

目标

- 1. 鉴定黑白
- 2. 详细静态分析，根据基础静态分析判定的结果，来详细分析样本的行为

原理

通过IDA阅读反汇编和伪代码，分析样本行为

实践过程

Lab05-01.dll

基础静态分析

VT检测

- 黑样本
- 58/68检出率，判定为黑样本。
- 后门

根据VT上众多扫描引擎的病毒名，判断为后门样本

58 / 68

Community Score

58 engines detected this file

eb1079bdd96bc9cc19c38b76342113a09666aad47518ff1a7536eebff8aad54a

X-doorc

armadillo corrupt overlay pedll

130.94 KB

Size

2019-09-03 06:06:26 UTC

5 days ago

DETECTION	DETAILS	COMMUNITY 5
Acronis	Suspicious	Ad-Aware Backdoor.XIW
AegisLab	Trojan.Win32.Agent.4lc	AhnLab-V3 Backdoor/Win32.Agent.R9408
Alibaba	Backdoor:Win32/Tdica.f6f5dba90	ALYac Backdoor.XIW
Antiy-AVL	Trojan[Backdoor]/Win32.Agent	SecureAge APEX Malicious
Arcabit	Backdoor.XIW	Avast Win32:Agent-OLH [Trj]
AVG	Win32:Agent-OLH [Trj]	Avira (no cloud) BDS/Agen.twe.134160
BitDefender	Backdoor.XIW	ClamAV Win.Trojan.Agent-118820
CMC	Backdoor.Win32.Agent!O	Comodo Backdoor.Win32.Agent.~TRE@92t1
CrowdStrike Falcon	Win/malicious_confidence_100% (D)	Cylance Unsafe

鉴定黑白后，进行对样本恶意行为进行进一步分析

信息收集

时间戳	信息类型	内容
文件类型		Mon Jun 09 20:49:29 2008
壳特征		32位GUI型DLL文件
		未加壳

从收集到的信息上看，是一款比较老的DLL恶意文件

简单行为分析

监控登陆窗口，记录登陆用户名密码

根据导入表函数：OpenDesktopA■SetThreadDesktop■和字符串表中的Winlogon，在线搜索发现相关API和字符可以实现这样的功能

枚举盘符

GetLogicalDrives■GetDriveTypeA，根据这些API可以知道

获取计算机信息

GetVersionExA■GetComputerNameA

创建服务，修改服务等操作

CreateServiceA■RegisterServiceCtrlHandlerA■StartServiceA■

文件操作，遍历、复制、删除等

WriteFile■CopyFileA■MoveFileExA■DeleteFileA■FindNextFileA FindFirstFileA

Socket连接

recv■send■connect■ntohs■htons

DLL注入

CreateToolhelp32Snapshot■Process32First■Process32Next■VirtualAllocEx■WriteProcessMemory■CreateRemoteThread

命令执行

WinExec■Sleep

注册表

- 设置IE浏览器路径SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\IEXPLORE.EXE
- 服务配置SYSTEM\CurrentControlSet\Services\等
- 获取设备信息HARDWARE\DEVICEMAP\VIDEO

反虚拟机

Found Virtual Machine,Install Cancel.

HTTP、FTP

```
anonymous
FTP://
ftp://
Content-Length:
HTTP/1.1 5
HTTP/1.1 3
HTTP/1.1 4
Expires: 0
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Connection: Keep-Alive
User-Agent: Mozilla/4.0 (compatible; MSIE 6.00; Windows NT 5.1)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/msword, application/x-msie
Host:
HTTP/1.1
GET
HTTP://
http://
```

命令参数

```
-warn
-erro
-stop
```

```
-shutdown
-reboot
attrib -a -r -s -h "%s"
rundll32.exe %s,StartEXS %s:%s
```

