

某某应用安全测试报告结果（样例）

安全测试结果

NO.01 开发信息

测试情况:

信息	描述	备注
开发语言	Java	
开发框架及版本	Apache Shiro 1.2.4	
数据库类型及版本	Mysql 5.7	
连接数据库框架或函数	MyBatis	
域名	www.dingtalk.com	
IP	SLB:100.100.100.100	若没有 EIP 则只填 SLB 的 IP。

NO.02 数据库信息

测试情况:

数据库字段名	字段描述	备注
id	自增 id	
username	用户昵称	
address	用户地址	

注：填写表格(数据库表、字段结构信息);如表和字段较多，仅填写与用户信息相关的表和字段。

要求：禁止存储用户的敏感信息(包括但不限于身份证号码、银行卡号等)。如有特殊需求请说明使用场景。

NO.03 域名/IP 对外开放端口信息

测试情况:

域名/IP	端口开放信息	备注
EIP:100.100.100.100	open:22	
SLB:200.200.200.200	open: 80 443	
域名:www.dingtalk.com	open: 80 443	

注: 填写表格并附扫描情况截图

要求: EIP 仅开放远程连接端口(如 22、3389 端口); SLB 仅开放 web 服务端口(如 80、443 端口); 域名仅开放 web 服务端口(如 80、443 端口)。如有特殊端口开放请说明场景。

截图信息:

EIP:

```
$ $ → ~ sudo nmap -sS -Pn -p1-65535 100.100.100.100

Starting Nmap 7.01 ( https://nmap.org ) at 2019-01-04 06:56 UTC
Nmap scan report for 100.100.100.100
Host is up (0.000010s latency).
rDNS record for 100.100.100.100
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
```

SLB:

```
$ $ → ~ sudo nmap -sS -Pn -p1-65535 200.200.200.200

Starting Nmap 7.01 ( https://nmap.org ) at 2019-01-04 06:56 UTC
Nmap scan report for 200.200.200.200
Host is up (0.000010s latency).
rDNS record for 200.200.200.200
Not shown: 65531 closed ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
```

域名:

```
$ $ → ~ sudo nmap -sS -Pn -p1-65535 www.dingtalk.com
```

```
Starting Nmap 7.01 ( https://nmap.org ) at 2019-01-04 06:56 UTC
```

```
Nmap scan report for 200.200.200.200
```

```
Host is up (0.000010s latency).
```

```
rDNS record for 100.100.100.100: www.dingtalk.com
```

```
Not shown: 65531 closed ports
```

PORT	STATE	SERVICE
------	-------	---------

80/tcp	open	http
--------	------	------

443/tcp	open	https
---------	------	-------

NO.04 敏感信息

测试情况:

场景信息	情况描述	备注
是否存在接口返回或展示用户较敏感信息	不存在	
服务端业务日志是否打印或存储敏感信息		

注：填写表格

要求：1. 返回或展示较敏感信息(例如手机号、邮箱)需进行打码处理 2. 服务端业务日志不允许打印或存储敏感信息

NO.05 越权漏洞

使用工具：抓包工具

测试结果

测试情况:

相关接口	功能描述	备注
dingtalk.com/my?orderid=	查看订单信息的接口	从 session 中获取用户信息，并判断订单是否属于该用户，不存在越权。
(POST)dingtalk.com/my (body)UpdateOrderid=&name=	修改订单信息的接口	从 session 中获取用户信息，并判断订单是否属于该用户，不存在越权。

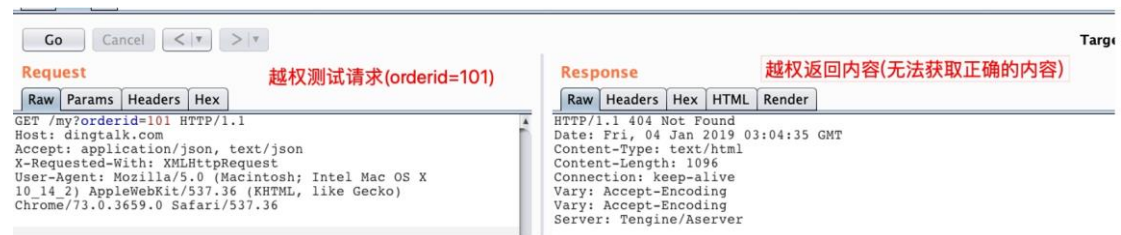
注：填写表格，并附测试情况截图。
要求：所有接口都需进行测试，并且都不存在越权漏洞，但截图可仅提供部分接口的测试截图。
截图要求：至少需要两张截图，一张 A 用户正常的请求，一张 A 用户修改了 URL 参数的请求。截图需能看出请求的 URL 及返回的内容。

截图信息:

A 用户查看 orderid=100 的信息。orderid=100 是属于 A 用户的订单信息



A 用户查看 orderid=101 的信息。Orderid=101 是属于 B 用户，不属于 A 用户的订单信息，A 无权查看。



NO.06 跨站脚本攻击漏洞

测试情况:

存在输入输出内容的页面	功能描述	备注
dingtalk.com/edit	编辑公告的页面	对特殊符号进行转义，不存在跨站脚本攻击
dingtalk.com/show	查看公告的页面	对特殊符号进行转义，不存在跨站脚本攻击

注：所有存在输入内容的页面都需进行测试，并且都不存在跨站漏洞，但截图可仅提供部分接口的测试截图。

要求：至少需要一张截图，查看内容被插入了 HTML 代码的页面。截图需能看出内容中只以文字形式展示">",并未执行恶意操作。

截图信息:

A 用户在 dingtalk.com/edit 页面编辑公告并发布，公告内容为

">



A 用户在 dintalk.com/show 查看公告信息，公告信息将代码以文字形式全部展示，未成功执行

11:15



[返回](#)

公告

[更多](#)

测试

测试 2019.1.4

[全部已读](#)

"

NO.07 SQL 注入漏洞

测试情况:

SQL 操作的代码实现方式	备注
使用 Mybatis	未使用拼接 SQL 语句的方式

注：将执行 SQL 语句相关的关键代码进行截图。使得能够看出来未使用拼接 SQL 语句的方式。

截图信息:

```
<select id="getUser" parameterType="int"
      responseType="me.gacl.domain.User">
    select * from users where id=#{id}
</select>
</mapper>
```

使用Mybatis，并且使用#{var}的方式

NO.08 跨站请求伪造漏洞

使用工具：抓包工具

测试情况：

相关接口	功能描述	备注
dingtalk.com/omp/api/profile	修改个人信息的接口	验证 referer, 不存在跨站脚本请求伪造
dingtalk.com/omp/api/delprofile	删除个人信息的接口	验证 referer, 不存在跨站脚本请求伪造

注：填写表格，并附测试情况截图。

所有接口都需进行测试，并且都不存跨站请求伪造漏洞，但截图可仅提供部分接口的测试截图。

截图要求：至少需要两张截图，一张正常的 Referer 或 token 的请求，一张修改了 Referer 或 token 的请求。截图需能看出请求的内容及返回的内容

截图信息：

A 用户在 dingtalk.com/omp/api/profile 页面正常修改个人信息，此时 Referer 为 https://dingtalk.com，修改成功。



A 用户在 dingtalk.com/omp/api/profile 页面正常修改个人信息，并将 Referer 改为错误 Referer https://aaaaaaaadingtalk.com，修改

失败

Go Cancel < >

Target: https:/

Request

Raw Params Headers Hex

POST /omp/api/profile HTTP/1.1
Host: dingtalk.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3660.0 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: https://aaaaadingtalk.com/contacts.htm

修改为错误Referer

Response

Raw Headers Hex

当Referer未通过校验, 修改失败

HTTP/1.1 400 Bad Request
Server: DingTalk/1.0.0
Date: Fri, 04 Jan 2019 03:45:52 GMT
Content-Type: application/json; charset=UTF-8
Content-Length: 67
Connection: close
Content-Language: zh-CN

NO.09 文件上传漏洞

测试情况:

检查项	检查结果
是否存在文件上传功能	是
文件上传至（）	服务器本地（如上传至 oss，则跳过下列所有检查项）
上传文件类型限制	<p>举例如下，根据应用实际情况详细说明。</p> <p>例 1：无，可上传任意文件</p> <p>例 2：有，只能上传 jpg、png、doc、xls 格式文件，使用白名单检验文件后缀，在前端页面进行限制</p> <p>例 3：有，不能上传 html、jsp、asp、cer 格式文件，使用黑名单检验文件后缀，在服务端进行限制。</p> <p>例 4：有，只能上传 jpg、png、doc、xls 文件，使用白名单校验文件后缀&&文件头，使用黑名单规则过滤<、script、eval 关键词校验文件内容，在服务端进行限制</p>
上传文件重命名	<p>例 1：无相关措施</p> <p>例 2：有，所有文件上传至服务器，统一命名为 xxxxxx.jpg</p> <p>例 3：有，文件重命名，但文件后缀保持原样。</p>
上传文件查看	<p>例 1：直接下载，不能在应用上预览</p> <p>例 2：可以在应用里查看，但所有文件按照图片渲染</p> <p>例 3：无特殊处理，所有文件保持原样在应用里查看</p>
其他处理措施补充说明	

注：填写表格，并附测试情况截图。

截图要求： 截图能支撑检查结果。

1) 如文件上传至 oss，访问已上传文件的 url：oss-

cnhangzhou.aliyuncs.com/1291723234/3e34j4u4dkdo.jpg，将地址栏信息一并截图

2) 其余对代码部分截图，要能看出应用对上传文件做了相应限制

截图信息:

