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;

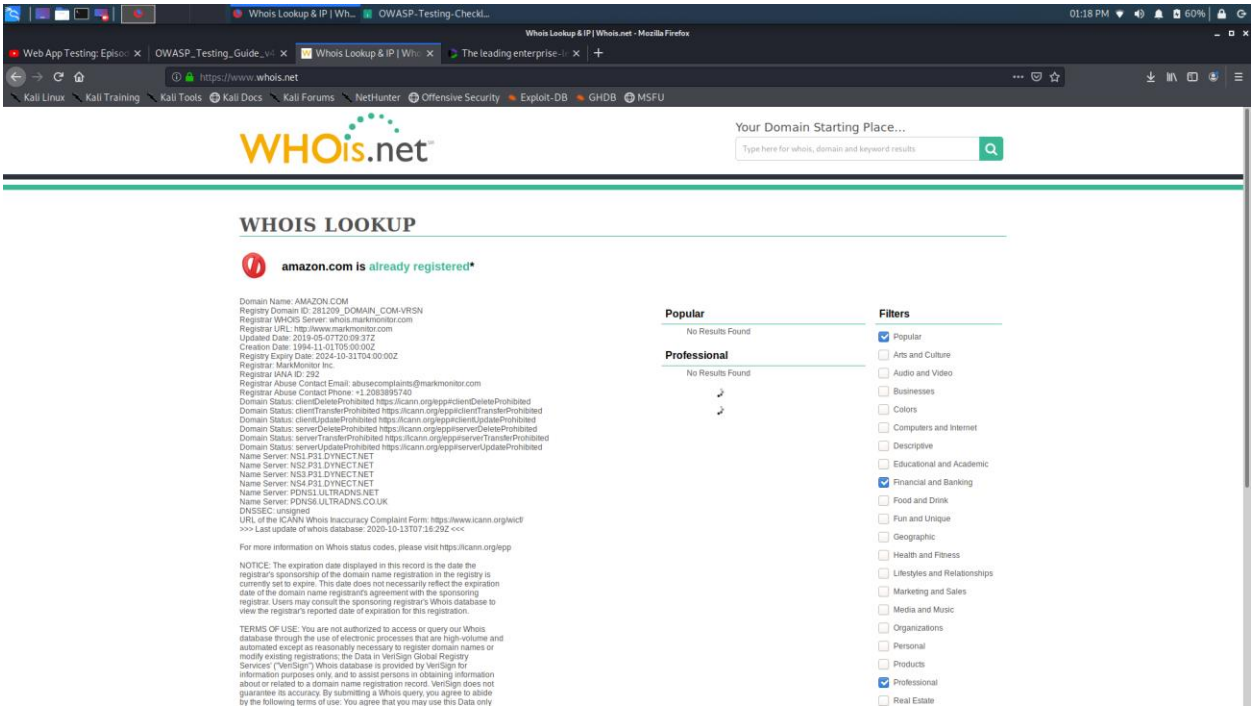
1. Passive – gathering information without any contact with the target web application
2. Active – gathering information while being in contact with the target web application

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.



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 utilize to find subdomains are:

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

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

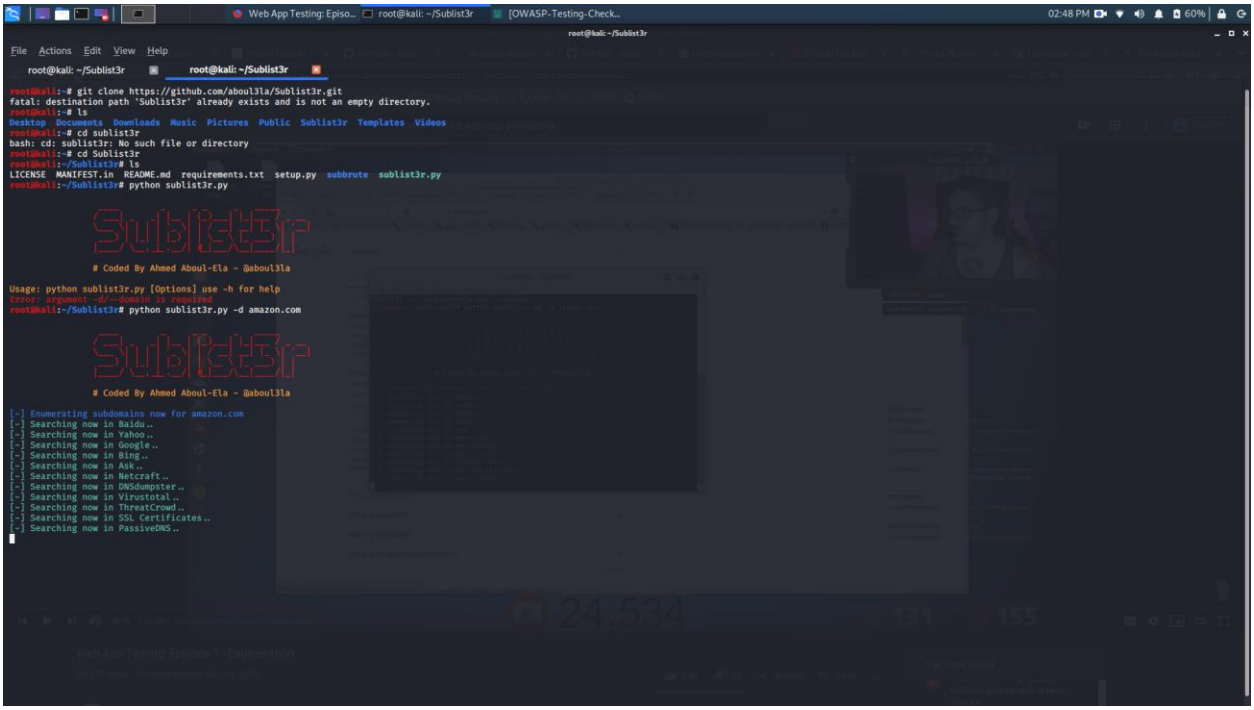
Sublist3r

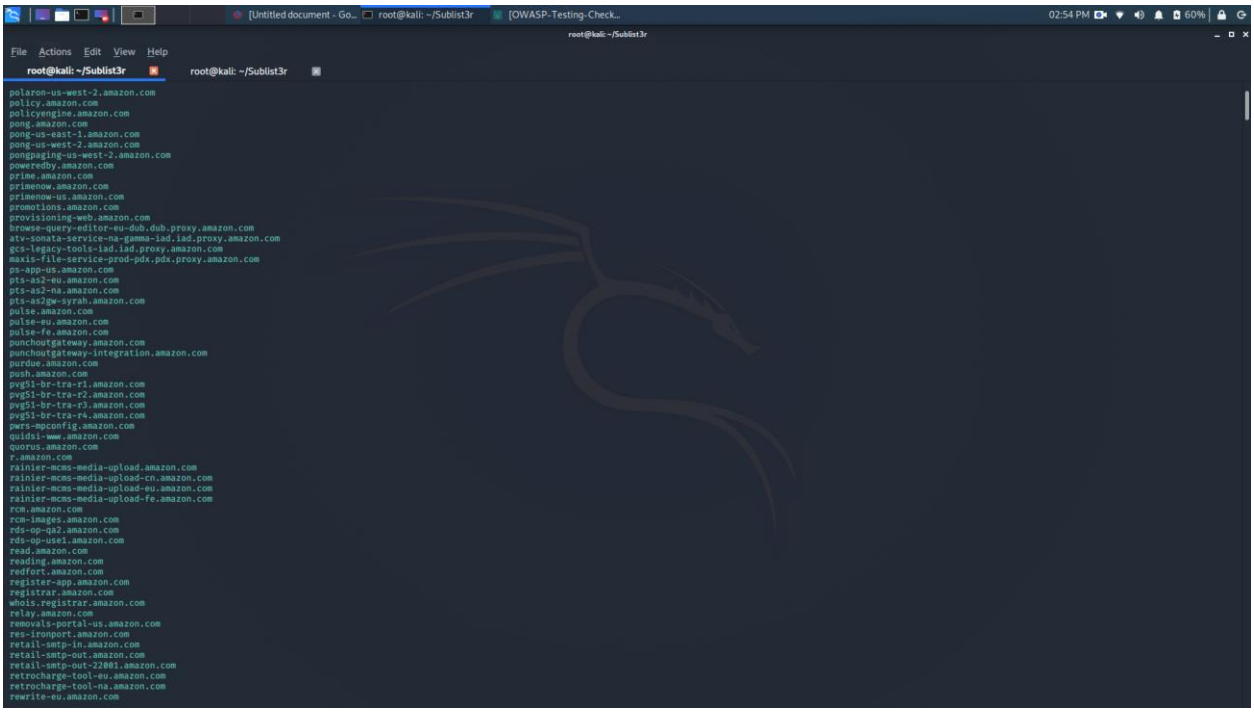
Sublist3r is a python-based tool that identify subdomains of a website. By using this tool, attackers and penetration testers collect subdomains of the targeted domain in order to run a penetration test or an attack.

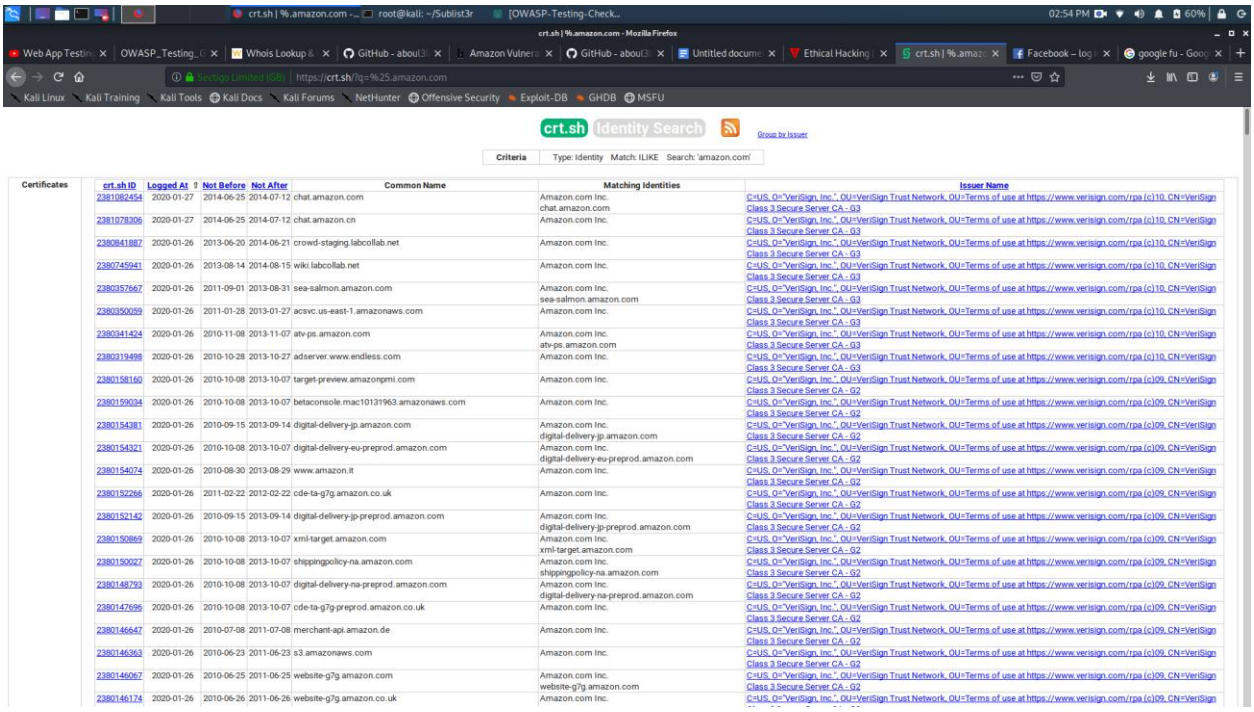
Sublist3r search subdomains using search engines such as: Google, Bing, Yahoo, Baidu etc. and other services such as Netcraft, ThreatCrowd, DNSdumpter, ReverseDNS etc.

In order to install sublst3r 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"







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.

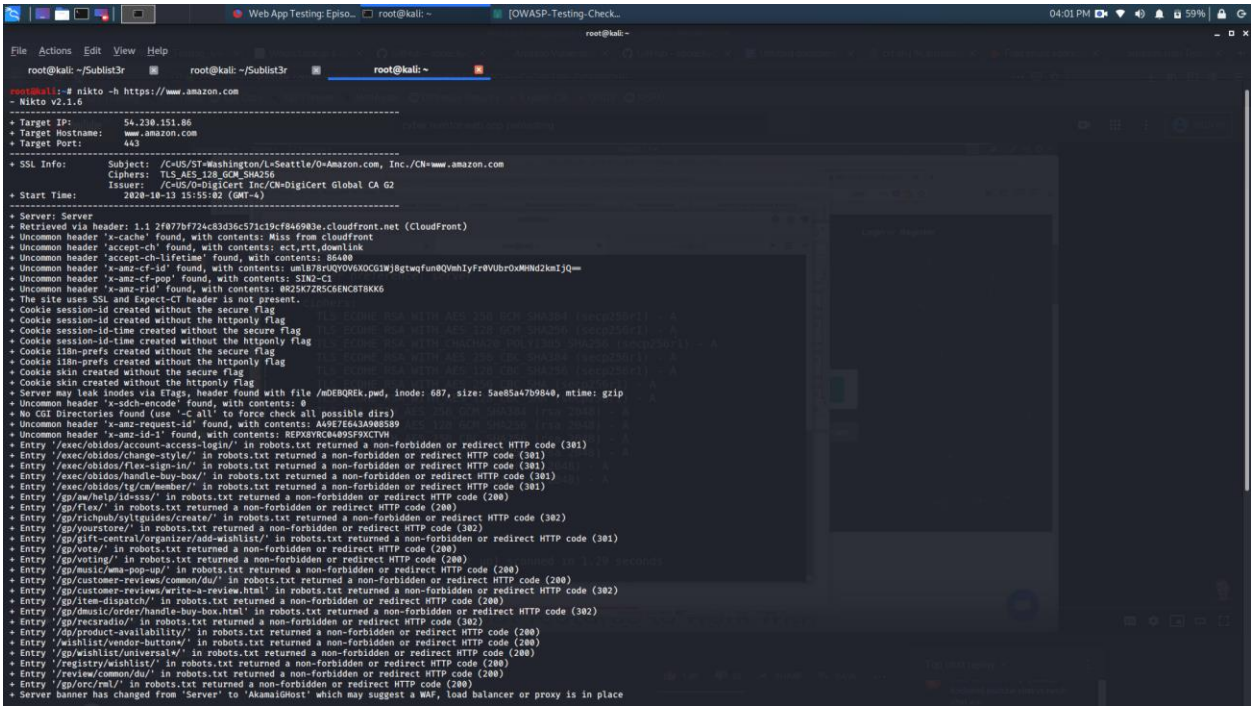
Vulnerability scanner is a tool that identifies and creates an inventory of all systems. Then it cross examines each of these systems against well-known vulnerabilities and scan whether such vulnerabilities are present in the system.

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.

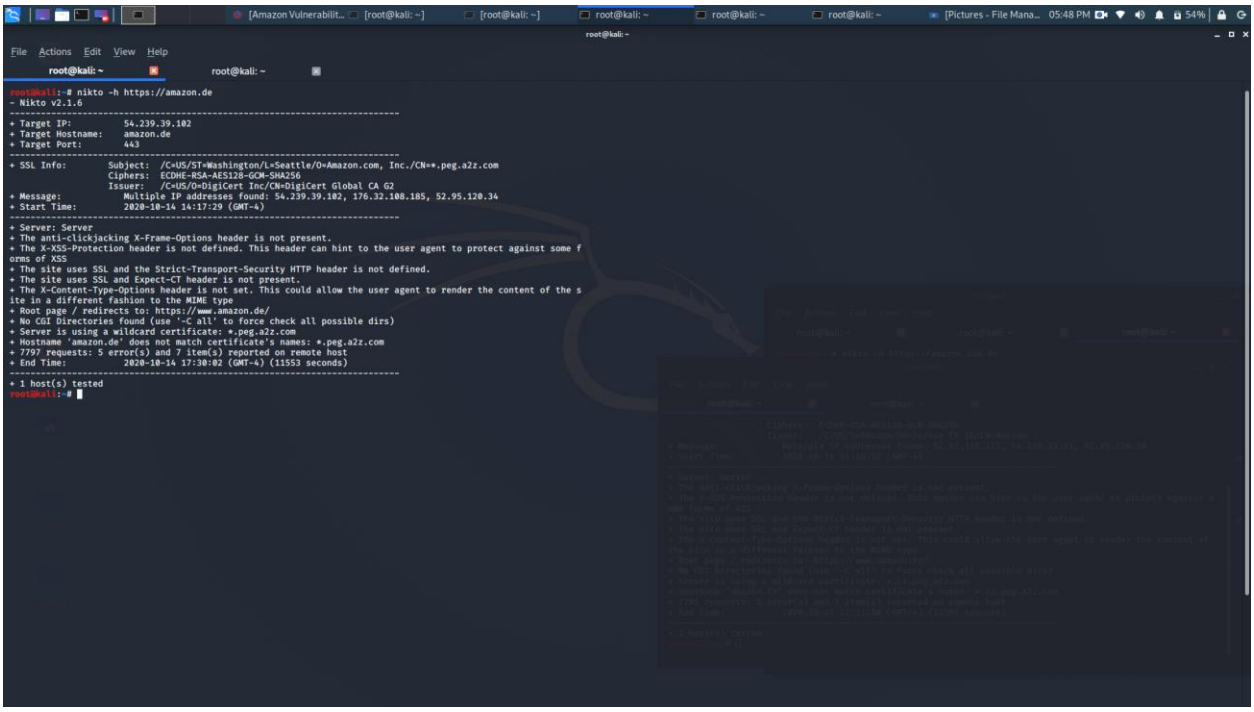


"Nikto -host" is used to specify the target host where the target is the website to scan. Here as example, the domain amazon is set as the target. Therefore "nikto -h http://www.amazon.com" is typed to run a basic scan against the target.

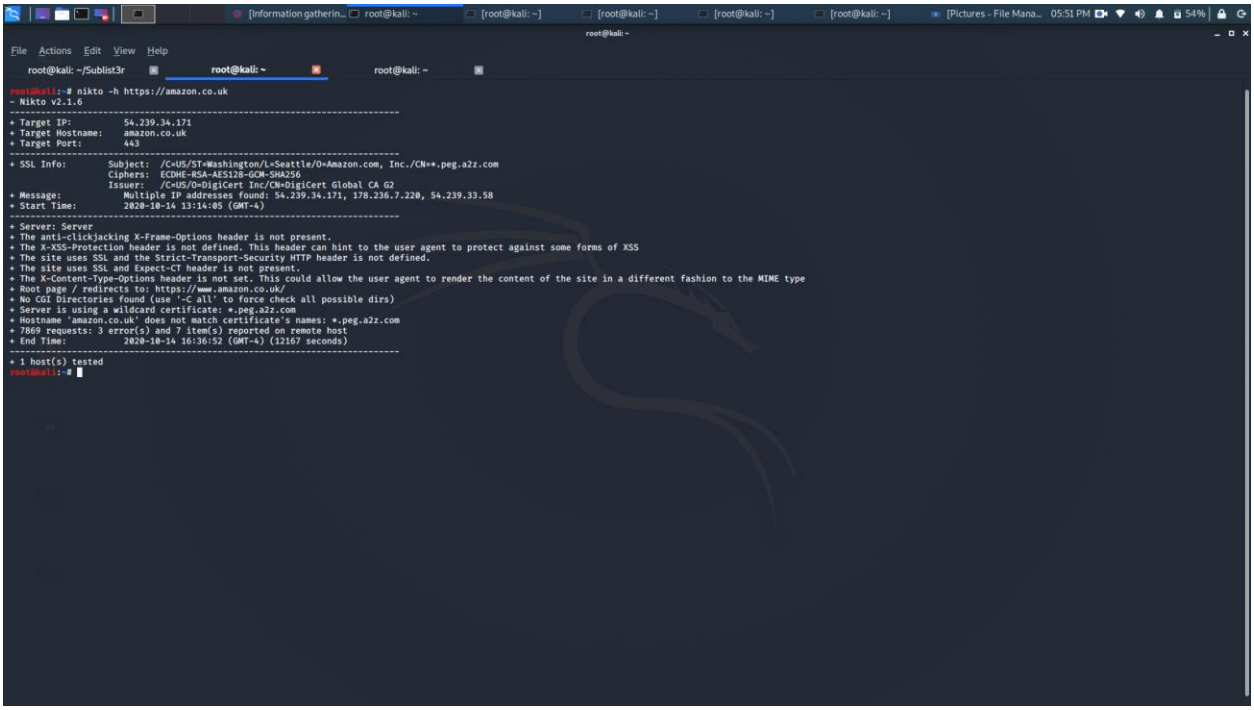
"-p" command could also be used here to specify which ports to examine. Nikto is able to scan single, multiple or range of ports anywhere between 1 - 1000. If port is not specified, it will only scan port 80 as default.

"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.

