Web Enumeration

When performing service scanning, we will often run into web servers running on ports 80 and 443. Webservers host web applications (sometimes more than 1) which often provide a considerable attack surface and a very high-value target during a penetration test. Proper web enumeration is critical, especially when an organization is not exposing many services or those services are appropriately patched.

Gobuster

After discovering a web application, it is always worth checking to see if we can uncover any hidden files or directories on the webserver that are not intended for public access. We can use a tool such as ffuf or GoBuster to perform this directory enumeration. Sometimes we will find hidden functionality or pages/directories exposing sensitive data that can be leveraged to access the web application or even remote code execution on the web server itself.

Directory/File Enumeration

GoBuster is a versatile tool that allows for performing DNS, vhost, and directory brute-forcing. The tool has additional functionality, such as enumeration of public AWS S3 buckets. For this module's purposes, we are interested in the directory (and file) brute-forcing modes specified with the switch `dir`. Let us run a simple scan using the `dirb common.txt` wordlist.

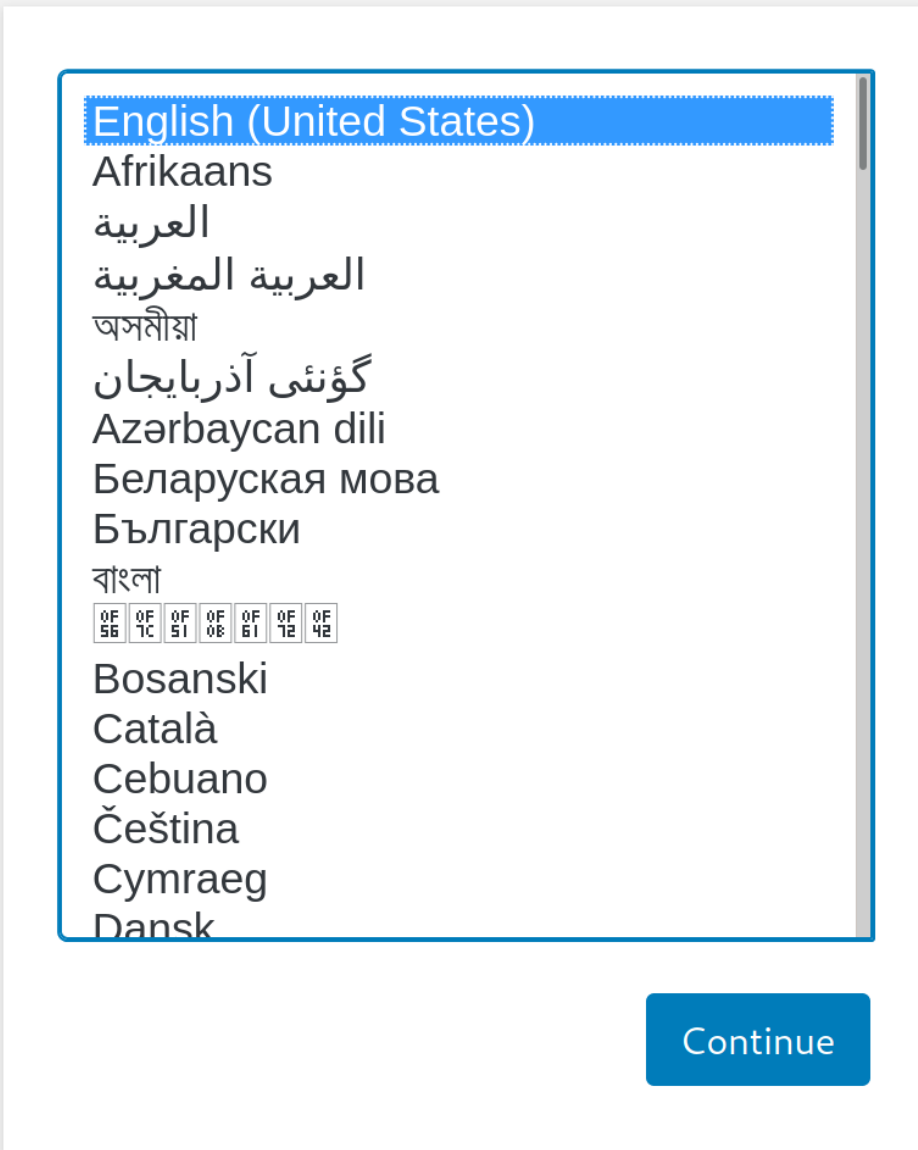
Directory/File Enumeration

```
MichaelLuka@htb[/htb]$ gobuster dir -u http://10.10.10.121/ -w /usr/share/dirb/wordlists/common.txt

===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.10.121/
[+] Threads:        10
[+] Wordlist:       /usr/share/dirb/wordlists/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/12/11 21:47:25 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/index.php (Status: 200)
/server-status (Status: 403)
/wordpress (Status: 301)
===============================================================
2020/12/11 21:47:46 Finished
===============================================================
```

