CCS 03.060
A11

Q/NXB

宁夏银行开发安全场景设计企业标准

Q/6401000NXYH-003-2021

宁夏银行开发安全场景设计标准

Develop security scenario design standards of NingXia-Bank

企业标准信息公共服务平台
公开 2021年07月30日 15点56分

2021－07－30 发布　　2021－08－1 实施

宁夏银行股份有限公司　　发布

目　　次

企业标准信息公共服务平台
公开 2021年07月30日 15点56分

前　　言

本标准依据 GB-T1.1-2020 规则起草。

本标准由宁夏银行股份有限公司提出并归口。

本标准起草单位：宁夏银行股份有限公司。

本标准主要起草人：耿道武、马宁、马颖涛、王颖武、白铁钢。

企业标准信息公共服务平台
公开 2021年07月30日 15点56分

引　　言

为提高宁夏银行应用系统安全运行水平，逐步建立安全开发管控体系与技术规范体系，以规划开发的应用系统为目标，从项目建设伊始即同步展开安全规划、设计和投入，将系统安全整合到系统建设的整个生命周期，从安全设计、开发到安全测试、部署安全、运行安全进行整体的规划，使得宁夏银行业务系统后期安全运营更有规划和条理，制定了宁夏银行开发安全场景设计标准。

本标准内容涉及通讯安全需求、应用安全需求、移动客户端安全需求三个方面。

宁夏银行开发安全场景设计企业标准

1. 范围

本标准规定了宁夏银行股份有限公司（以下简称“宁夏银行”）应用系统开发安全场景设计时，在通讯安全、应用安全、移动客户端安全等方面应满足的规范要求。

本标准适用于本版本发布之日宁夏银行股份有限公司的开发安全场景设计。

2. 规范性引用文件

本文件没有规范性引用文件。

3. 术语与定义

下列术语和定义适用于本文件。

3.1 通讯安全需求

通讯安全需求是指数据在通信传输时，需要使用HTTPS加密传输协议的安全需求。

3.2 应用安全需求

应用安全需求是指在身份认证、访问控制、数据安全、信息泄露、框架安全、中间件安全、第三方软件安全等方面的安全需求。

3.3 移动客户端安全需求

移动客户端安全需求是指在反编译、异常捕获、异常信息泄露、敏感信息安全、日志访问权限控制、APP最小化、软件加壳、APP完整性检查、自签名证书验证、Activity权限控制、service权限控制、Broadcast Receiver权限控制、Content Provider权限控制、Android Activity防劫持、日志记录功能开启、日志类型记录、时间戳要求、软键盘保护等方面的安全需求。

4 场景设计规范

4.1 通讯安全场景设计

4.1.1 需求说明

数据在通信传输时，应使用HTTPS加密传输协议。

4.1.2 适用范围

互联网交易类和涉及第三方通信（可信第三方除外，如与银联、人行的通讯）。

4.1.3 设计规范

- 客户端与服务器之间所有经过认证的连接都需要使用不低于TLS 1.0安全级别的加密通讯方式。禁用SSL 3.0及之前的版本。
- 完整性校验码密钥长度不低于128位。
- 加密密钥长度不应低于128位。
- 用于产生签名的RSA密钥长度不应低于1024位。

4.2 应用安全场景设计

4.2.1 身份认证类

4.2.1.1 安全需求一

4.2.1.1.1 需求说明

身份认证时，应防止身份鉴别信息被暴力猜解。注册时，应防止恶意批量注册。

4.2.1.1.2 设计规范

- 登录过程应有图形验证码保护，防止自动化程序暴力猜解。**例如：**客户端使用图片验证码，图片验证码在成功使用一次后，服务端立即重置，防止重复使用。
- 身份认证过程中对于用户名错误和密码错误提示应相同，降低账号、密码被猜解的风险；或在用户注册、登录时增加验证码，防止通过程序自动枚举账户。**例如：**在登录时，身份认证失败，给出通用的错误提示： 账号或密码错误 ，不透露用户ID是否存在等信息。
- 在用户注册、用户登录中增加对同一IP地址尝试次数的限制。
- 限制连续登录失败次数。**例如：**在2分钟内登录失败5次以上则限制登录用户，服务端记录登录失败尝试次数，当超过“5”次时，锁定用户，解锁方式为：次日自动解锁和管理员手动解锁。