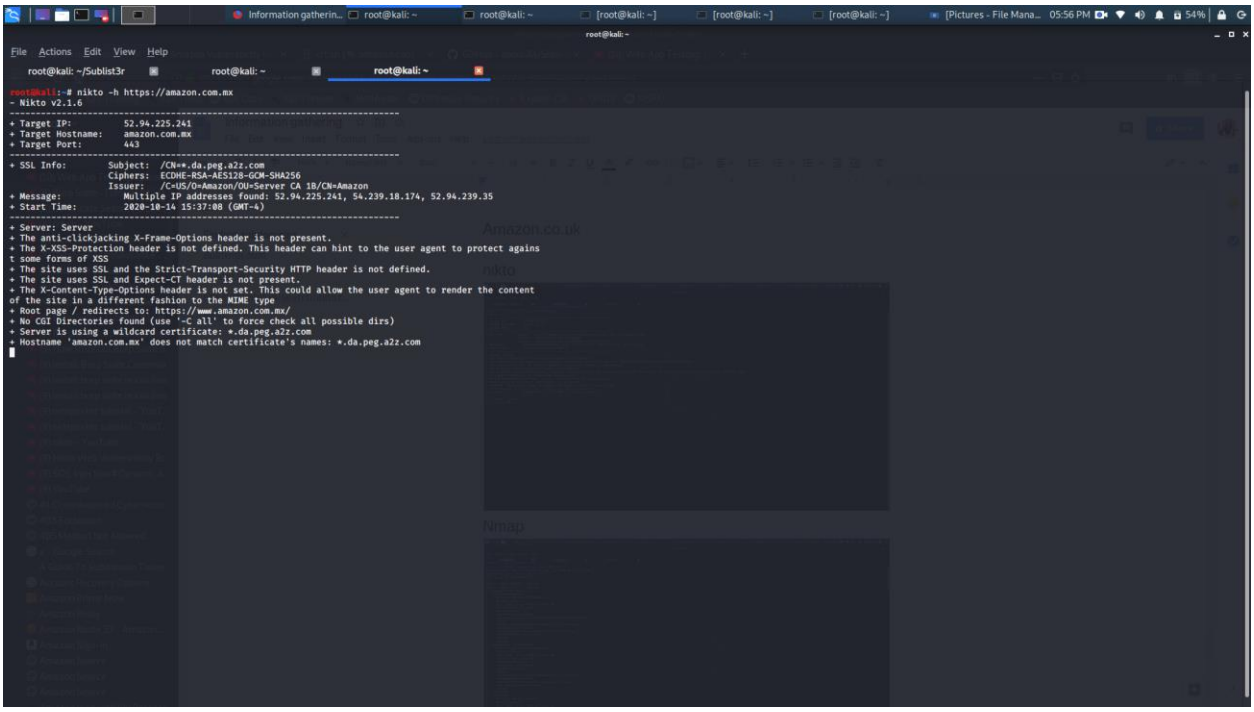
Amazon.de



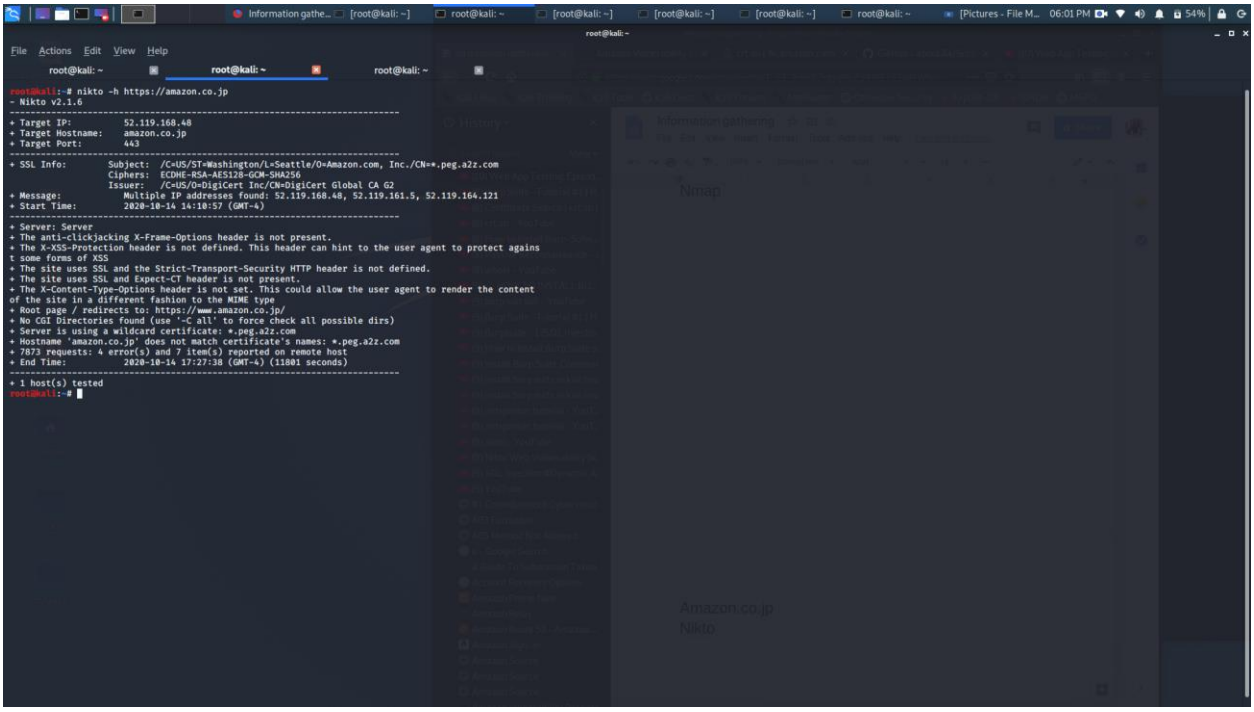
Amazon.co.uk



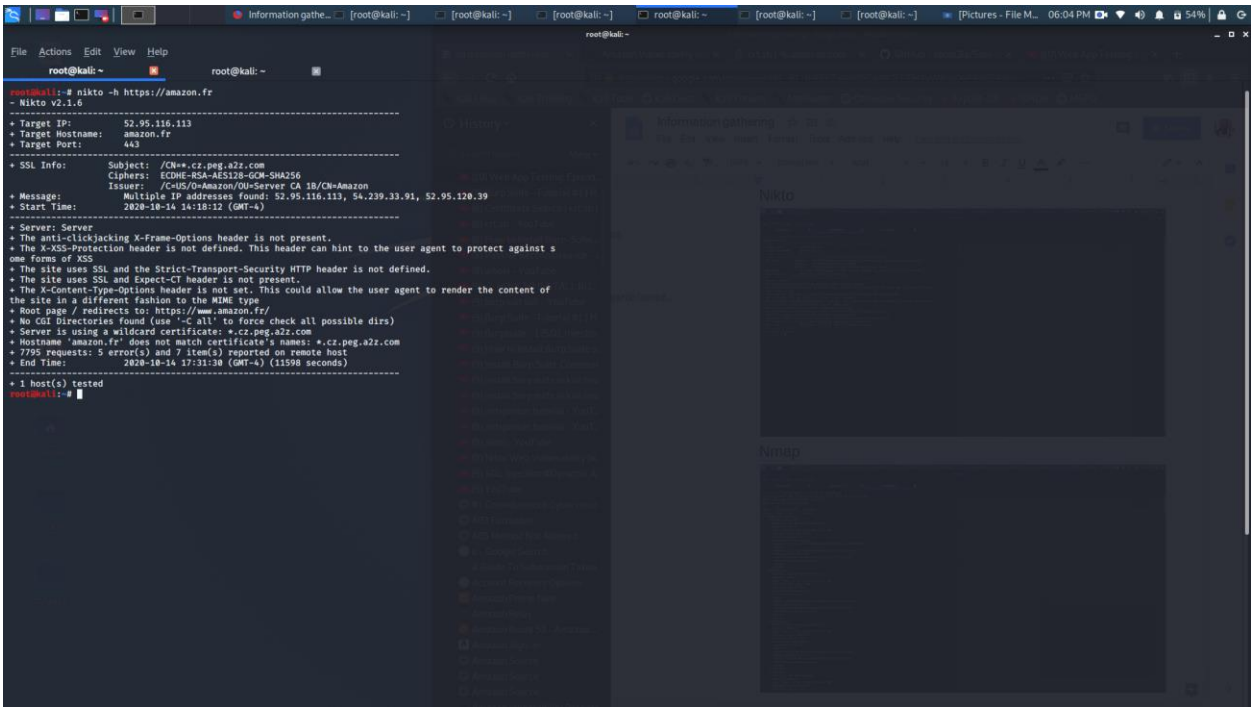
Amazon.com.mx



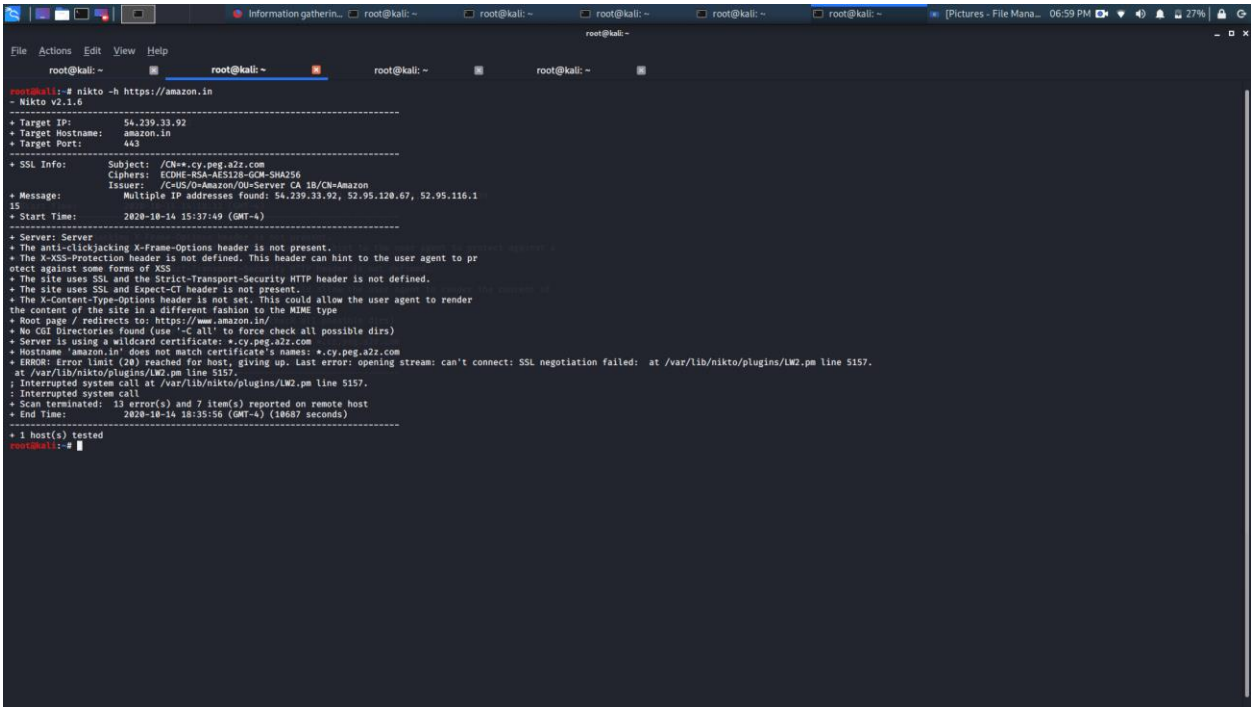
Amazon.co.jp



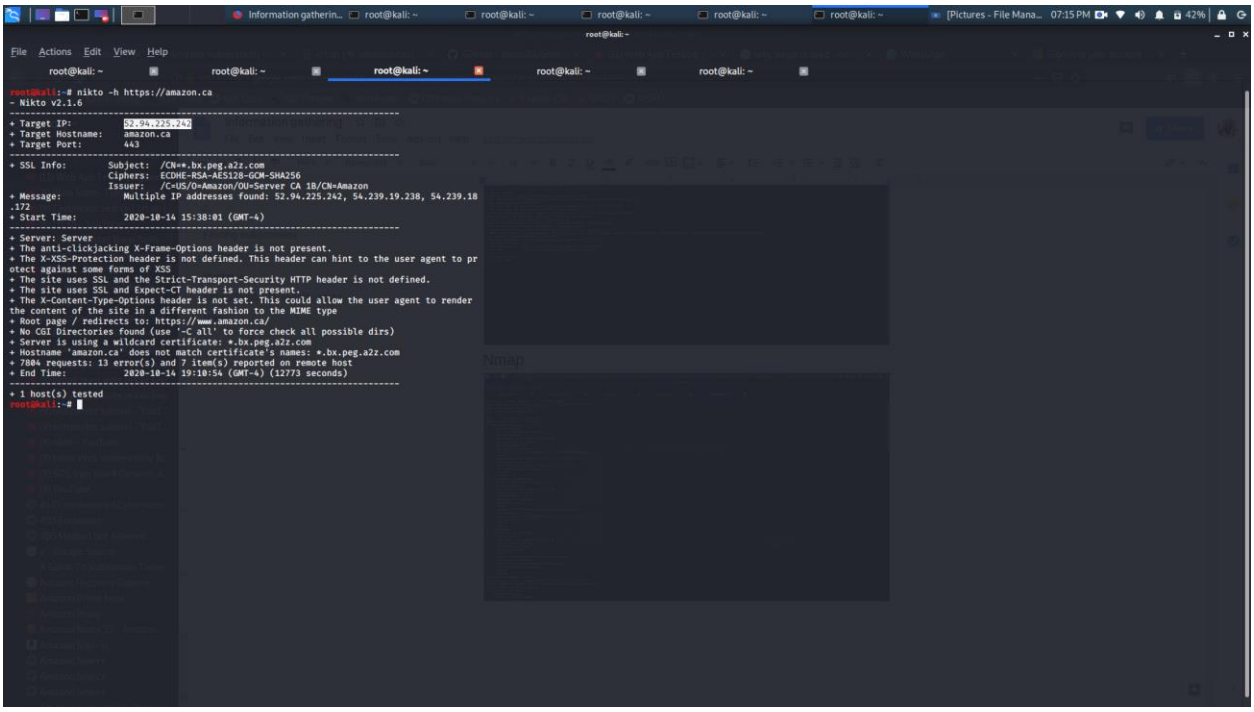
Amazon.fr



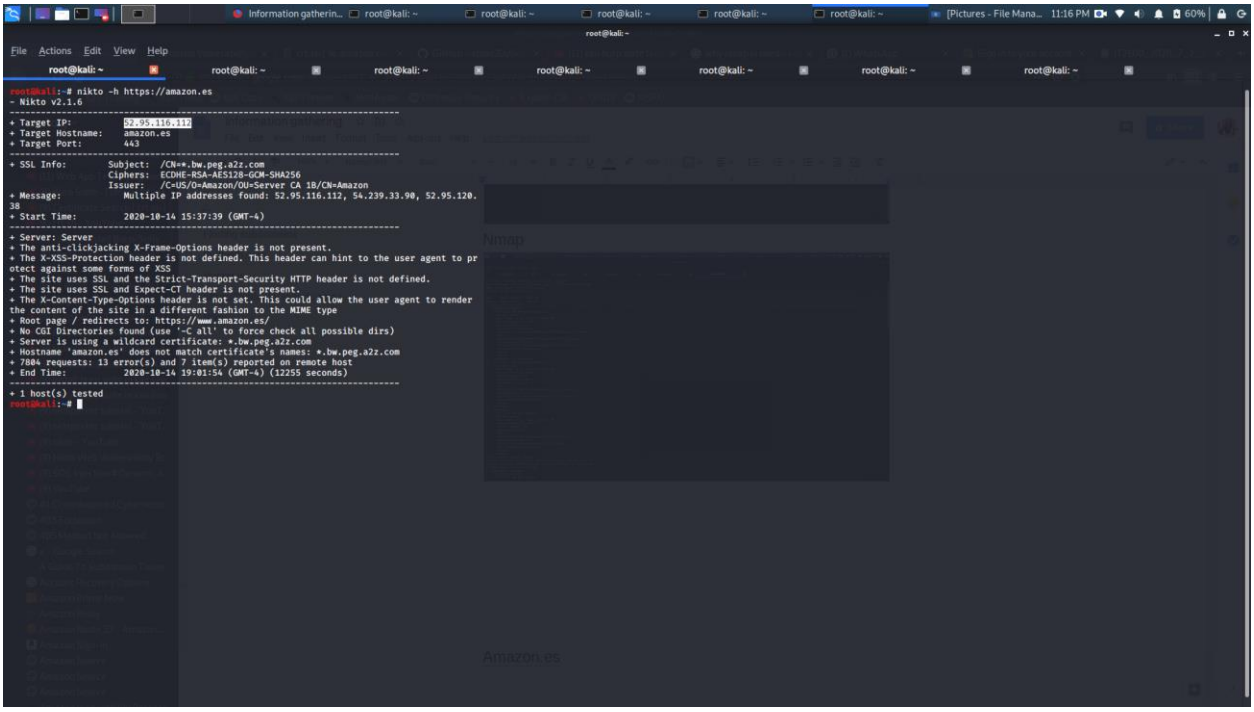
Amazon.in



Amazon.ca



Amazon.es



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

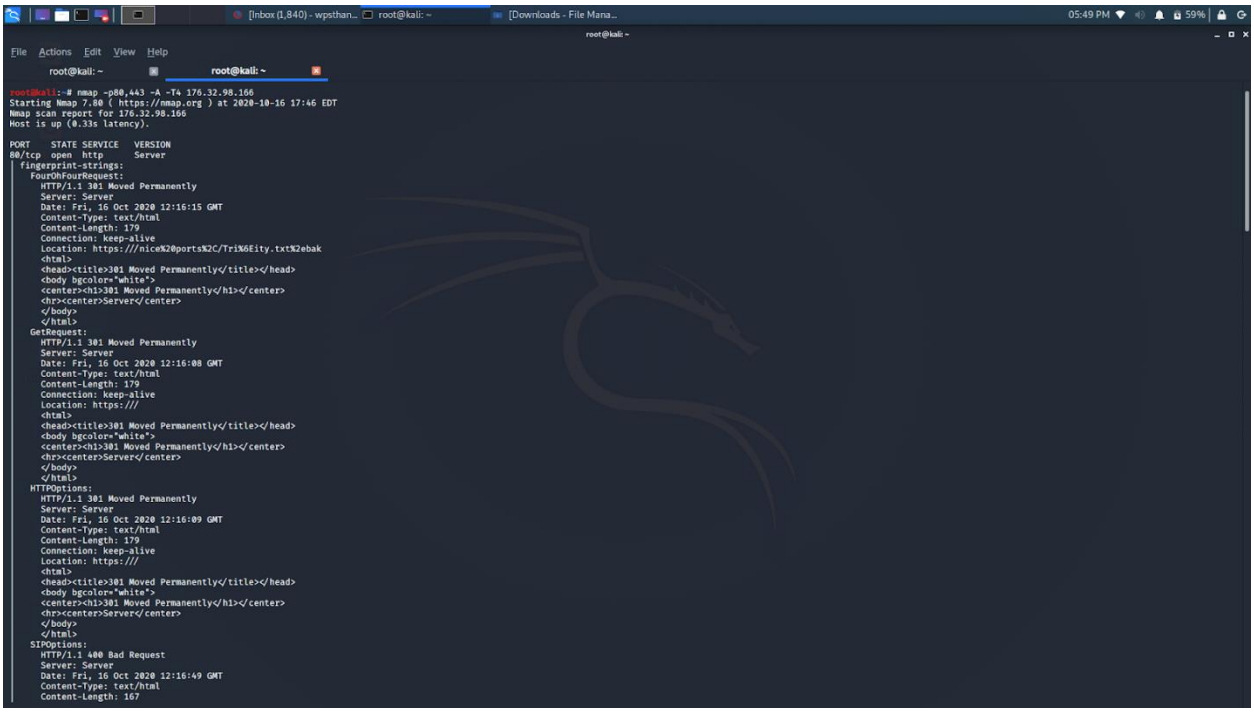
Nmap, also known as Network Mapper, is a comprehensive active stack fingerprinting tool. It identifies what devices are running on the targeted system, discover host that are available, find open ports, detect security risks and more.

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.

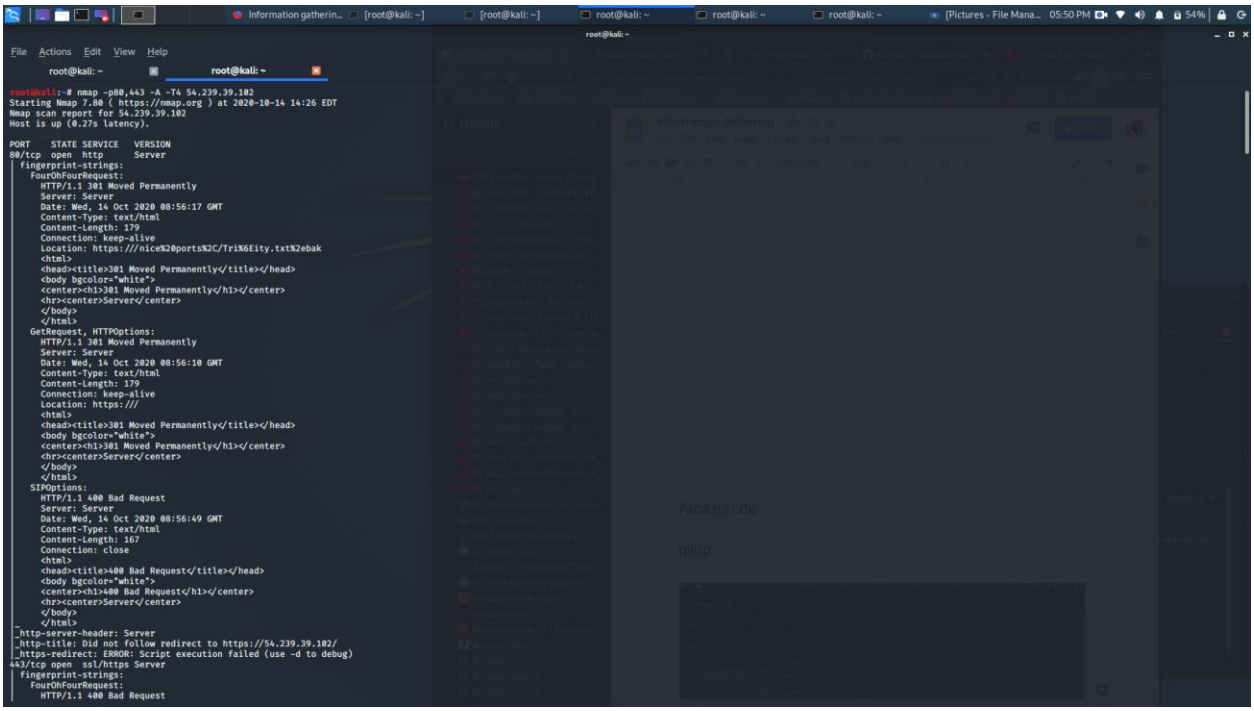
“nmap http://www.amazon.com (domain name/ IP address)”is one of the basic commands of nmap that scans a single IP address whereas “nmap http://www.amazon.com -p 20-3000”scans a range of ports in nmap.

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 that usual nmap commands. It also scans multiple ports (port 80 and 443) at once.

