Century

Century 2

The password for Century2 is the **build version** of the instance of PowerShell installed on this system.

Solution
In the Powershell, type $PSVersionTable
10.0.14393.7254

Century 3:

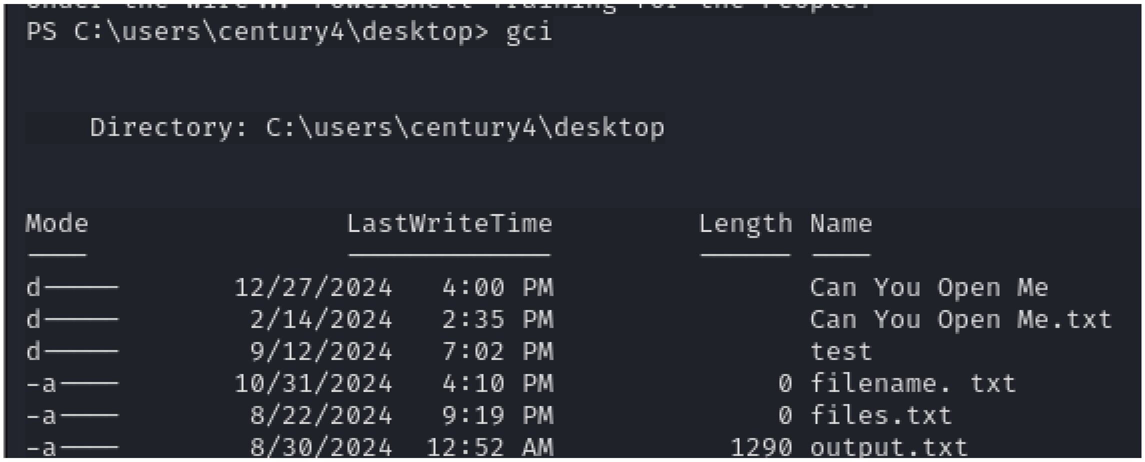
The password for Century3 is the name of the built-in cmdlet that performs the wget like function within PowerShell **PLUS** the name of the file on the desktop.
Solution
Use cmdlet invoke-webrequest
password: invoke-webrequest443

Century 4:

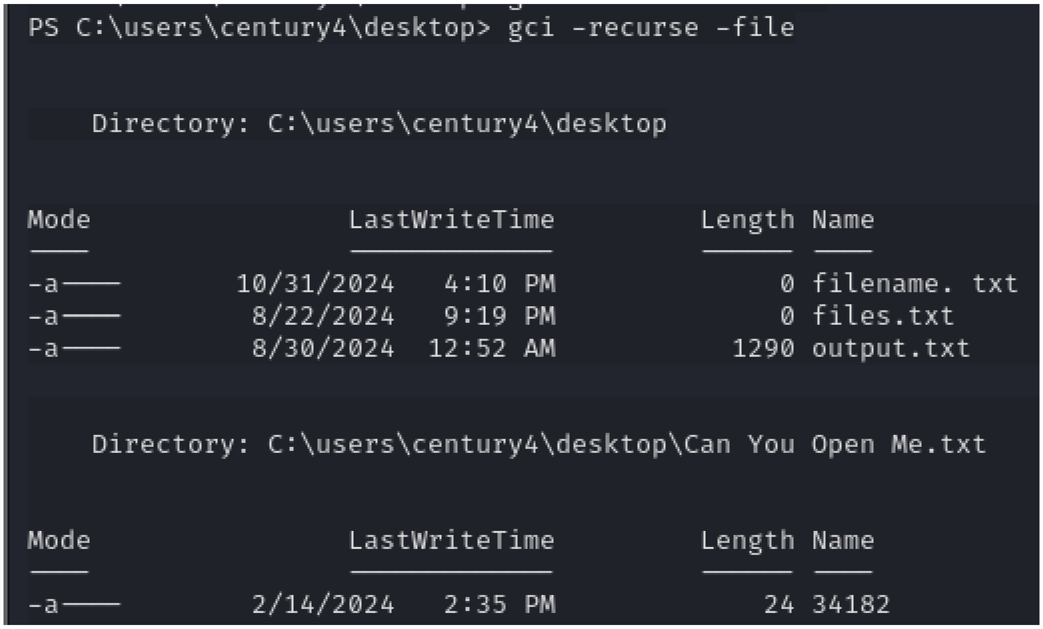
The password for Century4 is the number of files on the desktop.
Solution
Use cmdlet get-childitem | measure-object
Get-childitem lists all objects in a directory
Measure-object counts the objects retrieved from get-childitem
Password: 123

Century 5:

The password for Century 5 is the name of the file within a directory on the desktop that has spaces in its name.
Solution
This is my attempt to list out all the files in this “desktop” directory. And it seems like the directory that has spaces in its name doesn’t contain any files, but the .txt file that has spaces in its name might hold the information we need.
I tested it out and indeed the content of that .txt file was the password. The problem was misleading and took me some time to figure this out.



```
Under the Wire... PowerShell Training for the People!
PS C:\users\century4\desktop> gci


    Directory: C:\users\century4\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        12/27/2024   4:00 PM                Can You Open Me
d-----         2/14/2024   2:35 PM                Can You Open Me.txt
d-----         9/12/2024   7:02 PM                test
-a----        10/31/2024   4:10 PM              0 filename. txt
-a----         8/22/2024   9:19 PM              0 files.txt
-a----         8/30/2024  12:52 AM           1290 output.txt
```



```
PS C:\users\century4\desktop> gci -recurse -file


    Directory: C:\users\century4\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/31/2024   4:10 PM              0 filename. txt
-a----         8/22/2024   9:19 PM              0 files.txt
-a----         8/30/2024  12:52 AM           1290 output.txt


    Directory: C:\users\century4\desktop\Can You Open Me.txt


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/14/2024   2:35 PM             24 34182
```

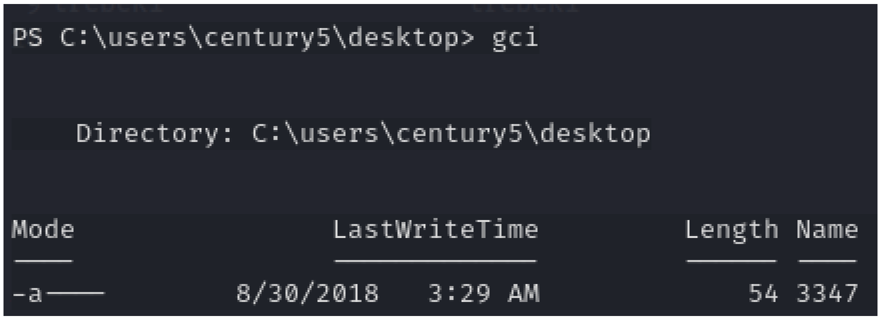
password: 34182

Century 6:

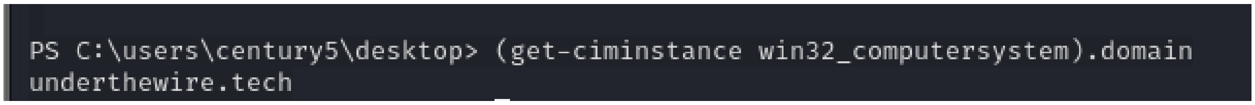
The password for Century6 is the short name of the domain in which this system resides in **PLUS** the name of the file on the desktop.

Solution

gci --> only one file named 3347



```
PS C:\users\century5\desktop> gci


    Directory: C:\users\century5\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:29 AM             54 3347
```



```
PS C:\users\century5\desktop> (get-ciminstance win32_computersystem).domain
underthewire.tech
```

To get the domain name, we can use the following cmdlet
(get-ciminstance win32_computersystem).domain
password: underthewire3347

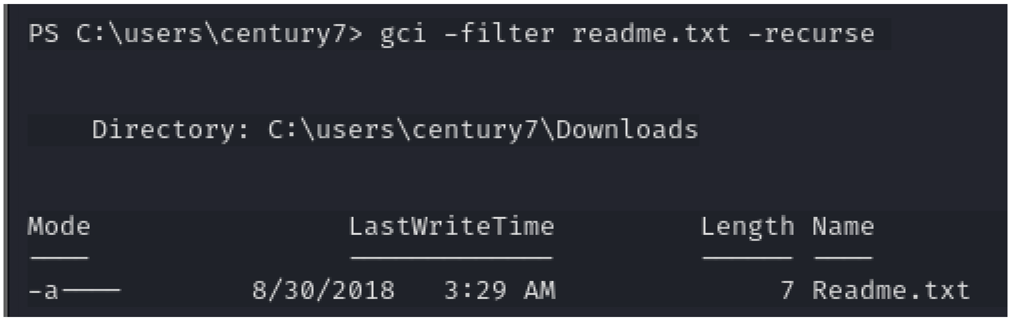
Century 7:

The password for Century7 is the number of folders on the desktop.

Solution
gci | measure
password: 197

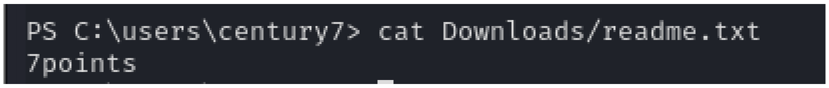
Century 8:

The password for Century8 is in a readme file somewhere within the contacts, desktop, documents, downloads, favorites, music, or videos folder in the user's profile.
Solution
First, I had to go back to the parent directory of "desktop", so I used cd .. to achieve this. Then, with the help of the -filter flag in gci cmdlet, I could specify a string-based query to filter the files or directories returned by the command. I also used the -recurse flag to make sure I could search through all subdirectories recursively.



```
PS C:\users\century7> gci -filter readme.txt -recurse


    Directory: C:\users\century7\Downloads


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:29 AM              7 Readme.txt
```

Once I was able to identify the path of the readme file, I can get the content of it using the cat command or get-content



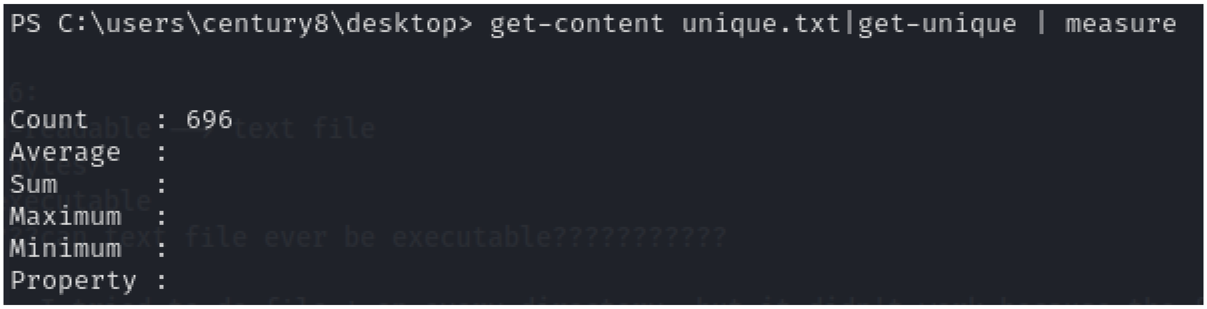
```
PS C:\users\century7> cat Downloads/readme.txt
7points
```

cd ..
gci -filter readme.txt -recurse
cat Downloads\Readme.txt

password: 7points

Century 9

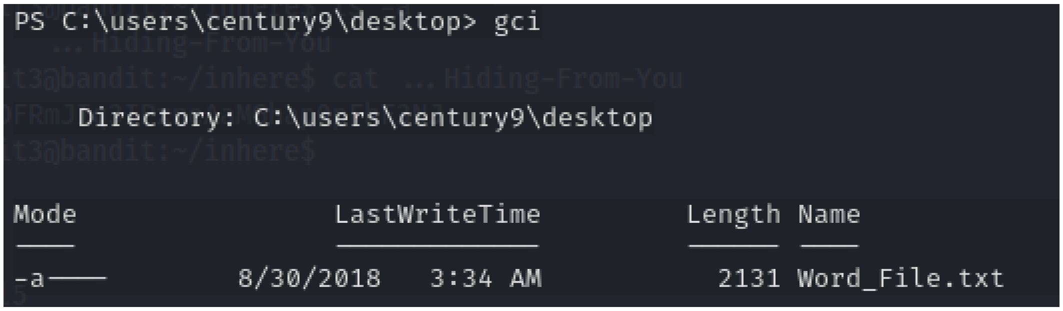
The password for Century9 is the number of unique entries within the file on the desktop.
Solution
PS C:\users\century8\desktop> get-content unique.txt | get-unique | measure
First, I used the Get-unique cmdlet to filter out duplicate entries, then use measure (or Measure-object) to count the entries I got from get-unique
password: 696



```
PS C:\users\century8\desktop> get-content unique.txt|get-unique | measure


Count    : 696
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

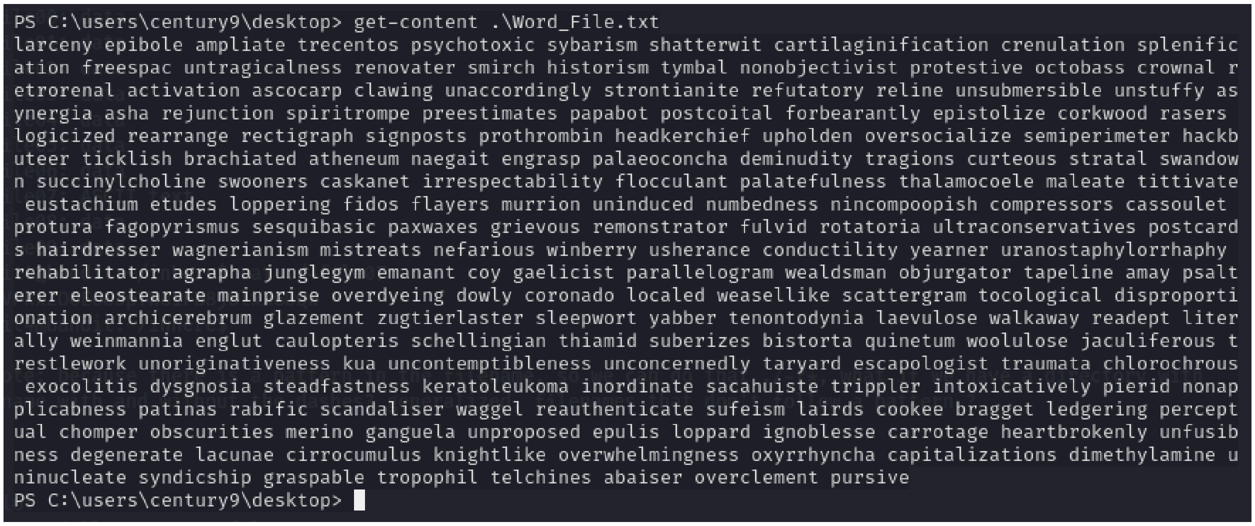
Century 10

The password for Century10 is the **161st** word within the file on the desktop.
Solution



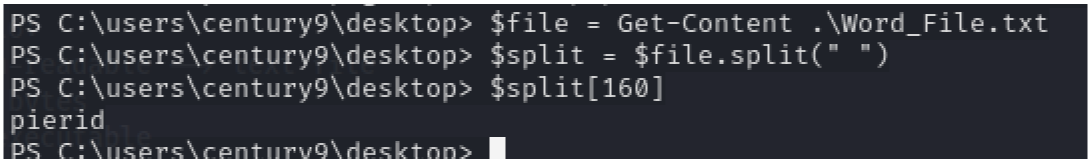
```
PS C:\users\century9\desktop> gci


