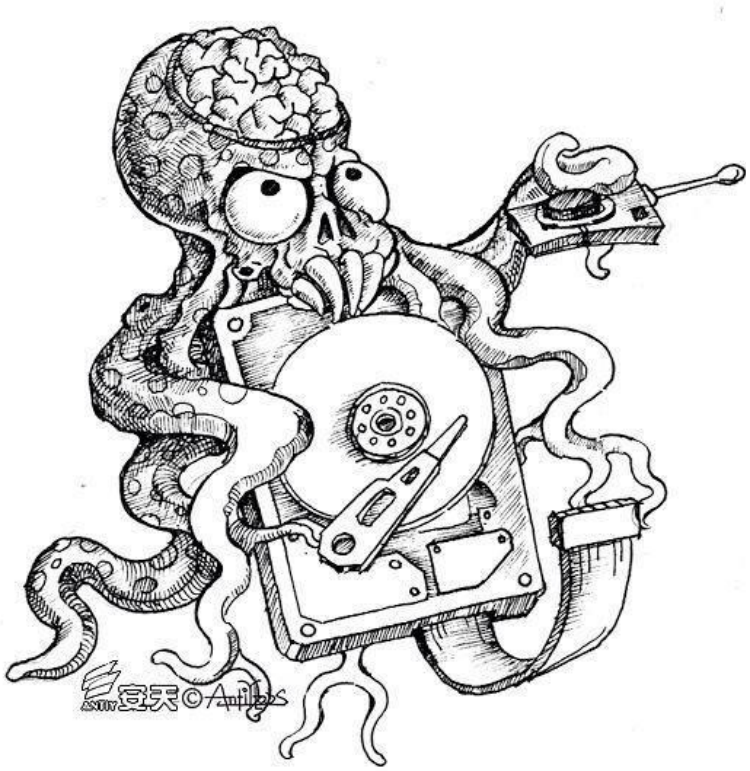
修改硬盘固件的木马

——探索方程式（EQUATION）组织的攻击组件

安天实验室



首次发布时间：2015 年 03 月 05 日 10 时 00 分

本版本更新时间：2015 年 03 月 05 日 09 时 45 分

目 录

1 背景

2015 年 2 月 18 日，安天实验室根据紧急研判，对被友商称为“方程式（Equation）”的攻击组织所使用的攻击组件，开始了初步的分析验证。后于 2 月 25 日正式组建了跨部门联合分析小组，于 3 月 4 日形成本报告第一版本。

事件相关背景为：卡巴斯基安全实验室在 2 月 16 日起发布系列报告（以下简称“友商报告”），披露了一个可能是目前世界上存在的最复杂的网络攻击组织——“方程式”组织（Equation Group）[1]。据卡巴斯基实验室称，该组织使用的 C&C 早在 1996 年就被注册，这暗示了该组织可能已经活跃了 20 年之久。多年以来，他们因总能比其他组织早发现漏洞，从而具有绝对的优势。该组织拥有一套用于植入恶意代码的超级制式信息武器库（在友商报告中披露了其中 6 个），其中包括两个可以对数十种常见品牌的硬盘固件重编程的恶意模块，这可能是该组织掌握的最具特色的攻击武器，同时也是首个已知的能够感染硬盘固件的恶意代码。在 2 月 17 日和 2 月 19 日的友商报告中，先后发布了其中 2 个模块的详细分析结果，它们分别是 Fanny[2]和 DoubleFantasy[3]。卡巴斯基根据相关线索分析，认为被攻击目标包括俄罗斯、印度、中国等国家，而相关媒体根据卡巴斯基的报告，推断出该攻击组织可能与美国情报机构相关。

鉴于样本的复杂性，以及攻击硬盘固件的特殊特点，分析进展极为缓慢，目前将有限的分析工作对外分享，旨在推动更多的业内参与协作。同时对友商报告中已经充分论述的内容，本报告未作更多引用和重复。因此建议读者先阅读友商报告，再阅读本报告以给予批评指正。

2 Equation 组织使用的组件

Equation 组织的被发现的武器库中至少有 6 件“装备”，它们是：EquationLaser、EquationDrug、DoubleFantasy、TripleFantasy、Fanny 和 GrayFish。安天的工程师称其为“组件”。除了这 6 个组件外，友商报告还提供了该组织有可能用到的其它恶意代码程序的哈希，这些哈希对应的程序包括：与 EquationDrug 相似的 EQUESTRE、键盘记录器程序 GROK keylogger、DoubleFantasy 安装程序和 LNK 漏洞利用程序 _SD_IP_CF.dll，以及需要重点关注的能够对硬盘重新编程的模块 nls_933w.dll。

组件名称	说明	时间
EquationLaser	Equation 组织早期使用的植入程序，大约在 2001 至 2004 年间被使用。兼容 Windows 95/98 系统。	2001-2003
EquationDrug	该组织使用的一个非常复杂的攻击组件，用于支持能够被攻击	2003-2013

	者动态上传和卸载的模块插件系统。怀疑是 EquationLaser 的升级版。	
DoubleFantasy	一个验证式的木马，旨在确定目标为预期目标。如果目标被确认，那么已植入恶意代码会升级到一个更为复杂的平台，如 EQUATIONDRUG 或 GRAYFISH。	2004-2012
TripleFantasy	全功能的后门程序，有时用于配合 GRAYFISH 使用。看起来像是 DOUBLEFANTASY 的升级版，可能是更新的验证式插件。	2012-至今
Fanny	创建于 2008 年的利用 USB 设备进行传播的蠕虫，可攻击物理隔离网络并回传收集到的信息。Fanny 被用于收集位于中东和亚洲的目标的信息。一些受害主机似乎已被升级到 DoubleFantasy，然后又升级为 EQUATIONDRUG。Fanny 利用了两个后来被应用到 Stuxnet 中的 0day 漏洞。	2008-2011
GrayFish	Equation 组织中最复杂的攻击组件，完全驻留在注册表中，依靠 bootkit 在操作系统启动时执行。	2008-至今

