Cracking Windows Access Control

Andrey Kolishchak

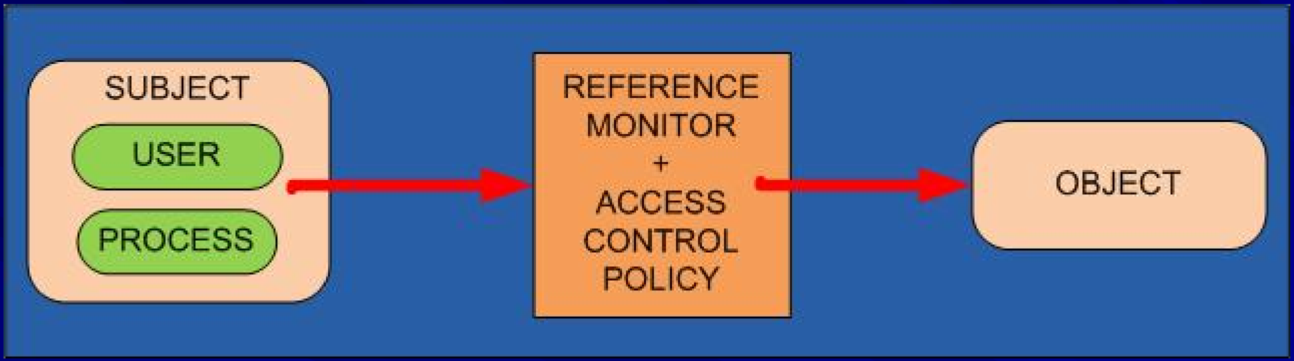
www.gentlesecurity.com

Hack.lu 2007

Outline

- Introduction into access control
- Windows access control weaknesses
- The demo
- Vista mandatory levels
- Exploiting mandatory levels
- Per-application access control

Discretional & Mandatory Access Control

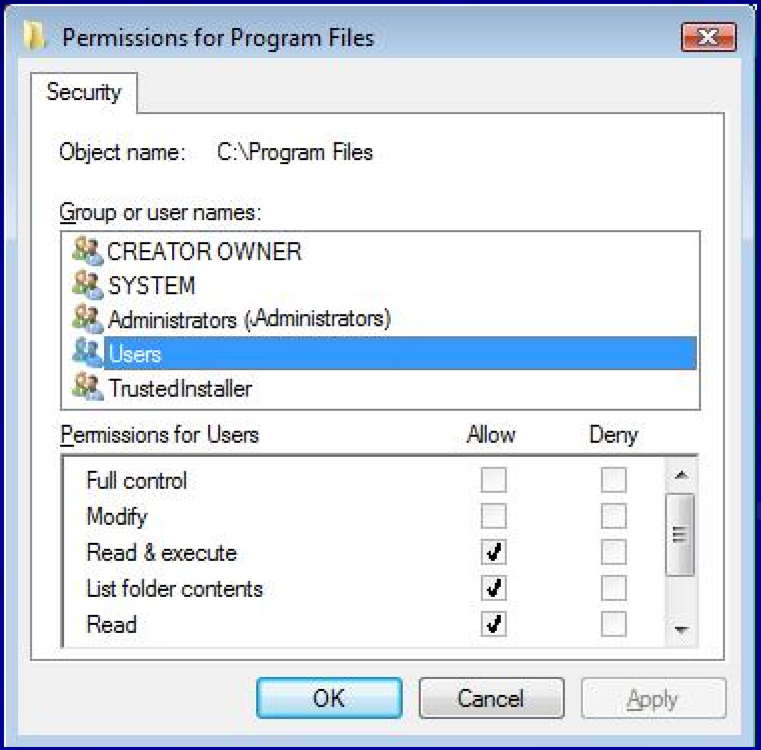


- Discretional Access Control
  - Access policy that depends on a user
  - Access Control Lists (ACL) and capabilities
- Mandatory Access Control (MAC)
  - Access policy decreed by system

Windows Access Control (DAC)

- A controllable object has a list of assigned permissions (ACL), USER x OBJECT

	Object_A	Object_B
USER_1	READ	WRITE
USER_2	EXECUTE	NONE
USER_N	READ WRITE	READ



Object name: C:\Program Files

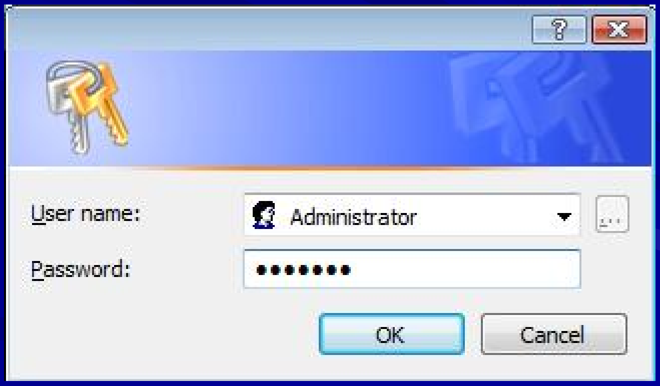
Group or user names:

