

WanaCrypt0r勒索蠕虫完全分析报告

日期：2017-5-13

0x1 前言

360互联网安全中心近日发现全球多个国家和地区的机构及个人电脑遭受到了一款新型勒索软件攻击，比于5月12日国内率先发布紧急预警，外媒和多家安全公司将该病毒命名为“WanaCrypt0r”（直译：“想哭勒索蠕虫”），常规的勒索病毒是一种趋利明显的恶意程序，它会使用非对称加密算法加密受害者电脑内的重要文件进行勒索，除非受害者交出勒索赎金，否则加密文件无法被恢复，而新的“想哭勒索蠕虫”尤其致命，它利用了窃取自美国国家安全局的黑客工具EternalBlue（直译：“永恒之蓝”）实现了全球范围内的快速传播，在短时间内造成了巨大损失。360追日团队对“想哭勒索蠕虫”国内首家对该蠕虫进行了完全的技术分析，帮助大家深入了解此次攻击！

0x2 抽样分析样本信息

MD5: DB349B97C37D22F5EA1D1841E3C89EB4

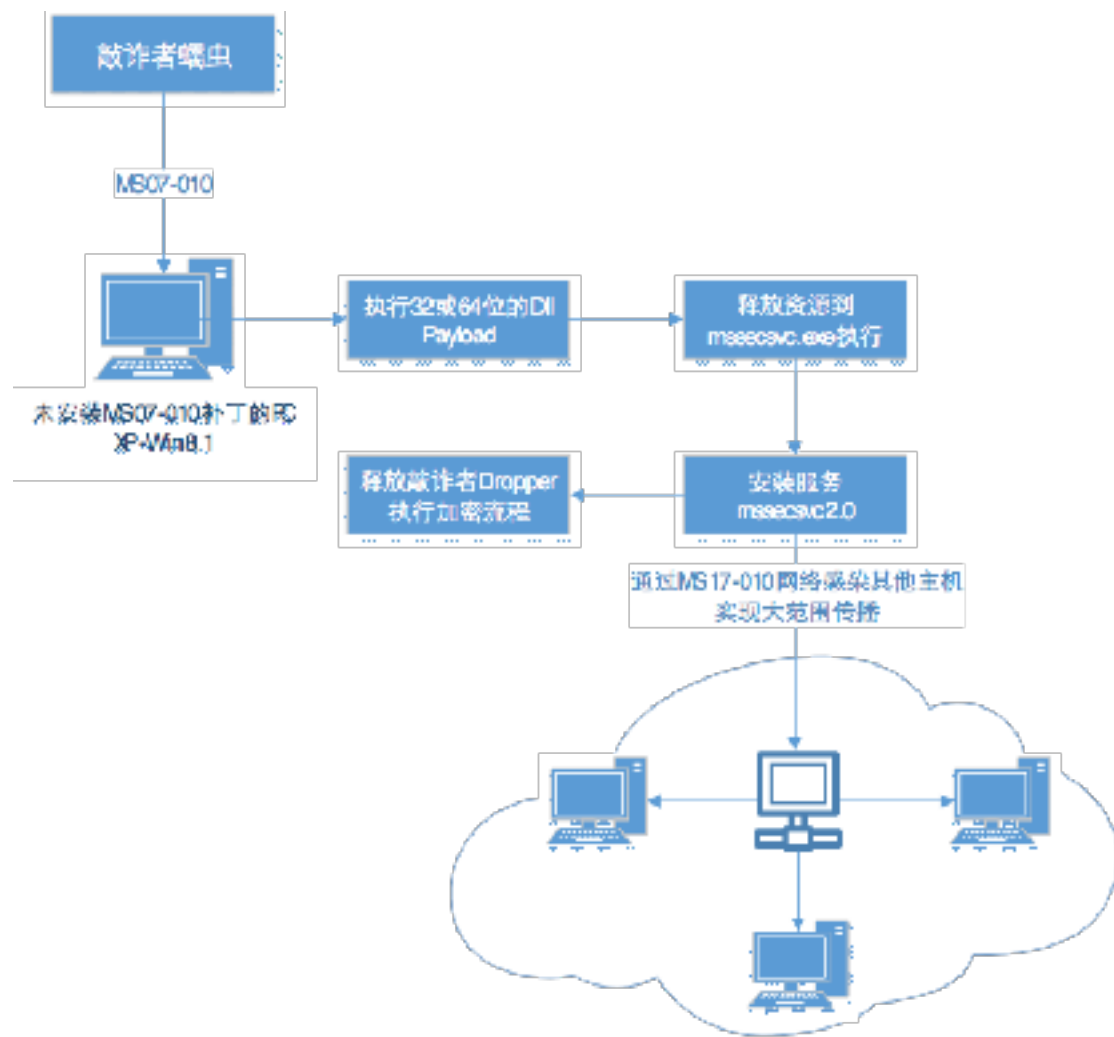
文件大小: 3,723,264

影响面: 除Windows 10外，所有未打MS-17-010补丁的Windows系统都可能被攻击

功能: 释放加密程序,通过MS17-010漏洞实现自身的快速感染和扩散。

0x03 蠕虫的攻击流程

该蠕虫病毒使用了ms017-010漏洞进行了传播，一旦某台电脑中招，相邻的存在漏洞的网络主机都会被其主动攻击，整个网络都可能被感染该蠕虫病毒，受害感染主机数量最终将呈几何级的增长。其完整攻击流程如下



0x04 蠕虫启动逻辑分析

1. 蠕虫启动时将连接固定url: <http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com>
 - a) 如果连接成功,则退出程序
 - b) 连接失败则继续攻击
2. 接下来蠕虫开始判断参数个数,小于2时,进入安装流程;大于等于2时,进入服务流程.
 - a) 安装流程
 - i. 创建服务,服务名称: mssecsvc2.0
 - 参数为当前程序路径 -m security
 - ii. 释放并启动exe程序
 - 移动当前 C:\WINDOWS\tasksche.exe到 C:\WINDOWS\qeriuwjhrf
 - 释放自身的1831资源(MD5: 84C82835A5D21BBCF75A61706D8AB549),到 C:\WINDOWS\tasksche.exe,并以 /i参数启动

b) 服务流程

- i. 服务函数中执行感染功能,执行完毕后等待24小时退出.
- ii. 感染功能

- 初始化网络和加密库,初始化payload dll内存.