Equation 组织的 6 个组件的攻击示意图：

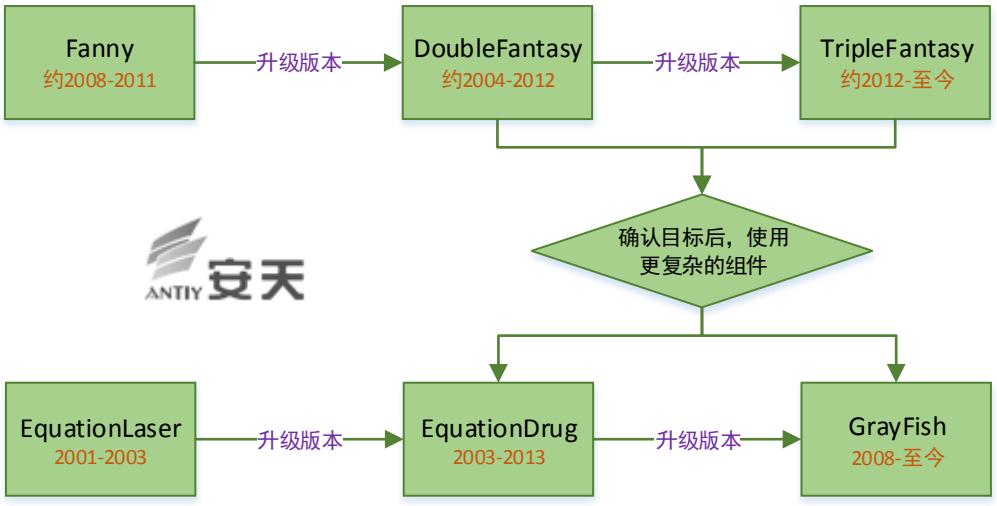


图 1 组件关系示意图

Equation 组织攻击时，选择 Fanny 或 DoubleFantasy 或 TripleFantasy 作为攻击前导，当确认被攻击端是攻击者的预期目标后，会使用更复杂的组件 EquationDrug 或 GrayFish。

安天分析小组目前将重点放在攻击前导组件（DoubleFantasy）、更复杂的组件（EquationDrug 和 GrayFish）。同时对具有硬盘固件重新编程功能的 nls_933w.dll 进行分析。

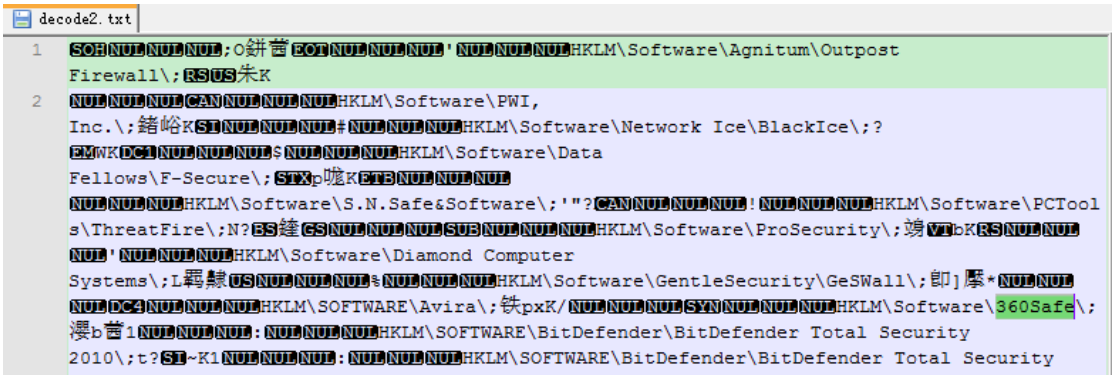
3 组件 DoubleFantasy 分析

组件 DoubleFantasy 是用来确认被攻击目标的，如果被攻击的目标属于被 Equation 组织感兴趣或关注的领域，那么更加复杂的其他组件就会从远端注入到被攻击的机器中。

友商报告已经对组件 DoubleFantasy 进行了详细分析，安天分析小组原本计划对组件 DoubleFantasy 进行分析验证，但在验证的过程中，分析小组发现该组件是以往分析过的，并找到了其他的关联的恶意代码；同时，安天也发现了友商报告未见披露的信息。

3.1 检测安全软件

组件 DoubleFantasy 枚举注册表键值，查找系统是否安装了安全软件，查询的安全软件列表存在资源节中，使用 0x79 异或加密。在友商报告中给出了其检测是否存在的安全软件列表，共计 10 种，而安天分析小组发现，实际上该组件一共检测 13 种安全软件的存在，除友商报告披露的 10 种产品外，还有 360、BitDefender 和 Avira 三家厂商的产品。



鉴于其中 360 安全卫士主要用户均在中国，这也进一步验证了中国也是 Equation 组织攻击的目标之一。

3.2 回传信息

DoubleFantasy 收集系统信息，并回传给攻击者，回传格式为：

000:MAC 地址 001:IP 地址.....019:当前时间

回传的详细信息如下：

标号	说明	标号	说明	标号	说明
000	MAC 地址	007	系统补丁信息（CSDVersion，例如 sp1）	014	网络连接类型
001	IP 地址	008	CurrentBuildNumber （例如	015	安装的软件信息

			2600）		
002	样本版本号	009	系统 CurrentVersion（5.1）	016	未知
003	样本 id	010	ProductID	017	此值不存在
004	代理设置信息	011	位置信息 1	018	32 位或 64 位
005	注册信息 1 (RegisteredOwner)	012	位置信息 2	019	当前时间
006	注册信息 2 (RegisteredOrganization)	013	系统目录		

3.3 通讯协议

DoubleFantasy 的被控端返回包格式是第一字节不加密，后面的数据加密。举例 0x42 指令如下：

0x42 指令分支详细功能

- 功能：重新上线，初始化通讯密钥，删除自身，清理感染痕迹。
- 控制端发包格式：第一字节为指令代码 0x42，第二字节为指令分支，分别有 3 种：00 立即重新上线，01 初始化通讯密钥，Sleep 60 秒后重新上线，02 删除自身，清除感染痕迹。
- 被控端返回包格式：无。

3.4 新的版本、C&C、密钥

友商报告给出了相关组件的版本、C&C 列表和密钥，经安天进一步分析，获得了更多相关信息。下文中绿色为友商报告中信息，红色（加粗）为安天分析出的新的信息。

版本列表：

8.1.0.4 (MSREGSTR.EXE)
008.002.000.006
008.002.001.001
008.002.001.004
008.002.001.04A (subversion "IMIL3.4.0-IMB1.8.0")
008.002.002.000
008.002.003.000
008.002.004.000
008.002.005.000
008.002.005.001
008.002.006.000
011.000.001.001
012.001.000.000
012.001.001.000

