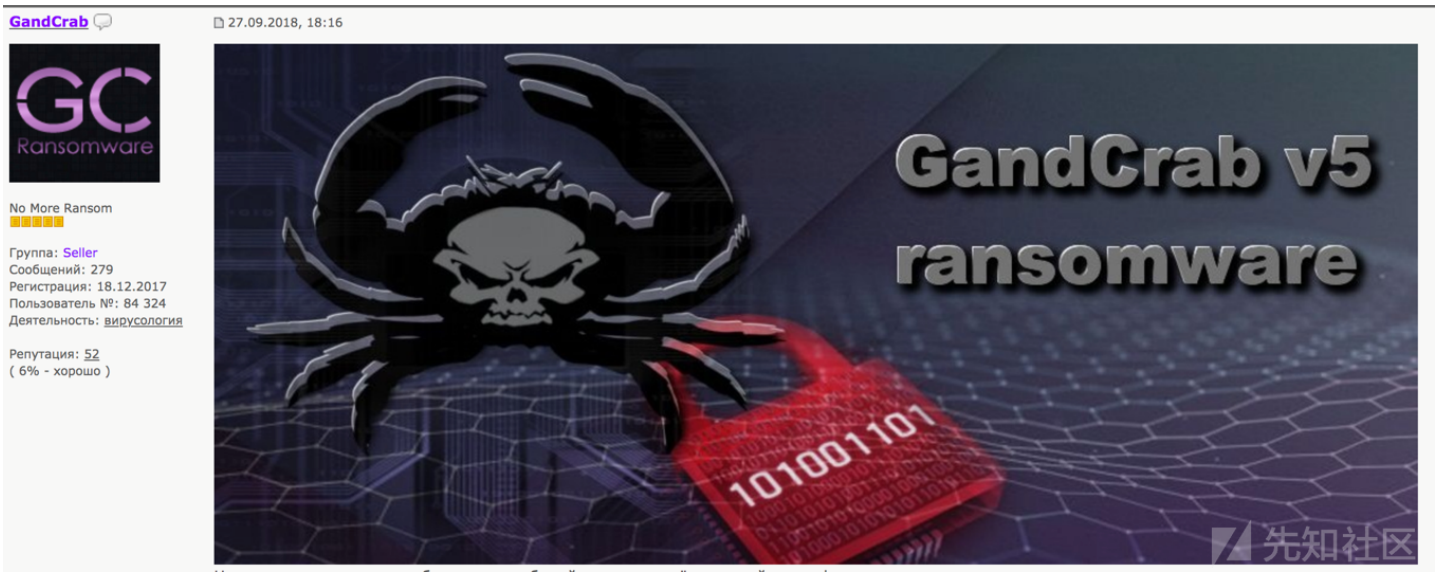
GandCrab V5分析

angel010 / 2018-10-13 23:20:50 / 浏览数 2848 技术文章 技术文章 顶(0) 踩(0)

