

Shadow Boxing Or How To Fight The New Windows Security Boundary



Mouad Abouhali
(@_m00dy_)

Agenda

>_ UAC

- >_ *Back to the UAC origins : Pre vista Era*
- >_ *Some internals about UAC mechanism*
 - >_ *Windows Token*
 - >_ *The Split Token model*
- >_ *Why UAC is not a security boundary*
- >_ *Demo: breaking the UAC*

>_ Windows Administrator Protection

- >_ *Windows Administrator Protection: new design*
- >_ *SMAA and its impacts*
 - >_ *WAP elevation flow*
 - >_ *Taxonomy a of a WAP Bypass*

0100010001100101100110011001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

THE PRE-VISTA ERA

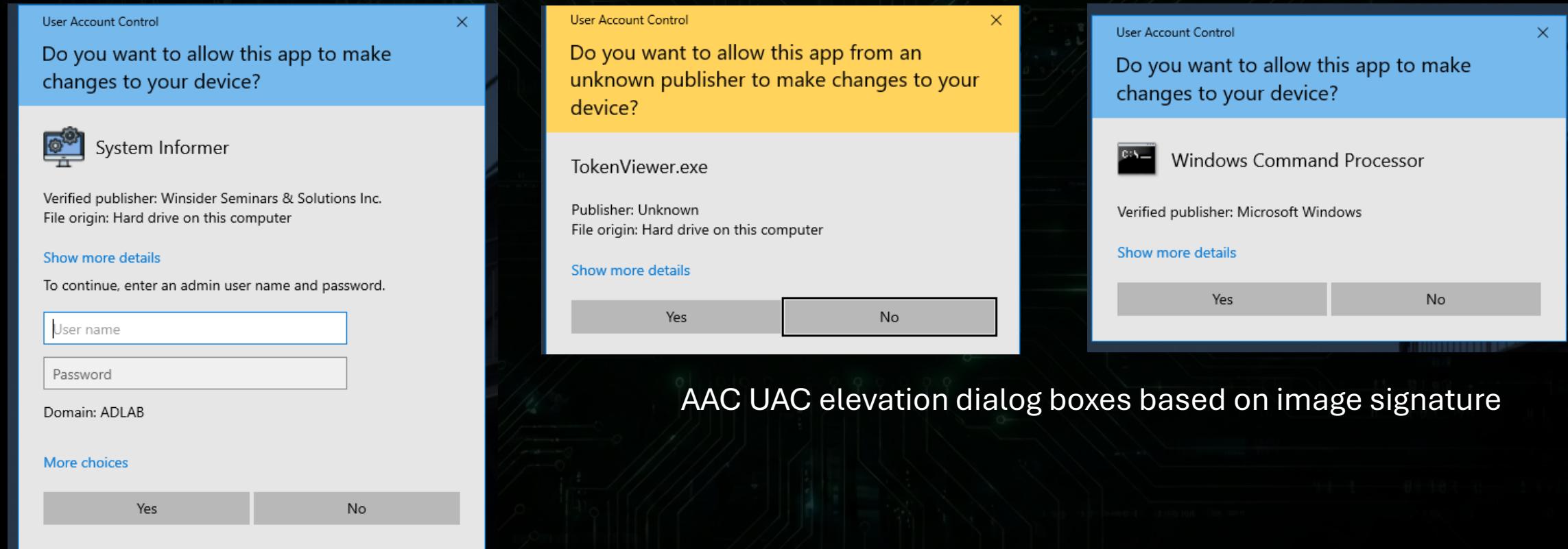


Default High Privileges = High Risk.

0100010001100101011001100011010110111101101110001000000100011101110010011011110111010101110000001000000100001101100001011100110110000101110011000110001101100001

```
#include "Windows UAC"
```

</> UAC Elevation: Mechanisms and Trust Models

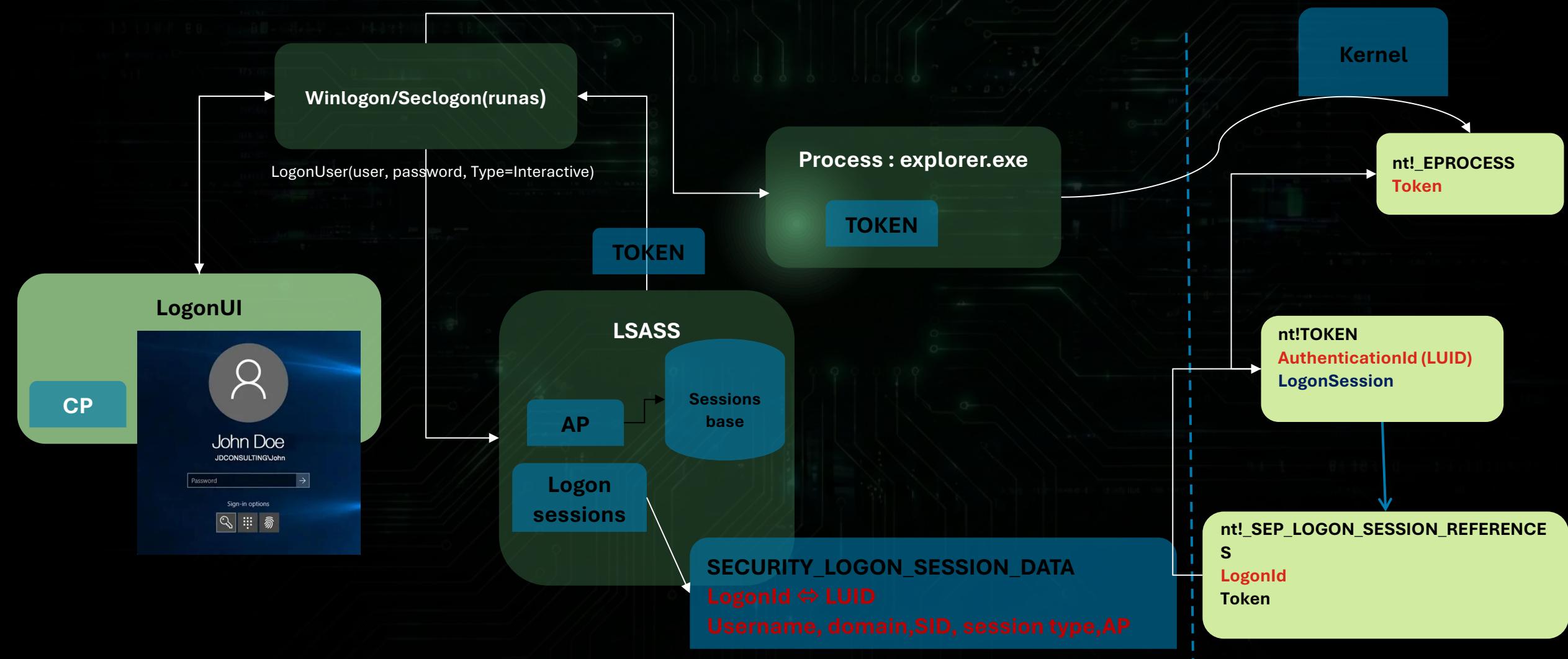


Over the shoulder (OTS) consent dialog box

0100010001100101011001100011010110111101101110001000000100011101110010011011110111010101110000001000000100001101100001011100110110000101110011000110001101100001

Windows Security Model : TOKEN 1/2

</> Windows Tokens : standard authentication process



Windows Security Model : TOKEN 2/2

</> Token = Security identity

cmd.exe:7660 - User DESKTOP-U50EUT1\abouh - TokenId 00000000-1A577409

Main Details	Groups	Privileges	Default Dacl	Misc	Operations	Token Source	Security
<u>Token</u>							
User:	DESKTOP-U50EUT1\abouh						
User SID:	S-1-5-21-3783879229-3626245844-3956583453-1001						
<u>Token Type:</u>	Primary						
Impersonation Level:	N/A						
<u>Token ID:</u>	00000000-1A577409						
Authentication ID:	00000000-01370D7D						
Origin Login ID:	00000000-0000003E7						
Modified ID:	00000000-1A51D851						
<u>Integrity Level:</u>	High			Set Integrity Level			
Session ID:	1						
Elevation Type:	Full						
Is Elevated:	True						
<u>Source</u>							
Name:	User32						
Id:	00000000-01370B51						