- CREATOR OWNER
- SYSTEM
- Administrators (Administrators)
- Users
- TrustedInstaller

Permissions for Users	Allow	Deny
Full control	□	□
Modify	□	□
Read & execute	☑	□
List folder contents	☑	□
Read	☑	□

Windows DAC Weaknesses, I

- Dependence on proper user authentication
  - – Social engineering;
  - – Stealing authentication information and keys;
  - – Passwords brute-forcing and sniffing over the network;
  - – Key-logging.
  - – Etc.



User name: Administrator

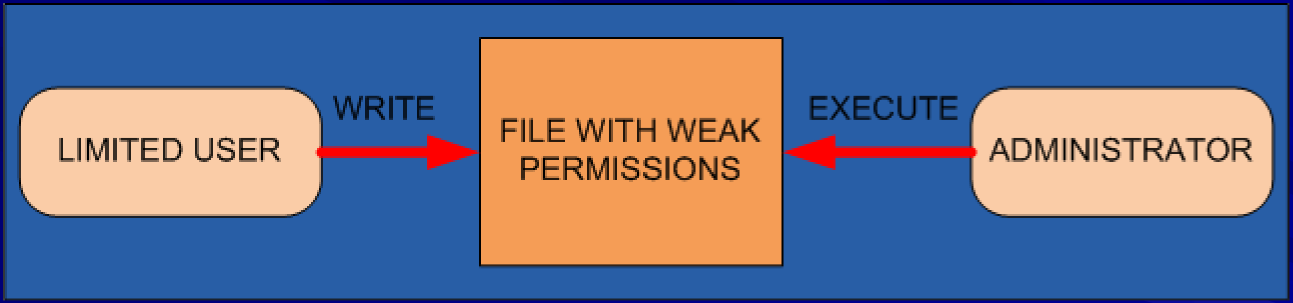
Password: •••••••

Windows DAC Weaknesses, II

- Impersonation
  - Allows a server application to substitute its security identity by the identity of client
  - Elevation: server receives privileges of client
  - Attacks
    - DOS + faked servers exposing RPC, named pipes, COM and other interfaces
    - Vulnerable services
    - All services are affected

Windows DAC Weaknesses, III

- Complexity of ACLs configuration
  - Weak permissions allow full access to Everyone, Users and Authenticated Users
  - Typical attack



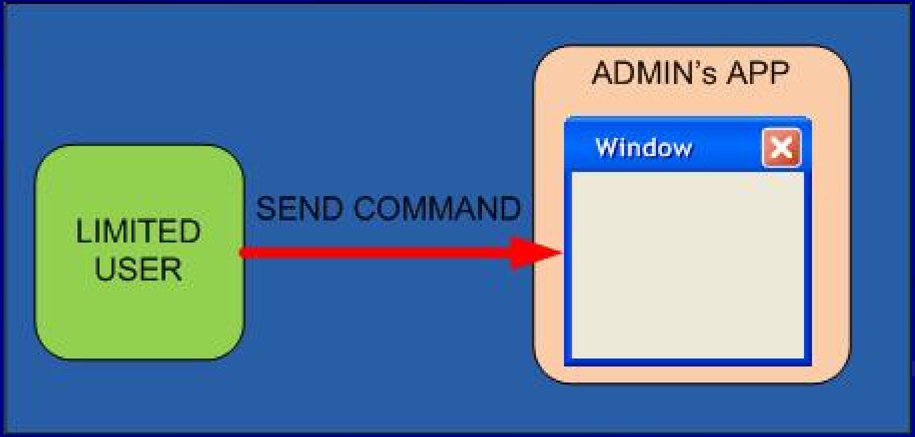
  - Affected: Microsoft, Adobe, Macromedia, AOL, Novell, etc.
  - Accesschk.exe users -wsu "%programfiles%"

