您的潜力，我们的动力

Microsoft
微软（中国）有限公司

ASP.NET安全性

讲师：邵志东

msdn

MSDN Webcasts

课前准备

- Dot Net FrameWork
- VS.NET 2002/2003
- C#/VB.NET

- Level 200

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

SQL Injection如何工作

您的潜力，我们的动力

Microsoft 微软(中国)有限公司

应用中的Query模型

```
SELECT COUNT (*) FROM Users
WHERE UserName='Jeff'
AND Password='imbatman'
```

恶意的Query语句

```
SELECT COUNT (*) FROM Users
WHERE UserName='' or 1=1--
AND Password=''
```

"or 1=1" matches every record in the table

"--" comments out the remainder of the query

输入验证

- 验证所有的输入
  - –使用ASP.NET验证控件
  - –对于其他情况，使用正则表达式 (e.g., web service 参数)
- 对于输出的数据要加密
- 使用参数化的存储过程和查询语句

DEMO1

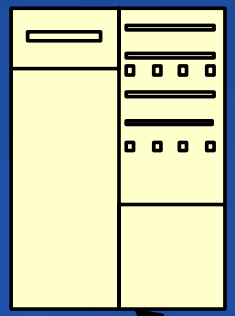
输入安全性

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

请求的安全性事件流

IIS 服务器



④根据 IIS 传送的经过身份验证的标记以及 Web 应用程序的配置设置，ASP.NET 决定是否在处理请求的线程上模拟用户

③IIS 对客户端进行身份验证，然后将经过身份验证的标记随客户端请求一起传送到 ASP.NET 工作进程。

②将客户端凭据传递给 IIS。

①客户端请求 aspx 页面

客户机

身份验证

- 身份验证是指以下过程：获取标识凭据（如用户名和密码），并对照某一颁发机构来验证这些凭据。
- ASP.NET 提供了四个身份验证提供程序：
  - 表单身份验证
  - Windows 身份验证
  - Passport 身份验证
  - 默认身份验证