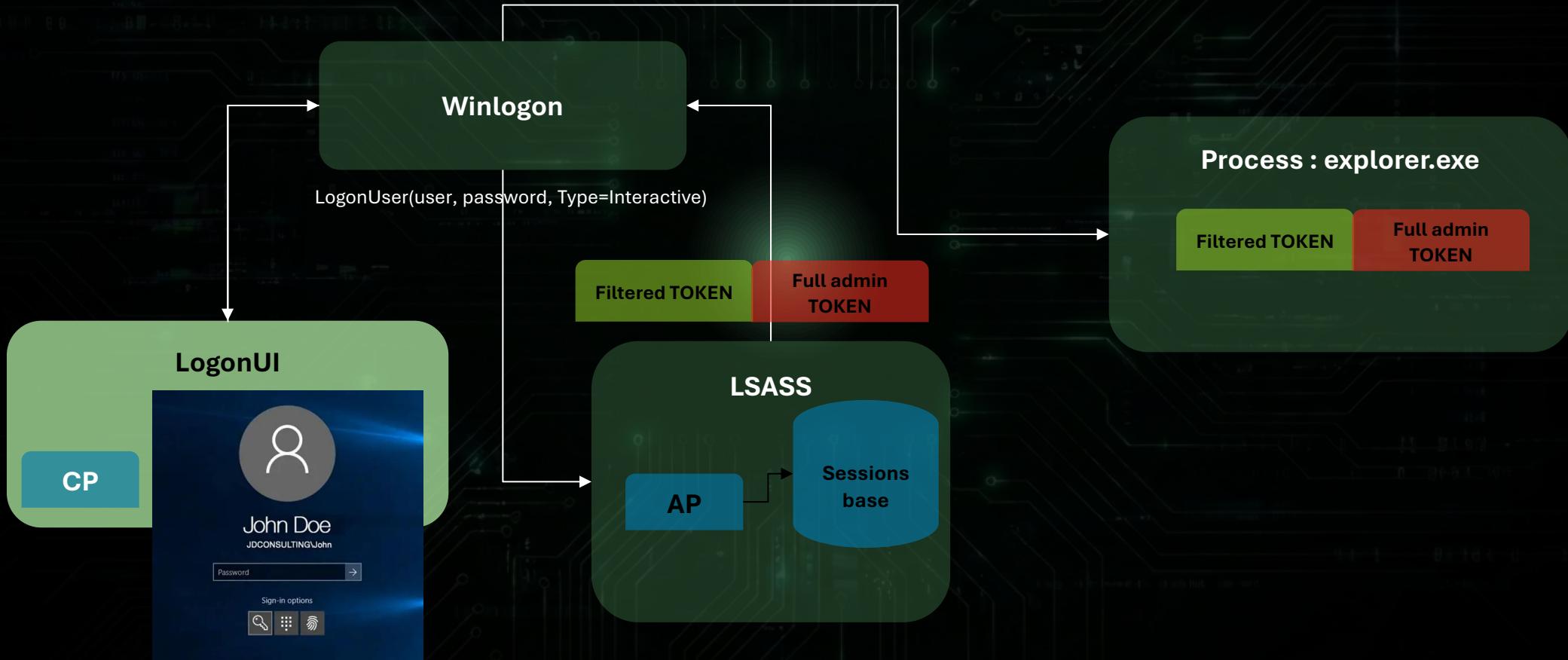
Main Details	Groups	Privileges	Default Dacl	Misc	Operations	Token Source	Security
Name						Flags	
AUTORITE NT\Authentification du compte cloud						Mandatory, Enabled	
AUTORITE NT\Cette organisation						Mandatory, Enabled	
AUTORITE NT\Compte local						Mandatory, Enabled	
AUTORITE NT\Compte local et membre du groupe Administrateurs						Mandatory, Enabled	
AUTORITE NT\INTERACTIF						Mandatory, Enabled	
AUTORITE NT\Utilisateurs authentifiés						Mandatory, Enabled	
BUILTIN\Administrators						Mandatory, Enabled, Owner	
BUILTIN\Performance Log Users						Mandatory, Enabled	
BUILTIN\Users						Mandatory, Enabled	
DESKTOP-U50EUT1\abouh						None	
DESKTOP-U50EUT1\docker-users						Mandatory, Enabled	
DESKTOP-U50EUT1\Netmon Users						Mandatory, Enabled	
LOCAL						Mandatory, Enabled	
MicrosoftAccount\abouh@signin.com						Mandatory, Enabled	
NT AUTHORITY\LogonSessionId_0_20384530						Mandatory, Enabled, LogonId	
OUVERTURE DE SESSION DE CONSOLE						Mandatory, Enabled	
Tout le monde						Mandatory, Enabled	

Main Details		Groups	Privileges	Default Dacl	Misc	Operations	Token Source	Security
Name			Flags					
SeBackupPrivilege			Disabled					
SeChangeNotifyPrivilege			Default Enabled					
SeCreateGlobalPrivilege			Default Enabled					
SeCreatePageFilePrivilege			Disabled					
SeCreateSymbolicLinkPrivilege			Disabled					
SeDebugPrivilege			Disabled					
SeDelegateSessionUserImpersonatePrivilege			Disabled					
SeImpersonatePrivilege			Default Enabled					
SeIncreaseBasePriorityPrivilege			Disabled					
SeIncreaseQuotaPrivilege			Disabled					
SeIncreaseWorkingSetPrivilege			Disabled					
SeLoadDriverPrivilege			Disabled					
SeManageVolumePrivilege			Disabled					
SeProfileSingleProcessPrivilege			Disabled					
SeRemoteShutdownPrivilege			Disabled					
SeRestorePrivilege			Disabled					
SeSecurityPrivilege			Disabled					
SeShutdownPrivilege			Disabled					
SeSystemEnvironmentPrivilege			Disabled					
SeSystemProfilePrivilege			Disabled					
SeSystemTimePrivilege			Disabled					
SeTakeOwnershipPrivilege			Disabled					
SeTimeZonePrivilege			Disabled					
SeUndockPrivilege			Disabled					

0100010001100101100110011001101101110110111000100011101110010011011101110101110000010000110110000101110011011000101100010011011000110000101101110011000110110001

UAC Design and Principles 1/7

</> Windows Tokens : Split-model Token



0100010001100101100110011001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

UAC Design and Principles 2/7

</> The Split-Token Administration model: Full vs Limited

The screenshot shows the Token Viewer application interface. At the top, there's a navigation bar with tabs: Processes, Threads, Handles, Services, Logon User, and Utilities. Below the navigation bar, a search bar displays "cmd.exe" and includes an "Apply Filter" button and a "Show Sandbox Only" checkbox. The main area lists two processes:

Process ID	Name	User	Integrity Level	Restricted	App Container	Command Line
14092	cmd.exe	DESKTOP-U50EUT1\abouh	High	False	False	"C:\WINDOWS\system32\cmd.exe"
14932	cmd.exe	DESKTOP-U50EUT1\abouh	Medium	False	False	"C:\WINDOWS\system32\cmd.exe"

Two windows are displayed side-by-side, each representing one of the cmd.exe processes. Both windows have a title bar showing "cmd.exe:14092 - User DESKTOP-U50EUT1\abouh - Token..." and a close button.