012.002.000.001
012.003.001.000
012.003.004.000
012.003.004.001
013.000.000.000

C&C 如下：

advancing-technology[.]com
avidnewssource[.]com
businessdealsblog[.]com
businessedgeadvance[.]com
charging-technology[.]com
computertechanalysis[.]com
config.getmyip[.]com - SINKHOLED BY KASPERSKY LAB
globalnetworkanalys[.]com
melding-technology[.]com
myhousetechnews[.]com - SINKHOLED BY KASPERSKY LAB
newsterminalvelocity[.]com - SINKHOLED BY KASPERSKY LAB
selective-business[.]com
slayinglance[.]com
successful-marketing-now[.]com - SINKHOLED BY KASPERSKY LAB
taking-technology[.]com
techasiamusicsvr[.]com - SINKHOLED BY KASPERSKY LAB
technicaldigitalreporting[.]com
timelywebsitehostesses[.]com
www.dt1blog[.]com
www.forboringbusinesses[.]com
Ignlist.com***
Datcemgmt.net***
Imptoday.com***
Budnessnews.com***

新的密钥：

37 08 EF 89 29 A7 4B 6B AB 3E 5D 03 F6 B0 B5 B3
66 39 71 3C 0F 85 99 81 20 19 35 43 FE 9A 84 11
8B 4C 25 04 56 85 C9 75 06 33 C0 5E C2 08 31 F6
32 EC 89 D8 0A 78 47 22 BD 58 2B A9 7F 12 AB 0C

组件 DoubleFantasy 通常是受害者被 Equation 组织感染的第一步，通过与后门的通信以及对不同系统参数的检查来确认受害者的信息。受害者一旦被确认，Equation 组织将使用更复杂的组件 EquationDrug 或 Grayfish。

4 组件 EquationDrug 分析

组件 EquationDrug 是一个很复杂的模块。其存活时间有近 10 年，后来被 GrayFish 升级给代替了。安天在分析中发现两个模块中的一些文件名称有相同处，从混淆加密等方面来看也有多处手法相同。它们都是从资源解密、解压缩和释放文件。分析中发现在资源里有一个 SYS 和一个 VXD 文件。VXD 是 Windows 9x 下的驱动机制，所在可以认定这个模块也有感染 Windows 9x 下的能力。其 EquationDrug 是一个插件平台，它具有安装与卸载插件功能。

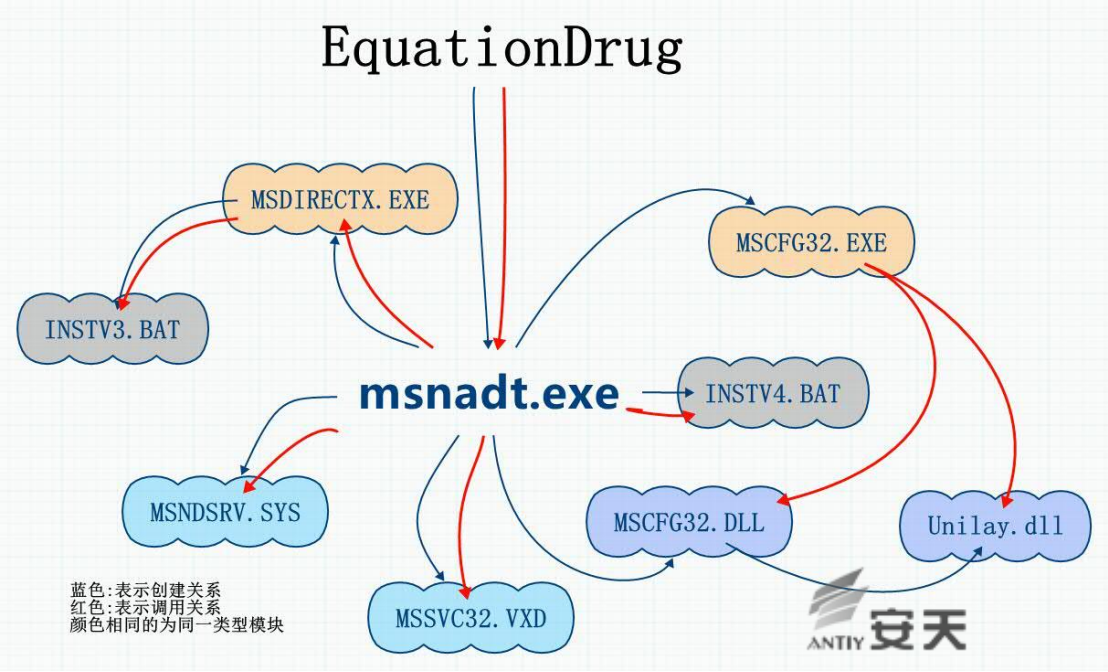


图 2 组件 EquationDrug 的创建与调用关系图

模块名称	功能
msnadt.exe	文件功能主要为释放文件、解密资源、判断系统类型、注入代码到指定进程和加载驱动等功能。
MSDIRECTX.EXE	创建 INSTV3.BAT 并运行自删除。
MSCFG32.exe	加载 MSCFG32.DLL，添加和修改注册表。
MSCFG32.DLL	该文件会添加和修改注册表，释放 unity.dll 文件。与驱动文件有关系，含有网络功能。
unity.dll	有大量文件操作和注册表操作

MSNDSRV.SYS MSSVC32.VXD	功能基本相同，但 VXD 是在 windows9.x 下用的，主要功能是 hook、网络监听和写文件等。还会判断系统中是否有 MSlog32.dat，有就打开写入数据，没有就创建一个新的。
INSTV3.BAT INSTV4.BAT	自删除文件。

4.1 检测安全软件

其枚举注册表键值，查找系统是否安装了安全软件，查询的安全软件列表存在资源节中。

其所检测的安全软件比 DoubleFantasy 组件更多，且类型也更加丰富，但同时其检测的中国安全软件是瑞星（Rising），但并未检测当前更为流行的 360。因此也可以验证前文关于这个组件已经被更新组件替代的结论。相关检测的注册表键值如下：

Zone Labs\TrueVector
Zone Labs\ZoneAlarm
KasperskyLab
Network Ice\BlackIce
Agnitum\Outpost Firewall
Sygate Technologies, Inc.\Sygate Personal Firewall
Norman
Data Fellows\F-Secure
PWI, Inc.
rising
Softwin
network associates\tvd\shared components\on access scanner\behaviourblocking\FileBlockEnabled_27!=0
network associates\tvd\shared components\on access scanner\behaviourblocking\FileBlockEnabled_28!=0
network associates\tvd\shared components\on access scanner\behaviourblocking\FileBlockEnabled_29!=0
network associates\tvd\shared components\on access scanner\behaviourblocking\FileBlockEnabled_30!=0
McAfee\ePolicy Orchestrator\Application Plugins\VIRUSCAN8600
Sophos
CA\CAPF
CA\HIPSEngine
Cisco
Symantec\IDS
Symantec\Norton 360
Symantec\Internet Security\SuiteOwnerGuid
Symantec\Norton AntiBot
Symantec\Symantec Endpoint Protection
Tiny Software\Tiny Firewall
CyberMedia Inc\Guard Dog
McAfee\Guard Dog