4.2.1.2 安全需求二

4.2.1.2.1 需求说明

用户更改手机号码时，系统应对用户的身份进行验证。

4.2.1.2.2 设计规范

- 通过向旧手机号码发送短信验证码，以验证用户身份。

- 通过除手机号以外的其他认证方式，如邮件、银行卡号、身份证号等多因素认证方式，以验证用户身份。

4.2.1.3 安全需求三

4.2.1.3.1 需求说明

身份认证时，系统应防范重放攻击。

4.2.1.3.2 设计规范

- 应采取有效措施防范登录操作的重放攻击。如在登录交互过程提交的认证数据中增加服务器生成的随机信息成分。**例如：**使用随机token。

4.2.1.4 安全需求四

4.2.1.4.1 需求说明

身份认证时，图片验证码设计应复杂有效。

4.2.1.4.2 设计规范

- 验证码复杂度应符合要求，防止OCR工具自动识别。**例如：**验证码使用一些线条或一些不规则的字符组成，字符黏连、变形，增加OCR工具识别难度，这些字符通常同时包含字母和数字。
- 验证码在使用过一次后自动刷新且使用后立即失效，验证码在服务端进行校验之后，同时在服务端立刻更新，避免被二次使用。
- 验证码明文不应被传送到客户端（如：页面、Cookie及DOM对象等），验证码明文禁止在响应包中进行回显或者返回到客户端存储中。
- 验证码应在被保护的操作进行前来验证，验证码校验和要保护的操作在一次交互内完成，后台逻辑上先进行验证码校验再进行其他操作。
- 先校验验证码，再检查用户名，最后比对密码的密文。**例如：**用户认证的表单提交后，首先校验验证码是否有效，并更新服务器端存储的验证码，再校验用户名是否合规，是否存在，最后对比用户提交的密码加密后的密文与数据库中的密码密文是否一致。

4.2.1.5 安全需求五

4.2.1.5.1 需求说明

身份认证时，手机验证码应当复杂随机，防止被线性遍历猜解。

4.2.1.5.2 设计规范

- 手机动态密码应随机产生。
- 手机动态密码长度应不少于4位，由数字和字母组成。
- 手机动态密码应具有有效时间，最长不超过5分钟，超过有效时间未使用应立即作废。**例如：**将手机动态密码及其生成时间存于服务器端，当用户提交动态密码时，该密码的提交时间与生成时间之间的时间差大于300秒，则将该密码作废，提示用户重新获取动态密码。

4.2.1.6 安全需求六

4.2.1.6.1 需求说明

手机验证码应当在服务端正确实现，防止被短信息定向无限制发送。

4.2.1.6.2 设计规范

- 短信验证码的接收手机号码应从服务端查询获得，不以前端提交的手机号为发送目标。即发送登录短信验证码时，应当校验接收短信的手机号码是否存储在数据库中；若存在，则在发送短信验证码时，应当从服务器端通过查询账号信息直接获得，不使用前端提交的手机号码。
- 限制验证码发送间隔。判断当前系统时间与发送验证码时间的时间差是否小于发送间隔，是则不发送；判断验证码是否在有效期内，是则不发送。

4.2.1.7 安全求需七

4.2.1.7.1 需求说明

密码应该保证复杂度，防止弱口令猜解。

4.2.1.7.2 设计规范

- 密码长度至少6位，最多不超过16位。
- 应用各级管理员的口令应在12位以上。
- 在客户设置密码时，不能使用简单密码。**例如：**通过正则表达式匹配密码字符串，判断其复杂度，密码不能为连续的数字，如123456，密码不能为相同的数字或字母，如111111。
- 密码复杂度应大写字母、小写字母、数字、特殊字符这四类中至少两类。可用正则表达式匹配密码字符串，判断其复杂度。
- 更改密码时，要求新密码与原密码不相同。执行密码更新操作之前，对比新密码密文与数据库中的旧密码密文，相同则提示用户更换密码，不相同则执行更新操作。
- 如有初始密码，首次登录时应强制客户修改初始密码。判断登录密码是否为初始密码，是则提示用户修改密码，并跳转到密码修改页面，密码未修改，则不能进行其它操作。

