|  |  |
| --- | --- |
| ICS | 35.080 |
| CCS | L 77 |

|  |
| --- |
| 2201 |

长春市地方标准

DB XX/T XXXX—XXXX

政务应用系统开发安全规范

Security specification for government application system development

XXXX - XX - XX发布

XXXX - XX - XX实施

长春市市场监督管理局  发布

目次

[前言 II](#_Toc175414031)

[1 范围 1](#_Toc175414032)

[2 规范性引用文件 1](#_Toc175414033)

[3 术语和定义 1](#_Toc175414034)

[4 缩略语 2](#_Toc175414035)

[5 需求阶段安全规范 2](#_Toc175414036)

[6 设计阶段安全规范 3](#_Toc175414037)

[6.1 环境安全要求 3](#_Toc175414038)

[6.2 身份识别和认证 3](#_Toc175414039)

[6.3 访问控制和授权 4](#_Toc175414040)

[6.4 输入数据验证 5](#_Toc175414041)

[6.5 数据处理控制 6](#_Toc175414042)

[6.6 敏感数据的保护 6](#_Toc175414043)

[6.7 配置管理 7](#_Toc175414044)

[6.8 异常处理 7](#_Toc175414045)

[7 开发阶段安全规范 7](#_Toc175414046)

[7.1 应用开发环境的安全要求 7](#_Toc175414047)

[7.2 应用开发文档的安全要求 7](#_Toc175414048)

[7.3 应用开发的代码安全要求 7](#_Toc175414049)

[8 测试阶段安全规范 8](#_Toc175414050)

[9 部署和维护阶段安全规范 9](#_Toc175414051)

[9.1 安全运维要求 9](#_Toc175414052)

[9.2 应用系统上线投产的安全要求 9](#_Toc175414053)

[9.3 应用系统代码安全审计要求 9](#_Toc175414054)

[9.4 应用系统web安全测试评价要求 9](#_Toc175414055)

[参考文献 10](#_Toc175414056)

1. 前言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由长春市政务服务和数字化建设管理局提出并归口。

本文件起草单位：杭州安恒信息技术股份有限公司、长春市标准研究院（长春市WTO/TBT咨询中心）。

本文件起草人：刘竞雄、柳羽辉、邓贺、戚志军、李聪、冷皓、李凌岳。

政务应用系统开发安全规范

* 1. 范围

本文件规范了长春市所属政务系统应用开发安全能力及其他相关要求。

本文件适用于指导长春市所属政务系统应用开发过程中的安全规范需求、设计、开发、测试、部署和维护等相关工作。

* 1. 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25069-2022 信息安全技术 术语

GB/T 22239-2019 信息安全技术 网络安全等级保护基本要求

GB/T 35273-2020 信息安全技术 个人信息安全规范

* 1. 术语和定义

GB/T25069-2022、GB/T 43632-2024、GB/T 43435-2023和GB/T43698-2024界定的以及下列术语和定义适用于本文件。

软件开发（这个及下面的所有术语都要把英文翻译加上）

是根据用户要求建造出软件系统或者系统中的软件部分的过程。软件开发是一项包括需求捕捉、需求分析、设计、实现和测试的系统工程。

需求分析

是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。

敏感信息

指不当使用或未经授权被人接触或修改会不利于国家利益计划的实行或不利于个人依法享有的个人隐私权的所有信息。

需求分析阶段

指收集和理解用户需求，确保最终软件产品能够完全满足用户需求的过程。

设计阶段

指在设计阶段完成需求分析后，开发团队根据收集到的需求，设计出软件架构、数据结构、用户界面、算法和模块等各个组件，创建稳定、可预测的系统结构的过程。

编码阶段

指在编码阶段设计阶段结束后，开发团队根据设计文档，遵循一定的编程规范和最佳实践，用编程语言编写软件代码的过程。

测试阶段

指在测试阶段编码阶段完成后，为发现并修复软件中问题所开展单元测试（测试单个组件或函数），集成测试（测试多个组件或函数的协作），系统测试（测试整个系统）以及验收测试（验证软件是否满足用户需求）的过程。

