Commands for Non AD

Host Discovery

```
# ping
sudo nmap -PE -sn -iL ranges.txt -oA ping
# tcp (discovery)
nmap -n -Pn -T4 --min-hostgroup 128 --max-retries 0 --top-por
ts 50 -iL ranges.txt -oA tcp-discovery
# udp (discovery)
sudo nmap -n -Pn --min-hostgroup 128 -sU -p 53,69,111,123,16
1,514,1900 -iL ranges.txt -oA udp
```

```
# get live hosts
grep Up ping.gnmap | cut -d ' ' -f 2 > live-ping.txt
grep 'open/' tcp-discovery.gnmap | cut -d ' ' -f 2 > live-tc
p.txt
grep 'open/' udp.gnmap | cut -d ' ' -f 2 > live-udp.txt
sort -uV live-ping.txt live-tcp.txt live-udp.txt > live.txt

# sample live hosts for further testing
shuf -n <num> live.txt | sort -uV > targets.txt
```

Service Scan

```
sudo nmap -n -Pn --min-hostgroup 128 --max-retries 0 -p- -sV
-O -iL targets.txt -oA tcp-full
```

```
# export to html for viewing in web browser
xsltproc ping.xml -o ping.html
xsltproc tcp-discovery.xml -o tcp-discovery.html
```

```
xsltproc udp.xml -o udp.html
xsltproc tcp-full.xml -o tcp-full.html
```

Outdated Versions

Find

```
nmap -Pn -p <ports> -sV --script vulners <target>
# manual review of tcp-full.html
# vulnerability scans (eg. Nessus)
```

Verify

```
curl -sik http(s)://<target>
nmap -Pn -p <port> -sV <target>
nmap -Pn -p <port> --script vmware-version <target>
```

Services

Terminal Access

SSH (22)

- default password
- weak password
- weak encryption
- password authentication

```
grep 22/open//ssh// tcp-full.gnmap | cut -d ' ' -f 2 > ssh.tx
t

# manually review for SSH on non-default ports with
grep -P ' (?!22)\d+/open//ssh//' tcp-full.gnmap
```

Default Password

1. get device type - check other accessible services, especially web

2. search default password on ChatGPT or Google
3. `sshpass -p <password> ssh <user>@<target>`

Weak Password

```
echo -en "root\nadmin" > usernames.txt
echo -en "\nroot\nadmin\npassword" > passwords.txt
hydra -M ssh.txt -L usernames.txt -P passwords.txt ssh -t 4
nxc ssh ssh.txt -u usernames.txt -p passwords.txt -t 4
```

Weak Encryption

```
nmap -Pn -p 22 --script ssh2-enum-algos <target>
ssh-audit -l warn -p 22 <target>
```

Password Authentication

```
nmap -Pn -p 22 --script ssh-auth-methods <target>
```

Telnet (23)

- default/no password
- weak password
- insecure protocol

```
grep 23/open//telnet// tcp-full.gnmap | cut -d ' ' -f 2 > tel
net.txt
cp telnet.txt temp.txt

# manually review for Telnet on non-default ports with
grep -P ' (?!23)\d+/open//telnet//' tcp-full.gnmap
```

Default/No Password

1. `for i in $(cat temp.txt); do telnet $i; done`

2. get device type - observe banner or check other accessible services, especially web
3. search default password on ChatGPT or Google
4. enter default credentials
5. on failure, `Ctrl+]` followed by `quit` If the service hangs, `Ctrl+C`, edit temp.txt, remove all hosts up to and including the offending host.

Weak Password

```
echo -en "root\nadmin" > usernames.txt
echo -en "\nroot\nadmin\npassword" > passwords.txt
hydra -M telnet.txt -L usernames.txt -P passwords.txt telnet
-t 4 # ***** verify this
```

Insecure Protocol

```
sed 's/$/ (tcp/23)/' telnet.txt
# copy-and-paste to affected hosts
```

File Sharing

FTP (21)

- anonymous login
- default password
- weak password
- insecure protocol

```
grep 21/open//ftp// tcp-full.gnmap | cut -d ' ' -f 2 > ftp.tx
t

# manually review for FTP on non-default ports with
grep -P ' (?!21)\d+/open//ftp//' tcp-full.gnmap
```

Anonymous Login

```
nmap -n -Pn -p 21 --script ftp-anon -iL ftp.txt -oA ftp-anon
grep allowed ftp-anon.nmap -B 6 | grep report | cut -d ' ' -f
5 > ftp-anon.txt
sed 's/$/ (tcp/21)/' ftp-anon.txt
```

Default Password

I don't check for default passwords on FTP because this is usually covered by other services, especially web (HTTP).