An HTTP status code of `200` reveals that the resource's request was successful, while a 403 HTTP status code indicates that we are forbidden to access the resource. A 301 status code indicates that we are being redirected, which is not a failure case. It is worth familiarizing ourselves with the various HTTP status codes, which can be found here. The `Web Requests` Academy Module also covers HTTP status codes further in-depth.

The scan was completed successfully, and it identifies a WordPress installation at `/wordpress`. WordPress is the most commonly used CMS (Content Management System) and has an enormous potential attack surface. In this case, visiting `http://10.10.10.121/wordpress` in a browser reveals that WordPress is still in setup mode, which will allow us to gain remote code execution (RCE) on the server.



DNS Subdomain Enumeration

There also may be essential resources hosted on subdomains, such as admin panels or applications with additional functionality that could be exploited. We can use `GoBuster` to enumerate available subdomains of a given domain using the `dns` flag to specify DNS mode. First, let us clone the SecLists GitHub repo, which contains many useful lists for fuzzing and exploitation:

Install SecLists

Install SecLists

```
MichaelLuka@htb[/htb]$ git clone https://github.com/danielmiessler/SecLists
```

Install SecLists

```
MichaelLuka@htb[/htb]$ sudo apt install seclists -y
```

Next, add a DNS Server such as 1.1.1.1 to the `/etc/resolv.conf` file. We will target the domain `inlanefreight.com`, the website for a fictional freight and logistics company.

Install SecLists

```
MichaelLuka@htb[/htb]$ gobuster dns -d inlanefreight.com -w /usr/share/SecLists/Discovery/DNS/namelist.txt

===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Domain:     inlanefreight.com
[+] Threads:    10
[+] Timeout:    1s
[+] Wordlist:   /usr/share/SecLists/Discovery/DNS/namelist.txt
===============================================================
2020/12/17 23:08:55 Starting gobuster
===============================================================
Found: blog.inlanefreight.com
Found: customer.inlanefreight.com
Found: my.inlanefreight.com
Found: ns1.inlanefreight.com
Found: ns2.inlanefreight.com
Found: ns3.inlanefreight.com
===============================================================
2020/12/17 23:10:34 Finished
===============================================================
```

This scan reveals several interesting subdomains that we could examine further. The Attacking Web Applications with Ffuf module goes into more details about web enumeration and fuzzing.

Web Enumeration Tips

Let us walk through a few additional web enumeration tips that will help complete machines on HTB and in the real world.

Banner Grabbing / Web Server Headers

In the last section, we discussed banner grabbing for general purposes. Web server headers provide a good picture of what is hosted on a web server. They can reveal the specific application framework in use, the authentication options, and whether the server is missing essential security options or has been misconfigured. We can use `cURL` to retrieve server header information from the command line. `cURL` is another essential addition to our penetration testing toolkit, and familiarity with its many options is encouraged.

Banner Grabbing / Web Server Headers

```
MichaelLuka@htb[/htb]$ curl -IL https://www.inlanefreight.com

HTTP/1.1 200 OK
Date: Fri, 18 Dec 2020 22:24:05 GMT
Server: Apache/2.4.29 (Ubuntu)
Link: <https://www.inlanefreight.com/index.php/wp-json/>; rel="https://api.w.org/"
Link: <https://www.inlanefreight.com/>; rel=shortlink
Content-Type: text/html; charset=UTF-8
```

Another handy tool is EyeWitness, which can be used to take screenshots of target web applications, fingerprint them, and identify possible default credentials.

Whatweb

We can extract the version of web servers, supporting frameworks, and applications using the command-line tool `whatweb`. This information can help us pinpoint the technologies in use and begin to search for potential vulnerabilities.

Whatweb

```
MichaelLuka@htb[/htb]$ whatweb 10.10.10.121

http://10.10.10.121 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], Email[license@php.net], HTTPServer[Ubuntu Linux]
```

Whatweb is a handy tool and contains much functionality to automate web application enumeration across a network.