a) Payload包含2个版本,x86和x64

```
U2 = EDLL_X86;
IF ( U1 )
    U2 = &DLL_X64;
U3 = *(void **) &FileName[4 * U1 + 260];
x(RU11 + U1) = (int)U3;
memcpy(U3, U2, U1 ? 0x1000 : 0x4000);
x(RU11 + U1) |= U1 ? 0x1000 : 0x4000;
```

b) 功能为释放资源到c:\windows\mssecsvc.exe并执行

- 启动线程,在循环中向局域网的随机ip发送SMB漏洞利用代码

```
mov     edi, 1
push    44h                ; hostshort
mov     word ptr [esp+12Ch+name.sa_data+18h], ax
mov     [esp+12Ch+argp], edi
mov     dword ptr [esp+12Ch+name.sa_data+2], ecx
mov     [esp+12Ch+name.sa_family], 2
call     htons
push    IPPROTO_TCP        ; protocol
push    edi                 ; type
push    AI_NUM              ; ai
mov     word ptr [esp+124h+name.sa_data], ax
call     socket
```

```
if ( q Connect &&5(a1) > 0 )
{
    u1 = (void *)beginthreadex(0, 0, q NS17 010, a1, 0, 0);
    u2 = u1;
    if ( u1 )
    {
        if ( WaitForSingleObject(u1, 0x000000u) == WAIT_TIMEOUT )
            TerminateThread(u2, 0);
        CloseHandle(u2);
    }
}
InterlockedDecrement((volatile LONG *) &FileName[268]);
endthreadex(0);
return 0;
```

0x05 蠕虫利用漏洞确认

通过对其中的发送的SMB包进行分析,我们发现其使用漏洞攻击代码和<https://github.com/rapid7/metasploit-framework>近乎一致,为Eternalblue工具使用的攻击包。