Left Window (Process ID 14092):

Main Details		Groups	Privileges	Default Dacl	Misc	Operations	Token Source	S
Token								
User:	DESKTOP-U50EUT1\abouh							
User SID:	S-1-5-21-3783879229-3626245844-3956583453-1001							
Token Type:	Primary							
Impersonation Level:	N/A							
Token ID:	00000000-291519DF							
Authentication ID:	00000000-01370D7D							
Origin Login ID:	00000000-000003E7							
Modified ID:	00000000-291519D2							
Integrity Level:	High	Set Integrity Level						
Session ID:	1							
Elevation Type:	Full	Linked Token						
Is Elevated:	True							
Source								
Name:	User32							
Id:	00000000-01370B51							

Right Window (Process ID 14932):

Main Details		Groups	Privileges	Default Dacl	Misc	Operations	Token Source	S
Token								
User:	DESKTOP-U50EUT1\abouh							
User SID:	S-1-5-21-3783879229-3626245844-3956583453-1001							
Token Type:	Primary							
Impersonation Level:	N/A							
Token ID:	00000000-1978C224							
Authentication ID:	00000000-01370DBD							
Origin Login ID:	00000000-000003E7							
Modified ID:	00000000-01370DC9							
Integrity Level:	Medium	Set Integrity Level						
Session ID:	1							
Elevation Type:	Limited	Linked Token						
Is Elevated:	False							
Source								
Name:	User32							
Id:	00000000-01370B51							

Annotations on the screenshot:

- A red arrow points from the "Full admin TOKEN" label at the bottom left to the "Elevation Type: Full" field in the left window.
- A green arrow points from the "Filtered TOKEN" label at the bottom right to the "Elevation Type: Limited" field in the right window.
- A green curved arrow originates from the "Full admin TOKEN" label and points to the "cmd.exe:14932" window.

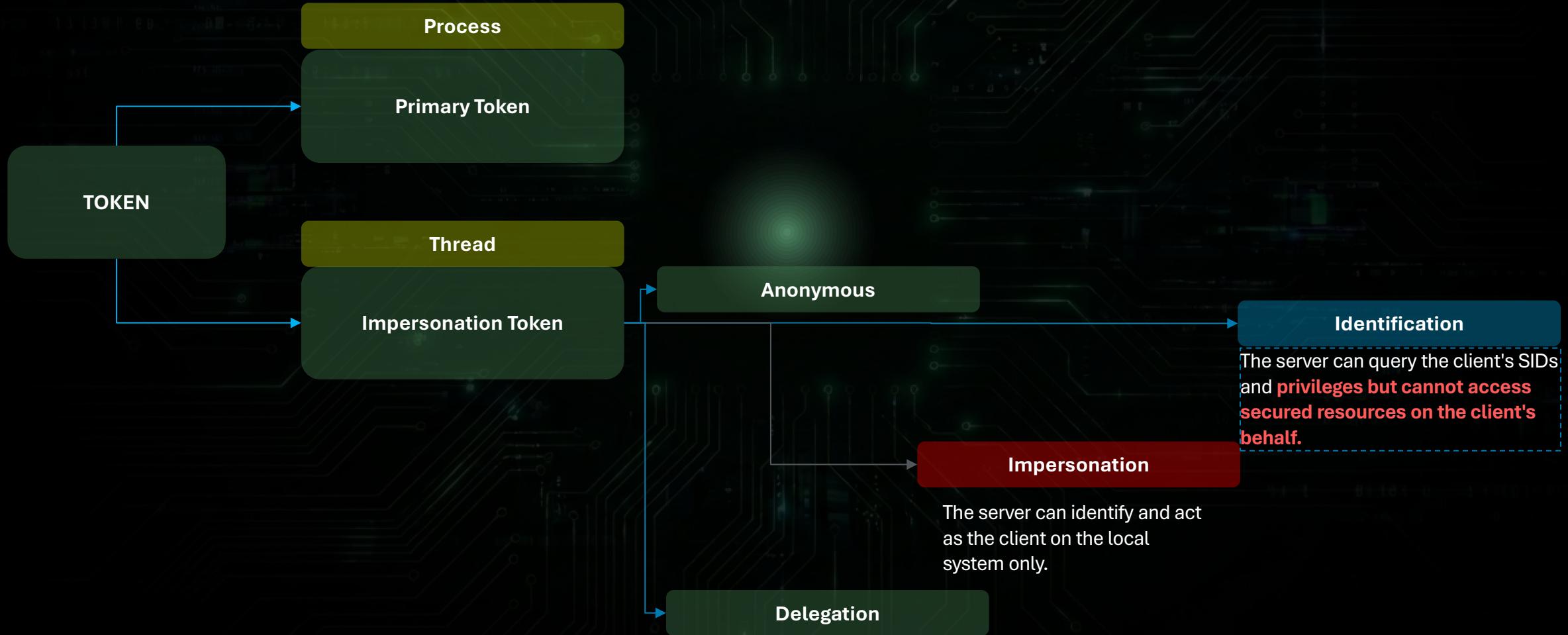
Labels at the bottom:

- Full admin TOKEN (Red box)
- Filtered TOKEN (Green box)

0100010001100101100110011001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

UAC Design and Principles 3/7

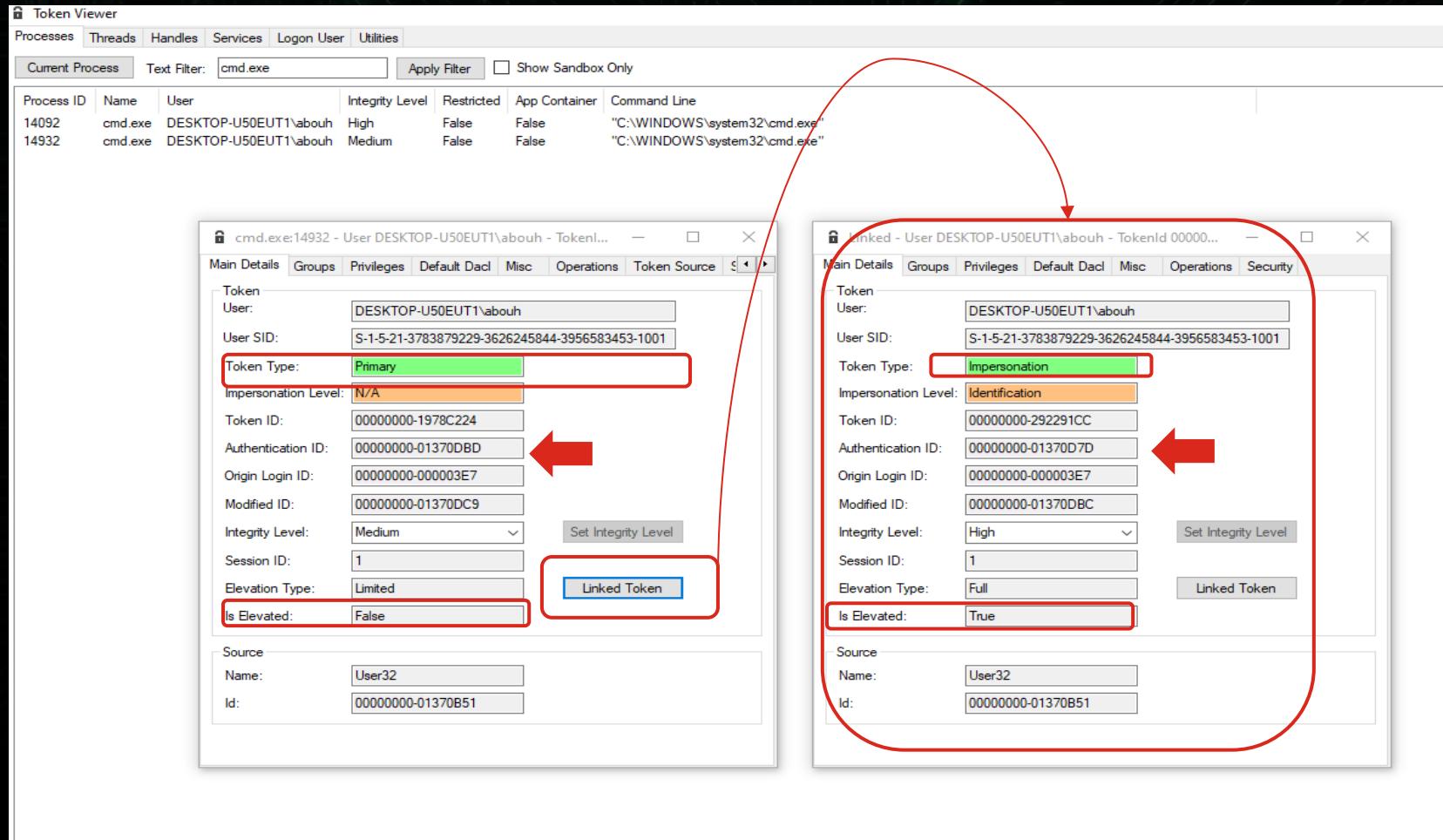
</> Tokens type: Primary vs Impersonation



