

0x04 输入输出

一、概述

输入输出的安全问题来源于开发人员编码过程中的粗心大意以及应有的安全意识的缺失。这些安全问题对于网站来说是非常严峻的，对数据库，网站管理权限，内网都有巨大威胁。而且利用手法很多，如利用任意文件上传漏洞可直接获取网站shell，利用命令执行漏洞可执行命令反弹shell，利用SQL注入漏洞可查看和修改数据库信息，利用服务端请求伪造（SSRF）漏洞可攻击内网等。因此关于输入输出的安全测试必不可少，并且是重中之重，丝毫不能疏忽和遗漏。以下是对交易所进行测试后列出的详细测试条目，以及一些存在输入输出安全问题的经典案例。

二、测试列表

输入输出安全

- 跨站脚本（XSS）
- 模板注入测试
- HTTP头注入测试
- HTTP参数污染测试
- 不安全的HTTP方法测试
- 服务端请求伪造（SSRF）测试
- 任意文件上传
- SQL注入测试
- XXE实体化测试
- 反序列化漏洞测试
- HTTP请求夹带（smuggling）攻击测试
- 代码注入测试
- 本地文件包含测试
- 远程文件包含测试
- 命令执行注入测试
- 缓冲区溢出测试
- 格式化字符串测试

三、案例分析

跨站脚本（XSS）测试

跨站脚本（XSS）是最常见的web漏洞之一，亦是客户端脚本安全的头号大敌，在各大漏洞提交平台也经常见到XSS的漏洞提交。跨站脚本攻击的危害巨大且可利用处繁多：如XSS钓鱼，Cookie劫持，获取用户真实IP等操作。

零时安全实验室在对某交易所进行测试时，发现在资产充值的付款钱包地址处，可以输入任意字符，并没有对输入的字符进行过滤和编码。初步测试时弹框成功，进一步写入利用XSS获取cookie的脚本。等待一段时间后，成功获取到后台管理员cookie，并以此登录后台成功，其后台可进行任意转账和充值操作，且存在大量用户信息。

下图为测试XSS时截图：

请填写支付信息

付款钱包地址

1<script>alert(1)</script>

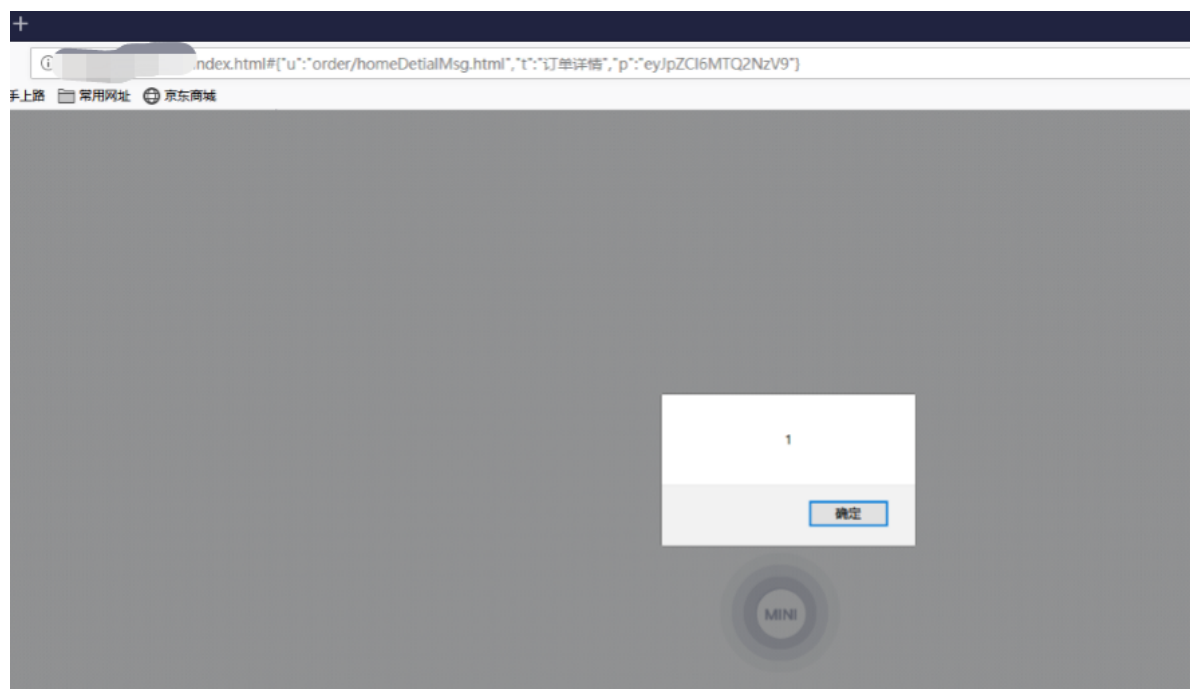
区块

1

充值数量

11111


成功弹框



成功获取后台管理员cookie:

- location : http://[REDACTED]#{%22u%22:%22order/pending.html%22,%22v%22:%22PART%22,%22_t%22:%22_top%22,%22t%22:%22%E6%96%B0%E6%98%A5%E5%8F%91%E8%BD%A6%22,%22o%22:%221%22,%22n%22:%222%22,%22m%22:%2236%22,%22p%22:%22%22}
- tolocation : http://[REDACTED]#{%22u%22:%22order/pending.html%22,%22v%22:%22PART%22,%22_t%22:%22_top%22,%22t%22:%22%E6%96%B0%E6%98%A5%E5%8F%91%E8%BD%A6%22,%22o%22:%221%22,%22n%22:%222%22,%22m%22:%2236%22,%22p%22:%22%22}
- cookie : MINI_SESSION_ID [REDACTED] 37
- opener :
- HTTP_REFERER : http://[REDACTED]
- HTTP_USER_AGENT : illa/5.0 (Linux; Android 9 N-AL00) AppleWebKit/536 (KHTML, like Gecko) Chrome/76.0.3809.132 Mobile Safari/537.36
- REMOTE_ADDR : 36.44.218
- IP-ADDR : [REDACTED]

用管理员cookie登录后台



| 订单号 | 创建时间 | 订单金额(元) | 订单金额 | 用户姓名 | 手机号码 | 商家名称 | 订单状态 |
|------------|---------------------|---------|--------------------|------|------------|------------|------|
| 2019832442 | 2019-05-02 10:58:13 | 3820 | 2007.00 3820元 | 刁瑞 | [REDACTED] | [REDACTED] | 取消 |
| 2019832441 | 2019-05-02 10:57:12 | 1372 | [REDACTED] 元 | 马松静 | [REDACTED] | [REDACTED] | 取消 |
| 2019832440 | 2019-05-02 10:43:18 | 1372 | [REDACTED] 元 | 马松静 | [REDACTED] | [REDACTED] | 取消 |
| 2019832439 | 2019-05-02 10:29:13 | 1372 | [REDACTED] 元 | 马松静 | [REDACTED] | [REDACTED] | 取消 |
| 2019832438 | 2019-05-02 10:23:21 | 7958.5 | 55.00 12.5元 397.5元 | 蔡萍 | [REDACTED] | [REDACTED] | 取消 |
| 2019832437 | 2019-05-02 10:20:31 | 7958.5 | 55.00 2.5元 397.5元 | 蔡萍 | [REDACTED] | [REDACTED] | 取消 |
| 2019832436 | 2019-05-02 10:19:54 | 7958.5 | 55.00 2.5元 397.5元 | 蔡萍 | [REDACTED] | [REDACTED] | 取消 |
| 2019832435 | 2019-05-02 10:19:20 | 7958.5 | 55.00 2.5元 397.5元 | 蔡萍 | [REDACTED] | [REDACTED] | 取消 |
| 2019832434 | 2019-05-02 10:12:11 | 1372 | 70.00 1372元 | 马松静 | [REDACTED] | [REDACTED] | 取消 |
| 2019832433 | 2019-05-02 10:11:21 | 2783.2 | 142.00 2783.2元 | 张永强 | [REDACTED] | [REDACTED] | 取消 |