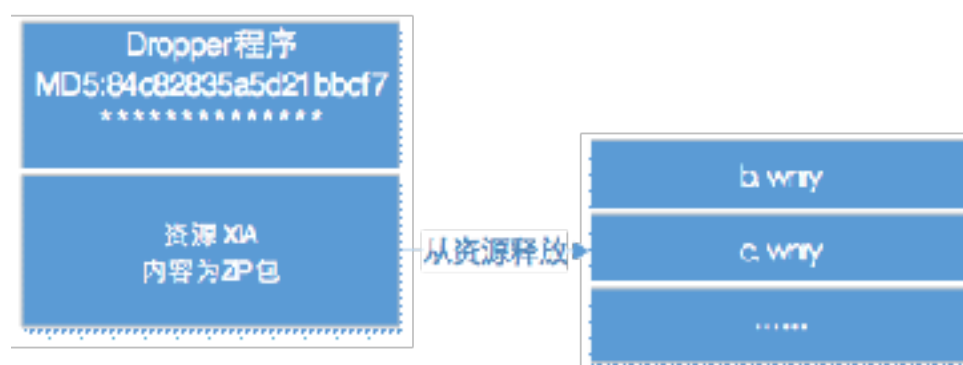
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	01 00 47 97 74 00 10 79 70 70 07 07 04 00 70 A7	1A0 1 A A5 A5															
00000001	10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	2A 8 1 0															
00000002	00 1F 52 A2 F0 FF FF FF 00 45 17 31 02 0F 00 02	20020 0 10 1A															
00000003	E9 23 00 00 00 6A 30 1F A1 2E 02 83 21 64 5B 00	10 j 1001A0															
00000004	40 20 00 00 00 61 04 1F 25 30 07 07 FF 00 00 0A	@ 1a 350909 10															
00000005	00 70 50 6A 02 00 02 70 50 70 00 21 01 02 6A 17	ARI 1A 1A 1A															
00000006	1F 31 4 0 0 0 1F 1A 0 0 0 0 0 0 0 0	00 00 00 00 00															
00000007	00 00 00 6A 33 8B E3 24 11 00 00 F2 33 31 00 45	00 00 00 00 00															
00000008	89 13 00 6A 23 6A 01 03 07 48 81 E0 90 02 00 00	1 10 10 10 10															
00000009	A1 00 33 03 F3 B9 76 00 00 00 31 02 0F 00 5B 33	10 39 00 10 00															
0000000A	1 70 00 00 FA 0 00 70 40 70 07 07 00 00 01 74 00	00 1 10 10															
0000000B	10 70 50 61 01 09 1F 70 70 70 05 07 00 00 00 00	10 00 00 00 00															
0000000C	52 48 00 F2 00 00 1F 1F 1F 1F 1F 1F 1F 1F 1F 1F	20020 00 00 00 00															
0000000D	48 00 00 0A 00 00 00 48 09 02 45 01 EA 20 1F 00	0 00 00 00 00															
0000000E	03 1F 00 F3 63 48 89 24 25 10 00 00 00 65 48 53	A 00 00 00 00															
0000000F	24 25 A2 01 00 00 50 00 00 00 50 50 50 50 50 50	00 00 00 00 00															

DB349B97C37D22F5EA1D1841E3C89EB4 文件:

00000000-0000000F: 24A004000															
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

0x06 蠕虫释放文件分析

蠕虫成功启动后将开始释放文件，流程如下：



释放文件与功能列表，如下：

名称	作用
b.wnry	敲 诈 图 片 资 源
c.wnry	配置文件，包含钱包信息，tor地址
r.wnry	Q&A
s.wnry	压缩包，包含TOR网络组件
t.wnry	加密的PAYLOAD，用于加密文件
u.wnry	解密程序（@WanaDecryptor@.exe）
taskdl.exe	删除临时文件
taskse.exe	在任意的远程桌面的session中运行指定的程序
taskhsvc.exe	网络通讯组件