Weak Password

```
echo -en "root\nadmin" > usernames.txt
echo -en "\nroot\nadmin\npassword" > passwords.txt
hydra -M ftp.txt -L usernames.txt -P passwords.txt ftp -t 4
nxc ftp ftp.txt -u usernames.txt -p passwords.txt -t 4
```

Insecure Protocol

```
sed 's/$/ (tcp/21)/' ftp.txt
# copy-and-paste to affected hosts
```

SMB (139,445)

- null session (blank username, blank password)
- guest session (username Guest, no password)
- default password - only special cases like printers or IoT devices
- weak password - tied to Active Directory on domain-joined machines
- known CVEs
- insecure protocol (SMBv1)
- signing not required
- excessive share or NTFS ACLs

```
# from network scan
grep 445/open//smb// tcp-full.gnmap | cut -d ' ' -f 2 > smb.t
xt
# manually review for SMB on non-default ports with
grep -P ' (?!445)\d+/open//smb//' tcp-full.gnmap

# from Active Directory (optional)
mkdir ldapdomaindump
ldapdomaindump -u '<domain>\<user>' -p <password> <dc> -o lda
pdomaindump
jq -r '.[].attributes.dNSHostName[0]' ldapdomaindump/domain_c
omputers.json | grep -v null | sort -u > computers.txt
nmap -n -Pn - *****
sort -u smb.txt computers.txt |
```

```
# get domain controllers
dnsrecon -d <domain> -t SRV | egrep -v 'Found$|\*' | awk '{pr
int $4 ":" $5}' | sort -u > dcs.txt
cut -d ':' -f 1 | sort -u > dcs-dns.txt
cut -d ':' -f 2 | sort -uV > dcs-ip.txt
```

Null Session

```
# list shares
nxc smb smb.txt -u '' -p '' --shares

# enumerate domain accounts
nxc smb dcs-ip.txt -u '' -p '' --rid-brute
for i in $(cat dcs-ip.txt); do rpcclient -U '' -N $i -c enumd
omusers; done
for i in $(cat dcs-ip.txt); do rpcclient -U '' -N $i -c 'quer
yuser 500'; done
```

Guest Session

```
# list shares
nxc smb smb.txt -u Guest -p '' --shares

# enumerate domain accounts
nxc smb dcs-ip.txt -u Guest -p '' --rid-brute
for i in $(cat dcs-ip.txt); do rpcclient -U Guest -N $i -c en
umdomusers; done
for i in $(cat dcs-ip.txt); do rpcclient -U Guest -N $i -c 'q
ueryuser 500'; done
```

Default Password

I don't check for default passwords on SMB because this only applies in niche cases and is usually covered by other services, especially web (HTTP).

Weak Password

```
echo -en "root\nadmin" > usernames.txt
echo -en "\nroot\nadmin\npassword" > passwords.txt
hydra -M smb.txt -L usernames.txt -P passwords.txt smb -t 4
nxc smb smb.txt -u usernames.txt -p passwords.txt -t 4
```

Excessive Share/NTFS ACLs

```
# password
runas /netonly /user:<domain>\<user> powershell.exe
# NT hash
.\Rubeus asktgt /domain:<domain> /dc:<dc> /user:<user> /rc4:<
NT hash> /ptt
# Kerberos ticket
.\Rubeus ptt /ticket:<ticket>

# search share contents
.\Snaffler.exe -d <domain> -v data -s -o snaffler.txt -y
.\snafflerParser.ps1 -in .\snaffler.txt
# review in browser, sorting by date
```

- Snaffler: https://github.com/SnaffCon/Snaffler
- SnafflerParser: https://github.com/zh54321/SnafflerParser

```
Get-Acl <share file> # this checks NTFS ACLs
Get-Content <share file>
```

In File Explorer, navigate to `<share directory>` to view files: `\\computer.domain.com\share\file`

NFS (2049)

- share enumeration
- publicly mountable shares

```
grep 2049/open//nfs// tcp-full.gnmap | cut -d ' ' -f 2 > nfs.
txt
cp telnet.txt temp.txt

# manually review for NFS on non-default ports with
grep -P ' (?!2049)\d+/open//nfs//' tcp-full.gnmap
```

Share Enumeration

```
nmap -Pn -p 2049 --script nfs-ls -iL nfs.txt -oA nfs-ls
nmap -Pn -p 2049 --script nfs-showmount -iL nfs.txt -oA nfs-s
howmount

rpcinfo -p <target>
showmount -e <target>
```

If you observe a .vmdk file,

```
apt install kpartx

mkdir /mnt/new
kpartx -av <flat vmdk>
```

```
mount /dev/mapper/loop0p1 /mnt/new # change, selecting loop w
ith largest size

# if windows
cd /mnt/new/Windows/System32/config
secretsdump.py -sam SAM -system SYSTEM -security SECURITY LOC
AL

# if linux
unshadow /mnt/new/etc/passwd /mnt/new/etc/shadow > ~/hashes.t
xt
john ~/hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

Publicly Mountable Shares

```
sudo su
mkdir /mnt/temp
mount -t nfs <target>:/<mount> /mnt/temp
cd /mnt/temp

umount /mnt/temp
```

Database

MSSQL (1433)

- default/weak password (unauth)
- weak password (auth)
- impersonation
- linked servers
- excessive login permissions
- sysadmin
- server account NTLM relay

Oracle (1521)

- default/weak password

MySQL (3306)

- default/weak password

PostgreSQL (5432)

- default/weak password

Redis (6379)

- default/weak password

Other

SMTP (25,587)

- open relay
- lack of sender validation
- insecure protocol

SNMP (udp/161)

- default RW community string
- default RO community string
- insecure protocol

RDP (3389)

- known CVEs
- weak encryption

Cisco

- Cisco Smart Install
- type 7 encryption
- exposed community strings