

企业安全 攻与防

平安SRC线上沙龙系列主题活动 第六期

🕒 时间：2024.5.31 14:00-17:00

主办方



合作伙伴



国内SRC漏洞挖掘经验分享

地图大师 (returnwrong)

Day1安全团队核心成员



ID: 地图大师
reutrnrwong

Day1安全团队核心成员热，爱漏洞挖掘，热爱技术分享，“希望能做你挖洞之路的导航地图”：

2023年智联SRC第5

2023年银联SRC第10

2023年知识星球SRC第4

2022年猎聘SRC第8

2022年BOSS直聘SRC第7

.....

安全研究：地图API漏洞、契约锁API接管漏洞



目录

1. 挖掘SRC的法律注意事项和公告内容解读
2. 国内SRC挖掘经验分享
3. 漏洞案例及插件讲解

每个人擅长的游戏不同，每个人擅长挖掘的漏洞领域也不同

有的人擅长挖掘银行、金融，有的人擅长挖掘互联网产品、文档类漏洞

找到自己擅长的领域才能坚持挖下去

建议新手师傅可以先每个SRC交几个漏洞，然后看那些符合自己的心理预期再去挖掘，不然遇到那种半个月不审核漏洞，或者内部已知的SRC心态直接崩。







挖掘SRC的法律注意事项和公告内容解读



《中华人民共和国刑法》中和挖洞有关的条例：

1、二百八十六条：违反国家规定，对计算机信息系统功能进行删除、修改、增加、干扰，造成计算机信息系统不能正常运行，后果严重的，处五年以下有期徒刑或者拘役；后果特别严重的，处五年以上有期徒刑。

解读：

- 1、不要删除正常业务数据
- 2、SQL注入拿到库名就点到为止
- 3、遇到不认识的接口（delete、drop）等，不要随便遍历
- 4、DoS类漏洞证明可以延时即可
- 5、Geshell类漏洞上传无害化脚本证明即可

《网络安全法》中和挖洞有关的条例：

- 1、第四十四条 任何个人和组织不得窃取或者以其他非法方式获取个人信息，不得非法出售或者非法向他人提供个人信息。
- 2、第四十八条 任何个人和组织发送的电子信息、提供的应用软件，不得设置恶意程序，不得含有法律、行政法规禁止发布或者传输的信息。

解读：

- 1、越权类漏洞用自己的测试账号进行测试，能读取的真实数据不超过5组，严禁批量读取
- 2、不要给上传的漏洞附件里塞木马（我见过有师傅想这么干过）

1

原本就是个低危漏洞非得按高危漏洞提，还和审核掰扯。

2

漏洞等级确认后不认可，在网上发小作文，引导舆论攻击该SRC或厂商。

3

违反SRC用户公告或众测条例，嘎嘎开漏扫，什么扫描器PoC多上啥扫描器。

4

交假洞，伪造客观事实，企图蒙混过关。

01

加各个src审核微信和公众号及时了解最新的活动，多种翻倍奖励

02

遇到有争议的漏洞及时和审核微信或qq沟通，把自己的想法说出来

03

子域名收集灯塔、subfinder、httpx等工具可以使用，因为这些不是poc扫描器不会对网站产生很大压力

04

漏洞通过后及时体现，避免被收回，有好多师傅不及时体现，会导致漏洞复核时被收回。被收回后及时和审核沟通

[TPSA19-22]SRC行业安全测试规范

公告编号: TPSA19-22 公告来源: TSRC 发布日期: 2019-11-08

 分享

参与此标准制定的组织:

腾讯SRC、蚂蚁金服SRC、ASRC、阿里云先知、百度SRC、本地生活SRC、菜鸟SRC、滴滴SRC、京东SRC、LYSRC、蘑菇街SRC、陌陌SRC、360SRC、苏宁SRC、同舟共测-企业安全响应联盟、唯品会SRC、微博SRC、VIPKID SRC、网易SRC、WiFi万能钥匙SRC、完美世界SRC、58SRC、小米SRC (排名不分先后)