    Directory: C:\users\century9\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:34 AM           2131 Word_File.txt
```

The only file that we have here is Word_File.txt



```
PS C:\users\century9\desktop> get-content .\Word_File.txt
larceny epibole ampliate trecentos psychotoxic sybarism shatterwit cartilaginification crenulation splenific
ation freespac untragicalness renovater smirch historism tymbal nonobjectivist protestive octobass crownal r
etrorenal activation ascocarp clawing unaccordingly strontianite refutatory reline unsubmersible unstuffy as
ynergia asha rejunction spiritrompe preestimates papabot postcoital forbearantly epistolize corkwood rasers
logicized rearrange rectigraph signposts prothrombin headkerchief upholden oversocialize semiperimeter hackb
uteer ticklish brachiated atheneum naegait engrasp palaeoconcha deminudity tragions curteous stratal swandow
n succinylcholine swooners caskanet irrespectability flocculant palatefulness thalamocoele maleate tittivate
 eustachium etudes loppering fidos flayers murrion uninduced numbedness nincompoopish compressors cassoulet
protura fagopyrismus sesquibasic paxwaxes grievous remonstrator fulvid rotatoria ultraconservatives postcard
s hairdresser wagnerianism mistreats nefarious winberry usherance conductility yearner uranostaphylorrhaphy
rehabilitator agrapha junglegym emanant coy gaelicist parallelogram wealdsman objurgator tapeline amay psalt
erer eleostearate mainprise overdyeing dowly coronado localed weasellike scattergram tocological disproporti
onation archicerebrum glazement zugtierlaster sleepwort yabber tenontodynia laevulose walkaway readept liter
ally weinmannia englut caulopteris schellingian thiamid suberizes bistorta quinetum woolulose jaculiferous t
restlework unoriginativeness kua uncontemptibleness unconcernedly taryard escapologist traumata chlorochrous
 exocolitis dysgnosia steadfastness keratoleukoma inordinate sacahuiste trippler intoxicatively pierid nonap
plicabness patinas rabific scandaliser waggel reauthenticate sufeism lairds cookee bragget ledgering percept
ual chomper obscurities merino ganguela unproposed epulis loppard ignoblesse carrotage heartbrokenly unfusib
ness degenerate lacunae cirrocumulus knightlike overwhelmingness oxyrrhyncha capitalizations dimethylamine u
ninucleate syndicship graspable tropophil telchines abaiser overclement pursive
PS C:\users\century9\desktop>
```

If we do get-content, we can see there are lots of words here. They are separated by a white space.
use .Split() function to break the text into an array of words.
PS C:\users\century9\desktop> $file = get-content .\Word_File.txt
PS C:\users\century9\desktop> $split = $file.Split(" ")
PS C:\users\century9\desktop> $split[160]
pierid



```
PS C:\users\century9\desktop> $file = Get-Content .\Word_File.txt
PS C:\users\century9\desktop> $split = $file.split(" ")
PS C:\users\century9\desktop> $split[160]
pierid
PS C:\users\century9\desktop>
```

password: pierid

Century 11

The password for Century11 is the **10th** and **8th** word of the Windows Update service description combined PLUS the name of the file on the desktop.
Solution
The Windows Update service is managed under the service name wuauserv.
get-service: gsv --> get the services on a local or remote computer

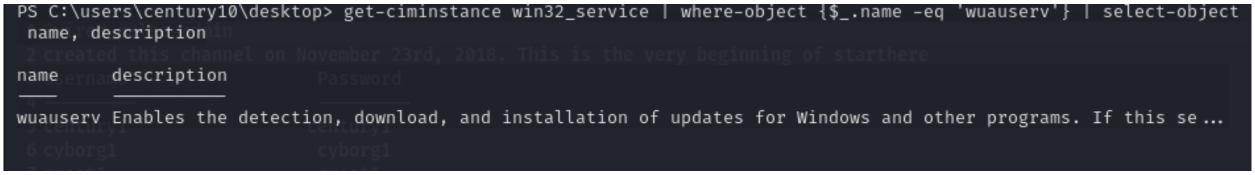
the windows update service has stopped, where to get the description?n

Why does this work but not gsv?
https://powershell.one/wmi/root/cimv2/win32_service

The Get-Service (gsv) cmdlet in PowerShell does not directly provide the service description. This is why we need to use the Get-CimInstance cmdlet, specifically querying the Win32_Service class, to retrieve detailed information, including the description.

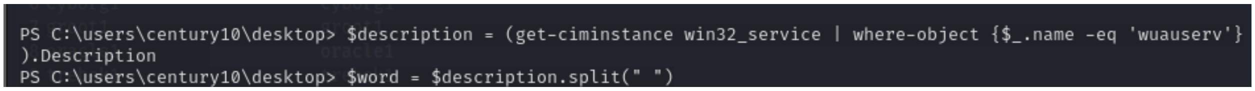
The syntax is Get-CimInstance Win32_Service | Where-Object {$_.Name -eq 'ServiceName'} | Select-Object Name, Description

Use Get-CimInstance to query the Win32_Service class and filter for the wuauserv (Windows Update) service:
get-ciminstance win32_service | where-object {$_.name -eq 'wuauserv'} | select-object
 name, description

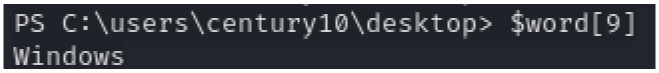


```
PS C:\users\century10\desktop> get-ciminstance win32_service | where-object {$_.name -eq 'wuauserv'} | select-object
 name, description

name     description
----     -----------
wuauserv Enables the detection, download, and installation of updates for Windows and other programs. If this se...
```

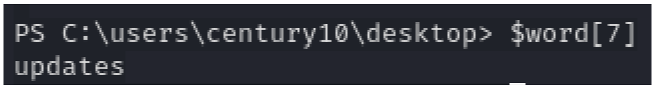
The next step is to split the service description into words. We can use the .Split() method to break the description into an array.
After splitting the string, we can directly index into the array to retrieve the 10th and 8th words. Since arrays are zero-indexed, we need to access the 9th and 7th indexes:



```
PS C:\users\century10\desktop> $description = (get-ciminstance win32_service | where-object {$_.name -eq 'wuauserv'}
).Description
PS C:\users\century10\desktop> $word = $description.split(" ")
```

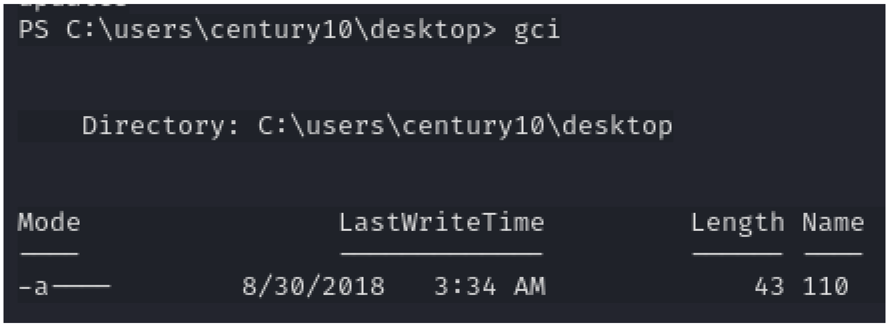


```
PS C:\users\century10\desktop> $word[9]
Windows
```



```
PS C:\users\century10\desktop> $word[7]
updates
```

Then we get the name of the only file in this directory using gci cmdlet:



```
PS C:\users\century10\desktop> gci


    Directory: C:\users\century10\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:34 AM             43 110
```

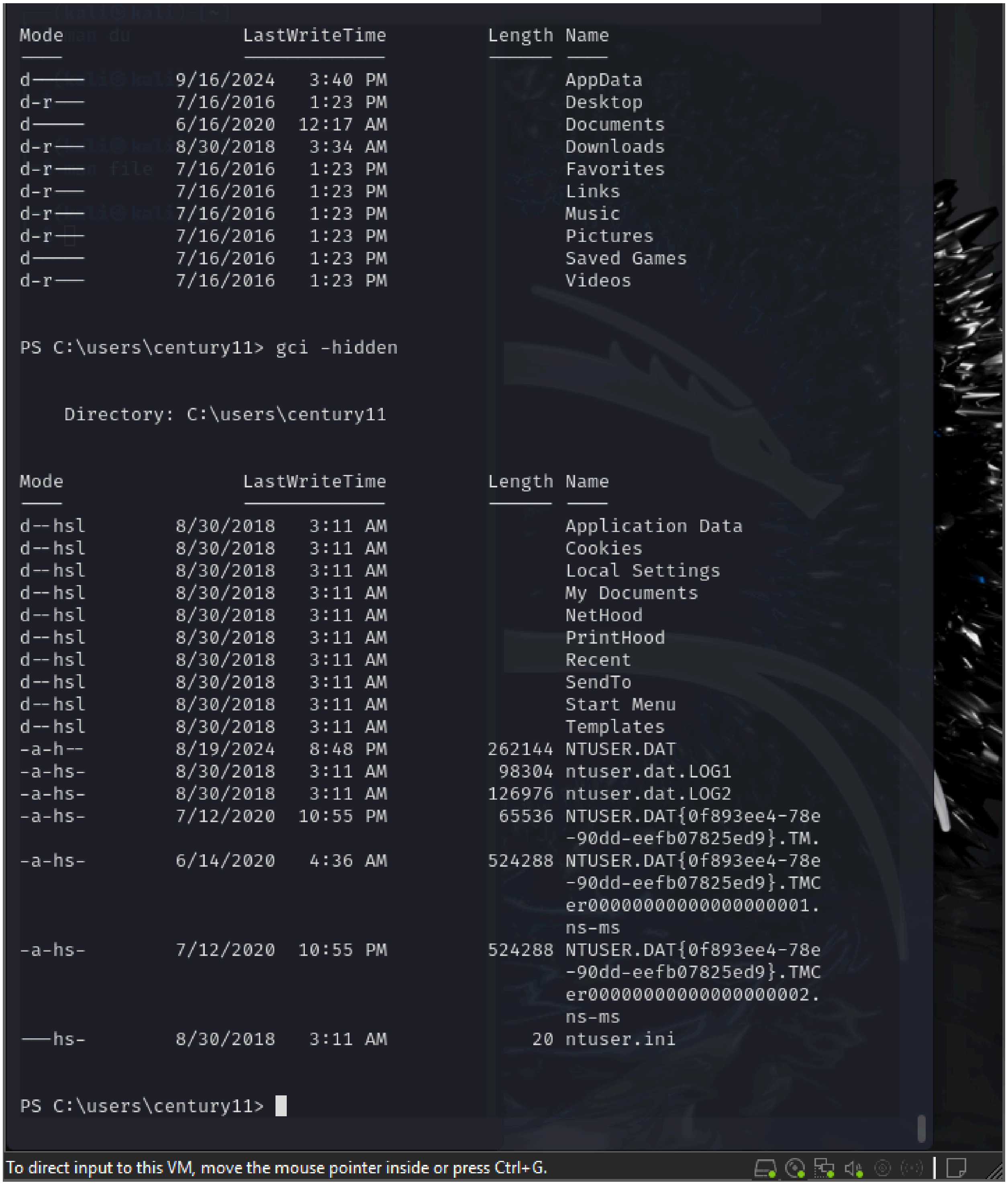
password (for century11): windowsupdates110

Century 12

The password for Century12 is the name of the hidden file within the contacts, desktop, documents, downloads, favorites, music, or videos folder in the user's profile.

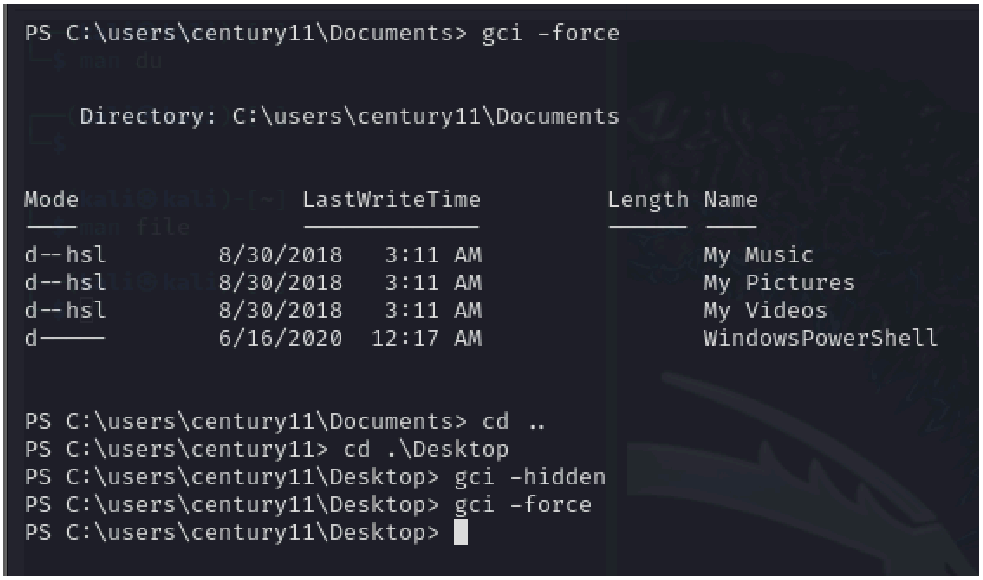
NOTE:
– Exclude "desktop.ini".
– The password will be lowercase no matter how it appears on the screen.

Solution

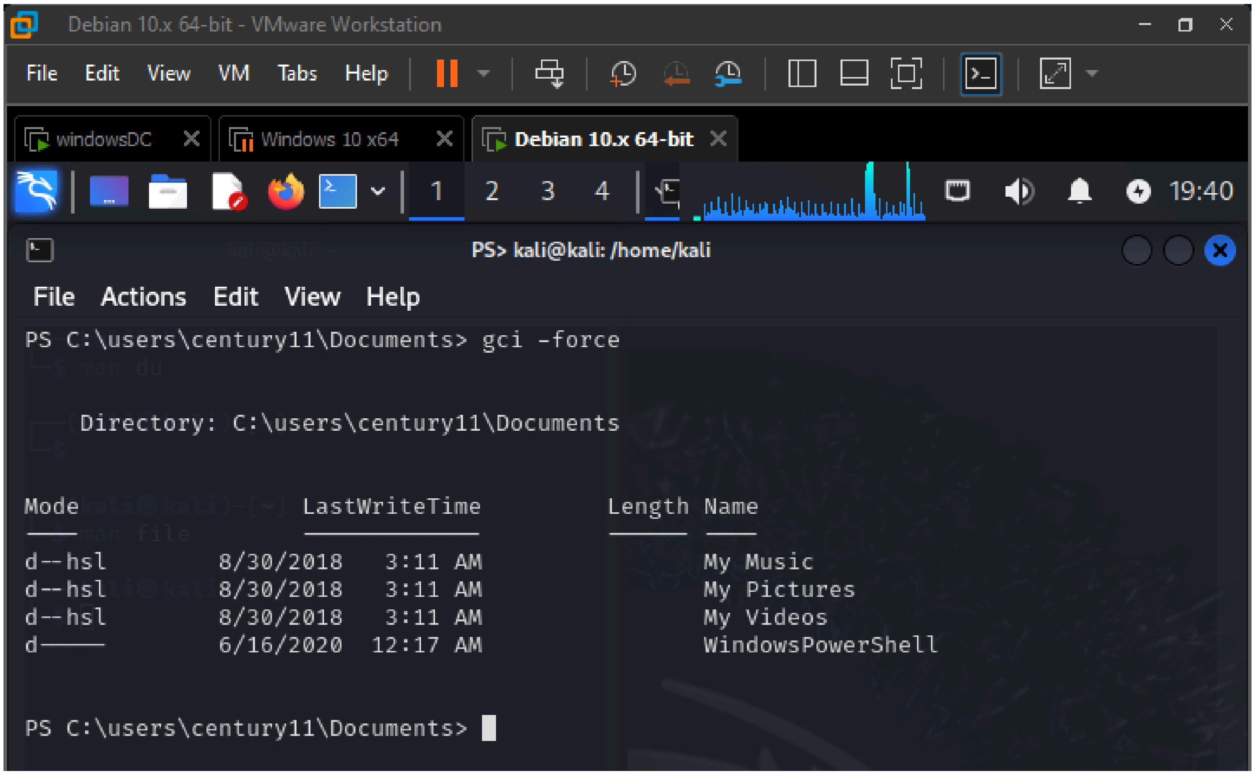