01000100011001011001100110011011011101101110001000111011100100110111011101011100000100001101100001011100110110000101100010011011000110000101101110011000110110001

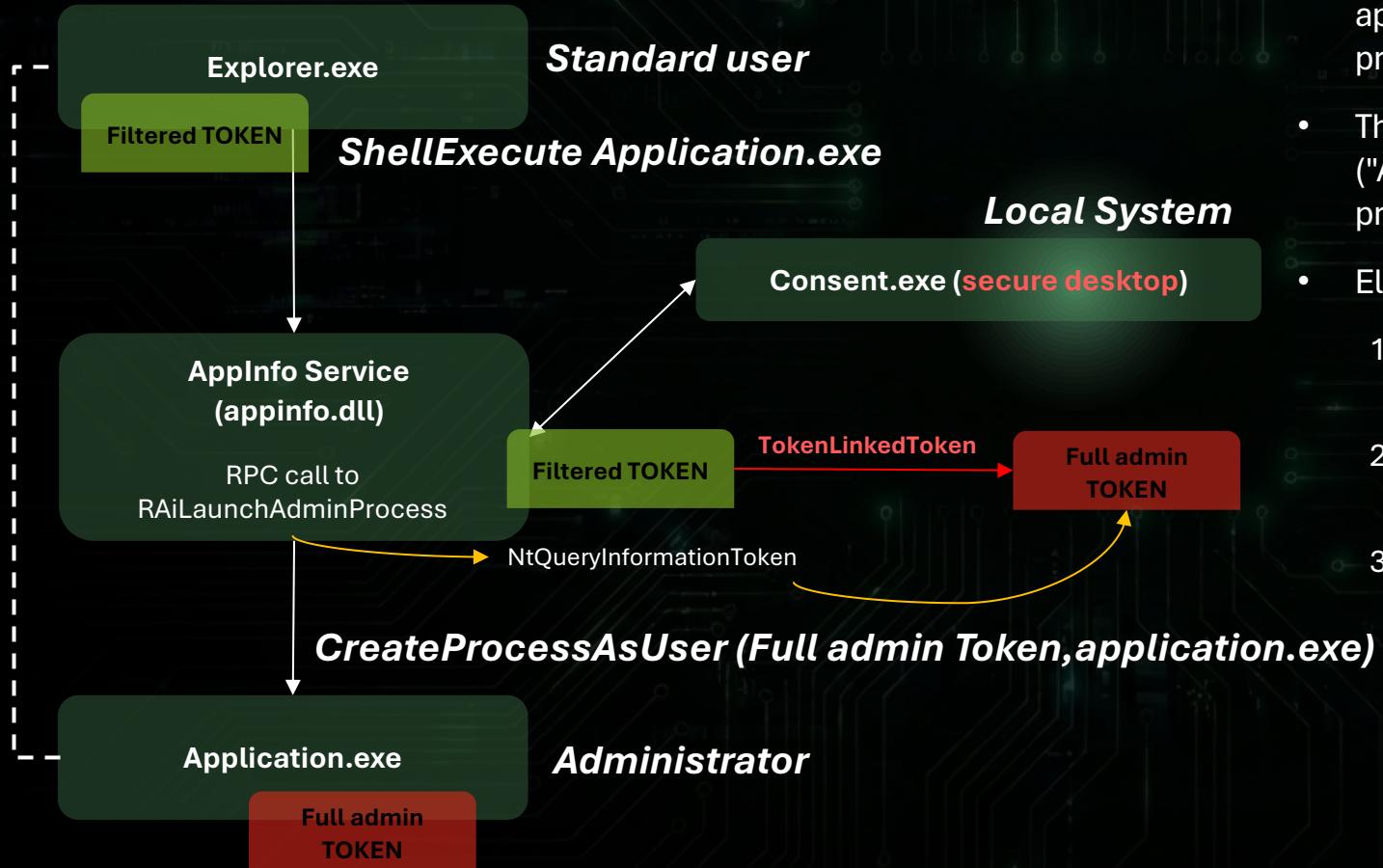
UAC Design and Principles 4/7

</> The Split-Token Administration model: LinkedToken



UAC Design and Principles 5/7

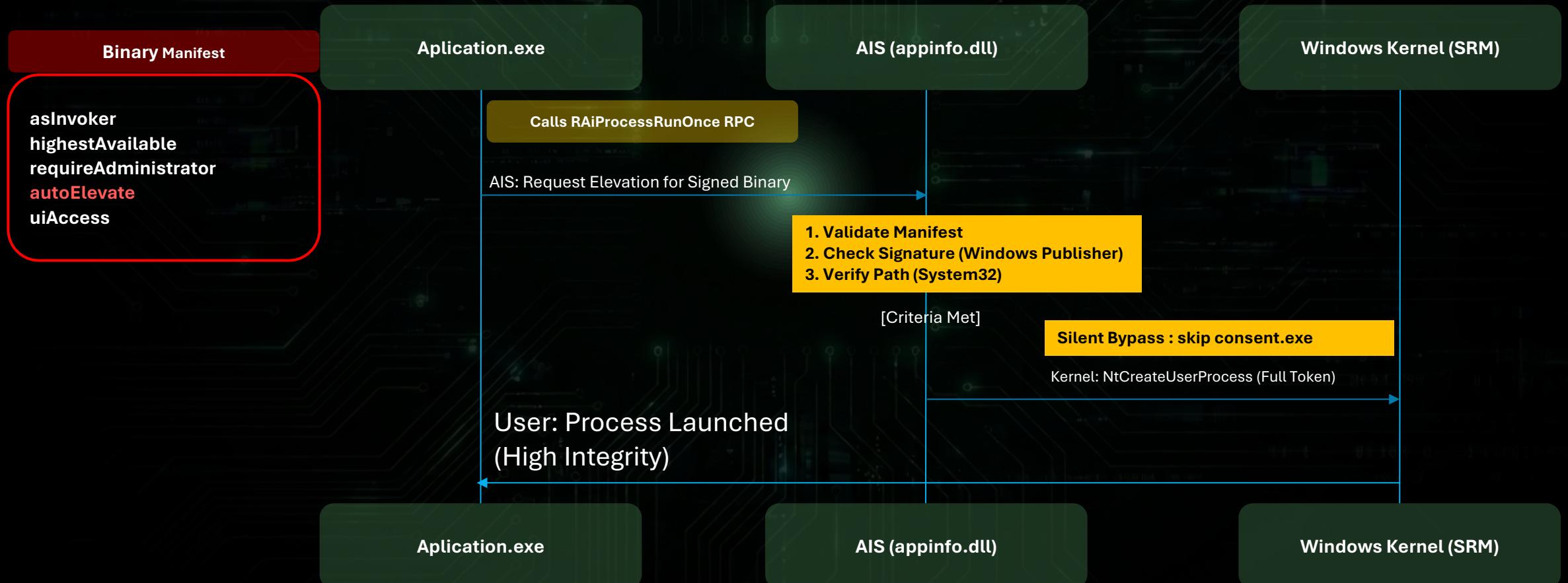
</> UAC Elevation: AIS flow



- The Service: The Application Information Service (AIS / appinfo.dll) runs as SYSTEM inside a svchost.exe process.
 - The "Super Power": AIS possesses the **SeTcbPrivilege** ("Act as part of the operating system"), the most powerful privilege in Windows.
 - Elevation Orchestration:
 1. When elevation is requested, AIS triggers consent.exe on the Secure Desktop.
 2. Upon approval, AIS uses its **SeTcbPrivilege** to "activate" the linked Full token.
 3. AIS then calls **CreateProcessAsUser** to launch the elevated application and "represents" it to the original caller (e.g., Explorer)