McAfee\McAfee Firewall
McAfee\Personal Firewall
McAfee.com\Personal Firewall
Network Associates\McAfee Fire
Kerio
BullGuard Ltd.\BullGuard
TheGreenBow
Panda Software\Firewall
TrendMicro\PC-cillin
ComputerAssociates\eTrust Suite Personal\pfw
Grisoft\Firewall

4.2 驱动模块 MSNDSRV.SYS 分析

1. 驱动初始化的时候从注册表中遍历了所有的网卡，然后调用函数 NdisRegisterProtocol 向 NDIS 库注册了一个 NDIS 协议相关的结构。注册后该驱动就可以收到本机上的所有的网络流量，这一点类似于 WinPcap 的捕包机理。相关代码如下：

```
lea     eax, [ebp+Status]
push    [ebp+NdisProtocolHandle] ; NdisProtocolHandle
mov     [ebp+ProtocolCharacteristics.OpenAdapterCompleteHandler], offset sub_B20779AA
mov     [ebp+ProtocolCharacteristics.CloseAdapterCompleteHandler], offset sub_B20779C2
mov     dword ptr [ebp+ProtocolCharacteristics.anonymous_1], offset sub_B207A170
push    eax             ; Status
mov     dword ptr [ebp+ProtocolCharacteristics.anonymous_2], offset sub_B2078970
mov     [ebp+ProtocolCharacteristics.ResetCompleteHandler], offset nullsub_2
mov     [ebp+ProtocolCharacteristics.RequestCompleteHandler], offset sub_B20776BA
mov     dword ptr [ebp+ProtocolCharacteristics.anonymous_3], offset sub_B20789E4
mov     [ebp+ProtocolCharacteristics.ReceiveCompleteHandler], offset nullsub_1
mov     [ebp+ProtocolCharacteristics.StatusHandler], offset sub_B2077728
mov     [ebp+ProtocolCharacteristics.StatusCompleteHandler], offset nullsub_1
mov     [ebp+var_38], offset sub_B2078B5A
mov     [ebp+var_34], offset sub_B20777B8
mov     [ebp+var_30], offset sub_B2077940
mov     [ebp+var_2C], offset sub_B2077AB8
call    ds:NdisRegisterProtocol
mov     eax, [ebp+Status]
pop     edi
```

2. 修改了 KeServiceDescriptorTable 中的函数的地址。

```
80528bac cc              int     3
kd> dd  KeServiceDescriptorTable
80553fa0  80502b8c 00000000 0000011c 80503000
80553fb0  00000000 00000000 00000000 00000000
80553fc0  00000000 00000000 00000000 00000000
80553fd0  00000000 00000000 00000000 00000000
80553fe0  00002710 bf80c0b6 00000000 00000000
80553ff0  f8b77a80 f830eb60 820fe550 806e2f40
80554000  00000000 00000000 cf09f27c 00000003
80554010  9776c53c 01d0562c 00000000 00000000
kd> dd  80502b8c
80502b8c  8059a948 805e7db6 805eb5fc 805e7de8
80502b9c  805eb636 805e7e1e 805eb67a 805eb6be
80502bac  8060cdfe 8060db50 805e31b4 805e2e0c
80502bbc  805cbde6 805cbd96 8060d424 805ac5ae
80502bcc  8060ca3c 8059edbe 805a6a00 805cd8c4
80502bdc  80500828 8060db42 8056ccd6 8053600e
80502bec  806060d4 805b2c3a 805ebb36 8061ae56
80502bfc  805f0028 8059b036 8061b0aa 8059a8e8
```

图 3 原始的 KeServiceDescriptorTable 中的函数地址

```
kd> dd  80502b8c
80502b8c  820742b1 820742bd 820742c9 820742d5
80502b9c  820742e1 820742ed 820742f9 82074305
80502bac  82074311 8207431d 82074329 82074335
80502bbc  82074341 8207434d 82074359 82074365
80502bcc  82074371 8207437d 82074389 82074395
80502bdc  820743a1 820743ad 820743b9 820743c5
80502bec  820743d1 820743dd 820743e9 820743f5
80502bfc  82074401 8207440d 82074419 82074425
kd> u  820742b1
```

图 4 修改后的 KeServiceDescriptorTable 中的函数地址

修改后的函数地址只是包含一个 JMP 指令。 如果 KeServiceDescriptorTable 中的函数不是它要 hook 的目标则直接跳回原始函数的地址，否则跳到驱动自己的函数中去。

如函数 nt!NtAcceptConnectPort 在 KeServiceDescriptorTable 中它的地址是 820742b1。

该处的指令如下：

```
820742b1 2eff25b8420782    jmp        dword ptr cs:[820742B8h]
```

820742B8 就是 NtAcceptConnectPort 对应的地址。而函数 NtTerminateProcess 在 KeServiceDescriptorTable 中的地址是 0x81cf9ebd.该处的指令为：

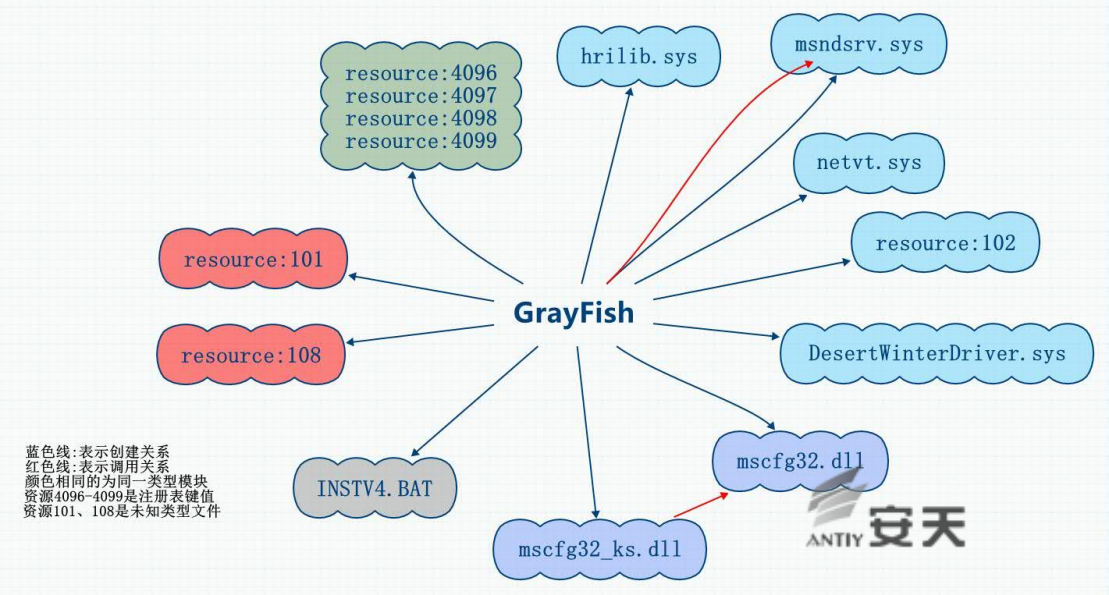
```
81cf9ebd 2eff25c49ecf81    jmp        dword ptr cs:[81CF9EC4h]
```