Windows DAC Weaknesses, IV

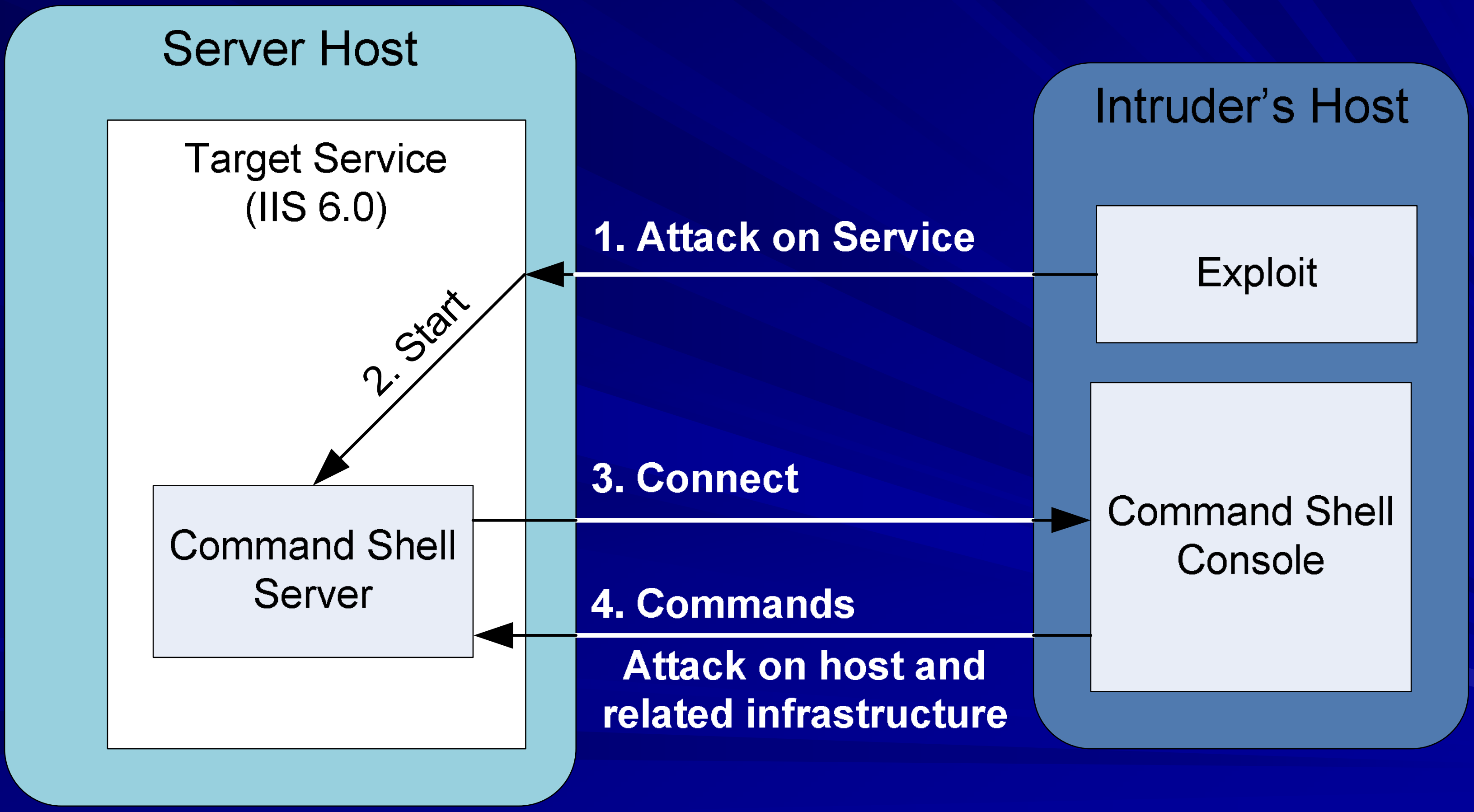
- Creator (owner) of object implicitly receives full permissions
  - Owner may write object's ACL
  - Attacks
    - Permissions revocation
    - Code injection in the processes run by the same user (NetworkService, LocalService)
  - Addressed in Windows Vista
    - Owner Rights SID
    - Unique service SID (requires updated service)

Windows DAC Weaknesses, V

- Permissions cannot be assigned to all objects, e.g.
  - Network
  - Windows subsystem
    - Shatter attacks
    - SetWindowsHook
      - Keyloggers
      - code injection



