Year of the Jellyfish (YotJF)

> Bradley, rnbochsr | 4/23/2021 - 4/30/2021

Target configuration info

- Website IP changes each time the machine spins up.
- URL: `https://robyns-petshop.thm`
- Website Platform: AmazonAWS. *Version #?*
- Web server: Apache 2.4.29 Ubuntu.
- Site runs using PicoCMS. *Version #?*
- OpenSSH 5.9p1 and 7.6p1.
- OpenSSL 1.1.1k
- vsFTPd 3.0.3

Initial Focus & Insights

As I began the challenge, I focused on the things that showed up in the `nmap` scan. Having found 7 ports, I felt confident that they would bear fruit and that was where I began. I just fell down the `ftp` rabbit hole. Hopeful, I tried a lot of variations to make the `ftp` and `sftp` interface my foothold. But, I couldn't figure out a way in.

My next plan was the port 8000 Costruction ID page. I tried lots of the same IDs. Robyn, admin, staff, pet types, pet names, admin, etc. None of my atempts were successful.

Port 8096 lands you on the Jellyfin login page. I had the same lack of success in trying to get into that portal.

I looked at software versions and tried to find CVEs that might work. Again drawing from my `nmap` scan, I looked at vsFTPd, PicoCMS, and other items noted above. Admittedly, as this was my first attempt at this type of challenge, I found myself dealing with information overload. Too many choices, and not knowing how best to evaluate it all.

Unable to determine the path that would yield the best chance of success, I found myself floundering. I was making no real headway in determining if an option was a rabbit hole, or if I simply hadn't yet gained the knowledge or tools to exploit it and hack into the box. I continued to plug along.

MuirlandOracle was right. Some boxes do sting.

Things to try

- `nmap` all ports - Done. Found 7 open ports. See `nmap/initial`.
- Figure out how to install the website's certificate from the nmap scan into Firefox to prevent the error and let me load the web page.
- Modify `/etc/hosts` file to add the machine IP so the DNS will resolve and display the website.
- `wget` website to get code - Initial results don't show much.
- `curl` website to get code - Initial results don't show much.
- `ftp` & `sftp` - In process. Try to find user IDs and passwords. Try:
  - Basic default ID:password combos
  - Robyn
  - Pet names & type (rabbit, Guinea pig, etc.)
  - Metasploit to look-up CVEs. They didn't work, or at least I couldn't make them work.
- Jellyfin login page. Done.
- Kestrel - Microsoft Web Server. But the website is running on an Apache Server??
- BurpSuite to look for cookies.
- Metasploit and Searchsploit for CVE analysis. I guess that Searchsploit looks up the software without launching Metasploit. Also can search directly from the CVE.Mitre.org or Exploit-DB.com websites.
- MD5 & SHA-1 hashes in `nmap` scan. Crackable??
- Users: Robyn MacKenzie `robyn@robyns-petshop.thm` `admin@robyns-petshop.thm` `staff@robyns-petshop.thm`
- There is a `Contact Us` page. See if and command injection might work there.
- AmazonAWS exploits - Lots of reading about this. I am completely unfamiliar with it.
- Email server - I can't find an email server running on the site. There has to be since the `Contact Us` page lists `staff@robyns-petshop.thm` as a data point. But that could just be a rabbit hole.
- Dirbuster - In process. See dirbuster_output.md for listing and notes.
- Gobuster `-k` flag? `gobuster -k dir -u http://IP -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt`.
- Hydra

- Nitko
- Use various `password attack` options in Kali
- Don't forget OSINT

Web Folders

assets
assets/pets
themes
themes/default/js

Domains

I had information on several other domains in my initial `nmap` scan. I didn't put any real emphasis on them. They were on my list of things to try, but not priorities. I didn't look into it until I saw the suggestion as a hint.

dev.robyns-petshop.thm

- This sounded promising. A development site would hopefully have some administrative credentials buried in it somewhere.
- All it does is go to the main website.