81CF9EC4 中包含的地址为 b1fd6eae，该地址指向驱动的一个函数。目前驱动 hook 的函数如下：

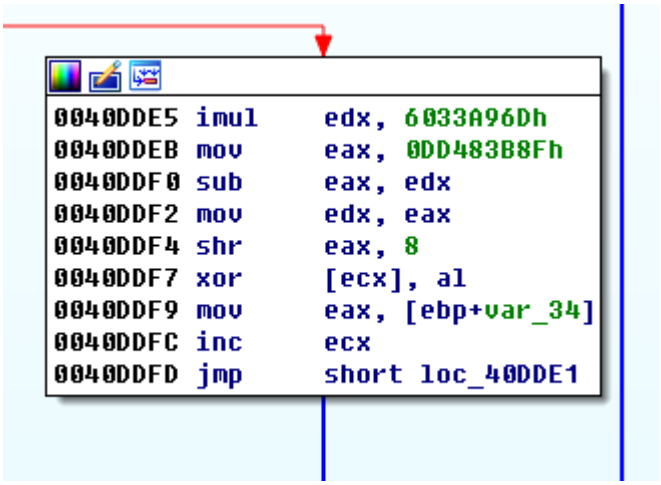
NtClose
NtCreateFile
NtCreateKey
NtCreateProcess
NtCreateProcessEx
NtCreateThread
NtEnumerateKey
NtOpenFile
NtOpenKey
NtOpenProcess
NtQueryAttributesFile
NtQueryDirectoryFile
NtQueryDirectoryObject
NtQueryFullAttributesFile
NtQueryKey
NtQuerySystemInformation
NtSetInformationFile
NtTerminateProcess

5 组件 GrayFish 分析

GrayFish 是 Equation 组织中最复杂的攻击组件，是 EquationDrug 的新一代版本，安天分析小组认为其最重要的特点是：**不依靠文件载体，而是完整的存在于注册表中，依靠 bootkit 在操作系统启动时执行，这一机制穿透了安全产品以文件为检测对象的机制，也穿透了相关基于白名单和可信计算的解决方案。**



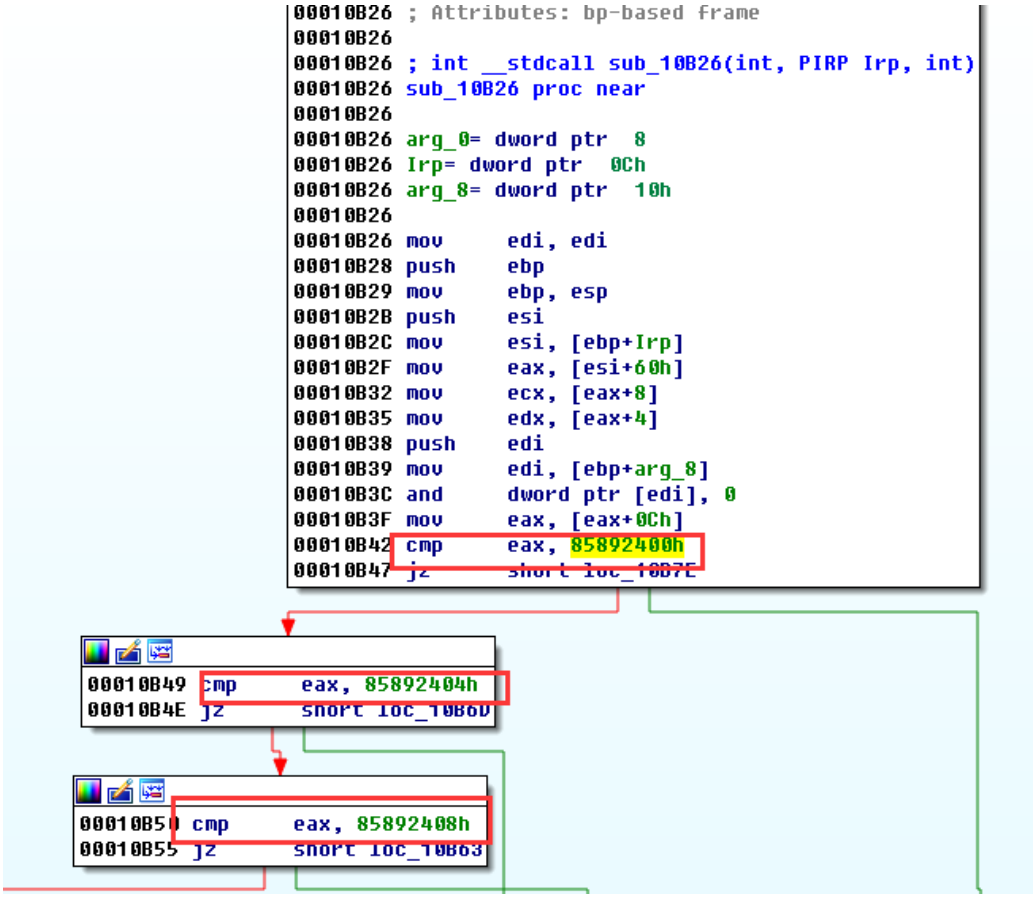
组件 GrayFish 的资源段中包含 13 个加密资源，均通过同一段解密算法进行解密：



解密后的 13 个文件中有 5 个驱动文件（sys）、2 个动态链接库文件（dll）、4 个包含注册表数据的文件、1 个含有字符串“services.exe”的配置文件及 1 个加密的数据文件。

动态调试后发现，其中 3 个驱动文件 hrilib.sys、msndsrv.sys 及 netvt.sys 是由原始样本释放，包含网络驱动及注册表相关操作函数。mscfg32_ks.dll 调用 mscfg32.dll，拥有创建远程线程、获取系统信息、创建和

