Nibbles - Web Footprinting

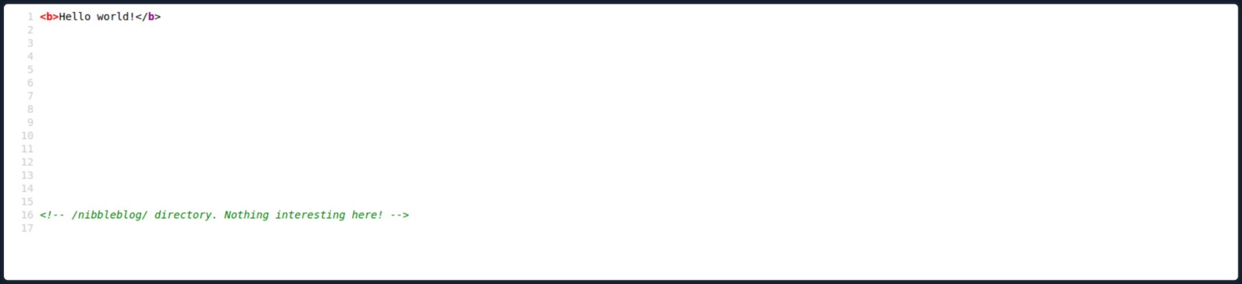
We can use `whatweb` to try to identify the web application in use.

```
MichaelLuka@htb[/htb]$ whatweb 10.129.42.190

http://10.129.42.190 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (U
```

This tool does not identify any standard web technologies in use. Browsing to the target in `Firefox` shows us a simple "Hello world!" message.

Hello world!

Checking the page source reveals an interesting comment.



```
<b>Hello world!</b>















<!-- /nibbleblog/ directory. Nothing interesting here! -->
```

We can also check this with cURL.

```
MichaelLuka@htb[/htb]$ curl http://10.129.42.190

<b>Hello world!</b>

<!-- /nibbleblog/ directory. Nothing interesting here! -->
```

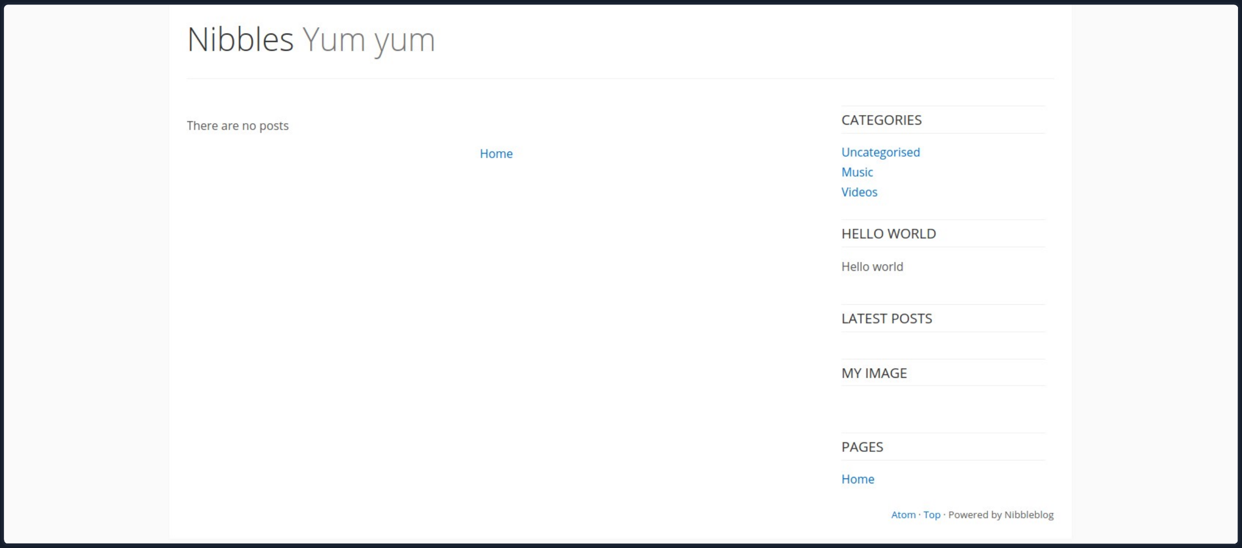
The HTML comment mentions a directory named `nibbleblog`. Let us check this with `whatweb`.

```
MichaelLuka@htb[/htb]$ whatweb http://10.129.42.190/nibbleblog

http://10.129.42.190/nibbleblog [301 Moved Permanently] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubun
http://10.129.42.190/nibbleblog/ [200 OK] Apache[2.4.18], Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HT
```