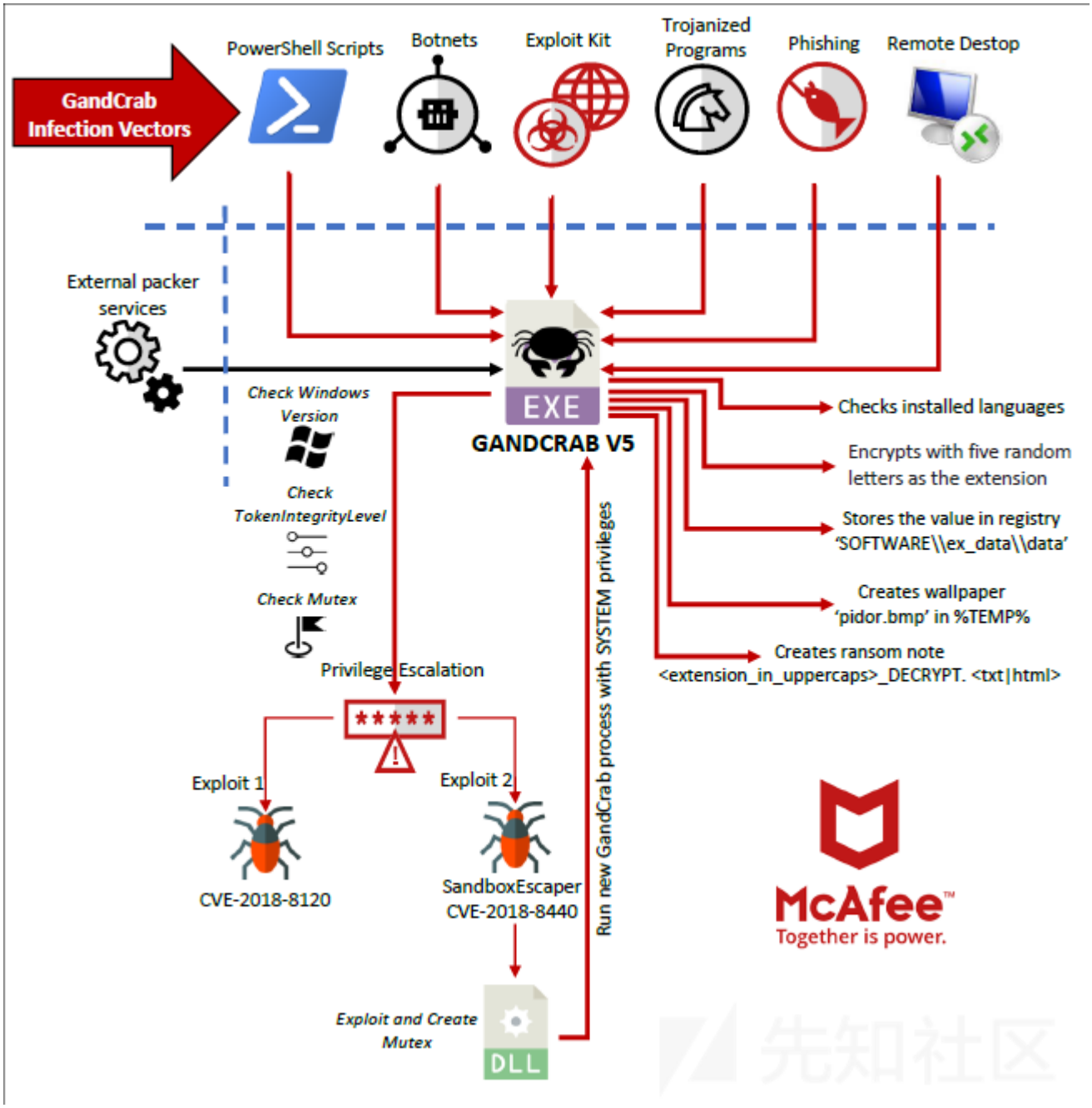
本文翻译自：
https://securingtomorrow.mcafee.com/mcafee-labs/rapidly-evolving-ransomware-gandcrab-version-5-partners-with-crypter-service-for-obfuscation/



2018年9月27日，GandCrab勒索软件称发布了V5版本。本文对GandCrab V5版本进行分析。

GandCrab V5感染

GandCrab V5版本使用了许多机制来感染系统。下图是GandCrab的行为概览。



entry vector

GandCrab有许多的entry vectors:

- 弱安全的远程桌面连接
- 含有链接或附件的钓鱼邮件
- 含有或下载或启动恶意软件的木马化的合法程序
- RigEK、FalloutEK这样的利用套件
- PowerShell脚本或在PowerShell进程的内存中
- 使用Phorpiex的僵尸网络