删除注册表键值等功能。除已释放的三个驱动文件外，资源 102 中含有对注册表进行操作的函数，而 DesertWinterDriver.sys 中包含对 IoControlCode 的比较，具体功能有待分析。



另外，原样本会生成批处理用以删除自身，该批处理文件名与 EquationDrug 用以自删除的文件名完全相同，这也说明二者之间具有密切的联系。

6 硬盘固件重新编程模块 nls_933w.dll 分析

nls_933w.dll 是具有硬盘固件重新编程能力的模块，由于硬盘固件是一个安天分析小组之前缺少储备的领域，因此分析进展非常缓慢。从目前分析来看，当 nls_933w.dll 模块被其他程序调用后，nls_933w.dll 从自身资源释放 win32m.sys 驱动文件，win32m.sys 驱动文件负责与硬盘控制器进行通信，它能够判断硬盘控制器类型，如：IDE、SATA 等，根据不同类型的硬盘控制器发送对应的控制指令。因此只要攻击者熟悉各硬盘厂商规定的 ATA 指令，那么就可以对硬盘固件进行恶意篡改。

图 5 修改硬盘固件的流程图

动态调试后安天分析小组发现，该模块调用函数 DeviceIoControl 与 win32m.sys 进行交互。在 win32m.sys 中，安天分析小组发现多个 IoControlCode 并分析了它们所对应的功能。

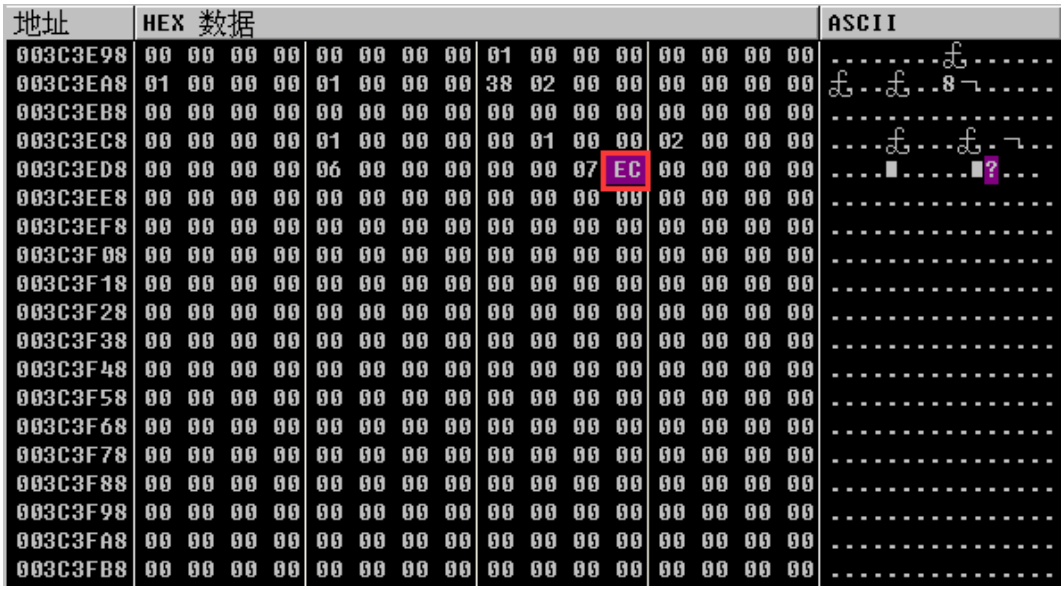
```
  goto LABEL_8;
}
CTL_CODE = *((_DWORD *)pCurrentStackLocation + 3);// Enter deviceIoControl
if ( CTL_CODE == 0x870021C0 )                   // 读版本号
{
  if ( pBuffer && OutBufferSize >= 8 )
  {
    Irp->IoStatus.Information = 8;
    qmemcpy(pBuffer, decode((unsigned __int8 *)&temp, (unsigned __int8 *)&v3_0_0_0), 8u);// 3.0.0.0
    goto LABEL_10;
  }
  ret = 0xC0000023;
}
else
{
  if ( CTL_CODE == 0x870021C4 )                 // 初始化硬盘控制器
  {
    if ( InBufferSize == 854 )
    {
      if ( *((_DWORD *)pDeviceExtension + 22) )
        goto LABEL_10;
      if ( Wait_TakeMutex(&Mutex) )
      {
        v12 = InitFunctionList_HookIRQL((int)pBuffer, (int)pDeviceExtension, 854);
        ReleaseMutex(&Mutex);
        v2 = Irp;
        if ( v12 )
          goto LABEL_10;
      }
      sub_103B2((int)pDeviceExtension);
      goto LABEL_18;
    }
    goto LABEL_22;
  }
  if ( CTL_CODE == 0x870021C8 )
  {
    sub_103B2((int)pDeviceObject->DeviceExtension);
    goto LABEL_10;
  }
  if ( CTL_CODE == 0x870021CC )                 // 检查C4初始化后驱动的状态
  {
    if ( !*((_DWORD *)pDeviceExtension + 24) )
      goto LABEL_20;
```

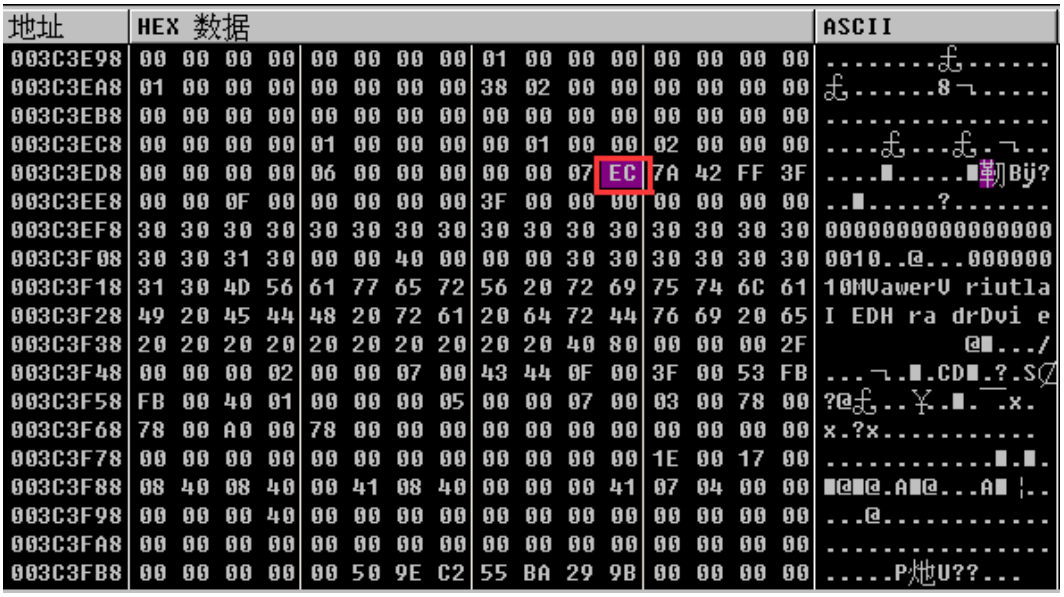
图 6 IoControlCode 所对应的功能图

安天分析小组发现，当 IoControlCode 为 0x870021D0 时，nls_933w.dll 对硬盘控制器发送 ATA 控制指令：0xEC，获取硬盘相关信息。