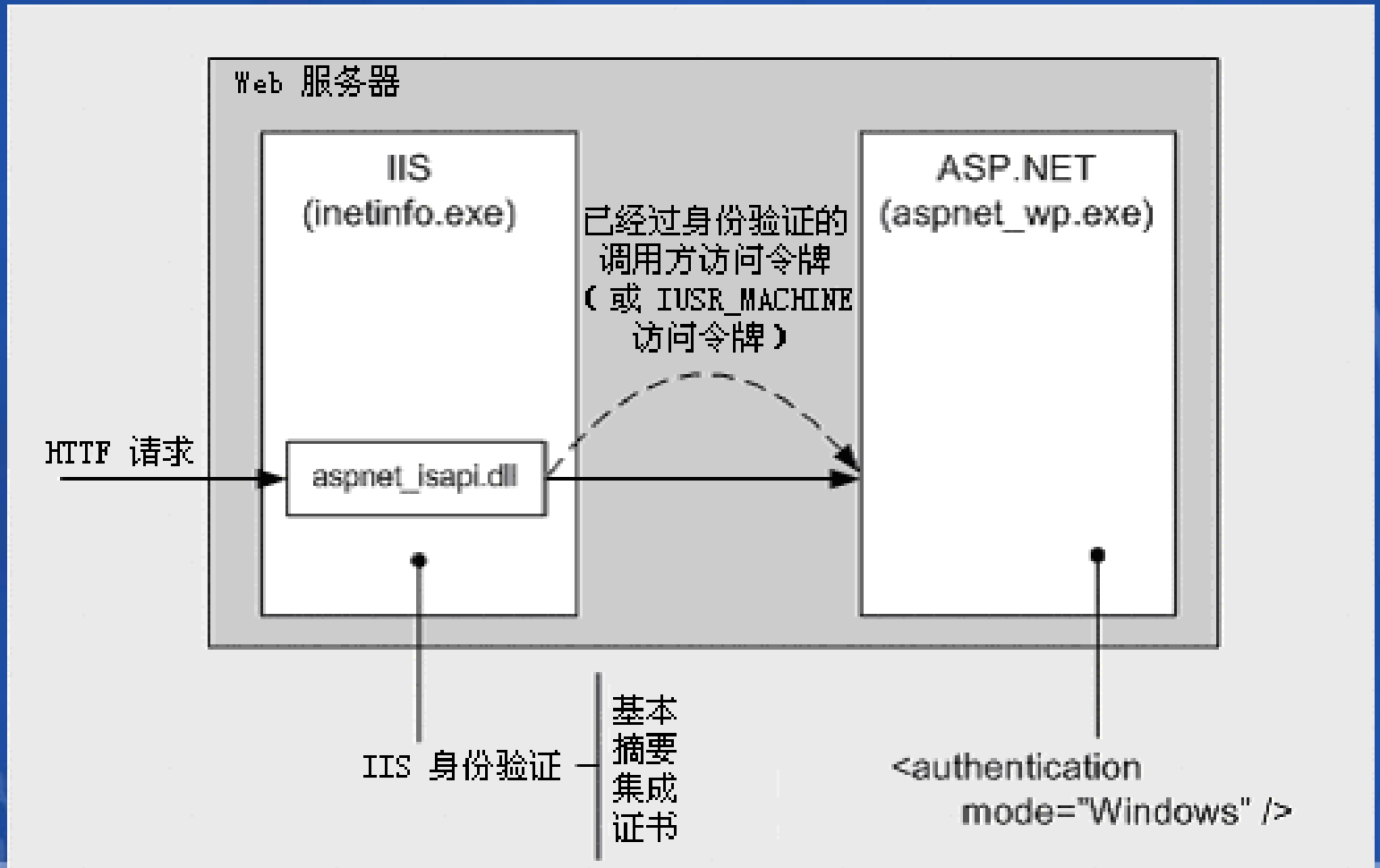
表单身份验证

表单身份验证是指以下系统：将未经身份验证的请求重定向到一个超文本标记语言 (HTML) 表单，使用户能够在其中键入他们的凭据。在用户提供凭据并提交该表单后，应用程序对请求进行身份验证，然后系统以 Cookie 的形式发出身份验证票证。此 Cookie 包含凭据或用于重新获取标识的密钥。浏览器的后续请求自动包含此 Cookie。

Windows 身份验证

在 Windows 身份验证中，IIS 执行身份验证，并将经过身份验证的标记传递给 ASP.NET 工作进程。使用 Windows 身份验证的优点是它需要的编码最少。在将请求传递给 ASP.NET 之前，您可能需要使用 Windows 身份验证来模拟 IIS 进行验证的 Windows 用户帐户。

ASP.NET Windows 身份验证使用 IIS 验证调用方的身份



Passport 身份验证

Passport 身份验证是 Microsoft 提供的集中式身份验证服务，它为成员站点提供单一登录和核心配置文件服务。通常，当您需要跨越多个域的单一登录功能时，将使用 Passport 身份验证。