部署和维护阶段

指软件开发完成并交付给用户使用，为确保软件的稳定性和可靠性，修复软件问题，增加新功能满足用户需求，对软件产品进行维护和更新的过程。

* 1. 缩略语

下列缩略语适用于本文件。

GDPR：一般数据保护条例（General Data Protection Regulation）

SSL/TLS：传输层安全协议（Secure Sockets Layer/ Transport Layer Security）

SSH：安全外壳协议（Secure Shell）

LDAP：轻型目录访问协议（Lightweight Directory Access Protocol）

Cookie：储存在用户本地终端上的数据（Cookie）

HTTPS：超文本传输安全协议（Hypertext Transfer Protocol Secure）

RBAC：基于角色的访问控制（Role-Based Access Control）

ABAC：基于属性的访问控制（Attribute Based Access Control）

URL：统一资源定位系统（Uniform Resource Locator）

SQL：结构化查询语言（Structured Query Language）

Nosql：非关系型的数据库（Not Only SQL）

CSRF ：跨站请求伪造（Cross-site request forgery）

XSS：跨站脚本攻击（Cross Site Scripting）

* 1. 需求阶段安全规范

应用软件在软件需求阶段应确认业务功能满足如下内容：

1. 安全需求分析：对系统的安全需求进行全面的分析和确定，包括身份验证、授权、数据保护、审计跟踪等方面的需求。
2. 敏感信息识别：确定系统中的敏感信息，如个人身份信息、财务数据等，并明确其在系统中的流动和处理方式。
3. 安全性能要求：确定系统在安全性能方面的要求，如认证和授权的响应速度、数据加密解密的效率等。
4. 合规性要求：确保系统满足相关法律法规和行业标准的安全合规性要求，如GDPR等。
5. 访问控制要求：确定系统对不同用户角色的访问控制策略，包括用户身份验证、权限分配和访问控制规则等。
6. 数据保护要求：确定系统对数据的加密、传输、存储和销毁等方面的保护要求，包括加密算法、密钥管理、数据备份和灾难恢复等。
7. 审计要求：确定系统需要记录和审计的安全事件和操作，包括登录记录、权限变更记录、数据访问记录等。
8. 安全培训要求：确定系统用户和管理人员需要接受的安全培训内容和频率，以提高其安全意识和技能。
9. 应急响应要求：确定系统对安全事件的应急响应计划和流程，包括漏洞修复、应急演练、事件响应和恢复等。
10. 供应商安全要求：确定与外部供应商合作时对其安全性的要求，包括安全评估、合同条款、监督和审查等。
    1. 设计阶段安全规范
       1. 环境安全要求

