公众号：安全先师

更多学习资料及干货请关注微信公众号获取

渗透篇

1、介绍一下自认为有趣的挖洞经历
挖洞也有分很多种类型，一种是以渗透、一种是以找漏洞为主，如果是前者会想各种办法获取权限继而获取想要的的东西完成渗透目标，这类跟 HW 类似，目标各种漏洞不算，要有 Shell，服务器权限才给分，这才是最接近实战渗透，跟某部门有合作的话也是属于这种打击网络犯罪获得权限、传销数据、组织架构，服务器权限、等.....

2、你平时用的比较多的漏洞是哪些？相关漏洞的原理？以及对应漏洞的修复方案？
SQL 注入、密码组合,前者防护分为几种，CDN -> Web -> 数据库 -> 主机,设置最小权限来应对。 密码组合根据个人习惯

3、php/java 反序列化漏洞的原理?解决方案?
php 中围绕着 serialize()，unserialize()这两个函数，序列化就是把一个对象变成可以传输的字符串,如果服务器能够接收我们反序列化过的字符串、并且未经过滤的把其中的变量直接放进这些魔术方法里面的话，就容易造成很严重的漏洞了。

O:7:"chybeta":1:{s:4:"test";s:3:"123";}
这里的 O 代表存储的是对象（object）,假如你给 serialize()传入的是一个数组，那它会变成字母 a。7 表示对象的名称有 7 个字符。"chybeta"表示对象的名称。1 表示有一个值。{s:4:"test";s:3:"123";}中，s 表示字符串，4 表示该字符串的长度，"test"为字符串的名称，之后的类似。 当传给 unserialize() 的参数可控时，我们可以通过传入一个精心构造的序列化字符串，从而控制对象内部的变量甚至是函数。

JAVA Java 序列化是指把 Java 对象转换为字节序列的过程便于保存在内存、文件、数据库中，ObjectOutputStream 类的 writeObject() 方法可以实现序列化。 Java 反序列化是指把字节序列恢复为 Java 对象的过程，ObjectInputStream 类的 readObject() 方法用于反序列化。

4、如果一台服务器被入侵后,你会如何做应急响应?
1.准备相关的工具，查后门等工具
2.初步判断事件类型,事件等级。
3.抑制范围，隔离使爱害面不继续扩大
4.查找原因，封堵攻击源。
5.业务恢复正常水平.
6.总结，报告，并修复、监控
以上是常规的回答，想知道你是否有这方面应急相关的经验，像这类问题甲方面试比较多。

5、你平时使用哪些工具?以及对应工具的特点?
AWVS、Masscan、BurpSuite
AWVS 常规漏洞扫描，masscan 快速查找端口，burp 重复提交数据包
想知道是否有自己开发工具,如果没有你对每个安全工具有哪些独特的见解以及别人不知道的技巧用法。如：awvs 如何批量扫描？burpsuite 如何爆破 401、脱库等、等等...

6、如果遇到 waf 的情况下如何进行 sql 注入/上传 Webshell 怎么做？请写出曾经绕过 WAF 的经过(SQLi，XSS，上传漏洞选一)
PHP 上传，无法上传 php、解析、后台没有办法拿到，只有一处点可以上传。通过 Windows 特性 shell.php::$DATA，是一个项目管理系统

7、如何判断 sql 注入，有哪些方法
提交错误语句是否有异常，除此之外这些显示的错误可以通过 sleep,修眠语句执行 5 秒等，除此之外通过 DNSlog 判断是还有传回值

8、如何判断 SQL 注入漏洞成因，如何防范？注入方式有哪些？除了数据库数据，利用方式还有哪些？
select * from news where id = '$SQL';
当程序执行访问新闻等一些操作都会执行到 sql 语句进行调用，如果在此调用过程中，提交了不合法的数据，而数据库无法识别则会报错。也就是一切输入都是有害的。

注入类型有 6 种，可以参考 SQLMAP，报错、盲注、联合、时间、内联、堆叠
注入提交方式：GET、POST、Cookies、文件头
利用方式：具体看什么数据库类型，像 SQLSERVER 可以命令执行，MYSQL 写 shell 有些权限大也可以执行命令但是条件是在 IINUX 环境下。
防范:边界,CDN -> 脚本语言过滤 -> 数据库过滤最小权限 -> 主机

9、为什么有的时候没有错误回显
没有进行错误打印或者错误屏蔽

10、宽字符注入的原理？如何利用宽字符注入漏洞，payload 如何构造？
在 mysql 中使用了 gbk 编码，占用 2 个字节,而 mysql 的一种特性,GBK 是多字节编码，它认为两个字节就代表一个汉字，所以%df 时候会和转义符\ %5c 进行结合,所以单引号就逃逸了出来,当第一个字节的 ascii 码大于 128，就可以了。