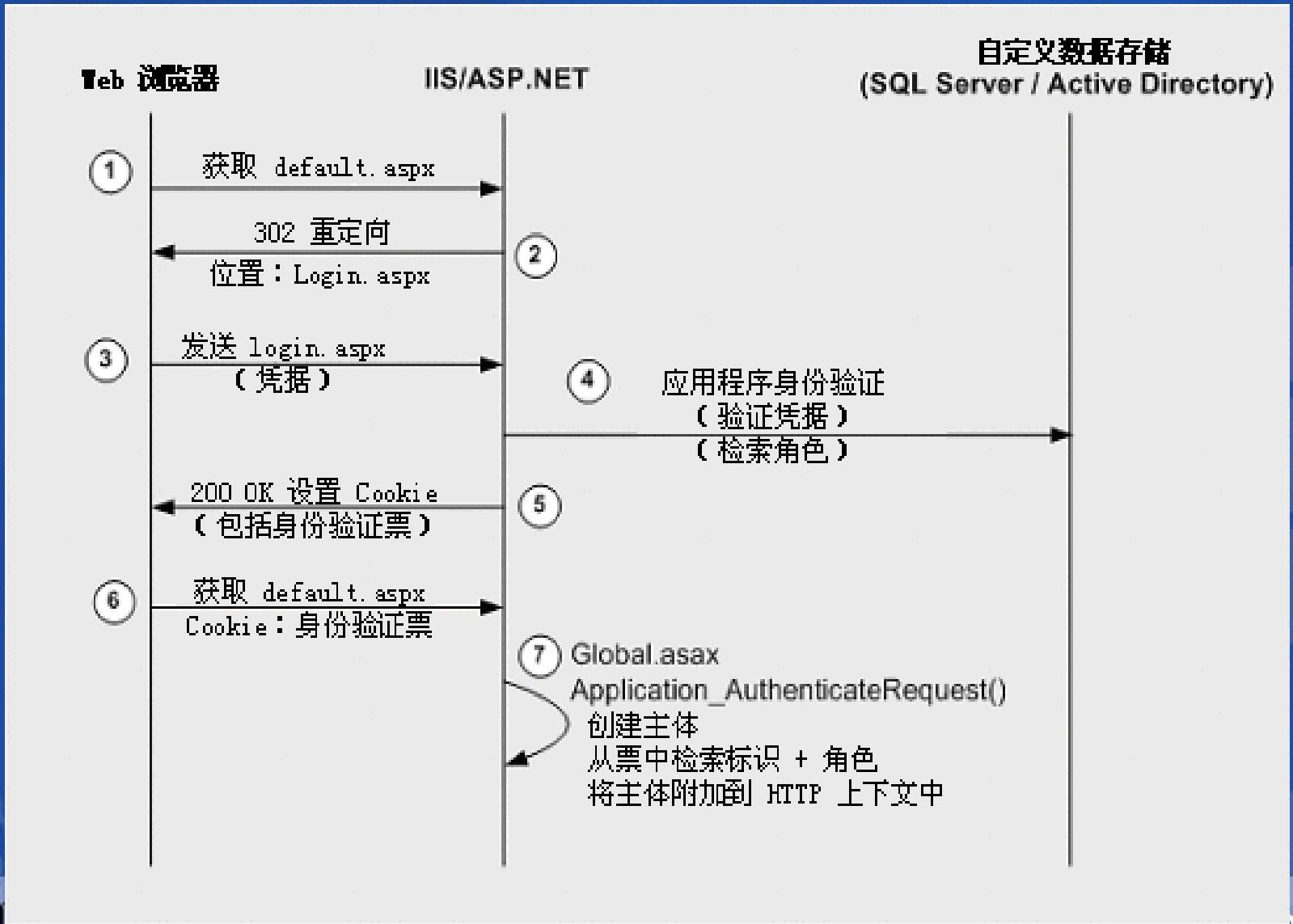
默认身份验证

当 Web 应用程序不需要任何安全功能时，将使用默认身份验证；此安全提供程序需要匿名访问。在所有的身份验证提供程序中，默认身份验证为应用程序提供了最高的性能。当您使用自己的自定义安全模块时，也可以使用此身份验证提供程序。

DEMO2

表单认证的最佳实践

表单身份验证的事件顺序：



表单身份验证的开发步骤

- 1. 将 IIS 配置为使用匿名访问。
  2. 将 ASP.NET 配置为使用表单身份验证。
  3. 创建登录 Web 表单并验证提供的凭据。
  4. 从自定义数据存储中检索角色列表。
  5. 创建表单身份验证票（在票中存储角色）。
  6. 创建一个 IPrincipal 对象。
  7. 将 IPrincipal 对象放到当前的 HTTP 上下文中。
  8. 基于用户名/角色成员身份对用户进行授权。

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

授权

- 授权是指验证经身份验证的用户是否可以访问请求资源的过程。
- ASP.NET 提供以下授权提供程序：
  - FileAuthorization：**FileAuthorizationModule** 类进行文件授权，而且在使用 Windows 身份验证时处于活动状态。
  - UrlAuthorization：**UrlAuthorizationModule** 类进行统一资源定位器 (URL) 授权，它基于 URI 命名空间来控制授权。URI 命名空间可能与 NTFS 权限使用的物理文件夹和文件路径存在很大的差异。

