Core Windows Processes

## **Core Windows Processes**

### **Task 1 – Introduction**

Core Windows processes are essential for the proper functioning of the Windows operating system. They handle system initialization, user logon, service management, and more. Knowing their normal behavior helps identify malware that disguises itself as a legitimate process.

### **Task 2 – Task Manager**

Task Manager is a utility that shows running processes, CPU/memory usage, startup programs, and more. It can be used to monitor core processes.

### **Task 3 –** System

**Normal Behavior:**

* Kernel-level process that manages system-level tasks and drivers.
* Runs under the SYSTEM account.
* No visible window, no user interaction.
* Path: N/A (it’s a kernel process, not a file).
* PID is usually 4 on most Windows systems.

**Unusual Activity:**

* High CPU usage for long periods.
* Hooks or code injections into the process.
* Presence of a user-space file pretending to be System.exe.

### **Task 4 –** smss.exe **(Session Manager Subsystem)**

**Normal Behavior:**

* Responsible for session creation during boot.
* Runs from C:\Windows\System32\smss.exe.
* Starts early in the boot process, launches wininit.exe and csrss.exe.
* Terminates after initialization.

**Unusual Activity:**

* Still running long after boot.
* Running from any path other than System32.
* Multiple instances (only one legitimate during boot).

### **Task 5 –** csrss.exe **(Client/Server Runtime Subsystem)**

**Normal Behavior:**

* Manages console windows, thread creation, and process shutdown.
* Runs from C:\Windows\System32\csrss.exe.
* Usually 1–2 instances, each tied to a session.
* Critical process—termination causes BSOD.

**Unusual Activity:**

* Located outside System32.
* High CPU or memory usage for extended time.
* Network connections (legit CSRSS should not communicate externally).

### **Task 6 –** wininit.exe

**Normal Behavior:**

* Starts system services during boot.
* Located in C:\Windows\System32\wininit.exe.
* Runs as SYSTEM.
* Spawns services.exe and lsass.exe.

**Unusual Activity:**

* Running from temp or user directories.
* Multiple unexpected instances.
* Parent process not smss.exe.

### **Task 7 –** services.exe **(Service Control Manager)**

**Normal Behavior:**

* Manages Windows services.
* Path: C:\Windows\System32\services.exe.
* Runs continuously as SYSTEM.
* Spawns multiple svchost.exe processes.

**Unusual Activity:**

* Located outside System32.
* Direct network communication (services.exe itself normally doesn’t).
* Spawns unknown executables.

### **Task 8 –** svchost.exe **(Service Host)**

**Normal Behavior:**

* Generic host for Windows services.
* Multiple instances are normal.
* Located in C:\Windows\System32\svchost.exe.
* Command-line arguments specify hosted services.

**Unusual Activity:**

* Located outside System32.
* No service name in command line.
* Unusual number of instances with high CPU usage.
* Network activity that doesn’t match hosted services.

### **Task 9 –** lsass.exe **(Local Security Authority Subsystem Service)**

**Normal Behavior:**

* Handles authentication, security policy, and password changes.
* Path: C:\Windows\System32\lsass.exe.
* Runs as SYSTEM.

**Unusual Activity:**

* Located outside System32.
* Dumping LSASS memory (credential theft attempt).
* Multiple instances running.
* Unexpected network connections.

### **Task 10 –** winlogon.exe

**Normal Behavior:**

* Manages user login and logout.
* Path: C:\Windows\System32\winlogon.exe.
* Runs as SYSTEM, one per active session.

**Unusual Activity:**

* Located outside System32.
* CPU spikes without logon/logoff activity.
* Multiple unexplained instances.

### **Task 11 –** explorer.exe

**Normal Behavior:**

* Windows shell, handles desktop, taskbar, and file browsing.
* Path: C:\Windows\explorer.exe.
* Runs under the logged-in user account.

**Unusual Activity:**

* Located outside Windows directory.
* Running as SYSTEM (should be user-level).
* High CPU/network activity without user interaction.

### **Task 12 – Conclusion**

Understanding normal vs. unusual process behavior is critical in detecting malware masquerading as system processes. Always verify **path**, **parent process**, and **runtime behavior**.