与其他勒索软件一样，GandCrab的目标是加密受感染系统上的所有文件，等用户支付赎金后再将这些文件解锁。开发者要求受害者以加密货币的形式支付赎金，因为加密货

恶意软件一般都是打包的，研究人员看到过含有DLL的.exe格式的变种。GandCrab也是一种有效的勒索软件即服务，其运营者可以选择需要的版本。

V5.0

V5.0版本一共发布过两个版本。

第一个版本由于编译时的重大错误，主要运行在win7及之后的平台上。V5.0利用两个漏洞来进行权限提升。首先检查操作系统的版本和进程的TokenIntegrityLevel类，如果Subauthority是SECURITY_MANDATORY_LOW_RID (0x1000)，并且通过mutex值检查，就尝试执行漏洞利用。

第二个版本是黑客SandboxEscaper今年8月在Twitter和GitHub上发布的漏洞利用：

- GitHub页面为https://github.com/SandboxEscaper/randomrepo
- Twitter页面为https://twitter.com/sandboxescaper

该漏洞利用尝试使用Windows操作系统处理高级本地过程调用（advanced local procedure call）不当时任务系统（Task System）的漏洞。GandCrab作者称没有CVE的漏洞利用，但实际上有CVE-2018-8440。该漏洞利用影响win7到win10的服务器。更多参见对CVE-2018-8440的分析。

在V5.0的第一个发布版本中，恶意软件作者用正常的函数调用来写利用代码。因此编译时，二进制文件的IAT充满了调用所需的DLL。而DLL并不存在于Windows Vista和XP系统中，因此恶意软件不能运行在win7以前的系统中，编译时就会出错。

```
.idata:004152E4 ;
.idata:004152E4 ; Imports from XPSPRINT.DLL
.idata:004152E4 ;
.idata:004152E4                 extrn StartXpsPrintJob:dword ; CODE XREF: .text:0040B58A↑p
.idata:004152E4                                              ; DATA XREF: .text:0040B58A↑r ...
```

xpsprint.dll不能运行在Windows XP和Vista中

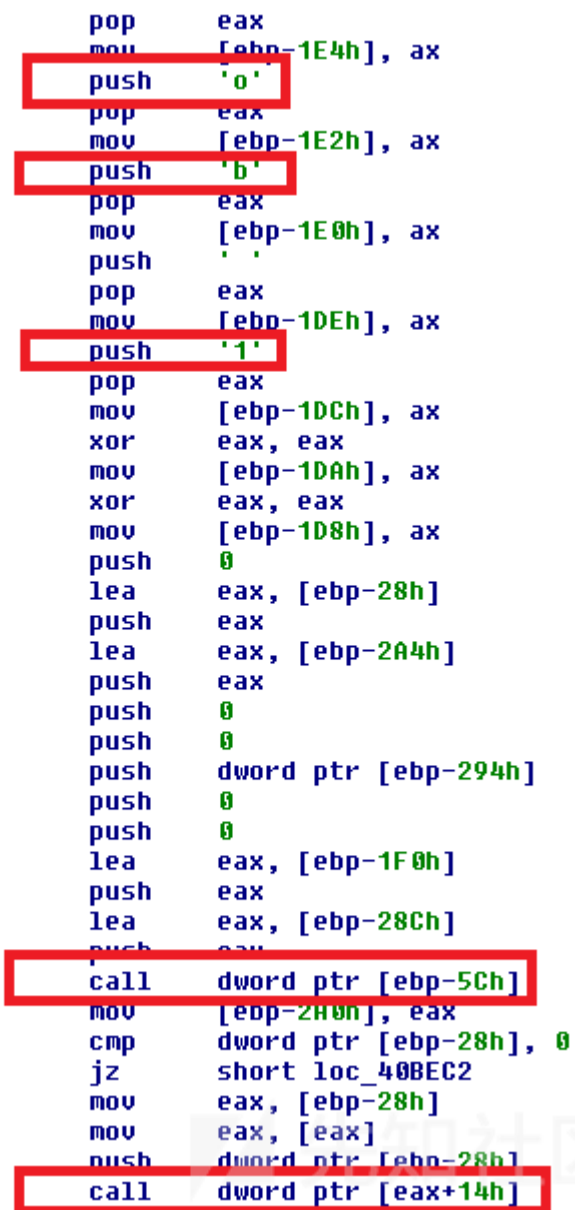
```
                xor     eax, eax
                mov     [ebp-1BCh], ax
                push    0
                lea     eax, [ebp-2Ch]
                push    eax
                lea     eax, [ebp-260h]
                push    eax
                push    0
                push    0
                push    dword ptr [ebp-274h]
                push    0
                push    0
                lea     eax, [ebp-1D4h]
                push    eax
                lea     eax, [ebp-254h]
                push    eax
                call    ds:StartXpsPrintJob
                mov     [ebp-264h], eax
                cmp     dword ptr [ebp-2Ch], 0
                jz      short loc_40B5A7
                mov     eax, [ebp-2Ch]
                mov     eax, [eax]
                push    dword ptr [ebp-2Ch]
                call    dword ptr [eax+14h]

loc_40B5A7:                             ; CODE XREF: .text:0040B59A↑j
                call    ds:CoUninitialize
                push    2710h
                push    dword ptr [ebp-268h]
                call    ds:WaitForSingleObject
                cmp     dword ptr [ebp-264h], 0
                jl      short loc_40B5CC
                xor     eax, eax
                inc     eax
                jmp     short loc_40B5CE
```