```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/16/2024   3:40 PM                AppData
d-r---         7/16/2016   1:23 PM                Desktop
d-----         6/16/2020  12:17 AM                Documents
d-r---         8/30/2018   3:34 AM                Downloads
d-r---         7/16/2016   1:23 PM                Favorites
d-r---         7/16/2016   1:23 PM                Links
d-r---         7/16/2016   1:23 PM                Music
d-r---         7/16/2016   1:23 PM                Pictures
d-----         7/16/2016   1:23 PM                Saved Games
d-r---         7/16/2016   1:23 PM                Videos


PS C:\users\century11> gci -hidden


    Directory: C:\users\century11


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hsl         8/30/2018   3:11 AM                Application Data
d--hsl         8/30/2018   3:11 AM                Cookies
d--hsl         8/30/2018   3:11 AM                Local Settings
d--hsl         8/30/2018   3:11 AM                My Documents
d--hsl         8/30/2018   3:11 AM                NetHood
d--hsl         8/30/2018   3:11 AM                PrintHood
d--hsl         8/30/2018   3:11 AM                Recent
d--hsl         8/30/2018   3:11 AM                SendTo
d--hsl         8/30/2018   3:11 AM                Start Menu
d--hsl         8/30/2018   3:11 AM                Templates
-a-h--         8/19/2024   8:48 PM         262144 NTUSER.DAT
-a-hs-         8/30/2018   3:11 AM          98304 ntuser.dat.LOG1
-a-hs-         8/30/2018   3:11 AM         126976 ntuser.dat.LOG2
-a-hs-         7/12/2020  10:55 PM          65536 NTUSER.DAT{0f893ee4-78e
                                                  -90dd-eefb07825ed9}.TM.
-a-hs-         6/14/2020   4:36 AM         524288 NTUSER.DAT{0f893ee4-78e
                                                  -90dd-eefb07825ed9}.TMC
                                                  er00000000000000000001.
                                                  ns-ms
-a-hs-         7/12/2020  10:55 PM         524288 NTUSER.DAT{0f893ee4-78e
                                                  -90dd-eefb07825ed9}.TMC
                                                  er00000000000000000002.
                                                  ns-ms
---hs-         8/30/2018   3:11 AM             20 ntuser.ini


PS C:\users\century11>
```

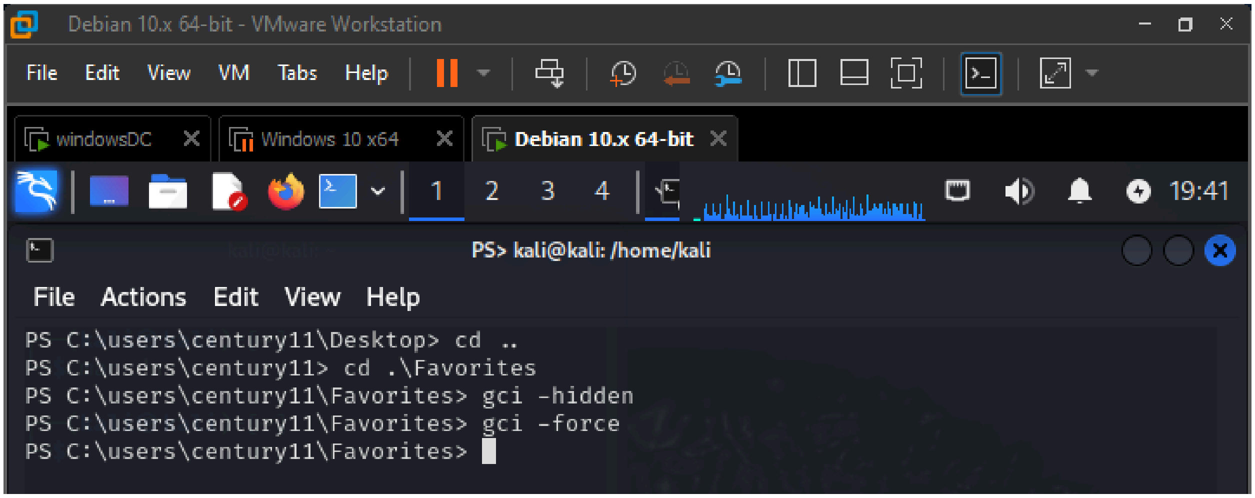
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



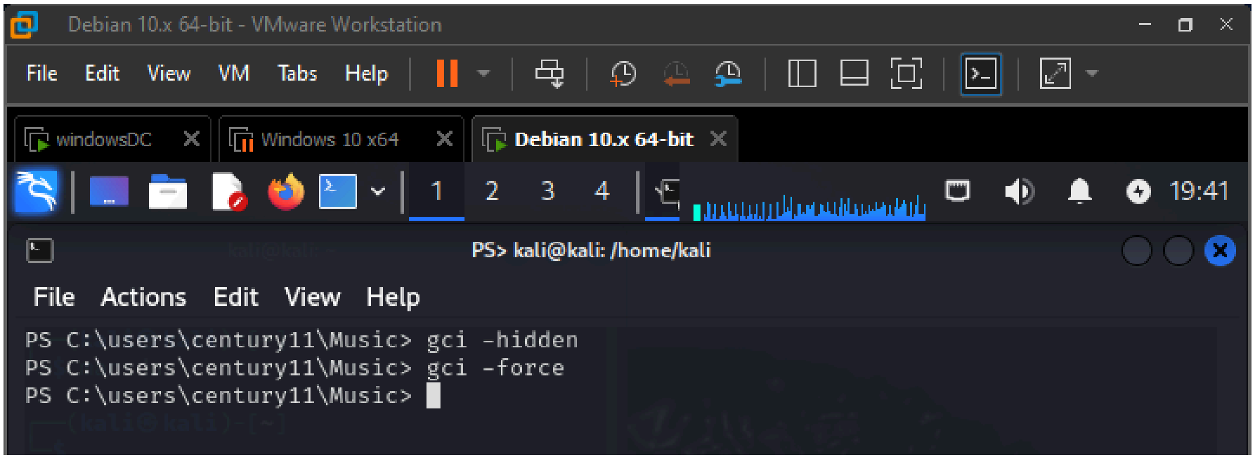
```
PS C:\users\century11\Documents> gci -force


    Directory: C:\users\century11\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hsl         8/30/2018   3:11 AM                My Music
d--hsl         8/30/2018   3:11 AM                My Pictures
d--hsl         8/30/2018   3:11 AM                My Videos
d-----         6/16/2020  12:17 AM                WindowsPowerShell


PS C:\users\century11\Documents> cd ..
PS C:\users\century11> cd .\Desktop
PS C:\users\century11\Desktop> gci -hidden
PS C:\users\century11\Desktop> gci -force
PS C:\users\century11\Desktop>
```



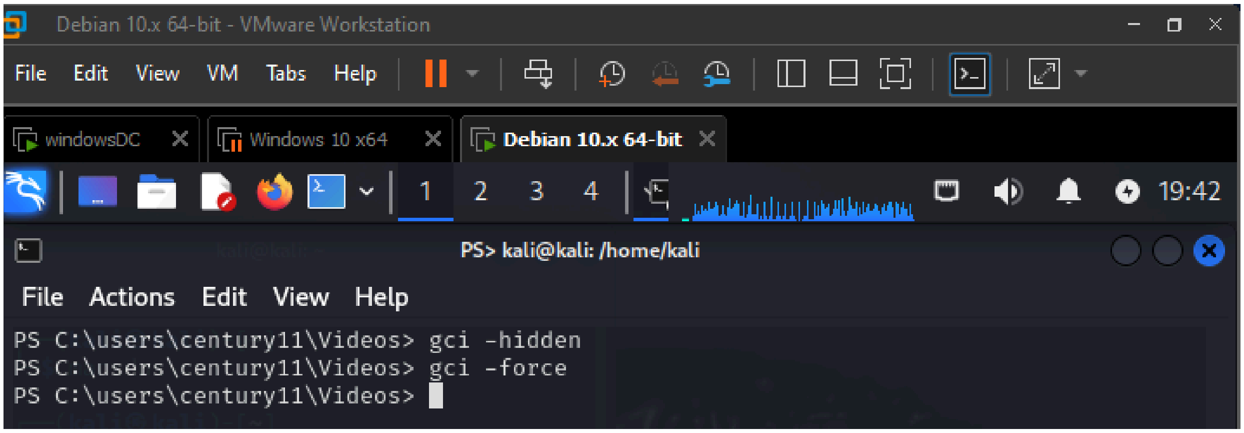
```
PS C:\users\century11\Documents> gci -force


    Directory: C:\users\century11\Documents


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hsl         8/30/2018   3:11 AM                My Music
d--hsl         8/30/2018   3:11 AM                My Pictures
d--hsl         8/30/2018   3:11 AM                My Videos
d-----         6/16/2020  12:17 AM                WindowsPowerShell


PS C:\users\century11\Documents>
```


Debian 10.x 64-bit - VMware Workstation
File Edit View VM Tabs Help
windowsDC
Windows 10 x64
Debian 10.x 64-bit
1 2 3 4
19:41
PS> kali@kali: /home/kali
File Actions Edit View Help

```
PS C:\users\century11\Desktop> cd ..
PS C:\users\century11> cd .\Favorites
PS C:\users\century11\Favorites> gci -hidden
PS C:\users\century11\Favorites> gci -force
PS C:\users\century11\Favorites> █
```

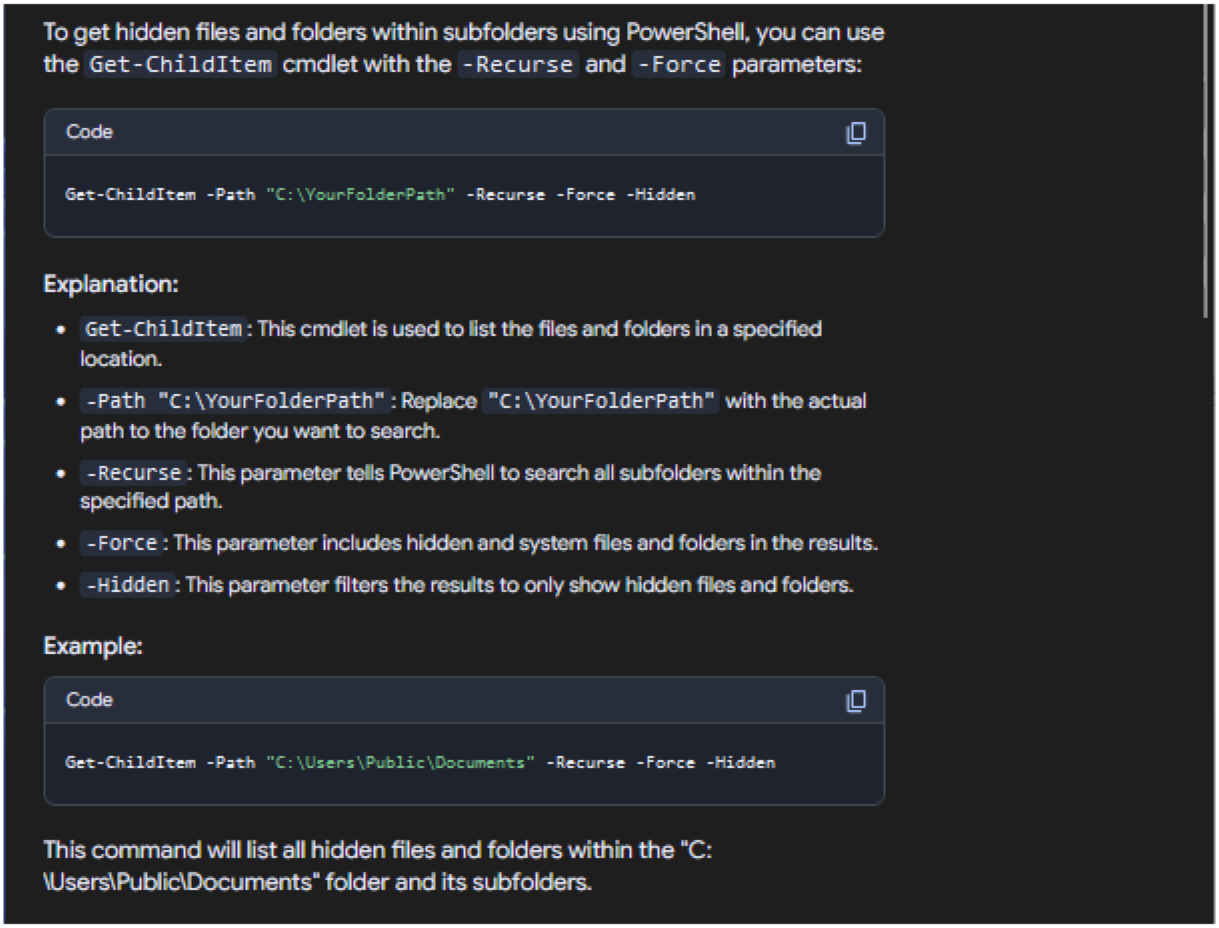

Debian 10.x 64-bit - VMware Workstation
File Edit View VM Tabs Help
windowsDC
Windows 10 x64
Debian 10.x 64-bit
1 2 3 4
19:41
PS> kali@kali: /home/kali
File Actions Edit View Help