感谢以下白帽子对此规范提供的建议和认可:

hackbar、SToNe、无心、mmmark、ayound、算命先生、泳少、离兮、lakes、Adam、羽_、小笼包 (随机排名, 不分先后)

一、测试规范:

1. 注入漏洞, 只要证明可以读取数据就行, 严禁读取表内数据。对于UPDATE、DELETE、INSERT等注入类型, 不允许使用自动化工具进行测试。
2. 越权漏洞, 越权读取的时候, 能读取到的真实数据不超过5组, 严禁进行批量读取。
3. 帐号可注册的情况下, 只允许用自己的2个帐号验证漏洞效果, 不要涉及线上正常用户的帐号, 越权增删改, 请使用自己测试帐号进行。
帐号不可注册的情况下, 如果获取到该系统的账密并验证成功, 如需进一步安全测试, 请咨询管理员得到同意后进行测试。
4. 存储xss漏洞, 正确的方法是插入不影响他人的测试payload, 严禁弹窗, 推荐使用console.log, 再通过自己的另一个帐号进行验证, 提供截图证明。对于盲打类xss, 仅允许外带domain信息。所有xss测试, 测试之后需删除插入数据, 如不能删除, 请在漏洞报告中备注插入点。

5. 如果可以shell或者命令执行的，推荐上传一个文本证明，如纯文本的1.php、1.jsp等证明问题存在即可，禁止下载和读取服务器上任何源代码文件和敏感文件，不要执行删除、写入命令，如果是上传的webshell，请写明shell文件地址和连接口令。
6. 在测试未限制发送短信或邮件次数等扫号类漏洞，测试成功的数量不超过50个。如果用户可以感知，例如会给用户发送登陆提醒短信，则不允许对他人真实手机号进行测试。
7. 如需要进行具有自动传播和扩散能力漏洞的测试（如社交蠕虫的测试），只允许使用和其他账号隔离的小号进行测试。不要使用有社交关系的账号，防止蠕虫扩散。
8. 禁止对网站后台和部分私密项目使用扫描器。
9. 除特别获准的情况下，严禁与漏洞无关的社工，严禁进行内网渗透。
10. 禁止进行可能引起业务异常运行的测试，例如：IIS的拒绝服务等可导致拒绝服务的漏洞测试以及DDOS攻击。
11. 请不要对未授权厂商、未分配给自己的项目、超出测试范围的列表进行漏洞挖掘，可与管理员联系确认是否属于资产范围后进行挖掘，否则未授权的法律风险将由漏洞挖掘者自己承担。
12. 禁止拖库、随意大量增删改他人信息，禁止可对服务稳定性造成影响的扫描、使用将漏洞进行黑灰产行为等恶意行为。
13. 敏感信息的泄漏会对用户、厂商及上报者都产生较大风险，禁止保存和传播和业务相关的敏感数据，包括但不限于业务服务器以及Github等平台泄露的源代码、运营数据、用户资料等，若存在不知情的下载行为，需及时说明和删除。
14. 尊重《中华人民共和国网络安全法》的相关规定。禁止一切以漏洞测试为借口，利用安全漏洞进行破坏、损害用户利益的行为，包括但不限于威胁、恐吓SRC要公开漏洞或数据，请不要在任何情况下泄露漏洞测试过程中所获知的任何信息，漏洞信息对第三方披露请先联系SRC获得授权。企业将对违法违规者保留采取进一步法律行动的权利。



