Fingerprinting

Fingerprinting focuses on extracting technical details about the technologies powering a website or web application. Similar to how a fingerprint uniquely identifies a person, the digital signatures of web servers, operating systems, and software components can reveal critical information about a target's infrastructure and potential security weaknesses. This knowledge empowers attackers to tailor attacks and exploit vulnerabilities specific to the identified technologies.

Fingerprinting serves as a cornerstone of web reconnaissance for several reasons:

- `Targeted Attacks`: By knowing the specific technologies in use, attackers can focus their efforts on exploits and vulnerabilities that are known to affect those systems. This significantly increases the chances of a successful compromise.
- `Identifying Misconfigurations`: Fingerprinting can expose misconfigured or outdated software, default settings, or other weaknesses that might not be apparent through other reconnaissance methods.
- `Prioritising Targets`: When faced with multiple potential targets, fingerprinting helps prioritise efforts by identifying systems more likely to be vulnerable or hold valuable information.
- `Building a Comprehensive Profile`: Combining fingerprint data with other reconnaissance findings creates a holistic view of the target's infrastructure, aiding in understanding its overall security posture and potential attack vectors.

Fingerprinting Techniques

There are several techniques used for web server and technology fingerprinting:

- `Banner Grabbing`: Banner grabbing involves analysing the banners presented by web servers and other services. These banners often reveal the server software, version numbers, and other details.
- `Analysing HTTP Headers`: HTTP headers transmitted with every web page request and response contain a wealth of information. The `Server` header typically discloses the web server software, while the `X-Powered-By` header might reveal additional technologies like scripting languages or frameworks.
- `Probing for Specific Responses`: Sending specially crafted requests to the target can elicit unique responses that reveal specific technologies or versions. For example, certain error messages or behaviours are characteristic of particular web servers or software components.
- `Analysing Page Content`: A web page's content, including its structure, scripts, and other elements, can often provide clues about the underlying technologies. There may be a copyright header that indicates specific software being used, for example.

A variety of tools exist that automate the fingerprinting process, combining various techniques to identify web servers, operating systems, content management systems, and other technologies:

Tool	Description	Features
`Wappalyzer`	Browser extension and online service for website technology profiling.	Identifies a wide range of web technologies, including CMSs, frameworks, analytics tools, and more.
`BuiltWith`	Web technology profiler that provides detailed reports on a website's technology stack.	Offers both free and paid plans with varying levels of detail.
`WhatWeb`	Command-line tool for website fingerprinting.	Uses a vast database of signatures to identify various web technologies.
`Nmap`	Versatile network scanner that can be used for various reconnaissance tasks, including service and OS fingerprinting.	Can be used with scripts (NSE) to perform more specialised fingerprinting.
`Netcraft`	Offers a range of web security services, including website fingerprinting and security reporting.	Provides detailed reports on a website's technology, hosting provider, and security posture.
`wafw00f`	Command-line tool specifically designed for identifying Web Application Firewalls (WAFs).	Helps determine if a WAF is present and, if so, its type and configuration.

Fingerprinting inlanefreight.com

Let's apply our fingerprinting knowledge to uncover the digital DNA of our purpose-built host, `inlanefreight.com`. We'll leverage both manual and automated techniques to gather information about its web server, technologies, and potential vulnerabilities.

Banner Grabbing

Our first step is to gather information directly from the web server itself. We can do this using the `curl` command with the `-I` flag (or `--head`) to fetch only the HTTP headers, not the entire page content.

Fingerprinting

```
MisaelMacias@htb[/htb]$ curl -I inlanefreight.com
```

The output will include the server banner, revealing the web server software and version number:

Fingerprinting

```
MisaelMacias@htb[/htb]$ curl -I inlanefreight.com

HTTP/1.1 301 Moved Permanently
Date: Fri, 31 May 2024 12:07:44 GMT
Server: Apache/2.4.41 (Ubuntu)
Location: https://inlanefreight.com/
Content-Type: text/html; charset=iso-8859-1
```

In this case, we see that `inlanefreight.com` is running on `Apache/2.4.41`, specifically the `Ubuntu` version. This information is our first clue

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

In this case, we see that inlanefreight.com is running on Apache/2.4.41, specifically the Ubuntu version. This information is our first clue, hinting at the underlying technology stack. It's also trying to redirect to `https://inlanefreight.com/` so grab those banners too

Fingerprinting

```
MisaelMacias@htb[/htb]$ curl -I https://inlanefreight.com

HTTP/1.1 301 Moved Permanently
Date: Fri, 31 May 2024 12:12:12 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Redirect-By: WordPress
Location: https://www.inlanefreight.com/
Content-Type: text/html; charset=UTF-8
```