11、CRLF 注入的原理
CRLF 注入在 OWASP 里面被称为 HTTP 拆分攻击（HTTP Splitting）CRLF 是"回车 + 换行"（\r\n）的简称,在 HTTP 协议中，HTTP Header 与 HTTP Body 是用两个 CRLF 分隔的，浏览器就是根据这两个 CRLF 来取出 HTTP 内容并显示出来。所以，一旦我们能够控制 HTTP 消息头中的字符，注入一些恶意的换行

12、mysql 的网站注入，5.0 以上和 5.0 以下有什么区别？
5.0 以下没有 information_schema 这个系统表，无法列表名等，只能暴力跑表名。
5.0 以下是多用户单操作，5.0 以上是多用户多操做。

13、php.ini 可以设置哪些安全特性
禁用 PHP 函数
允许 include 或打开访问远程资源

14、php 的%00 截断的原理是什么？
因为在 C 语言中字符串的结束标识符%00 是结束符号，而 PHP 就是 C 写的，所以继承了 C 的特性，所以判断为%00 是结束符号不会继续往后执行

条件：PHP<5.3.29，且 GPC 关闭

15、webshell 检测，有哪些方法
grep、关键词、关键函数
安全狗、D 盾

16、php 的 LFI，本地包含漏洞原理是什么？写一段带有漏洞的代码。手工的话如何发掘？如果无报错回显，你是怎么遍历文件的？

```
if ($_GET['file']){
    include $_GET['file'];
}
```

包含的文件设置为变量，并且无过滤导致可以调用恶意文件 还可以对远程文件包含，但需要开启 allow_url_include = ON 通过测试参数的地方进行本地文件/etc/passwd 等包含 如何存在漏洞而且没有回显，有可能没有显示在页面而是在网页源代码中，除些可以利用 DNSlog 进行获取包含的信息。从 index.php 文件一级级往读取 也可以利用 PHP 封装协议读取文件

17、说说常见的中间件解析漏洞利用方式
IIS 6.0

/xx.asp/xx.jpg "xx.asp"是文件夹名
IIS 7.0/7.5

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
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%00.php
Apache

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

18、mysql 的用户名密码是存放在那张表里面？mysql 密码采用哪种加密方式？
mysql -> users
SHA1

19、Windows、Linux、数据库的加固降权思路，任选其一
禁用 root
禁止远程访问

禁止写入
单独帐号
禁止执行 system 等函数

20、你使用什么工具来判断系统是否存在后门
Chkrootkit
Rkhunter

31、如何绕过 CDN 获取目标网站真实 IP，谈谈你的思路?
类似 phpinfo、网站信息
C 段、子域名
历史解析记录
DDOS
zmap 全网扫描识别 http 头
网站域名管理员邮箱，注册过的域名等相关信息关联

22、如果给你一个网站,你的渗透测试思路是什么？在获取书面授权的前提下。
其实这是一个非常大的话题，渗透大部分思路都是如此，而面试官是想听到你回答不一样的答案让人眼前一亮 如何才做到让人眼前一亮都需要看你的经验，把你实践的过程拿出来说，以及遇到什么问题如何解决，最终取得成果 渗透其它大同小异,而做为渗透者知识的储备、基础扎实、耐心、细心都是必不可少。

23、谈一谈 Windows 系统与 Linux 系统提权的思路?
Windows
Windows 服务比较多所以方法也如此，最基本的就是 Exp 提权，数据库 SQLServer、MYSQL UDF 等、第三方软件提权。
除此之外提权的成功与否和在于信息收集也非常重要，你对这台服务器和管理员了解多少。
windows 权限提升(二)
Linux

Linux 也是类似，除了 EXP 或者高版本的内核无法提权之外，通过第三方软件和服务，除了提权也可以考虑把这台机器当跳版，
达到先进入内网安全防线最弱的地方寻找有用的信息，再迂回战术。
linux 权限提升

Brief
枚举脚本
以 root 权限运行的程序
用户安装的软件

弱口令或者明文密码
只能内部访问的服务
suid 和 guid 错误配置
滥用 sudo 权限
以 root 权限运行的脚本文件
错误的路径配置
计划任务
未挂载的文件系统
NFS 共享
通过键盘记录仪窃取密码
其它有用的和提权相关的东西
内核提权

24、列举出您所知道的所有开源组件高危漏洞(十个以上)
Tomcat
Nginx
Apache
Hadhoop
Docker
Jenkins
Zenoss
Jboss
MongoDB
Redis
GlassFish