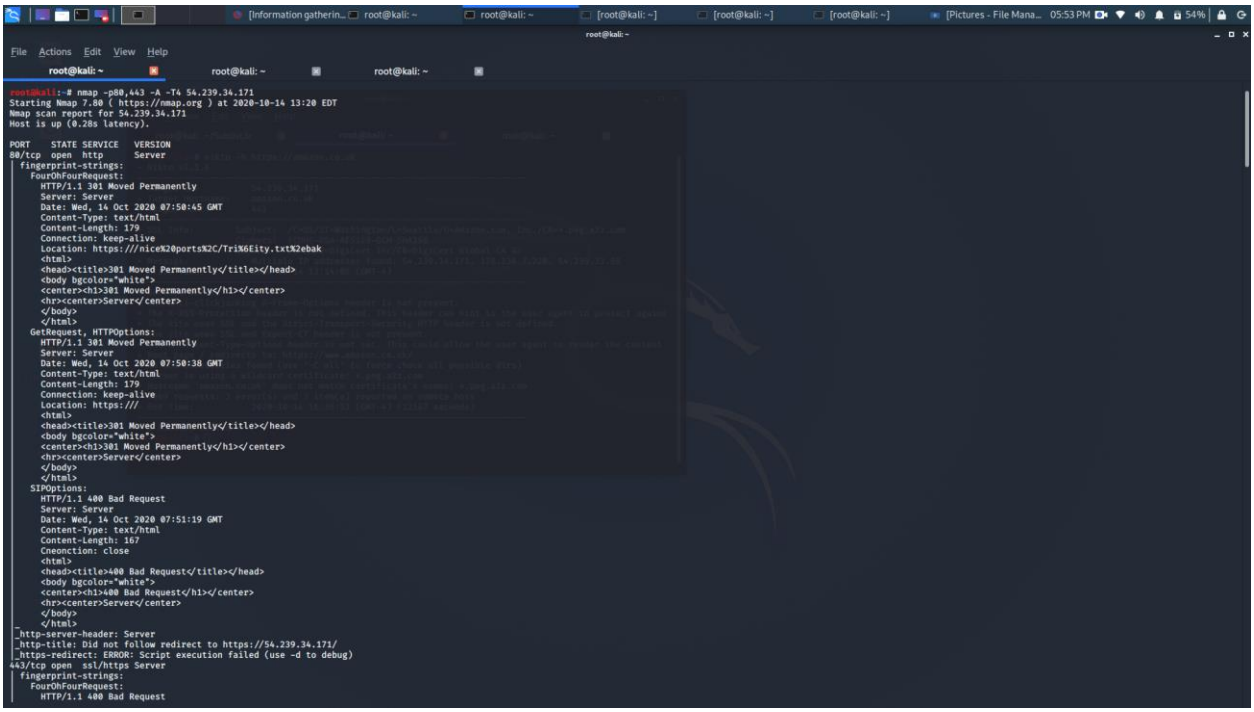
Amazon.com



Amazon.de

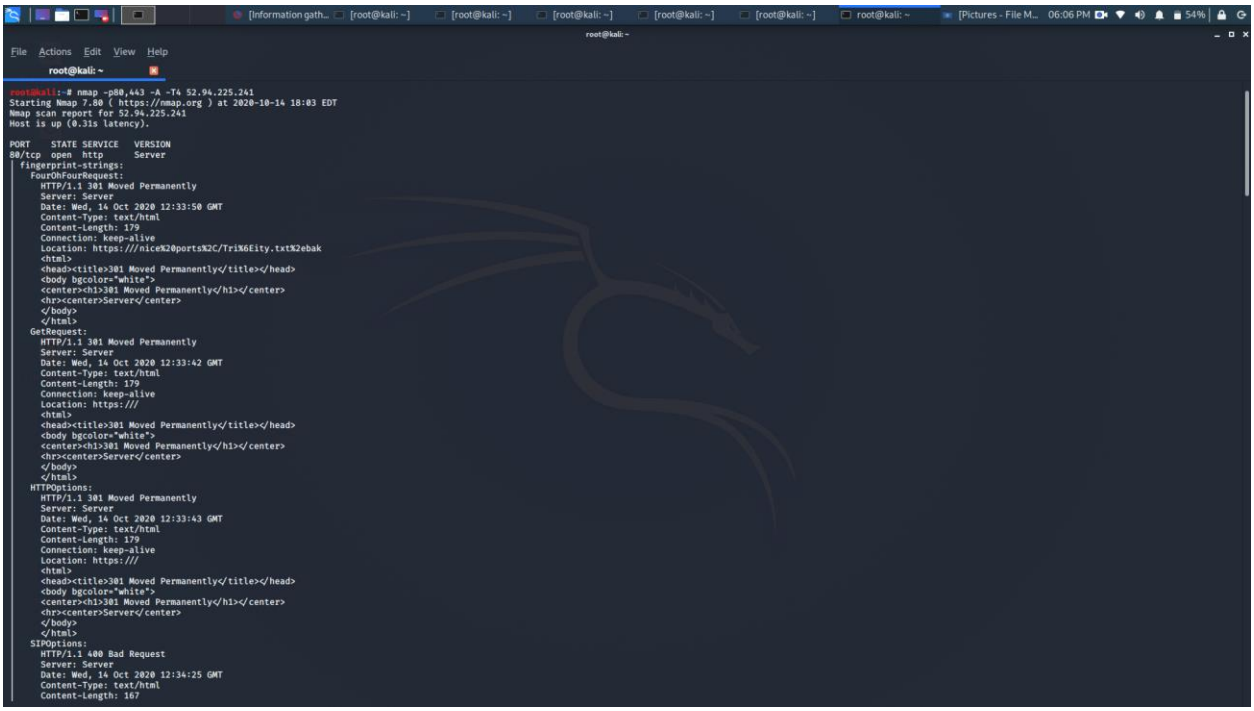


Amazon.co.uk



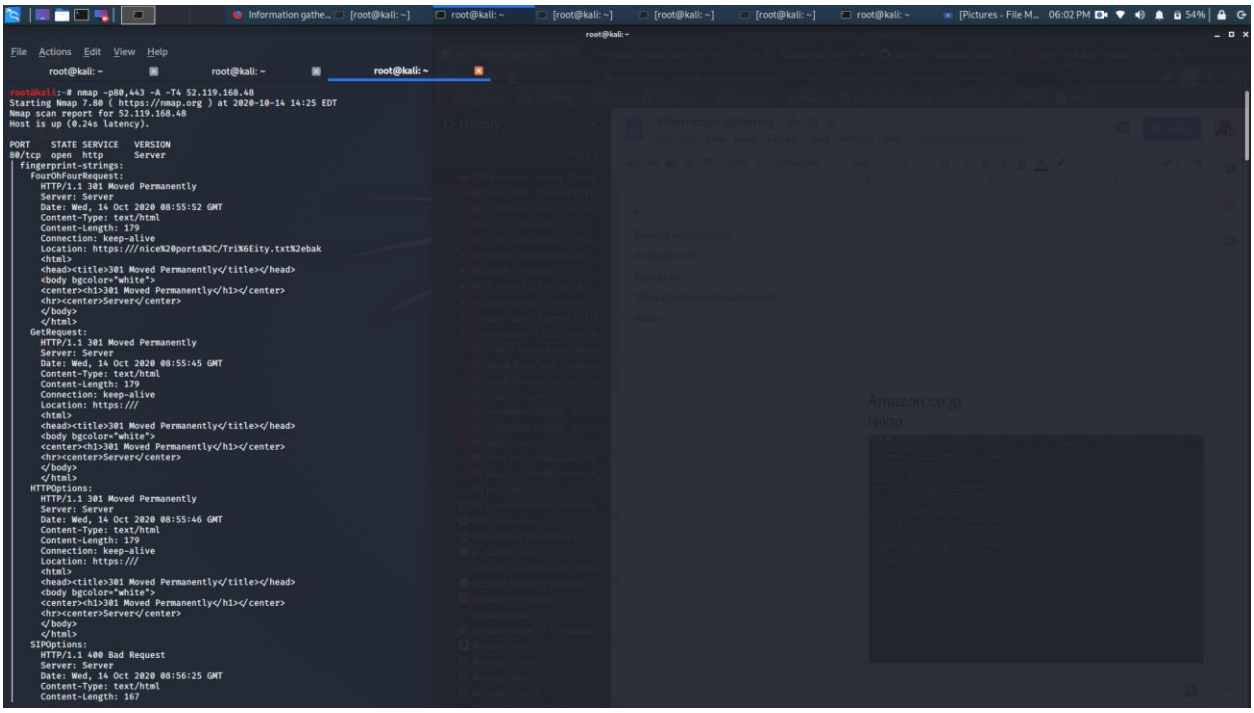
```
root@kali:~# nmap -p80,443 -A -T4 54.239.34.171
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 13:20 EDT
Nmap scan report for 54.239.34.171
Host is up (0.28s latency).

PORT    STATE SERVICE   VERSION
80/tcp  open  http      Server
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 301 Moved Permanently
|     Server: Server
|     Date: Wed, 14 Oct 2020 07:50:45 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>301 Moved Permanently</h1></center>
|     <hr><center>Server</center>
|     </body>
|     </html>
|   GetRequest, HTTPOptions:
|     HTTP/1.1 301 Moved Permanently
|     Server: Server
|     Date: Wed, 14 Oct 2020 07:50:38 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 07:51:19 GMT
|     Content-Type: text/html
|     Content-Length: 167
|     Cneonction: close
|     <html>
|     <head><title>400 Bad Request</title></head>
|     <body bgcolor="white">
|     <center><h1>400 Bad Request</h1></center>
|     <hr><center>Server</center>
|     </body>
|_    </html>
|_http-server-header: Server
|_http-title: Did not follow redirect to https://54.239.34.171/
|_https-redirect: ERROR: Script execution failed (use -d to debug)
443/tcp open  ssl/https Server
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 400 Bad Request
```

Amazon.com.mx

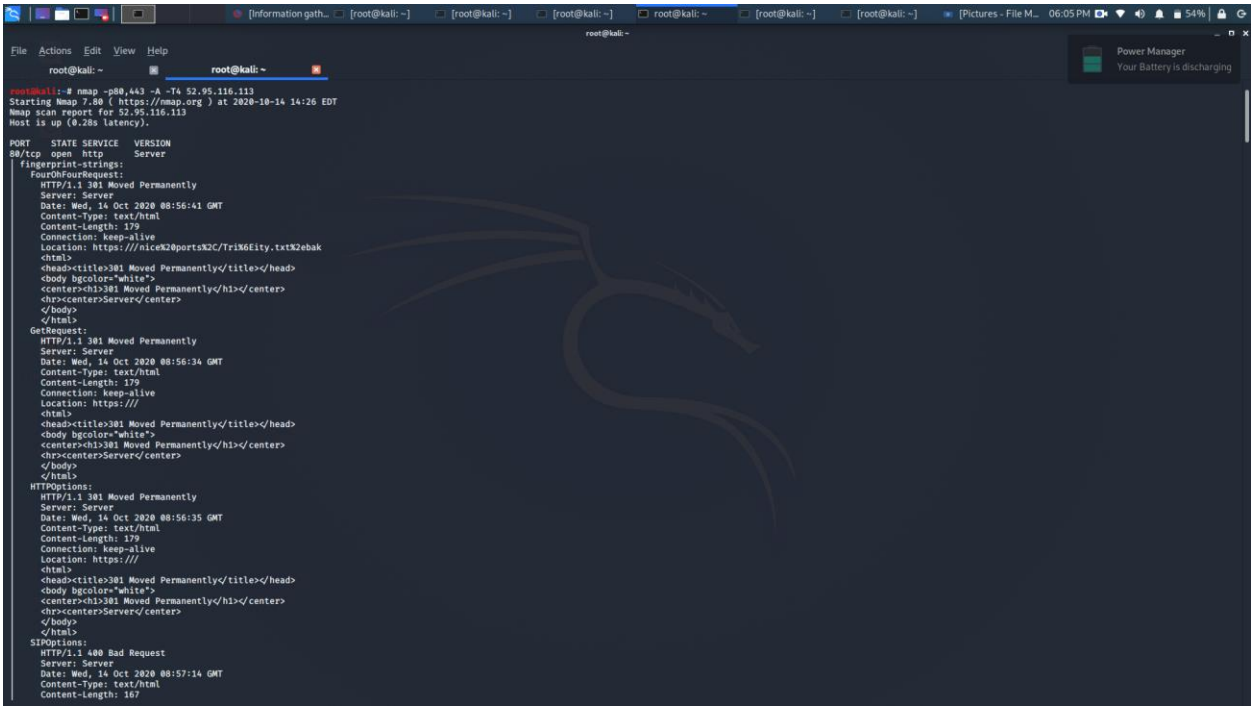


```
root@kali:~# nmap -p80,443 -A -T4 52.94.225.241
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-14 18:03 EDT
Nmap scan report for 52.94.225.241
Host is up (0.31s latency).

PORT    STATE SERVICE   VERSION
80/tcp  open  http      Server
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 301 Moved Permanently
|     Server: Server
|     Date: Wed, 14 Oct 2020 12:33:50 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>301 Moved Permanently</h1></center>
|     <hr><center>Server</center>
|     </body>
|     </html>
|   GetRequest:
|     HTTP/1.1 301 Moved Permanently
|     Server: Server
|     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
```