使用直接调用的漏洞利用

该版本在加密了用户的文件后会发布一个HTML文件，但该文件没有加密用户文件所需的信息，所以该文件是有缺陷的。

第二个发布的版本使用动态调用，并混淆了漏洞利用中的字符串，如下图所示。



使用动态调用和混淆字符串的漏洞利用

第二个漏洞利用涵盖了CVE-2018-8120
，可以在win7、Windows server 2008 R2和Windows server
2008上进行权限提升。因为system进程token的对象有错误，改变了恶意软件中的token，最终导致恶意软件以system权限运行了。

```
loc_402960:                             ; CODE XREF: .text:00402947↑j
                push    dword ptr [ebp-0E0h]
                call    ds:SetProcessWindowStation
                test    eax, eax
                jnz     short loc_402987
                push    8000h
                push    0
                push    dword ptr [ebp-8]
                call    ds:VirtualFree
                xor     eax, eax
                jmp     loc_402FC6
; ---------------------------------------------------------------------------

loc_402987:                             ; CODE XREF: .text:0040296E↑j
                push    180h
                mov     dl, 90h
                lea     ecx, [ebp-558h]
                call    sub_40163C
                pop     ecx
                lea     eax, [ebp-558h]
                push    eax
                push    20h
                push    1
                push    1
                push    60h
                call    ds:CreateBitmap
                mov     [ebp-20h], eax
                lea     eax, [ebp-558h]
                push    eax
                push    20h
                push    1
                push    1
                push    60h
                call    ds:CreateBitmap
```


执行CVE-2018-8120漏洞利用

恶意软件会检查操作系统的版本和用户的类型，以确定在应用漏洞利用前是否可以获取进程的token提权信息。在一些案例中，是不能成功感染的。比如，在Windows XP系统中，v5的第二个发布版本可以允许，但是不能加密文件。