国内SRC漏洞挖掘经验分享

举例：某src接收漏洞类型

【严重】	【中危】	【低危】
额外奖励（暂定为1万-2万）		
1、可获得关键服务器权限；	1、需要交互才能获取用户身份信息的漏洞：包括但不限于经评估可获得管理员权限的存储型XSS；	1、无法获得数据的SQL注入漏洞；
2、可获得大量持卡人账户敏感数据，经评估影响资金安全；	2、越权操作：包括但不限于普通用户权限批量越权访问、查看其他用户信息、管理后台访问；	2、Jsonp Hijacking、CSRF等漏洞；
3、可获得大量商户清算数据，造成公司、用户损失；	3、应用缺陷导致的远程拒绝服务漏洞，不包括DDOS或CC方式；	3、一般信息泄露漏洞：包括但不限于路径泄露、SVN文件泄露、LOG文件泄露、Phpinfo、页面文件遍历；
4、可破解核心加密算法，影响敏感数据的存储、传输等；	4、应用系统管理员弱口令：可以获得该应用系统的管理员权限；	4、一般越权操作：包括但不限于普通用户权限越权修改、删除其他用户信息等；
5、经综合评估认定的其他严重安全漏洞。	5、直接通过客户端可获取操作系统管理员（或手机root）的权限；	5、一般逻辑设计缺陷：包括但不限于无限制短信/邮件发送、图形验证码绕过、非关键环节短信验证码绕过等；
	6、可远程窃取客户端数据等；	6、无法利用或者难以利用的漏洞：包括但不限于反射型XSS；
	7、经综合评估认定的其他中危安全漏洞。	7、URL跳转：包括但不限于未验证的重定向和转发；
		8、客户端本地漏洞：包括但不限于本地拒绝服务漏洞、命令截断、应用程序目录下的dll劫持；
		9、因业务需要可能导致的撞库、爆破、遍历接口（获得的数据不涉及敏感信息）；
【高危】		
1、直接获取操作系统权限（服务器权限、客户端权限）：包括但不限于远程任意命令执行、上传Webshell、缓冲区溢出、服务器解析漏洞等可获得服务器权限；		
2、逻辑设计缺陷：包括但不限于任意账号登录、任意账号密码修改、任意金额支付、关键环节短信邮件验证码绕过、任意账号支付等；		
3、敏感信息泄露：包括但不限于SQL注入、任意文件包含、任意文件读取、源代码泄露、账户及支付交易等敏感信息批量泄露、客户端加密算法可被破解；		
4、经综合评估认定的其他高危安全漏洞。		

高危漏洞以任意用户、信息泄露为主。

任意用户这块，我首先是用灯塔、subfinder、fofa、hunter等子域名导出去重，然后httpx识别响应码，然后挨个子域名看。

漏洞挖掘方法：我挖SRC的方法简单粗暴，就是收集齐了子域名，挨个域名挖。能注册的就挨个注册，把所有能想到的任意用户漏洞的点，比如登录、注册、密码重置都试一遍。




信息泄露主要是findsomething，找到隐藏在js里的接口，或者用dirsearch找到那种备份文件。

js-可通过前端绕过+弱口令-获取大量员工信息 (姓名+部门+岗位+电话)	Web漏洞	已确认	1500	25	已确认	2023-05-15 15:21
存在4位数验证码可爆破-导致任意用户登录或任意用户密码重置	Web漏洞	已确认	1200	24	已确认	2023-03-27 12:14
数据库备份文件泄露-存在数据库源码	Web漏洞	已确认	500	20	已确认	2023-03-16 20:10

中危漏洞主要以**越权**为主

漏洞挖掘方法：我挖掘越权的方法也是和上页说的一样，能注册就注册进去，注册不了就找找接口看能不能未授权进去。



越权大家都会挖，这里主要说下大家一定要对SRC的根域名收集的够齐，你的攻击面才越大。


 平台-存在越权删除机构信息-可删除全站机构信息	Web漏洞	已确认	600	12	已确认	2023-12-08 11:01
 -存在越权删除联系人信息漏洞	Web漏洞	已确认	600	12	已确认	2023-12-07 22:40
 企业版-存在垂直越权漏洞	Web漏洞	已确认	600	12	已确认	2023-06-27 16:56

低危漏洞就花样繁多了，我的技巧每挖一个SRC是把我知道的所有低危都交一遍，看看收那个，剩下的我在批量挨个挖，有很多师傅其实看不上挖掘低危漏洞，反而我对这块很有热衷。

低危这里给大家着重说几个，

- **地图API**：这个现在很多已经不收了，之前100一个，我大概刷了30多个
- **用户名枚举**：这个没有写在公告里，所以需要按我前面说的把自己会的都挖一遍，很多师傅可能看不上挖这个，但这个在银联这块可以尝试挖挖，一个100，我大概也挖了20来个吧
- jsonpxss、并发、短信轰炸
- cors json信息泄露
- 任意二维码生成、URL跳转等等