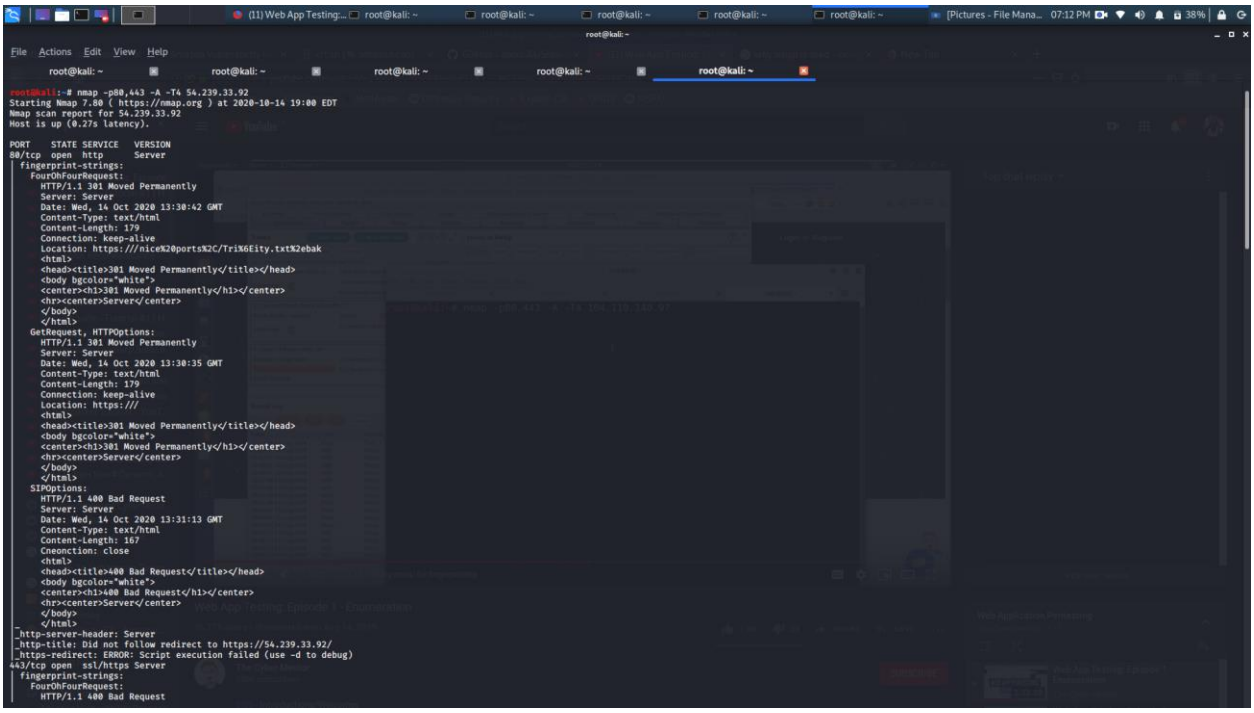
Amazon.co.jp



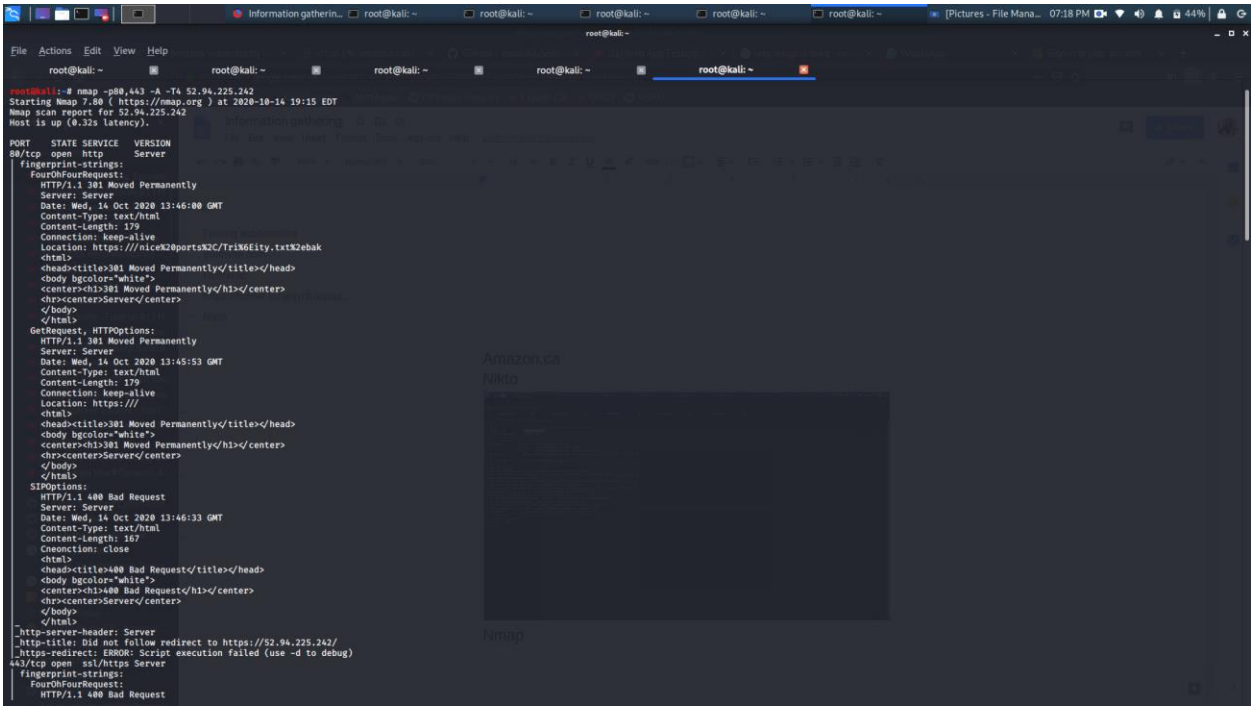
Amazon.fr



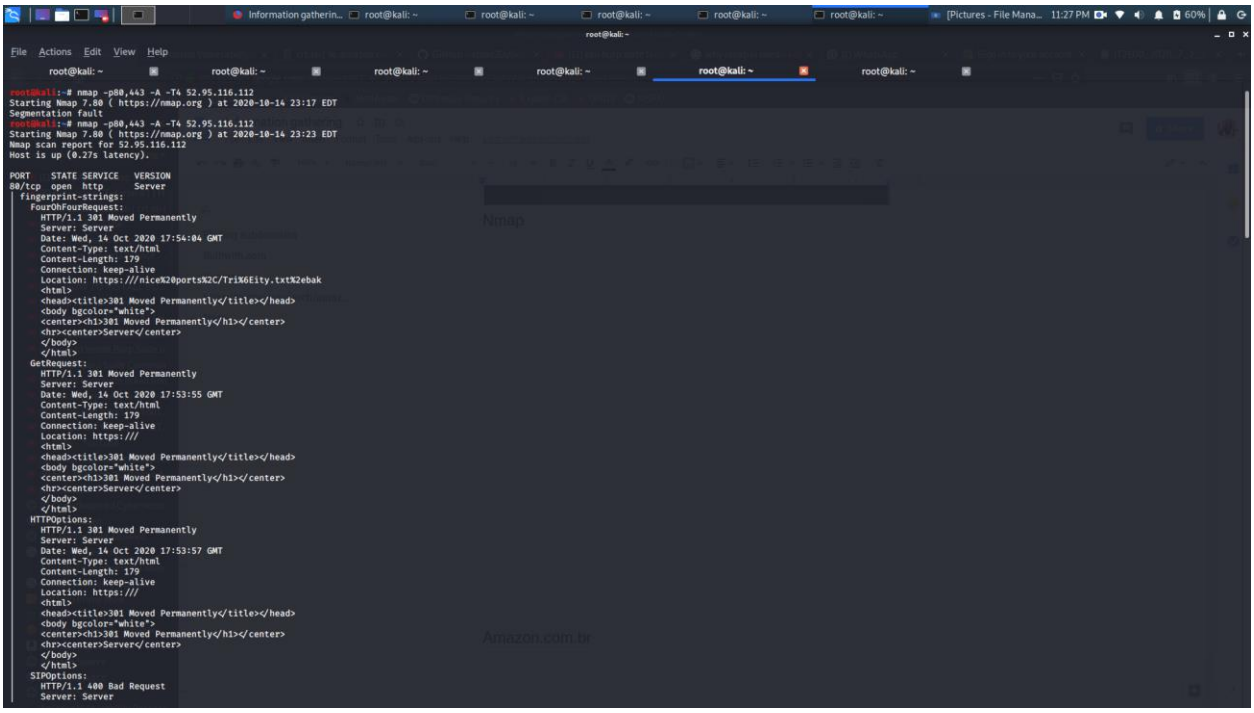
Amazon.in



Amazon.ca



Amazon.es


root@kali: ~
File Actions Edit View Help
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
| fingerprint-strings:
| FourOhFourRequest:
| 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:///nice%20ports%2C/Tri%6Eity.txt%2ebak
| <html>
| <head><title>301 Moved Permanently</title></head>
| <body bgcolor="white">
| <h1>301 Moved Permanently</h1>
| <hr> Server
| </body>
| </html>
| GetRequest:
| 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">
| <h1>301 Moved Permanently</h1>
| <hr> Server
| </body>
| </html>
| HTTPOptions:
| 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">
| <h1>301 Moved Permanently</h1>
| <hr> Server
| </body>
| </html>
| SIPOptions:
| HTTP/1.1 400 Bad Request
| Server: Server

Vulnerabilities

Vulnerability is a weakness within a system that can be exploit during a cyber attack to preform 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 – Privilege information is presented to only those who can has clearance
- Integrity – Protects data from being unauthorizedly 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.

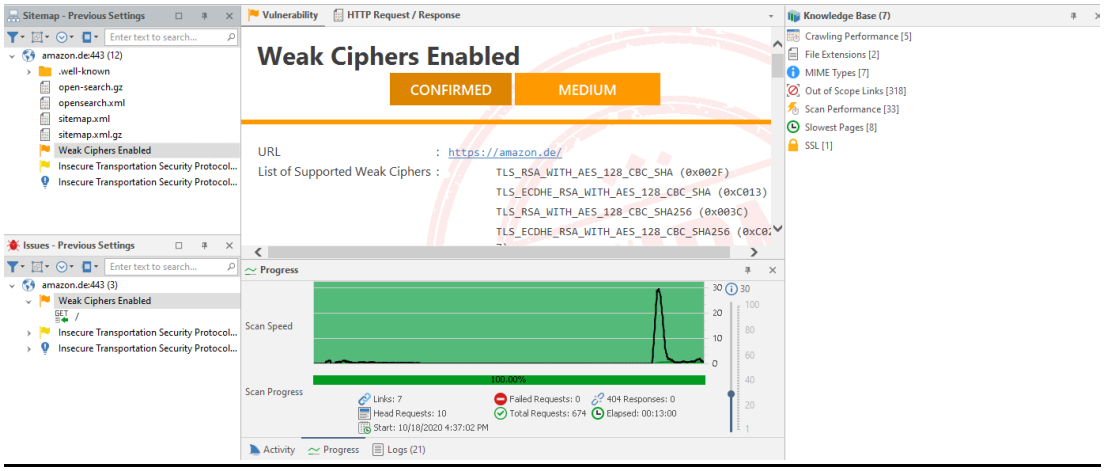
Such attacks are lunched using a vulnerability that is hidden within the system itself. Therefore, such vulnerabilities must be found and patch in a routine checkup.

Here, using Netsparker and nikto all chosen subdomains where 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 recognize 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"
```

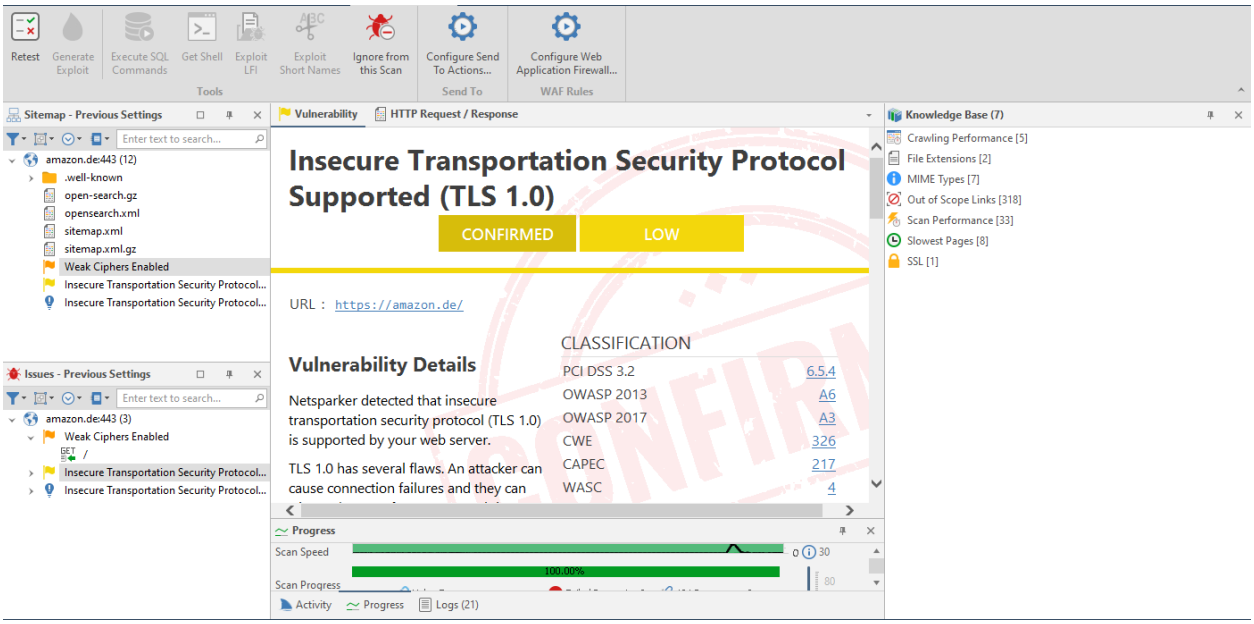
For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
b. In Registry Editor, locate the following registry key:
`HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c. Set "Enabled" DWORD to "0x0" for the following registry keys:

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

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low



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 `regedt32` 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\
```

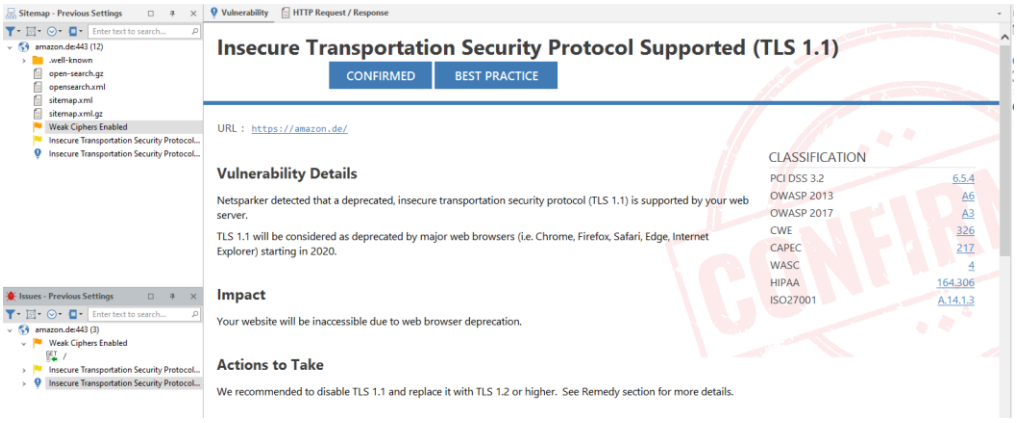
Locate a key named `Server` or create if it doesn't exist.

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

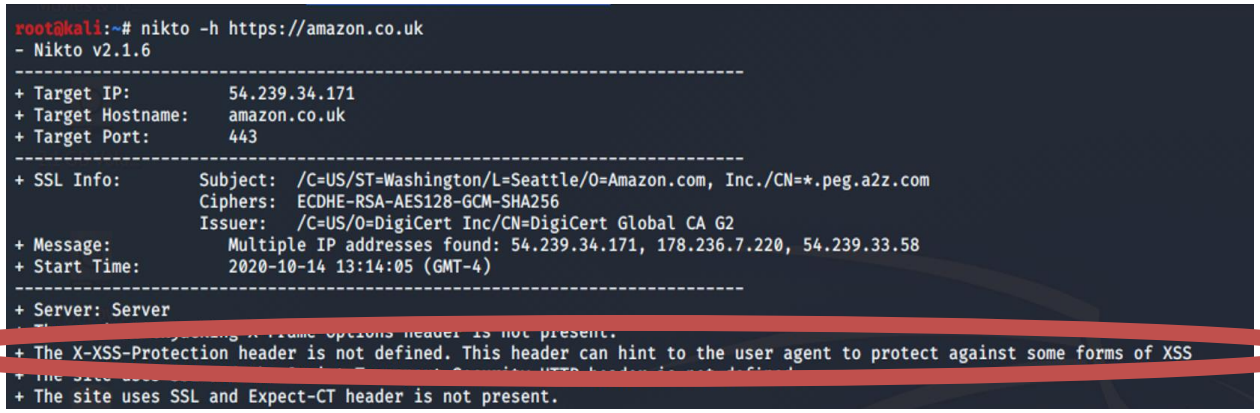


TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

Amazon.co.uk

Missing X-XXS protection header

Severity: Best Practice



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)