Now we are starting to get a better picture of things. We can see some of the technologies in use such as HTML5, jQuery, and PHP. We can also see that the site is running Nibbleblog, which is a free blogging engine built using PHP.

Directory Enumeration

Browsing to the `/nibbleblog` directory in `Firefox`, we do not see anything exciting on the main page.



A quick Google search for "nibbleblog exploit" yields this Nibblblog File Upload Vulnerability. The flaw allows an authenticated attacker to upload and execute arbitrary PHP code on the underlying web server. The `Metasploit` module in question works for version `4.0.3`. We do not know the exact version of `Nibbleblog` in use yet, but it is a good bet that it is vulnerable to this. If we look at the source code of the `Metasploit` module, we can see that the exploit uses user-supplied credentials to authenticate the admin portal at `/admin.php`.

Let us use Gobuster to be thorough and check for any other accessible pages/directories.

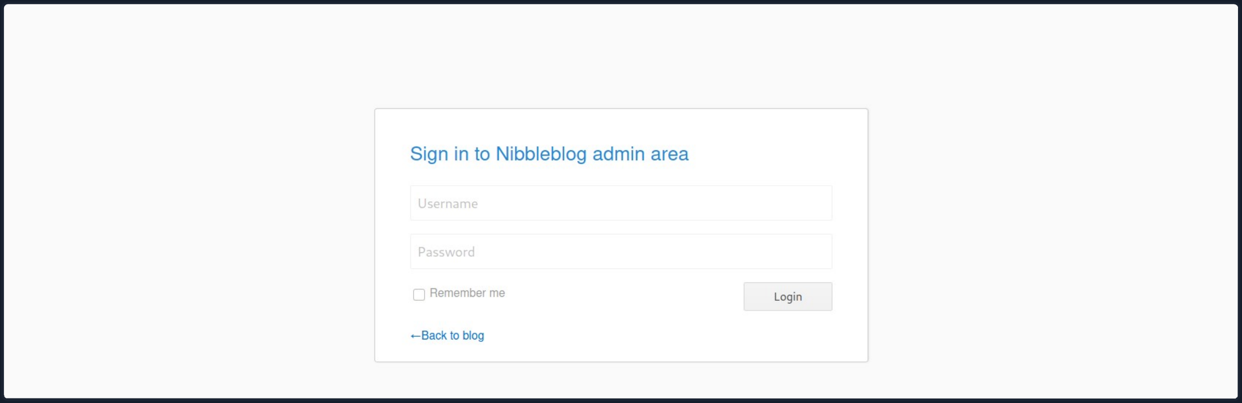
```
MichaelLuka@htb[/htb]$ gobuster dir -u http://10.129.42.190/nibbleblog/ --wordlist /usr/share/dirb/wordlists/c

===============================================================

Gobuster v3.0.1

by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================

[+] Url:            http://10.129.42.190/nibbleblog/
[+] 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
===============================================================
2020/12/17 00:10:47 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/admin (Status: 301)
/admin.php (Status: 200)
/content (Status: 301)
/index.php (Status: 200)
/languages (Status: 301)
/plugins (Status: 301)
/README (Status: 200)
/themes (Status: 301)
===============================================================
2020/12/17 00:11:38 Finished
```

```
===============================================================
```

Gobuster finishes very quickly and confirms the presence of the admin.php page. We can check the README page for interesting information, such as the version number.