图 7 获取硬盘相关信息

调用 DeviceIoControl 前后内存中的数据对比，调用后返回硬盘信息：





其他的 IoControlCode 所对应的功能及 ATA 指令有待进一步分析和发现。

7 攻击硬盘固件的机理分析

7.1 硬盘的结构和工作原理

不论是传统的机械硬盘还是固态硬盘，其总体的结构都是相似的。硬盘主要由处理器、缓存、Boot ROM 和主存储介质等几部分构成，对于机械硬盘，还有电机驱动电路和磁头控制电路等。其简化原理框图如下图所示：

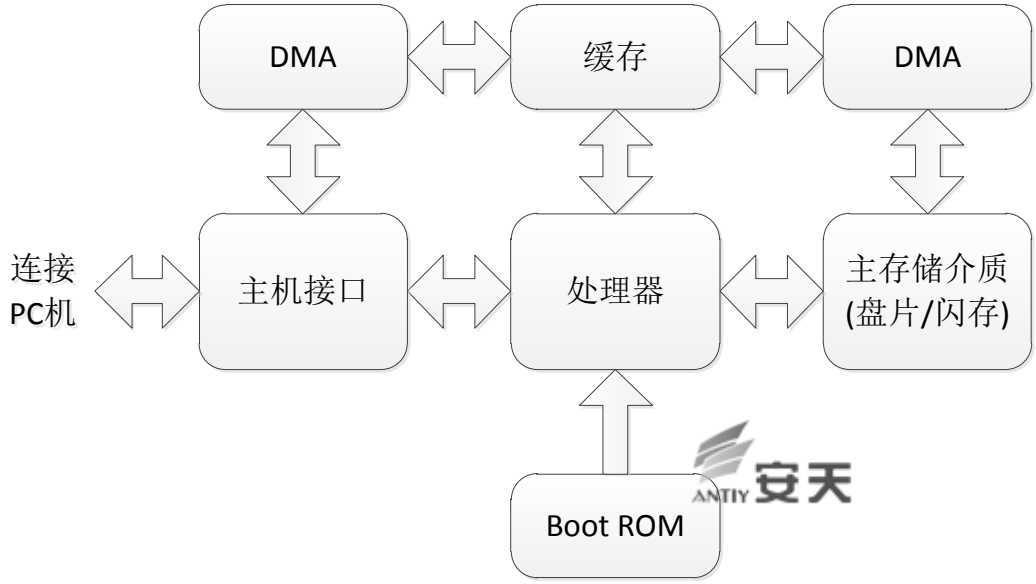


图 8 硬盘原理框图

由于硬盘的电路板上已经具有了 CPU、内存和 ROM，硬盘可以看做是一个小型的计算机系统，在固件的控制下可以有自己的行为。目前常见的硬盘处理器都是基于 ARM 核心的，新型的硬盘控制器甚至采用多核结构来保证高速的数据传输。

硬盘通电时，处理器执行片内的 Loader 代码，这部分代码会加载 Boot ROM 到缓存中，并执行（对硬盘上的嵌入式处理器来说，就是内存）。Boot ROM 可能存放在主控的片内 FLASH，独立的 I2C EEPROM，SPI FLASH 芯片或者固态硬盘上的 NAND FLASH 阵列中。Boot ROM 得到控制权之后，会依次初始化基本外设，初始化主存储介质，从主存储介质上加载固件主体，启动 IDE/SATA 总线接口驱动模块，并进入待命状态，此时计算机即可对硬盘进行操作。

1) 传统机械硬盘

对于目前大部分的机械硬盘来说，其固件的主体部分通常存放在盘片上的隐藏扇区中，Boot ROM 按照校准数据初始化磁头组件之后，从隐藏扇区中读取固件数据，并将控制权转交给固件主体，固件主体完成自身初始化之后，加载并启动总线接口驱动模块。至此，硬盘完成上电启动过程。

机械硬盘的内部结构如图所示：

图 9 机械硬盘组成与结构

（http://jingyan.baidu.com/article/ab0b5630d88efdc15bfa7d60.html）

硬盘的数据存储在磁盘盘片上，硬盘工作时，主轴带动盘片高速旋转，读写磁头悬浮于盘片上方几微米处，通过巨磁阻效应来进行读写操作。传动手臂通过强磁铁与线圈构成的音圈电机进行寻道，以定位要读写的内容。图中的返利局弹簧装置是给传动手臂提供回复力的，该装置能够保证硬盘在断电的时候，磁头能够自动归位到 Park 区。Park 区有一块柔软的支撑垫，可以在硬盘不工作时固定读写手臂，以免因外界震动而划伤盘片。