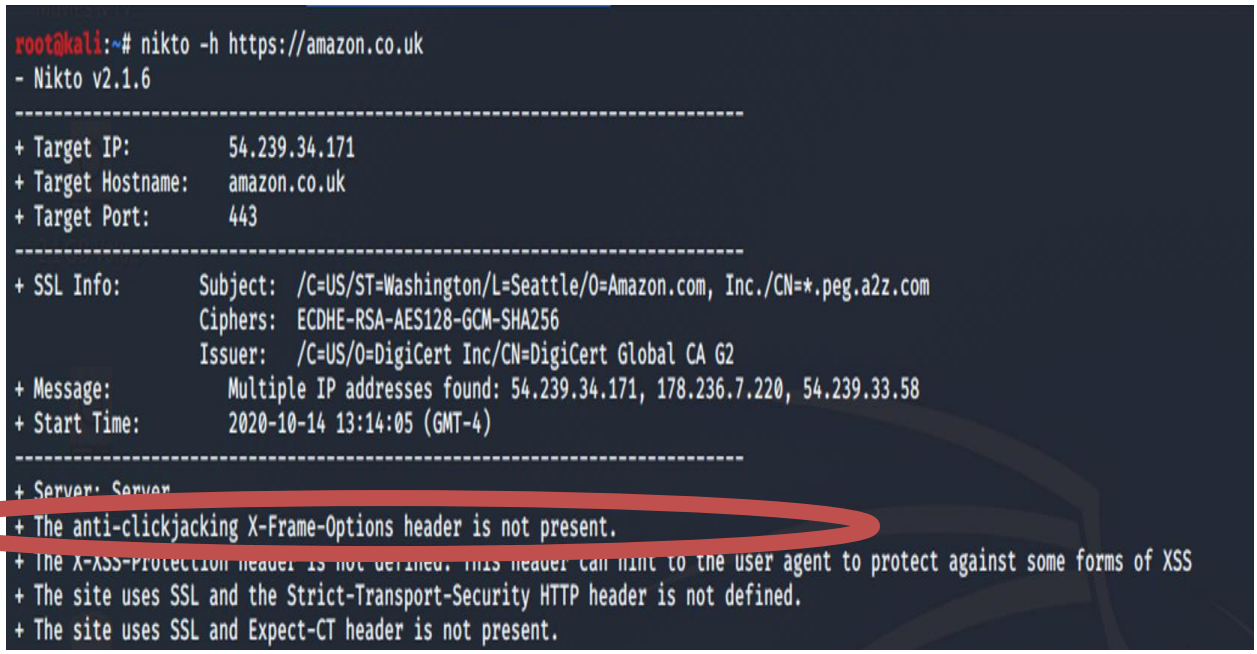
```
<IfModule mod_headers.c>
Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

Nginx

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

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

Severity: Low



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

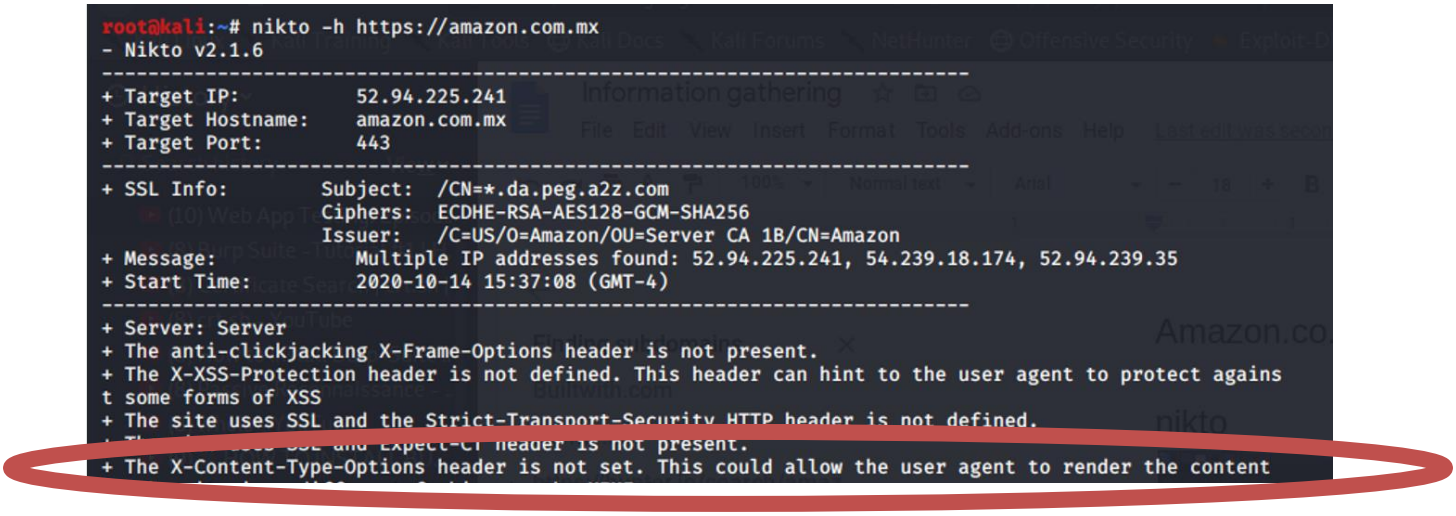
There are three values for the x-frame option header that instruct the browser to allow/disallow framing from other domains:

- 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



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.

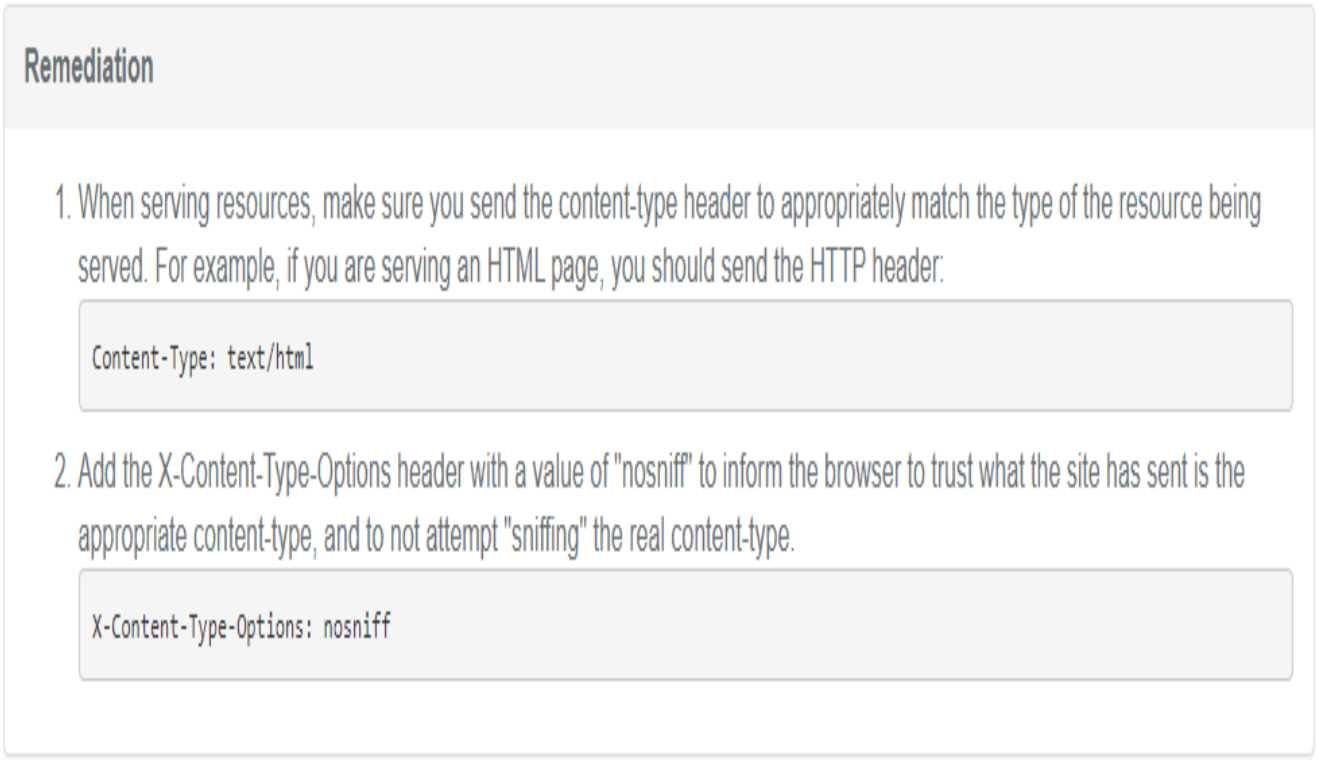
If the server does not return a correct x-content-type-options header, the website would be at risk of XXS attacks.

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

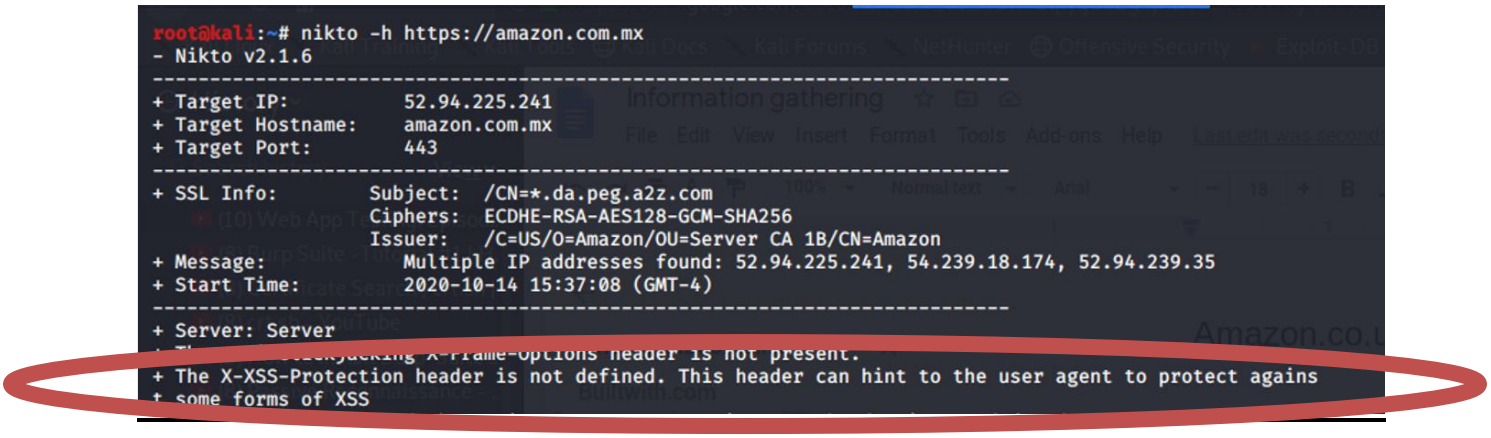
and to render the content as the HTML browser, it is possible to inject malicious code and make the victim execute it.

How to fix this issue.



Missing X-XXS protection header

Severity: Best Practice



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>

Cross-Site Scripting (XXS)

The attacker injects malicious scripts into a vulnerable website. 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)

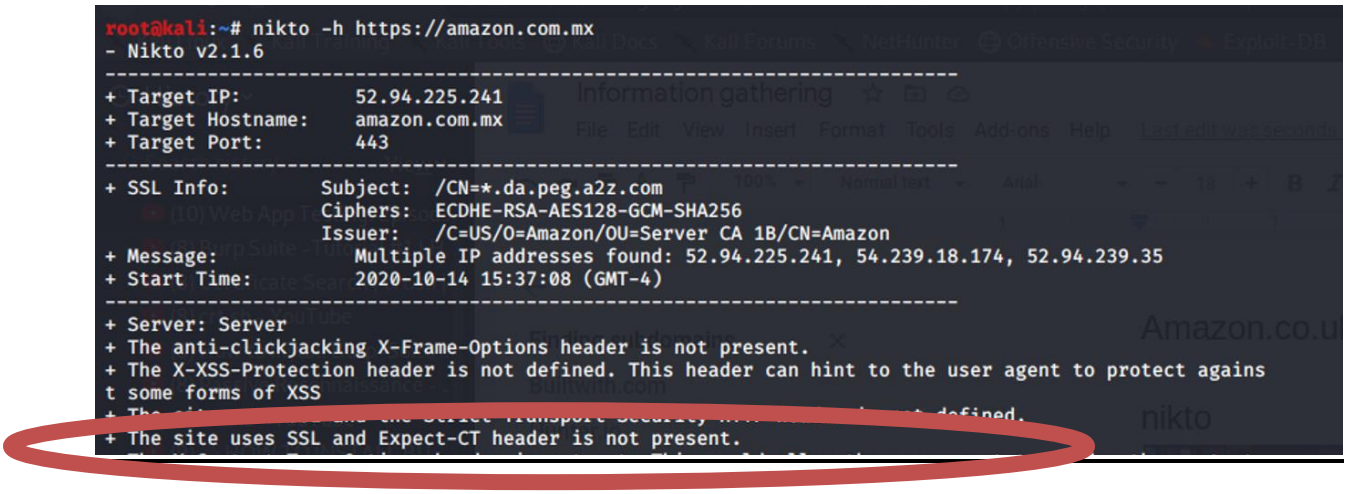
```
<IfModule mod_headers.c>
Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

Nginx

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

Expect – CT header is not defined

Severity: Best Practice



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

Remediation

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

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

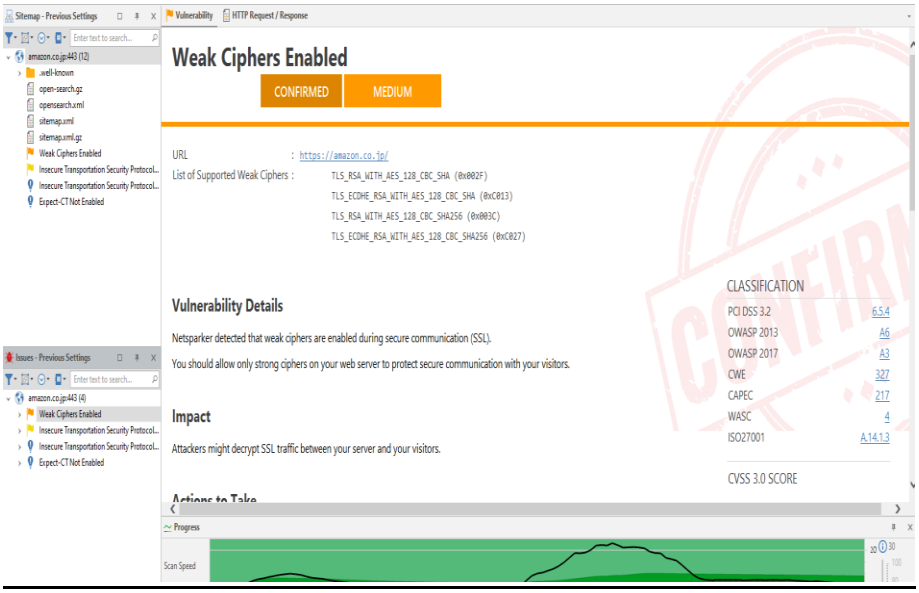
Note: We strongly suggest you to use Expect-CT header in **report-only mode** first. If everything goes well and your certificate is ready, go with the Expect-CT enforce mode. To use **report-only mode** first, omit **enforce** flag and see the browser's behavior with your deployed certificate.

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

Amazon.co.jp

Weak Ciphers Enabled

Severity: Medium



Weak 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 recognize 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"
```

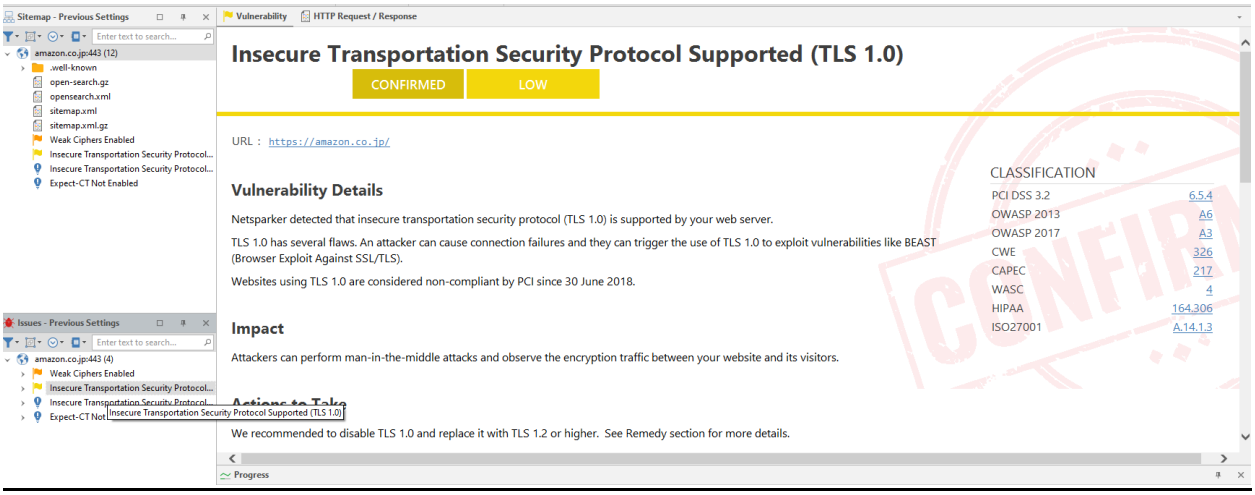
For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
b. In Registry Editor, locate the following registry key:
`HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c. Set "Enabled" DWORD to "0x0" for the following registry keys:

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

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low



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 `regedt32` 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.

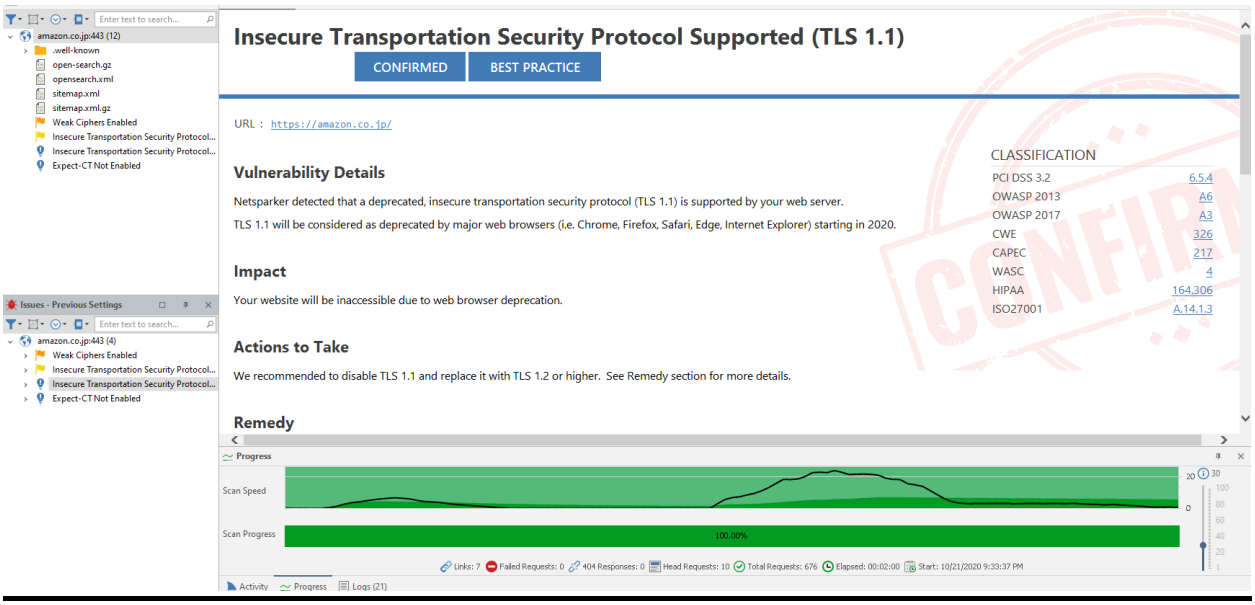
Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

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.

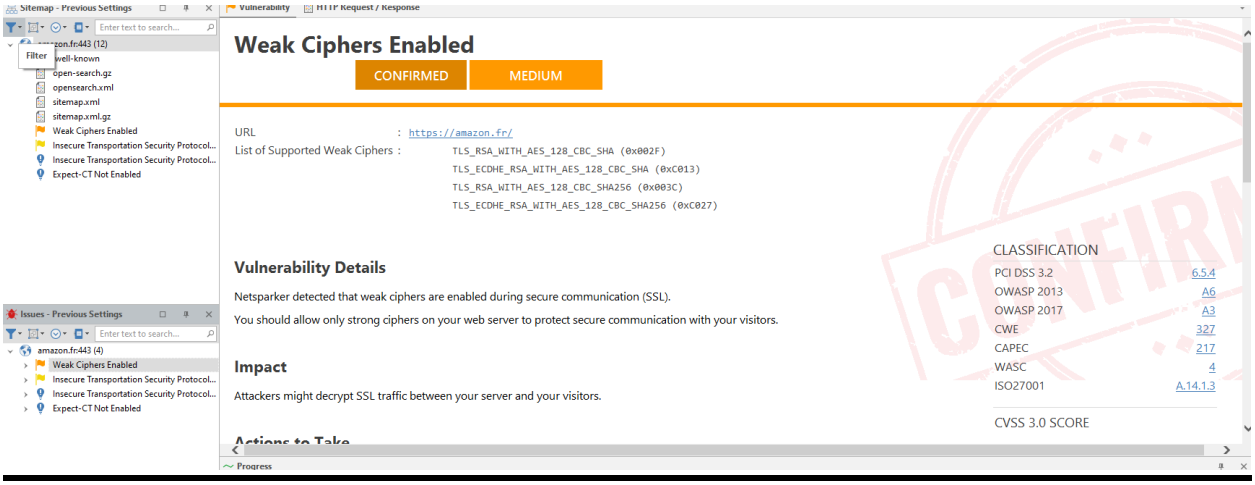
How to fix this issue

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

Amazon.fr

Weak Ciphers Enabled

Severity: Medium



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 recognize 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"
```

