



逻辑漏洞和src小技巧

分享人彦语



逻辑漏洞

- 注册覆盖
- 系统会将已经注册的用户手机号识别出来，并返回ture
- 如果检测该手机号没注册的话，会返回false。那我们就修改返回包，达到覆盖注册的目的
- 某功能点对于状态的判断仅限制于前端，那就有可能通过修改返回包来证明逻辑缺陷。
- 例如修改状态码 200 等
- 注册遍历
- 短信验证码与用户绑定（在注册时切换手机号） 手机号、邮箱换绑
- 密码找回/修改 存在任意密码找回，（存一个修改成功的返回包，再输入错的密码进行替换）
- 短信轰炸
- 验证码爆破等等
- 支付逻辑漏洞
- 水平越权、垂直越权 （通过修改代表用户身份的某变量来达到越权效果。）

并发漏洞

- 应用场景 一切对于次数有限制的功能点
- 思考互动
- 大家认为在什么功能点中，系统会对用户有次数限制？

- 比如
 - 点赞、阅读量、热度、评论
 - 验证码短信轰炸
 - 有领取限制的代金券、优惠券 有使用次数限制的代金券
 - VIP限制的功能点 领取次数 各种每天签到才能获得的东西 金币、时长、数据卡等
-
- **高危 和金钱挂钩的功能点 （限购商品）**

功能点示例



MITM

Fuzzer

Codec

数据对比

解码

编码

Web

WS

Codec

DNSLog

History

Web Fuzzer

MITM 交互式劫持

增加管理员1包

导出信息1包

修改管理员密码包

在一般添加管理员包不可时, ...

查看admin密码包

WF-[10]

WF-[7]

强制 HTTPS

国密TLS

真实Host

设置代理

禁用系统代理

请求包配置

并发配置

并发线程

随机延迟

重试配置

重定向配置

DNS配置

匹配器

数据提取器

设置变量

重置

重置

重置

重置

添加/测试

添加/测试

添加

发送请求

强制 HTTPS

历史

Request

数据包扫描

热加载

构造请求

Responses

成功[1000]

失败[0]

提取响应数据

导出数据

请求	Method	状态	响应大小	延迟(ms)	Payloads	操作
1	GET	200	124	17		
2	GET	200	124	25		
3	GET	200	124	28		
4	GET	200	124	33		
5	GET	200	124	32		
6	GET	200	124	28		
7	GET	200	124	28		
8	GET	200	124	30		
9	GET	200	125	32		
10	GET	200	125	32		
11	GET	200	125	35		
12	GET	200	125	33		
13	GET	200	125	32		
14	GET	200	125	33		
15	GET	200	125	34		
16	GET	200	125	35		
17	GET	200	125	33		
18	GET	200	125	34		
19	GET	200	125	31		

强制 HTTPS



国密TLS



真实Host

请输入...

设置代理

请输入...

