Easy and quick vulnerability hunting in Windows

Cesar Cerrudo
CTO at IOActive Labs
Who am I?

• CTO at IOActive Labs
  • Leading efforts to produce cutting edge research
• I have been working on security for +9 years
• I have found and helped to fix hundreds of vulnerabilities in software such as MS Windows, MS SQL Server, Oracle Database Server, IBM DB2, and more...
• +50 vulnerabilities found on MS products (+20 on Windows operating systems)
• I have researched and created novel attacks and exploitation techniques
Introduction

• With every application you install you are weakening your system security

• Sometimes you need to audit Windows applications
  – Before installing them in hardened servers or in hundreds of desktops.
  – For fun, etc.

• A quick and easy security audit can help to find out vulnerabilities
  – Maybe you can convince your boss of not installing the application or to assume the risks.
  – Make some $$$ by selling it, etc.
Introduction

• Finding some kind of vulnerabilities is not difficult, you just need to know how and where to look for
  – Hopefully today you will learn some how and where
The tools

• Sysinternals tools
  – Process explorer, Process monitor
  – TCPview, accesschk, WinObj, etc.

• Windows debugger
  – WinDbg

• Windows tools
  – Registry editor, Windows explorer, Component services, WMI Control, netstat, cacls, etc.

• Other
  – Wireshark, DeviceTree, etc.
The process

• Always observe and ask yourself What, How, When and Why, be always curious
  – What is that? What does that?
  – How it does that?
  – When it does that?
  – Why it does that?

• Knowledge will get you free and also help you to find vulnerabilities
Targets

• Privileged applications
  – Windows Services
    • Services processes run under privileged accounts
  – Some non services processes run with higher privileges than regular ones
    • WMI processes
    • Windows Installer processes
    • Windows task processes
    • COM Servers, etc.

– Device drivers
– Vulnerabilities could allow to elevate privileges
Targets

• Regular applications
  – ActiveX components
    • Can be accessed remotely by web sites from Internet or Intranet
    • Vulnerabilities could allow to execute code or perform dangerous actions
  – Can save sensitive information on files or registry
  – On Windows >= Vista there is privilege elevation by running with different Integrity Levels
Attack surface

- Network (TCP/IP, etc.)
- File system
- Registry
- Kernel Drivers
- GUI
- DACLs
- COM servers, WMI and ActiveX
Attack surface: GUI

• Unless the application is an interactive service (which is not common nowadays) there isn’t much to look for here.
  – If it’s an interactive service, GUI is protected on Windows >= Vista with new protection against shatter attacks
• If it’s a web app then is just web app security related and not covered here
• So, we won’t focus on the GUI
Attack surface: GUI

• Ask yourself
  – Is an interactive service?
    • Can I manipulate the input in some way?
Attack surface: File System

• Applications save and read data from files
• Applications load binaries (.exe, .dll) read from files that are stored on folders
  – Dll loading search order weaknesses
• Files and folders have DACL
  – If DACL is weak then low privileged users can read, modify, delete, create, etc. files and folders.
    • This could allow elevation of privileges.
      – Reading a password (cleartext or hashed)
      – Modifying/creating a binary, configuration file, etc.
Attack surface: File System

• Ask yourself
  – Have any application file or folder weak permissions?
    • Is the file used to save configuration data?
      – Does configuration data include security options?
    • Is the file used to save sensitive information?
    • Are Dlls or other binaries loaded from that folder?
  – Does the application fail to load Dlls from regular folders?
    • There is any folder with weak permission in Path environment variable?
Attack surface: File System

- **Accesschk.exe users c:\windows –wsu**
  - Searches for file and folder write permissions on all files and folder under c:\windows for users group.
- **ProcMon can monitor for file writing, reading and Dll loading, etc.**
- **Windows Explorer allows to view files and folders, also to view and modify DACLs.**
- **Demo**
  - A couple of Windows 0days, one for NSA only and the other one for lazy people that doesn’t patch often
Attack surface: Registry

• Applications save and read data from registry
• Registry keys have DACL
  • If DACL is weak then low privileged users can read, modify, delete, create, etc. values and keys.
    – This could allow elevation of privileges
      » Changing a folder path or file name
      » Changing some value that alters application execution
      » Reading a password (cleartext or hashed)
      » Etc.
Attack surface: Registry

• Ask yourself
  – Have an application registry key weak permissions?
    • Is the key used to save configuration data?
      – Does configuration data include security options?
    • Is the key used to save sensitive information?
    • Is the key used to save files or folder paths?
      – Are those paths used by the application to access files and folders?
Attack surface: Registry

• Accesschk.exe users hklm –kwsu
  –Searches for registry key write permissions on keys under HKEY LOCAL MACHINE for users group.