 存在任意二维码	Web漏洞	已确认	50	5	已确认	2023-06-1
生成漏洞						
 存在并发漏洞	Web漏洞	已确认	100	9	已确认	2023-06-15 11:30





漏洞案例讲解



案例 1：组合拳实现高危漏洞

PINGANSECURITYRESPONSECENTER

漏洞场景：在我们信息收集完之后，发现很多子域名都没法打开或者一闪而过，这种有时候用很简单的技巧也能实现高危漏洞

步骤1：打开子域名发现网站一闪而过，抓包看响应包，发现响应里面有一段js关闭代码

```
var userAgent = navigator.userAgent;  
if (userAgent.indexOf("Firefox") != -1 || userAgent.indexOf("Chrome") != -1) {  
    window.location.href = "about:blank";  
    window.close();  
}
```

案例1：组合拳实现高危漏洞2

PINGANSECURITYRESPONSECENTER

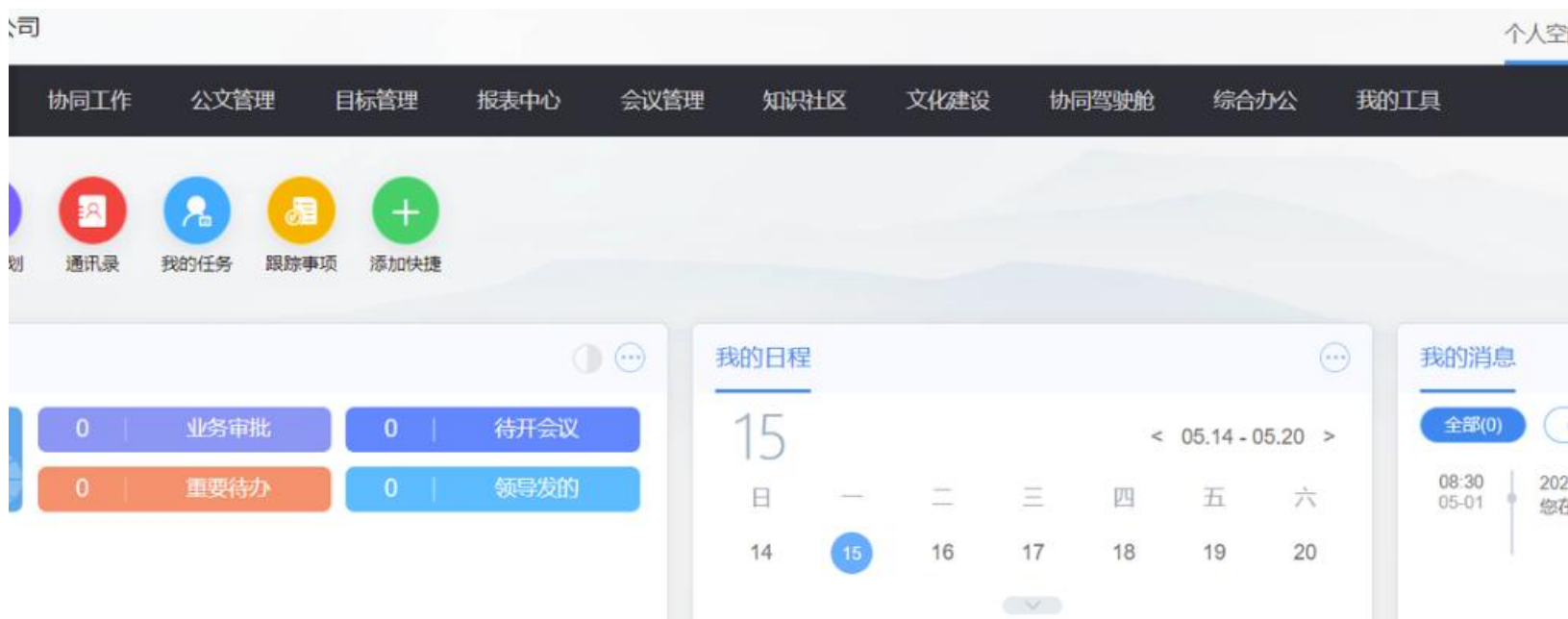
步骤2：成功来到登录页面，发现此处存在用户名枚举漏洞，会提示用户名不存在，果断枚举