UAC Design and Principles 6/7

</> Application Manifests and UAC auto elevation



010001000110010110011001100110110111011011100100000010001110111001001101111011010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

UAC Design and Principles 6/7

</> Application Manifests and UAC elevation

```
C:\SysinternalsSuite>sigcheck.exe -m c:\Windows\System32\taskmgr.exe

Sigcheck v2.90 - File version and signature viewer
Copyright (C) 2004-2022 Mark Russinovich
Sysinternals - www.sysinternals.com

c:\windows\system32\Taskmgr.exe:
    Verified:      Signed
    Signing date: 2:22 PM 8/18/2025
    Publisher:    Microsoft Windows
    Company:     Microsoft Corporation
    Description: Task Manager
    Product:    Microsoft® Windows® Operating System
    Prod version: 10.0.19041.6280
    File version: 10.0.19041.6280 (WinBuild.160101.0800)
    MachineType: 64-bit
    Manifest:
        <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
        <!-- Copyright (c) Microsoft Corporation -->
        <assembly xmlns="urn:schemas-microsoft-com:asm.v1" xmlns:asmv3="urn:schemas-microsoft-com:asm.v3" manifestVersion="1.0">
            <assemblyIdentity
                processorArchitecture="amd64"
                version="5.1.0.0"
                name="Microsoft.Windows.Diagnosis.AdvancedTaskManager"
                type="win32"
            />
            <description>Task Manager</description>
            <dependency>
                <dependentAssembly>
                    <assemblyIdentity
                        type="win32"
                        name="Microsoft.Windows.Common-Controls"
                        version="6.0.0.0"
                        processorArchitecture="amd64"
                        publicKeyToken="6595b64144ccf1df"
                        language="*"
                    />
                </dependentAssembly>
            </dependency>
            <trustInfo xmlns="urn:schemas-microsoft-com:asm.v3">
                <security>
                    <requestedPrivileges>
                        <requestedExecutionLevel
                            level="highestAvailable"
                        />
                    </requestedPrivileges>
                </security>
            </trustInfo>
            <asmv3:application
                <asmv3:windowsSettings xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">
                    <dpiAwareness>true</dpiAwareness>
                    <autoElevate>true</autoElevate>
                </asmv3:windowsSettings>
            </asmv3:application>
        </assembly>
    
```

0100010001100101100110011001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

UAC Design and Principles : the end !

</> The UAC Design Flaw: Shared Resource Environment

Same-Account Model

- Traditional UAC (Admin Approval Mode) runs both limited and elevated processes under the same user account
- Unlike a true security boundary, there is no isolation between the environment of a "Standard" user and an "Administrator"

Common Profile Resources

- Registry Hive: Both tokens share access to HKEY_CURRENT_USER (HKCU)
- File System: Both tokens share the same User Profile directories (%AppData%, %TEMP%, Documents)

The "Pollution" Attack Vector

- A Medium Integrity process can modify configuration files or registry keys that a High Integrity process will later read and trust
- DLL Planting: Malware can place a malicious DLL in a user-writable directory (like %TEMP%) that a system binary might load upon elevation

01000100011001010110011001100011011011110110111100010000000100011101110010011011110101011100000100000010000011011100010111001101100001011000100110110001100001011011100110001101100001

Bypassing UAC for fun and profit !

</> DLL planting demo : SilentCleanup

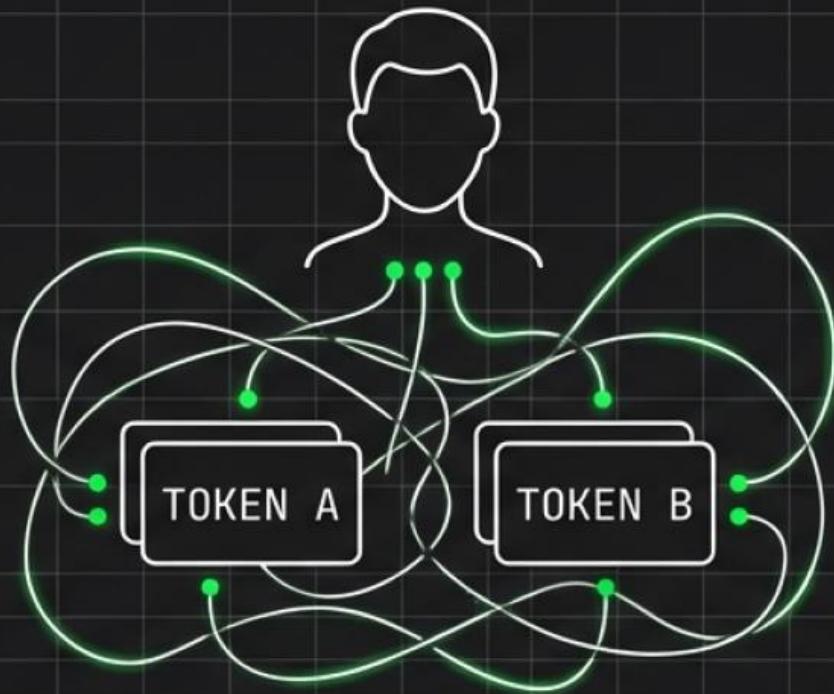
- **Target: The SilentCleanup Scheduled Task:**
 - **Location:** \Microsoft\Windows\DiskCleanup\SilentCleanup.
 - **Privilege Level:** Configured to "Run with highest privileges" (High Integrity).
 - **The "Silent" Advantage:** Because it is a trusted Windows component, it can be triggered by a standard user without a UAC prompt (Auto-elevation).
 - **The Vulnerability:** Path Hijacking & Shared Resources
 - **Shared Environment:** Traditional UAC allows Limited and Elevated tokens to share the same User Profile (Registry HKCU and directories like %LocalAppData%).
 - **Search Order Exploitation:** When triggered, the task's process (e.g., cleanmgr.exe or dism.exe) attempts to load specific DLLs.
 - **Environment Overloading:** Attackers can manipulate environment variables in HKCU or place a malicious DLL in a user-writable directory that the elevated process trusts and searches first.

01000100011000101100110001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

DEMO

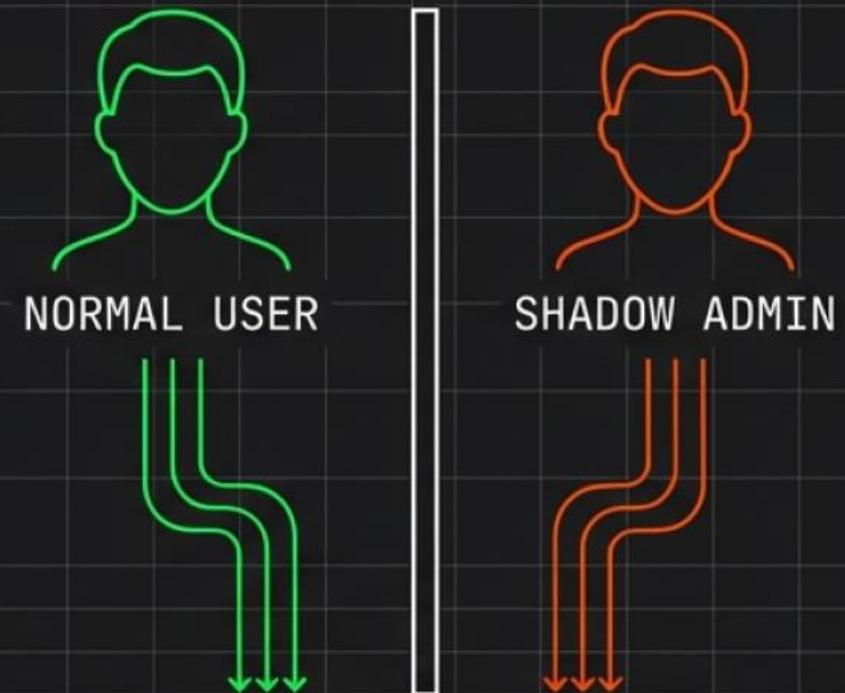
THE PARADIGM SHIFT

UAC (Legacy)



Shared Session

WAP (Modern)



Isolated Session

Windows Administration Protection : new design 1/3

</> Introduction to Windows Administrator Protection (WAP)

- **From Convenience to Security Boundary:**

- From Convenience to Security Boundary:
 - ✓ Traditional UAC is a convenience feature where limited and elevated tokens share the same user account and environment.
 - ✓ WAP transforms elevation into a **hard security boundary**, isolating administrative tasks from the standard user environment.