The Demo

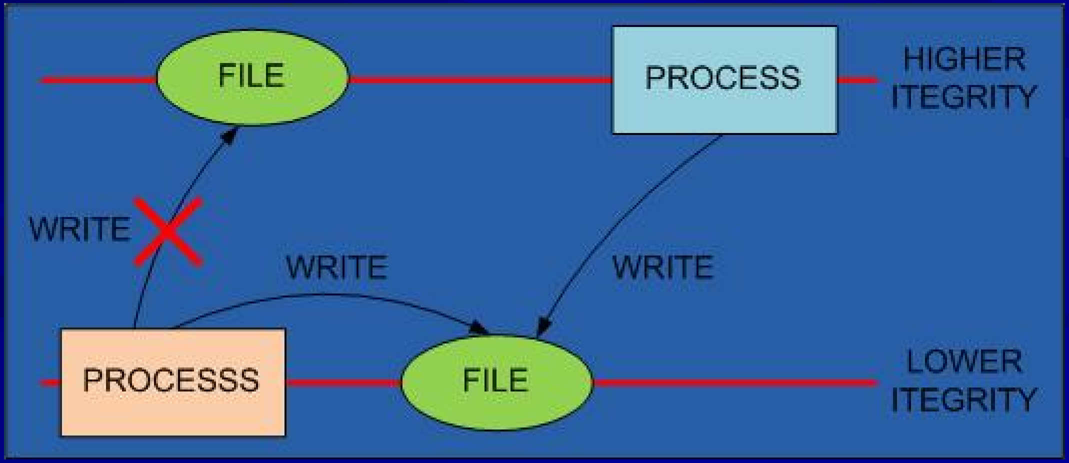

Server Host
Target Service
(IIS 6.0)
2. Start
Command Shell
Server
1. Attack on Service
3. Connect
4. Commands
Attack on host and
related infrastructure
Intruder's Host
Exploit
Command Shell
Console

Interesting Facts

- NetworkService account is nearly the same as LocalSystem
- MS SQL service running as a unique user account can be elevated up to LocalSystem
- Any service's context could be elevated to LocalSystem
- NetworkService account has permissions to sniff network traffic
- An intruder can conduct attacks without introducing additional executable files
  - CodeRed
  - Remote shell via FTP tunnel is just 20 lines VBS script

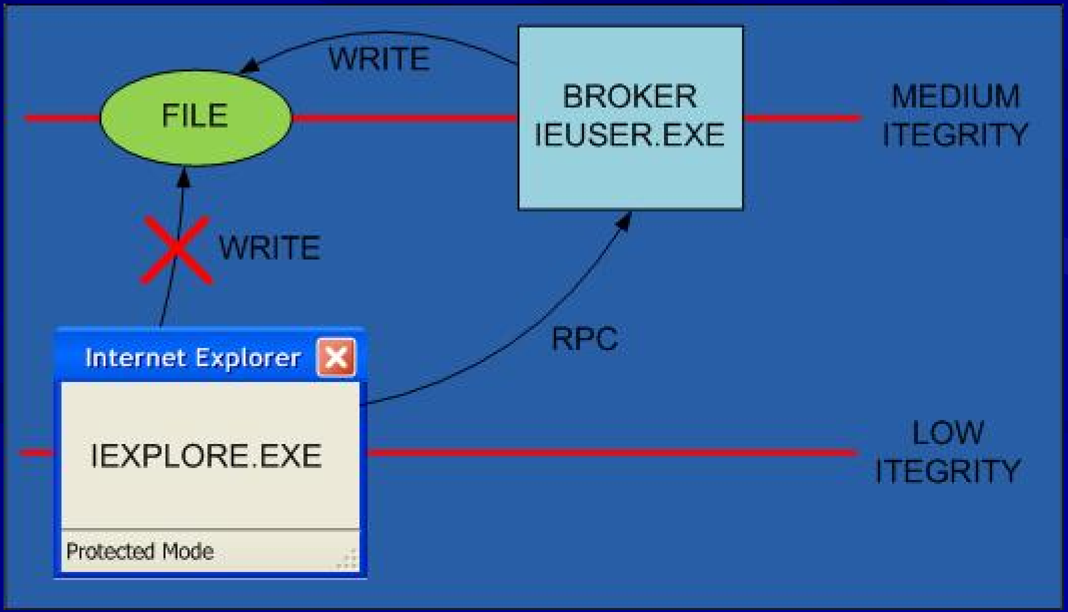
Mandatory Integrity Levels (IL), I

- Integrity Level is an ordered label that define trustworthy of running applications and objects
  - Low, Medium, High and System
  - Mapped to users
- Mandatory Policies restrict lower IL applications
  - No-Write-Up, No-Read-Up and No-Exec-Up



