Windows Breakout Workshop

Bastian Kanbach

whoami

- Pentester since 2014 (@CTX since 2016)
- Twitter: @_bka_

Agenda

1. Windows Breakout
    - Desktop Lockdown
    - Kiosks
    - AppLocker

Setting up the Vulnerable VM

1) Import the Windows 10 VM into VMWare/VirtualBox
2) Boot machine
3) You can find a set of tools in "C:\Workshop-Tools“

PS: All passwords are set to ‘123’

Windows Breakout

Windows Breakout Agenda

- Legacy Lockdowns
  - Set directly from Registry Keys / Group Policy
  - Very simple; Mostly Visual
  - Easily Bypassed
- Kiosks
  - Published Application (Locked to a single application)
  - Can be only Visual Lockdowns (Bad idea !)
- AppLocker
  - Modern/Enterprise Lockdowns
  - Common failures – rundll32

Desktop Lockdown

Lockdown Policies

- Can be set using Group Policy or the Registry
- Group Policy Registry Table:
  - https://msdn.microsoft.com/en-gb/library/ms815238.aspx
- Major Issues
  - Difficult to manage
  - Rigid settings
  - Restricted to the available Registry Keys/Group Policies

Desktop Lockdown

Bypass Folder Visual Restrictions

- Alternative File Paths
  - file:///C:/Windows/System32
  - \\127.0.0.1\C$\Windows\System32
- Alternative Location Shortcuts
  - %WINDIR%, %SYSTEMDRIVE%, %USERPROFILE%
  - shell:System, shell:MyComputerFolder, shell:Personal
  - shell:::{031E4825-7B94-4dc3-B131-E946B44C8DD5}
  - My Control Panel.{ED7BA470-8E54-465E-825C-99712043E01C}

Desktop Lockdown

Native & Custom Shells

- Native Shells
  - Most obvious: CMD, Powershell, Powershell_ISE
  - Scripting: bat, vbs, ps1, Macros
  - Others: ftp, command.com
- Custom Shells (emulate Windows functionality)
  - cmd.exe (ReactOS cmd)
  - p0wnedShell (@Cn33liz)

Lab 1: Restricted1

Setup

- Log into Restricted1 (password: 123)
- It should automatically log out after a while
- Log back in
- If you cannot right-click, then it worked

Tasks

- Break out of the weak visual lockdown
- Execute ‘whoami’ in 4 different ways using native shells ONLY

Lab: Restricted2

Setup

- Log into Restricted2 (password: 123)

Tasks

- Run a native shell (eg: cmd.exe)
- Run a custom shell (eg: ReactOS cmd.exe)

Tips

- Which program(s) is/are allowed to run ?
- Can you guess how whitelisting is achieved ?

Kiosks

- Think lockdown + published application
  - Very common in public areas such as hotels
  - Always fun to bypass
- Real OS under the hood
- Typically only visual lockdowns are applied
  - If bypassed, we gain full control of the machine
  - We need to get an Explorer Windows (.. Somehow)
    - We’ll look at some examples of this in the next slide
  - Other techniques we have seen so far might apply

Getting an Explorer Window

- Native Application Functionality:
  - File -> Open
  - Help Menus
  - Print dialogs
- Shortcut Keys
  - Shift * 5
  - Win+R
  - Ctrl+Shift+Esc
  - Alt+F4

Lab: Kiosk1

Setup
- Log into Kiosk1 (password: 123)

Task
- Get an Explorer Window in 6 different ways
- Run cmd each time

Tips
- Click on everything and anything

Lab: Kiosk2

Setup

- Log into Kiosk2 (password: 123)
- It should automatically log out after a while
- Log back in
- Machine should be locked to Microsoft Photos Application

Task

- Obtain a reverse shell

Tips

- If the window is not visible, it does not mean it’s not running

AppLocker

- De facto standard to lockdown Windows machines
  - Though turning to Device Guard (Windows Defender Application Control) in more recent Windows 10 versions
- New(ish) to Windows 7 and Windows Server 2008 (Enterprise and Ultimate)
- Successor to SRP (sometimes called SRPv2)
- Can Import/Export policies
- Offers Audit-only mode!

AppLocker Rules

- Apply to the following file types
  - Executables: exe, com, dll, ocx
  - Installer: msi, msp (install and uninstall)
  - Scripts: ps1, cmd, bat, vbs, js
  - Packaged Apps: Appx
- Applied in different ways:
  - Path
  - File hash
  - Publisher (Publisher, Product name, File name, File Version)

AppLocker Rule Precedence

1) Explicit Deny
   - Rule exists that denies a file
2) Explicit Allow
   - Rule exists that allows a file
3) Implicit Deny
   - Anything else which is not covered by the rules

AppLocker Default Rules

- Default Applocker Executable Rules:

Action	User	Name	Condition	Exceptions
Allow	Everyone	(Default Rule) All files located in the Program Files folder	Path	
Allow	Everyone	(Default Rule) All files located in the Windows folder	Path	
Allow	BUILTIN\Administrators	(Default Rule) All files	Path	

AppLocker Bypass Strategy

- Mainly focus on Default Executable rules
  - This excludes DLLs by default due to performance
- Target AppLocker configurations not AppLocker itself
  - Find different ways to achieve the same goal
- Should convince you that find the right balance between usability, manageability and security is a daunting task

Lab: AppLocker1 (Misconfiguration)

Setup
- Log into Applocker1 (password: 123)

Task
- Run Powershell
- Find a way to execute any binary

Tips
- Enumerate AppLocker policy
- Which policies applies to us?

