

HTB: Paper (User Flag)

Starting out with HackTheBox, I did not know where to begin, nor did I truly understand the dynamic of “hacking boxes.” After a peer showed me a starting point, pinging the IP of Paper and suggesting the use of Nmap, I was off to the races.

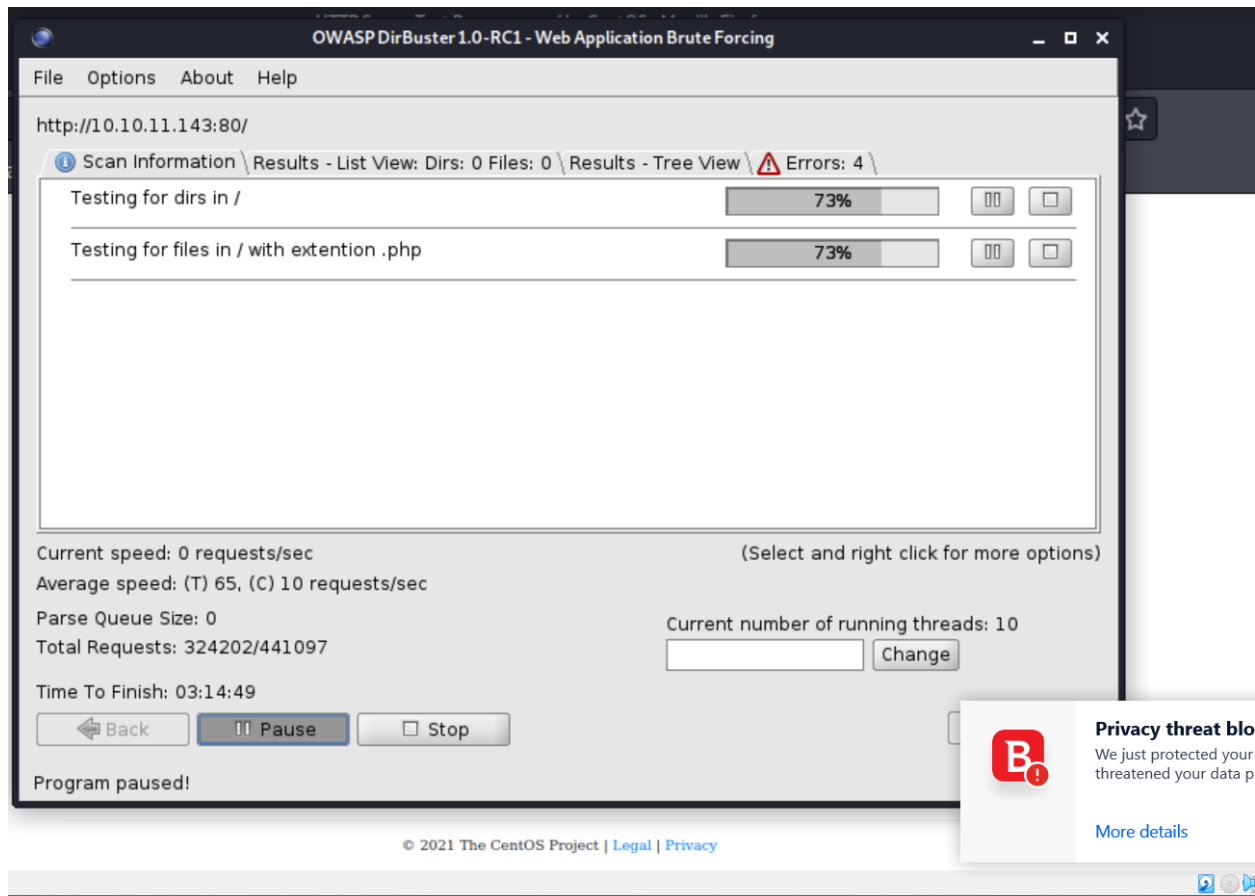
I realized that I was going to need lots of tools to work with the boxes, so I ran a Kali VM on my desktop to interact with Paper.