Whatweb

```
MichaelLuka@htb[/htb]$ whatweb --no-errors 10.10.10.0/24

http://10.10.10.11 [200 OK] Country[RESERVED][ZZ], HTTPServer[nginx/1.14.1], IP[10.10.10.11], PoweredBy[Red,nginx],
http://10.10.10.100 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)]
http://10.10.10.121 [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], Email[license@php.net], HTTPServer[Ubuntu Linux]
http://10.10.10.247 [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[contact@cross-fit.htb], Frame, HTML5, HTTPServe
```

Certificates

SSL/TLS certificates are another potentially valuable source of information if HTTPS is in use. Browsing to `https://10.10.10.121/` and viewing the certificate reveals the details below, including the email address and company name. These could potentially be used to conduct a phishing attack if this is within the scope of an assessment.

Certificate

Networks

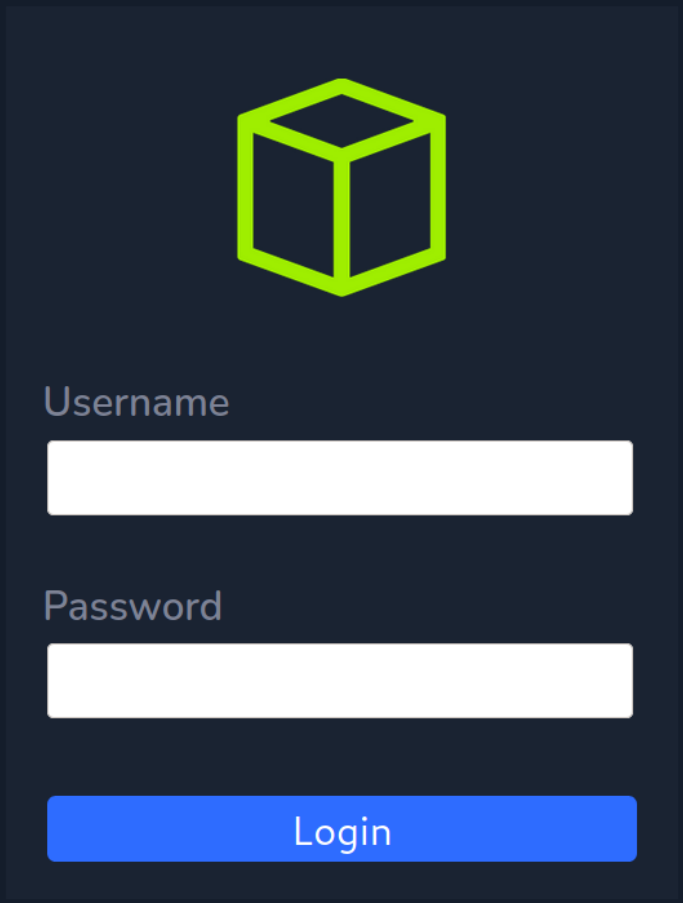
Subject Name	
Country	GB
State/Province	London
Locality	London
Organization	Megabank Limited
Organizational Unit	IT
Common Name	Networks
Email Address	networks@megabank.htb
Issuer Name	
Country	GB
State/Province	London
Locality	London
Organization	Megabank Limited
Organizational Unit	IT
Common Name	Networks
Email Address	networks@megabank.htb

Robots.txt

It is common for websites to contain a `robots.txt` file, whose purpose is to instruct search engine web crawlers such as Googlebot which resources can and cannot be accessed for indexing. The `robots.txt` file can provide valuable information such as the location of private files and admin pages. In this case, we see that the `robots.txt` file contains two disallowed entries.

```
User-agent: *
Disallow: /private
Disallow: /uploaded_files
```

Navigating to `http://10.10.10.121/private` in a browser reveals a HTB admin login page.



Source Code

It is also worth checking the source code for any web pages we come across. We can hit `[CTRL + U]` to bring up the source code window in a browser. This example reveals a developer comment containing credentials for a test account, which could be used to log in to the website.

```
<!--test account: egre55 / password1-->

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Admin Login</title>
<link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">

```

Start Instance

1 / 1 spawns left

Waiting to start...

Questions

Cheat Sheet

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

Target: Click here to spawn the target system!

+ 1 Try running some of the web enumeration techniques you learned in this section on the server above, and use the info you get to get the flag.

Submit your answer here...

Submit

Hint

Previous Next

Cheat Sheet

Go to Questions

Table of Contents

My Workstation

OFFLINE

Start Instance

1 / 1 spawns left