```
PS C:\users\century11\Music> gci -hidden
PS C:\users\century11\Music> gci -force
PS C:\users\century11\Music> █
```


Debian 10.x 64-bit - VMware Workstation
File Edit View VM Tabs Help
windowsDC
Windows 10 x64
Debian 10.x 64-bit
1 2 3 4
19:42
PS> kali@kali: /home/kali
File Actions Edit View Help

```
PS C:\users\century11\Videos> gci -hidden
PS C:\users\century11\Videos> gci -force
PS C:\users\century11\Videos> █
```

Get-ChildItem -Path "C:\YourFolderPath" -Recurse -Force -Hidden



To get hidden files and folders within subfolders using PowerShell, you can use the `Get-ChildItem` cmdlet with the `-Recurse` and `-Force` parameters:

```
Get-ChildItem -Path "C:\YourFolderPath" -Recurse -Force -Hidden
```

Explanation:

- `Get-ChildItem`: This cmdlet is used to list the files and folders in a specified location.
- `-Path "C:\YourFolderPath"`: Replace `"C:\YourFolderPath"` with the actual path to the folder you want to search.
- `-Recurse`: This parameter tells PowerShell to search all subfolders within the specified path.
- `-Force`: This parameter includes hidden and system files and folders in the results.
- `-Hidden`: This parameter filters the results to only show hidden files and folders.

Example:

```
Get-ChildItem -Path "C:\Users\Public\Documents" -Recurse -Force -Hidden
```

This command will list all hidden files and folders within the "C:\Users\Public\Documents" folder and its subfolders.

Password: secret_sauce

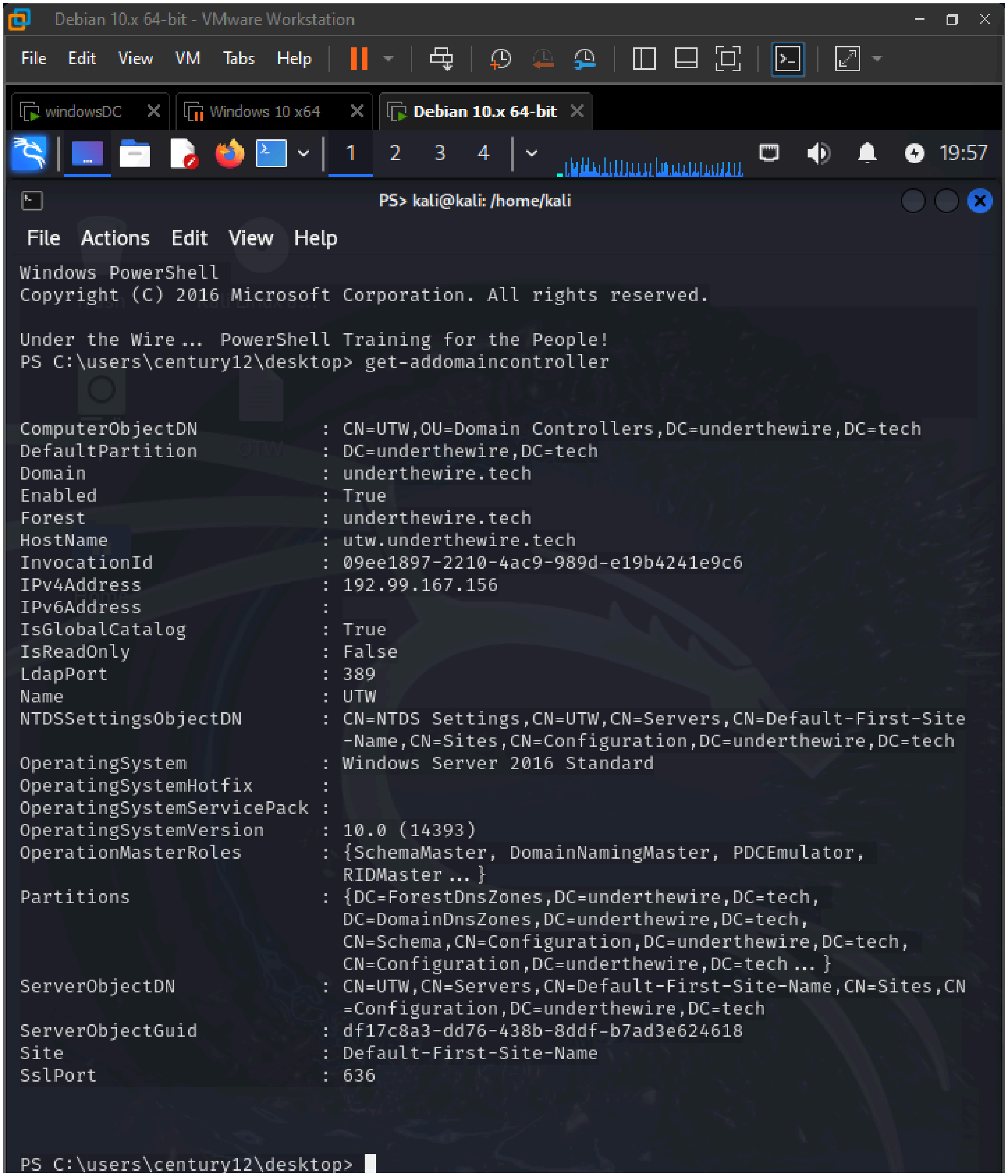
Century 13

The password for Century13 is the description of the computer designated as a Domain Controller within this domain **PLUS** the name of the file on the desktop.

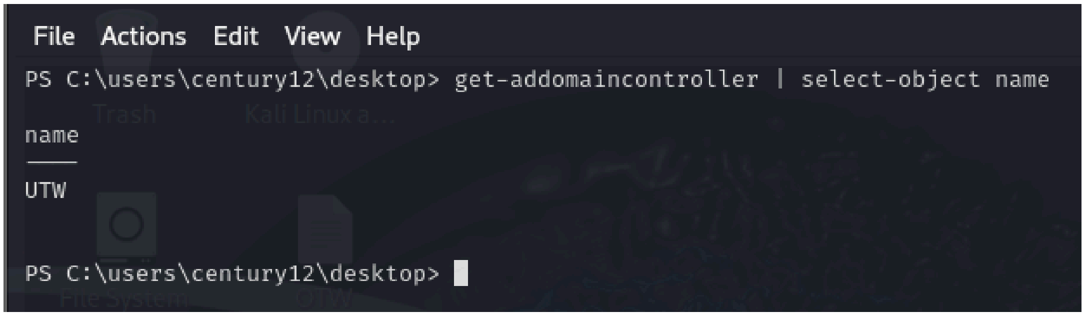
NOTE:
– The password will be lowercase no matter how it appears on the screen.
– If the description “today_is” and the file on the desktop is named “_cool”, the password would be “today_is_cool”.

Solution

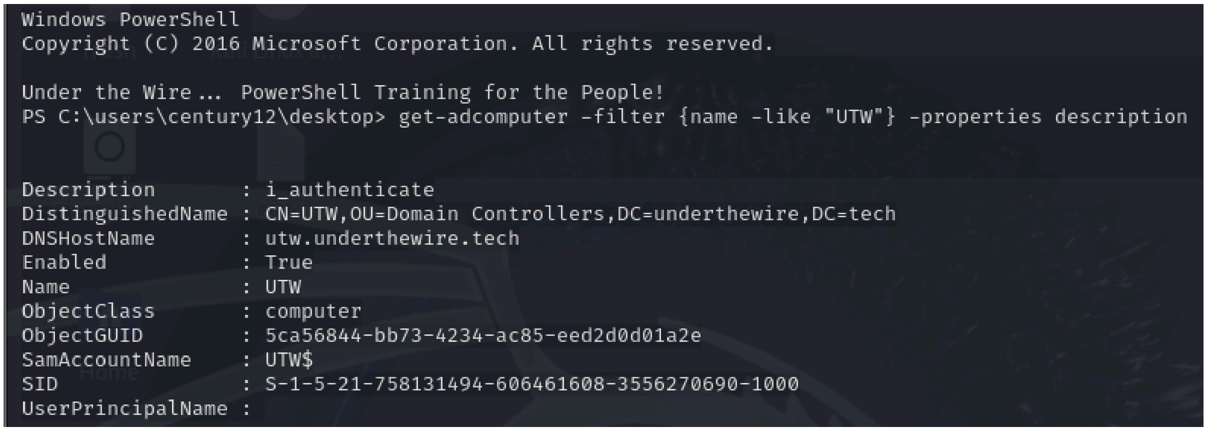
Get-ADDomainController will return a list of all domain controllers in your domain.
I didn’t know how to exactly get the “description” of a computer? I tried get-computerinfo and get-adcomputer -identity UTW but none of them gives “description” property.


Debian 10.x 64-bit - VMware Workstation
File Edit View VM Tabs Help
windowsDC
Windows 10 x64
Debian 10.x 64-bit
1 2 3 4
19:57
PS> kali@kali: /home/kali
File Actions Edit View Help

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Under the Wire ... PowerShell Training for the People!
PS C:\users\century12\desktop> get-addomaincontroller


ComputerObjectDN           : CN=UTW,OU=Domain Controllers,DC=underthewire,DC=tech
DefaultPartition           : DC=underthewire,DC=tech
Domain                     : underthewire.tech
Enabled                    : True
Forest                     : underthewire.tech
HostName                   : utw.underthewire.tech
InvocationId               : 09ee1897-2210-4ac9-989d-e19b4241e9c6
IPv4Address                : 192.99.167.156
IPv6Address                :
IsGlobalCatalog            : True
IsReadOnly                 : False
LdapPort                   : 389
Name                       : UTW
NTDSSettingsObjectDN       : CN=NTDS Settings,CN=UTW,CN=Servers,CN=Default-First-Site
                             -Name,CN=Sites,CN=Configuration,DC=underthewire,DC=tech
OperatingSystem            : Windows Server 2016 Standard
OperatingSystemHotfix      :
OperatingSystemServicePack :
OperatingSystemVersion     : 10.0 (14393)
OperationMasterRoles       : {SchemaMaster, DomainNamingMaster, PDCEmulator,
                             RIDMaster ... }
Partitions                 : {DC=ForestDnsZones,DC=underthewire,DC=tech,
                             DC=DomainDnsZones,DC=underthewire,DC=tech,
                             CN=Schema,CN=Configuration,DC=underthewire,DC=tech,
                             CN=Configuration,DC=underthewire,DC=tech ... }
ServerObjectDN             : CN=UTW,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN
                             =Configuration,DC=underthewire,DC=tech
ServerObjectGuid           : df17c8a3-dd76-438b-8ddf-b7ad3e624618
Site                       : Default-First-Site-Name
SslPort                    : 636


PS C:\users\century12\desktop>
```

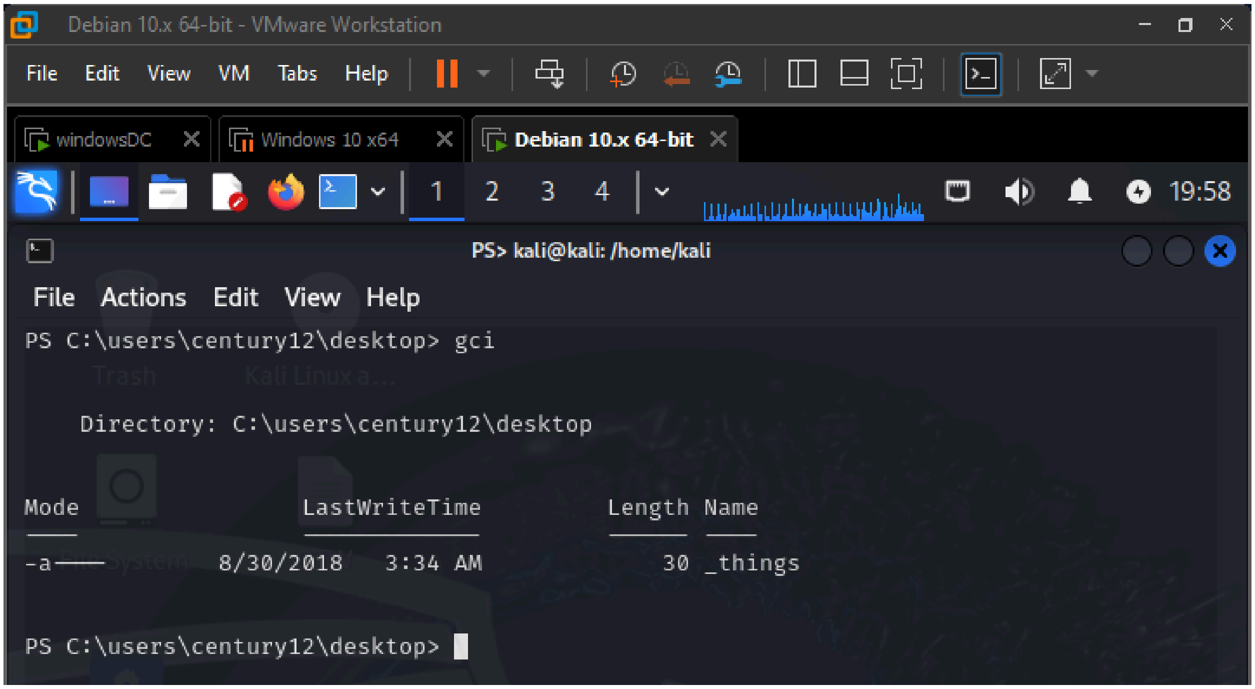


```
PS C:\users\century12\desktop> get-addomaincontroller | select-object name

name
----
UTW


PS C:\users\century12\desktop>
```



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Under the Wire ... PowerShell Training for the People!
PS C:\users\century12\desktop> get-adcomputer -filter {name -like "UTW"} -properties description


Description       : i_authenticate
DistinguishedName : CN=UTW,OU=Domain Controllers,DC=underthewire,DC=tech
DNSHostName       : utw.underthewire.tech
Enabled           : True
Name              : UTW
ObjectClass       : computer
ObjectGUID        : 5ca56844-bb73-4234-ac85-eed2d0d01a2e
SamAccountName    : UTW$
SID               : S-1-5-21-758131494-606461608-3556270690-1000
UserPrincipalName :
```

source



```
PS C:\users\century12\desktop> gci


    Directory: C:\users\century12\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:34 AM             30 _things