25、反弹 shell 的常用命令？一般常反弹哪一种 shell？为什么？
nc -lvvp 7777 -e /bin/bash
bash 是交互式,否则像 useradd 无法执行交互

26、CMD 命令行如何查询远程终端开放端口
tasklist /svc
netstat -ano

27、服务器为 IIS+PHP+MySQL，发现 root 权限注入漏洞，讲讲你的渗透思路
可以读取 IIS 信息，知道路径,如果像 WAMMP 类似构建，通过@@datadir 知道数据库路径也可以猜测网站路径。
或者直接写 Shell

28、请写出 Mysql5 数据库中查询库'helloworld'中'users'表所有列名的语句
select COLUMN_NAME from information_schema.COLUMNS where table_name = 'your_table_name' and table_schema = 'your_db_name';

29、下面这段代码存在漏洞吗？如果存在请说出存在什么漏洞并利用
http://www.exp.com/1.php
<?php
$s_func = $_GET['s_func'];
$info = $_GET['info'];
$s_func($info);
?>
代码执行,通过 assert 调用

30、udf 提权
MySQL 可以自定义函数,通过自定义函数做到类似 xp_cmdshell 效果

31、SQL 头注入点
UserAgent
Referer
Cookie
X-FOR-IP

32、php 中命令执行涉及到的函数
eval()
assert()
system()
exec()
shell_exec()

33、SSRF 漏洞的成因 防御 绕过
模拟服务器对其它资源进行请求 IP 探测, 如果想漏洞利用必需要构造好 Payload 禁止跳转, 限制协议，内外网限制，URL 限制 针对 IP 格式

34、mysql 写 shell 有几种方法
outfile、dumpfile、开启 log 写 webshell

35、Metasploit 打开反向监听的命令
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp

36、应急响应的步骤
1.准备已经编译好的工具以及取证分析等工具干净可靠放 U 盘
2.初步判断事件的类型，是被入侵、ddos 还是其它的原因
3.首先抑制范围、影响范围，隔离使受害面不继续扩大。
4.寻找原因，封堵攻击源。
5.把业务恢复至正常水平
6.监控有无异常，报告、管理环节的自省和改进措施。

37、有哪些反向代理的工具?
reGeirg、EW、lcx、Ngrok、frp

38、有什么比较曲折的渗透经历
这个问题想知道你工作渗透到什么样的程度，只是简单的漏扫搬砖，还是有毅力坚持完成整个渗透，如：对目标不放弃，坚持一个月最终通过各种手段，曲折的过程拿下目标。

39、UpdateTime:2019.5.11

怎么查找域控

方法有很多

1.通过 DNS 查询

dig -t SRV _gc._tcp.lab.ropnop.com

dig -t SRV _ldap._tcp.lab.ropnop.com

dig -t SRV _kerberos._tcp.lab.ropnop.com

dig -t SRV _kpasswd._tcp.lab.ropnop.com

2.端口扫描

域服务器都会开启 389 端口，所以可以通过扫描端口进行识别。

3.其实很多域环境里，DNS 服务器就是域控制根本不需要怎么找。

4.各种命令

dsquery

net group "Domain controllers"

nltest /DCLIST:pentest.com

.....

前端篇

1、什么是同源策略?
源就是主机、协议、端口名的一个三元组 同源策略(Same Origin Policy, SOP)是 Web 应用程序的一种安全模型，被广泛地应用在处理 WEB 内容的各种客户端上，比如各大浏览器，微软的 Silverlight，Adobe 的 Flash/Acrobat 等等。

2、XSS 能用来做什么?
网络钓鱼、窃取用户 Cookies、弹广告刷流量、具备改页面信息、删除文章、获取客户端信息、传播蠕虫

3、XSS 的三种类型，防御方法
反射型、Dom Base XSS、存储型 防御方法这个只能说个大概，毕竟这是一个比较大的话题，而且防御的方法还得看所在的业务等。 从网络层、主机层、Web 层、数据库，通过 CDN 都有过滤常见一些攻击手法，但不能有 CDN 就以为可以了，添加 CDN 只是让攻击成本增高，开启 HttpOnly，以防确实存在避免 cookies 被获取，CSP 策略、再就是语言中提供的函数对输入过滤，以及输出编码以及 ModSecurity 类的防火墙。

4、存储型 xss 原理?
如网站留言版，把插入的记录存储在数据库中，插入的代码会一直留在页面上，当其它用户访问会从数据库中读取并触发漏洞。

5、你怎么理解 xss 攻击?
是一种被动型，在不知道的情况下触发类似无感型，在渗透很多情况下平常的渗透手段以及取得目标的信息，而 XSS 就能轻松获取，类似 QQ 邮箱你不可能渗透这么大的互联网就算可以时间成本都非常的高，XSS 比较有针对性。

6、如何快速发现 xss 位置?