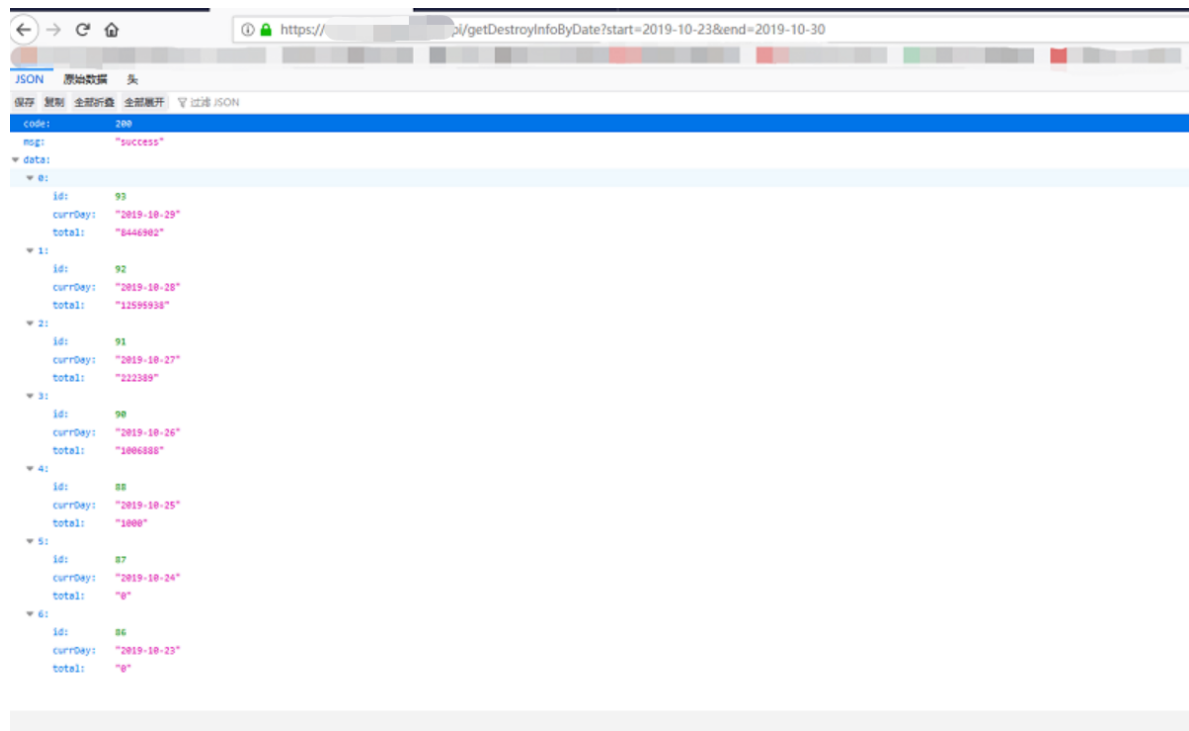
SQL注入测试

数据库 (SQL) 对于开发者和网络安全人员来说, 应该是非常熟悉了。OWASP TOP 10多次把数据库相关的SQL注入攻击列在榜首, 也足以见得危害。SQL注入漏洞的产生主要是因为Web应用程序对用户输入的数据没有进行合法性判断, 攻击者可以控制前端传入到后端的参数, 利用参数代入数据库查询, 通过构造不同的SQL语句实现对数据库的任意操作。鉴于其危害性巨大, 零时安全实验室在对交易所测

