Windows Services

Introduction

• Motivation
  ◦ Implement server side solution (IIS, FTP, DNS, Print, Indexing...)
  ◦ Perform background task (Antivirus, Event log...)
  ◦ Perform privileged operations on behalf of less privileged users

• Definition
  ◦ Background process
    • daemon
  ◦ User-mode process
    • Win32 process with some additional code to interact with the SCM
Windows Services

**Introduction**

- **Model**
  - Typically start during boot process
  - Automatically started (dependency)
    - Once started never stop (when no more needed!)
  - Usually run when no user is logged
  - A host (often) contains several services

- **Infrastructure**
  - Services are controlled by the system
  - RPC based mechanism
    - local and remote administration
Windows Services

Services Components – Logical View

- Layered Architecture

Service Control Program
  Management
    Specific

Service Control Manager
  Routing
    Generic

Service Application
  Implementation
    Specific

 MMC, Net.exe, Sc.exe, ....

Command

Update

Request

Status

Services.exe

Service A
  started

Service B
  stopped

... Service N
  paused
Windows Services

Service Control Program - SCP

- Layered Interfaces

- MMC.EXE, NET.EXE
- ADVAPI32.DLL
- SCM
- Registry

APIs:
- CreateService
- StopService
- StartService
- ...

Pipe connections:
- MyService.exe
- SCM
Windows Services

Services Control Manager - SCM

- Owns all services
- Controls their lifetime
- Routes requests (install/start/stop/pause/delete)
- Maintains services database
  - HKLM\system\CurrentControlSet\Services
Windows Services

Database

- Registration
  - `CreateService(..)` to register a service to the system

- Configuration
  - Name, startup, process type, image file location, error reaction, description...

![Registry Editor](image)
Internal Mechanism

• Types
  ◦ SERVICE_WIN32_OWN_PROCESS
  ◦ SERVICE_WIN32_SHARE_PROCESS
  ◦ SERVICE_INTERACTIVE_PROCESS

• Interface
  ◦ Interaction with the SCM
    • Receive commands
    • Send status feedback
  ◦ Service receive SCM requests – WinMain(…)
    • Windows applications receive messages – WinMain(..)
    • Console applications receive keyboard input – Main(..)
Controls Flow

- Host have at least two threads
Controls Steps

- SCM
  - Logon/Authenticate the user (LsaLogonUser)
  - Load user profile
  - Create service’s process
  - Assign user’s token to the process
  - Call Main(..)

- WinMain
  - Connect main thread to the SCM thread Dispatcher
  - Register service’s function(s) - Collect service’s entry point(s)
  - Start a thread for each service function and wait for them to terminate
  - Wait for any incoming request from the SCM (start/stop/pause/resume) and notify the appropriate thread

- Service Thread
  - Register Service’s request handler
Multiple Hosted Services

• Motivation
  ◦ Save resources
  ◦ Boost performance

• Sample
  ◦ Svchost.exe is the home of many Windows services
Multiple Hosted Services

• Implications
  ◦ Threads share same logon – Identity is the same
  ◦ Threads share security context – Audit is difficult
  ◦ One failure affects the host – Security is critical
    • One vulnerability compromises all hosted threads

• Best Practices
  ◦ Services should only be consolidated if their security requirements exactly match
## Windows Common Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Provides RPC, file, print and named pipe sharing (LanMan server)</td>
</tr>
<tr>
<td>Workstation</td>
<td>Provides network connections to RPC, file, print and named pipe sharing (LanMan client)</td>
</tr>
<tr>
<td>TCP/IP NetBios Helper</td>
<td>Enables support for NetBIOS over TCP/IP (NetBT) service and NetBIOS name resolution (LanMan host)</td>
</tr>
<tr>
<td>NetMeeting</td>
<td>Allows authorized users to remotely access the Windows desktop using NetMeeting</td>
</tr>
<tr>
<td>Windows Installer</td>
<td>Installs, repairs and removes software according to instructions contained in .MSI files</td>
</tr>
<tr>
<td>Remote Registry</td>
<td>Allows remote registry administration (performance monitoring,...)</td>
</tr>
<tr>
<td>System Event</td>
<td>Tracks system events such as Windows logon, network and power events</td>
</tr>
<tr>
<td>SNMP</td>
<td>Monitor the activity and provide Management information</td>
</tr>
<tr>
<td>SNMP Trap</td>
<td>Receive traps generated by local or remote SNMP agents</td>
</tr>
<tr>
<td>Terminal Services</td>
<td>Provide multisession environment</td>
</tr>
<tr>
<td>Telnet</td>
<td>Allows a remote user to log on to the system and run console program using the command line</td>
</tr>
<tr>
<td>WWW</td>
<td>Provides Web connectivity</td>
</tr>
<tr>
<td>WINS</td>
<td>Naming service for NetBIOS network.</td>
</tr>
</tbody>
</table>
Windows Services