beta.robyns-petshop.thm

- Goes to the port 8000 Under Construction page.
- beta.robyns-petshop.thm/ID_HERE

monitorr.robyns-petshop.thm

- https://monitorr.robyns-petshop.thm
  Settings on page point to *LOCAL* directories
  - User database dir: /var/www/monitorr/data
  - User database file: /var/www/monitorr/datausers.db

Exploit CVE List

Apache 2.4.29

- CVE-2018-1312 Apache httpd 2.2.0 - 2.4.29
- CVE-2018-1283 Apache httpd 2.4.0 - 2.4.29
- CVE-2017-15715 Apache httpd 2.4.0 - 2.4.29
- CVE-2017-15710 Apache httpd 2.4.0 - 2.4.29

PicoCMS

- http://picocms.org/
- https://github.com/picocms
- CVE-2008-6604 PicoFlat CMS 0.5.9
- CVE-2007-5920 PicoFlat CMS pre 0.4.18
- CVE-2007-5390 PicoFlat CMS 0.4.14 and earlier

Amazon AWS lists 28 CVEs.

OpenSSH 5.9p1 or 7.6p1

- 1 Exploit listed but it didn't bypass username:password.
- 1 Enumeration Scanner but it didn't bypass username:password.

OpenSSL 1.1.1k

- CVE-2021-3450 OpenSSL 1.1.1k
- CVE-2021-3449 OpenSSL 1.1.1k

Jellyfin - No CVEs listed.

Results that worked

- `nmap` of all ports found 7 open ports.
- Modifying the `/etc/hosts` file allowed the website to load! Remember this for future pentests.
- `ftp` didn't connect in a useful manner from my IP. It would connect but then I couldn't pass anything.
- `sftp` connected and was *interactive!!* Now try to use some password cracking for `robyn`, `admin`, & `anonymous`.
- `ftp` on the THM Kali attack box connects without any issues. It allowed me to interact. Now to see if I can brute the password.
- `http://robyns-petshop.thm:8000` gets you to a page that says enter your ID `...:8000/ID_Here`. `robyn`, `admin`, & `staff` don't work. Look for other options.
- `http://robyns-petshop.thm:8096` is a login page for Jellyfin. It seems to be a media player and control panel. I'll try to manual brute forcing logins.
- `https://robyns-petshop.thm/business` is a login page. It is asking for `Business Credentials Please`. User Name/password.
- Dirbuster produced a good website directory tree. I had trouble exporting it from the Kali attackbox, so I transcribed some notes into `dirbuster_output.md`.

Notes and possible additional resources

- `https://github.com/wwong99/pentest-notes/blob/master/oscp_resources/OSCP-Survival-Guide.md#enumeration`
- `https://github.com/theonlykernel/enumeration/wiki`
- `https://github.com/theonlykernel/EasyEnumeration/wiki`
- Possible new 0day exploit 4/25/21. I read it as LATimes, but it is LATlmes. Use caution if I pursue this. Might be a malicious website. `https://www.latlmes.com/tech/the-ultimate-0day-1` <= Rickroll!
- Honeypot `https://github.com/cowrie/cowrie`. Someone posted this link, but why would you need a honeypot when trying to hack into a box? Not trying to trip up others.