PS C:\users\century12\desktop>
```

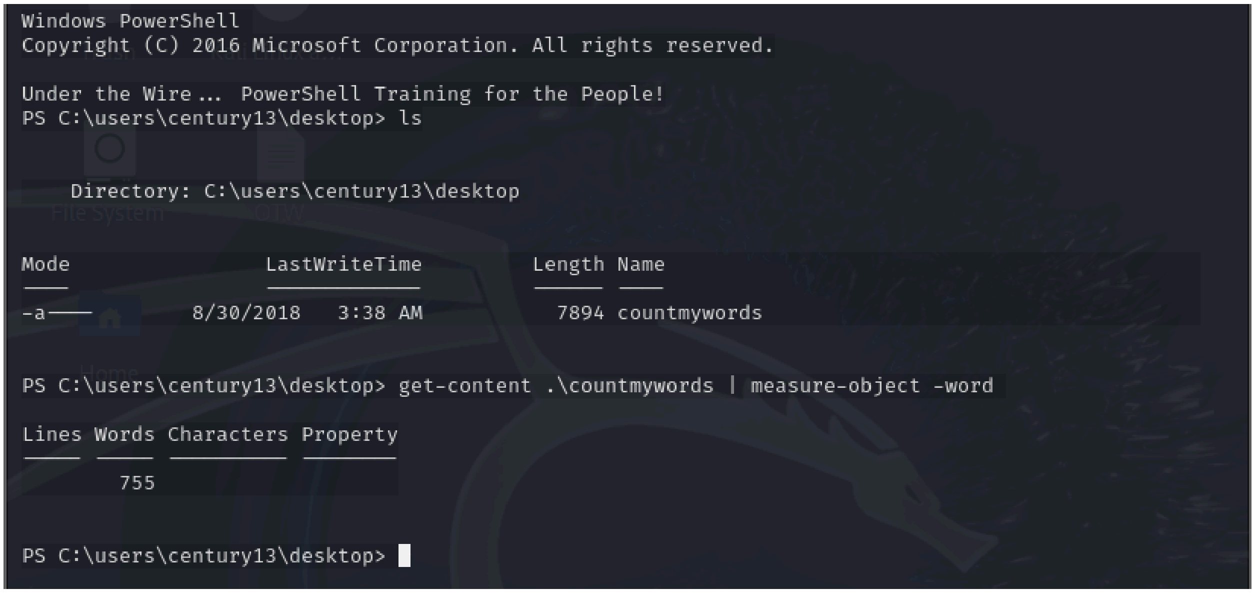
Password: i_authenticate_things

Century 14

The password for Century14 is the number of words within the file on the desktop.

Solution

Use: get-content .\countmywords | measure-object -word



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Under the Wire ... PowerShell Training for the People!
PS C:\users\century13\desktop> ls


    Directory: C:\users\century13\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018   3:38 AM           7894 countmywords


PS C:\users\century13\desktop> get-content .\countmywords | measure-object -word

Lines Words Characters Property
----- ----- ---------- --------
        755


PS C:\users\century13\desktop>
```

Password: 755

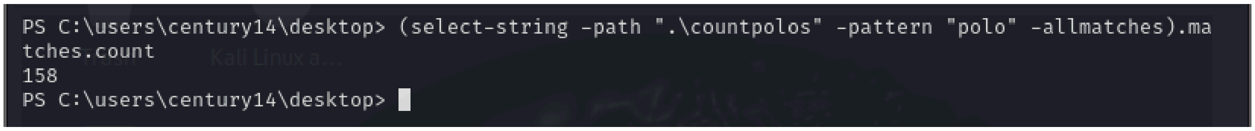
Century 15

Count the number of times a specific word shows up

The password for Century15 is the number of times the word “polo” appears within the file on the desktop.

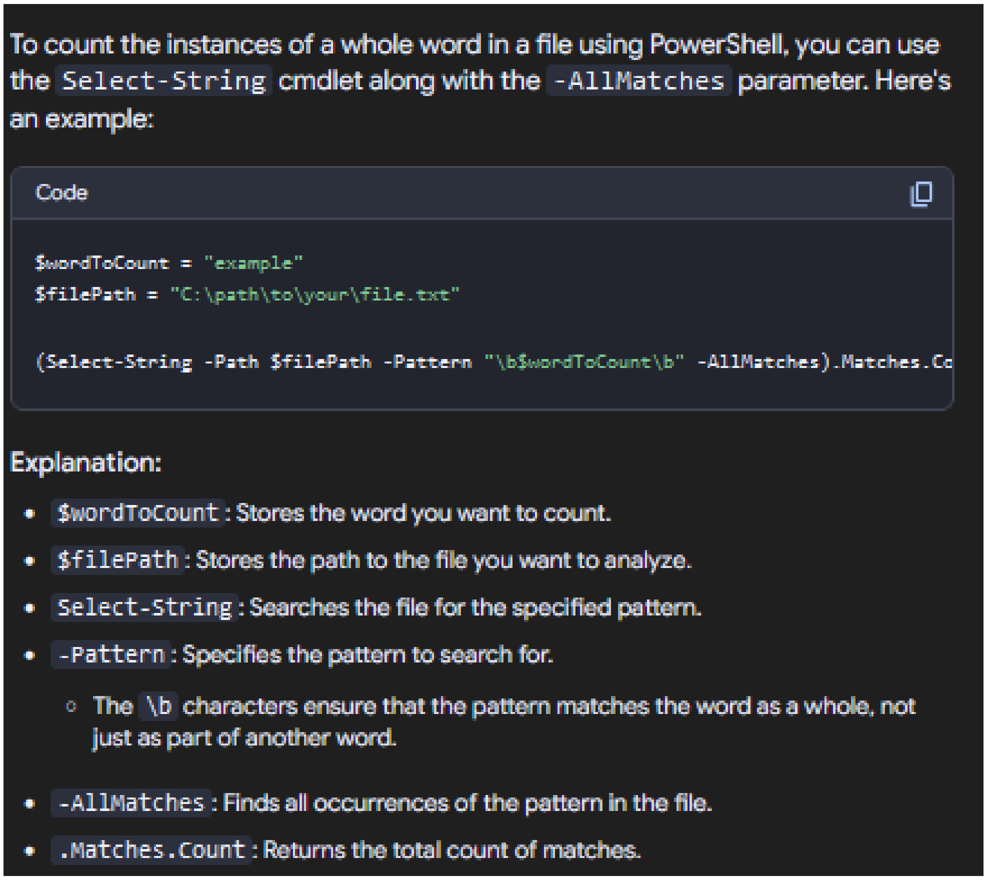
NOTE:
– You should count the instances of the **whole word** only.

Solution



```
PS C:\users\century14\desktop> (select-string -path ".\countpolos" -pattern "polo" -allmatches).ma
tches.count
158
PS C:\users\century14\desktop>
```

This gives the wrong answer since the match could not be a whole word.

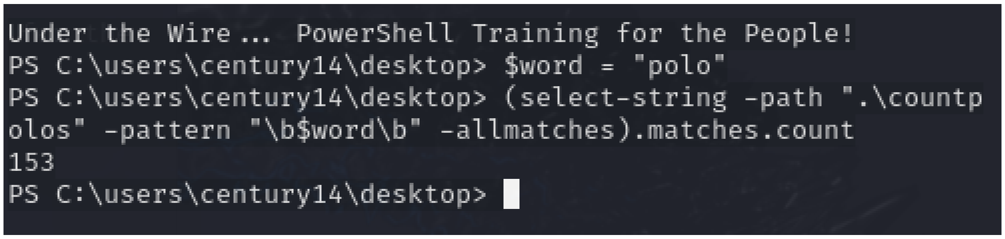


To count the instances of a whole word in a file using PowerShell, you can use the `Select-String` cmdlet along with the `-AllMatches` parameter. Here's an example:

```
$wordToCount = "example"
$filePath = "C:\path\to\your\file.txt"

(Select-String -Path $filePath -Pattern "\b$wordToCount\b" -AllMatches).Matches.Co
```

Explanation:

- `$wordToCount`: Stores the word you want to count.
- `$filePath`: Stores the path to the file you want to analyze.
- `Select-String`: Searches the file for the specified pattern.
- `-Pattern`: Specifies the pattern to search for.
  - The `\b` characters ensure that the pattern matches the word as a whole, not just as part of another word.
- `-AllMatches`: Finds all occurrences of the pattern in the file.
- `.Matches.Count`: Returns the total count of matches.

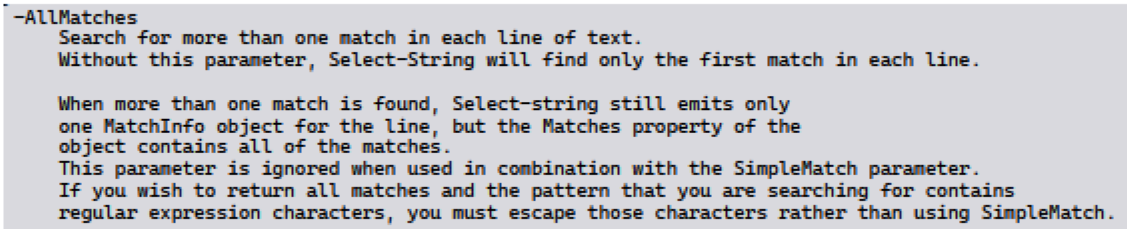


```
Under the Wire... PowerShell Training for the People!
PS C:\users\century14\desktop> $word = "polo"
PS C:\users\century14\desktop> (select-string -path ".\countp
olos" -pattern "\b$word\b" -allmatches).matches.count
153
PS C:\users\century14\desktop>
```

Password: 153

(select-string -path “.\countpolos” -pattern “\b$word\b” -allmatches).matches.count

-allmatches parameter:



```
-AllMatches
    Search for more than one match in each line of text.
    Without this parameter, Select-String will find only the first match in each line.

    When more than one match is found, Select-string still emits only
    one MatchInfo object for the line, but the Matches property of the
    object contains all of the matches.
    This parameter is ignored when used in combination with the SimpleMatch parameter.
    If you wish to return all matches and the pattern that you are searching for contains
    regular expression characters, you must escape those characters rather than using SimpleMatch.
```

Select-string and Grep

Select-string official documentation

Most common usage of select-string: find text patterns in field and strings, search through multiple files, report the location including the line number of the string for each find, find only the first occurrence of string, or return only the files that don't match a string, etc. :https://lazyadmin.nl/powershell/powershell-grep-select-string/

Powershell pipeline (good to know)

Cyborg

Cyborg 1

Password: cyborg1

Cyborg 2

The password for cyborg2 is the state that the user Chris Rogers is from as stated within Active Directory.

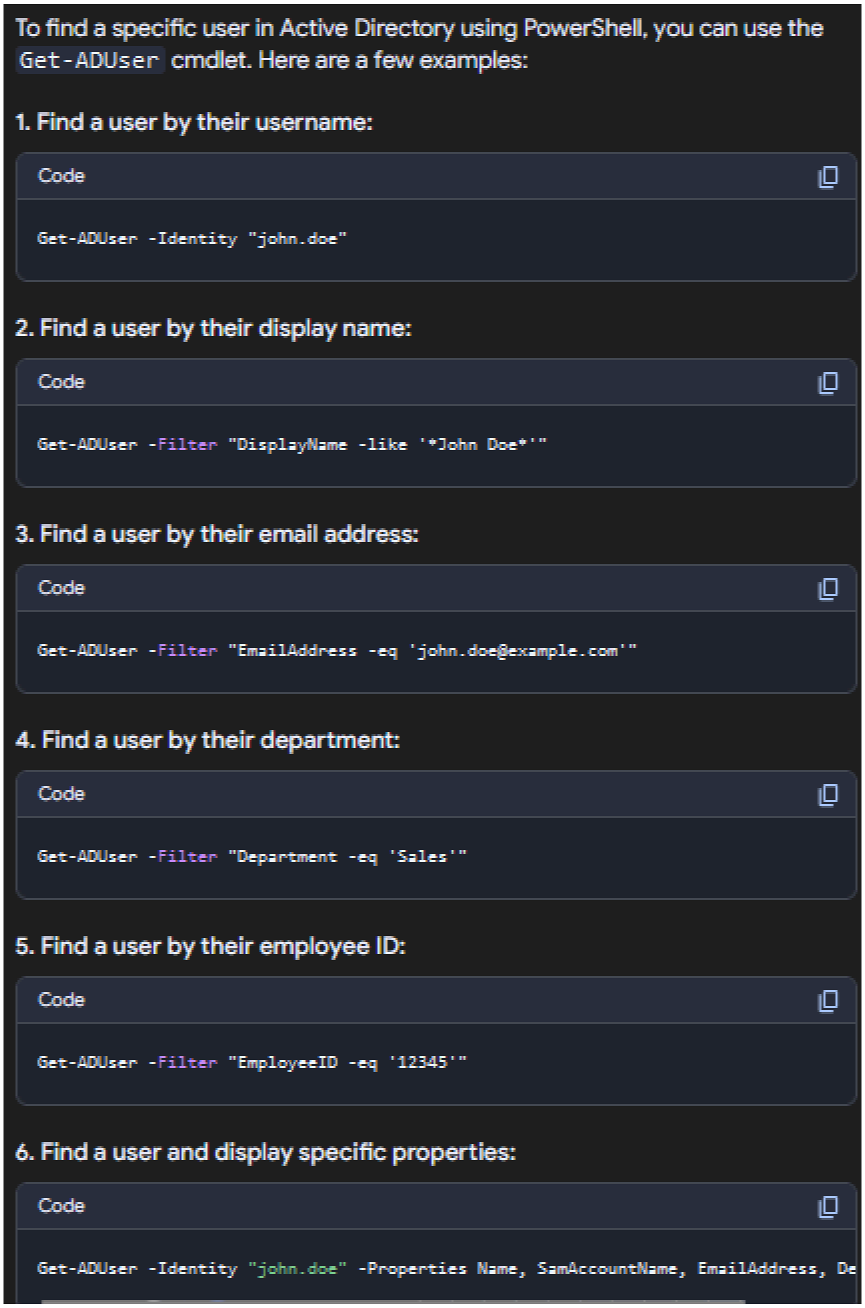
NOTE:
– The password will be lowercase no matter how it appears on the screen.
– "State" refers to the location within the country and NOT the "state" of the account (enabled/ disabled).

IMPORTANT:
Once you feel you have completed the Cyborg1 challenge, start a new connection to the server, and log in with the username of Cyborg2 and this password will be the answer from Cyborg1. If successful, close out the Cyborg1 connection and begin to solve the Cyborg2 challenge. This concept is repeated over and over until you reach the end of the game.

▼ **HINT:**

List the available modules, there may be a useful one available…

Solution



To find a specific user in Active Directory using PowerShell, you can use the `Get-ADUser` cmdlet. Here are a few examples:

1. Find a user by their username:

```
Get-ADUser -Identity "john.doe"
```

2. Find a user by their display name:

```
Get-ADUser -Filter "DisplayName -like '*John Doe*'"
```

3. Find a user by their email address:

```
Get-ADUser -Filter "EmailAddress -eq 'john.doe@example.com'"
```

4. Find a user by their department:

```
Get-ADUser -Filter "Department -eq 'Sales'"
```

5. Find a user by their employee ID:

```
Get-ADUser -Filter "EmployeeID -eq '12345'"
```

6. Find a user and display specific properties:

```
Get-ADUser -Identity "john.doe" -Properties Name, SamAccountName, EmailAddress, De
```

Use the cmdlet *Get-aduser -identity “chris.rogers” -properties **
Password: kansas

Cyborg 3

The password for cyborg3 is the host A record IP address for CYBORG718W100N **PLUS** the name of the file on the desktop.

NOTE:

– If the IP is “10.10.1.5” and the file on the desktop is called “_address”, then the password is “10.10.1.5_address”.
– The password will be lowercase no matter how it appears on the screen.

