OWASP 应用程序安全设计项目

中文版本团队成员

王颉　王文君

欢迎大家指正翻译错误或不恰当的地方。我们将在后续版本中不断完善该项目。

原项目地址：https://www.owasp.org/index.php/OWASP_Secure_Application_Design_Project

目录

1. 简介

我们都知道在应用程序开发过程中“安全编码”的重要性。但是，我们对“安全设计”会给予同等的重视吗？相信我们大多数人可能会说，不会！

“OWASP 应用程序安全设计项目”突出了安全设计的重要性和其中的一些重要步骤。虽然设计级别的缺陷并非为每个人所知，但它们的存在对应用程序来说确实是一个非常大的风险。这种缺陷很难通过静态或动态的应用程序扫描被发现，而需要对应用程序架构和布局深入了解而进行手工发掘。随着业务需求的增长，应用程序设计和架构的复杂度也随之增加。今天，越来越多的应用程序使用定制和多样化技术，这使得对设计迫切需要进行审核。

本项目的重点是突出一些重要的安全设计原则与步骤，开发人员和架构师必须遵循它们来进行安全的应用程序设计。通过设计审核，我们可以发现其中的风险，然后采取措施在设计中避免这种风险。

2. 了解设计

2.1 什么是应用程序设计？

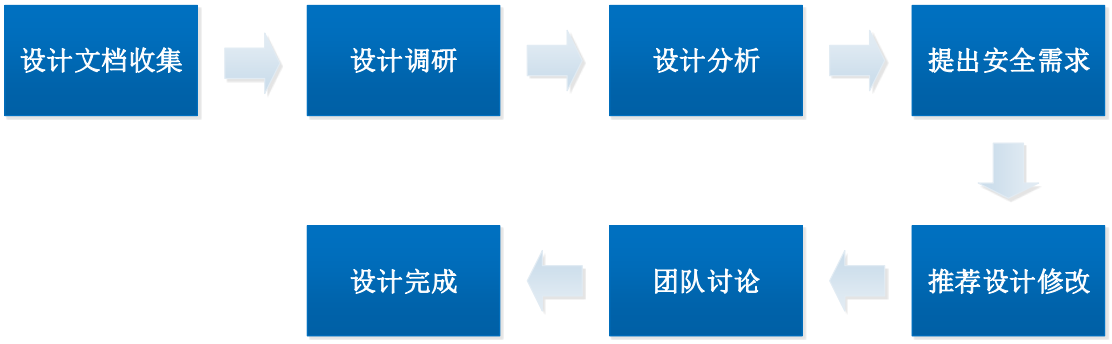
设计是应用程序的蓝图，是应用程序开发的基础。设计展示了应用程序的结构，并确定了应用程序需要的不同组件。通常，设计确定了应用程序执行流程的架构。并且，大多数的应用程序设计都基于 MVC 的概念。在这样的设计中，不同的组件彼此交互，并有序的处理任何用户请求服务。

2.2 为什么需要设计审核？

设计审核应当是软件安全开发过程中设计阶段的一个重要组成部分。如果在设计阶段对应用程序进行安全性审核，可以发现其中的很多后门漏洞。另外，设计审核还有利于以更好的方式实现安全需求。

3. 设计审核方法

设计审核的流程如下图所示。下面我们将对设计审核所采用的方法进行详细的分析。



3.1 设计文档收集

本阶段需要收集进行安全设计所需的信息。这将涉及到由开发团队维护的所有跟设计有关的文档，如：流程图、序列图、类图等。另外，还需要获得需求分析文档，以了解所提出设计的目标。

3.2 设计调研

在这个阶段中，我们需要从数据流、不同应用程序组件及它们之间的交互、数据处理等方面进行深入了解与研究。通过人工分析、与设计或技术架构团队进行讨论来达到目的。应用程序的设计和架构必须彻底理解，以分析可能导致应用程序中存在安全漏洞的脆弱区域。下面给出了在威胁分析时必须考虑的几个关键设计方面：

- 数据流或代码布局；
- 访问控制；
- 现有的或内置的安全控制；
- 非用户输入的入口点；
- 与外部服务的集成；
- 配置文件和数据源的位置；
- 插件和定制化展现（在内置设计框架的情况下）。

以上设计内容将有助于确定应用程序的信任边界，从而有助于对漏洞和对应用程序造成的风险等级做出判断。

3.3 设计分析

在了解了设计内容之后，下一个阶段是分析威胁，此阶段包括“威胁建模”。

对在前一阶段确定的不同设计方面，必须确定风险。它包括从攻击者的角度分析设计，并发现当前设计中存在的后门和不安全的地方。该分析可以大致基于以下两个重要的方面完成：

（1）不安全的开发：这意味着设计存在漏洞，并可能会成为应用程序中的安全漏洞。例如，对业务逻辑功能的不安全引用。

（2）缺乏安全的开发：这意味着设计中没有纳入安全实践。例如，在连接到外部服务器时，没有使用不同的安全需求保护数据的机密性和完整性。

下面列出了类似的实例来说明，在分析不同设计方面的同时，应考虑：

- 数据流：
  - ✧ 用户输入是否被直接用于引用业务逻辑的类或函数？
  - ✧ 是否有一个数据绑定缺陷？
  - ✧ 是否暴露任何后门参数来调用业务逻辑？
  - ✧ 应用程序的执行流程是否正确？
- 身份验证和访问控制：
  - ✧ 是否对所有文件实现访问控制？
  - ✧ 是否安全地处理会话？
  - ✧ 是否存在单点登录？单点登录是否留下后门？
- 已有或内置的安全控制：
  - ✧ 在现有任意安全控制中的弱点；
  - ✧ 安全控制的部署是否正确？
- 架构：
  - ✧ 对所有的输入是否有验证？