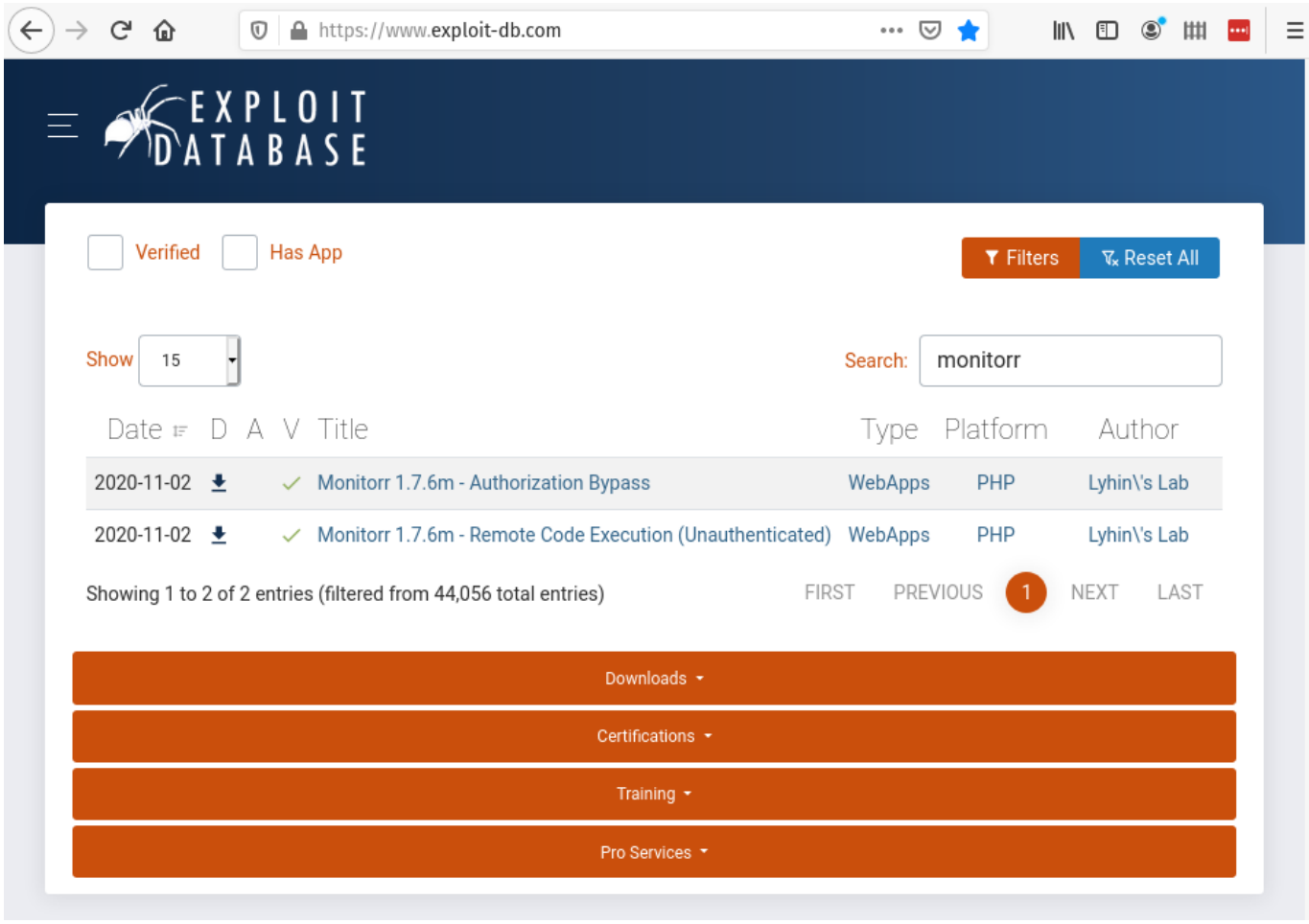
Things that didn't work

Website won't load. Domain doesn't resolve until `/etc/hosts` updated.
Initially didn't get anything useful from `wget` or `curl` of website.
My first real use of Hashcat. It couldn't crack the MD5 from the `nmap` scan. That was disappointing. Hoped it was a site or user specific password.
I didn't have any luck with Hydra on the login pages.
Jellyfin login page led nowhere.
Business login page led nowhere.
`ftp` and `sftp` led nowhere.

