源代码审计报告

目录

一. 概述

1.1 源代码审计概述

源代码审计工作通过分析当前应用系统的源代码，熟悉业务系统，从应用系统结构方面检查其各模块和功能之间的关联、权限验证等内容；从安全性方面检查其脆弱性和缺陷。在明确当前安全现状和需求的情况下，对下一步的编码安全规范性建设有重大的意义。

源代码审计工作利用一定的编程规范和标准，针对应用程序源代码，从结构、脆弱性以及缺陷等方面进行审查，以发现当前应用程序中存在的安全缺陷以及代码的规范性缺陷。

审核目的

本次源代码审计工作是通过对当前系统各模块的源代码进行审查，以检查代码在程序编写上可能引起的安全性和脆弱性问题。

审核依据

本次源代码审计工作主要突出代码编写的缺陷和脆弱性，以 OWASP TOP 10 2010 为检查依据，针对 OWASP 统计的问题作重点检查。

点击打开文档 OWASP TOP 10 2010

审计范围

根据 XX 给出的代码，对其 WEB 应用作脆弱性和缺陷、以及结构上的检查。通过了解业务系统，确定重点检查模块以及重要文件，提供可行性的解决方法。

审计方法

通过白盒（代码审计）的方式检查应用系统的安全性，白盒测试所采用的方法是工具审查+人工确认 +人工抽取代码检查，依照 OWASP 2010 TOP 10 所披露的脆弱性，根据业务流来检查目标系统的脆弱性、缺陷以及结构上的问题。

本次源代码审计分为三个阶段：

信息收集

此阶段中，源代码审计人员熟悉待审计 WEB 应用的结构设计、功能模块，并与客户相关人员商议、协调审计重点及源代码提供等方面的信息。

代码安全性分析

此阶段中，源代码审计人员会使用工具对源代码的脆弱性和安全缺陷进行初步的分析，然后根据客户关注的重点对部分代码进行手工审计，主要包含以下内容：

输入/输出验证 。SQL 注入、跨站脚本、拒绝服务攻击，对上传文件的控制等因为未能较好的控制用户提交的内容造成的问题；

安全功能 。请求的参数没有限制范围导致信息泄露， Cookie 超时机制和有效域控制，权限控制、日志审计等方面的内容；

程序异常处理 。忽略处理的异常、 异常处理不恰当造成的信息泄露或是不便于进行错误定位等问题；

代码规范性检查

此阶段中，源代码审计人员主要是利用一些代码规范检查工具对网站各功能模块的代码进行合规性检查，主要目的在于提高代码质量，使其更符合编码规范的要求，主要包括以下内容：

代码质量 。例如对象错误或不适合调用导致程序未能按预期的方式执行， 功能缺失；类成员与其封装类同名，变量赋值后不使用等；

封装 。多余的注释信息、调试信息问题导致应用系统信息暴露，错误的变量声明等。

API 滥用 。例如调用非本单位直接控制的资源、对象过于频繁调用、直接调用空对象导致系统资源消耗过大或是程序执行效率低下等问题。

1.2 项目概述

在 XX 及 WEB 应用开发单位 XX 公司相关人员的协调与配合下， **公司安全测试小组于 XXXX 年 XX 月 XX 日至 XX 日对 XX 应用进行了源代码审计工作。在此期间内， **公司安全测试小组利用各种主流的代码审计工具以及手工检查等方式对网站主要功能模块的源代码进行了安全性及规范性检查，发现了源代码中存在的一些脆弱性、合规性问题及缺陷。

本文档即为 **公司安全测试小组在进行代码审计工作完成后所提交的报告资料，用于对 XXWEB 应用的安全状况从代码层面作出分析和建议。

**公司代码审计服务是经过授权的，也是有时间限制的。

二. 审核对象

2.1 应用列表

本次代码审核的对象包括：

基本信息	
应用系统名称	XX WEB 应用
语言类型	□ ASP □ ASP.NET （VB） □ ASP.NET （C#） □ PHP □ JSP （JAVA ） □ 其它 _____