• ProcMon can monitor for registry writing, reading, modification, etc.

• Registry Editor allows to view and modify registry key and values, also to view and set DACLs

• Demo
Attack surface: DACLs

- Processes, Threads, Files, File Mappings, Pipes, Interprocess synchronization objects, etc. are Kernel objects
  - If they are securable they have a security descriptor then a DACL
  - Named Kernel objects can be accesses from other processes
    - Some unnamed such as processes and threads can be accessed too
Attack surface: DACLs

• Windows services are securable objects
  – Weak DACL means that low privileged users can change services permissions and elevate privileges
Attack surface: DACLs

• Ask yourself
  – Can the Kernel object be accessed by other processes?
    • Has it a NULL DACL?
    • Has it a weak DACL?
  – What kind of Kernel object is?
    • What are the known attack vectors for processes, threads, file mappings, pipes, etc.?
  – Can a low privileged user change the service DACL or configuration?
  – Demo
Attack surface: COM Servers, WMI and ActiveX

• Applications can install COM Servers, WMI providers and ActiveX controls
  – COM Servers and WMI providers can run under high privileged accounts
    • New Windows versions enforce a strong ACL on COM Servers
      – Applications can modify ACL but with limits
    • If dangerous functionality is exposed to low privileged users it could be abused (most servers will impersonate the caller)
  – ActiveX could be remotely accessed by web sites
    • This could allow abuse of functionality or exploitation of known or unknown vulnerabilities
Attack surface: COM Servers, WMI and ActiveX

• Ask yourself
  – Are there COM Servers or WMI providers with weak DACLs?
    • Do they provide dangerous functionality?
  – Does the COM Server or WMI provider run under a high privileged account and allow low privileged accounts to access them?
    – Can the functionality be abused in some way?
  – Are there ActiveX components with non secure settings?
    • Have these ActiveX vulnerabilities or expose dangerous functionality?
Attack surface: COM Servers, WMI and ActiveX

• Component services tool displays COM Servers permissions and WMI Control tool displays WMI ones

• ActiveX safe for scripting and safe for initialization
  – Subkeys \{7DD95801-9882-11CF-9FA9-00AA006C42C4\} and \{7DD95802-9882-11CF-9FA9-00AA006C42C4\} under key HKCR\CLSID\{ActiveXGUID\}\Implemented Categories
  – Kill bit set if value named Compatibility Flags = 0x00000400 on HKLM \SOFTWARE\Microsoft\Internet Explorer\ActiveX Compatibility \ActiveXGUID
Attack surface: COM Servers, WMI and ActiveX

• To detect COM objects (Servers, WMI and ActiveX) installed by an application monitor (ProcMon tool) key HKLM\SOFTWARE\Classes\CLSID
  – WMI providers are also listed on key HKLM\SOFTWARE\Microsoft \WBEM\CIMOM\SecuredHostProviders

• Demo
Attack surface: Network

• Services can be accessed locally or from the network, using TCP/IP or other protocols
  – We need to identify what ports the application is listening on
    • netstat –anob
    • TCPView

• Services can make outbound connections
  – netstat –anob
  – TCPView
  – Wireshark
Attack surface: Network

• Ask yourself
  – Does the application listen in some ports?
    • What ports?
    • Does it accept remote and/or local connections?
    • What protocols are used?
  – Does the application make outbound connections?
    • What protocols are used?
    • Does it update itself?
      – Update is done in a secure way?
Attack surface: Network

• Fuzz protocols on open ports
  – Time consuming unless you do simple fuzzing
    • Simple fuzzing could be just changing bytes incrementally
      – Just capture a network packet and build a simple tool to change bytes in the packet and send it while target application is attached to a debugger
      – Could easily find some DOS if application is buggy

• Demo
Attack surface: Kernel drivers

• Some applications install device drivers
  – Weak DACLs could allow abuse of functionality and elevation of privileges
    • WinObj and accessenum tools can be used to see DACLs
      – Accesschk.exe -wuo everyone \device
  – They can have vulnerabilities allowing elevation of privileges
  – Need to RE and debug to find out functionality and audit it
    • DeviceTree displays a lot of information about device drivers
Attack surface: Kernel drivers

• Ask yourself
  – Does the application install device drivers?
    • Do they have a proper DACL?
    • What functionality do they provide?
      – Can the functionality be abused/exploited in some way?

• Demo
Conclusions

• Finding vulnerabilities is not difficult if you know how and where to look for.

• Be always aware and ask yourself What, How, When and Why
Fin

• Questions?
• Thanks

• E-mail: ccerrudo@ioactive.com
• twitter: @cesarcer