授权

- 若要建立访问特定目录的条件，则必须将一个包含 **<authorization>** 部分的配置文件放置在该目录中。为该目录设置的条件也会应用到其子目录，除非子目录中的配置文件重写这些条件。此部分的常规语法如下所示。
- **<[element]** [*users*] [*roles*] [*verbs*]***/>*** 元素是必需的。必须包含 *users* 或 *roles* 属性。可以同时包含二者，但这不是必需的。

- 以下示例向 Kim 和管理角色的成员授予权限，而拒绝 John 和所有匿名用户：

```
<authorization>
    <allow users="Kim"/>
    <allow roles="Admins"/>
    <deny users="John"/>
    <deny users="?"/>
</authorization>
```

- 若要允许 John 并拒绝其他任何人，可以构造下面的配置部分。
  - <authorization> <allow users="John"/> <deny users="*"/> </authorization>
- 下面的示例允许每个人使用 **GET**，但只有 Kim 可以使用 **POST**。
  - <authorization>
  - <allow verb="GET" users="*"/>
  - <allow verb="POST" users="Kim"/>
  - <deny verb="POST" users="*"/>
  - </authorization>

DEMO3

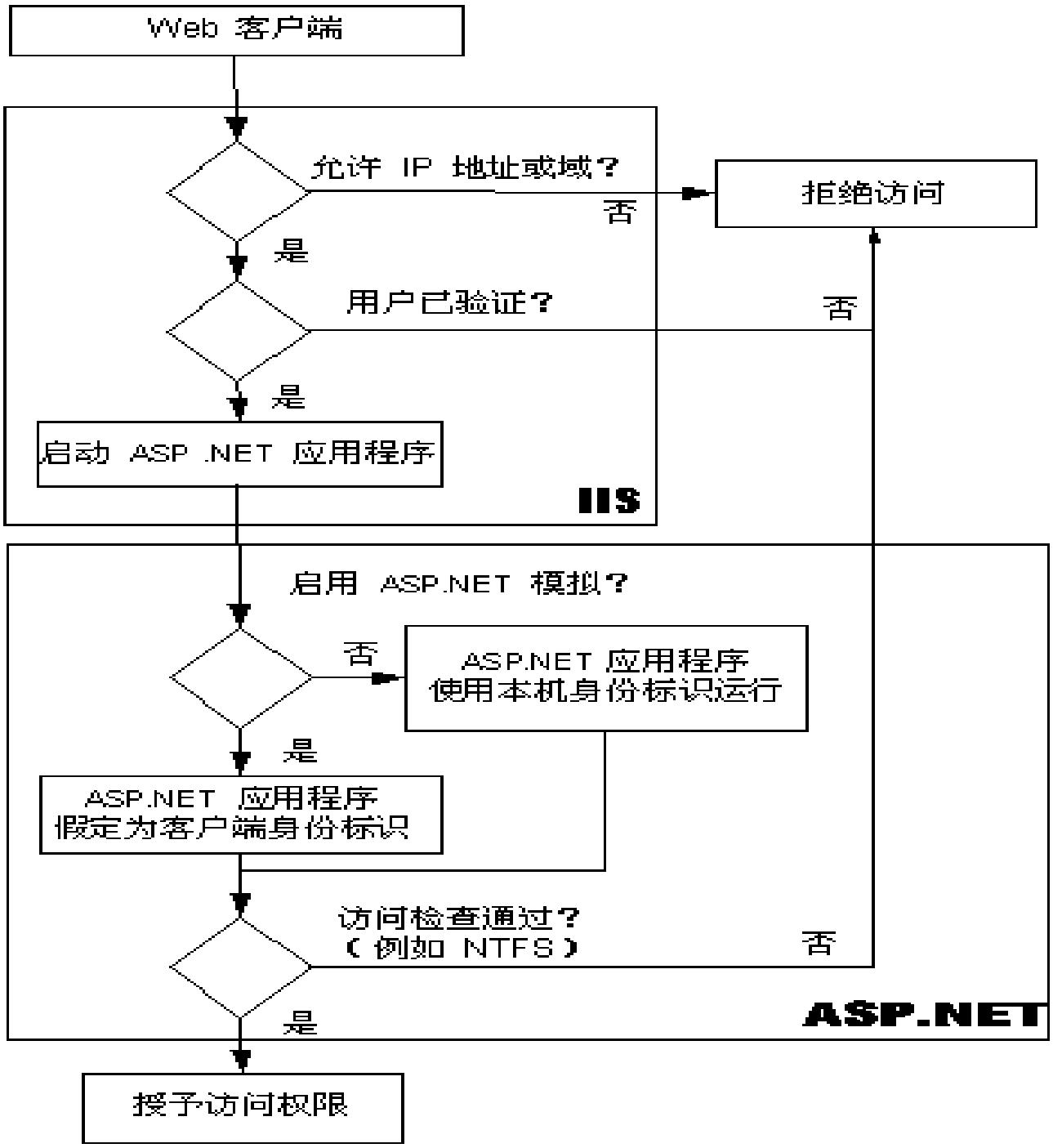
授权

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

ASP.NET模拟

缺省情况下，ASP.NET应用程序以本机的ASPNET帐号运行，该帐号属于普通用户组，权限受到一定的限制，以保障ASP.NET应用程序运行的安全。但是有时需要某个ASP.NET应用程序或者程序中的某段代码执行需要特定权限的操作，比如某个文件的存取，这时就需要给该程序或相应的某段代码赋予某个帐号的权限以执行该操作，这种方法称之为身份模拟（Impersonation）。



启用模拟的方法

- 在ASP.NET应用程序中使用身份模拟一般用于资源访问控制，主要有如下几种方法：
  - 模拟IIS认证帐号：

```
<IDENTITY impersonate="true" />
```

  - 在某个ASP.NET应用程序中模拟指定的用户帐号

```
<IDENTITY impersonate="true"
    userName="accountname" password="password" />