AppLocker Bypasses

“Trusted Things That Execute Things” – @subTee

- Default rules allow anything in %WINDIR% and %PROGRAMFILES% to be executed by anyone
  - A lot of trusted binaries and directories
- Anything interesting in there which gives us Code Execution?
  - rundll32, regsvr32, reg etc

AppLocker Bypass – Default Weak Folder Permissions

- The %windir% folder is huge; makes a good target
- If we can write to any location under C:\Windows, we can execute it
- ONLY an administrator can write to subdirectories in C:\Windows
  - Right ?
  - Are you sure ?
  - You would hope so !!

AppLocker Bypass – rundll32

- Surprise surprise, it runs a DLL
- Full path: C:\Windows\System32\rundll32.exe
  - Under C:\Windows so trusted
- Invoke: rundll32 <dll_name>,<entry_point>
- Uses LoadLibrary() to load the DLL
  - Executes DllMain() when library is loaded

AppLocker Bypass – regsvr32

- Registers or unregisters a COM DLL in the Windows Registry
- Invoke: regsvr32 [/u] <dllname1>
- Uses LoadLibrary() to load the DLL
  - <empty> - Registers DLL with DllRegisterServer()
  - /u – Unregisters DLL with DllUnregisterServer()

Lab 6: Applocker2 (Default Configuration 1)

Setup

- Log into Applocker2 (password: 123)

Tasks

- Obtain meterpreter reverse shell through the following AppLocker bypass methods:
  - Weak default folder permissions (generate exe)
  - rundll32 (generate dll)
  - regsvr32 (generate shellcode)

AppLocker Bypass – Registry Key Manipulation

Pre-requisites to the understanding the bypass:

1. What registries can we modify as low privileged users?
   - Only our own hive (HKCU)
2. What are (Control Panel) .cpl files?
   - DLLs which export the CPIApplet callback function
   - Used by Control Panel to collate settings into a central place

AppLocker Bypass – Registry Key Manipulation

- Our own custom CPL is blocked but default CPLs are still loaded
- Can we mimic this behaviour and load our CPL in the same way?
- Procmon reveals the truth:

Process Name	PID	Operation	Path	Result
Explorer.EXE	4268	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Control Panel\CPLs	NAME NOT FOUND
Explorer.EXE	4268	RegOpenKey	HKLM\Software\Microsoft\Windows\CurrentVersion\Control Panel\CPLs	SUCCESS
Explorer.EXE	4268	RegEnumValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Control Panel\Cpls	NO MORE ENTRIES
Explorer.EXE	4268	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Control Panel\Cpls	SUCCESS
DllHost.exe	5632	RegOpenKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Control Panel\CPLs	NAME NOT FOUND
DllHost.exe	5632	RegOpenKey	HKLM\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\Control Panel\CPLs	SUCCESS
DllHost.exe	5632	RegSetInfoKey	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Control Panel\Cpls	SUCCESS
DllHost.exe	5632	RegEnumValue	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Control Panel\Cpls	NO MORE ENTRIES
DllHost.exe	5632	RegCloseKey	HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Control Panel\Cpls	SUCCESS

AppLocker Bypass – Registry Key Manipulation

- Modifying the registry:
  - Binaries: reg, regedit; regedt32
  - Scripts: vbscript, jscript
- Launching Control Panel:
  - Control.exe
  - Control Panel.lnk
  - shell:::{ED7BA470-8E54-465E-825C-99712043E01C}
  - My Control Panel.{ED7BA470-8E54-465E-825C-99712043E01C}

AppLocker – InstallUtil

- “… allows you to install and uninstall server resources by executing the installer components in specified assemblies.” – MSDN
- Steps:
  - Loads the binary reflectively with READ permissions
  - Locates class decorated by `[System.ComponentModel.RunInstaller(true)]`
  - Locates Install/Uninstall function (depending if “/U” is specified)
  - Changes permissions to EXECUTE
  - Runs the function

Lab: AppLocker2 (Default Configuration 2)

Setup

- Log into Applocker2 (password: 123)

Tasks

- Obtain code execution through the following AppLocker bypass methods:
  - Registry Key Manipulation (ReactOS DLL or Meterpreter Shell)
  - InstallUtil (p0wned shell)

All The Things

- Combines most AppLocker bypass techniques into a single DLL
- Created by @SubTee (https://github.com/subTee/AllTheThings)
- Covered Techniques:
  - InstallUtil
  - Regsvr32
  - Rundll32
  - Regasm
  - Regsvcs

Windows Breakout Recap

- Covered multiple paths on bypassing restrictions on a Windows machine
  - Legacy Configurations
  - Kiosks
  - AppLocker
- By now you should convinced that it’s not easy to lock down a Windows machine
- Now we have code execution as a low privileged user; What’s next !?

Shutting down