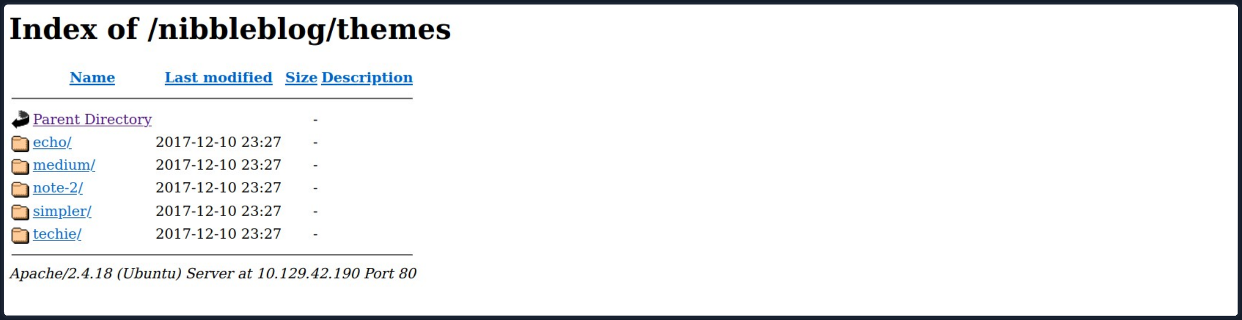
```
MichaelLuka@htb[/htb]$ curl http://10.129.42.190/nibbleblog/README

====== Nibbleblog ======
Version: v4.0.3
Codename: Coffee
Release date: 2014-04-01

Site: http://www.nibbleblog.com
Blog: http://blog.nibbleblog.com
Help & Support: http://forum.nibbleblog.com
Documentation: http://docs.nibbleblog.com

===== Social =====

* Twitter: http://twitter.com/nibbleblog
* Facebook: http://www.facebook.com/nibbleblog
* Google+: http://google.com/+nibbleblog

===== System Requirements =====

* PHP v5.2 or higher
* PHP module - DOM
* PHP module - SimpleXML
* PHP module - GD
* Directory “content” writable by Apache/PHP

<SNIP>
```

So we validate that version 4.0.3 is in use, confirming that this version is likely vulnerable to the Metasploit module (though this could be an old README page). Nothing else interesting pops out at us. Let us check out the admin portal login page.



Now, to use the exploit mentioned above, we will need valid admin credentials. We can try some authorization bypass techniques and common credential pairs manually, such as admin:admin and admin:password, to no avail. There is a reset password function, but we receive an e-mail error. Also, too many login attempts too quickly trigger a lockout with the message Nibbleblog security error - Blacklist protection.

Let us go back to our directory brute-forcing results. The 200 status codes show pages/directories that are directly accessible. The 403 status codes in the output indicate that access to these resources is forbidden. Finally, the 301 is a permanent redirect. Let us explore each of these. Browsing to nibbleblog/themes/. We can see that directory listing is enabled on the web application. Maybe we can find something interesting while poking around?


Index of /nibbleblog/themes

Name	Last modified	Size	Description
Parent Directory		-	
echo/	2017-12-10 23:27	-	
medium/	2017-12-10 23:27	-	
note-2/	2017-12-10 23:27	-	
simpler/	2017-12-10 23:27	-	
techie/	2017-12-10 23:27	-	

Apache/2.4.18 (Ubuntu) Server at 10.129.42.190 Port 80

Browsing to nibbleblog/content shows some interesting subdirectories public, private, and tmp. Digging around for a while, we find a users.xml file which at least seems to confirm the username is indeed admin. It also shows blacklisted IP addresses. We can request this file with cURL and prettify the XML output using xmllint.

```
MichaelLuka@htb[/htb]$ curl -s http://10.129.42.190/nibbleblog/content/private/users.xml | xmllint  --format -

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<users>
  <user username="admin">
    <id type="integer">0</id>
    <session_fail_count type="integer">2</session_fail_count>
    <session_date type="integer">1608182184</session_date>
  </user>
  <blacklist type="string" ip="10.10.10.1">
    <date type="integer">1512964659</date>
    <fail_count type="integer">1</fail_count>
  </blacklist>
  <blacklist type="string" ip="10.10.14.2">
    <date type="integer">1608182171</date>
    <fail_count type="integer">5</fail_count>
  </blacklist>
</users>
```