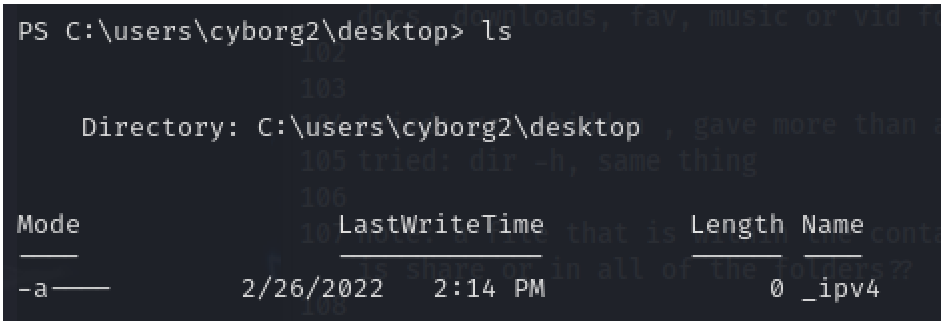
▼ **HINT:**

WMI or cmdlets… choices, choices.

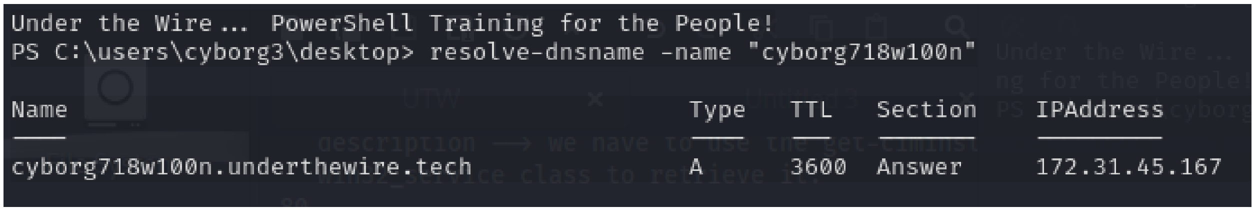
▼ **HINT:**

Each domain client has its own specific Zone Name.

Solution



```
PS C:\users\cyborg2\desktop> ls


    Directory: C:\users\cyborg2\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          2/26/2022   2:14 PM              0 _ipv4
```

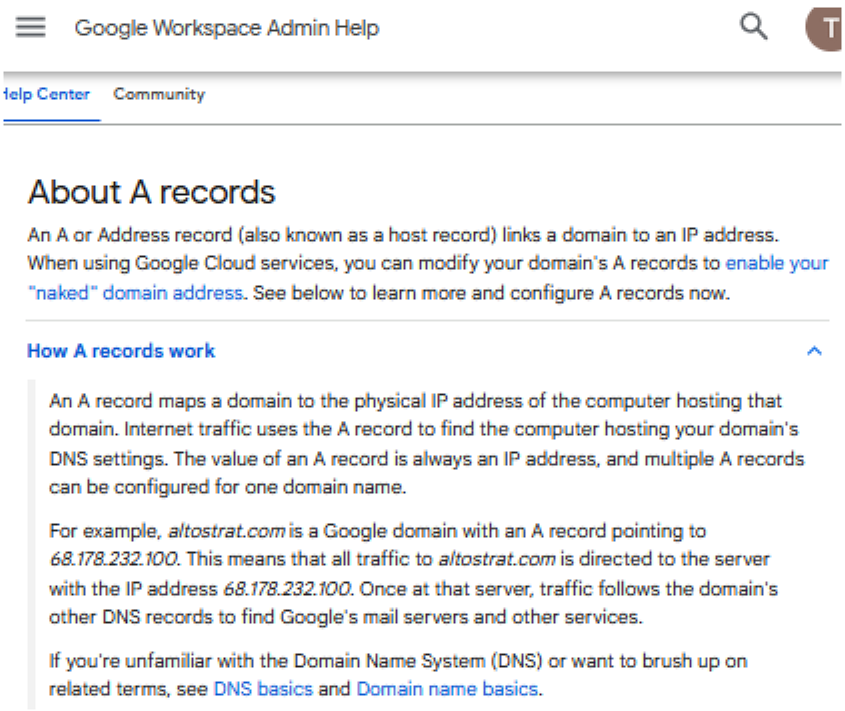


```
Under the Wire... PowerShell Training for the People!
PS C:\users\cyborg3\desktop> resolve-dnsname -name "cyborg718w100n"

Name                                           Type   TTL   Section    IPAddress
----                                           ----   ---   -------    ---------
cyborg718w100n.underthewire.tech               A      3600  Answer     172.31.45.167
```

To get the name of the file in desktop directory, simply do ls or gci.
Password: 172.31.45.167_ipv4

Q&A

- What is A record? – or address record (or host record) that indicates the IP address of a domain cloudfare



About A records

An A or Address record (also known as a host record) links a domain to an IP address. When using Google Cloud services, you can modify your domain's A records to enable your "naked" domain address. See below to learn more and configure A records now.

How A records work

> An A record maps a domain to the physical IP address of the computer hosting that domain. Internet traffic uses the A record to find the computer hosting your domain's DNS settings. The value of an A record is always an IP address, and multiple A records can be configured for one domain name.
>
> For example, *altostrat.com* is a Google domain with an A record pointing to *68.178.232.100*. This means that all traffic to *altostrat.com* is directed to the server with the IP address *68.178.232.100*. Once at that server, traffic follows the domain's other DNS records to find Google's mail servers and other services.
>
> If you're unfamiliar with the Domain Name System (DNS) or want to brush up on related terms, see DNS basics and Domain name basics.

Useful links: DNS basics Domain Name basics

- What is WMI? How is it different from cmdlets?
- What is zone name?
- *******resolve -dnname ~ NSLookup
  https://www.pdq.com/blog/what-is-the-powershell-equivalent-of-nslookup/

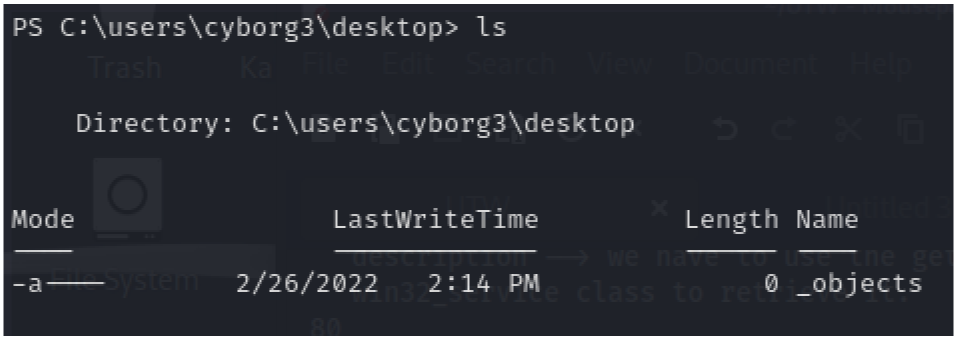
Cyborg 4

The password for cyborg4 is the number of users in the Cyborg group within Active Directory PLUS the name of the file on the desktop.

NOTE:
– If the number of users is “20” and the file on the desktop is called “_users”, then the password is “20_users”.
– The password will be lowercase no matter how it appears on the screen.

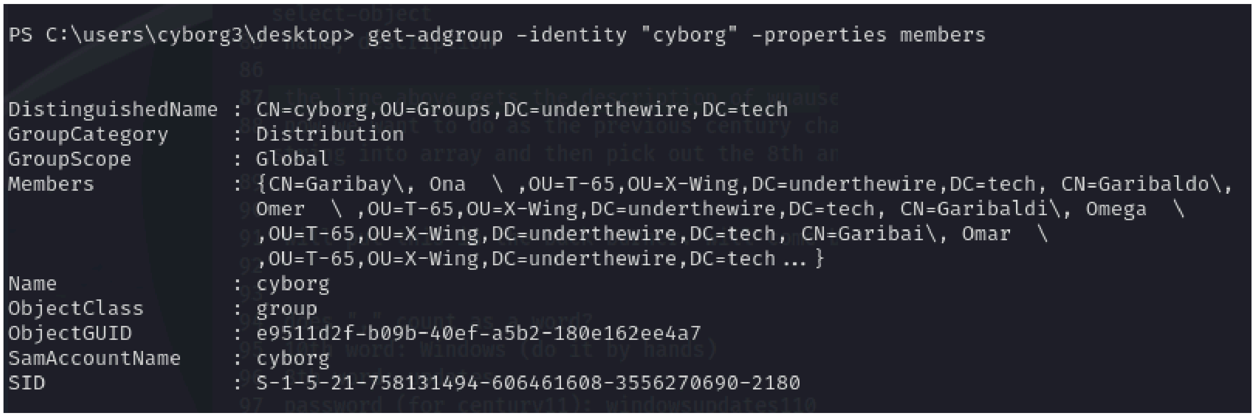
▼ HINT:

https://technet.microsoft.com/en-us/library/ee617195.aspx
Solution



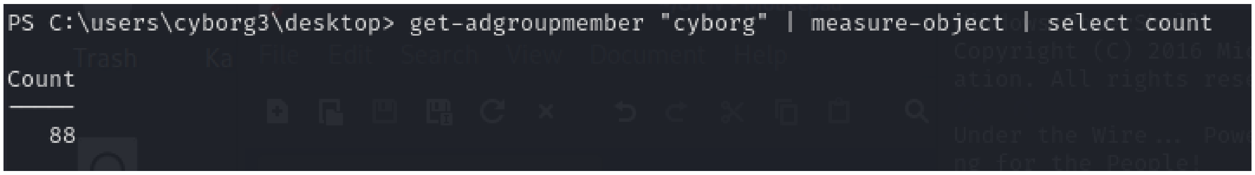
```
PS C:\users\cyborg3\desktop> ls


    Directory: C:\users\cyborg3\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        2/26/2022   2:14 PM              0 _objects
```

To get a specific group by name, use this cmdlet: `Get-ADGroup -Identity "GroupName"`



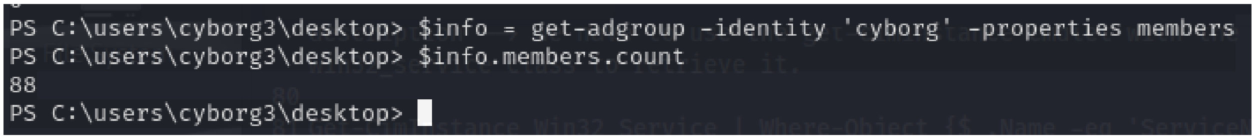
```
PS C:\users\cyborg3\desktop> get-adgroup -identity "cyborg" -properties members


DistinguishedName : CN=cyborg,OU=Groups,DC=underthewire,DC=tech
GroupCategory     : Distribution
GroupScope        : Global
Members           : {CN=Garibay\, Ona  \ ,OU=T-65,OU=X-Wing,DC=underthewire,DC=tech, CN=Garibaldo\,
                    Omer  \ ,OU=T-65,OU=X-Wing,DC=underthewire,DC=tech, CN=Garibaldi\, Omega  \
                    ,OU=T-65,OU=X-Wing,DC=underthewire,DC=tech, CN=Garibai\, Omar  \
                    ,OU=T-65,OU=X-Wing,DC=underthewire,DC=tech...}
Name              : cyborg
ObjectClass       : group
ObjectGUID        : e9511d2f-b09b-40ef-a5b2-180e162ee4a7
SamAccountName    : cyborg
SID               : S-1-5-21-758131494-606461608-3556270690-2180
```

- Doing this would lead to the wrong answers since we can't see how many members are left after the fourth one (notice there are the three dots which indicate there are more)
- Found this simple, one-line cmdlet that actually solves the problem



```
PS C:\users\cyborg3\desktop> get-adgroupmember "cyborg" | measure-object | select count

Count
-----
   88
```

- This also works: *source*



```
PS C:\users\cyborg3\desktop> $info = get-adgroup -identity 'cyborg' -properties members
PS C:\users\cyborg3\desktop> $info.members.count
88
PS C:\users\cyborg3\desktop> 
```

Password: 88_objects

Q&A

- What is an AD module? Module provider?

Cyborg 5

The password for cyborg5 is the PowerShell module name with a version number of 8.9.8.9 **PLUS** the name of the file on the desktop.

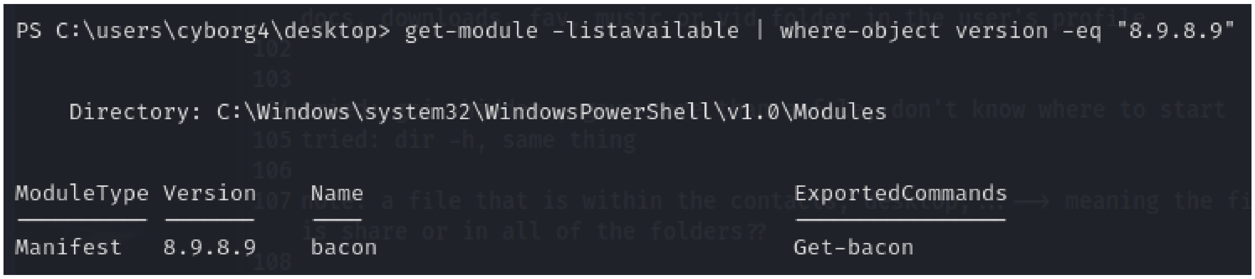
NOTE:
– If the module name is “bob” and the file on the desktop is called “_settings”, then the password is “bob_settings”.
– The password will be lowercase no matter how it appears on the screen.

▼**HINT:**

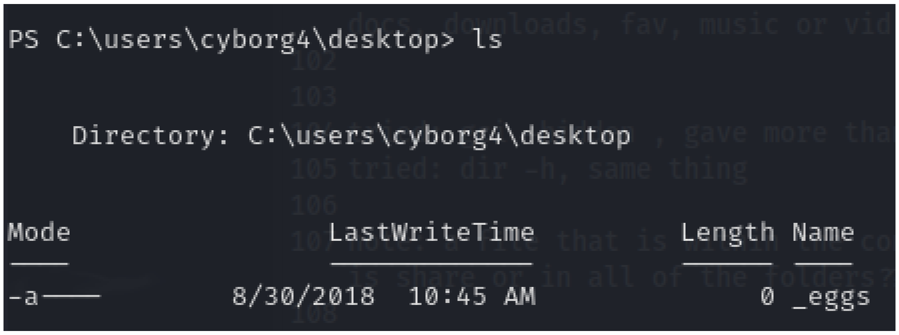
List the modules…

Solution

Get-module -listavailable : see all modules installed on the system



```
PS C:\users\cyborg4\desktop> get-module -listavailable | where-object version -eq "8.9.8.9"


    Directory: C:\Windows\system32\WindowsPowerShell\v1.0\Modules


ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   8.9.8.9    bacon                               Get-bacon
```

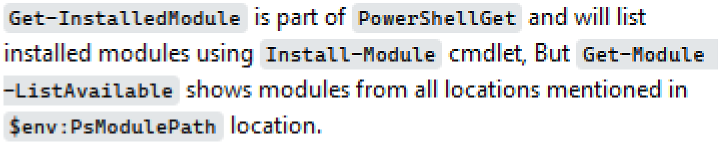


```
PS C:\users\cyborg4\desktop> ls


    Directory: C:\users\cyborg4\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018  10:45 AM              0 _eggs
```

Password: bacon_eggs

Q&A

- Get-installedmodule vs. get-module -listavailable? answer



> `Get-InstalledModule` is part of `PowerShellGet` and will list installed modules using `Install-Module` cmdlet, But `Get-Module -ListAvailable` shows modules from all locations mentioned in `$env:PsModulePath` location.

