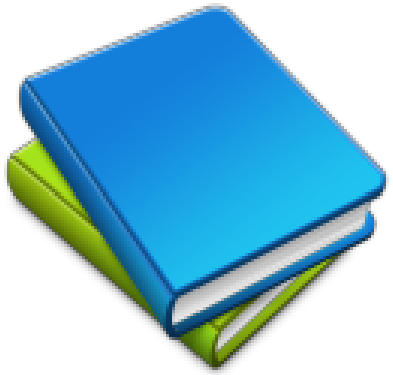
Windows OS Security

Internet Information Server (IIS) Security

Table of Contents



Internet Information Server History

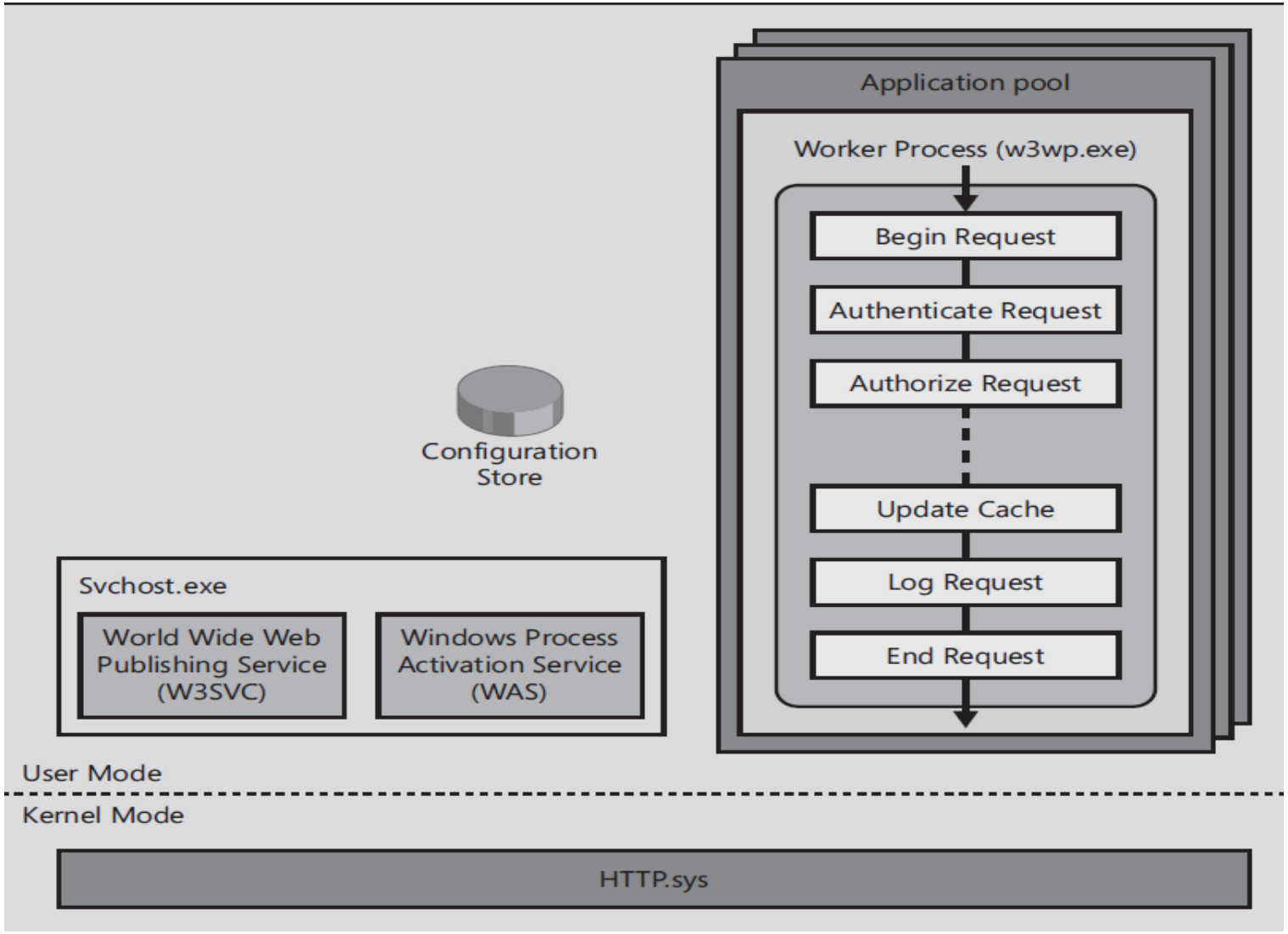
What is Internet Information Server?