0x07 关键勒索加密过程分析

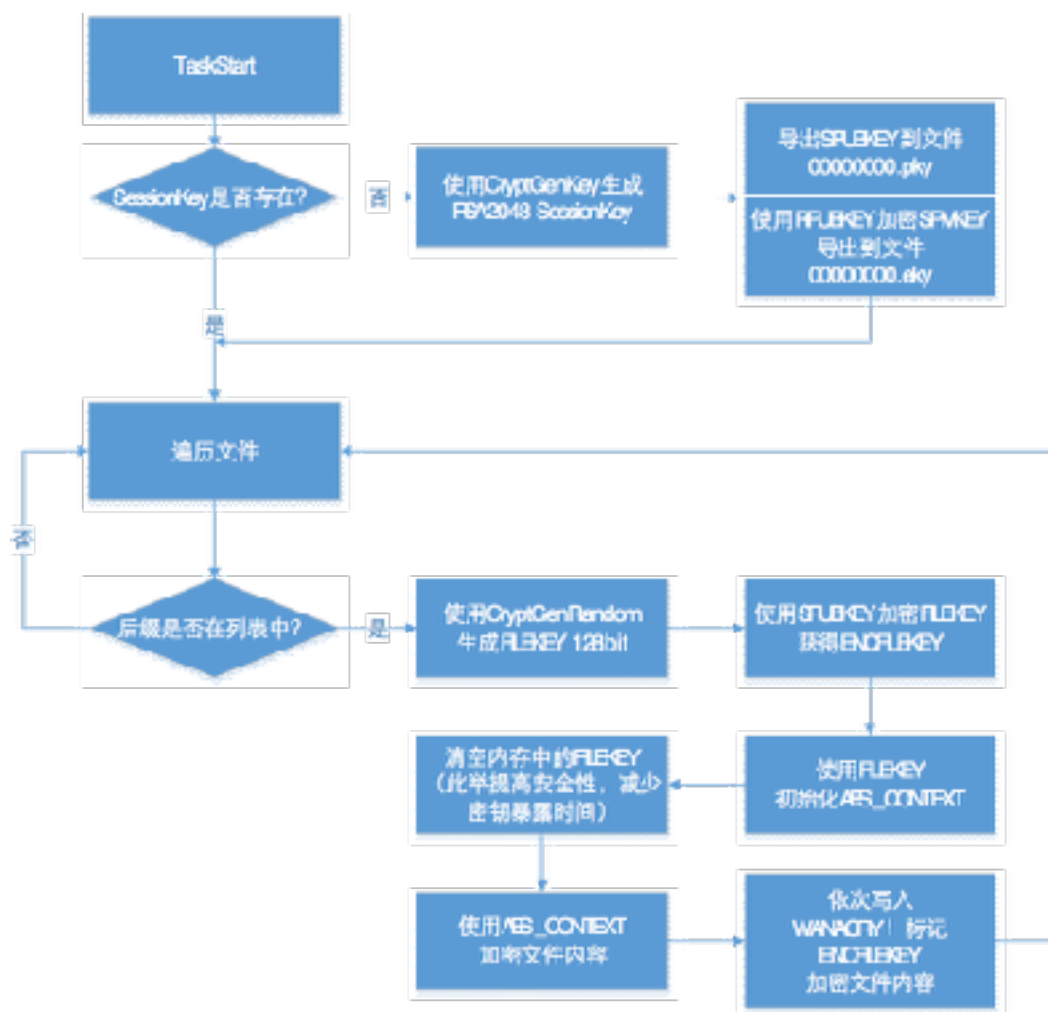
蠕虫会释放一个加密模块到内存，直接在内存加载该DLL。DLL导出一个函数TaskStart用于启动整个加密的流程。程序动态获取了文件系统和加密相关的API函数，以此来躲避静态查杀。

```

10004466 push     edi
10004467 mov     edi, ds:CallProcAddress
10004469 push    offset aCryptAcquireContext ; "CryptAcquireContext"
10004472 push    esi ; hModule
10004473 call    GetProcAddress
10004475 push    offset aCryptImportKey ; "CryptImportKey"
1000447A push    esi ; hModule
1000447B mov     g_CryptAcquireContextA, eax
1000447E call    GetProcAddress
10004482 push    offset aCryptDestroyKey ; "CryptDestroyKey"
10004487 push    esi ; hModule
10004489 mov     g_CryptImportKey, eax
1000448D call    GetProcAddress
10004491 push    offset aCryptEncrypt ; "CryptEncrypt"
10004494 push    esi ; hModule
10004495 mov     g_CryptDestroyKey, eax
1000449A call    GetProcAddress
1000449C push    offset aCryptDecrypt ; "CryptDecrypt"
100044A1 push    esi ; hModule
100044A2 mov     q_CryptEncrypt, eax
100044A7 call    GetProcAddress
100044A9 push    offset aCryptGenKey ; "CryptGenKey"
100044AC push    esi ; hModule
100044AF mov     g_CryptDecrypt, eax
100044B4 call    GetProcAddress
100044B6 mov     ecx, q_CryptAcquireContextA
100044BC mov     g_CryptGenKey, eax
100044C1 test    ecx, ecx