At this point, we have a valid username but no password. Searches of Nibbleblog related documentation show that the password is set during installation, and there is no known default password. Up to this point, have the following pieces of the puzzle:

- A Nibbleblog install potentially vulnerable to an authenticated file upload vulnerability
- An admin portal at nibbleblog/admin.php
- Directory listing which confirmed that admin is a valid username
- Login brute-forcing protection blacklists our IP address after too many invalid login attempts. This takes login brute-forcing with a tool such as Hydra off the table

There are no other ports open, and we did not find any other directories. Which we can confirm by performing additional directory brute-forcing against the root of the web application

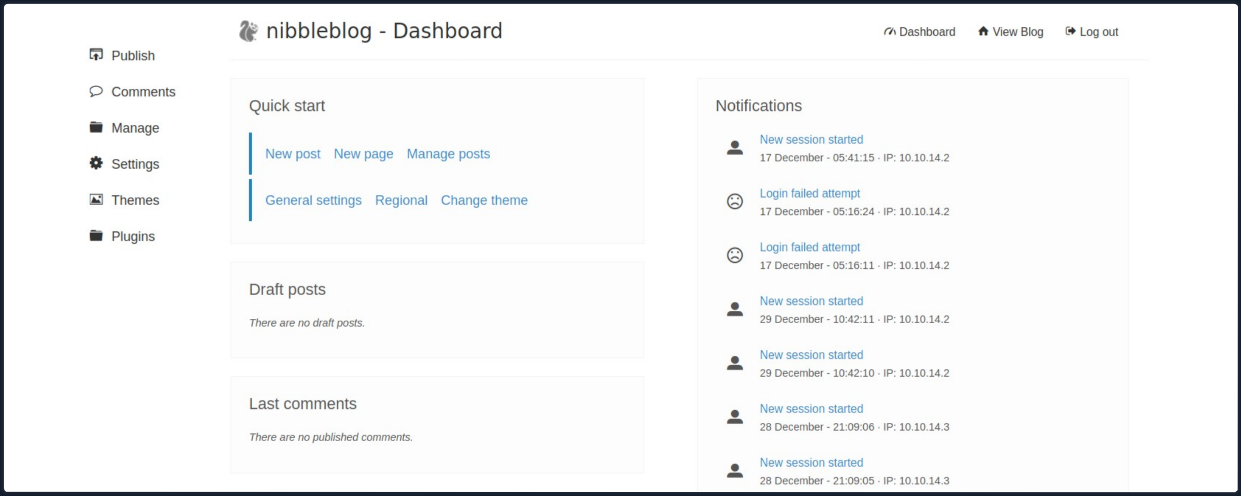
```
MichaelLuka@htb[/htb]$ gobuster dir -u http://10.129.42.190/ --wordlist /usr/share/dirb/wordlists/common.txt

===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.129.42.190/
[+] Threads:        10
[+] Wordlist:       /usr/share/dirb/wordlists/common.txt
[+] Wordlist:       /usr/share/dirb/wordlists/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/12/17 00:36:55 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/index.html (Status: 200)
/server-status (Status: 403)
===============================================================
2020/12/17 00:37:46 Finished
===============================================================
```