- Internet Information Services (IIS) is an extensible web server created by Microsoft
- Web Server Oops!?
  - HTTP and HTTPS
  - FTP and FTPS
  - SMTP
- IIS 1.0 was released as a free add-on for Windows NT 3.51
- IIS 8.5 was released in Windows 2012 R2

If(IIS.Version < 7) { saygoodbay(); }

- IIS 7 was introduced in Windows Server 2008
- All version prior 7 have different architecture!
- IIS 7 and all successors are based on new modern architecture
  - Modern Modular Architecture
    - Install only modules that you really need (Small attack surface)
    - You can find many modules from Microsoft and other companies
  - Modules extend server functionalities trough a public module API
  - Most of the integrated IIS functions are provided by modules
  - Support for Managed Code Modules
  - Extensible XML Based Configuration Engine
  - Control APIs for managing state

IIS Architecture



HTTP.SYS

- What is it?
  - Kernel-mode HTTP stack/listener
  - Always running
- Reliability Features
  - Process routing based on URL
  - Request queues: kernel-mode queuing
- Performance Features
  - Kernel-mode response cache
  - Text-based and binary logging

MS15-034: Vulnerability in HTTP.sys could allow remote code execution: April 14, 2015

Processing Client Requests

- When an HTTP request arrives at the server
  - HTTP.SYS intercepts the request and check the configuration information
  - HTTP.SYS parses the URL path to determine which site/app the request is for
  - HTTP.SYS forwards the request to a worker process
  - The worker process begins a request processing pipeline to execute the request
  - At the end of the processing, a response is generated and returned to HTTP.SYS
  - HTTP.SYS sends a response to the client

❖Each application runs within an isolated application pool

❖One or more worker processes serve an application pool

Security Context

IIS AppPools

- What is an IIS AppPool
  - Application pools host one or more web applications
  - Worker Process share the same configuration
  - Enables applications isolation for better security
- You can configure Web applications to run in:
  - Default application pools
  - You can create a new AppPool
- AppPool Identity
  - Identity under which worker processes in the application pool will run
  - You can create custom user account

ASP.NET Impersonation

- Impersonation is the ability of a thread to execute using different security context
- Typically, this allows the server thread to act on behalf of a client user when access objects
- By default, it is disabled
- You can also programmatically impersonate users

Authentication

Authentication Modules

- Anonymous Authentication
- Basic Authentication
- Digest Authentication
- Forms Authentication
- Windows Authentication

Windows Integrated Authentication

- Encapsulate SSPI (Security Support Provider Interface) authentication schema in HTTP Authorization/WWW-Authentication
- Supports Kerberos and NTLM
- Provides Single Sign On (SSO)
- Browser
  - Internet Explorer
  - Mozilla
  - Chrome 8.0
  - Safari
- Works only on Windows…….

Creating a Managed Code IIS Module

Required Steps

- Create class that implements IHttpModule
- Write code for the Init Method
- Initialize module
- Subscribe to events
- Write code for the subscribed events
- Implement the Dispose method (required)
- Register the module in the Web.config or Applicationhost.config file

Creating a Class from IHttpModule

```
public class CustomAuthenticationModule : IHttpModule
{
 void Init(HttpApplication context)
 {
 }

 void Dispose()
 {
 }
}
```

Integrated pipeline: Events

- Request Events
  - Begin
  - Authenticate
  - Authorize
  - Resolve Cache
  - Map Handler
  - Acquire State
  - PreExecute Handler
  - Execute Handler
  - Release State
  - Update Cache
  - Log
  - End
- On Demand Events
  - SendResponse
  - ReadEntityBody
  - MapPath
- Global Events
  - Initialize / Shutdown
  - Config Change / File Change
  - Application Start / Stop
  - Health Check
  - Trace Event
  - More

Creating a Class from IHttpModule

```
public void Init(HttpApplication context)
{
   //
   // Subscribe to the authenticate event to perform the
   // authentication.
   //
   context.AuthenticateRequest += new
                     EventHandler(this.AuthenticateUser);
   //
   // Subscribe to the EndRequest event to issue the
   // challenge if necessary.
   //
   context.EndRequest += new
                     EventHandler(this.IssueAuthenticationChallenge);
}
```


Questions?