I first scanned 10.10.11.143 with Nmap, using flags -A (for OS detection, although had I just looked at the info card, I could have seen that Paper was a Linux box) and -T4 for faster execution of the process. The initial port scanning was to gather whatever info from the outside of the box so that I could start plotting my way in.

```
File Actions Edit View Help
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-22 18:58 EST
Nmap scan report for 10.10.11.143
Host is up (0.029s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.0 (protocol 2.0)
|_ ssh-hostkey:
|   2048 10:05:ea:50:56:a6:00:cb:1c:9c:93:df:5f:83:e0:64 (RSA)
|   256 58:8c:82:1c:c6:63:2a:83:87:5c:2f:2b:4f:4d:c3:79 (ECDSA)
|_  256 31:78:af:d1:3b:c4:2e:9d:60:4e:eb:5d:03:ec:a0:22 (ED25519)
80/tcp    open  http     Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_ http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
|_ http-title: HTTP Server Test Page powered by CentOS
443/tcp   open  ssl/http Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_ http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
|_ http-title: HTTP Server Test Page powered by CentOS
|_ ssl-cert: Subject: commonName=localhost.localdomain/organizationName=Unspecified/countryName=US
|_   Subject Alternative Name: DNS:localhost.localdomain
|_   Not valid before: 2021-07-03T08:52:34
|_   Not valid after:  2022-07-08T10:32:34
|_ ssl-date: TLS randomness does not represent time
|_ tls-alpn:
|_   http/1.1
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.51 seconds
kali@CyberRange:~$
```

From Nmap, I discovered three open ports: port 22 (ssh), port 80 (http) and port 443 (https). I figured my best bet was going to be with port 80, as http seemed like the easiest to find a way in through.

From here, I tried using DirBuster to see if I could find any directories I could exploit. As soon as I started the scan, my BitDefender (yes I know, I have an Antivirus) started flipping out, clouding my screen with notifications.



Despite the notifications, I let DirBuster run, but found nothing I could use to move forward with. I read up on the comments of the box, and a few people suggested using a tool called Nikto to make some progress.

After researching how to use Nikto, I ran it against the IP address of the box.

```

kali@CyberRange:~$ nikto -h 10.10.11.143
- Nikto v2.1.6

+ Target IP: 10.10.11.143
+ Target Hostname: 10.10.11.143
+ Target Port: 80
+ Start Time: 2022-02-23 21:50:25 (GMT-5)

+ Server: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
+ 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
+ Uncommon header 'x-backend-server' found, with contents: office.paper
+ 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 fashion to the MIME type
+ Retrieved x-powered-by header: PHP/7.2.24
+ Allowed HTTP Methods: OPTIONS, HEAD, GET, POST, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ ERROR: Error limit (20) reached for host, giving up. Last error:
+ Scan terminated: 0 error(s) and 7 item(s) reported on remote host
+ End Time: 2022-02-23 21:52:50 (GMT-5) (145 seconds)

+ 1 host(s) tested

```

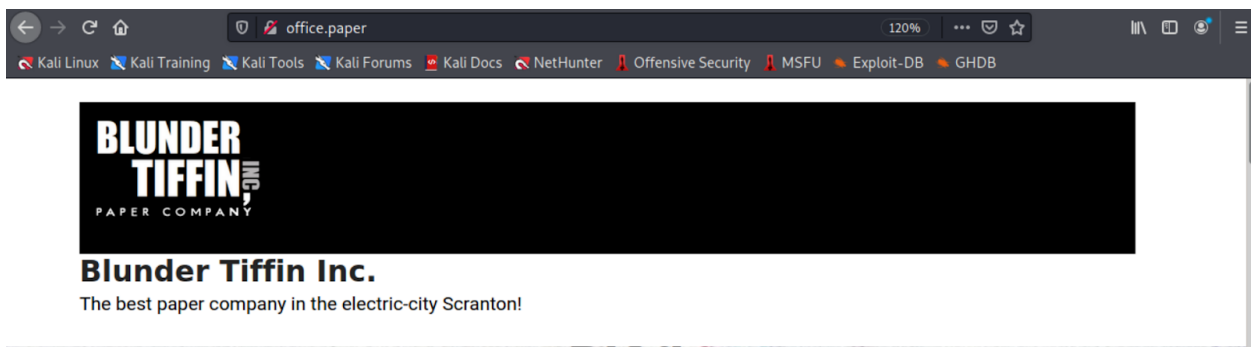
With Nikto, I discovered a back-end server called `office.paper`. Now at first I didn't know what to do with this information, however, I knew I had a new lead. I put `office.paper` in my browser but couldn't load the webpage. Stumped yet again, I turned to the internet and found that I may want to try adding it to my `/etc/hosts` file before I gave up on that lead.