Taking another look through all of the exposed directories, we find a `config.xml` file.

```
MichaelLuka@htb[/htb]$ curl -s http://10.129.42.190/nibbleblog/content/private/config.xml | xmllint --format -

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<config>
  <name type="string">Nibbles</name>
  <slogan type="string">Yum yum</slogan>
  <footer type="string">Powered by Nibbleblog</footer>
  <advanced_post_options type="integer">0</advanced_post_options>
  <url type="string">http://10.129.42.190/nibbleblog/</url>
  <path type="string">/nibbleblog/</path>
  <items_rss type="integer">4</items_rss>
  <items_page type="integer">6</items_page>
  <language type="string">en_US</language>
  <timezone type="string">UTC</timezone>
  <timestamp_format type="string">%d %B, %Y</timestamp_format>
  <locale type="string">en_US</locale>
  <img_resize type="integer">1</img_resize>
  <img_resize_width type="integer">1000</img_resize_width>
  <img_resize_height type="integer">600</img_resize_height>
  <img_resize_quality type="integer">100</img_resize_quality>
  <img_resize_option type="string">auto</img_resize_option>
  <img_thumbnail type="integer">1</img_thumbnail>
  <img_thumbnail_width type="integer">190</img_thumbnail_width>
  <img_thumbnail_height type="integer">190</img_thumbnail_height>
  <img_thumbnail_quality type="integer">100</img_thumbnail_quality>
  <img_thumbnail_option type="string">landscape</img_thumbnail_option>
  <theme type="string">simpler</theme>
  <notification_comments type="integer">1</notification_comments>
  <notification_session_fail type="integer">0</notification_session_fail>
  <notification_session_start type="integer">0</notification_session_start>
  <notification_email_to type="string">admin@nibbles.com</notification_email_to>
  <notification_email_from type="string">noreply@10.10.10.134</notification_email_from>
  <seo_site_title type="string">Nibbles - Yum yum</seo_site_title>
  <seo_site_description type="string"/>
  <seo_keywords type="string"/>
  <seo_robots type="string"/>
  <seo_google_code type="string"/>
  <seo_bing_code type="string"/>
  <seo_author type="string"/>
  <friendly_urls type="integer">0</friendly_urls>
  <default_homepage type="integer">0</default_homepage>
</config>
```

Checking it, hoping for passwords proofs fruitless, but we do see two mentions of `nibbles` in the site title as well as the notification e-mail address. This is also the name of the box. Could this be the admin password?

When performing password cracking offline with a tool such as `Hashcat` or attempting to guess a password, it is important to consider all of the information in front of us. It is not uncommon to successfully crack a password hash (such as a company's wireless network passphrase) using a wordlist generated by crawling their website using a tool such as CeWL.



There are no draft posts.

There are no published comments.

New session started
17 December - 05:41:15 - IP: 10.10.14.2

Login failed attempt
17 December - 05:16:24 - IP: 10.10.14.2

Login failed attempt
17 December - 05:16:11 - IP: 10.10.14.2

New session started
29 December - 10:42:11 - IP: 10.10.14.2

New session started
29 December - 10:42:10 - IP: 10.10.14.2

New session started
28 December - 21:09:06 - IP: 10.10.14.3

New session started
28 December - 21:09:05 - IP: 10.10.14.3

This shows us how crucial thorough enumeration is. Let us recap what we have found so far:

- We started with a simple `nmap` scan showing two open ports
- Discovered an instance of `Nibbleblog`
- Analyzed the technologies in use using `whatweb`
- Found the admin login portal page at `admin.php`
- Discovered that directory listing is enabled and browsed several directories
- Confirmed that `admin` was the valid username
- Found out the hard way that IP blacklisting is enabled to prevent brute-force login attempts
- Uncovered clues that led us to a valid admin password of nibbles

This proves that we need a clear, repeatable process that we will use time and time again, no matter if we are attacking a single box on HTB, performing a web application penetration test for a client, or attacking a large Active Directory environment. Keep in mind that iterative enumeration, along with detailed notetaking, is one of the keys to success in this field. As you progress in your career, you will often marvel at how the initial scope of a penetration test seemed extremely small and "boring," yet once you dig in and perform rounds and rounds of enumeration and peel back the layers, you may find an exposed service on a high port or some forgotten page or directory that can lead to sensitive data exposure or even a foothold.

Start Instance

1 / 1 spawns left

Waiting to start...

Previous Next Mark Complete & Next

Cheat Sheet

Table of Contents

Problem Solving

Common Pitfalls

Getting Help

What's Next?

Next Steps

Knowledge Check

My Workstation

OFFLINE

Start Instance

1 / 1 spawns left