- ✧ 到外部服务器的连接是安全的吗？
- 配置或代码文件和数据存储：
  - ✧ 配置文件中是否含有敏感数据？
  - ✧ 是否支持任何不安全的数据源？

详细的安全设计清单，请见附件 1。

在本阶段结束时，我们得到了一个关于威胁或不安全区域的列表。

3.4 提出安全需求

在分析了设计中存在的不安全问题后，必须在本步骤中创建一组对应的安全需求。这些安全需求是高级别的变更或是在设计过程中需与设计合并的需求，比如：在处理来自 web 服务响应的输入信息前进行验证。任何需要对设计中确定的漏洞而提供的保护措施，都将作为设计的安全需求。这种基于风险的方法，将有助于开发团队对安全需求确定优先级。

3.5 推荐设计变更

在这个阶段中，每个安全需求必须与一个安全控制相关联。对于设计安全控制的最佳匹配被提出并被记录。这些安全控制是对安全要求的详细说明。在这里，我们将明确需变更的内容，或在设计中需合并的内容，以满足需求或缓解威胁。对设计推荐的变更或控制应当明确和详细，类似于下面给出的实例：

- 从以下方面详细说明验证策略：
  - ✧ 明确正确的应用程序组件，比如：servlet 过滤器、拦截器、验证器的类；
  - ✧ 检查机制的部署；
  - ✧ 验证机制；
  - ✧ 使用第三方的安全 API 或框架的内置设计功能；
- 加密技术；
- 设计模式；

- 其它在设计中依赖的控制。

3.6 团队讨论

安全需求列表和提出的控制必须与开发团队讨论。来自团队的询问必须被解决，并且安全控制的可行性必须被决定。如果有例外的话，必须被考虑；并且备用方案的建议也应该提出。在这个阶段中，需要就最终的安全控制与团队达成一致。

3.7 设计完成

与开发团队共同完成的最终设计必须被再次审核，并为未来的开发过程最终确定设计。

附件 1. 应用程序安全设计清单

应用程序安全设计清单

类别	存在漏洞的地方		分析内容
设计	代码流—基于 MVC 的代码	后门参数、功能、文件的存在	1. 是否有后门或没有暴露的业务逻辑类？
			2. 是否有与业务逻辑相关的、但未使用的配置？
			3. 如果请求参数被用于确定业务逻辑的方法，是否有一个用户权限与该权限允许使用的方法或活动的恰当匹配？
		安全检查部署	1. 是否在处理输入的信息前部署安全检查？
		不安全的数据绑定机制	1. 检查未暴露的实参是否出现在表格对象中与用户输入绑定。如果出现，检查它们是否设有默认值。
			2. 检查未暴露的实参是否出现在表格对象中与用户输入绑定。如果出现，检查它们在与表格绑定前是否初始化。
	身份认证和访问控制机制	不安全的身份认证和访问控制逻辑	1. 身份认证和授权检查的部署是否正确？
			2. 在收到非法的请求时，程序运行是否停止或被终止？比如，当身份验证或授权检查失败时。
			3. 安全检查是否被正确执行？是否有后门参数？
			4. 在 Web 的根目录中是否对所有需要的文件和文件夹使用了安全检查？
		多余的配置	1. 是否有像 Access-ALL 这样的默认配置？
			2. 配置是否用于所有的文件和用户？
			3. 如有容器托管的身份认证，是否只有基于 Web 的身份认证方法？
			4. 如有容器托管的身份认证，是否对所有的资源使用了身份认证？
		不安全的会话管理	1. 设计是否安全地处理会话？

		脆弱的密码处理	1. 密码是否强制使用了密码复杂度检查？
			2. 密码是否以加密的形式被存储？
			3. 密码是否向用户泄漏，或写入文件、日志或控制台？
	数据访问机制	配置文件或代码中存在敏感数据	1. 数据库凭证是否以加密的形式被存储？
		存在或支持对不同不安全数据和它们相关的漏洞	1. 设计是否支持弱数据存储，如：扁平文件。
	集中验证和截获器	任何已有安全控制中的弱点	1. 集中验证是否应用于所有的请求和所有的输入？
			2. 集中验证检查是否阻止了所有的特殊字符？
			3. 验证过程中是否有特殊请求被忽略？
			4. 设计中是否对被验证的参数或功能维护特定列表？

架构	**入口点**	不安全的数据处理和验证	1. 所有不可信的输入数据是否得到验证？
	外部一体化	不安全的数据传输	1. 数据是否在加密通道中传输？应用程序是否使用 HTTPClient 进行外部连接？
			2. 设计是否包含组件或模块之间的会话共享？在会话两端会话是否被正确验证？
		被提高的权限等级	1. 设计是否对外部连接或命令使用了被提高的 OS 或系统权限？

配置	**使用外部 API**	在第三方 API 或功能中出现已知漏洞	1. 在使用的 API 或技术中是否含有已知漏洞？如：DWR。
	内置安全控制	常见安全控制	1. 设计框架中是否提供内置的安全控制？如：ASP.NET MVC 中的<%: %> 。
			2. 在已有的内置控制中是否有漏洞或弱点？
			3. 在设计中是否启用了所有的安全设置？