I added it to `/etc/hosts` but I still could not access the webpage. I was sure that I was supposed to, so again, cue the internet.

`10.10.11.143 office.paper`

After some more digging, I found that flushing my DNS may work in updating the new host rule. So I found a command to flush my DNS, `sudo systemd-resolve --flush-caches`, however, I got a big red error on my screen: **Failed to flush caches: Unit dbusorg.freedesktop.resolve1.service not found**. After even more digging to figure out how to *flush my DNS* I came across another command to enable the thing that was prompting an error: `sudo systemctl enable systemd-resolved.service`. After running this command, I was successfully able to flush my DNS and was able to visit the webpage.

On the webpage, I was presented with this:



It was a WordPress site. I ended up running WPScan to see what info it could gather.

```
[+] WordPress version 5.2.3 identified (Insecure, released on 2019-09-05) .  
  | Found By: Rss Generator (Passive Detection)  
  | - http://office.paper/index.php/feed/,  
<generator>https://wordpress.org/?v=5.2.3</generator>  
  | - http://office.paper/index.php/comments/feed/,  
<generator>https://wordpress.org/?v=5.2.3</generator>
```

With the version number, I started digging for vulnerabilities for the website.



WordPress < 5.4.1 - Authenticated Cross-Site Scripting (XSS) in File Uploads

✓ Fixed in version 5.3.3

2020-04-29

WordPress < 5.4.1 - Cross-Site Scripting (XSS) in wp-object-cache

✓ Fixed in version 5.3.3

2020-04-29

WordPress < 5.4.1 - Authenticated Cross-Site Scripting (XSS) in Search Block

✓ Fixed in version 5.3.3

2020-04-29

WordPress < 5.4.1 - Authenticated Cross-Site Scripting (XSS) in Customizer

✓ Fixed in version 5.3.3

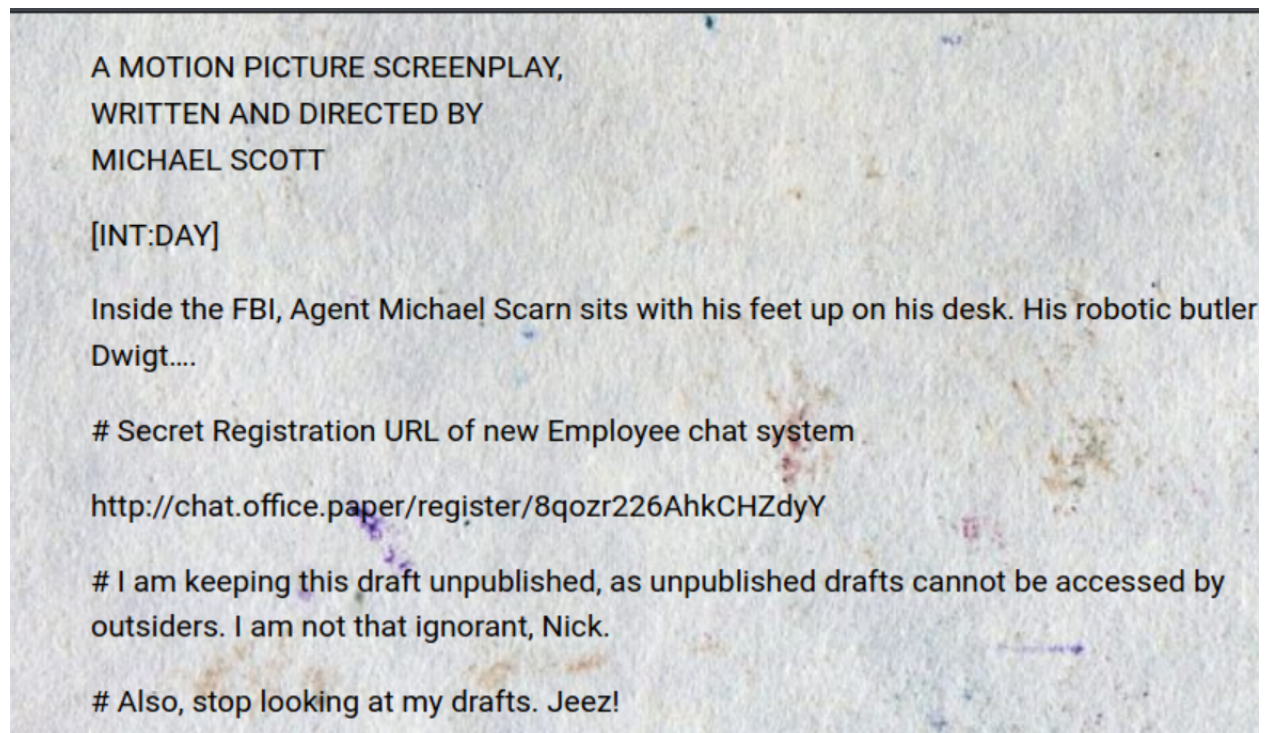
2020-04-29

WordPress < 5.4.1 - Unauthenticated Users View Private Posts

✓ Fixed in version 5.3.3

I explored a couple of these and ended up trying the Unauthenticated Users View Private Posts bug with this version of WordPress. I appended `/?static=1` to the URL and I was

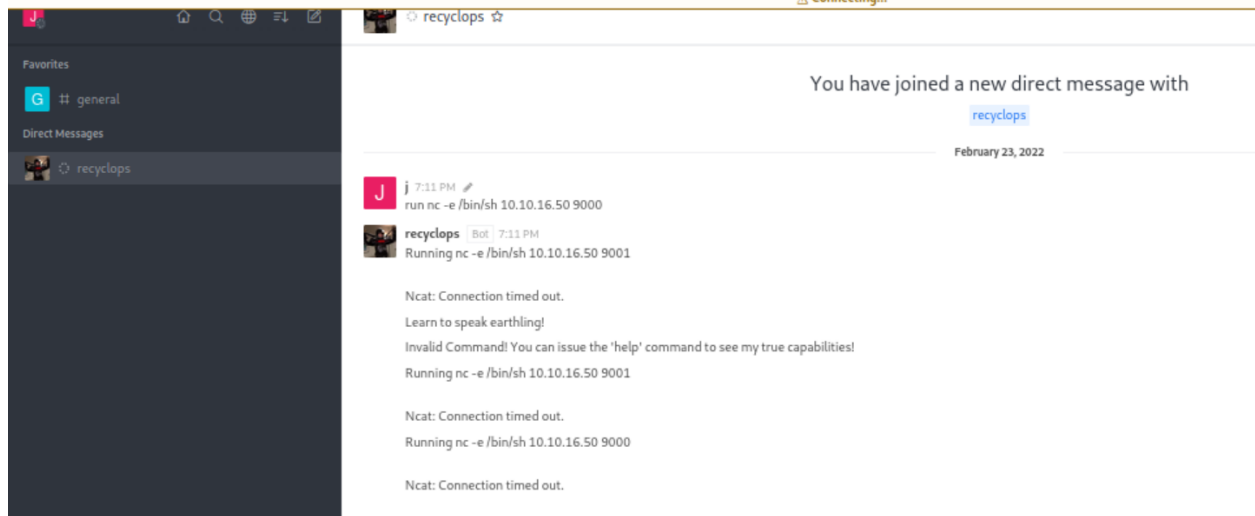
brought to a page that I “wasn’t supposed to see.” Scrolling down on the page, I found another link to follow.