2.2 参与人员

参与人员	工作职责	联系方式
XXX		
XXX		
XXX		

2.3 审计工具

工具一	
工具名称	XXXX
工具用途	
相关信息	

工具二	
工具名称	XXXX

工具用途	
相关信息	

… …

三. 现状分析

XX 门户网站是由 XX 公司开发的基于 XXX 语言的网站， 主要功能有产品及解决方案、 合作伙伴、客户支持、工作机会、 eDM 以及贯穿多个模块的讨论组。根据模块的不同进行访问权限的控制。

整个网站采用唯一的访问入口 default.aspx ，所有模块均由系统根据权限和参数来进行控制。系统用户根据权限的不同分为超级管理员、模块管理员和用户三个级别。前台用户访问使用 HTTP 协议，后台管理员维护使用 HTTPS 协议，以保证通讯安全。

除了产品及解决方案、合作伙伴、讨论组、工作机会、客户支持五个模块进行了定制开发以外，整个网站的基础架构（如用户管理、权限管理、网站安全、文件上传下载等）均采用成熟的平台来构建。因此，最可能出现各种问题的地方也集中在各个定制模块当中，源代码审计的重点也集中在这几部分的代码上。

四. 审计结果

4.1 XX 模块

4.1.1 XXXXXX

编 号	NS-SCA-XXXX-XX
描 述	
潜在威胁	
所在页面	
问题行数	
修改建议	

4.1.2 XXXXXX

编　　号	NS-SCA-XXXX-XX
描　　述	
潜在威胁	
所在页面	
问题行数	
修改建议	

五. 审计结论与建议

5.1 审计结果简评

通过对 XX WEB 应用进行为期 XX 天的源代码审计，我们得出如下结论：

底层平台采用了较为成熟的用户管理、权限控制、模块动态加载及访问控制技术，代码的编写基本符合编码规范的要求。但在部分功能模块上还存在一些问题，需要加于改进，主要体现在以下几个方面：

XXXXXX

… …

XXXXXX

… …

注意事项

5.2 脆弱性和缺陷编程意见

经过本次代码审计，也发现了被检测 WEB 应用存在的一些问题或缺陷，在本节我们会根据我们的经验来提出一些改进意见或建议，供 WEB 应用开发、管理人员参考。这部分内容对于后期的维护和扩展也有一定的指导意义。

永远不要相信用户的输入

用户的输入主要包括以下几类：

WEB 访问请求中 URL 的参数部分；

HTML 表单通过 POST 或 GET 请求提交的数据 ；

在客户端临时保存的数据（也就是 Cookie ）；

数据库查询。

安全功能方面

不要过于信任应用程序访问控制规则；

身份鉴别系统和会话管理可能会被绕过或是被篡改；

存储的敏感信息可能被抽取。

其它：

服务器：安装最新的补丁，降低 WEB 应用运行用户的权限，适当设置应用所在目录的读写权限。

WEB 服务器软件：不要开启目录浏览、写入、脚本资源访问等功能。

错误处理：必须关闭详细错误显示，比较好的处理方式是开启错误重定向功能在出错后重定向到指定页面（如网站首页），并且这个页面不能把异常信息发送给客户端，如：

<customErrors mode="On" defaultRedirect ="Default.aspx " />

代码质量：主要是指可用性、可维护性、运行效率、重复代码量等等指标，高质量的代码不仅易于维护，而且运行效率高，因为当受到拒绝服务攻击时可以有效降低对系统的影响。好的代码依赖于合理的系统架构、优秀的程序编写人员和严谨的工作作风。

5.3 定期进行代码抽样审计

虽然我们在本次代码审计中发现了这些问题，并且相信这些安全隐患能够在短时间内解决。我们仍然建议您定期进行类似的安全抽样审计，保障不断发展的动态网络的持续安全。

5.4 系统上线前进行全面的测试

在网站新上线或是部分功能更新时，建议进行全面的测试，确保无问题后再在正式环境中上线使用。

5.5 制定完善的开发文档

应该为网站制定完善的开发文档，不建议在开发过程中实现开发文档要求以外的功能，应该注重并严格遵守以下几方面内容：

输入输出实现

程序变更准则

修改程序代码准则

程序验证准则

功能需求