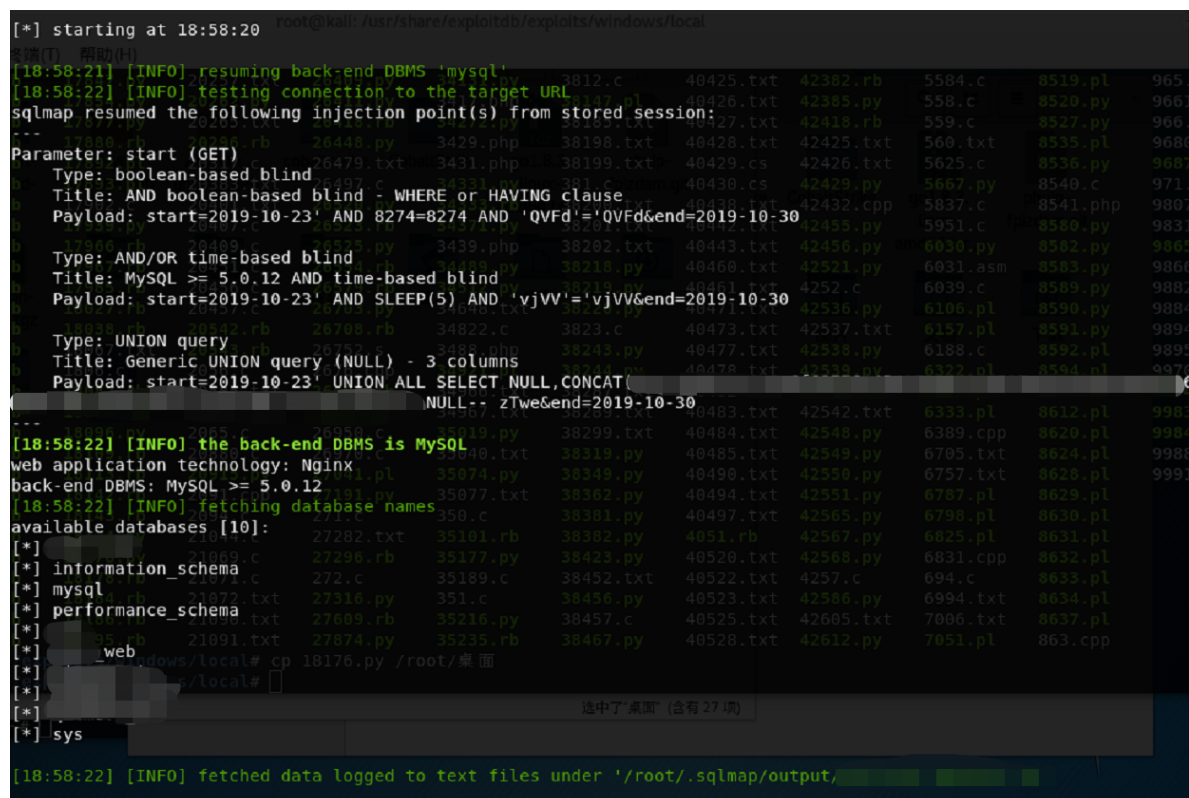
试时，对SQL注入类漏洞会进行全面细微的排查。

零时安全实验室被授权测试某交易所时，在一个提交GET请求的数据包中，发现疑似的注入点。进行手工测试后，验证存在延时注入，后使用工具进一步测试，得到了数据库表，用户名，密码等数据库存储的相关信息。

下图为手工注入测试过程：



工具测试过程：



代码注入测试

代码注入漏洞与SQL注入漏洞同隶属于注入类型，其危害也相当巨大，是经常被攻击者能用来拿下网站权限的“利器”。这种漏洞的产生大多是因为部分开发人员的编程习惯并不安全，例如经常使用一些类似于eval(), system()的危险函数等。下面用一个实例说明一下此类漏洞的危害。

零时安全实验室在对某交易所进行测试时，发现交易所的某相关域名下报错信息提示中间件及版本为Thinkphp 5.0.11，随即利用其已公开的一个代码执行漏洞写入shell，连接成功后拿下网站权限。