禁用系统代理



请求包配置

重置

并发配置

重置

重复发包

1000

一般用来测试条件竞争或者大并发的情况

并发线程

20

随机延迟

Min

0 s

Max

0 s

重试配置

重置

重定向配置

重置

DNS配置

重置

匹配器

0

重置

添加/测试

数据提取器

0

重置

添加/测试

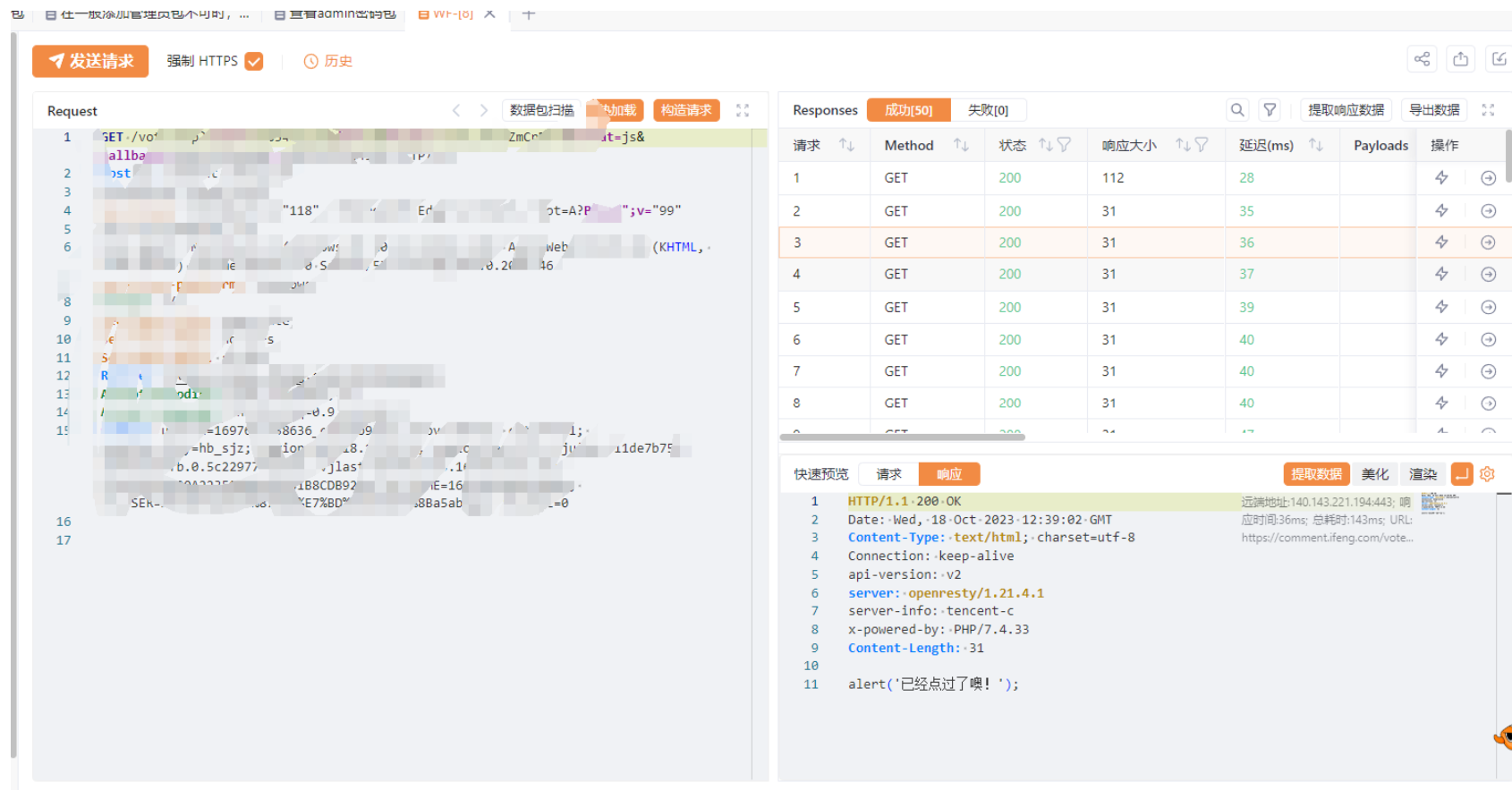
设置变量

重置

预览

添加

并发失败示例

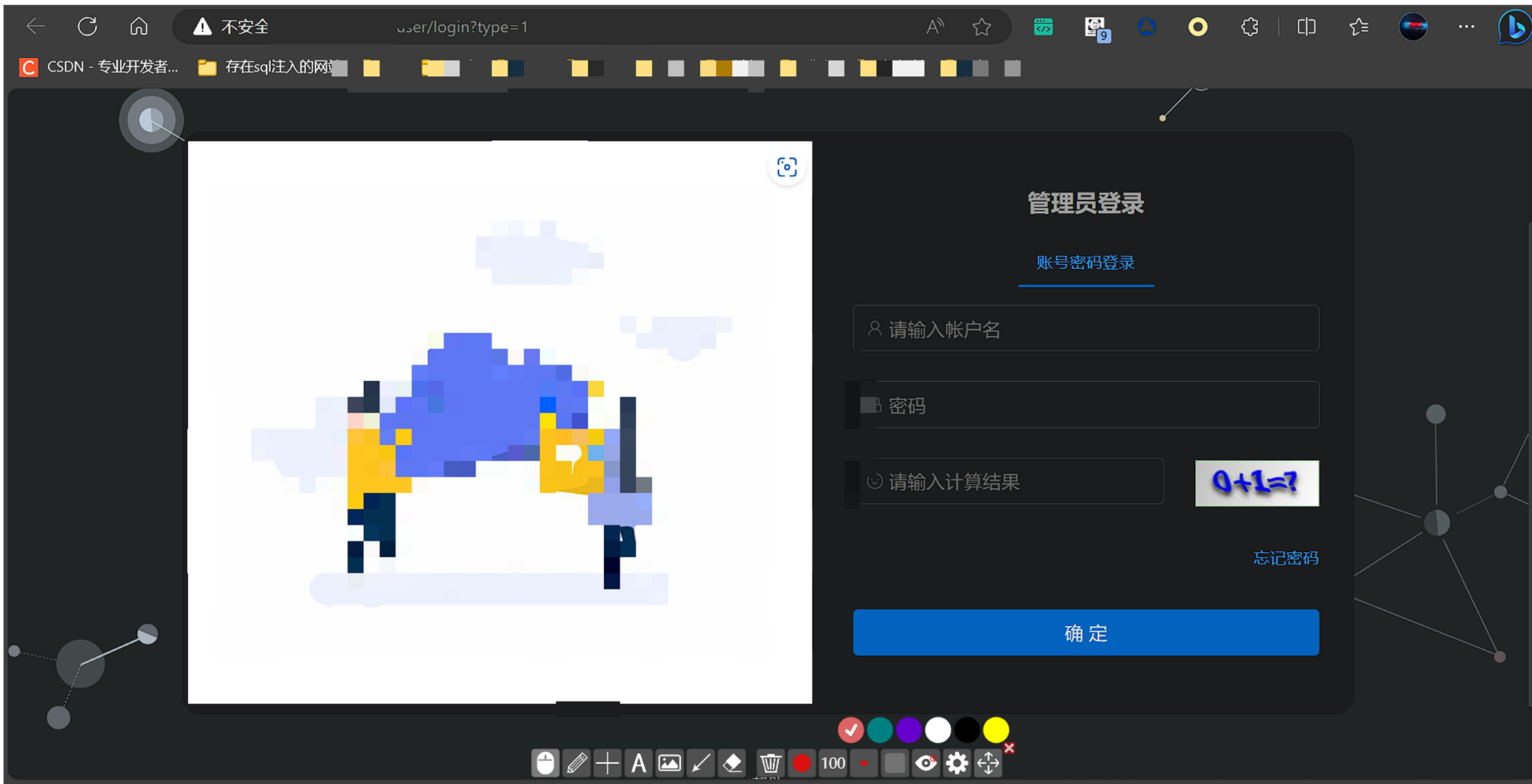


- 问题： 它和burp的重放器的区别在哪？

- 答案： 线程，重发器和爆破（inturder）是单线程的，而并发是多线程的

src小技巧

- 多观察数据包
- 将有用的数据包一个个重放，看他的返回包



找回密码

1 确认账号

2 邮箱验证

3 发送成功

请输入用户账号

请输入验证码

7*9≈?




[返回登录](#)

[下一步](#)

可以测的点

- 右键看看源码
- f12刷新界面 网络 看看数据包 js接口 (jsfinder、findsomething)
- 敏感目录扫描工具
- sql注入 (框 + url)
- 弱口令
- xss (登录框里 url里)
- 一系列逻辑
- 不同用户的登录口可能是不一样的, 管理员、普通学生、老师、职工等等身份


万能密码（碰运气）



[报名缴费](#) [学员登录](#) [培训部主页](#)


姓名

密码




*默认为身份证后六位


验证码



登录

QQ咨询

 李老師

 李老師

实例

× 全屏



Response

```
HTTP/1.1 200
Content-Type: application/json
Connection: keep-alive
Server: Apache/2.4.18 (Ubuntu)
Vary: Origin
Vary: Access-Control-Request-Headers
Content-Length: 338964
```

```
{
  "success": true,
  "message": "操作成功!",
  "code": 0,
  "result": [
    {
      "id": "1e08c50f428446f55072",
      "parent": "1e08c50f428446f55072",
      "departName": "部门名称",
      "departNameEn": "Department Name",
      "departNameAbbr": "部门名称缩写",
      "departOrder": 1,
      "departCode": "1e08c50f428446f55072",
      "description": "部门描述",
      "departType": "1",
      "orgCategory": "1",
      "orgType": "2",
      "majorType": "1",
      "orgCode": "1e08c50f428446f55072",
      "mobile": null,
      "fax": null,
      "address": "部门地址",
      "memo": null,
      "type": "1",
      "status": null,
      "delFlag": "0",
      "createBy": "1e08c50f428446f55072",
      "createTime": "2017-07-11 11:11:11",
      "updateBy": "1e08c50f428446f55072",
      "updateTime": "2017-07-11 11:11:11",
      "disabled": "1",
      "pid": null,
      "hasChild": "1"
    }
  ],
  {
    "id": "1e08c50f428446f55072"
  }
}
```