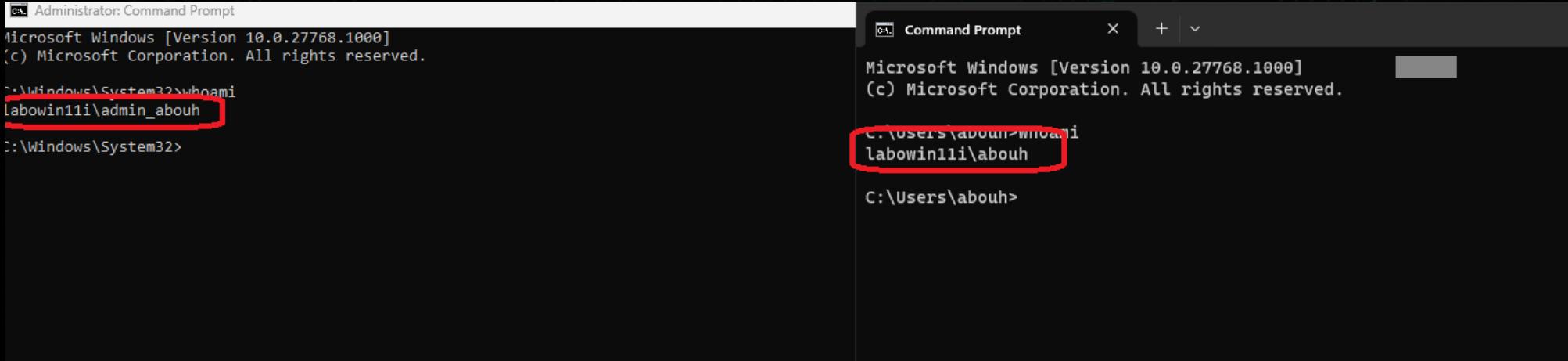
- **Three main principles :**

- Three main principles :
 - The SMAA Mechanism (System Managed Administrator Account)
 - Isolated User Profile
 - Mandatory Windows Hello Confirmation
 - WAP removes "silent" auto-elevations.
 - Every elevation request requires an interactive confirmation via Windows Hello (PIN, Biometrics),

0100010001100101100110011001100110110111011011100100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

Windows Administration Protection : let me duplicate everything!

</> Technical Anatomy of the Shadow Admin Account



```
Administrator: Command Prompt
microsoft Windows [Version 10.0.27768.1000]
(c) Microsoft Corporation. All rights reserved.

>C:\Windows\System32\whoami
labowin11i\admin_abouh
C:\Windows\System32>

Command Prompt
X + ▾
Microsoft Windows [Version 10.0.27768.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abouh>whoami
labowin11i\abouh

C:\Users\abouh>
```

The SMAA Mechanism (System Managed Administrator Account):

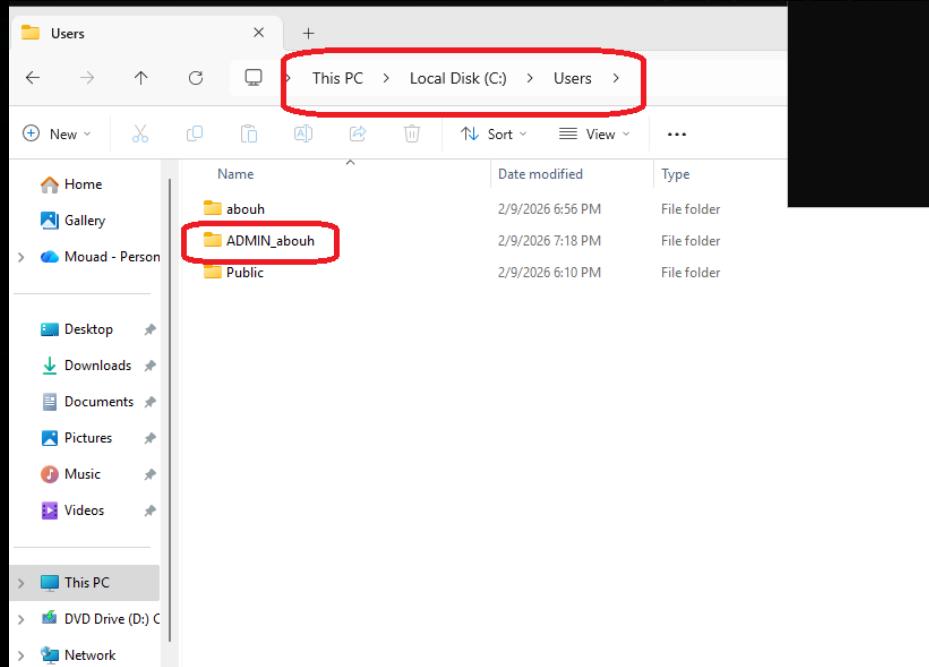
Elevated processes no longer run under the user's identity.

They are executed by a System Managed Administrator Account (SMAA), a dedicated virtual account created for the administrative task.

0100010001100101100110011001100110110111011011100100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

Windows Administration Protection : let me duplicate everything!

</> Technical Anatomy of the Shadow Admin Account



Isolated User Profile:

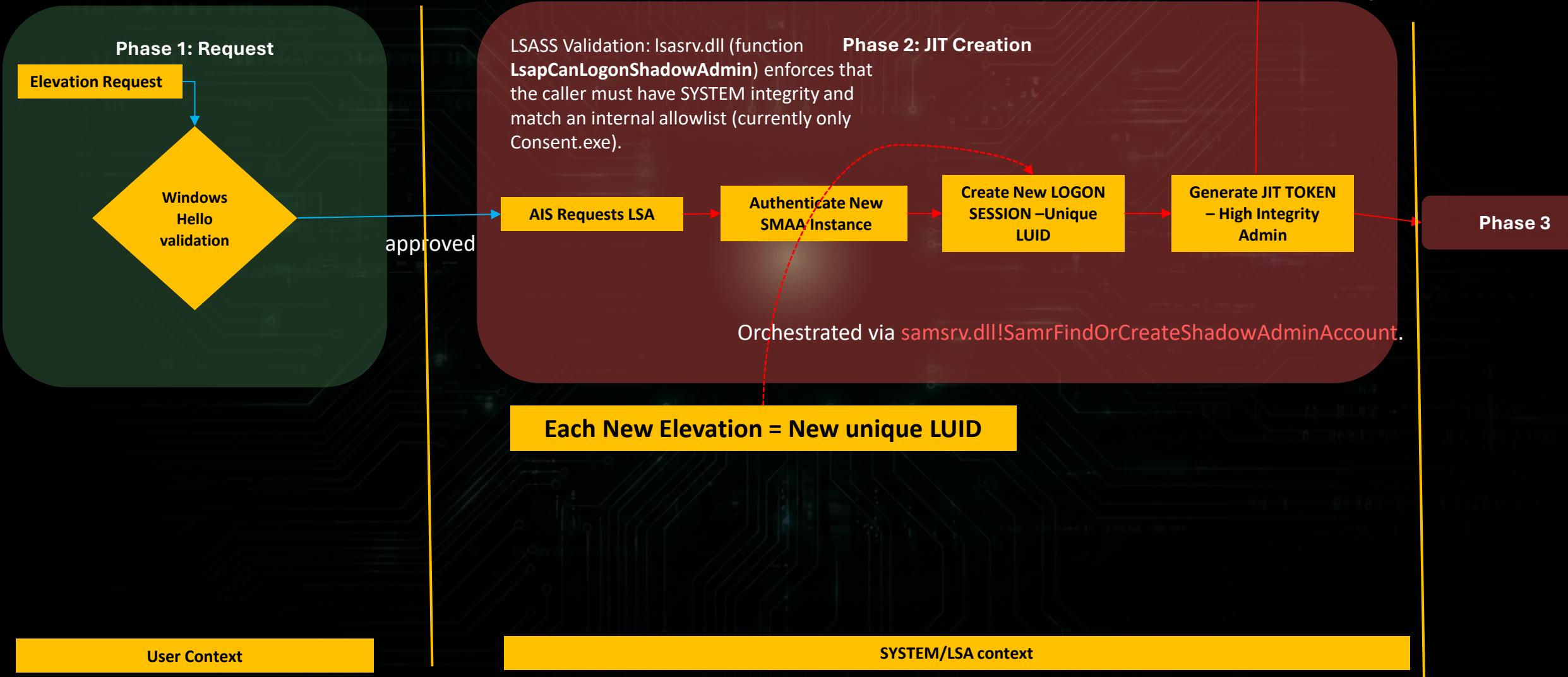
- **Strict Separation:** The SMAA uses an isolated profile (visible as ADMIN_ folders) instead of the standard user profile.
- **No Resource Sharing:** Registry keys (HKCU) and file system paths (AppData, Temp) of the standard user are invisible to the elevated process, preventing "environment pollution" attacks like DLL planting or registry hijacking [History].