应用软件在设计阶段应确认：

1. 应用系统服务器硬件物理安全

物理访问控制：应确保服务器设备存放在安全的物理环境中，只有授权人员可以接触和操作服务器设备。

防盗措施：应部署物理安全设备，如门禁系统、监控摄像头等，以防止未经授权的人员进入服务器设备所在的区域。

1. 应用系统服务器操作系统安全

及时更新补丁：应定期更新服务器操作系统和相关软件的补丁，及时修补已知漏洞，降低系统被攻击的风险。

强化身份验证：应配置强密码策略、禁用不必要的服务和端口，限制远程访问等，提高服务器操作系统的安全性。

1. 应用系统数据库的安全

访问控制：应设置严格的数据库访问权限，只允许授权用户或应用程序访问数据库，并限制其权限范围。

加密存储：应采用加密算法对数据库中的敏感数据进行加密存储，提高数据的保密性。

备份和恢复：应定期备份数据库，并确保备份数据的安全存储，以便在发生数据丢失或损坏时进行快速恢复。

1. 应用系统的存储安全

数据加密：应对存储在服务器或云端的敏感数据进行加密保护，确保即使数据泄露也不会导致敏感信息泄露。

访问控制：应限制对存储系统的访问权限，只允许授权人员或系统进行访问，并监控访问行为。

1. 应用系统网络通信安全

加密通信：应采用安全的通信协议，如SSL/TLS、SSH等，确保网络通信过程中的数据机密性和完整性。

防火墙和入侵检测系统：应部署防火墙和入侵检测系统，监控网络流量和检测恶意行为，及时阻止攻击。

* + 1. 身份识别和认证

身份识别和认证过程设计应包括：

1. 用户身份验证：
   1. 用户名和密码：应为用户提供唯一的用户名和相应的密码进行身份验证。密码应该采用安全的存储方式，如哈希算法加盐存储等。
   2. 多因素认证：应结合密码与其他因素，如手机验证码、硬件令牌或生物识别等，提高身份验证的安全性。
   3. 身份认证凭据：应加密并通过安全的通信通道传输，以保护认证的凭据（如Cookie）安全。
   4. 限制认证凭据的有效期，以减少攻击的威胁，并定期更新凭据以保持安全性。
2. 用户账号管理：
   1. 注册与注销：应提供用户注册功能，并确保注册过程中的输入合法性和安全性。同时，应该允许用户注销账号并删除相关信息。
   2. 密码策略：应制定密码复杂性策略，要求用户设置安全性较高的密码，并定期提示用户更新密码。
   3. 登录策略：应在多次尝试登陆后，锁定账户一定时间并记录日志
   4. 密码重置：应提供用户密码重置功能，通常通过验证注册时提供的备用邮箱、手机号码或安全问题等方式进行。
3. 会话管理：
   1. 登录会话管理：应记录用户登录状态并维持会话，确保用户在登录状态下可以持续访问系统，并在一定时间内自动登出以保护用户安全。
   2. 会话安全：应采用安全的会话标识和管理机制，防止会话劫持和会话固定等攻击。
4. 权限管理：
   1. 用户角色与权限：应将用户分配到不同的角色，并授予相应的权限，以控制用户对系统资源的访问和操作。
   2. 细粒度权限控制：应支持对每个用户或角色进行细粒度的权限控制，以实现最小权限原则，确保用户只能访问其需要的资源。
5. 安全传输：

使用安全协议：应在用户身份验证和会话管理过程中使用安全的传输协议，如HTTPS，确保用户的身份和数据在传输过程中受到保护。

1. 安全审计：
   1. 记录日志：应记录用户的身份验证和操作行为，并定期审计日志以检测异常行为和安全事件。
   2. 报告与分析：应生成安全审计报告，对系统的安全状况进行分析和评估，及时发现潜在的安全风险。

（之前层次有问题，看我改的对不对）

* + 1. 访问控制和授权

访问控制和授权的开发设计应包括：

1. 应优先考虑采用统一的认证和授权平台。认证过程需要进行加密，密钥长度不低于128位，以确保信息安全。
2. 需要对用户的权限进行设计，包括限制用户对系统级资源的访问，如文件、文件夹、注册表项、LDAP对象、数据库对象、日志文件等。
3. 应用系统使用的数据库帐号应为普通权限帐号，只允许访问特定的数据库。此外，应降低应用系统启动进程的权限，并且合理设计用户权限的颗粒度，满足授权最小原则。
4. 输入数据方面，应采用输入复核或其他输入检查方式，如边界检查、限制数据输入字段的范围和类型等，以防止输入错误。常见的输入错误包括输入过长、输入数据字段中有非法字符、输入为空或不完整、输入值超过上限或下限等。通过这些方式，可以有效地保护应用系统免受输入错误的影响，提高系统的稳定性和安全性。
5. 用户权限分配和管理功能：需设计系统的读、写、执行权限，以及查看、配置、修改、删除、登录、运行等权限。还需要设计数据访问范围的权限，以及应用功能模块的使用权限。
6. 身份验证：应确认用户的身份是否合法的过程，通常包括用户名和密码验证、双因素认证等方式。
7. 授权：应对用户访问特定资源或执行特定操作进行授权，例如基于角色的访问控制（RBAC）或基于属性的访问控制（ABAC）。
8. 访问控制列表：应定义用户或系统进程可以访问特定资源以及对资源的操作权限。
9. 会话管理：应管理用户与应用程序之间的会话状态，包括会话创建、终止和过期等操作，以确保会话的安全性和完整性。
10. API 访问控制：应对应用程序接口（API）的访问进行控制，包括限制调用频率、使用 API 密钥进行身份验证等。
11. 数据访问控制：应控制用户对数据的访问权限，包括对数据库、文件系统或其他数据存储系统的访问权限管理。
12. 安全标头：应在HTTP报文响应中包含安全标头，以保护应用程序免受常见的网络攻击，如点击劫持、内容嗅探等。
13. 安全日志：应记录应用程序的安全事件和行为，以监控和调查潜在的安全威胁或违规行为。
    * 1. 输入数据验证