研究人员和Lemmou发现了Version
5.0.2中的问题，对注册表做一些修改就可以使恶意软件正常运行，但白帽怎么可以帮黑客修复恶意软件呢。但第二个版本使用的扩展是随机的5个字母，而不是之前版本中看

```
                dd 0583E900h, 45C700F0h
                db 0F4h
                dd offset aSoftwareEx_dat ; "SOFTWARE\\ex_data\\data"
; ---------------------------------------------------------------------------
                mov     dword ptr [ebp-14h], offset aExt ; "ext"
                and     dword ptr [ebp-4], 0
                push    0
                lea     eax, [ebp-8]
                push    eax
                push    0
                push    0F003Fh
                push    0
                push    0
                push    0
                push    dword ptr [ebp-0Ch]
                push    80000002h
                call    ds:RegCreateKeyExW
                mov     [ebp-4], eax
                cmp     dword ptr [ebp-4], 0
                jz      short loc_404956
                push    0
                lea     eax, [ebp-8]
                push    eax
                push    0
                push    0F003Fh
                push    0
                push    0
                push    0
                push    dword ptr [ebp-0Ch]
                push    80000001h
                call    ds:RegCreateKeyExW
                mov     [ebp-4], eax

loc_404956:                             ; CODE XREF: .text:00404930↑j
                cmp     dword ptr [ebp-4], 0
                jnz     short loc_404995
                push    dword ptr [ebp+8]
                call    ds:lstrlenW
                lea     eax, [eax+eax+2]
                push    eax
                push    dword ptr [ebp+8]
                push    3               ; BINARY TYPE
```

保存随机扩展名的新注册表记录

恶意软件会在HKEY_LOCAL_MACHINE的根key下创建新的记录。如果用户没有管理权限，就不能成功创建，然后将会将该记录放在HKEY_CURRENT_USER的根key中。文

V5.0.1

该版本修复了恶意软件中的一些内部bug，其他没有明显修改。

V5.0.2

该本比将随机扩展名的长度从5个字符修改为10个，并修复了一些内部bug。文件不能加密漏洞仍未修复。

最新版本

本节是对最新版本的GandCrab（10月4日的V5.0.2）恶意软件进行分析。从V5版本开始，恶意软件开始使用两个漏洞利用进行权限提升。

第一个漏洞利用对函数`IsWoW64Process`进行动态调用来检测操作系统运行的32位还是64位。

```
sub     esp, 28h
push    'k'
pop     eax
push    'e'
pop     edx
push    'r'
pop     ecx
push    'n'
mov     [ebp+ModuleName], ax
pop     eax
push    'l'
mov     [ebp+var_22], ax
pop     eax
push    '3'
mov     [ebp+var_1E], ax
pop     eax
mov     [ebp+var_1C], ax
push    '2'
pop     eax
mov     [ebp+var_1A], ax
xor     eax, eax
mov     [ebp+var_18], eax
mov     [ebp+var_6], al
lea     eax, [ebp+ProcName]
push    eax             ; lpProcName
lea     eax, [ebp+ModuleName]
mov     [ebp+var_26], dx
push    eax             ; lpModuleName
mov     [ebp+var_24], cx
mov     [ebp+var_20], dx
mov     dword ptr [ebp+ProcName], 'oWsI'
mov     [ebp+var_10], 'P46w'
mov     [ebp+var_C], cl
mov     [ebp+var_B], 'co'
mov     [ebp+var_9], dl
mov     [ebp+var_8], 'ss'
call    ds:GetModuleHandleW ; kernel32
push    eax             ; hModule
call    ds:GetProcAddress ; IsWoW64Process
```

对含有混淆字符串的IsWoW64Process的动态调用

根据结果，恶意软件有两个嵌入的DLL，用`XOR 0x18`进行简单加密操作。

```
                mov     dword ptr [ebp-0Ch], offset unk_42F970
                mov     dword ptr [ebp-10h], 1400h
                jmp     short loc_40B173
; ---------------------------------------------------------------------------

loc_40B165:                             ; CODE XREF: .text:0040B1E3↑j
                mov     dword ptr [ebp-0Ch], offset unk_42E770
                mov     dword ptr [ebp-10h], 1200h

loc_40B173:                             ; CODE XREF: .text:0040B163↑j
                and     dword ptr [ebp-1Ch], 0
                jmp     short loc_40B180
; ---------------------------------------------------------------------------

loc_40B179:                             ; CODE XREF: .text:0040B19C↓j
                mov     eax, [ebp-1Ch]
                inc     eax
                mov     [ebp-1Ch], eax

loc_40B180:                             ; CODE XREF: .text:0040B177↑j
                mov     eax, [ebp-1Ch]
                cmp     eax, [ebp-10h]
                jnb     short loc_40B19E
                mov     eax, [ebp-0Ch]
                add     eax, [ebp-1Ch]
                movsx   eax, byte ptr [eax]
                xor     eax, 18h
                mov     ecx, [ebp-0Ch]
                add     ecx, [ebp-1Ch]
                mov     [ecx], al
                jmp     short loc_40B179
; ---------------------------------------------------------------------------

loc_40B19E:                             ; CODE XREF: .text:0040B186↑j
                xor     eax, eax
                inc     eax
                imul    eax, 0
                mov     ecx, [ebp-0Ch]
                mov     byte ptr [ecx+eax], 'M'
                xor     eax, eax
                inc     eax
                shl     eax, 0
```