When I pasted it in the URL bar, nothing happened. I figured I had to add this webpage to my hosts list. After adding it and flushing my DNS once more, I was presented with... Invalid page. The page showed a rocket logo, so I knew I managed to get through to the server, however, it was bugging out on me. After many refreshes, flushing my DNS yet again and clicking around, it magically took me to a registration page.

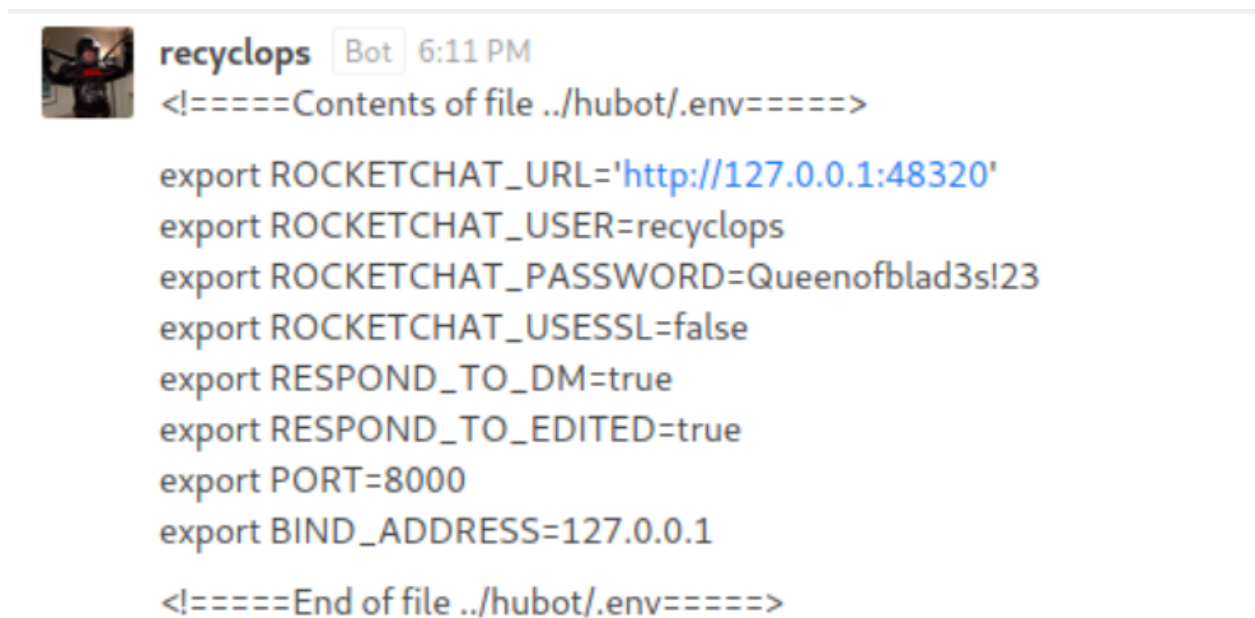
I registered with the client with bogus information and logged in. I was brought to a Slack-like The Office-themed chat service. The #general channel was read-only, but I could message other people, including the bot on the server, recyclops.

I found out that I could use Linux commands with the bot. I tried to open a reverse shell (as I had never done one before) but I was unsuccessful in my attempt.

