

安全编码规范

（通用版）

**—— 安全团队**

**目录**

[一. 通用安全开发要求](#_bookmark0) [1](#_bookmark0)

[二. 输入验证与输出净化](#_bookmark1) [1](#_bookmark1)

[三. 身份验证与权限控制](#_bookmark2) [2](#_bookmark2)

[四. 文件及资源管理](#_bookmark3) [3](#_bookmark3)

[五. 会话管理与通信安全](#_bookmark4) [3](#_bookmark4)

[六. 异常处理与日志审计](#_bookmark5) [4](#_bookmark5)

[七. 敏感数据加密与保护](#_bookmark6) [5](#_bookmark6)

[八. 业务安全](#_bookmark7) [6](#_bookmark7)

# 一. 通用安全开发要求

SC-1-1 使用“经测试和可信的平台/框架代码”开发应用程序。

SC-1-2 应用系统应当保持对所依赖的框架、第三方组件的更新，以避免出现已知漏洞。

SC-1-3 应用程序应避免于页面（比如：HTML、JavaScript）中包含技术性注释语句、功能说明或解释等信息。

SC-1-4 应用系统上线前，应删除相应的测试内容，包括但不限于：测试页面、测试用例、测试代码、控制台输出等。

SC-1-5 应用系统部署后，应删除默认部署页面，禁止留存 SVN/Git 相关文件、备份文件等。

SC-1-6 如果应用软件部署在客户端，例如移动 APP，应使用混淆、签名、加固等措施防止逆向获取源代码。

# 二. 输入验证与输出净化

SC-2-1 应用系统应对所有输入的参数进行合法性、合理性验证，拒绝接受验证失败的数据，包括

但不仅限于验证数据的类型、长度、格式、范围和内容等。说明：

* 对于输入数据范围可确定的场景，合法性检测建议使用“白名单”的方式（比如：日期、身份 证号、银行卡号、手机号、数字等可明确格式的数据，须在服务器端验证格式是否正确）。
* 对于输入数据范围不确定的场景，合法性检测可采用“黑名单”的方式（比如：在可自由输入 的文本框过滤或转义 SQL 关键字、HTML 标签、XML 标签、单引号、双引号、路径字符、换行符、空字节等）。根据实际情况设置黑白名单，避免影响业务正常使用。

SC-2-2 应用系统应对所有输出到客户端、操作系统、Web 页面等位置的数据进行编码或过滤净化，避免潜在危险字符，导致安全问题发生，包括但不仅限于：SQL 注入漏洞、XSS 漏洞、命令注入漏洞等。

说明：常见危险字符如\、'、"、..、/、\r、\n、<、>、^、|、!、`、\*、(、)、&、;、 -、:、%等， 应当在服务器端进行安全过滤或转义编码。

# 三. 身份验证与权限控制

SC-3-1 密码输入界面应采取安全保护措施，包括但不限于：不以明文形式显示密码、利用图形验

证码防止暴力猜解攻击等。

SC-3-2 密码的创建应当符合复杂度检测要求。

包括但不限于以下要求：静态密码长度至少 8 位，至少应同时包含大小写字母、数字和特殊字符这三类字符中的两类，密码不能与用户名相同，避免密码中包含公司特征密码等等。

SC-3-3 所有的身份验证过程应在可信任环境中执行，且在每次用户登录时进行身份验证。避免仅在客户端而非服务器端执行身份验证。

SC-3-4 服务端应对验证尝试的频率进行限制，连续多次登录失败的，应当对其进行限制。说明：

* 限制同一 IP 地址能够进行验证尝试的频率和次数。在超过尝试次数后，封禁 IP 地址一段时间，防止攻击者持续进行暴力破解。
* 限制同一帐号能够进行验证尝试的频率和次数。在超过尝试次数后，锁定账户一段时间，防止攻击者持续进行暴力破解。

SC-3-5 在用户执行关键或者不可逆的操作，如交易时、修改密码之前，应再次验证用户身份，以减少不安全会话带来的损失，防范跨站请求伪造攻击。

SC-3-6 遵循最小权限原则进行权限管理的设计和实现，将许可权限尽可能地细化，建议使用细粒度权限管理的访问控制。

SC-3-7 隶属于用户个人的页面或者功能必须进行权限控制校验。

说明：防止水平越权漏洞，避免任意访问、修改、删除他人的数据，比如查看他人的私信内容、修改他人的订单。

SC-3-8 确保只有授权的用户才能访问秘密数据或敏感数据。

说明：通常，这类数据包括：与用户信息或应用软件自身安全密切相关的状态信息、文件或其他资源、受保护的URL、受保护的功能、直接对象引用、应用程序数据以及与安全相关的配置信息等。避免未授权访问。

# 四. 文件及资源管理

SC-4-1 在上传文件前应对用户身份进行验证，并对上传文件进行验证和控制。

包括但不仅限于：

* 应对上传文件的文件类型扩展名进行验证，还应通过检查文件报头信息的方式，验证上传文档是否是满足业务需要的文件类型，避免攻击者上传恶意文件。
* 应对上传文件进行重命名（比如：保存文件的命名格式为: 随机值.白名单[index]）。
* 文件上传后不应向用户返回文件保存的绝对路径，避免泄露服务器信息。
* 取消上传目录下文件的执行权限。