```

  - 在代码中模拟IIS认证帐号
  - 在代码中模拟指定的用户帐号

启用模拟的方法

- 在代码中使用身份模拟更加灵活，可以在指定的代码段中使用身份模拟，在该代码段之外恢复使用ASPNET本机帐号。该方法要求必须使用Windows的认证身份标识。

ASP.NET进程标识

- 使用权限最小的账户
  - 使用权限最少的帐户可以减少与进程攻击相关的威胁。
- 避免作为SYSTEM运行
- 域控制器和 ASP.NET 进程帐
  - 一般情况下，不建议在域控制器上运行 Web 服务器，因为对服务器的攻击就是对域的攻击。
- 使用默认 ASPNET 帐户
  - 已将本地 ASPNET 帐户明确配置为使用尽可能最少的权限集运行 ASP.NET Web 应用程序。如果可能，尽量使用 ASPNET。

DEMO4

ASP.NET模拟

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

存储机密

- Web 应用程序通常需要存储机密。需要妥善保管这些机密，以防止不道德的管理员和有恶意的 Web 用户进行访问
- 机密的典型示例包括：
  - SQL 连接字符串。一个常见的错误是以纯文本形式存储用户名和密码。
  - Web.config 中的固定标识。
  - Machine.config 中的进程标识。
  - 用于安全地存储数据的密钥。
  - 用于根据数据库进行表单身份验证的密码。

在 ASP.NET 中存储机密的选项

NET Web 应用程序开发人员可以使用多种方法来存储机密。它们包括：

- .NET 加密类。.NET 框架包含可用于加密和解密的类。这些方法要求安全地存储加密密钥。
- 数据保护 API (DPAPI)。DPAPI 是一对 Win32 API，它使用从用户密码派生的密钥对数据进行加密和解密。在使用 DPAPI 时，您并不需要进行密钥管理。操作系统对作为用户密码的密钥进行管理。
- COM+ 构造函数字符串。如果应用程序使用服务组件，则可以在对象构造字符串中存储机密。该字符串以明文形式存储在 COM+ 目录中。
- CAPICOM。这是一个 Microsoft COM 对象，它提供对基础加密 API 基于 COM 的访问。
- 加密 API。这些 API 是执行加密和解密的低级 Win32 API。

DEMO5

存储机密

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

使用加密

- 使用名称空间System.Security.Cryptography：该命名空间提供加密服务，包括安全的数据编码和解码，以及许多其他操作，例如散列法、随机数字生成和消息身份验证。
- 使用RNGCryptoServiceProvider，代替System.Random.