0100010001100101100110011001101101110110111000100000010001110111001001101110111010111000001000000100011011000101110011011000101100010011011000110000101101100011011000110000101101100011011000110110001

Windows Administration Protection : new design 1/3

</> Introduction to Windows Administrator Protection (WAP)

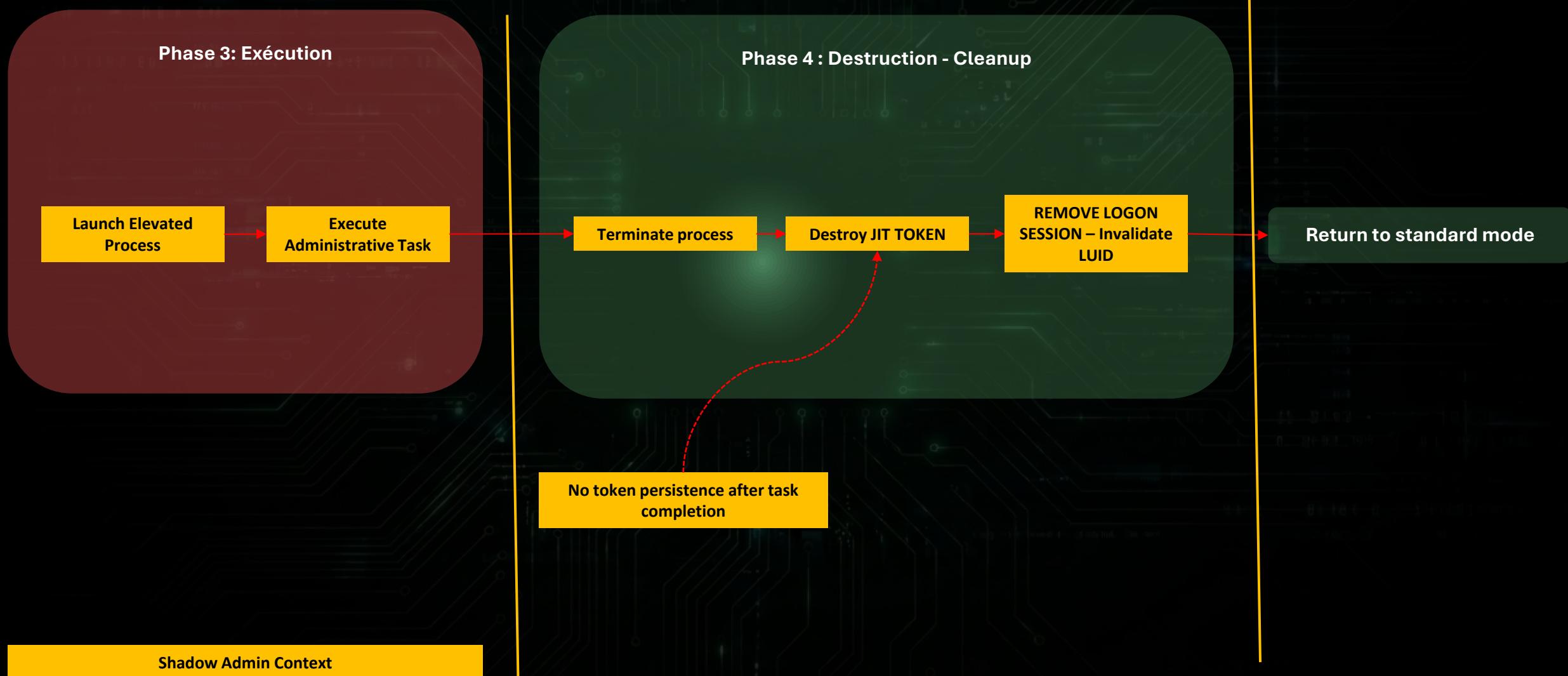
Consent.exe triggers token creation via **LogonUserExExW** using a blank password



01000100011001011001100110011011011101101110001000000100011101110010011011101110101110000010000110110000101110011011000101100010011011000110000101101100011011000110000101101100011011000110110001

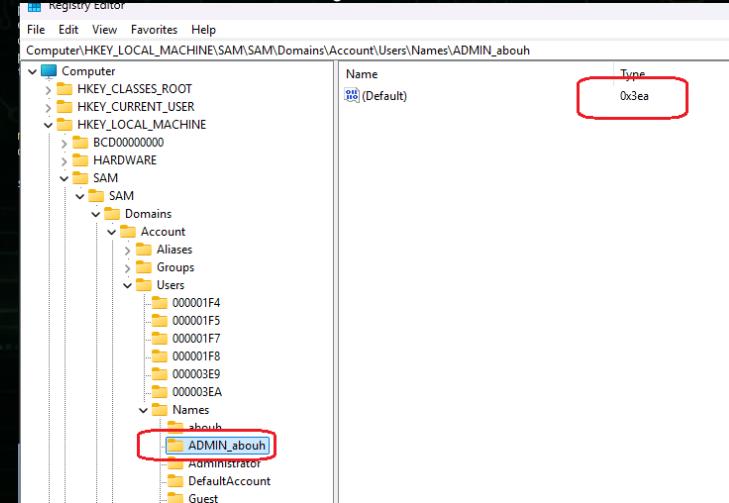
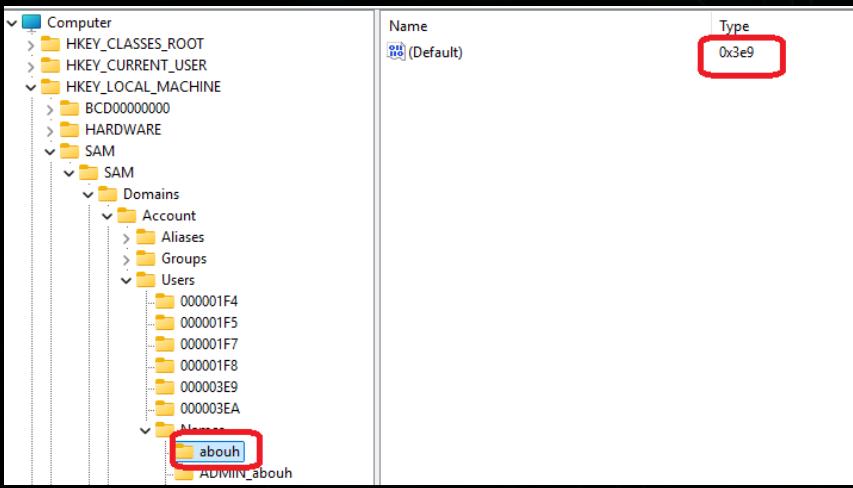
Windows Administration Protection : new design 1/3

</> Introduction to Windows Administrator Protection (WAP)