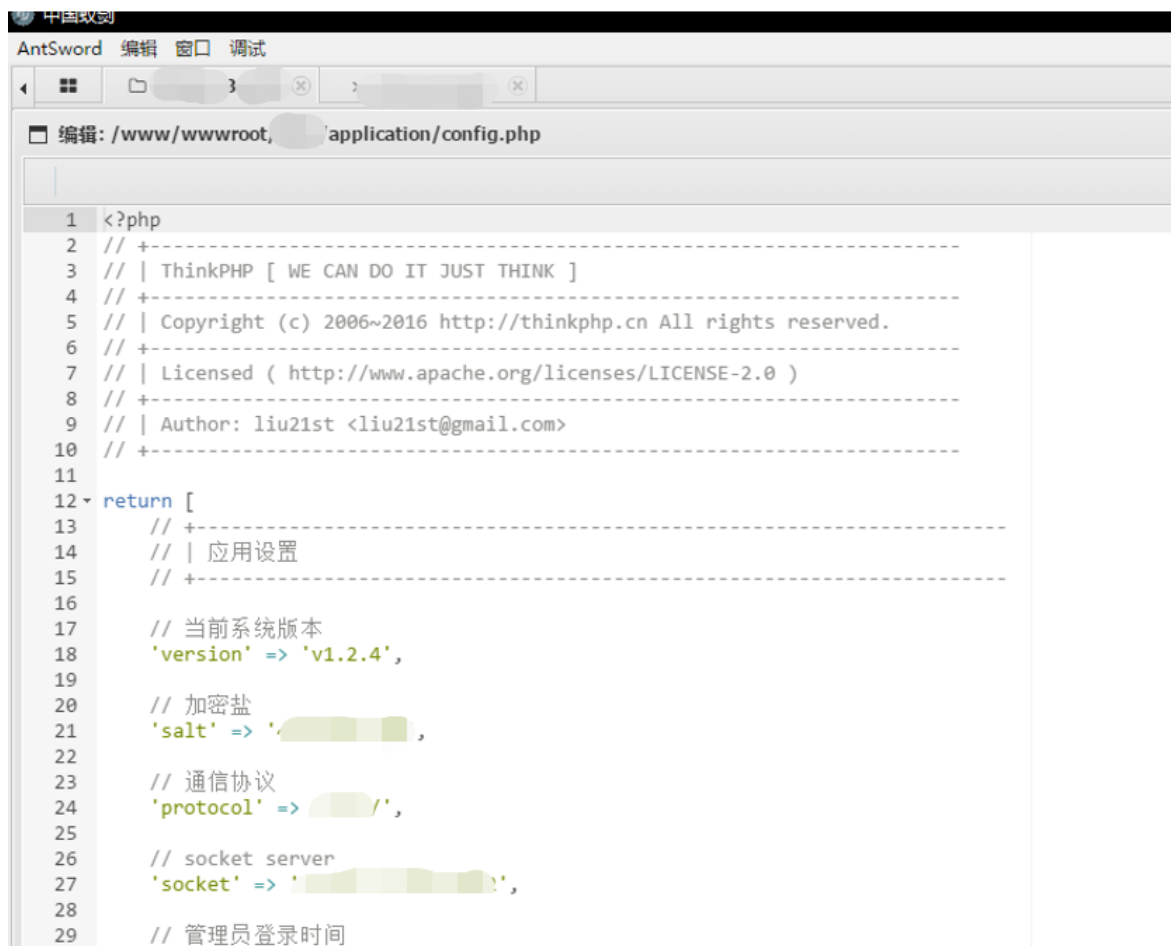
访问网站不存在的目录，管理员开启了debug信息：



根据ThinkPHP相关版本漏洞，验证漏洞是否存在，可以看到禁用方法：

| PHP Version | 5.6.30 | |
|-------------------------------|---|---|
| Directive | Local Value | Master Value |
| allow_url_fopen | On | On |
| allow_url_include | Off | Off |
| always_populate_raw_post_data | 0 | 0 |
| arg_separator.input | & | & |
| arg_separator.output | & | & |
| asp_tags | Off | Off |
| auto_append_file | no value | no value |
| auto_globals_jit | On | On |
| auto_prepend_file | no value | no value |
| browscap | no value | no value |
| default_charset | UTF-8 | UTF-8 |
| default_mimetype | text/html | text/html |
| disable_classes | no value | no value |
| disable_functions | passthru,exec,system,chroot,chr,chrgrp,chmod,shell_exec,proc_open,proc_get_status,popen,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru | passthru,exec,system,chroot,chr,chrgrp,chmod,shell_exec,proc_open,proc_get_status,popen,ini_alter,ini_restore,dl,openlog,syslog,readlink,symlink,popepassthru |
| display_errors | On | On |
| display_startup_errors | Off | Off |
| doc_root | no value | no value |
| docref_ext | no value | no value |
| docref_root | no value | no value |
| enable_dl | Off | Off |
| enable_post_data_reading | On | On |
| error_append_string | no value | no value |
| error_log | no value | no value |
| error_prepend_string | no value | no value |
| error_reporting | 32767 | 32759 |
| exit_on_timeout | Off | Off |

写入木马后，连接成功，拿到网站管理员权限：



```
1 <?php
2 // +-----+
3 // | ThinkPHP [ WE CAN DO IT JUST THINK ]
4 // +-----+
5 // | Copyright (c) 2006~2016 http://thinkphp.cn All rights reserved.
6 // +-----+
7 // | Licensed ( http://www.apache.org/licenses/LICENSE-2.0 )
8 // +-----+
9 // | Author: liu21st <liu21st@gmail.com>
10 // +-----+
11
12 return [
13     // +-----+
14     // | 应用设置
15     // +-----+
16
17     // 当前系统版本
18     'version' => 'v1.2.4',
19
20     // 加密盐
21     'salt' => '1234567890',
22
23     // 通信协议
24     'protocol' => 'http',
25
26     // socket server
27     'socket' => 'tcp://127.0.0.1:8080',
28
29     // 管理员登录时间
```

人和机器总是要产生交互

交互就意味着危险

有输入输出的地方就是危险的多发地

注：以上所有测试均已经过相关交易所授权，请勿自行非法测试。