```

整个加密过程采用RSA+AES的方式完成，其中RSA加密过程使用了微软的CryptAPI，AES代码静态编译到dll。加密流程如下图所示。



使用的密钥概述:

RPUBKEY	RSA 2048 Root Public Key, 硬编码于程序中
RPIVKEY	RSA 2048 Root Private Key, 作者持有, 目前未公开
SPUBKEY	RSA 2048 Session Public Key, 每个受害用户唯一的会话密钥 (公钥), 用于加密AES KEY, 导出到文件00000000.pky
SPIVKEY	RSA 2048 Session Private Key, 每个受害用户唯一的会话密钥 (私钥), 用于解密AES KEY, Encrypt (RPUBKEY, SPIVKEY), 即用 RPUBKEY加密后导出到文件00000000.eky
FILEKEY	AES 128Bit KEY, 每一个文件生成一个, 通过CryptGenRandom生成
ENCFILEKEY	被SPUBKEY加密的FILEKEY, 存在于被加密的文件当中

目前加密的文件后缀名列表:

".docx", ".xls", ".xlsx", ".ppt", ".pptx", ".pst", ".ost", ".msg", ".eml", ".vsd", ".vsdx", ".txt", ".csv", ".rtf", ".123", ".wks", ".wk1", ".pdf", ".dwg", ".onetoc2", ".snt", ".jpeg", ".jpg", ".docb", ".docm", ".dot", ".dotm", ".dotx", ".xlsm", ".xlsb", ".xlw", ".xlt", ".xlm", ".xlc", ".xltx", ".xltm", ".pptm", ".pot", ".pps", ".ppsm", ".ppsx", ".ppam", ".potx", ".potm", ".edb", ".hwp", ". "

值得注意的是，在加密过程中，程序会随机选取一部分文件使用内置的RSA公钥来进行加密，这里的目的是解密程序提供的免费解密部分文件功能。

```

62 LABEL_29:
63 if ( u4 == 0x00000000 )
64 {
65     if ( *((_DWORD *)u4 + 582) )
66     {
67         if ( !((unsigned int)rand() % ((_DWORD *)u4 + 582)) )
68         {
69             u10 = *((_DWORD *)u4 + 584);
70             if ( u10 < *((_DWORD *)u4 + 583) )
71             {
72                 u62 = 1;
73                 u64 = u4 + 44;
74                 *((_DWORD *)u4 + 584) = u10 + 1;
75             }
76         }
77     }
78 }
79 u52 = 512;
80 if ( !_init_dbg((int)u64, RspBuf+ec, RspBuf, (int)u62, (int)u52) )
81     goto LABEL_22;

