## 验证测试

### 渗透测试

#### 渗透测试过程说明

**1、此次渗透测试所使用的工具如下**：

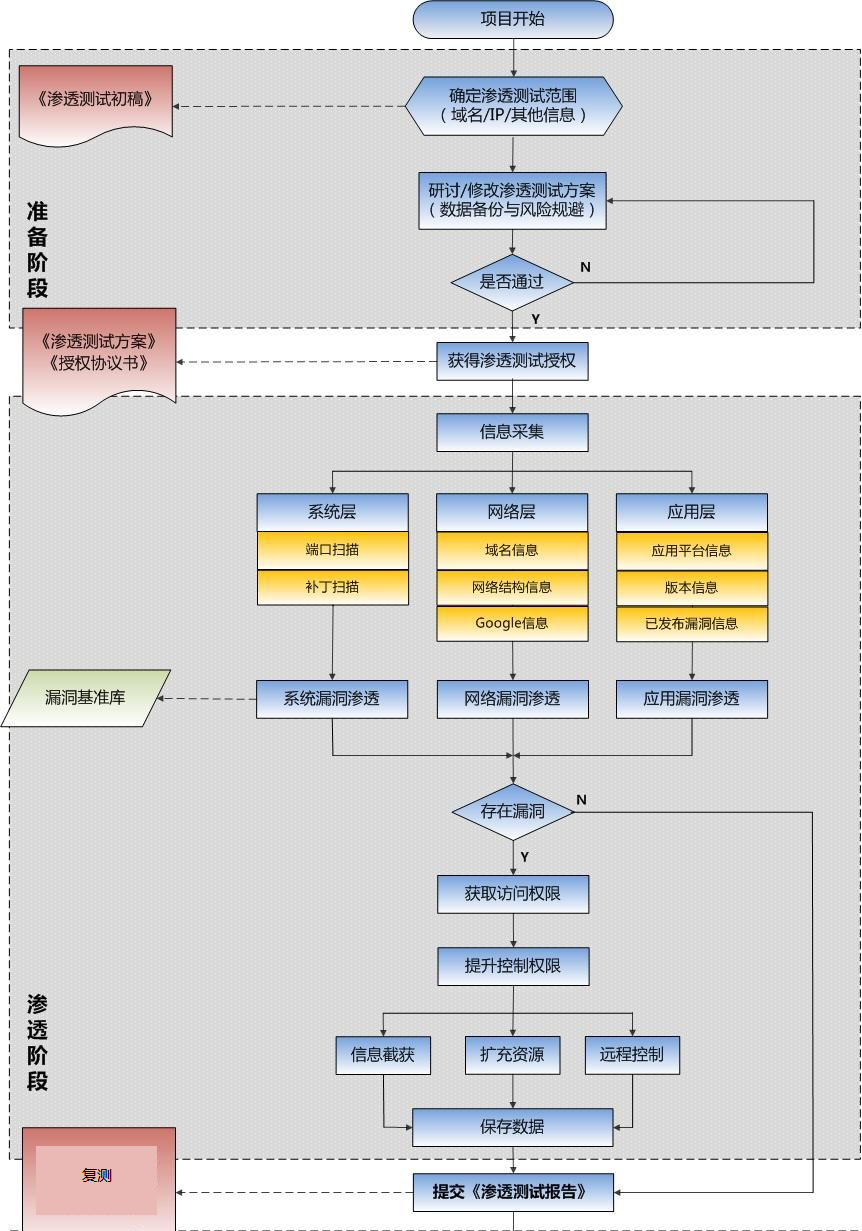
Nmap、Sqlmap、AWVS、Burpsuite、Metasploit、[MasScan](https://www.cesafe.com/7840.html)等。

**2、渗透测试方法：**

1）信息收集:

信息收集是渗透攻击的前提，通过信息收集可以有针对性地制定模拟攻击测试计划，提高模拟攻击的成功率，同时可以有效的降低攻击测试对系统正常运行造成的不利影响。  
 信息收集的方法包括端口扫描、操作系统指纹判别、应用判别、账号扫描、配置判别等。  
 2）端口扫描  
 通过对目标地址的TCP/UDP端口扫描，确定其所开放的服务的数量和类型，这是所有渗透测试的基础。通过端口扫描，可以基本确定一个系统的基本信息，结合安全工程师的经验可以确定其可能存在以及被利用的安全弱点，为进行深层次的渗透提供依据。  
 3）漏洞扫描  
 漏洞扫描通常是指基于漏洞数据库，通过扫描等手段，对指定的远程或者本地计算机系统的安全脆弱性进行检测，发现可利用的漏洞的一种安全检测（渗透攻击）行为。  
 漏洞扫描技术是渗透测试的核心部分，同时漏洞扫描技术也是一类重要的网络安全技术。漏洞扫描技术与防火墙、入侵检测系统相互配合，能够有效提高网络的安全性。通过对漏洞扫描，渗透测试人员能了解到网络安全配置和运行的应用服务，及时发现安全漏洞，一般进行后续攻击。  
 4）口令猜测  
 本阶段将对暴露在公网的所有登陆口进行口令猜解的测试，找出各个系统可能存在的弱口令或易被猜解的口令。猜解成功后将继续对系统进行渗透测试，挖掘嵌套在登录口背后的漏洞、寻找新的突破口以及可能泄漏的敏感信息，并评估相应的危害性。猜解的对象包括：WEB登录口、FTP端口、数据库端口、远程管理端口等。  
 5）脚本测试  
 脚本测试专门针对Web服务器进行。根据最新的技术统计，脚本安全弱点为当前Web系统尤其存在动态内容的Web系统存在的主要比较严重的安全弱点之一。利用脚本相关弱点轻则可以获取系统其他目录的访问权限，重则将有可能取得系统的控制权限。因此对于含有动态页面的Web系统，脚本测试将是必不可少的一个环节。  
 6）权限获取  
 通过初步信息收集分析，存在两种可能性，一种是目标系统存在重大的安全弱点，测试可以直接控制目标系统；另一种是目标系统没有远程重大的安全弱点，但是可以获得普通用户权限，这时可以通过该普通用户权限进一步收集目标系统信息。这样不停的进行信息收集分析、权限提升的结果形成了整个的渗透测试过程。  
 7）其他测试方法  
 渗透测试过程中，根据系统漏洞情况，采取其他测试方法。

**渗透测试流程将采用如下方案开展：**



**图 3‑1渗透测试流程图**

#### 渗透测试问题描述

2023-03-22 08:30至17:30从内网对内蒙古自治区公安厅技术侦察总队 – 北疆云二期云平台综合应用平台系统(http://www.ses.bjy.nm.jz/)开展渗透测试，测试IP为：26.9.27.171。本次测试共发现2个高危漏洞，包括：弱口令、SQL注入漏洞；0个中危漏洞；0个低危漏洞。

以下为高危问题及所造成危害描述：

弱口令:攻击者可利用该漏洞，非法登录他人账户，甚至系统后台账户，进行恶意操作。SQL注入漏洞:攻击者可利用SQL注入获取数据库全部数据，甚至获取服务器权限。

详细渗透测试过程记录描述参见报告附录。

1. 渗透测试结果记录

F.1 弱口令

**漏洞链接**:

http://www.ses.bjy.nm.jz/

**风险程度**:【高危】

**漏洞概述**：

弱口令没有严格和准确的定义，通常认为容易被别人猜测、有规律或易被破解工具破解的口令均为弱口令。

**渗透过程**：



图 F-1

**漏洞危害**：

攻击者可利用该漏洞，非法登录他人账户，甚至系统后台账户，进行恶意操作。

**整改建议**：

1. 不使用空口令或系统缺省默认的口令，因为这些口令众所周知，为典型的弱口令。

2. 口令长度不小于8个字符。

3. 口令不应该为连续的某个字符（例如：AAAAAAAA）或重复某些字符的组合（例如：tzf.tzf.）。

4. 口令应该为以下四类字符的组合，大写字母(A-Z)、小写字母(a-z)、数字(0-9)和特殊字符。

5. 口令中不应包含账户本人有关的信息，如生日、姓名等。

6. 口令不应该为用数字或符号代替某些字母的单词。

7. 口令应该易记且可以快速输入，防止他人从你身后很容易看到你的输入。

8. 至少90天内更换一次口令，防止未被发现的入侵者继续使用该口令。

F.2 SQL注入漏洞

**漏洞链接**:

http://www.ses.bjy.nm.jz/synqc/app/qc

**风险程度**:【高危】

**漏洞概述**：

SQL注入即是指web应用程序对用户输入数据的合法性没有判断或过滤不严，攻击者可以在web应用程序中事先定义好的查询语句的结尾上添加额外的SQL语句，在管理员不知情的情况下实现非法操作，以此来实现欺骗数据库服务器执行非授权的任意查询，从而进一步得到相应的数据信息。