解密DLL来加载漏洞利用并修复header

恶意软件作者用fuzzing技巧来绕过检测：DLL的前2个字节是垃圾信息，在之后的版本中修复了。

解密和加载漏洞利用后，DLL会在系统中创建一个mutex和一些管道来与主恶意软件进行通信。恶意软件会创建一个DLL之后读取的管道，并准备一些字符串作为DLL的mute

```
                pop     edx
                mov     [ebp-10Ah], ax
                push    offset Name     ; "Global\\XlAKFoxSKGOfSGOoSFOOFNOLPE"
                push    14h
                lea     eax, [ebp-11Ch]
                push    eax
                mov     edx, [ebp-10h]
                mov     ecx, [ebp-0Ch]
                call    GandCrabPrepareStringFunctionForDLL
```

为DLL准备字符串

DLL的这些字符串有dummy string（虚拟字符串）。

```
push    ebp
mov     ebp, esp
sub     esp, 58h
push    esi
push    edi
xorps   xmm0, xmm0
mov     [ebp+StartupInfo.hStdError], 0
push    offset Name     ; "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"...
push    0               ; bInitialOwner
movdqu  xmmword ptr [ebp+StartupInfo.cb], xmm0
push    0               ; lpMutexAttributes
mov     esi, ecx
mov     [ebp+StartupInfo.cb], 44h
movdqu  xmmword ptr [ebp+StartupInfo.dwX], xmm0
movdqu  xmmword ptr [ebp+StartupInfo.dwXCountChars], xmm0
movdqu  xmmword ptr [ebp+StartupInfo.wShowWindow], xmm0
movdqu  xmmword ptr [ebp+ProcessInformation.hProcess], xmm0
call    ds:CreateMutexW
mov     edi, eax
lea     eax, [ebp+ProcessInformation]
push    eax             ; lpProcessInformation
lea     eax, [ebp+StartupInfo]
push    eax             ; lpStartupInfo
push    0               ; lpCurrentDirectory
push    0               ; lpEnvironment
push    0               ; dwCreationFlags
push    0               ; bInheritHandles
push    0               ; lpThreadAttributes
push    0               ; lpProcessAttributes
push    0               ; lpCommandLine
push    esi             ; lpApplicationName
call    ds:CreateProcessW
```

创建新的mutex并重启进程

恶意软件开始时会检查该mutex。函数返回的1或0却决于是否可以打开mutex。然后检查结果，如果mutex可以打开，恶意软件就不会检查版本并不会用这两个漏洞利用来

```
sub_4059EF      proc near               ; CODE XREF: .text:loc_405A57↓p
                push    offset Name     ; "Global\\XlAKFoxSKGOfSGOoSFOOFNOLPE"
                push    0               ; bInheritHandle
                push    100000h         ; dwDesiredAccess
                call    ds:OpenMutexW
                test    eax, eax
                jz      short loc_405A10
                push    eax             ; hObject
                call    ds:CloseHandle
                xor     eax, eax
                inc     eax
                retn
; ---------------------------------------------------------------------------

loc_405A10:                             ; CODE XREF: sub_4059EF+14↑j
                xor     eax, eax
                retn
sub_4059EF      endp
```