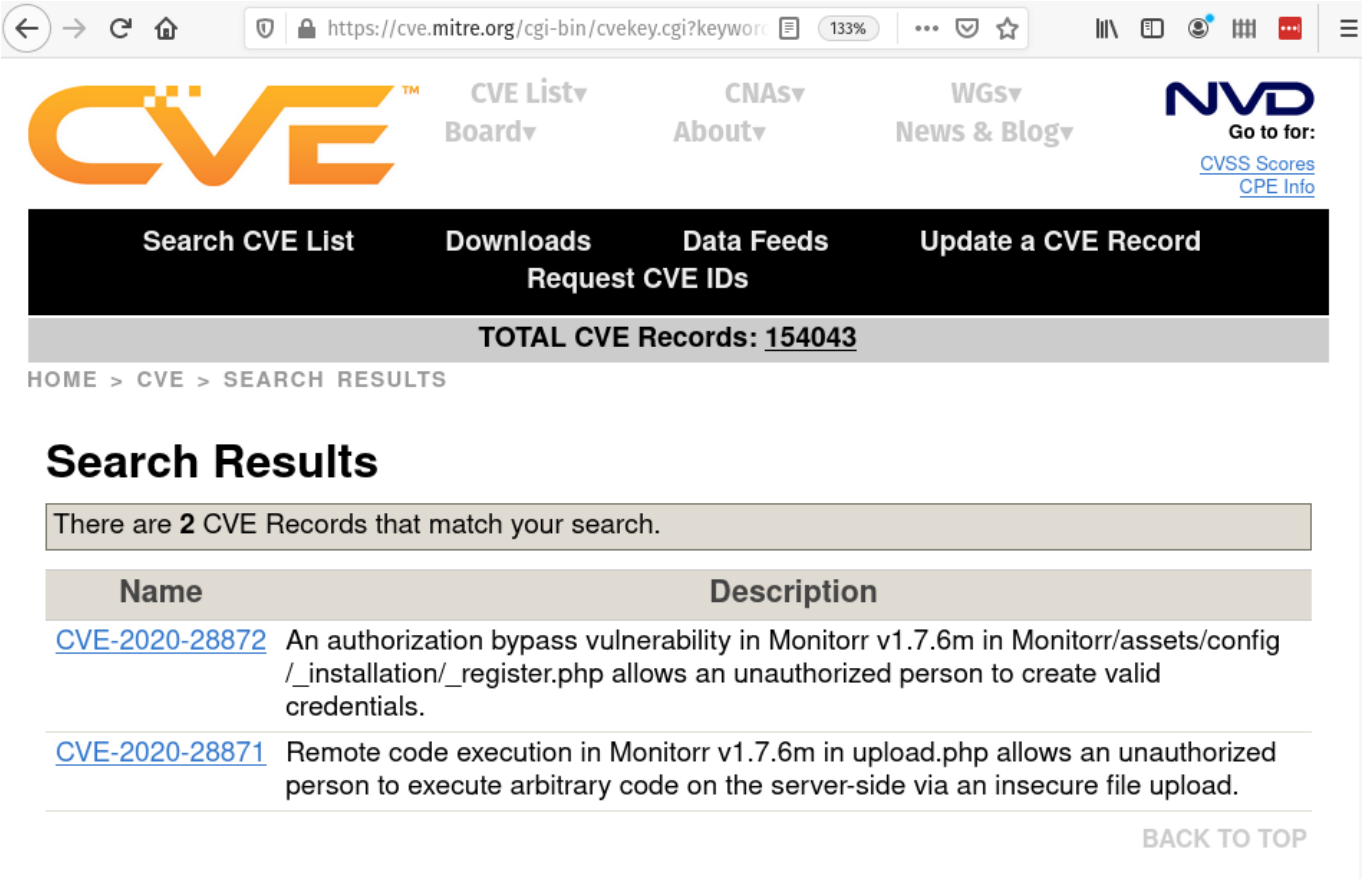
A Hint and The Solution

After the challenge was over I finally asked for a hint from Muirland. He said that I missed a CVE on Monitorr. I didn't know how because I had searched for it. I know you couldn't use metasploit to run exploits, but I had used it to search and came up with nothing. I searched again, and nothing. I went to try the exploit-db.com site directly. There it was. A

remote code execution vulnerability.



I checked the CVE.Mitre.org site. It was there too.



Apparently my version of metasploit had an out of date database. Serves me right for not ensuring everything was up to date before I started.

