**fs.597.com系统技术检测报告**

系统属地： 厦门

系统名称： fs.597.com

域名（或IP地址）： fs.597.com

技术检测报告

经技术检测发现，你单位系统（域名或IP地址:fs.597.com:80存在安全漏洞1个，其中高风险漏洞1个，中风险漏洞0个，低风险漏洞0个，已验证漏洞1个。

**1.检测结果（检测结果包含但不限于以下漏洞）**

**1.1 高危风险漏洞**

**1.1.1 SQL注入漏洞（盲注）**

https://fs.597.com/zhaopin/wenyuan/?q=%E6%96%87%E5%91%98&n=4406:select 4406

**漏洞描述:**

目标存在SQL注入漏洞。SQL注入漏洞是目前互联网最常见也是影响非常广泛的漏洞。由于编写代码时没有对用户输入数据的合法性进行判断，使应用程序存在安全隐患。攻击者通过在应用程序中预先定义好的查询语句结尾加上额外的SQL语句元素，就可以欺骗数据库服务器执行非授权的任意查询。

**1.2 中危风险漏洞**

**1.3 低危风险漏洞**

**2.验证结果**

仅对上述漏洞中威胁较高的漏洞进行验证，请你单位自行验证其他漏洞。

验证地址或请求：

验证截图

请你单位高度重视系统安全保障工作，对系统进行全面检测、整改，有关整改情况及时反馈同级公安机关网安部门。

附件：漏洞危害及整改建议

系统漏洞危害及整改建议

1. SQL注入漏洞（盲注）

*漏洞危害:*

攻击者利用该漏洞可能导致1.网页被篡改2.数据被篡改3. 核心数据被窃取4. 数据库所在服务器被攻击变成傀儡主机

*整改意见:*

若干问题的补救方法在于对用户输入进行清理。 通过验证用户输入未包含危险字符，便可能防止恶意的用户导致应用程序执行计划外的任务，例如：启动任意 SQL 查询、嵌入将在客户端执行的 Javascript 代码、运行各种操作系统命令，等等。 建议过滤出所有以下字符：[1] |（竖线符号）[2] （ 符号）[3];（分号）[4] $（美元符号）[5] %（百分比符号）[6] @（at 符号）[7] （单引号）[8] （引号）[9] \（反斜杠转义单引号）[10] \（反斜杠转义引号）[11] （尖括号）[12] ()（括号）[13] +（加号）[14] CR（回车符，ASCII 0x0d）[15] LF（换行，ASCII 0x0a）[16] ,（逗号）[17] \（反斜杠）以下部分描述各种问题、问题的修订建议以及可能触发这些问题的危险字符： SQL 注入和 SQL 盲注： A. 确保用户输入的值和类型（如 Integer、Date 等）有效，且符合应用程序预期。 B. 利用存储过程，将数据访问抽象化，让用户不直接访问表或视图。当使用存储过程时，请利用 ADO 命令对象来实施它们，以强化变量类型。 C. 清理输入以排除上下文更改符号，例如：[1] （单引号）[2] （引号）[3] \（反斜线转义单引号）[4] \（反斜杠转义引号）[5] )（结束括号）[6] ;（分号）跨站点脚本编制： A. 清理用户输入，并过滤出 JavaScript 代码。我们建议您过滤下列字符：[1] （尖括号）[2] （引号）[3] （单引号）[4] %（百分比符号）[5] ;（分号）[6] ()（括号）[7] （ 符号）[8] +（加号）B. 如果要修订 %00script 变体，请参阅 MS 文章 821349 C. 对于 UTF-7 攻击： [-] 可能的话，建议您施行特定字符集编码（使用 Content-Type 头或 标记）。 HTTP 响应分割：清理用户输入（至少是稍后嵌入在 HTTP 响应中的输入）。 请确保输入未包含恶意的字符，例如：[1] CR（回车符，ASCII 0x0d）[2] LF（换行，ASCII 0x0a）远程命令执行：清理输入以排除对执行操作系统命令有意义的符号，例如：[1] |（竖线符号）[2] （ 符号）[3];（分号）执行 shell 命令： A. 绝不将未检查的用户输入传递给 eval()、open()、sysopen()、system() 之类的 Perl 命令。 B. 确保输入未包含恶意的字符，例如：[1] $（美元符号）[2] %（百分比符号）[3] @（at 符号）XPath 注入：清理输入以排除上下文更改符号，例如：[1] （单引号）[2] （引号） 等LDAP 注入： A. 使用正面验证。字母数字过滤（A..Z,a..z,0..9）适合大部分 LDAP 查询。 B. 应该过滤出或进行转义的特殊 LDAP 字符：[1] 在字符串开头的空格或“#”字符[2] 在字符串结尾的空格字符[3] ,（逗号）[4] +（加号）[5] （引号）[6] \（反斜杠）[7] （尖括号）[8] ;（分号）[9] ()（括号）MX 注入： 应该过滤出特殊 MX 字符：[1] CR（回车符，ASCII 0x0d）[2] LF（换行，ASCII 0x0a）记录伪造：应该过滤出特殊记录字符：[1] CR（回车符，ASCII 0x0d）[2] LF（换行，ASCII 0x0a）[3] BS（退格，ASCII 0x08）ORM 注入： A. 确保用户输入的值和类型（如 Integer、Date 等）有效，且符合应用程序预期。 B. 利用存储过程，将数据访问抽象化，让用户不直接访问表或视图。 C. 使用参数化查询 API D. 清理输入以排除上下文更改符号，例如： (\*)：[1] （单引号）[2] （引号）[3] \（反斜线转义单引号）[4] \（反斜杠转义引号）[5] )（结束括号）[6] ;（分号） 如下一些方法能够防止注入攻击：1.在网页代码中需要对用户输入的数据进行严格过滤。2.部署Web应用防火墙3.对数据库操作进行监控建议过滤用户输入的数据，切记用户的所有输入都要认为是不安全的。