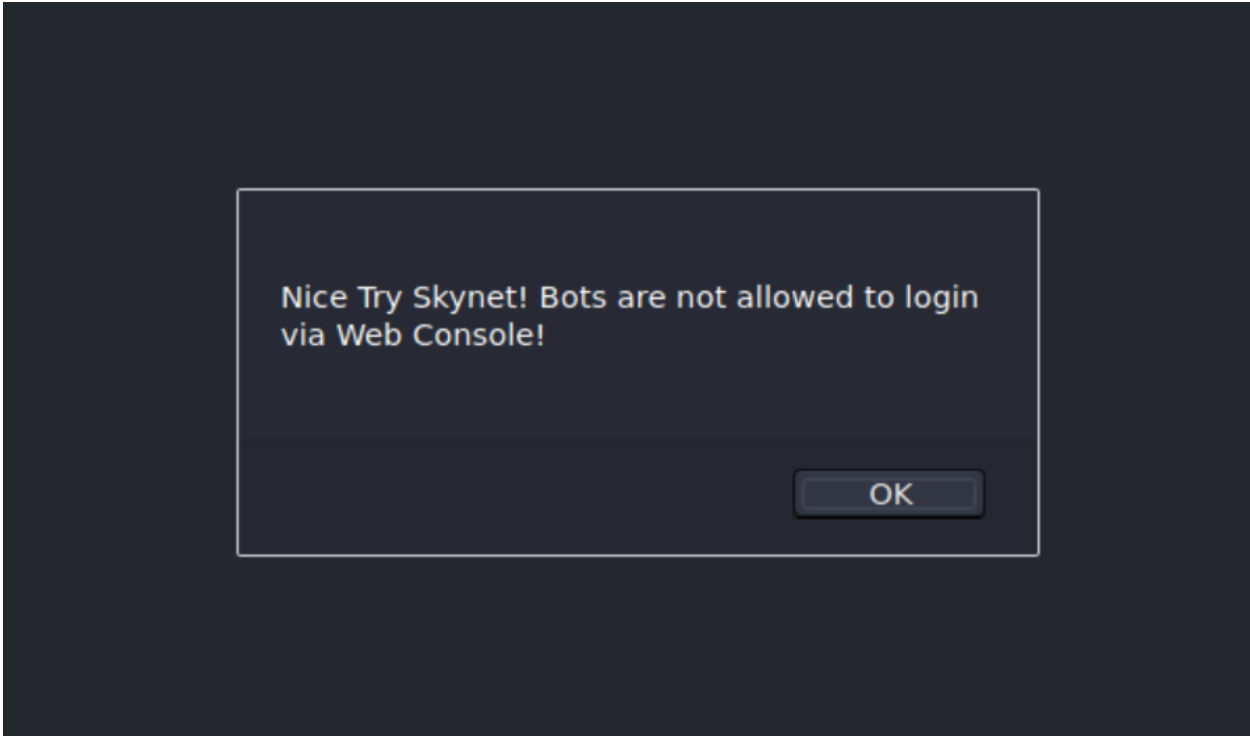


I later found out that this was a valid method of exploiting the bot, however, the VPN was not running inside of my virtual machine which apparently made Ncat unable to connect properly.

My backup was to try to get to the other files on the machine through the chatbot text. After more than an hour of exploring, I found a file by using the command `recyclops file ../hubot/.env` that contained some very juicy information.



Of course, I didn't know what to do with this at first. I tried logging into the chat site with the bot's information, and surprisingly, it let me after flashing a message on the screen.



Nice Try Skynet! Bots are not allowed to login
via Web Console!

OK

I realized I was not going to be able to do much from the chat site, so I tried SSHing into the box using the robot's credentials, to no avail. Little did I know that the answer was in my face: I could try logging in as Dwight (the owner of the bot who *probably* reused his password for the bot elsewhere...).

The VPN service went down for the remainder of the day, but as soon as I could access the box again, I was able to log in as Dwight.

```
kali@CyberRange:~$ ssh dwight@10.10.11.143
dwight@10.10.11.143's password:
Permission denied, please try again.
dwight@10.10.11.143's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last failed login: Thu Feb 24 15:21:45 EST 2022 from 10.10.14.235 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu Feb 24 13:59:57 2022 from 10.10.16.24
```


And as Dwight, I was able to access user . txt. Flag obtained!

```
[dwight@paper ~]$ ls
bot_restart.sh  exp.py  hubot  sales  user.txt
[dwight@paper ~]$ cat user.txt
5217cc0a7ef2178857226b67c75f0d00
[dwight@paper ~]$
```