I downloaded the CVE info and exploit file. The exploit abuses an upload.php file fault. I'm familiar with PHP, but not very good at programming in it, so a little reading is the plan.

Giving it a try out of the box I gave it my IP and port, but it didn't work. The script isn't uploading. I found a method to print responses as they come in from the server. I added a variable to the post request and a `print(r.text)` to give me some feedback that I can hopefully use. The server won't upload php files. And it is throwing errors relating to the certificate. A little more reading and discovered a `verify=False` flag that ignores certificate errors. It also may help to the sessions rather than just post.

A `.gif` uploads but nothing happened. The browser won't open it. `.jpg` same thing. I need to try stacking extensions. And it won't overwrite files either. Need to rename the file with each attempt. Note the file name is in 2 spots in the script so be certain to match the names or it won't work.

Anything with `.php` in it won't upload. `.jpg.php .gif.php .png.php` are all not working. I need to look for another file extension that will both bypass the upload filter and execute once on the server. I'm also adding a cookie value as the server is not accepting the file

without one. It says "You are an exploit." The browser tools show the cookie value as `"isHuman" = "1"` so I am adding that to the session post and get requests.

Found an extension, `.phtml` but while it uploads, I am not getting a shell back. Tried a few different ports with no luck. More reading.

Ok, I'm learning a lot with this one and at the same time I'm annoyed with myself. The target is on the internet, and not the internal THM virtual environment. So I gave the script my WAN ISP IP. I wasn't getting a reverse shell. All my reading said this should work. And Muirland said this was what I missed. I tired it on the attack box in case I was having the same issues I had with `FTP`. I gave the script the attack box's IP. It still didn't work. I'm going to have to read some more.

As I relaunched the machine, I saw the same notice I'd been seeing for weeks. The challenge kept putting up the `You have to use OpenVPN or the attack box` and although I didn't think it would work, I put my THM OpenVPN IP into the script and ran through all the variations and ports again. It worked! It had to be on port 443 and I don't know why a 10.x.x.x IP works, but I'm not going to argue.

Checking the contents of each directory as I move up the tree. I find the flag in `/var/www`. Nice. Now to figure out how to escalate my privileges.

Poking around I see only 1 other user, `robyn` and there isn't anything in her home directory. Lots of stuff isn't available to the www-data user. More reading.

Using Exploit-DB.com to check for CVEs this time, I listed all the software I could see then ran searches. Finally I landed on snapd. It had a vulnerability to `dirty_sock`. I downloaded the CVE and the exploit file. Now the reading resumes.

It seems there are 2 versions. Version #1 is asking for an account on Ubuntu and a user ID:password combo. I don't have any of that yet. Version #2 just runs on the local server without the Ubuntu account or id:password combo. I'll try that one.

I started a local PHP HTTP server and sent the files to the target. Ran the file with Python and it ran! No errors, but I can't change users. More reading.

As it turns out, stabilizing a shell does more than let you use tab-complete and arrow keys. It also allows certain commands to work like `su`. I didn't know that. I hadn't stabilized the shell properly before so even though the exploit ran, the `su` command was blocked. I'll try again and this time fully stabilize the shell.

Once I stabilized the shell I'm successfully able to:

```
su dirty_sock
enter the password: dirty_sock
```

`id` shows I'm in the `sudo` group. I still don't have permission to `cd /root`. A little more poking around and reading leads me to:

```
sudo bash
enter the password: dirty_sock
cd /root
```

Now I'm in the root user directory and get the flag.

Summary & Lessons Learned

It was a long road for me, but I learned a lot. Still pretty basic stuff in the pentester world I'm sure, but for me it was a fun journey.

- Update my tools before I begin.
- Checking subdomains for CVEs.
- Using CVE.Mitre.org, the Exploit-DB.com website, or Searchsploit to search for CVEs.
- New way to have my file extensions hide their true nature.
- And a new toy, Dirty_sock.

Yes, all in all, a fun ride.