打开新mutex来检查是否有必要运行漏洞利用

GandCrab
V4.x版本之后，恶意软件会在之后检查版本。如果是Vista或之后版本，会尝试获取TokenIntegrityLevel类并重启二进制文件来提权，并以runas应用调用ShellExecu
XP，代码就会继续正常流。

并不会为主恶意软件创建mutex，mutex是为利用漏洞加载的DLL创建的。为了更好地理解，看一下下面的IDA片段：

```
_check_exploit_mutex:                   ; CODE XREF: .text:00405A4D↑j
                call    GandCrabCheckMutexOfExploits
                test    eax, eax
                jnz     short _no_exploits
                call    GandCrabGetOSVersionAndCheckWithVistaAtLeast
                test    eax, eax
                jz      short _no_exploits
                call    GandCrabGetTokenClassTokenImpersonationAndGetSidSubAuthority
                cmp     eax, 1000h
                ja      short _no_exploits
                call    GandCrabExploitCVE_2018_8120
                test    eax, eax
                jz      short _use_second_exploit
                call    GandCrabGetTokenClassTokenImpersonationAndGetSidSubAuthority
                cmp     eax, 1000h
                ja      short _no_exploits

_use_second_exploit:                    ; CODE XREF: .text:00405A7C↑j
                call    GandCrabExploitCVE_2018_8440
                test    eax, eax
                jz      short _no_exploits
                push    0
                call    ds:ExitProcess
; ---------------------------------------------------------------------------

_no_exploits:                           ; CODE XREF: .text:00405A5E↑j
                                        ; .text:00405A67↑j ...
                call    GandCrabGetOSVersionAndCheckWithVistaAtLeast
                test    eax, eax
                jz      short _prepare_normal_flow_to_check_language
                call    GandCrabGetTokenClassTokenImpersonationAndGetSidSubAuthority
                cmp     eax, 1000h
                ja      short _prepare_normal_flow_to_check_language
                call    GandCrabUseRunasProcessToLaunchItselfToElevatePrivileges
                push    0
                call    ds:ExitProcess
; ---------------------------------------------------------------------------

_prepare_normal_flow_to_check_language: ; CODE XREF: .text:00405AA2↑j
                                        ; .text:00405AAE↑j
                call    loc_40558B
```

解释mutex检查和漏洞利用

本版本还修改了桌面墙纸，是在运行时创建的，并用加密文件的扩展填充。勒索信文本或HTML的名为<extension_in_uppercase>_DECRYPT.<txt|html>和机器用户名。

```
push    0
call    ds:GetDC
mov     [ebp-8], eax
cmp     dword ptr [ebp-8], 0
jz      loc_40A2A7
push    dword ptr [ebp-8]
call    ds:CreateCompatibleDC
mov     [ebp-4], eax
cmp     dword ptr [ebp-4], 0
jz      loc_40A29C
push    8
push    dword ptr [ebp-8]
call    ds:GetDeviceCaps
mov     [ebp-14h], eax
push    0Ah
push    dword ptr [ebp-8]
call    ds:GetDeviceCaps
mov     [ebp-0Ch], eax
push    dword ptr [ebp-0Ch]
push    dword ptr [ebp-14h]
push    dword ptr [ebp-8]
call    ds:CreateCompatibleBitmap
mov     [ebp-34h], eax
cmp     dword ptr [ebp-34h], 0
jz      loc_40A293
push    dword ptr [ebp-34h]
push    dword ptr [ebp-4]
call    ds:SelectObject
mov     [ebp-134h], eax
push    5Ah
push    dword ptr [ebp-8]
call    ds:GetDeviceCaps
```