各种输入的点，名称、上传、留言、可交互的地方，一切输入都是在害原则。

7、Dom xss 原理/防范
DOM 型 XSS 并不需要服务器解析响应的直接参与触发 XSS 靠的是浏览器 DOM 解析 DOM—based XSS 漏洞是基于文档对象模型 Document Objeet Model,DOM)的一种漏洞。

cument.getElementByld("a").innerHTML="yyyyyy";
在输入点过滤敏感关键字

8、DOM 型 XSS 与反射型 XSS 区别?
DOM 型就是 JavaScript 中的 Document 对象 HTML 注入，直接浏览器处理。

9、如何使得前端 referer 为空
通过地址栏输入、从书签里面选择或者浏览器的插件 BurpSuite 修改。

10、cookie 参数，security 干什么的
Httponly：防止 cookie 被 xss 偷

https：防止 cookie 在网络中被偷

Secure：阻止 cookie 在非 https 下传输，很多全站 https 时会漏掉

Path :区分 cookie 的标识，安全上作用不大，和浏览器同源冲突

11、如果 SRC 上报了一个 XSS 漏洞，payload 已经写入页面，但未给出具体位置，如何快速介入?
看是否什么类型的 XSS，XSS 反射型看提交的地址，指的参数是哪个位置，通过这个页面进行 fuzzing 测试。如果是存储型页面查找关键字。

12、XSS， CSRF， CRLF 比较容易弄混，说说三者的原理，防御方法
CSRF 跨站请求伪造，构靠已知的所有参数让对方访问，

防护 CSRF: 防御原理：不让你那么容易伪造请求(cookie 中加入随机数，要求请求中带上，而攻击者获取不到 cookie 中的随机数,验证 HTTP Referer 字段,在请求地址中添加 takon 验证

CRLF 原理:

HTTP 拆分攻击（HTTP Splitting），CRLF 是"回车 + 换行"（\r\n）的简称。

在 HTTP 协议中，HTTP Header 与 HTTP Body 是用两个 CRLF 分隔的，浏览器就是根据这两

个 CRLF 来取出 HTTP 内容并显示出来。所以，一旦我们能够控制 HTTP 消息头中的字符，注入一些恶意的换行，这样我们就能注入一些会话 Cookie 或者 HTML 代码，所以 CRLF Injection 又叫 HTTP Response Splitting，简称 HRS。

13、csrf 如何不带 referer 访问
通过地址栏，手动输入；从书签里面选择；通过实现设定好的手势。上面说的这三种都是用户自己去操作，因此不算 CSRF。

跨协议间提交请求。常见的协议：ftp://,http://,https://,file://,javascript:,data:.最简单的情况就是我们在本地打开一个 HTML 页面，这个时候浏览器地址栏是 file://开头的，如果这个 HTML 页面向任何 http 站点提交请求的话，这些请求的 Referer 都是空的。那么我们接下来可以利用 data:协议来构造一个自动提交的 CSRF 攻击。当然这个协议是 IE 不支持的，我们可以换用 javascript:

14、CSRF 成因及防御措施；如果不用 token 如何做防御?
X-Frame-Options
DENY(禁止被 加载进任何 frame)
SAMEORIGIN(仅允许被加载进同域内的 frame)

X-XSS-Protection
0（表示禁止用这个策略）
1（默认，对危险脚本做一些标志或修改，以阻止在浏览器上熏染执行。）
1;mode=block（强制不熏染，在 Chrome 下直接跳转到空白页，在 IE 下返回一个#符号）

这个策略仅针对反射型，对付不了存储型 XSS，能识别出反射型是因为提交请求的 URL 中带有可疑的 XSS 代码片段。

X-Content-Security-Policy

15、Xss worm 原理
攻击者发现目标网站存在 XSS 漏洞，并且可以编写 XSS 蠕虫。 利用一个宿主（如博客空间）作为传播源头进行 XSS 攻击。

16、Cookie 的 P3P 性质
HTTP 响应头的 p3 字段是 W3C 公布的一项隐私保护推荐标准，该字段用于标识是否允许目标网站的 cookie 被另一个域通过加载目标网站而设置或发送，仅 IE 执行了该策略。

17、CSRF 有何危害?
篡改目标网站上的用户数据 盗取用户隐私数据 传播 CSRF 蠕虫