4.2.1.8 安全求需八

4.2.1.8.1 需求说明

登录密码与交易密码不能相同。

4.2.1.8.2 设计规范

修改交易密码时，须验证用户身份，然后判断提交交易密码密文与数据库中查询到的该账户的登录密文是否相同，相同则提示用户更换密码，不同则执行更新操作。

4.2.1.9 安全求需九

4.2.1.9.1 需求说明

禁止明文显示密码。

4.2.1.9.2 设计规范

禁止明文显示密码，应使用相同位数的同一特殊字符（例如*和#）代替。

4.2.2 访问控制类

4.2.2.1 安全求需一

4.2.2.1.1 需求说明

管理员用户密码安全策略。

4.2.2.1.2 设计规范

管理员用户只能通过系统后台修改、重置密码。**例如：**修改和重置密码时，判断用户类型，禁止管理员用户通过Http请求修改和重置密码。

4.2.2.2 安全求需二

4.2.2.2.1 需求说明

在服务端会话中完成对权限的访问控制。

4.2.2.2.2 设计规范

防止非授权访问及操作涉及权限的操作，应以会话中保存的用户标识为依据进行操作。**例如：**对所有涉及权限划分的操作都应严格检查后再确定用户是否可以操作请求的数据。开发人员应避免只注重功能权限的划分，即不同角色的人员只能看到并使用自己拥有的功能菜单，而忽视了对于数据权限的判断。并且对关键数据应进行加密或增加校验，防止通过修改关键字段来越权操作他人数据。

4.2.2.3 安全求需三

4.2.2.3.1 需求说明

系统应建立完善的会话管理机制。

4.2.2.3.2 设计规范

- 系统应通过身份鉴别方式实现会话的建立。
- 应具备用户注销功能，用户注销时应清理了当前用户会话。
- 系统身份鉴别过程会话标识验证生成仅在服务器端实现，不应由第三方提供会话标识。
- 系统在通过身份鉴别后必须分配新的会话标识，不能使用未认证前的标识，即会话不能被复用。
- 系统应不容许使用相同的user ID进行同时重复的登录，即用户账号不能被复用。
- 系统应指定用户会话的空闲时间，当超出此时间，用户的所有操作必须重新由系统进行身份鉴别，否则自动终止会话，空闲时间建议为10分钟。
- 会话标识应当采用随机且唯一的不可预测的散列值，会话标识字符串推荐128位长，避免暴力散列攻击。
- 系统应禁止通过Get参数传递会话标识值，即使是在客户端Cookie被禁止的情况下也应如此。
- 系统使用Cookie时，应设置Cookie的Secure属性及Httponly属性 ，不在非SSL通道中传输cookie值。

4.2.2.4 安全求需四

4.2.2.4.1 需求说明

系统中对于关键业务操作应确保使用浏览器“后退”功能无法回到上一步操作界面。

4.2.2.4.2 设计规范

关键业务操作应确保使用浏览器“后退”功能无法回到上一步操作界面。**例如：**设置页面禁止缓存response.setHeader("Pragma", "No-cache");response.setHeader("Cache-Control", "no-cache");response.setDateHeader("Expires", 0)。

4.2.2.5 安全求需五

4.2.2.5.1 需求说明

系统应对非认证的请求进行重定向。

4.2.2.5.2 设计规范

系统应能够识别非认证的地址访问并拒绝访问或者重定向到其他提示页面，并对请求提示警告。

4.2.2.6 安全求需六

4.2.2.6.1 需求说明

系统应当建立完善的交易验证机制，避免出现流程跳跃、不完整验证。

4.2.2.6.2 设计规范

系统应建立完善的交易验证机制，每次处理的客户信息均以服务器端数据为准，并对客户请求指令的逻辑顺序进行合理控制。

4.2.3 数据安全类

4.2.3.1 安全求需一

4.2.3.1.1 需求说明

系统应当在输入过程中对数据进行校验、过滤。

4.2.3.1.2 设计规范

- 在客户端应对客户信息进行格式和类型检测。对于可枚举格式的数据输入,可通过正则表达式的方式对数据进行白名单检测,限制用户输入的数据格式，如检测中国大陆手机号码、检测日期格式、检测邮编、检测邮箱、检测url、检测中国大陆身份证号等（JS实现）。
- 在服务器端应对关键客户信息进行格式和类型的检测。
- 系统应当对关键数据进行参数合理性校验。**例如：**判断参数是否应该为负数。
- 应通过全局过滤器或过滤函数对SQL注入、命令注入及跨站脚本攻击常见的敏感字符进行过滤或全角转换。从sql注入、xss、路径遍历、xml、http响应分割、LDAP注入、CRLF注入、命令注入等和输入输出相关的漏洞中分析出利用的字符，并进行过滤。