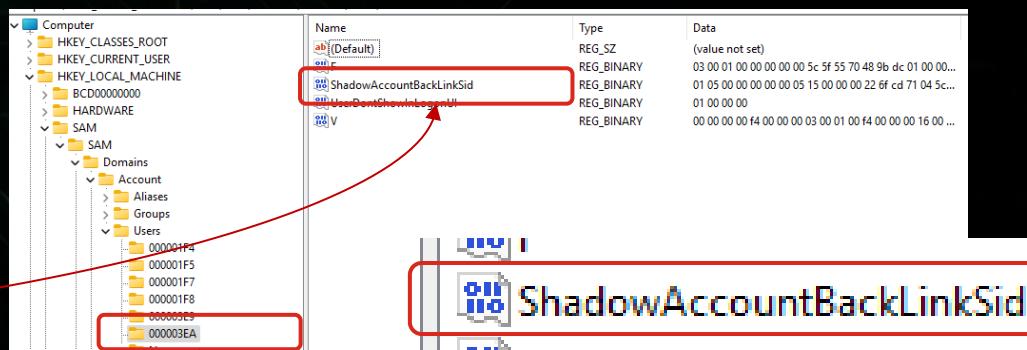
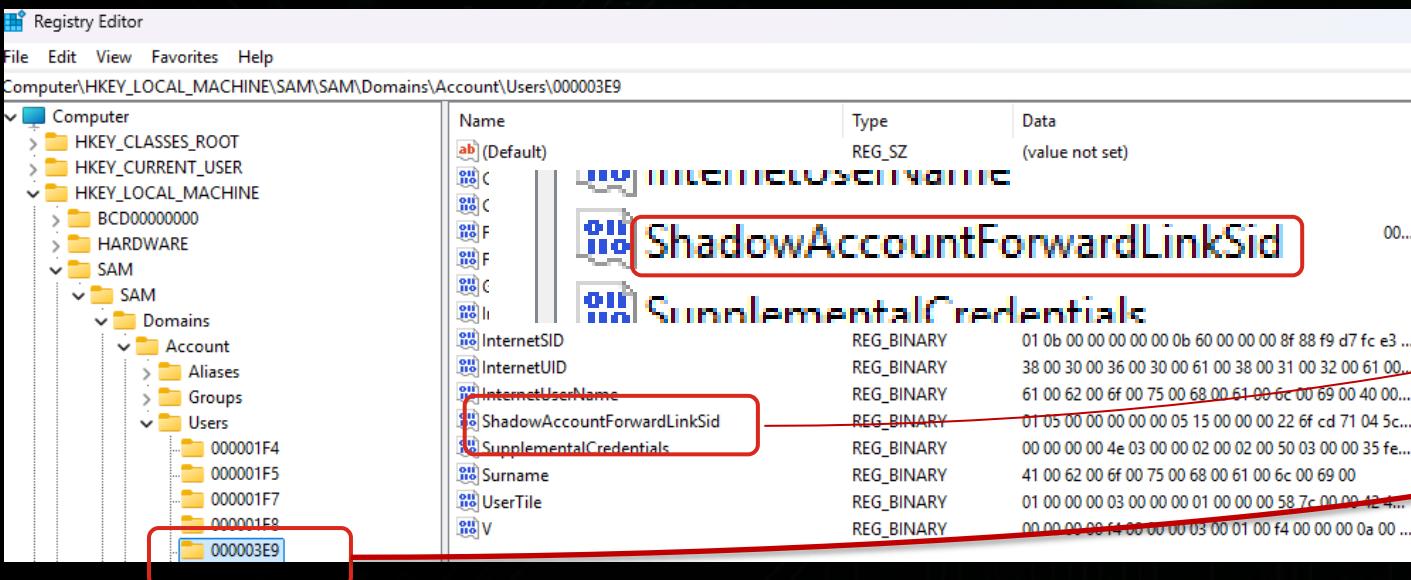
Windows Administrator Protection: Registry artefacts

</> Technical Anatomy of the Shadow Admin Account



Registry Linkage: Trust is maintained through reciprocal links in the SAM:

ShadowAccountForwardLinkSid (on the user account)
and
ShadowAccountBackLinkSid (on the shadow account).

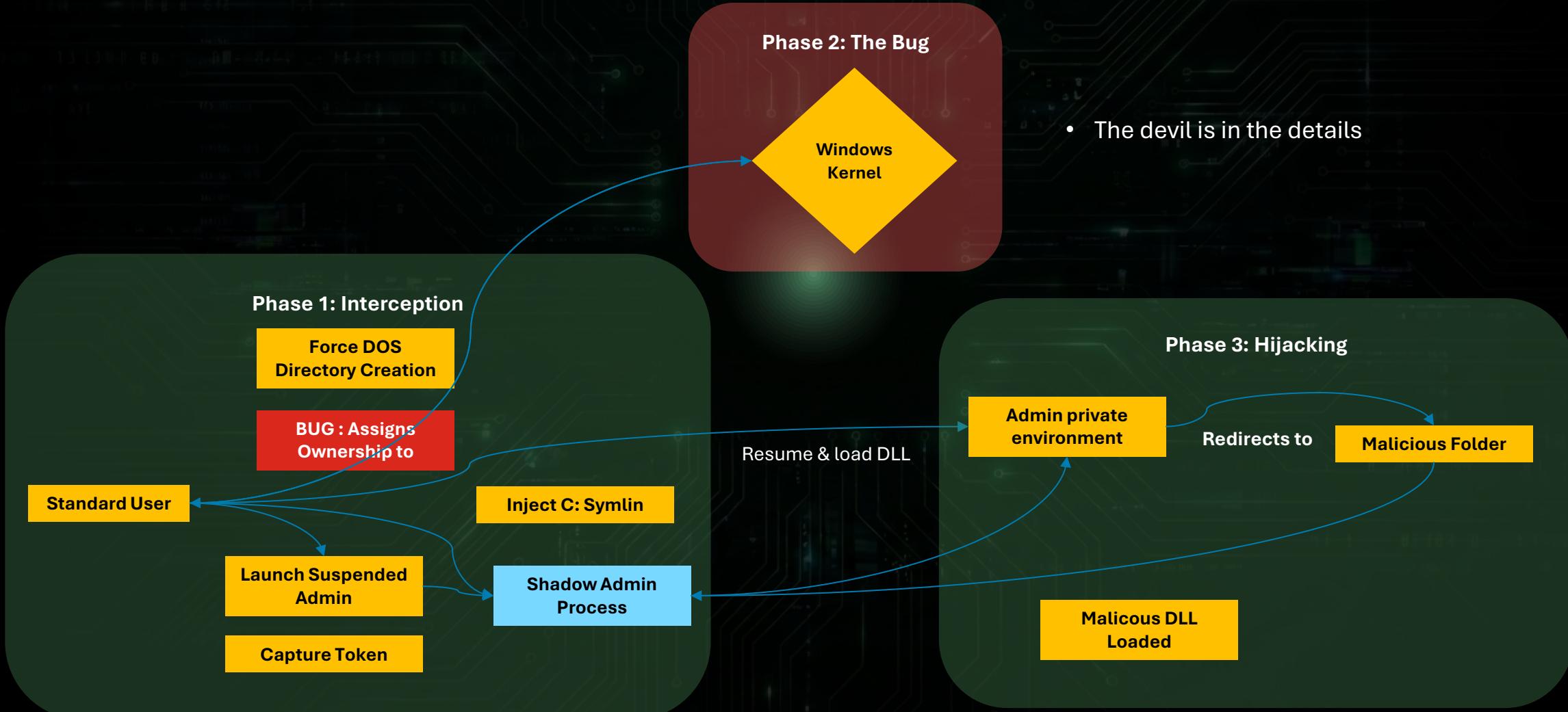


Manipulating these links can potentially redirect elevation to a different shadow account

010001000110010110011001100110110111011011100010000001000111011100100110111011101011100000100001101100001011100110110000101100010011011000110000101101110011000110110001

Windows Administration Protection : Still bypassable ?

</> Taxonomy of a Windows Administrator bypass



01000100011000101100110001101101110110111000100000010001110111001001101110111010111000001000000100001101100001011100110110000101100010011011000110000101101110011000110110001

DEMO