渗透测试 安全工具 插件 反连 数据库

MITM Web WS Codec 解码 编码 数据对比

导入协作资源 Codec Payload Yak Runner

首页 History MITM 交互式劫持 Web Fuzzer

劫持 HTTP Request

规则配置 过滤器 证书下载 免配置启动

全部 已启用 热加载 关键字 手动劫持 自动放行 被动日志 清空 已屏蔽条件 5

- 插件组 1 添加至组
- ☐ 基础 XSS 检测
 - ☐ 启发式SQL注入检测
 - ☐ Shiro 指纹识别 + 弱密码检测
 - ☐ FastJSON 漏洞检测 via DNSLog
 - ☐ GraphQL API Fingerprint Checking
 - ☐ Spring Actuator 敏感信息泄露
 - ☐ ThinkPHP RCE 被动扫描
 - ☐ ActiveMQ 默认密码检查
 - ☐ HTTP Web 目录爆破: 敏感中间件
 - ☐ /crossdomain.xml allow-access-from...
 - ☐ Elasticsearch 未授权访问
 - ☐ Git Leak Detection [ScanPort]

序号	方法	状态码	URL	Title	Tag	操作
122	GET	200	http://inc.net/.../sys/user/.../19844&username=admin	-		
121	GET	200	http://inc.net/.../sys/user/check/.../16961754&username=admin	-		
120	GET	200	http://inc.net/.../sys/user/check/.../1754&username=admin	-		
119	GET	200	http://inc.net/.../sys/user/check/.../1754&username=admin	-		
118	GET	200	http://inc.net/.../sys/user/check/.../198475&username=admin	-		
117	GET	200	http://inc.net/.../sys/user/check/.../596984&username=admin	-		
115	GET	200	http://inc.net/.../sys/user/check/.../696984226	-		
113	GET	200	http://inc.net/.../sys/user/check/.../1696984226	-		
112	GET	200	http://inc.net/.../sys/user/check/.../42267447×tamp=1696984226	-		

Request

```
GET /.../sys/user/check/.../1754&username=admin HTTP/1.1
Host: inc.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36
Referer: http://inc.net/.../sys/user/check/.../1754&username=admin
Accept-Encoding: deflate, gzip
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,es;q=0.6
Accept: application/json, text/javascript, */*; q=0.01
tenant id: ...
```

Response

```
{
  "success": true,
  "message": "操作成功!",
  "code": 0,
  "result": true,
  "timestamp": 1696984755837
}
```


小技巧二

- 不知道数据包中的哪个参数是决定返回包中数据的，那就控制变量，一个一个删除，看看哪些删了是不影响的哪些删除之后会报错。
- 找到了此参数后，就可以尝试去更改，尝试去挖掘

通杀洞经验

- 要想达到通杀洞，大概率我们要首先进入一个系统，进入了之后可以测试的功能点就多起来了，比如说一个截取一个系统中的某功能点的数据包，然后替换host，ip域名端口之类的。
- 去在其他使用一个系统的站点，在未登录的情况下导出信息、增加账户、修改密码、查看密码、删除用户
- 文件上传、xss、越权等等

2023-10-15	任店 级中学	中危	等待修复	
2023-10-15	叶 庄 级中学打包	中危	等待修复	
2023-10-15	级中学	中危	等待修复	
2023-10-15	级中学	中危	等待修复	
2023-10-15	级中学	中危	等待修复	
2023-10-15	级中学	中危	等待修复	
2023-10-15	中学	中危	等待修复	
2023-10-15	中学	中危	等待修复	
2023-10-14	通 专科	中危	等待修复	
2023-10-11	通 专科	中危	未通过	修改
2023-08-24	术学	低危	未通过	修改
2023-08-18		低危	等待修复	
2023-08-07	广东 贺	中危	已修复	



感谢聆听

欢迎加入我们, qq群: 343380539



< 8 >