数据验证应在服务器端进行，数据验证开发设计应包括：

1. 数据类型验证：
   1. 确保输入数据符合预期的数据类型，如整数、浮点数、字符串等。
   2. 对于敏感数据，如日期、时间等，应确保格式正确并符合预期。
2. 长度验证：
   1. 针对字符串输入，限制其长度，防止缓冲区溢出等安全问题。
   2. 对于数字输入，可以限制其最小值和最大值，以确保数据在合理范围内。
3. 格式验证：
   1. 对于特定格式的数据，如邮件地址、URL、电话号码等，应进行格式验证，确保数据符合标准格式。
   2. 使用正则表达式或专门的验证库来验证格式。
4. 范围验证：
   1. 对于数值型数据，确保其在预期的范围内，如年龄必须大于0，金额不能为负等。
   2. 对于日期和时间，确保其在合理的时间范围内。
5. 唯一性验证：
   1. 对于需要唯一性的数据，如用户名、邮箱等，确保其在系统中唯一，避免重复数据造成的混乱。
   2. 在数据库层面使用唯一索引或约束来确保唯一性。
6. 安全性验证：

针对用户输入的敏感数据，如 SQL 注入、XSS 攻击等，采取相应的安全措施，如参数化查询、HTML 标签转义等，防止恶意输入导致的安全问题。

1. 业务逻辑验证：
   1. 根据业务需求对输入数据进行额外的业务逻辑验证，确保数据符合业务规则。
   2. 针对不同的业务场景，可能需要不同的验证规则，例如对于电子商务系统，验证商品库存是否足够等。
2. 错误处理和反馈：
   1. 当输入数据验证失败时，向用户提供清晰的错误提示，说明输入数据不符合要求，并指导用户如何正确输入。
   2. 日志应记录验证过程和失败原因，以进行排查和分析问题。

（之前层次有问题，看我改的对不对）

* + 1. 数据处理控制

数据处理控制开发设计应包括：

1. 数据收集合法性：

应确保收集的数据来源合法，遵循用户授权和同意的原则，不得擅自收集用户未授权的数据。

1. 数据处理合规性：

应对于敏感信息的收集，如个人身份信息、金融信息等，需符合相关法律法规的规定，并遵循数据最小化原则，只收集必要的信息。在数据处理过程中，应采取措施确保数据的完整性、可用性和保密性，防止数据泄露、篡改或丢失。

1. 数据处理透明度：

应对数据处理操作进行记录和审计，确保数据处理过程的透明度和可追溯性。

应提供用户数据访问和管理的途径，使用户可以随时了解自己的数据被如何处理和使用。

1. 数据处理风险管理：

应定期进行数据处理风险评估，识别潜在的安全风险和威胁。

应制定相应的风险管理措施，及时修复安全漏洞和弱点，降低数据处理风险的影响。

原始数据应进行检错和校验操作，保证原始数据的正确性和完整性。并保留处理日志。

1. 数据处理管理：

应用开发过程应根据数据的类型、数据的处理方式、数据的安全性要求，与其它接口有关的敏感等级、数据相关业务应用的重要性程度来进行数据处理过程的安全性设计。

* + 1. 敏感数据的保护

敏感数据管理应：

1. 最小化数据收集：应只收集应用所需的最小量的敏感数据。避免过渡收集和数据关联带来的数据泄露风险。
2. 数据分类：应对数据进行分类，区分敏感数据和非敏感数据。针对敏感数据采取更严格的保护措施，如加密、访问控制等。
3. 加密存储：应使用强加密算法对敏感数据进行加密存储，包括数据库中存储的数据、本地存储的数据以及在网络传输过程中的数据。
4. 访问控制：应制订访问控制策略，限制对敏感数据的访问权限，只允许授权用户或系统访问。
5. 安全传输：应在数据在网络传输过程中采用安全的传输协议，如SSL/TLS、HTTPS等，确保数据传输过程中的机密性和完整性。
6. 安全存储：应采用安全的存储方式，如加密文件系统、安全存储服务等，确保数据在存储时的安全性。
   * 1. 配置管理

应用系统配置管理开发设计应包括：

1. 权限控制：应由经过授权的操作员和管理员进行，应采用强身份认证手段，例如双因素认证等方式确认。应禁止远程配置，以减少安全风险。
2. 基于角色的授权策略：配置管理应基于角色（操作员和管理员等）进行授权，各角色只能访问其所需的配置管理功能，以降低误操作和滥用权限的风险。
3. 权限最小原则：对于系统内部帐号，如进程帐号、服务帐号等，应遵循权限最小原则进行权限配置，只分配完成任务所需最小权限，以减少潜在安全风险。
4. 配置信息的安全存储：应确保配置信息本身的安全性。常见的存储应用程序配置信息的方法包括基于文本的配置文件、注册表和数据库。
5. 应禁止使用Web空间配置文件，避免因服务器漏洞而导致配置文件被下载或篡改。
6. 应限制对包含加密数据的注册表项、文件或表的访问权限，以防止未授权访问或篡改配置信息。
7. 应加强配置管理，禁止明文形式存储配置信息，配置信息（如数据库连接字符串、API 密钥等）应采取安全方式进行存储和使用，禁止敏感信息在代码中以硬编码方式使用。
   * 1. 异常处理

应在应用系统异常时应避免向客户端泄漏应用程序内部敏感信息，不应暴露包括函数名以及调试信息（出问题的行数，堆栈信息，判断逻辑等）。

* + - 1. 应在日志中详细记录错误信息，但不记录密码或其他敏感数据。
  1. 开发阶段安全规范
     1. 应用开发环境的安全要求

开发环境应划定专门的网络安全区域，应禁止或限制访问互联网互联网。

存储项目文档、代码的服务器应建立访问控制管理、备份制度，定期验证备份有效性，备份应有在线备份和离线备份两种以上方式。

应严格管理访问项目文档和代码的权限，做好记录并定期审计访问记录。

应对项目文档和代码进行版本管理和控制。

应对开发终端进行必要的安全限制措施，避免数据、文档、代码的泄露。

* + 1. 应用开发文档的安全要求

应在开发过程中明确安全设计需求，并在文档中明确说明：

1. 需求说明书中应明确描述用户的安全需求。
2. 设计文档中应明确针对用户安全需求提出的设计方案，方案应进行评审并通过。
3. 测试计划或测试方案中应有安全性测试方案，应开展安全性测试并保留测试记录。
4. 应用开发文档应参照安全分类要求设定安全级别，读者范围。
5. 应在应用软件开发过程中确认安全配置要求、流程和基线，并统一进行安全配置管理。
   * 1. 应用开发的代码安全要求

应对输入进行验证与过滤，对所有函数入口参数进行合法性和准确性的检查，验证输入数据的合法性和安全性。规范并定义输入数据的字符长度，进行数据合法性检查，拒绝不满足安全标准的数据输入。

应使用参数化查询或预处理语句来执行数据库操作，以防止 SQL 注入、NoSQL 注入等类型的攻击。

应实施有效的登录验证机制，禁止开发人员为普通软件进程分配系统特权帐号，确保登录密码满足安全要求，并禁止以明文方式传递和存储用户密码。确保软件能自动拒绝所有登录请求当登录验证机制不可正常运作时。

应加强对用户会话管理，包括安全存储会话令牌、限制会话过期时间、防止会话固定攻击等。WEB服务应确保用户注销会话后，包含敏感信息的页面不能通过使用浏览器的回退按钮再次显示。

应加强数据存储安全，对敏感数据使用加密算法进行加密，并采取数据库加密、文件加密等方式进行数据存储。

应禁止开发人员未经授权和验证地使用第三方代码。

应对用户上传的文件，实施严格的文件类型检查和限制，应检查和验证是否存在恶意代码。

应对从用户输入、URL 参数等处接收的数据进行适当的转义和过滤，以防止 XSS 攻击。

应实施 CSRF 保护机制。

应使用安全协议（如 HTTPS）以保护数据在客户端和服务器之间的传输，防止窃听和中间人攻击。

应及时更新与漏洞修复，更新应用程序的依赖库和框架并及时修复已知的漏洞和安全问题。

应加强代码设计与开发规范管理，代码设计应简单清晰，遵循规范的变量、函数命名和程序书写格式，以提升设计过程的安全性和可读性。

应代码进行版本控制，注释无用的语句，确保代码的可用性。禁止程序员非授权修改代码，以及在程序中添加隐藏“恶意”的代码，防止与应用系统相关的程序员对系统的非授权修改。

