSS-Protection" "1; mode=block";
```

Strict transport security HTTP header not defined

Severity: Medium

```
root@kali:~# nikto -h https://amazon.es
- Nikto v2.1.6

+ Target IP:      52.95.116.112
+ Target Hostname: amazon.es
+ Target Port:    443
-----
+ SSL Info:       Subject: /CN=*.bw.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.95.116.112, 54.239.33.90, 52.95.120.
38
+ Start Time:    2020-10-14 15:37:39 (GMT-4)
-----
+ Server: Server
+ 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 pr
otect against some forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined. The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The Site's 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
```

HSTS, otherwise known as HTTP Strict Transport Security is a web security policy where the server declares that complying user agents such as web browser are to interact with it only using secure HTTPS connections. The HSTS policy communicated to the user is via the strict transport security header. This policy specifies a time period where the user can access the server only in a secure fashion.

This policy automatically turns any insecure HTTP links to secure links and if the security of a connection could not be ensured, an error message is displayed and user is not allowed to access the web application.

How to fix this issue

Remediation

Configure your webserver to redirect HTTP requests to HTTPS.

i.e. for Apache, you should have modification in the httpd.conf. For more configurations, please refer to External References section.

```
# load module
LoadModule headers_module modules/mod_headers.so

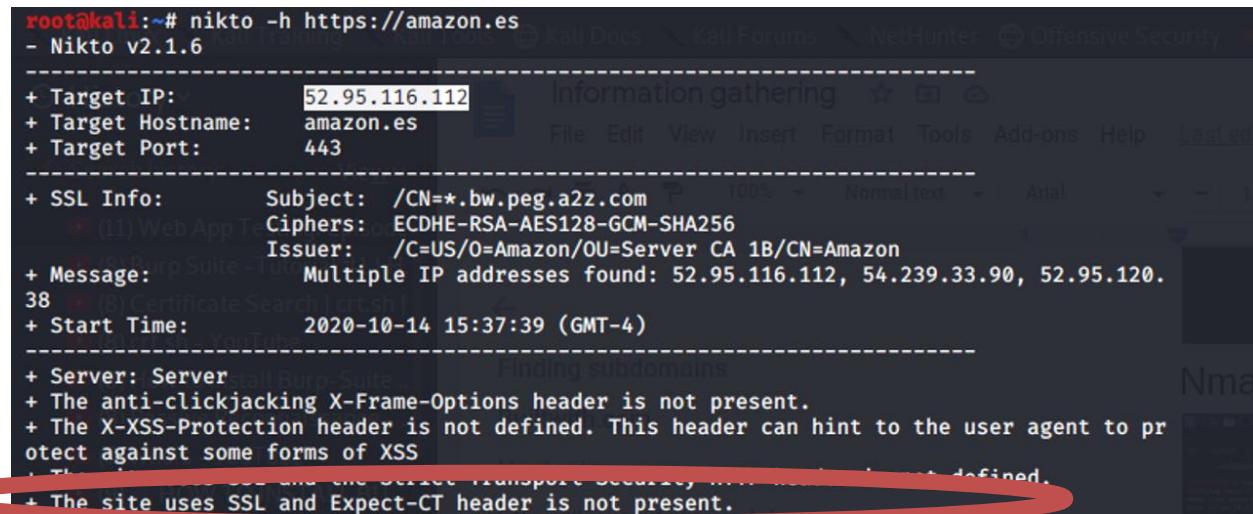
# redirect all HTTP to HTTPS (optional)
<VirtualHost *:80>
    ServerAlias *
    RewriteEngine On
    RewriteRule ^(.*)$ https:// %{HTTP_HOST}$1 [redirect=301]
</VirtualHost>

# HTTPS-Host-Configuration
<VirtualHost *:443>
    # Use HTTP Strict Transport Security to force client to use secure connections only
    Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

    # Further Configuration goes here
    [...]
</VirtualHost>
```

Expect – CT header is not defined

Severity: Best Practice



The screenshot shows the terminal output of the Nikto tool scanning the website <https://amazon.es>. The output includes details about the target IP (52.95.116.112), hostname (amazon.es), port (443), SSL info (Subject: /CN=*.bw.peg.a2z.com, Ciphers: ECDHE-RSA-AES128-GCM-SHA256, Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon), message (Multiple IP addresses found: 52.95.116.112, 54.239.33.90, 52.95.120.), start time (2020-10-14 15:37:39 (GMT-4)), and server information. A red oval highlights the final line of the output: '+ The site uses SSL and Expect-CT header is not present.'

```
root@kali:~# nikto -h https://amazon.es
- Nikto v2.1.6
-----
+ Target IP:      52.95.116.112
+ Target Hostname: amazon.es
+ Target Port:    443
-----
+ SSL Info:       Subject: /CN=*.bw.peg.a2z.com
                  Ciphers: ECDHE-RSA-AES128-GCM-SHA256
                  Issuer: /C=US/O=Amazon/OU=Server CA 1B/CN=Amazon
+ Message:        Multiple IP addresses found: 52.95.116.112, 54.239.33.90, 52.95.120.
38
+ Start Time:    2020-10-14 15:37:39 (GMT-4)
-----
+ Server: Server
+ 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 Content-Security-Policy and the Strict-Transport-Security headers are not defined.
+ The site uses SSL and Expect-CT header is not present.
```

Certificate transparency, shortly known as CT is a technology that makes it very difficult for a certificate authority to issue an SSL certificate for a domain without the certificate being visible to the owner of the domain.

In 2018 Google announced that if it finds a website with a certificate that is not in the CT log it will consider the certificate invalid and reject the connection.

How to fix this issue

Remediation

Configure your web server to respond with Expect-CT header.

```
Expect-CT: enforce, max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"
```

Note: We strongly suggest you to use Expect-CT header in **report-only mode** first. If everything goes well and your certificate is ready, go with the Expect-CT enforce mode. To use **report-only mode** first, omit **enforce** flag and see the browser's behavior with your deployed certificate.

```
Expect-CT: max-age=7776000, report-uri="https://ABSOLUTE_REPORT_URL"
```

Conclusion

Overall, the web application under the audit was well designed and has implemented many security measures to protect itself from cyber-attacks. During the testing, there were only medium to low risk vulnerabilities and best practices were observed and there were no high-risk vulnerabilities, which indeed a good sign for a website to appear secured and well protected.

However, it was also seen that the website vulnerable for few of the most common cyber attacks at present day such as Cross Site Scripting and SQL injections. Amazon is one of the leading e-commerce companies in the world and as a result, seeing that when such a well provided company cannot successfully protect itself from the entire field of cyber attacks shows how complicated the world of web and cyber security really is in the present.