衍生文件

.\vmselfdel.bat

小结

简单从导入表和字符串表中粗略概括以上恶意行为，下面用IDA对照上面的信息，详细分析

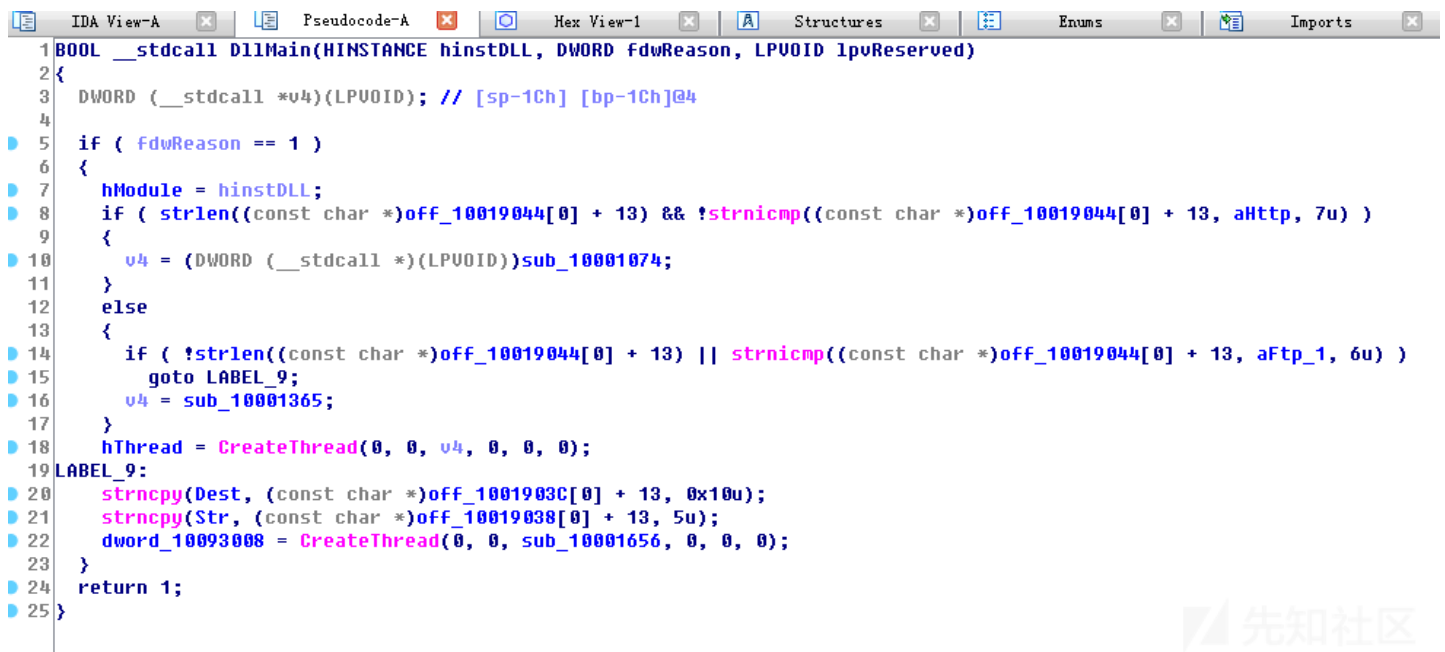
高级静态分析

这里我们大概看一下，简单从导入表和导出表来看他的行为

1. 入口位置：DllMain

IDA直接识别出入口位置，并用其最重要的功能之一的F5大法来查看伪代码。如果使用rundll32.exe启动这个DLL文件，就会从这里开始执行。

很明显从下面API可以看出这里有创建多条线程的操作

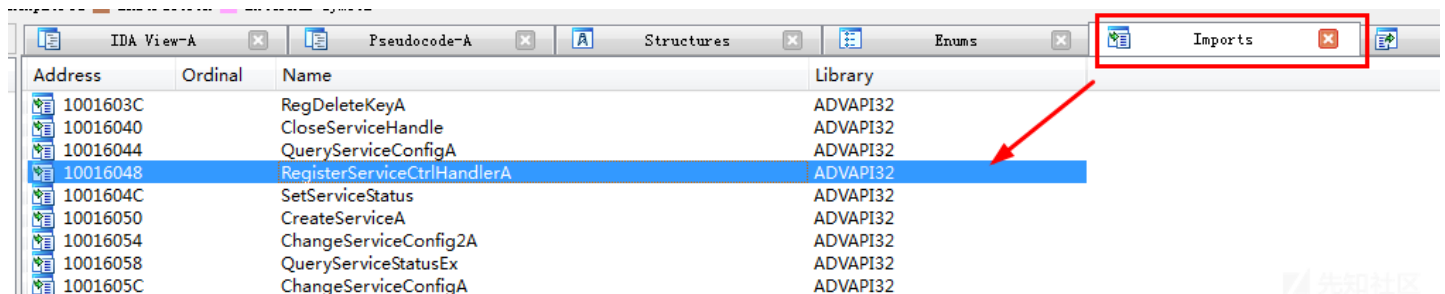


```
1 BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
2 {
3     DWORD (__stdcall *u4)(LPVOID); // [sp-1Ch] [bp-1Ch]@4
4
5     if ( fdwReason == 1 )
6     {
7         hModule = hinstDLL;
8         if ( strlen((const char *)off_10019044[0] + 13) && !strnicmp((const char *)off_10019044[0] + 13, aHttp, 7u) )
9         {
10             u4 = (DWORD (__stdcall *) (LPVOID))sub_10001074;
11         }
12         else
13         {
14             if ( !strlen((const char *)off_10019044[0] + 13) || strnicmp((const char *)off_10019044[0] + 13, aFtp_1, 6u) )
15                 goto LABEL_9;
16             u4 = sub_10001365;
17         }
18         hThread = CreateThread(0, 0, u4, 0, 0, 0);
19 LABEL_9:
20         strncpy(Dest, (const char *)off_1001903C[0] + 13, 0x10u);
21         strncpy(Str, (const char *)off_10019038[0] + 13, 5u);
22         dword_10093008 = CreateThread(0, 0, sub_10001656, 0, 0, 0);
23     }
24     return 1;
25 }
```

2. 导入表

直接定位关键函数。