应加强用户体验并提供安全提示，在所有警告或提示对话窗口中应使用准确、明了的描述性语言，并提供有关帮助链接。

应隐藏敏感信息，如在输入密码等敏感信息时使用星号来代替输入的字符。

* 1. 测试阶段安全规范

应用系统的测试过程应符合以下要求：

1. 应用系统的安全性测试应制订测试方案，明确测试的各项要求和测试用例。
2. 测试前应明确测试目的，所有测试人员应熟悉测试的目的和意图。
3. 应明确测试的安全要点、测试参与人员、测试流程，并编写测试计划。
4. 测试环境的硬件、软件环境和基础架构应模拟真实环境。
5. 测试数据不能直接使用真实数据进行测试，测试数据应通过编制、脱敏、混淆等方式获得。
6. 系统测试和验收测试应尽可能接近实际运行数据的测试数据，应避免使用含有个人信息的业务数据库。应开展脱敏、混淆等方式对测试数据进行处理。当将运行数据用于测试目的时，应采取措施保护运行数据。
7. 在与其他系统的交互性测试中，应充分考虑对其他系统的影响，选择适当的时间、方法。
8. 测试过程使用的后门、用户名及口令不可保存并同步发布到生产版本，应同步开展安全检查，确保生产部署版本不存在测试账号。
9. 确保测试环境的安全。应将测试环境与开发环境、生产环境相隔离，避免测试工作对业务的影响。
10. 应详细记录测试过程发生的各个事件，列出测试过程中发现的问题。
11. 应根据测试的过程和测试结果，提出被测试系统、测试过程等方面的改进说明。
12. 应确保测试用例、测试内容和测试结果的保密性。
    1. 部署和维护阶段安全规范
       1. 安全运维要求

在维护阶段应按照GB/T 36626-2018 《信息安全技术 信息系统安全运维管理指南》的要求开展。如涉及远程运维应按照GB/T 39837-2021 《信息技术 远程运维 技术参考模型》的要求开展。

1. 应建立安全运维体系，建立安全运维模型，做好安全运维分类，明确安全运维活动要素和管理原则。
2. 应制定安全运维策略并定期开展评审。
3. 应组建安全运维团队，明确安全运维的角色和责任。
4. 应建立安全运维规程。加强资产、日志、访问、密码、漏洞和备份管理，建立安全事件管理及响应机制。
5. 应采用安全运维支撑系统，以辅助安全运维规程的实现。
   * 1. 应用系统上线投产的安全要求

应用系统上线应建立上线验收标准和流程，上线过程应制订规划，上线前应完成功能、性能和安全测试，明确所需资源和人员：

1. 应明确系统应用系统上线所软件、硬件、网络和安全防护的要求。
2. 应明确应用系统上线对相关部门、人员的职责及角色。
3. 应明确应用系统上线的时间进度。
4. 应明确应用系统上线方案并经主管部门审核批准。
5. 应明确应用系统上线可能存在的风险，并制订预案及回退方案。制订错误恢复和重新启动流程，编制意外事故处理计划并准备必要的软硬件资源。
6. 上线前应对应用系统上线环境（硬件环境、操作系统、物理环境）进行确认。
7. 应对应用系统变更涉及的系统在变更前进行备份及确认。
8. 应用系统上线变更过程应进行记录。
9. 应用系统上线应使用最小账号权限。
10. 其它资源的详细清单。
    * 1. 应用系统代码安全审计要求

应按照GB∕T 39412-2020 《信息安全技术 代码安全审计规范》中安全功能缺陷审计、代码实现安全缺陷审计、资源使用安全缺陷审计、环境安全缺陷审计等几个方面开展。

* + 1. 应用系统web安全测试评价要求

应按照GB/T 37931-2019 《信息安全技术 Web应用安全检测系统安全技术要求和测试评价方法》中的安全功能要求、自身安全要求、安全保障要求等几个方面开展测评工作。

参考文献

[1] GB/T 30998-2014 信息技术 软件安全保障规范

[2] GB/T 36626-2018 信息安全技术 信息系统安全运维管理指南

[3] GB/T 39837-2021 信息技术 远程运维 技术参考模型

[4] GB/T 39412-2020 信息安全技术 代码安全审计规范

[5] GB/T 37931-2019 信息安全技术 Web应用安全检测系统安全技术要求和测试评价方法

[6] GB/T 36626-2018 信息安全技术 信息系统安全运维管理指南

[7] GB/T 39837-2021 信息技术 远程运维 技术参考模型