**渗透过程**：

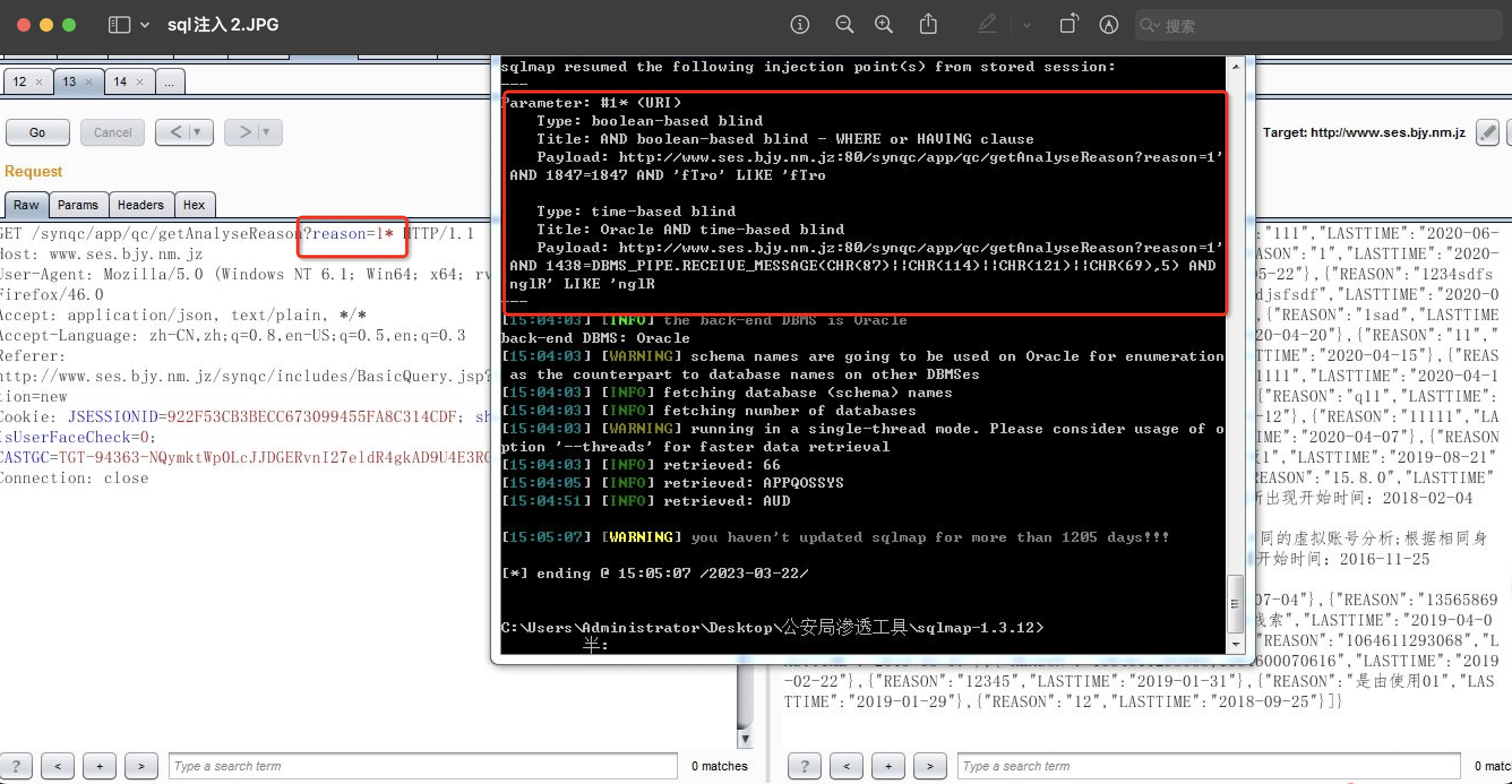


图 F-2

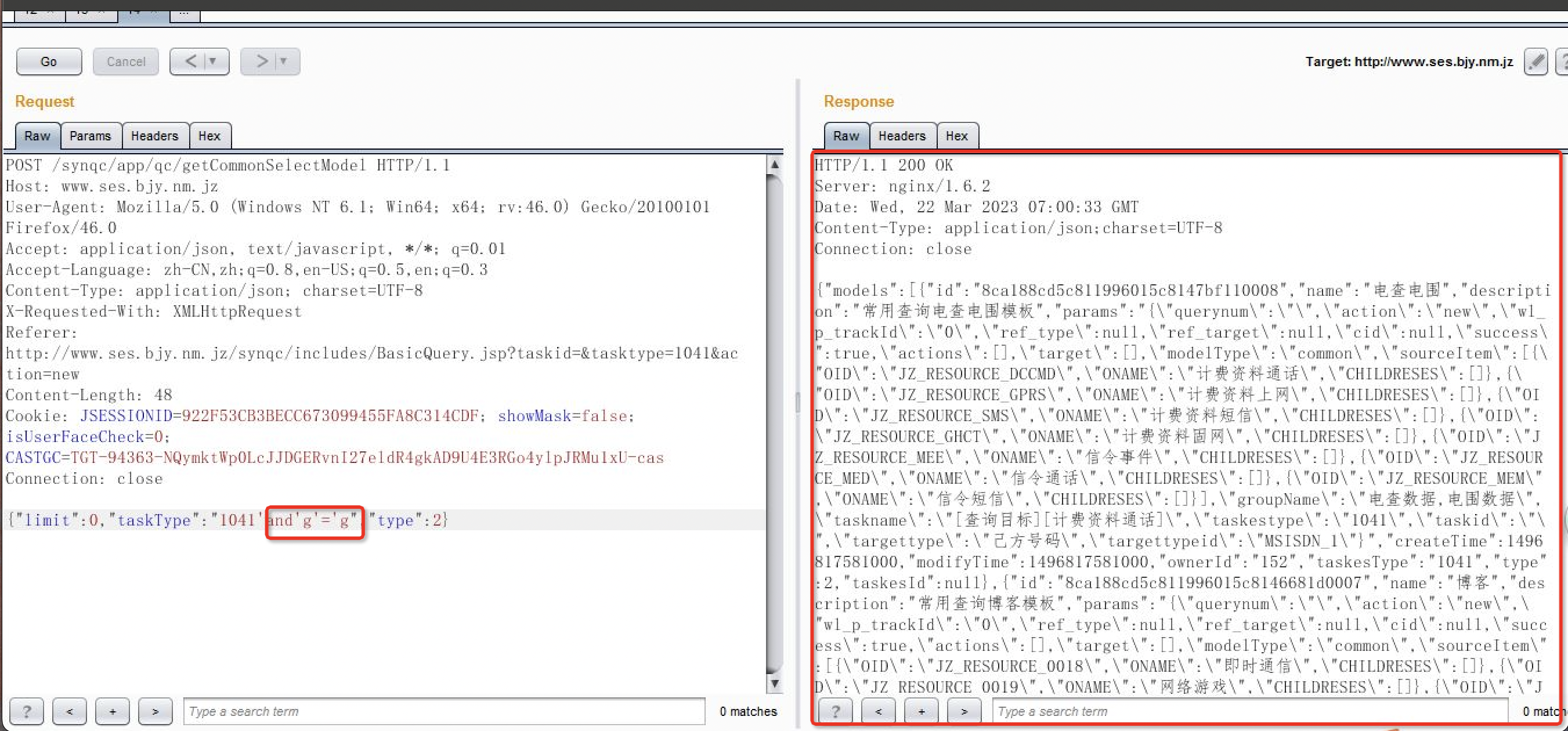


图 F-3

**漏洞危害**：

攻击者可利用SQL注入获取数据库全部数据，甚至获取服务器权限。

**整改建议**：

1. 对用户的输入进行校验，可以通过正则表达式、限制输入长度、对特殊符号进行转义等。

2. 不要使用动态拼接的SQL语句，可以使用参数化的SQL或者直接使用存储过程进行数据的增删改查。

3. 不要使用管理员权限的数据库连接，应为每个应用使用单独权限的数据库连接账号。

4. 不要把敏感信息明文存储，应加密内容或将密码、手机号、身份证号等敏感内容进行脱敏处理。

5. 屏蔽应用程序的原生错误信息，最好使用自定义的错误信息对原始错误信息进行包装、统一处理。