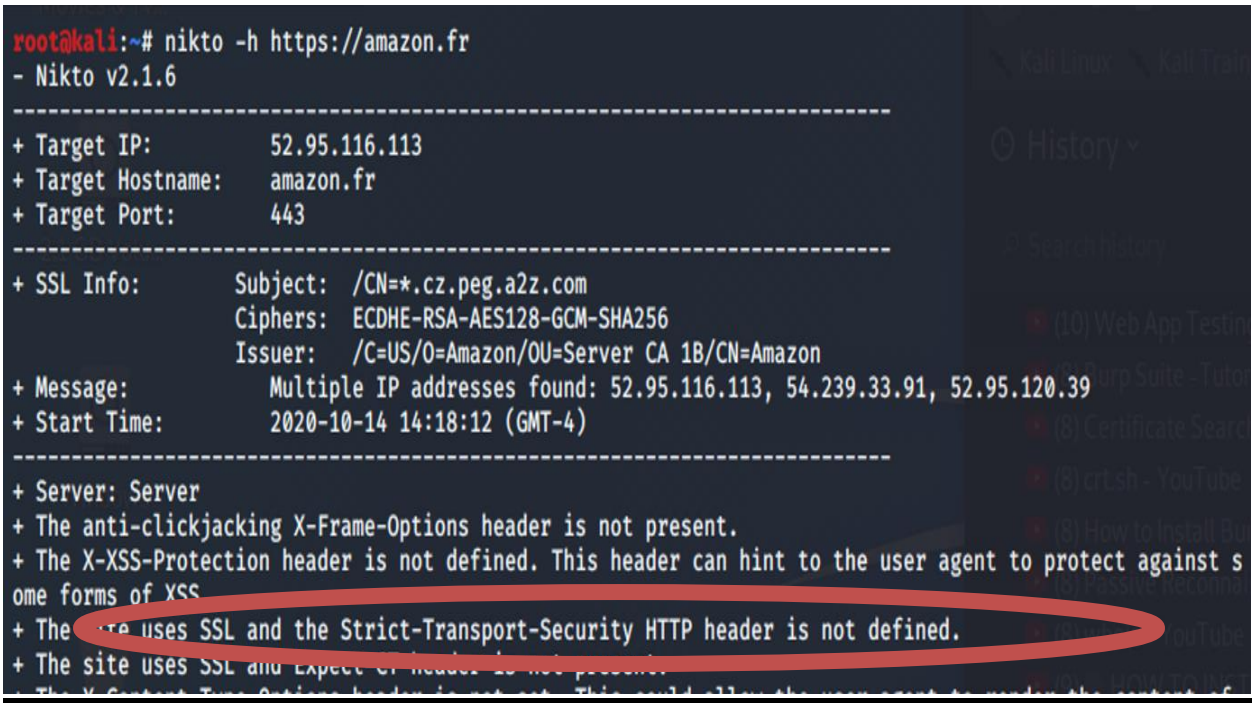
For Microsoft IIS, you should make some changes to the system registry. **Incorrectly editing the registry may severely damage your system. Before making changes to the registry, you should back up any valued data on your computer.**

a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
b. In Registry Editor, locate the following registry key:
`HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c. Set "Enabled" DWORD to "0x0" for the following registry keys:

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

Strict transport security HTTP header not defined

Severity: Medium



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

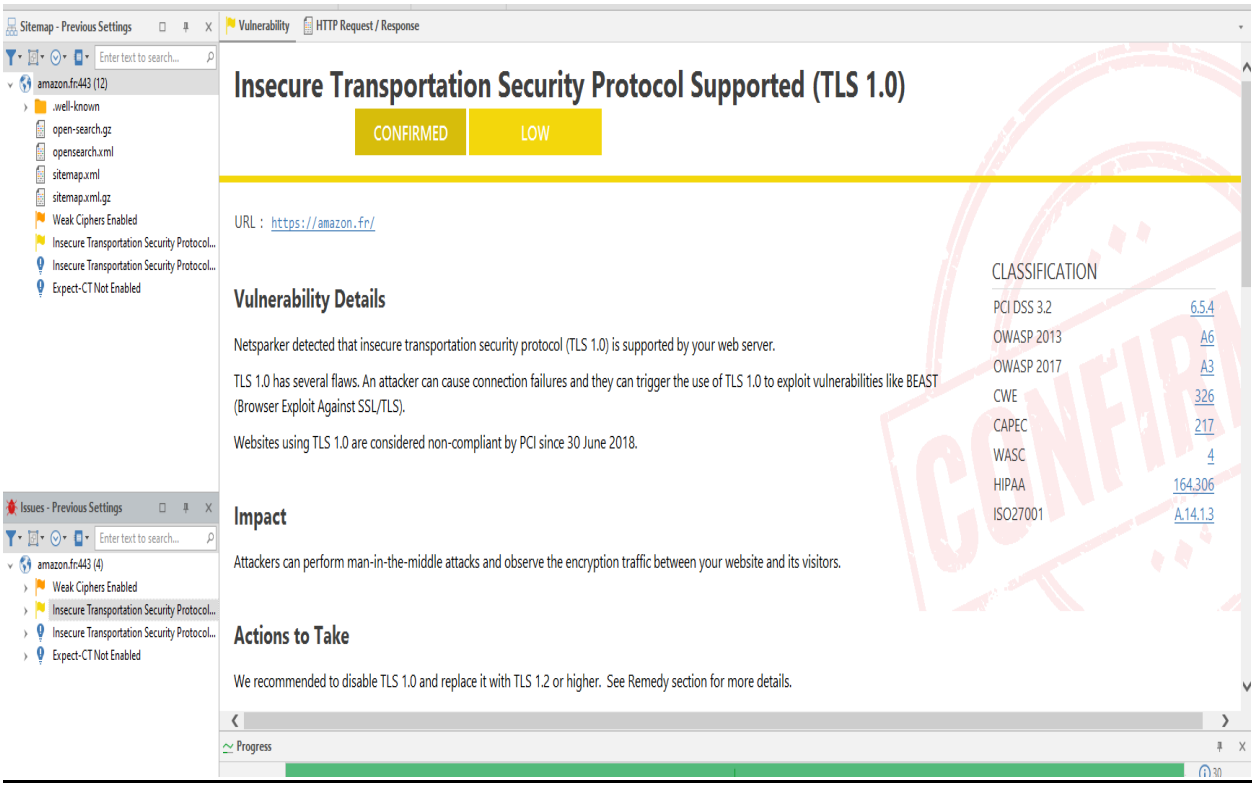
Configure your webserver to redirect HTTP requests to HTTPS.

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



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 `regedt32` 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.

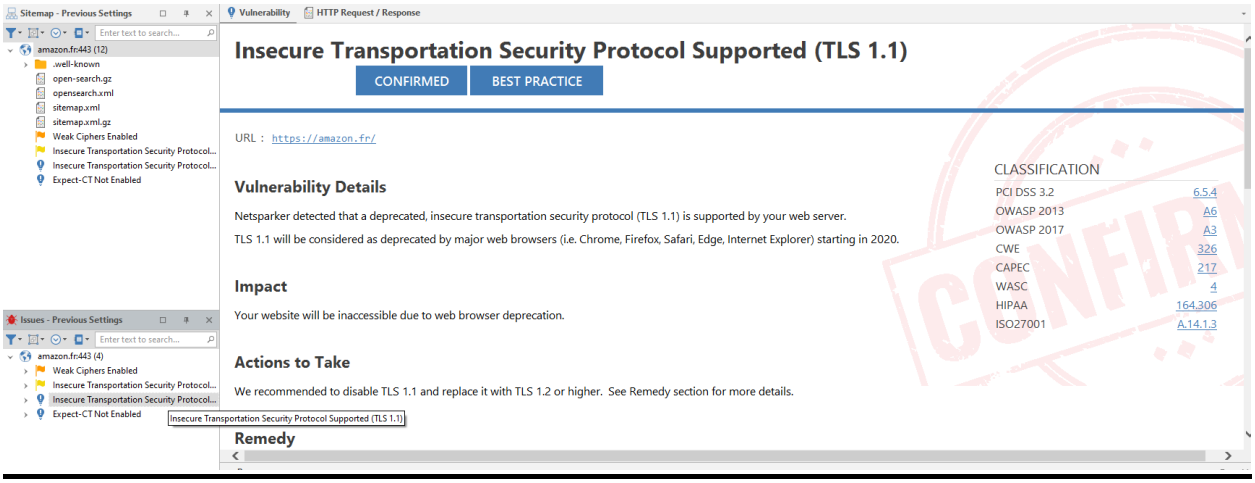
Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

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.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, 52.95.116.1
15
+ Start Time:         2020-10-14 15:37:49 (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 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.

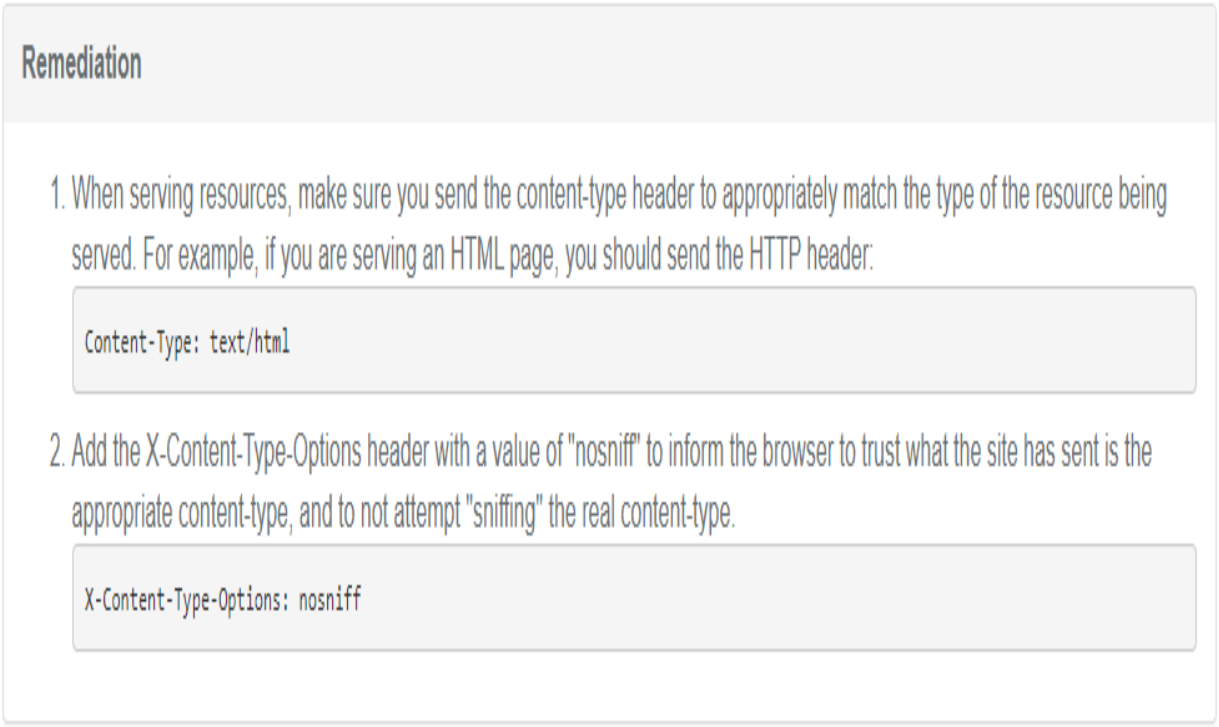
If the server does not return a correct x-content-type-options header, the website would be at risk of XXS attacks.

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

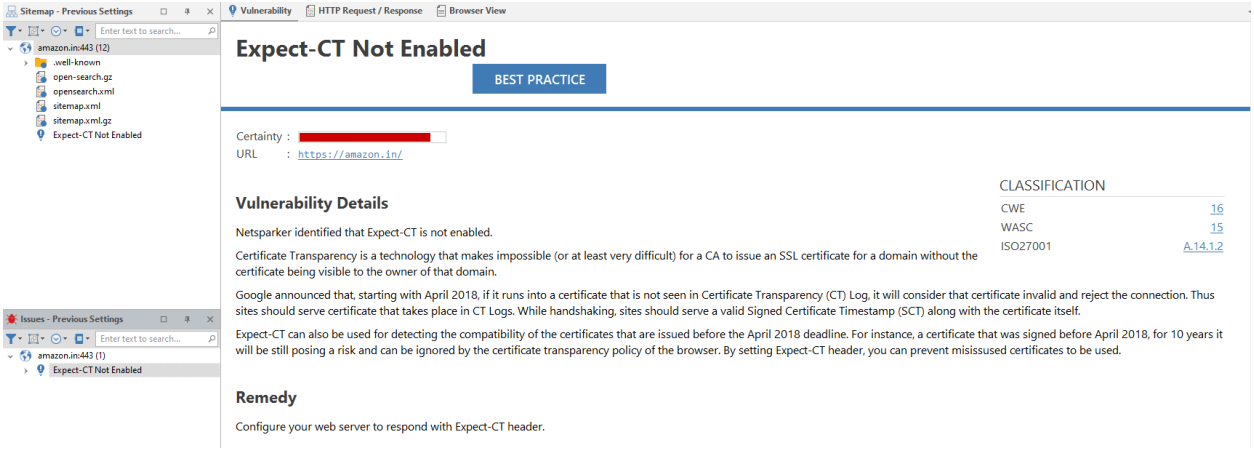
and to render the content as the HTML browser, it is possible to inject malicious code and make the victim execute it.

How to fix this issue.



Expect – CT header is not defined

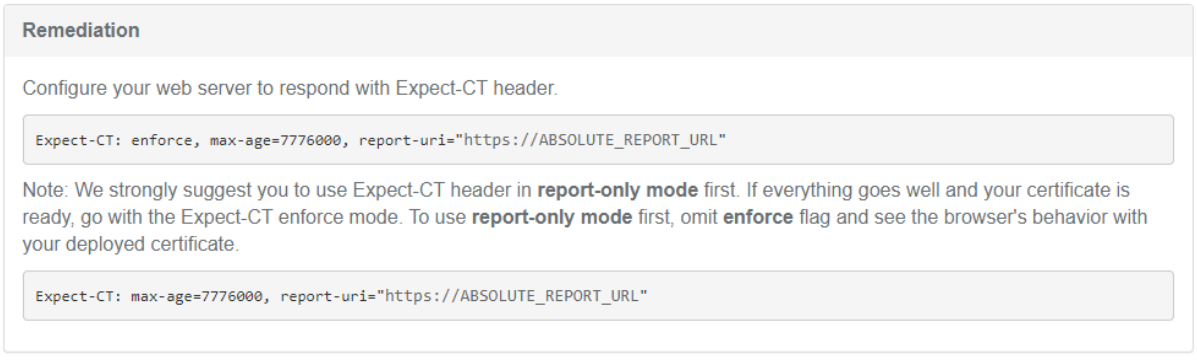
Severity: Best Practice



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.