跟踪关键函数 RegisterServiceCtrlHandlerA，接着用交叉引用和F5大法就可以跟到打开服务的行为



Address	Ordinal	Name	Library
1001603C		RegDeleteKeyA	ADVAPI32
10016040		CloseServiceHandle	ADVAPI32
10016044		QueryServiceConfigA	ADVAPI32
10016048		RegisterServiceCtrlHandlerA	ADVAPI32
1001604C		SetServiceStatus	ADVAPI32
10016050		CreateServiceA	ADVAPI32
10016054		ChangeServiceConfig2A	ADVAPI32
10016058		QueryServiceStatusEx	ADVAPI32
1001605C		ChangeServiceConfigA	ADVAPI32

```

>
5 Dest = 0;
7 memset(&v9, 0, 0x3FCu);
3 v10 = 0;
2 v11 = 0;
8 sprintf(&Dest, aServiceSstartT, lpServiceName);
1 v2 = (struct _QUERY_SERVICE_CONFIGA *)LocalAlloc(0x40u, 0x400u);
2 if ( v2 )
3 {
4     v3 = OpenSCManager(0, 0, 0xF003Fu);
5     hSCObject = v3;
6     if ( v3 )
7     {
8         v5 = OpenServiceA(v3, lpServiceName, 0xF01FFu);
9         if ( v5 )
10         {
11             if ( !QueryServiceConfigA(v5, v2, 0x400u, &pcbBytesNeeded) )
12                 sprintf(&Dest, aQueryRegistryS);
13             v6 = v2->dwStartType;
14             *a2 = v6;
15             sprintf(&Dest, aQueryServiceSt, v6);
16         }
17         else
18         {
19             sprintf(&Dest, aMyqueryservice, lpServiceName);
20         }
21         sub_10003592(&Dest, v7);
22         result = CloseServiceHandle(hSCObject);
23     }
24     else
25