We now get a really interesting header, the server is trying to redirect us again, but this time we see that it's `WordPress` that is doing the redirection to `https://www.inlanefreight.com/`

Fingerprinting

```
MisaelMacias@htb[/htb]$ curl -I https://www.inlanefreight.com

HTTP/1.1 200 OK
Date: Fri, 31 May 2024 12:12:26 GMT
Server: Apache/2.4.41 (Ubuntu)
Link: <https://www.inlanefreight.com/index.php/wp-json/>; rel="https://api.w.org/"
Link: <https://www.inlanefreight.com/index.php/wp-json/wp/v2/pages/7>; rel="alternate"; type="application/json"
Link: <https://www.inlanefreight.com/>; rel=shortlink
Content-Type: text/html; charset=UTF-8
```

A few more interesting headers, including an interesting path that contains `wp-json`. The `wp-` prefix is common to WordPress.

Wafw00f

`Web Application Firewalls` (`WAFs`) are security solutions designed to protect web applications from various attacks. Before proceeding with further fingerprinting, it's crucial to determine if `inlanefreight.com` employs a WAF, as it could interfere with our probes or potentially block our requests.

To detect the presence of a WAF, we'll use the `wafw00f` tool. To install `wafw00f`, you can use pip3:

Fingerprinting

```
MisaelMacias@htb[/htb]$ pip3 install git+https://github.com/EnableSecurity/wafw00f
```

Once it's installed, pass the domain you want to check as an argument to the tool:

Fingerprinting

```
MisaelMacias@htb[/htb]$ wafw00f inlanefreight.com

                ______
               /      \
              (  W00f! )
               \  ____/
                ,,    __           404 Hack Not Found
           |`-.__   / /                      __     __
           /"  _/  /_/                      \ \   / /
          *===*    /                         \ \_/ /  405 Not Allowed
         /     )__//                         \   /
    /|  /     /---`                          403 Forbidden
    \\/`   \ |                                 / _ \
    `\    /_\\_              502 Bad Gateway  / / \ \  500 Internal Error
      `_____ ``                              /_/   \_\

                     ~ WAFW00F : v2.2.0 ~
        The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://inlanefreight.com
[+] The site https://inlanefreight.com is behind Wordfence (Defiant) WAF.
[~] Number of requests: 2
```

The `wafw00f` scan on `inlanefreight.com` reveals that the website is protected by the `Wordfence Web Application Firewall` (`WAF`), developed by Defiant.

This means the site has an additional security layer that could block or filter our reconnaissance attempts. In a real-world scenario, it would be crucial to keep this in mind as you proceed with further investigation, as you might need to adapt techniques to bypass or evade the WAF's detection mechanisms.

Nikto

`Nikto` is a powerful open-source web server scanner. In addition to its primary function as a vulnerability assessment tool, `Nikto's` fingerprinting capabilities provide insights into a website's technology stack.

`Nikto` is pre-installed on pwnbox, but if you need to install it, you can run the following commands:

Fingerprinting

```
MisaelMacias@htb[/htb]$ sudo apt update && sudo apt install -y perl
MisaelMacias@htb[/htb]$ git clone https://github.com/sullo/nikto
MisaelMacias@htb[/htb]$ cd nikto/program
MisaelMacias@htb[/htb]$ chmod +x ./nikto.pl
```

To scan `inlanefreight.com` using `Nikto`, only running the fingerprinting modules, execute the following command:

Fingerprinting

```
MisaelMacias@htb[/htb]$ nikto -h inlanefreight.com -Tuning b
```

The `-h` flag specifies the target host. The `-Tuning b` flag tells `Nikto` to only run the Software Identification modules.

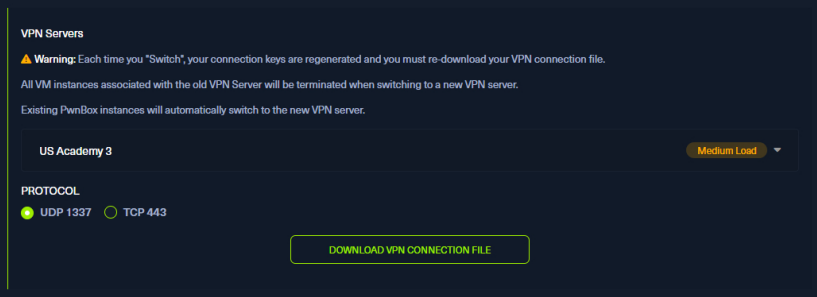
`Nikto` will then initiate a series of tests, attempting to identify outdated software, insecure files or configurations, and other potential security risks.