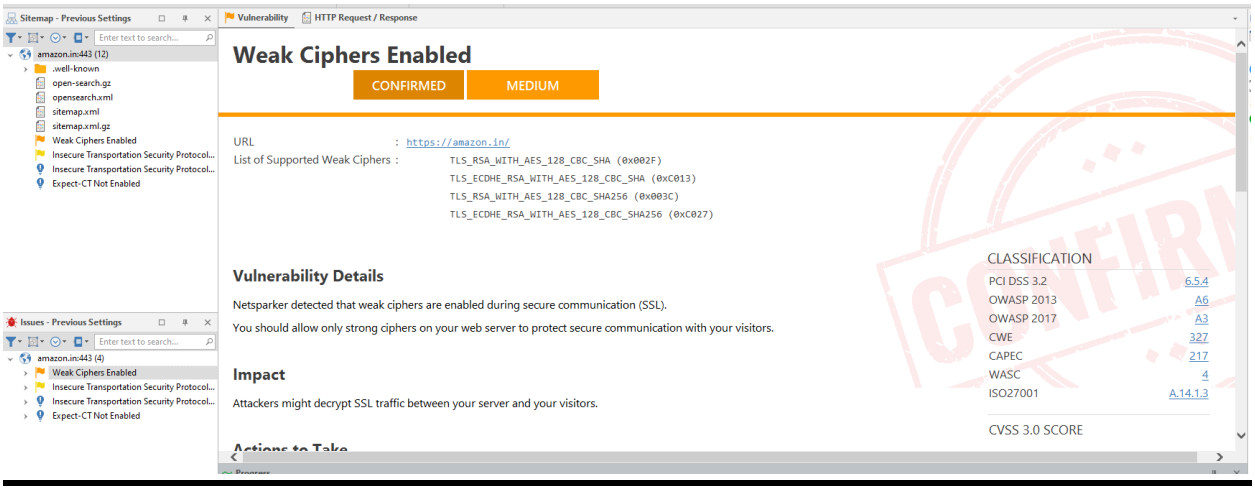
In 2018 Google announce that if it finds a website with a certificate that is not in the CT log it will consider the certificate is invalid and reject the connection.

How to fix this issue



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.

How to fix this issue

Configuring the web server to disallow using weak ciphers can be recognize 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.**

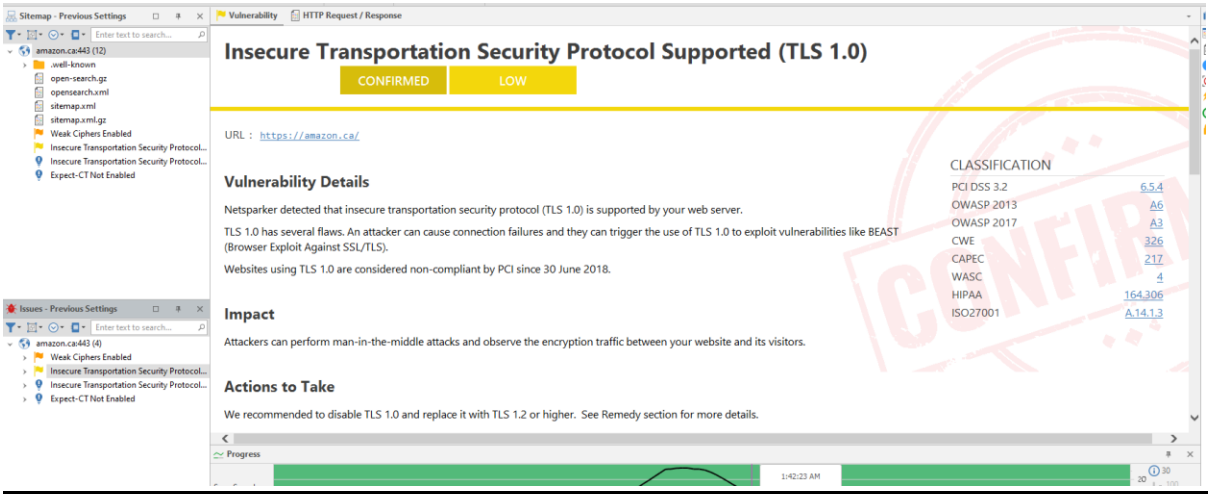
a. Click Start, click Run, type `regedt32` or type `regedit`, and then click OK.
b. In Registry Editor, locate the following registry key:
`HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders`
c. Set "Enabled" DWORD to "0x0" for the following registry keys:

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

Amazon.ca

Insecure Transport Security Protocol Supported (TSL 1.0)

Severity: Low



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 `regedt32` 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\
```

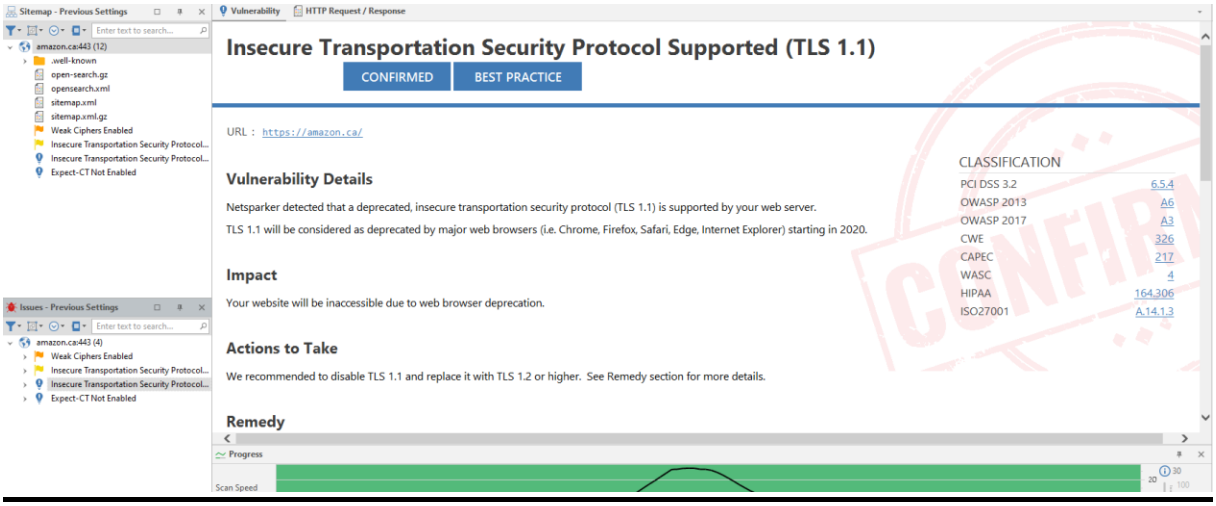
Locate a key named `Server` or create if it doesn't exist.
Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

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



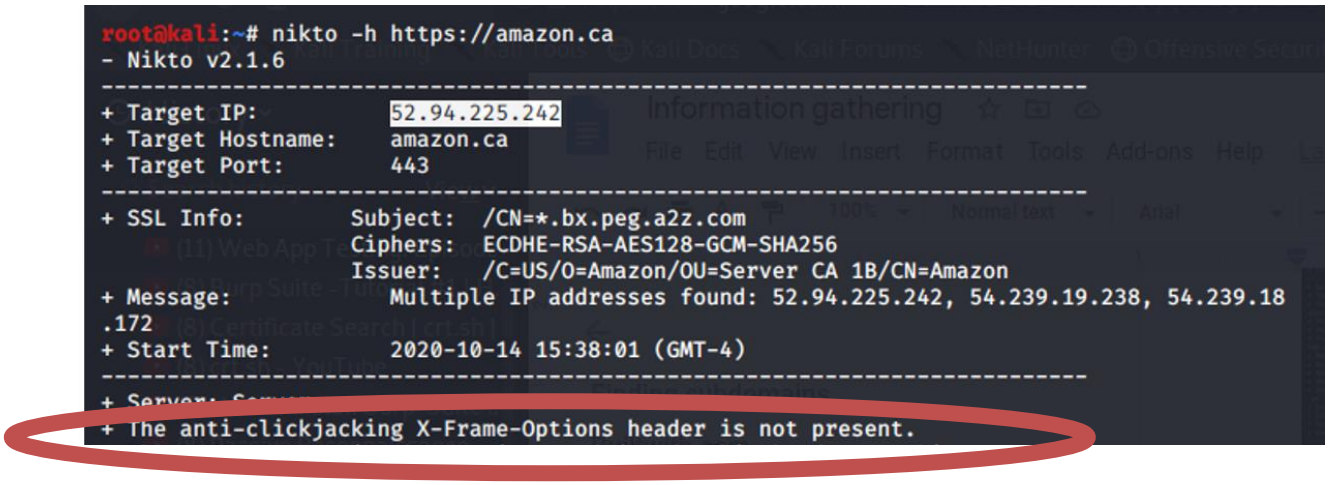
TSL 1.1 is also vulnerable and insecure and the impact of it is that the website will be inaccessible due to web browser deprecation. By 2021, google will stop loading websites with TSL 1.0 or 1.1.

How to fix this issue

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

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

Severity: Low



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

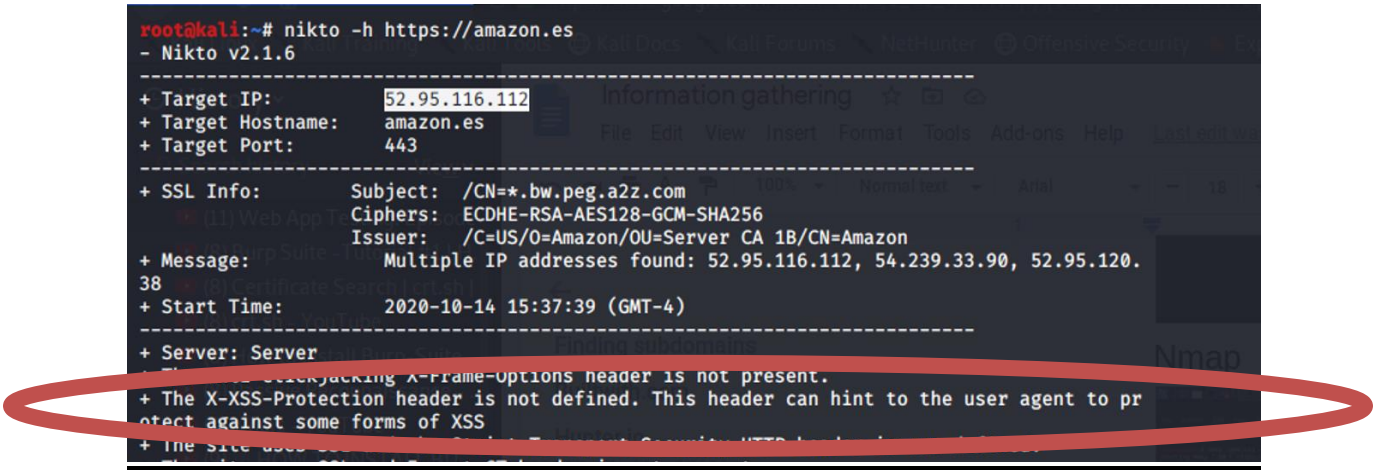
There are three values for the x-frame option header that instruct the browser to allow/disallow framing from other domains:

- 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



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>

How to fix this issue

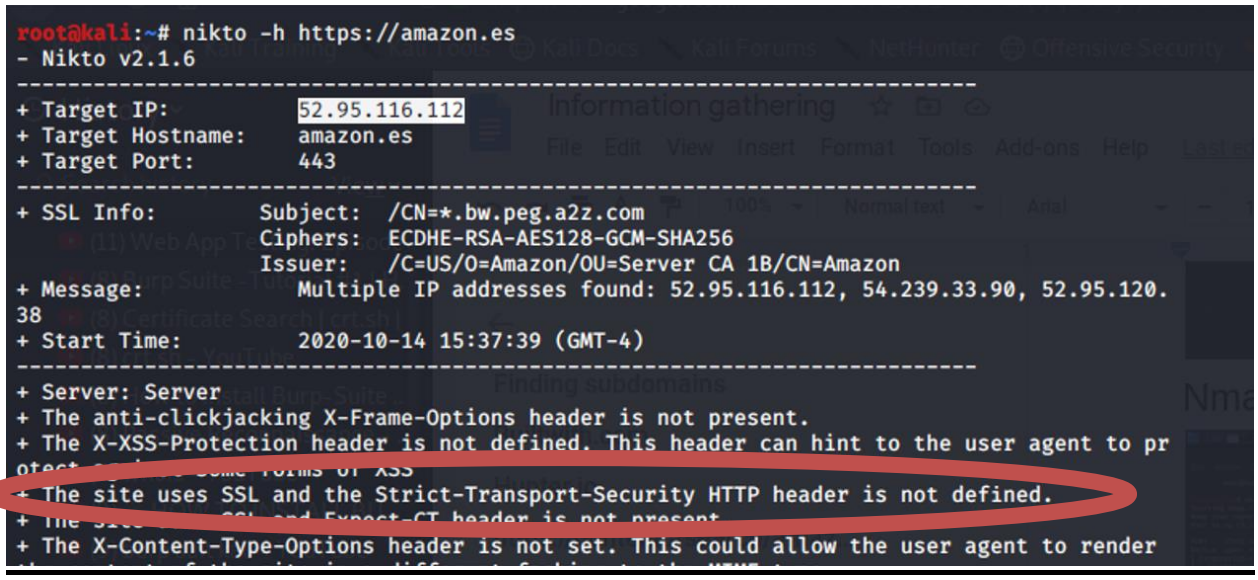
Add X-XXS protection header with value “1; mode=block”

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



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

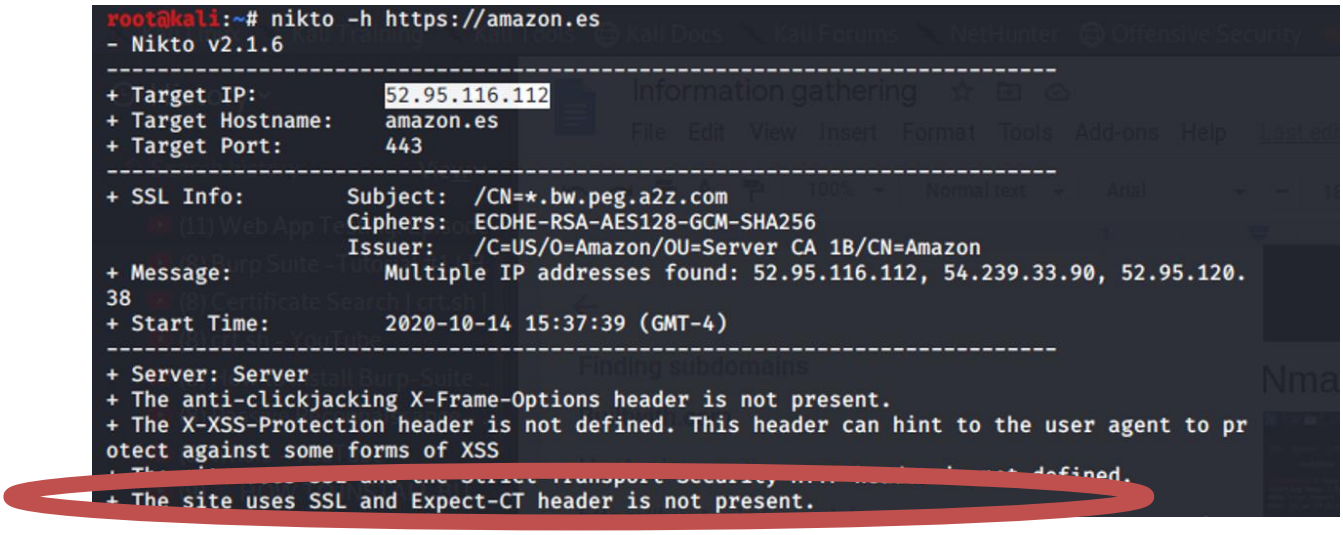
Configure your webserver to redirect HTTP requests to HTTPS.

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



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

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.