案例 1：组合拳实现高危漏洞3

PINGANSECURITYRESPONSECENTER

步骤3：找到一个存在的账户，xiaomingwang，发现在这个系统里姓名是反过来的，于是调整爆破字典将常见中文名进行替换，爆破出多个账户，再用弱口令字典，成功进去其中某一个账户



案例 1：组合拳实现高危漏洞 4

步骤4：成功进入系统，通过联系人功能拿到该公司全部员工信息（姓名、职位、部门、电话等），提交后获得赏金（1500+1000京东卡），

[illegible]

案例2：多个数据包实现越权解绑

漏洞场景：“套娃”越权，在网站测试的时候，发现直接修改userid是不能实现越权的。但是回头看数据包日志的时候发现三个数据包之间似乎存在联系

步骤1：重新在页面抓取包含userid的包

OPTIO...	/wwwopen/openData/agentConfig?f=json&r=j7i9oIn2gq	✓	20
GET	/getTicket?userid=ARTCA PRO 1615	✓	20
POST	/wwwopen/openData/agentConfig?f=json&r=i7i9oIn2gq	✓	20
POST	/web?env=trtc-inspection-test-2bhec82d41e	✓	20
POST	/collect?id=bRLDot6R4Kymzz0iPO&from=https://ando...	✓	20
POST	/web?env=trtc-inspection-test-2bhec82d41e	✓	20
POST	/web?env=trtc-inspection-test-2bhec82d41e	✓	20
POST	/web?env=trtc-inspection-test-2bhec82d41e	✓	20

案例2：多个数据包实现越权解绑

步骤2：将userid改成小号的解绑id 1616，发现响应包里生成了一个类似于token的字段，将此处生成的数据填入下一个请求包

request

RawParamsHeadersHex

GET /getTicket?userid=ARTCA_PRO_1616 HTTP/1.1

response

RawHeadersHexRender

HTTP/1.1 200 OK

70d9573d-a50e-4601-930d-d/evJhbGciOiJSUzI1NiIsImV
udil6lnRydGMtaW5zcGVjdGlveHAiOiE3MDI3NDMxOTU
5NjYsInVpZC16lkFSVENBX1BSczk1OTY2fQ. Vpz4BZpZ
k1QQ4nqc6fq6FHZL_sdbG1911u7Jq7YB1kh7WxNh1MFDV7
tLqv1yb9NA5YE6WhmF1LwdpKcw42p

将这个值填写到下一个数据包

案例2：多个数据包实现越权解绑

步骤3：当我们将上个包的token填写到如下数据包的ticket字段后，我们有获取到了一个refresh_toekn，同样的再把该数据包填写到第三个数据包中

The image displays a network traffic analysis tool interface. On the left, the 'Request' tab is active, showing a POST request to `/web?env=trtc-inspection-test-2bhec82d41e`. The request body is a JSON object with an action, dataVersion, env, and ticket. On the right, the 'Response' tab is active, showing a 200 OK status and a JSON response. The response contains a 'refresh_token' field, which is highlighted with a red box. A red text annotation below the response states: '将这个数值，填写到第三个数据包中' (Copy this value and paste it into the third data packet).

```
Request
Raw Params Headers Hex
POST /web?env=trtc-inspection-test-2bhec82d41e HTTP/1.1
{"action": "auth.signInWithTicket", "dataVersion": "2020-01-10", "env": "prod", "ticket": "70d9573d-50a-4000-8000-000000000000"}

Response
Raw Headers Hex
HTTP/1.1 200 OK
{"requestId": "5d7cbec16b531", "refresh_token": "08cfd03690834e629cff0ca4634de5689076fa5bf84092a881f1", "seqId": "5d7cbec16b531"}
```

将这个数值，填写到第三个数据包中