4.2.3.2 安全求需二

4.2.3.2.1 需求说明

系统应当对上传功能进行有效验证。

4.2.3.2.2 设计规范

以白名单形式指定允许上传的扩展名；以黑名单形式指定禁止上传的文件名；二次渲染图片；将图片保存到无法解析目录下。**例如：**在服务器端对上传的文件类型和格式进行检查，对不符合要求的文件禁止上传和存储。对上传的文件进行重命名，并禁用文件上传目录的脚本执行权限。

4.2.3.3 安全求需三

4.2.3.3.1 需求说明

系统应当对下载功能或者文件访问进行有效防护。

4.2.3.3.2 设计规范

针对下载模块应过滤掉../敏感字符。**例如：**Web应用应限定文件下载目录，并且在程序中过滤客户端提交的路径参数中的特殊字符，例如：./、../等。如非必要，避免使用客户端提交的文件绝对路径，可采用服务器端配置的路径或建立下载文件ID与文件路径的对应表，由客户端传递文件ID，服务器通过ID获取文件路径。

4.2.3.4 安全求需四

4.2.3.4.1 需求说明

系统应当对页面来源进行检测，严格限制页面间的前后访问继承关系，对于重要过程可通过标识的方法进行控制。

4.2.3.4.2 设计规范

系统限制页面访问来源时，通过referer信息进行判断，也可针对关键操作应采用token与表单一起提交方式避免CSRF并确保token生成安全性。**例如：**表单生成的同时创建token并存储在服务器端，token和表单一起返回到客户端。表单提交后，先校验token的有效性，并更新服务器端存储的token值，确保token的一次有效性和唯一性。

4.2.3.5 安全求需五

4.2.3.5.1 需求说明

系统应当对输出进行转义处理。

4.2.3.5.2 设计规范

返回数据时，应对htmL、JavaScript相关的标签进行转义处理。**例如：**通过使用ESAPI对输入输出的数据进行HTML实体转义。

4.2.3.6 安全求需六

4.2.3.6.1 需求说明

数据传输过程中应当采用加密技术。

4.2.3.6.2 设计规范

- 系统中应采用密码技术支持的保密性保护机制或其他具有相应安全强度的手段提供保密性保护。**例如：**对敏感或者重要数据进行加密保护，不得使用弱加密算法，包括MD5等算法。系统中涉及的用户功能页面，不应使用用户ID等易猜测的明文作为页面的URL信息。
- 系统中不得使用已经被证明为不安全的算法或者自定义不安全算法进行用户数据加密。
- 身份鉴别信息应使用端对端加密技术加密传输。客户端与服务器之间所有经过认证的连接都需要使用不低于TLS 1.0安全级别的加密通讯方式。
- 和第三方对接，进行实时交易、对账文件传输、清算文件传输等，应使用加密技术防止敏感信息泄露或被篡改，推荐使用对称加密技术。对称加密算法以SM4为主，也可以同第三方一样采用3DES、AES等其他对称算法。

4.2.3.7 安全求需七

4.2.3.7.1 需求说明

系统应对数据的完整性进行校验。

4.2.3.7.2 设计规范

- 系统中应对用户关键数据进行完整性校验，建议采用明文加上随机码或时间戳因子再加密或散列方式。**例如：**MD5应用防篡改方案或者SM2算法防篡改方案。
- 系统中应指定必须检测的异常情况，包括数据篡改、数据替换、数据不可恢复的排序改变、数据重放、不完全的数据以及其它完整性错误，针对异常情况应采取一些动作，包括忽略用户数据、重新请求数据、告知授权管理员等操作。

4.2.3.8 安全求需八

4.2.3.8.1 需求说明

用户数据需要安全存储。

4.2.3.8.2 设计规范

- 用户数据存储时，身份属性信息必须进行不可逆的加密存储。**例如：**使用安全哈希算法如SM3及更高级别的算法对身份属性信息进行加密存储。
- 用户数据存储时，对于关键数据可存储于服务器端会话中，以便后续数据鉴别使用。**例如：**用户登录认证成功后，将通过用户名和密码查询到的用户ID、用户角色等关键数据，存储在服务器端的Session中，如需通过这些关键数据查询其他信息时，使用服务器端存储的session值，不使用客户端提交的参数。