运行时创建新墙纸

检查用户名，如果用户是SYSTEM，恶意软件就在墙纸上显示USER。

```
loc_409BE9:                             ; CODE XREF: .text:00409BC3↑j
                mov     dword ptr [ebp-13Ch], 80h
                lea     eax, [ebp-13Ch]
                push    eax
                lea     eax, [ebp-340h]
                push    eax
                call    ds:GetUserNameW
                test    eax, eax
                jz      short loc_409C78
                push    offset aSystem  ; "SYSTEM"
                lea     eax, [ebp-340h]
                push    eax
                call    ds:lstrcmpiW
                test    eax, eax
                jz      short loc_409C3F
                lea     eax, [ebp-340h]
                push    eax
                push    offset aDearS   ; "DEAR %s, "
                lea     eax, [ebp-0B40h]
                push    eax
                call    ds:wsprintfW
                add     esp, 0Ch
                jmp     short loc_409C53
```

检查墙纸的用户名

在文件夹 %TEMP%中创建名为pidor.bmp的墙纸：

```
push    4
push    3000h
push    200h
push    0
call    ds:VirtualAlloc
mov     ecx, [ebp+8]
mov     [ecx], eax
mov     eax, [ebp+8]
cmp     dword ptr [eax], 0
jz      short loc_40A260
mov     dword ptr [ebp-128h], 1
mov     eax, [ebp+8]
push    dword ptr [eax]
push    100h
call    ds:GetTempPathW
push    offset aPidor_bmp ; "\\pidor.bmp"
mov     eax, [ebp+8]
push    dword ptr [eax]
call    ds:lstrcatW
mov     eax, [ebp+8]
push    dword ptr [eax]
mov     edx, [ebp-8]
mov     ecx, [ebp-34h]
call    GandCrabGetBitsFromBMPOfMemoryAndWriteInFileInDisk
pop     ecx
```

在temp文件夹创建墙纸

这是墙纸名所用的字符串，检查用户名和格式的字符串：

```
                db    0
aSystem:                                ; DATA XREF: .text:00409C0B↑o
                unicode 0, <SYSTEM>,0
                align 4
aDearS:                                 ; DATA XREF: .text:00409C28↑o
                unicode 0, <DEAR %s, >,0
aDearUser:                              ; DATA XREF: .text:loc_409C3F↑o
                unicode 0, <DEAR USER, >,0
aPidor_bmp:                             ; DATA XREF: .text:0040A23F↑o
                unicode 0, <\pidor.bmp>,0
```

墙纸名和特殊字符串

最后，对所有用户设置墙纸：

```
call    GandCrabGetTokenClassTokenImpersonationAndGetSidSubAuthority
cmp     eax, 4000h      ; SECURITY_MANDATORY_SYSTEM_RID
jnb     short _check_thread_handle
and     dword ptr [ebp-8], 0
push    630h
lea     eax, [ebp-8]
push    eax
call    GandCrabCreateWallpaperAndWriteInDisk
pop     ecx
pop     ecx
test    eax, eax
jz      short _check_thread_handle
push    3
push    dword ptr [ebp-8]
push    0
push    14h
call    ds:SystemParametersInfoW
```

修改墙纸

恶意软件会检查系统语言，解密字符串，并根据系统语言写出对应的勒索信。

安全防护

独立研究员Twitter用户Valthek创建了一些非常有效的免疫工具，可以预防GandCrab 4.x 到5.0.2不被加密。

免疫工具：https://29wspy.ru/reversing.html

对Version 4.x版本，删除影子卷是不可避免的，但至少文件本身是安全的。

对Version
5.x版本，加密文件是可以避免的，但创建和修改墙纸无法避免。恶意软件不能创建随机扩展来加密文件，但是这些随机字符串已经准备好的。如果墙纸在%TEMP%文件夹中，

免疫工具有不同的版本，有的有驻留机制，有的没有驻留机制。有驻留机制的工具在每次运行时，会特殊的文件夹中创建一个随机的文件名并将特殊的随机记录写入注册表中

上一篇：CanSecWest2017 Pw... 下一篇：2018护网杯线上赛 Writeu...
1. 0 条回复
   - 动动手指，沙发就是你的了！