```

能免费解密的文件路径在文件f.wnry中

```
1 C:\reverse\Cil\ydhgs7\Plugin\shourtcute.txt.MNCRY
2 C:\reverse\Cil\ydhgs7\脱壳脚本\Acprotect\U.TFAPROTECT 1.x - ACPROTECT 1.22 VB.txt.MNCRY
3 C:\reverse\Cil\ydhgs7\脱壳脚本\Armadillo\Armadillo 5.xx OEP finder (Standard Protection + Debug
4 C:\reverse\Cil\ydhgs7\脱壳脚本\ASProtect\ASProtect 1.2x - 1(1).3x (Registered) OEP Finder.txt.MN
5 C:\reverse\Cil\ydhgs7\脱壳脚本\SECURITY\SECURITY OEP SCRIPT 1.1 [MAIN EXE].TXT.MNCRY
6 C:\Documents and Settings\All Users\Application Data\Microsoft\User Account Pictures\Default P
7 C:\Python27\include\longintrep.h.MNCRY
8 C:\Python27\include\methobj.h.MNCRY
9 C:\reverse\android\lib\small.jar.MNCRY
```

0x08 蠕虫赎金解密过程分析

首先，解密程序通过释放的taskhsvc.exe向服务器查询付款信息，若用户已经支付过，则将eky文件发送给作者，作者解密后获得dky文件，这就是解密之后的Key
解密流程与加密流程相反，解密程序将从服务器获取的dky文件中导入Key

```
push    offset a08x_dky ; "%08x.dky"
push    ecx              ; best
call    edi ; __imp_printf
```

```

v2 = This;
if ( !g_CryptRequireContext() )
{
    q_DestroyKey(v2);
    return 0;
}
if ( !pFileName )
{
    if ( !q_ImportKeyFromFile(*((_DWORD *)v2 + 1), (int)(char *)v2 + 8, pFileName) )
    {
        q_DestroyKey(v2);
        return 0;
    }
}
else if ( !g_CryptImportKey(*((_DWORD *)v2 + 1), q_InsideKey, 11/2, 0, 0, (char *)v2 + 8) )
{
    q_DestroyKey(v2);
    return 0;
}
return 1;

```

可以看到，当不存在dkey文件名的时候，使用的是内置的Key，此时是用来解密免费解密的文件使用的。

05C0	test eax, eax
75 00	jnz NQManaDec.00404769
00CC	mov ecx, esi
EB 60666666	call NQManaDec.00404776
33C0	xor eax, eax
5F	jmp esi
E2 444444	call ebx
8B4424 8B	mov eax, dword ptr ds:[esp+0x8]
85E8	test eax, eax
75 20	jnz NQManaDec.00404776
8B4E 04	mov ecx, dword ptr ds:[esi+0x4]
0D46 0B	lea eax, dword ptr ds:[esi+0x0]
50	push eax
6A 00	push 0x0
6A 00	push 0x0
6B 94046666	push 0x494
6B 94074266	push NQManaDec.004047D4
51	push ecx
E8 15 04124244	call dword ptr ds:[0x421714]
	api 27.CryptDecryptKey
	mov esi, [esp+0Ch+arg_0]
	mov ecx, [ebx+8]
	lea eax, [esp+0Ch+arg_4]
	push eax
	push esi
	push 0
	push 1
	push 0
	push ecx
	call q_CryptDecrypt

之后解密程序从文件头读取加密的数据，使用导入的Key调用函数CryptDecrypt解密，解密出的数据作为AES的Key再次解密得到原文件。

```

v24 = (unsigned int)v25;
q_AES_Decrypt(*( (_DWORD *)v10 + 306), *( (_DWORD *)v10 + 307), v25, 1);
if ( !q_WriteFile(v5, *( (_DWORD *)v10 + 307), v25, 3026, 0) || v26 != v25 )
goto LABEL_33;
}
SelfFilePointerEx(v5, LiDistanceToMove, 0, 0);

```

总结

该蠕虫在勒索类病毒中全球首例使用了远程高危漏洞进行自我传播复制，危害不小于冲击波和震荡波蠕虫，并且该敲诈者在文件加密方面的编程较为规范，流程符合密码学标准，因此在作者不公开私钥的情况下，很难通过其他手段对勒索文件进行解密（但是，因为其删除文件不彻底，可以通过文件删除恢复工具来恢复部分文件），同时微软已对停止安全更新的xp和2003操作系统紧急发布了漏洞补丁，请大家通过更新MS17-010漏洞补丁及时预先防止被蠕虫被攻击。