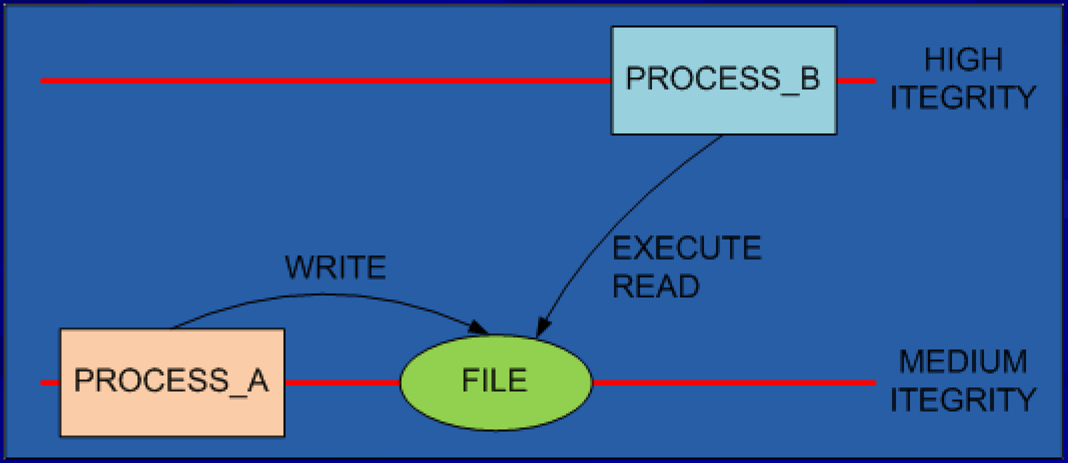
Mandatory Integrity Levels (IL), II

- User Interface Privilege Isolation (UIPI)
- IE Protected Mode
  - Iexplore.exe at Low, renders html
  - Ieuser.exe at Medium, broker for privileged operations



Exploiting Integrity Levels, I

- Medium IL assigned to all objects created at MI and above levels
  - all objects, such as files, are shared
  - No strict boundary between MI and above



Exploiting Integrity Levels, II

- Bypassing UIPI via automation applications
  - Restrictions
    - UIAccess=”true” in the manifest
    - Digital signature
    - %ProgramFiles% or %WinDir%
    - High or +16 IL
  - Attacks
    - Side-by-side DLL injection in writable a %ProgramFiles%
    - Medium-16+16 = Medium

Exploiting Integrity Levels, III

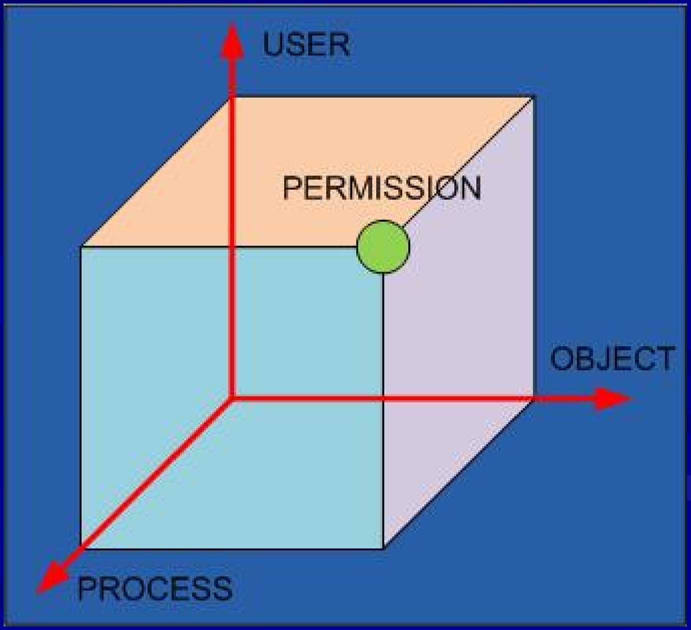
- Vulnerable brokers
  - AppInfo’s handle leak bug found by Skywing (fix in SP1)
    - Bypassing IE’s Protected Mode
  - Any RPC interface might be affected
- ILs are not enforced over the network
- No-Read-Up is not used for files in the default configuration
  - Low Integrity process may read files

Integrity Levels Limitations

- A strict security boundary enforced for Low Integrity processes
- The usage is limited
  - Configuration is restricted, requires re-design of applications
  - Capacity of Low Integrity pool is limited due to shared resources, e.g.
    - An e-mail database accessible by browser

Per-Application Access Control

- New dimension in access control matrix, a process: PROCESS x USER x OBJECT
  - True least privileges
  - Over-complicated



Addressing The Complexity

- Application permissions repository
  - Centralized
  - Attached to applications, e.g. manifests
- Hiding part of permissions behind a mandatory model, such as
  - Windows Integrity Levels
  - Information-flow control
  - Role-based

Thank You!

- Questions?