4.2.4 防信息泄露类

4.2.4.1 安全求需一

4.2.4.1.1 需求说明

系统应当对输出的内容进行脱敏，避免信息泄露，不向用户提示过多技术细节。

4.2.4.1.2 设计规范

系统应当避免向用户提示过多的技术细节，特别是错误信息。**例如：**不向客户端返回与程序相关的技术信息、调试信息、异常信息等。

4.2.4.2 安全求需二

4.2.4.2.1 需求说明

系统应当屏蔽在客户端显示的用户资源。

4.2.4.2.2 设计规范

- 显示客户身份证件信息时，应屏蔽部分关键内容。**例如：**敏感信息返回客户端之前，在服务器端进行脱敏处理，即将字符串中的部分字符转换为“*”等字符，再返回客户端。
- 不容许客户端浏览器启用缓存用户名、密码等账户信息。**例如：**登录操作不使用记住密码的功能设计；或采用加密控件。

4.2.4.3 安全求需三

4.2.4.3.1 需求说明

系统应当屏蔽在WEB页面显示的用户资源。

4.2.4.3.2 设计规范

- 系统应避免WEB目录残留可下载文件，并对外提供访问。
- 系统应避免管理功能页面面向普通用户。**例如：**设置基于URL的访问控制，对所有管理功能页面的URL访问，校验其权限，拒绝普通用户的请求。
- 禁止在Web应用程序错误提示中包含详细信息，不向客户显示调试信息。
- 系统发生异常、错误时，应采取统一的错误提示页面。**例如：**在web.xml中指定错误处理页面，可以通过“异常类型”或“错误码”来指定错误处理页面。

```
<error-page>
    <error-code>404</error-code>
    <location>/error404.jsp</location>
</error-page>
-----------------------------
<error-page>
    <exception-type>java.lang.Exception<exception-type>
```

<location>/exception.jsp<location>
</error-page>

- 系统应避免于页面中，如Html中包含技术性的注释语句，功能说明解释等信息。系统上线前删除html和js等文件中的注释语句。

4.2.4.4 安全求需四

4.2.4.4.1 需求说明

系统应当正确使用文件路径。

4.2.4.4.2 设计规范

- 系统应避免通过构造文件路径的方式直接查看文件，禁止目录列表浏览，防止网上银行站点重要数据被未授权下载。
- 系统应避免向客户端暴露服务器端绝对路径。
- 系统应合理设置robots.txt，防止来自于搜索引擎的信息泄漏。**例如：**对于敏感目录，采用通配符的方式，编写，如禁止爬取admin目录，应写成“Disallow: /ad”，避免完整的目录名暴露后台管理地址。

4.2.4.5 安全求需五

4.2.4.5.1 需求说明

系统应避免残留垃圾页面面向普通用户。

4.2.4.5.2 设计规范

- 在系统上线前，应删除Web目录下所有测试脚本、程序。
- 如果在生产服务器上保留部分与Web应用程序无关的文件，应为其创建单独的目录，使其与Web应用程序隔离，并对此目录进行严格的访问控制。

4.2.5 框架安全类

4.2.5.1 需求说明

开源框架spring、hibernate、dubbox、zookeeper等建议部署最新版本。

4.2.5.2 设计规范

开源框架在系统集成阶段应确定所需的版本号，建议部署最新版本的框架。

4.2.6 中间件安全类

4.2.6.1 需求说明

中间件建议部署最新版本。

4.2.6.2 设计规范

在系统集成阶段应确定所需中间件的版本号，建议部署最新版本的中间件。

4.2.7 第三方软件类

4.2.7.1 需求说明

系统应关注第三方软件的安全需求。

4.2.7.2 设计规范

系统设计阶段应该关注第三方软件的安全需求。**例如：**

Redis安全需求：

- 不建议使用 root 权限启动 redis 服务。
- 对 redis 访问启用密码认证，并且添加 IP 访问限制；
- 尽可能不对公网直接开放 SSH 服务。

4.3 移动客户端安全场景设计

4.3.1 反编译

4.3.1.1 需求说明

使用客户端加固，防止被反编译。

4.3.1.2 设计规范

需要对程序代码进行混淆保护防止反编译。例如：为了解决Android二次打包以及信息泄露的风险，Android应用应通过加壳进行安全防护，可使用市场上成熟的加壳方案，来防止内存信息泄露、防键盘记录木马、防截屏木马、防二次打包、防范Activity劫持、以及代码混淆等。