Hosting processes

- Many services are implemented as Libraries and must be therefore hosted
  - Hosted services share resources
Recovery

- Motivation
  - Flexible and automatic healing and monitoring features
Services - Status

- Requests
- Transitions

![Service Status Diagram]

1. Running
   - Start
   - Pause
   - Pending
2. Stopped
   - Pending
3. Paused
   - Pending

Status transitions:
- Running → Stopped
- Running → Paused
- Stopped → Running
- Paused → Running
- Pending → Running
- Running → Pending
- Stopped → Pending
- Paused → Pending
- Pending → Stopped
- Pending → Paused
Windows Services

Security - Context

- Running a service under the Local System account
  - Advantages
    - Member of the local Administrators group
    - Owns virtually every defined privilege
    - Potential access to any local resource (SeTakeOwnership)
    - Only one window station is created – good for performance
  - Disadvantages
    - Only access to HKEY_CURRENT_USER\Default
    - Service settings will be place in machine-specific registry area
    - No access to any other interactive users resources
    - Restricted access to network resource (printers, shares, named pipes that allow null session – connection without credential)
    - Very restricted access to WinSta0 (visible/interactive) window station
    - Only one window station is created – bad for security
Windows Services

Security - Context

- Running an interactive service under the Local System account
  - Share user window station (desktop, clipboard,...)
  - SERVICE_INTERACTIVE_PROCESS
Security - Context

- Running a service under a specific user account
  - No access to WinSta0
  - Incompatible with any interactive usage
  - Access to HKEY_CURRENT_USER\sid profile (mapped drives, shares, printers...)
Security - Recommendations

- Services are largest exposure to attack
  - buffer overflow, DOS, privilege escalation
- Use Discrete Account
  - LocalSystem has complete control of the system
  - Better security granularity – LocalSystem cf. multiple users sharing the same user account and password
  - Better auditing (for most exposed services)
Impersonation

- Definition
  - Take the identity of another user and perform a task in his security context
  - A thread is assigned a token different of its process

- Motivation
  - The reason why token exist at all!
  - Allow adjustable settings in a security context (privileges, ACL...)
  - Allow local customization without race condition on other parts of the application
  - Allow a thread to slip into a different security context

- Levels
  - Server creates a token with the trust indicated by the client
    - SecurityAnonymous: Client cannot be identified nor impersonated
    - SecurityIdentification: Client can be identified but not impersonated
    - SecurityImpersonation: Client can be used to open local objects
Impersonation

- Perform privileged operations on behalf of less privileged users
Impersonation

- Typically all threads within a process share (copy) the same token
  - In some situations a thread can have its own token
- A thread token overrides the host process token
  - There is no API to change a process token
  - There is an API to change a thread token
## Windows Services

### Impersonation

- Running a program with other credentials

![Process Explorer](https://www.sysinternals.com/)

<table>
<thead>
<tr>
<th>Process</th>
<th>PID</th>
<th>CPU</th>
<th>Description</th>
<th>Owner</th>
<th>Session</th>
<th>Handles</th>
<th>Window Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Idle Process</td>
<td>0</td>
<td>96</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td>4</td>
<td>1</td>
<td></td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>explorer.exe</td>
<td>784</td>
<td>0</td>
<td>Windows NT Session Manager</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>163</td>
<td></td>
</tr>
<tr>
<td>cscript.exe</td>
<td>340</td>
<td>0</td>
<td>Client Server Runtime Process</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>winlogon.exe</td>
<td>364</td>
<td>0</td>
<td>Windows NT Login Process</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>381</td>
<td></td>
</tr>
<tr>
<td>services.exe</td>
<td>483</td>
<td>0</td>
<td>Services and Controller apps</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>235</td>
<td></td>
</tr>
<tr>
<td>svchost.exe</td>
<td>594</td>
<td>0</td>
<td>Generic Host Process for Win32 Serv...</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>236</td>
<td></td>
</tr>
<tr>
<td>svchost.exe</td>
<td>600</td>
<td>0</td>
<td>Generic Host Process for Win32 Serv...</td>
<td>NT AUTHORITY\SYSTEM</td>
<td>0</td>
<td>249</td>
<td></td>
</tr>
</tbody>
</table>

![Notepad.exe](https://www.winitor.com/)