```

3.字符串表

根据可以的网络访问字符串，再结合跟进去后看见的socket连接行为，很明显是后门访问获取特定指令来进行HTTP请求

Address	Length	Type	String
xdoors_d:1009...	00000017	C	Connection: Keep-Alive
xdoors_d:1009...	00000040	C	User-Agent: Mozilla/4.0 (compatible; MSIE 6.00; Windows NT 5.1)
xdoors_d:1009...	000000AD	C	Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, a...
xdoors_d:1009...	00000007	C	Host:
xdoors_d:1009...	0000000A	C	HTTP/1.1
xdoors_d:1009...	00000005	C	GET
xdoors_d:1009...	00000008	C	HTTP://
xdoors_d:1009...	00000008	C	http://
xdoors_d:1009...	00000011	C	GetLastInputInfo
xdoors_d:1009...	00000008	C	user32.dll

```

225 {
226     *(_DWORD *)a6 = 444;
227     return 0;
228 }
229 if ( inet_addr(&Str) == -1 )
230 {
231     v18 = gethostbyname(&Str);
232     if ( !v18 )
233     {
234         *(_DWORD *)a6 = 499;
235 LABEL_85:
236         WSACleanup();
237         return v6;
238     }
239     Dst = 2; |
240     v19 = atoi(&v70);
241     v68 = htons(v19);
242     memcpy(&v69, *(const void **)v18->h_addr_list, v18->h_length)
243 }
244 else
245 {
246     Dst = 2;
247     v17 = atoi(&v70);
248     v68 = htons(v17);
249     v69 = inet_addr(&Str);
250 }
251 v20 = socket(2, 1, 0);
252 sa = v20;
253 if ( v20 == -1 )
254 {
255     *(_DWORD *)a6 = 400;
256     printf(aSocketError__);
257     goto LABEL_85;
258 }
259 setsockopt(v20, 0xFFFF, 4102, optval, 4);
260 if ( connect(v20, (const struct sockaddr *)&Dst, 16) )

```

4.导出表

根据符号信息可以初步判断是一些安装卸载服务和其他一些行为的操作。

IDA View-A Pseudocode-A Structures Enums Imports Strings window Exports			
Name	Address	Ordinal	
InstallRT	1000D847	1	
InstallSA	1000DEC1	2	
InstallSB	1000E892	3	
PSLIST	10007025	4	
ServiceMain	1000CF30	5	
StartEXS	10007ECB	6	
UninstallRT	1000F405	7	
UninstallSA	1000EA05	8	
UninstallSB	1000F138	9	
DllEntryPoint	1001516D	[main entry]	

跟进InstallSA导出函数发现存在反虚拟机行为。

```

xdoors_d:10094F87 align 4
xdoors_d:10094F88 ; char aFoundVirtualMa[]
xdoors_d:10094F88 aFoundVirtualMa text "UTF-8", 'Found Virtual Machine,Install Cancel.',0
xdoors_d:10094F88 ; DATA XREF: InstallRT+33↑o
xdoors_d:10094F88 ; InstallSA+33↑n

```

```
20 v5 = 0;
21 v6 = 0;
22 GetModuleFileNameA(hModule, &Filename, 0x104u);
23 sprintf(&Dest, a_UMselfdel_bat);
24 v0 = fopen(&Dest, aM);
25 v1 = v0;
26 if ( v0 )
27 {
28     fprintf(v0, a_echoOff);
29     fprintf(v1, aSelfkill);
30     fprintf(v1, aAttribARSHS, &Filename);
31     fprintf(v1, aDelS, &Filename);
32     fprintf(v1, aIfExistSGotoSe, &Filename);
33     fprintf(v1, aDel0);
34 }
35 fclose(v1);
36 return WinExec(&Dest, 0);
37 }
```

先知社区

小结

这个简单分析初步探索一下静态逆向过程。很明显这个过程如果对Windows API不熟的话需要不断的查询,当然我们的关注点应该更专注于恶意行为会用到的API。

点击收藏 | 0 关注 | 1

[上一篇：meterpreter学习笔记](#) [下一篇：【漏洞分析】泛微OA E-colo...](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟贴

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)