SC-4-2 在下载文件前应对用户身份进行验证，并对下载文件进行验证和控制。包括但不仅限于：

* 验证文件的真实路径是否在允许下载范围内。
* 验证下载文件名是否含有“../”（Linux 服务器）和“..\” (Windows 服务器)，防止路径遍历导致的任意文件下载。

SC-4-3 应正确释放资源，及时释放系统资源，禁止再次释放已释放的资源，确保释放资源前完全清除敏感信息。

SC-4-4 发生异常时，应及时回收并释放系统资源。

# 五. 会话管理与通信安全

SC-5-1 会话标识应当采用随机且唯一的不可预测的散列值，具备足够强度（比如：128 位），使

用服务器或者框架的会话管理控制，确保会话标识符的随机性，避免暴力散列攻击。

SC-5-2 用户登录成功后，服务端应更新会话标识符，或者增加 IP 等其他判断标志，并周期性地使旧会话标识符失效。

说明：这可以缓解那些原标识符被获得的特定会话劫持情况。

SC-5-2 不应在 URL、错误信息或日志中暴露会话标识符。说明：

* 敏感操作应当采用POST 请求。
* 不要将会话标识符以 GET 参数进行传递，会记录在服务器访问日志中。SC-5-3 应设置会话令牌有效期，建议公网系统会话有效时间不超过 30 分钟。SC-5-4 用户注销时应当立即清理当前用户会话。

SC-5-5 在通信过程中，对敏感信息（如：密码等）应进行加密并使用加密通道传输。使用安全的协议，禁止使用SSL v2 和 v3，建议使用 TLS v1.2。

建议：为所有要求身份验证的访问内容和所有其他的敏感信息提供 TLS 连接。

# 六. 异常处理与日志审计

SC-6-1 精确捕获异常，并对捕获的异常进行恰当的处理，避免在捕获异常后不做任何处理。

SC-6-2 当异常发生时，不要将系统产生的异常信息直接反馈给用户。

包括但不限于：系统的详细信息、会话标识、帐号信息、调试或堆栈跟踪信息、源代码片段等。

SC-6-3 应创建默认的错误页面，使用通用的错误消息提示，防止攻击者从出错页面中得到一些敏感的系统信息。

SC-6-4 不要在日志中保存敏感信息（比如：密码和个人隐私数据）。对日志记录的内容进行审核，防止将关键级敏感信息写入日志。

说明：日志记录中不得以任何形式保存密码、密钥等高敏感性数据，其他敏感级信息仅应在脱敏或加密的前提下保存。

SC-6-5 使用信息摘要算法以验证日志记录的完整性。

SC-6-6 对日志中的特殊元素进行过滤和验证。确保日志记录中的不可信数据，不会在查看界面或者运行软件时以代码的形式被执行。

说明：

* \r、\n 等换行符的录入，可能会造成日志内容换行，从而篡改日志内容。
* 如果日志中存在恶意代码，在页面读取展示时，可能会造成恶意代码执行，如 XSS 攻击、一句话木马等。

# 七. 敏感数据加密与保护

SC-7-1 保护重要秘密信息免受未授权访问。

包括但不仅限于：

* 明确应用系统中的敏感数据、隐私数据的范围，以及有权访问这些数据的用户范围。
* 对数据的授权访问遵循最小权限原则。

SC-7-2 用户敏感信息（比如：姓名、密码、手机号、银行卡号、身份证号等），在页面显示和传输过程中应注意脱敏和加密，数据库中也应当进行加密存储，避免数据库端信息泄露。

SC-7-3 用于访问系统资源的系统用户、密码、密钥必须加密保存在配置文件中，同时要求配置参数要统一集中管理，避免同一个参数在多个配置文件中保存。

SC-7-4 代码中不应当出现硬编码的敏感信息，如用户名、密码、IP 地址、公司邮箱等。

SC-7-5 数据库中存储的用户密码等鉴别信息，应使用不可逆的加密算法（比如：国密 SM3、SHA-512），并进行加盐（Salted）处理，然后将盐和密码哈希值一起保存在数据库中，或通过加密机完成加密。

说明：

推荐如下加密算法选择：

* 非对称加密算法：SM2、RSA
* 对称加密算法：SM4、3DES、AES
* 散列算法：SM3、SHA256

SC-7-6 使用安全的随机数生成器，避免将具有密码学弱点的伪随机数生成器用于加密场景。

# 八. 业务安全

SC-8-1 在使用平台资源，譬如短信、邮件、电话、下单、支付，必须实现正确的防重放的机制，

如数量限制、接口请求频率限制、验证码人机校验，避免被滥刷而导致资损。

说明：如注册时发送验证码到手机，如果没有限制次数和频率，那么可以利用此功能骚扰到其它用户，并造成短信平台资源浪费。

SC-8-2 在调用接口时，应该采取 API 签名机制，防止数据在传输过程中被篡改。如支付订单等重要且敏感的数据，应当采取数字签名等方式，保证数据完整性。