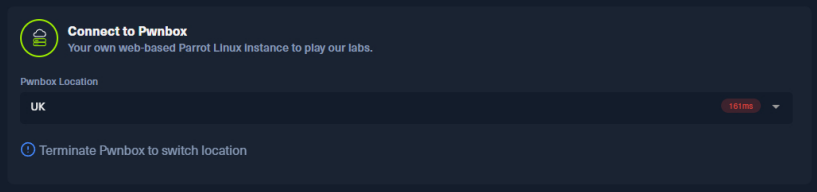
Fingerprinting

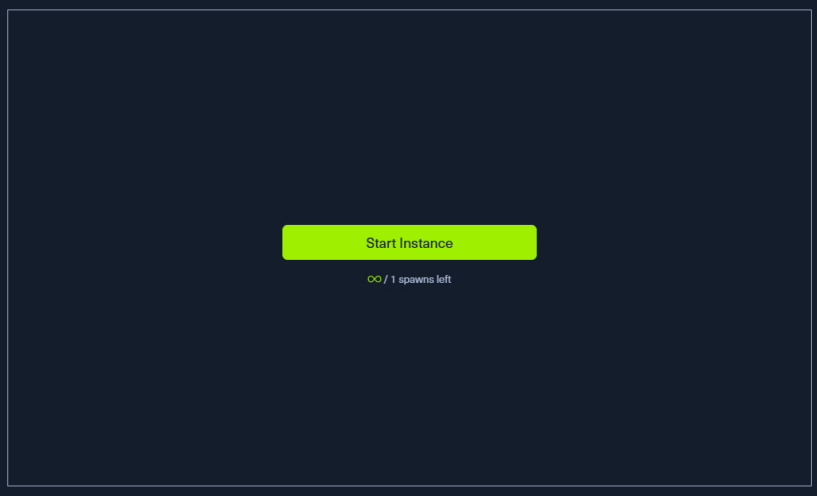
```
MisaelMacias@htb[/htb]$ nikto -h inlanefreight.com -Tuning b

- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 134.209.24.248, 2a03:b0c0:1:e0::32c:b001
+ Target IP:          134.209.24.248
+ Target Hostname:    www.inlanefreight.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=inlanefreight.com
                   Altnames: inlanefreight.com, www.inlanefreight.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=US/O=Let's Encrypt/CN=R3
+ Start Time:         2024-05-31 13:35:54 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: Link header found with value: ARRAY(0x558e78790248). See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Heade
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not defined. See: https://developer.mozilla.or
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site i
+ /index.php?: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: The Content-Encoding header is set to "deflate" which may mean that the server is vulnerable to the BREACH attack
+ Apache/2.4.41 appears to be outdated (current is at least 2.4.59). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /license.txt: License file found may identify site software.
+ /: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://develope
+ /wp-login.php:X-Frame-Options header is deprecated and has been replaced with the Content-Security-Policy HTTP heade
+ /wp-login.php: Wordpress login found.
+ 1316 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2024-05-31 13:47:27 (GMT0) (693 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

The reconnaissance scan on `inlanefreight.com` reveals several key findings:

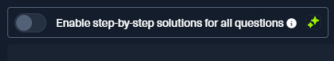
- `IPs`: The website resolves to both IPv4 (`134.209.24.248`) and IPv6 (`2a03:b0c0:1:e0::32c:b001`) addresses.
- `Server Technology`: The website runs on `Apache/2.4.41 (Ubuntu)`
- `WordPress Presence`: The scan identified a WordPress installation, including the login page (`/wp-login.php`). This suggests the site might be a potential target for common WordPress-related exploits.
- `Information Disclosure`: The presence of a `license.txt` file could reveal additional details about the website's software components.
- `Headers`: Several non-standard or insecure headers were found, including a missing `Strict-Transport-Security` header and a potentially insecure `x-redirect-by` header.







Waiting to start...



Questions

Answer the question(s) below to complete this Section and earn cubes!

Cheat Sheet

Download VPN Connection File

Target(s): Click here to spawn the target system!

vHosts needed for these questions:

- app.inlanefreight.local
- dev.inlanefreight.local

+ 1 Determine the Apache version running on app.inlanefreight.local on the target system. (Format: 0.0.0)

2.4.41

Submit

+ 1 Which CMS is used on app.inlanefreight.local on the target system? Respond with the name only, e.g., WordPress.

Joomla

Submit

+ 1 On which operating system is the dev.inlanefreight.local webserver running in the target system? Respond with the name only, e.g., Debian.

ubuntu

Submit

Previous Next

Mark Complete & Next