DEMO6

使用加密

议程

- 输入安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

使用最佳实践

- 如果应用程序使用表单身份验证，而且在用户身份验证中需要考虑性能问题，则检索角色列表并将其存储在身份验证票中。
- 如果使用表单身份验证，则始终给每个请求创建一个主体并将其存储在上下文中。
- 如果角色太多而无法存储在身份验证 Cookie 中，则使用全局应用程序缓存来存储角色。
- 不要创建权限最少的自定义帐户来运行 ASP.NET。而是应更改 ASPNET 帐户密码，并在应用程序需要访问的任何远程 Windows 服务器上创建重复帐户。
- 如果必须创建自定义帐户以运行 ASP.NET，请使用权限最少的用户。例如：
- 如果主要考虑管理问题，请使用权限最少的域帐户。

使用最佳实践

- 如果使用本地帐户，则必须在 Web 应用程序需要访问的任何远程计算机上创建重复帐户；如果应用程序需要访问非信任域中的资源或者防火墙禁止 Windows 身份验证，则必须使用本地帐户。
- 不要使用本地 SYSTEM 帐户来运行 ASP.NET。
- 不要给 ASPNET 帐户授予“充当操作系统的一部分”权限。
- 在以下情况中使用 SSL：
  - 在浏览器和 Web 服务器之间传送安全敏感信息时。
  - 使用基本身份验证（以保护凭据）时。
  - 使用表单身份验证进行身份验证（与个性化相对）时。
- 避免使用纯文本形式存储机密。

小结

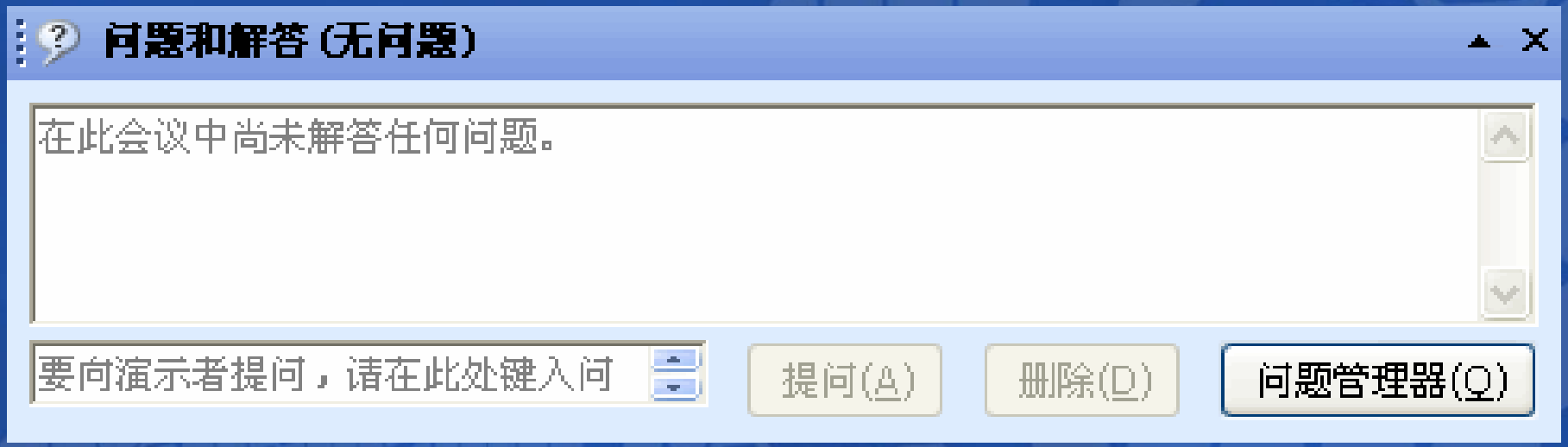
- 输入的安全性
- 身份验证
- 授权
- ASP.NET模拟
- 存储机密
- 使用加密
- ASP.NET安全使用最佳实践

更多信息.....

- MSDN网站： http://msdn.microsoft.com
- 程序员大本营:**www.csdn.net**

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。




您的潜力，我们的动力
Microsoft
微软（中国）有限公司
Microsoft
msdn
MSDN Webcasts