Cyborg 6

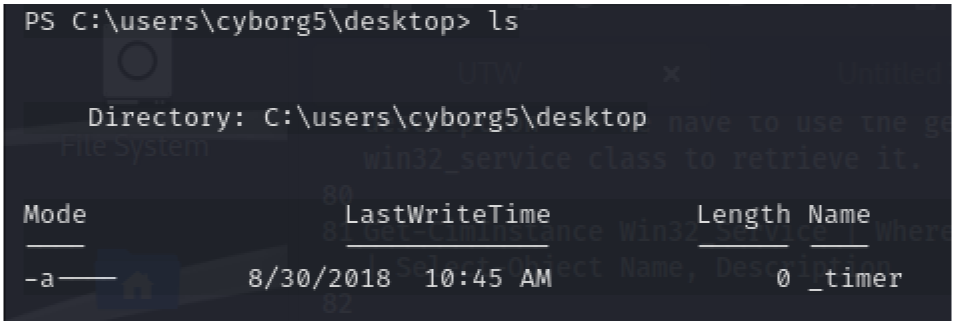
The password for cyborg6 is the last name of the user who has logon hours set on their account **PLUS** the name of the file on the desktop.

NOTE:
– If the last name is “fields” and the file on the desktop is called “_address”, then the password is “fields_address”.
– The password will be lowercase no matter how it appears on the screen.

▼ HINT:

https://technet.microsoft.com/en-us/library/ee617195.aspx

Solution



```
PS C:\users\cyborg5\desktop> ls


    Directory: C:\users\cyborg5\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018  10:45 AM              0 _timer
```

-filter: to filter the list of user accounts by one or more attributes. We can combine conditions using the logical powershell comparison operators like -and, -or, -not and comparison operators like -eq,-ne,-gt,-ge,-lt,-le,-like,-notlike, etc
Additionally, we can sort the resulting list of users with the sort-object cmdlet or the where-object cmdlet to specify multiple criteria at once

I just wanted to know what properties a user has

Get-aduser -filter * -properties logonhours | ft name, logonhours

Didn’t see the attribute logonhours when I tried to list out all properties of a user. This needs to be studied more!!

Password; rowray_timer

Q&A

- How to view all properties of a user object of an AD?????????????????????????????????????

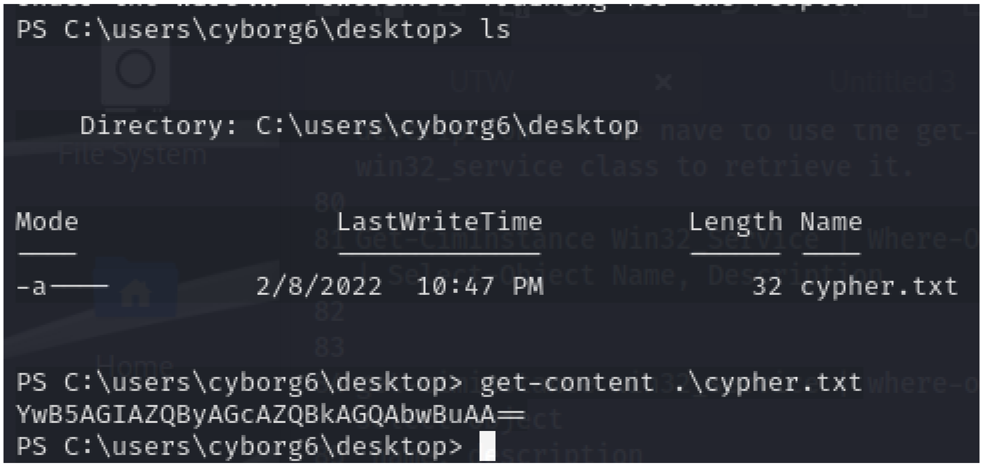
Cyborg 7

The password for cyborg7 is the decoded text of the string within the file on the desktop.

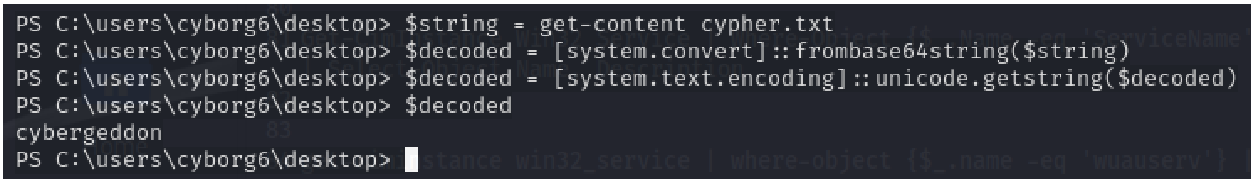
NOTE:
– The password is the last word of the string. For example, if it is “I like PowerShell”, the password would be “powershell”.
– The password will be lowercase no matter how it appears on the screen.
– There are no spaces in the answer.

▼ HINT:

PowerShell has access to the .Net Framework which can convert text encoding between formats. Find the right system call and you will be able to convert text strings.



```
PS C:\users\cyborg6\desktop> ls


    Directory: C:\users\cyborg6\desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----          2/8/2022  10:47 PM             32 cypher.txt


PS C:\users\cyborg6\desktop> get-content .\cypher.txt
YwB5AGIAZQByAGcAZQBkAGQAbwBuAA==
PS C:\users\cyborg6\desktop>
```

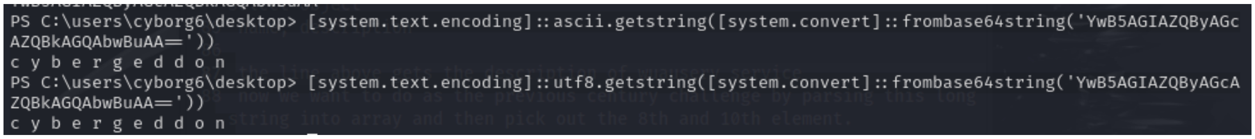


```
PS C:\users\cyborg6\desktop> $string = get-content cypher.txt
PS C:\users\cyborg6\desktop> $decoded = [system.convert]::frombase64string($string)
PS C:\users\cyborg6\desktop> $decoded = [system.text.encoding]::unicode.getstring($decoded)
PS C:\users\cyborg6\desktop> $decoded
cybergeddon
PS C:\users\cyborg6\desktop>
```

Password: cybergeddon

I tried to decode the text through cyberchef but I tested out all the rules from base64 and still can’t decode the text for some reason. !!!

An alternative:



```
PS C:\users\cyborg6\desktop> [system.text.encoding]::ascii.getstring([system.convert]::frombase64string('YwB5AGIAZQByAGc
AZQBkAGQAbwBuAA=='))
c y b e r g e d d o n
PS C:\users\cyborg6\desktop> [system.text.encoding]::utf8.getstring([system.convert]::frombase64string('YwB5AGIAZQByAGcA
ZQBkAGQAbwBuAA=='))
c y b e r g e d d o n
```

Q&A

- How to know that this is base64 code??????????? That is the biggest question – I guess because base64 is one of the most used bases to encode binary data as text.
    - There are other methods to encode/decode strings in powershell like URLand HTML, but the content of the file definitely doesn't look like a URL or a HTML script.
-
- What is .NET framework??????????????????????/
-

To encode and decode strings in PowerShell, use Base64 encoding by converting the string to bytes with [System.Text.Encoding]::UTF8.GetBytes($string) and then encoding with [Convert]::ToBase64String($bytes). Decode by using [Convert]::FromBase64String($base64String) and converting back to a string with [System.Text.Encoding]::UTF8.GetString($bytes).

Cyborg 8

The password for cyborg8 is the executable name of a program that will start automatically when cyborg7 logs in.

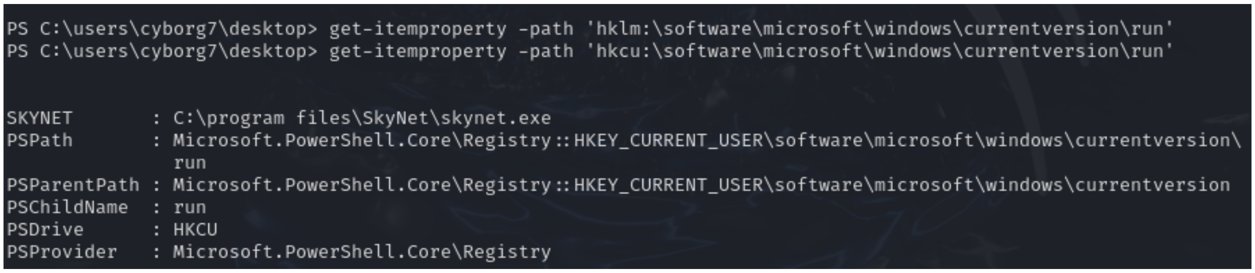
NOTE:

– The password will be lowercase no matter how it appears on the screen.

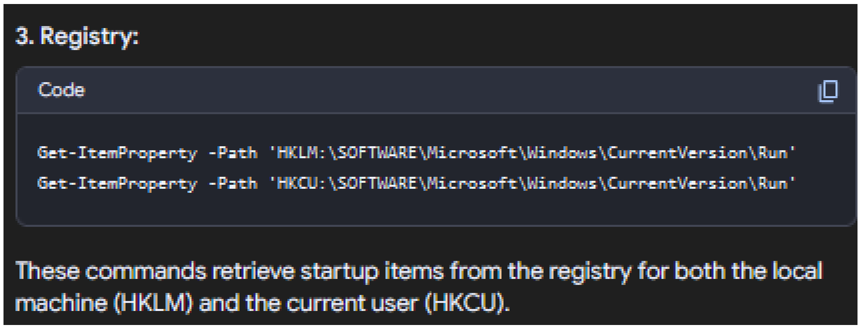
▼ **HINT:**

The Run key in the registry seems like a good place to look…

answer



```
PS C:\users\cyborg7\desktop> get-itemproperty -path 'hklm:\software\microsoft\windows\currentversion\run'
PS C:\users\cyborg7\desktop> get-itemproperty -path 'hkcu:\software\microsoft\windows\currentversion\run'


SKYNET       : C:\program files\SkyNet\skynet.exe
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\software\microsoft\windows\currentversion\
               run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\software\microsoft\windows\currentversion
PSChildName  : run
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```



3. Registry:

```
Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run'
Get-ItemProperty -Path 'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run'
```

These commands retrieve startup items from the registry for both the local machine (HKLM) and the current user (HKCU).

Password: skynet

Q&A

- What is windows registry? How does it work?

Cyborg 9

The password for cyborg9 is the Internet zone that the picture on the desktop was downloaded from.

NOTE:
– The password will be lowercase no matter how it appears on the screen.

▼ **HINT:**

Alternate NTFS data streams contain valuable information. Get information for the item with appropriate parameters to solve this level.

answer

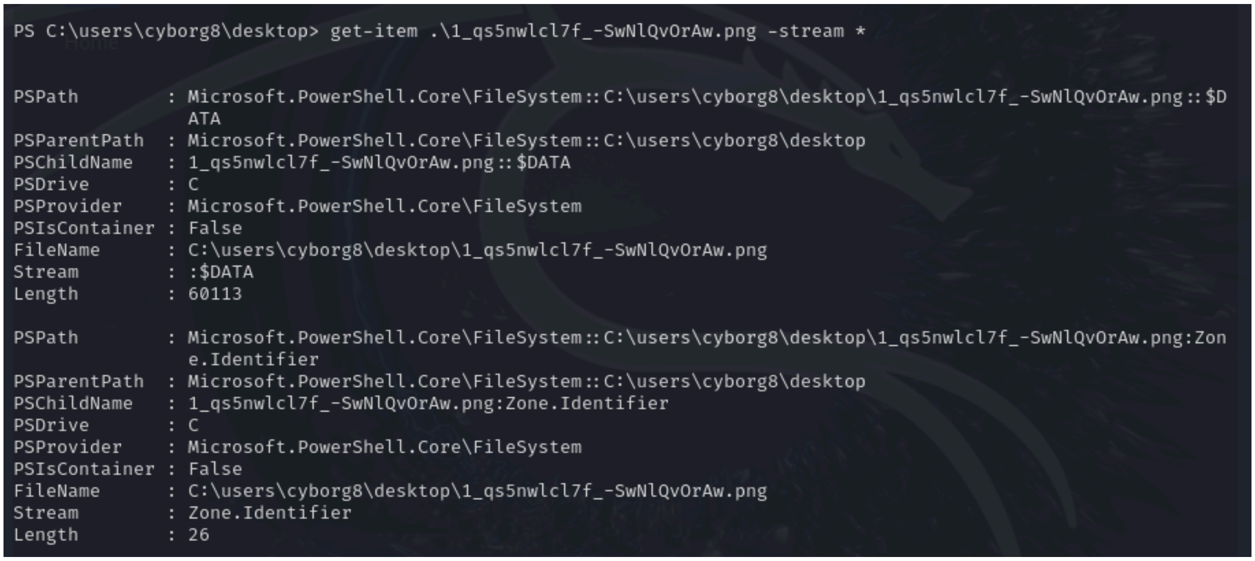
There are a number of ways to access Alternate Data Streams. In powershell, we use the get-Item cmdlet to list all available streams for a specific png file.

Get-item [filename] -stream *

If we look at the output (below), we can see two streams, the unmanned data stream and a stream with the name zone.identifier. We will find out what this alternate stream contains.

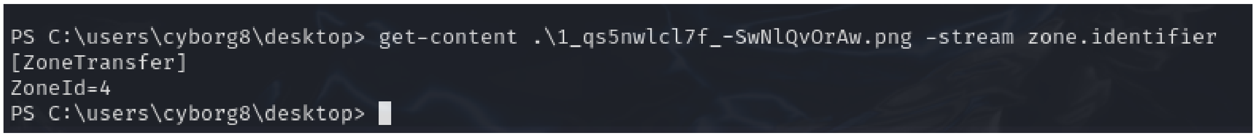
The zone.identifier stream was first designed as a security feature and provided storage for URL security zone information. It allows Windows to determine whether a file should be trusted or not.

How did it get attached to the .png file? Internet Explorer added a zone.identifier stream to all downloaded files and set an ID that indicated which Zone the file originated from (such as Zone 3, the Internet Zone).