<table>
<thead>
<tr>
<th>Type</th>
<th>Access</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-8</td>
<td>0x00000003</td>
<td>Windows10</td>
</tr>
<tr>
<td>D-C</td>
<td>0x00000003</td>
<td>C:\windows\svchost.exe</td>
</tr>
<tr>
<td>D-10</td>
<td>0x00000001</td>
<td>\ntsc.exe</td>
</tr>
<tr>
<td>D-14</td>
<td>0x00000003</td>
<td>\windows</td>
</tr>
<tr>
<td>D-C</td>
<td>0x00000003</td>
<td>C:\windows\svchost.exe</td>
</tr>
<tr>
<td>D-20</td>
<td>0x00002001</td>
<td>Windows\WindowsStation\ntsc.exe</td>
</tr>
</tbody>
</table>
Services - Profiles

- Motivation
  - Many running services are unnecessary
    - "Computer Browser" on a computer without a network interface!
  - Some services should not be running under certain situations
    - "NetBIOS" on a computer accessible over the Internet
  - Adjust services configuration...without taking too much risk

- Solution
  - Load services dependant to environment
  - Associate a service with a hardware profile
  - Adjust services configuration...without taking too much risk
Windows Services

Hardware Profiles

- “General” settings affect
  - All hardware profiles
  - All users accounts
- “Log On” settings affect
  - All users accounts
- Some services cannot be disabled
  - “Event Log”
  - “PnP”
- On service should not be disabled
  - “RPC” - system unbootable
Windows Services

Services - Management

- Microsoft Management Console (MMC)

![Microsoft Management Console (MMC) Screenshot]
Windows Services

Services - Management

- Net.exe
  - Net start
  - Net start <servicename>
  - Net stop <servicename>

![Image of command prompt showing services list]
Windows Services

Services - Management

- Sc.exe

![Image of Sc.exe command prompt]

**DESCRIPTION:**

`SC` is a command line program used for communicating with the Service Control Manager and services.

**USAGE:**

```
sc (server) [command] [service name] [option1] [option2] ...
```

The option `<server>` has the form "\ServerName"

Further help on commands can be obtained by typing: "sc [command]"

**Commands:**

- **query** -- Queries the status for a service, or
- enumerates the status for types of services.
- **queryex** -- Queries the extended status for a service, or
  enumerates the status for types of services.
- **start** -- Starts a service.
- **pause** -- Sends a PAUSE control request to a service.
- **interrogate** -- Sends an INTERROGATE control request to a service.
- **continue** -- Sends a CONTINUE control request to a service.
- **stop** -- Sends a STOP request to a service.
- **config** -- Changes the configuration of a service (persistent).
- **description** -- Changes the description of a service.
- **failure** -- Changes the actions taken by a service upon failure.
- **failuresflag** -- Changes the failure actions flag of a service.
- **sidtype** -- Changes the service SIDs type of a service.
- **privs** -- Changes the required privileges of a service.
- **go** -- Queries the configuration information for a service.
- **qdescription** -- Queries the description for a service.
- **qfailure** -- Queries the actions taken by a service upon failure.
- **qfailuresflag** -- Queries the failure actions flag of a service.
- **qsidtype** -- Queries the service SIDs type of a service.
- **qprivs** -- Queries the required privileges of a service.
- **delete** -- Deletes a service (from the registry).
- **create** -- Creates a service. (adds it to the registry).
- **control** -- Sends a control to a service.
- **sdshow** -- Displays a service's security descriptor.
- **sdset** -- Sets a service's security descriptor.
- **show** -- Displays the service SIDs string corresponding to an
  arbitrary name.
- **GetDisplayName** -- Gets the Display Name for a service.
- **GetKeyName** -- Gets the Service Key Name for a service.
- **EnumDepend** -- Enumerates Service Dependencies.

The following commands don't require a service name:

```
sc (server) (command) (option)
```

- **bind** -- (ok | bad) Indicates whether the last boot should be saved as the last-known-good boot configuration
- **lock** -- Locks the Service Database
- **queryLock** -- Queries the Lock Status for the SCManager Database

**EXAMPLE:**

```
sc start MyService
```

Would you like to see help for the QUER Y and QUER YEX commands? [y | n]: 
Windows Services

Services - Management

• Tasklist /svc
Windows Services

Services - Management

- OpenSCManager( )
- CreateService( )
- OpenService( )
- ControlService( )
- QueryServiceStatus( )
- DeleteService( )
Links

- NT Services, Wrox Press, Kevin Miller
- Programming NT Security, Addison-Wesley, Keith Brown
- Windows NT Security, R&D Books Miller Freeman, N.Okuntseff
- Microsoft Windows Internals, Microsoft Press, Solomon, Russinovich
- www.sysinternals.com