4.3.2 异常捕获

4.3.2.1 需求说明

系统需要正确处理异常。

4.3.2.2 设计规范

系统应对需要捕获的异常进行处理，禁止直接忽略异常。

4.3.3 异常信息防泄露

4.3.3.1 需求说明

系统需要正确处理异常，防止信息泄露。

4.3.3.2 设计规范

系统应对出现的异常进行处理，避免直接向客户端直接展示异常信息。系统上线前关闭日志打印的相关代码。

4.3.4 敏感信息安全

4.3.4.1 需求说明

客户端应按要求进行敏感信息确认，并将敏感信息放入到服务器中存储，防止敏感信息泄露。

4.3.4.2 设计规范

- 客户端应用中禁止以任何形式保存用户密码信息。包括但不局限于SQLite、文件、SharedPreferences（Android）、NSUserDefaults（iOS）、keyChain（iOS）、LocalStorage（HTML5）。
- 客户端应用中需要存储用户密码以外的敏感数据时，应进行加密保存，例如：使用SQLCipher对SQLite数据库进行加密。

4.3.5 日志安全

4.3.5.1 需求说明

系统需要记录日志，用于追溯事件，防止越权查看日志。

4.3.5.2 设计规范

- 系统应开启日志记录功能。
- 系统应确保在重要行为发生时有相应的日志记录，以便捕获系统异常信息。
- 日志记录中必须明确包含时间戳。
- 控制其他应用程序对日志的访问权限。

4.3.6 权限最小化

4.3.6.1 需求说明

移动客户端遵循授权最小化原则。

4.3.6.2 设计规范

只应授予移动客户端必须的功能和权限。不必要的权限不应在AndroidManifest.xml文件中添加。

4.3.7 加壳防护

4.3.7.1 需求说明

移动客户端应加壳防止二次打包和分析。

4.3.7.2 设计规范

应对移动客户端进行软件加壳，防止内存信息泄露、防范键盘记录木马、防范截屏木马、防范二次打包。

4.3.8 防篡改

4.3.8.1 需求说明

系统需要防止移动客户端被篡改。

4.3.8.2 设计规范

- 系统需要对移动客户端进行完整性校验，确保未被篡改。
- 使用自签名的SSL证书防篡改，移动客户端应使用公钥锁定方式进行HTTPS连接。

4.3.9 软键盘保护

4.3.9.1 需求说明

移动客户端应该提供自主开发的软键盘功能，而不调用系统默认软键盘。

4.3.9.2 设计规范

为了保证用户输入的保密性，移动客户端应该提供自主开发的软键盘功能，在输入敏感信息时，使用自定义软键盘而不调用系统默认软键盘。软键盘设计应满足以下几点：

- 密码软键盘分布应随机化，每次启动软键盘时键位均不相同。
- 输入字符在内存保存时，应加密处理。

- 密码软键盘按下时，屏蔽回显功能，可使用声音或者震动的方式代替回显，避免被截屏木马记录。

4.3.10 第三方组件安全

4.3.10.1 需求说明

系统需要使用安全的第三方组件。

4.3.10.2 设计规范

系统调用第三方代码或库文件，应确保其来源的可靠性。

4.3.11 移动客户端更新

4.3.11.1 需求说明

移动客户端应关注软件更新安全，防止下载到恶意程序。

4.3.11.2 设计规范

- 系统应设计并提供安全更新的功能。
- 系统应采用动态加载更新补丁的方式，需校验补丁程序的签名证书是否合法，防止补丁被替换并植入恶意代码。

4.3.12 安卓客户端安全需求

4.3.12.1 需求说明

安卓客户端应对Activity、service、Broadcast Receiver、Content Provider等进行权限控制，并具备Android Activity防劫持功能。

4.3.12.2 设计规范

- 安卓客户端外部调用Activity时，应严格设置调用接口的访问权限。安卓客户端应严格设置Activity的访问权限，不需要外部应用调用时，应关闭导出接口，将exported属性设置为false。
- 安卓客户端需要外部调用Service时，应严格设置调用接口的访问权限。安卓客户端应严格设置service的访问权限，不需要外部应用调用时，应关闭导出接口，将exported属性设置为false。
- 安卓客户端使用Broadcast Receiver时，应严格设置访问权限，避免接收到攻击者发送的恶意广播，执行恶意指令。防范恶意应用向本应用发送广播在AndroidManifest.xml中注册receiver的时候将export设置为false
- 安卓客户端使用Content Provider向外部应用提供数据访问时，应严格设置访问权限。由于API level 在17以下的所有应用的“android:exported”属性默认值都为true，因此如果应用的Content

Provider不必要导出，建议显式设置注册的Content Provider组件的“android:exported”属性为false。

- 安卓客户端应具备Activity防劫持功能，防止攻击者通过启用后台木马服务，切换给用户钓鱼登录界面，泄漏用户输入的帐号和密码等敏感信息。

企业标准信息公共服务平台
公开 2021年07月30日 15点56分