```
PS C:\users\cyborg8\desktop> get-item .\1_qs5nwlcl7f_-SwNlQvOrAw.png -stream *


PSPath        : Microsoft.PowerShell.Core\FileSystem::C:\users\cyborg8\desktop\1_qs5nwlcl7f_-SwNlQvOrAw.png::$D
                ATA
PSParentPath  : Microsoft.PowerShell.Core\FileSystem::C:\users\cyborg8\desktop
PSChildName   : 1_qs5nwlcl7f_-SwNlQvOrAw.png::$DATA
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\users\cyborg8\desktop\1_qs5nwlcl7f_-SwNlQvOrAw.png
Stream        : :$DATA
Length        : 60113

PSPath        : Microsoft.PowerShell.Core\FileSystem::C:\users\cyborg8\desktop\1_qs5nwlcl7f_-SwNlQvOrAw.png:Zon
                e.Identifier
PSParentPath  : Microsoft.PowerShell.Core\FileSystem::C:\users\cyborg8\desktop
PSChildName   : 1_qs5nwlcl7f_-SwNlQvOrAw.png:Zone.Identifier
PSDrive       : C
PSProvider    : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName      : C:\users\cyborg8\desktop\1_qs5nwlcl7f_-SwNlQvOrAw.png
Stream        : Zone.Identifier
Length        : 26
```

We will find out what is stored in the Zone.Identifier by running the get-content cmdlet:
Get-content [filename] -stream zone.identifier



```
PS C:\users\cyborg8\desktop> get-content .\1_qs5nwlcl7f_-SwNlQvOrAw.png -stream zone.identifier
[ZoneTransfer]
ZoneId=4
PS C:\users\cyborg8\desktop> █
```

Password: 4

Q&A

- What is internet zone?
- What is ntfs alternate data stream?
    - NTFS: a file system. Files stored on an NTFS system have many attribute, one of these is $DATA
- ADS threat? Why is an alternate data stream a security vulnerability?
- forensic analysis with zone.identifier

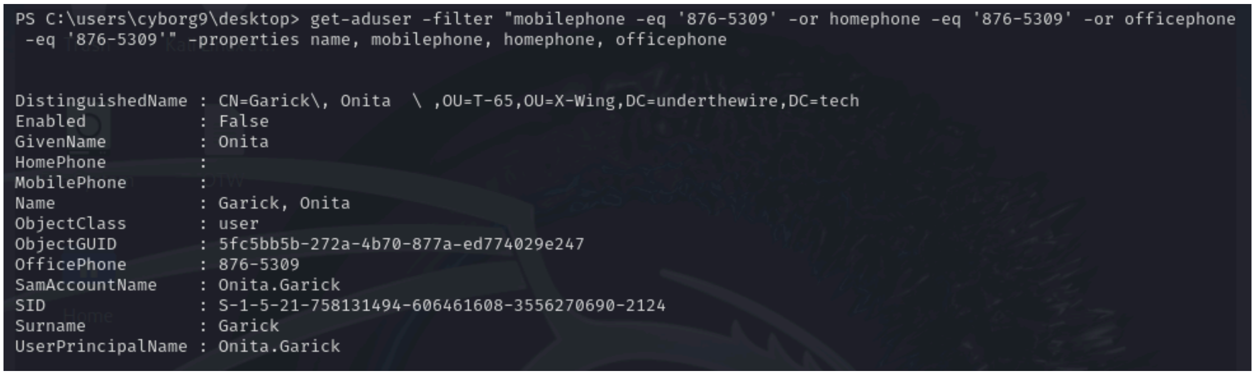
Cyborg 10

The password for cyborg10 is the first name of the user with the phone number of 876-5309 listed in Active Directory **PLUS** the name of the file on the desktop.

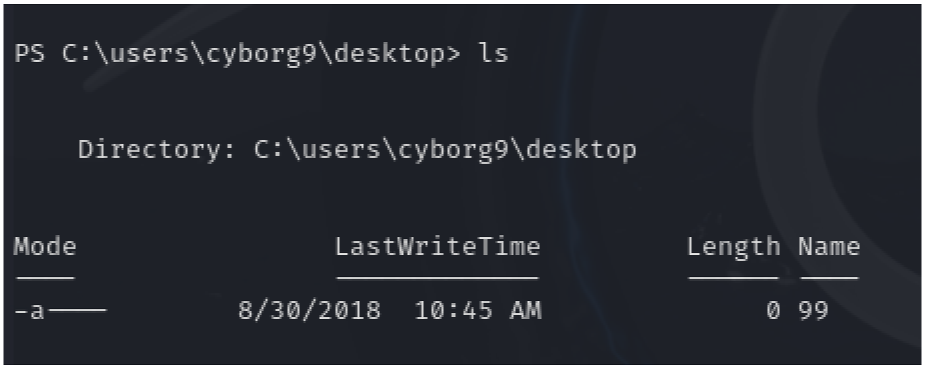
NOTE:
– If the first name “chris” and the file on the desktop is called “23”, then the password is “chris23”.
– The password will be lowercase no matter how it appears on the screen.

▼ HINT:

https://learn.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2016-ps



```
PS C:\users\cyborg9\desktop> get-aduser -filter "mobilephone -eq '876-5309' -or homephone -eq '876-5309' -or officephone
 -eq '876-5309'" -properties name, mobilephone, homephone, officephone


DistinguishedName : CN=Garick\, Onita  \ ,OU=T-65,OU=X-Wing,DC=underthewire,DC=tech
Enabled           : False
GivenName         : Onita
HomePhone         :
MobilePhone       :
Name              : Garick, Onita
ObjectClass       : user
ObjectGUID        : 5fc5bb5b-272a-4b70-877a-ed774029e247
OfficePhone       : 876-5309
SamAccountName    : Onita.Garick
SID               : S-1-5-21-758131494-606461608-3556270690-2124
Surname           : Garick
UserPrincipalName : Onita.Garick
```



```
PS C:\users\cyborg9\desktop> ls


    Directory: C:\users\cyborg9\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018  10:45 AM              0 99
```

Password: onita99

Q&A

Is there a way to combine the 3 conditions mobilephone, homephone, officephone -eq ‘876-5309’ into one????????????????????????????????????

Cyborg 11 – App Locker

The password for cyborg11 is the description of the Applocker Executable deny policy for ill_be_back.exe PLUS the name of the file on the desktop.

NOTE:

– If the description is “green$” and the file on the desktop is called “28”, then the password is “green$28”.

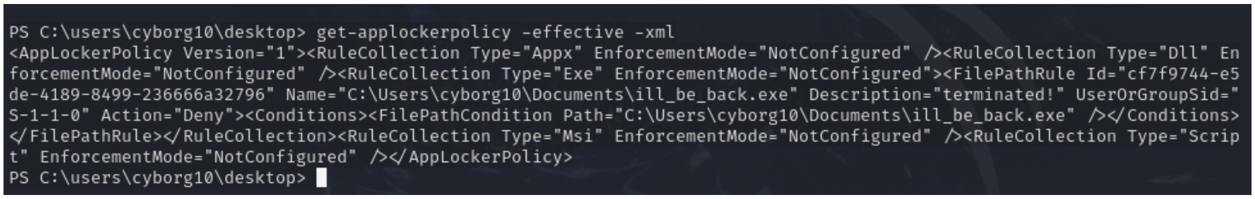
– The password will be lowercase no matter how it appears on the screen.

▼ HINT:

Powershell is a great applockerpolicy tool just go GET it.

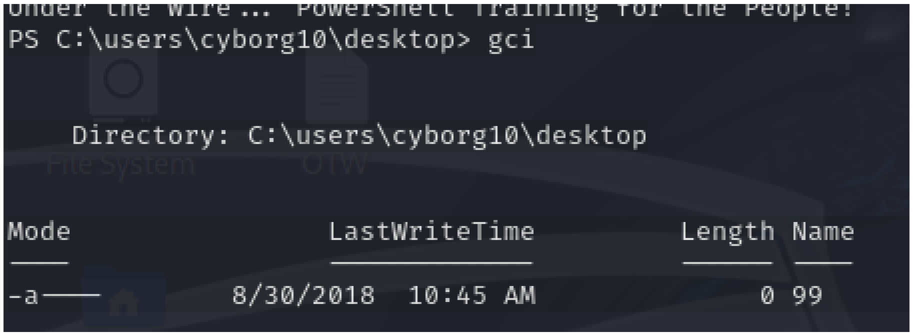
Q&A

- What is AppLocker?
    - When a user runs a process, that process has the same level of access to data that the user has. AppLocker restricts the files/processes that users or groups are allowed to run to avoid sensitive data accessed by unauthorized software.
    - Although we have AD, it only controls what users are allowed to access. AppLocker helps control which apps and files users can run – including executable files, scripts, windows installer files, dynamic-link libraries DLLS, packaged apps, and packaged app installers. (official doc)



```
PS C:\users\cyborg10\desktop> get-applockerpolicy -effective -xml
<AppLockerPolicy Version="1"><RuleCollection Type="Appx" EnforcementMode="NotConfigured" /><RuleCollection Type="Dll" En
forcementMode="NotConfigured" /><RuleCollection Type="Exe" EnforcementMode="NotConfigured"><FilePathRule Id="cf7f9744-e5
de-4189-8499-236666a32796" Name="C:\Users\cyborg10\Documents\ill_be_back.exe" Description="terminated!" UserOrGroupSid="
S-1-1-0" Action="Deny"><Conditions><FilePathCondition Path="C:\Users\cyborg10\Documents\ill_be_back.exe" /></Conditions>
</FilePathRule></RuleCollection><RuleCollection Type="Msi" EnforcementMode="NotConfigured" /><RuleCollection Type="Scrip
t" EnforcementMode="NotConfigured" /></AppLockerPolicy>
PS C:\users\cyborg10\desktop>
```

-xml parameter specifies that the AppLocker policy be output as an XML-formatted string. Get-applocker cmdlet can get the local, effective, or a domain AppLocker policy via the respective parameters -local, -effective, -domain. (Get-AppLocker official doc)



```
Under the Wire... PowerShell Training for the People!
PS C:\users\cyborg10\desktop> gci


    Directory: C:\users\cyborg10\desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        8/30/2018  10:45 AM              0 99
```

Password: terminated!99

Cyborg 12 – IIS Logs

The password for cyborg12 is located in the IIS log. The password is not Mozilla or Opera.

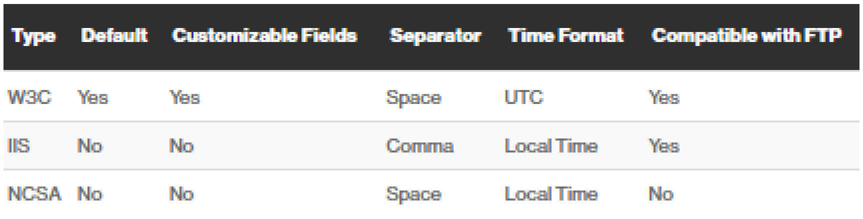
NOTE:
– The password will be lowercase no matter how it appears on the screen.

▼ **HINT:**

A log is just a file, load the content then search what you are looking for or not what you are looking for. Sometimes extra noise is a good thing.

Q&A

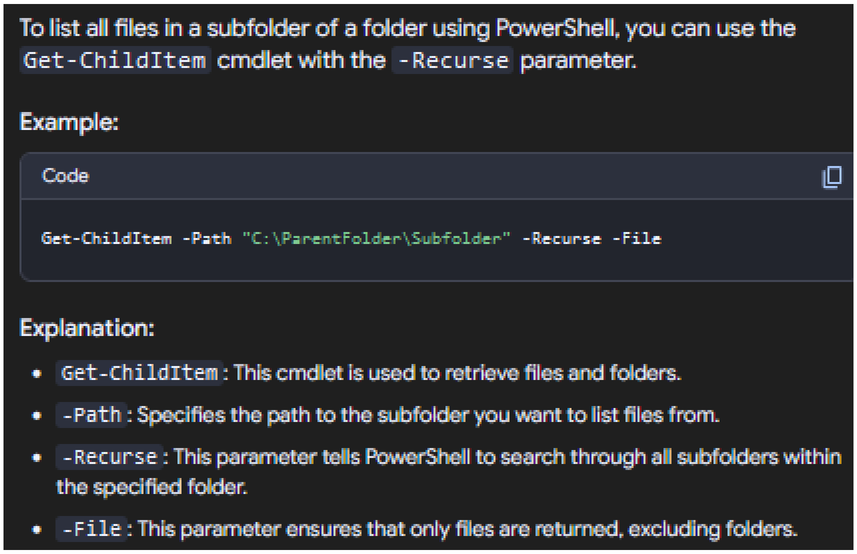
- What is the IIS log? (Internet information services log) (crowdstrike article)
    - IIS logs are for troubleshooting web applications. IIS creates log files in ASCII text that contains like source IP address, web pages accessed, URI queries, http methods, http status codes returned, etc. for each website it serves. Each website has a site ID. There are 3 different log formats – W3C (default, most flexible), IIS (fixed), NCSA (fixed) – because they are different in log event fields, field separators, and the time format; thus, each IIS log file entry can contain different fields. To manage IIS logs, we use a log management system.



Type	Default	Customizable Fields	Separator	Time Format	Compatible with FTP
W3C	Yes	Yes	Space	UTC	Yes
IIS	No	No	Comma	Local Time	Yes
NCSA	No	No	Space	Local Time	No

    - Where do IIS logs locate?

To get all files in a subfolder of a folder
Gci -path “” -recurse -file



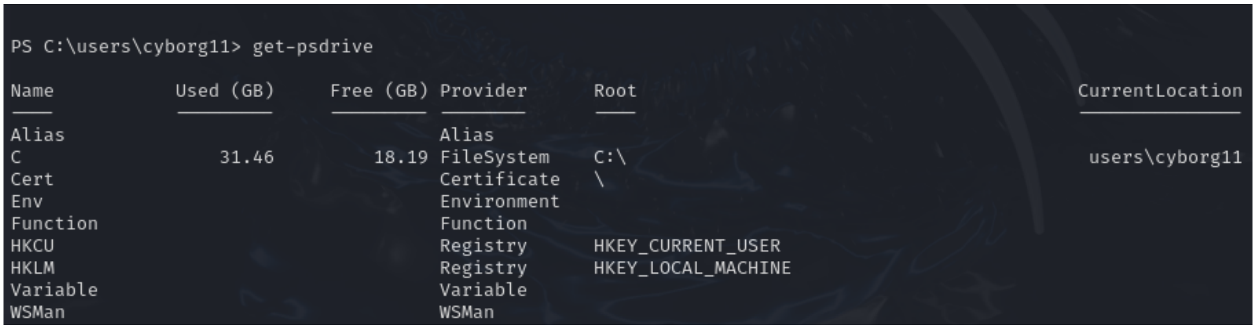
To list all files in a subfolder of a folder using PowerShell, you can use the `Get-ChildItem` cmdlet with the `-Recurse` parameter.

Example:

```
Get-ChildItem -Path "C:\ParentFolder\Subfolder" -Recurse -File
```

Explanation:

- `Get-ChildItem`: This cmdlet is used to retrieve files and folders.
- `-Path`: Specifies the path to the subfolder you want to list files from.
- `-Recurse`: This parameter tells PowerShell to search through all subfolders within the specified folder.
- `-File`: This parameter ensures that only files are returned, excluding folders.



```
PS C:\users\cyborg11> get-psdrive

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
Alias                                  Alias
C                  31.46         18.19 FileSystem    C:\                                                 users\cyborg11
Cert                                   Certificate   \
Env                                    Environment
Function                               Function
HKCU                                   Registry      HKEY_CURRENT_USER
HKLM                                   Registry      HKEY_LOCAL_MACHINE
Variable                               Variable
WSMan                                  WSMan
```

I’m stuck for over an hour.