2) 固态硬盘

与机械硬盘相比，由于没有机械结构，固态硬盘的结构要简单不少，通常的固态硬盘都可以用下面的图来描述：

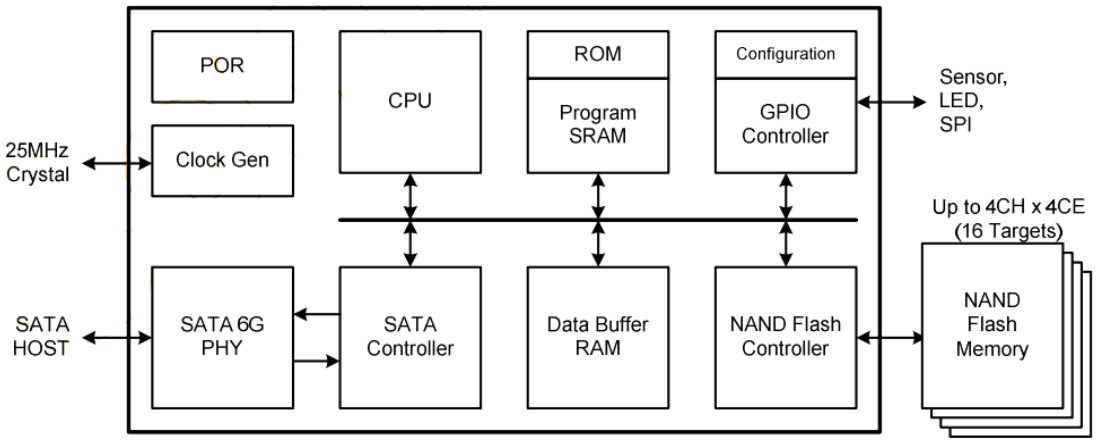


图 10 固态盘及其控制器结构框图

（《JMF608SATA III NAND Flash Controller datasheet》）

图中左边的框是固态硬盘的控制器，右边是板载外设和 NAND FLASH 阵列，部分型号的控制器还需要外置的 Data Buffer RAM，也就是缓存。从图中可以看出，控制器本身就可以构成一个完整的计算机系统，其引导过程与机械硬盘类似，在此就不再重复了。

3) 硬盘的接口规范

目前常见的 IDE 和 SATA 硬盘都遵循 ATA 指令集，PC 机通过发送 ATA 命令来对硬盘进行读写操作。

ATA 技术是一个关于 IDE（Integrated Device Electronics）的技术规范族。最初，IDE 只是一项以把控制器与盘体集成在一起为主要意图的硬盘接口技术。 随着 IDE/EIDE 得到的日益广泛的应用，全球标准化协议将该接口自诞生以来使用的技术规范归纳成为全球硬盘标准，这样就产生了 ATA（Advanced Technology Attachment）。ATA 发展至今经过多次修改和升级，每新一代的接口都建立在前一代标准之上，并保持着向后兼容性。除了读写命令以外，硬盘还支持一些高级功能，比如自我监测功能（SMART），容量设置（HPA）、噪音管理（AAS）等。详见《ATA/ATAPI Command Set - 2 (ACS-2)》（一部 500 多页的大部头标准文档！）。

7.2 硬盘的信息安全脆弱性

特别要值得注意的是，目前大部分硬盘都支持固件升级功能（通过下载微码命令或者厂商的私有命令实现），用户可以通过厂商规定的 ATA 指令来对硬盘驱动器上的固件进行更新。这使得硬盘厂商无需召回有固件 bug 的产品，而可以在用户系统上通过软件工具升级固件，修补缺陷。例如，希捷 2008.12 月的硬盘有故障，官方发布了固件更新工具和使用说明，使用户刷新固件解决问题。类似的还有硬盘厂商西部数据的 C1 门事件。

下图以 Seagate SandForce SF-2200 系列固态硬盘为例，说明硬盘固件的升级过程：

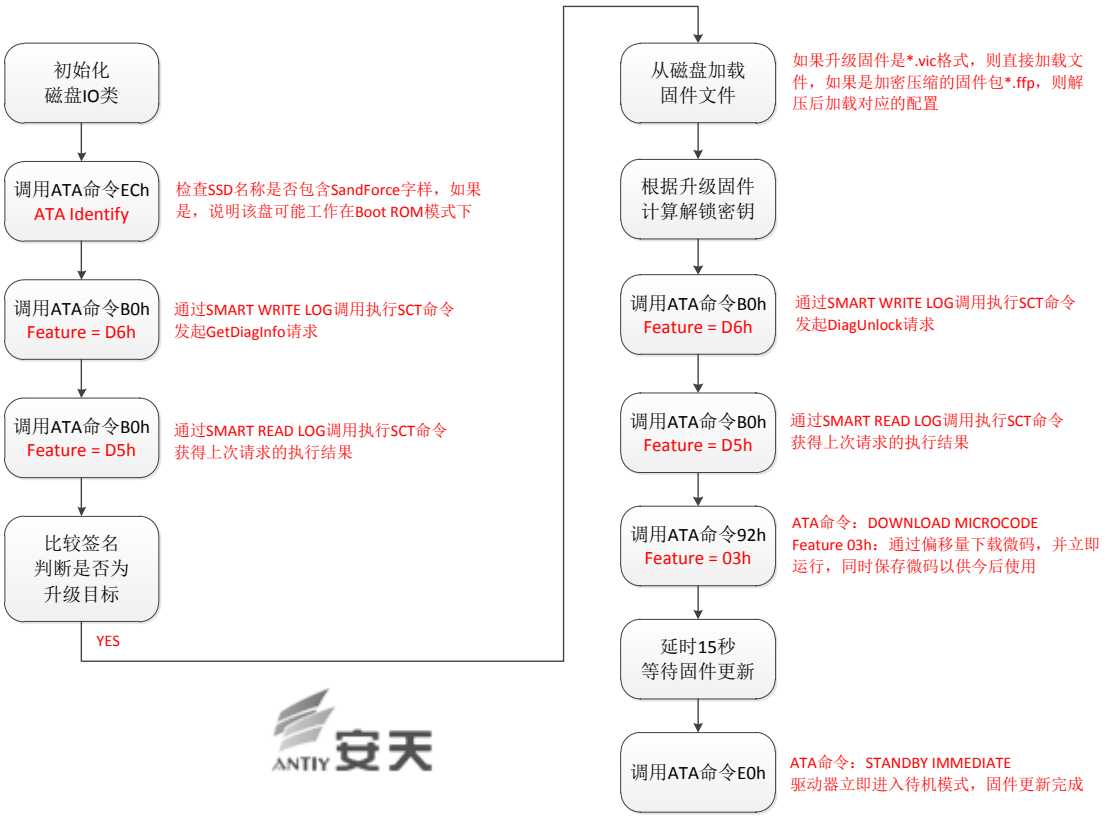


图 11 Seagate SandForce SF-2200 系列固态硬盘固件升级流程图

这种通过主机软件在系统（In System）升级固件的机制使用很方便，但也意味着存在着固件被恶意篡改的可能性。而且这种篡改可以是通过软件操作，在用户毫不知情的情况下进行。

如前文所述，硬盘本身就是一套完整的嵌入式系统，其内部的固件独立于计算机软硬件而运行。固件完全决定了硬盘的读写操作行为，甚至可以在主机不知情的时候自主处理数据。如果攻击者在硬盘的固件中设计了精巧的代码，则可以对用户的读写操作进行拦截和干扰，或者通过这种手段获得系统的最高控制权，而所有这一切都是在硬盘上完成的，计算机前面的用户、计算机上的软硬件根本无法感知这一过程，甚至知道也无法干预这样的动作发生。

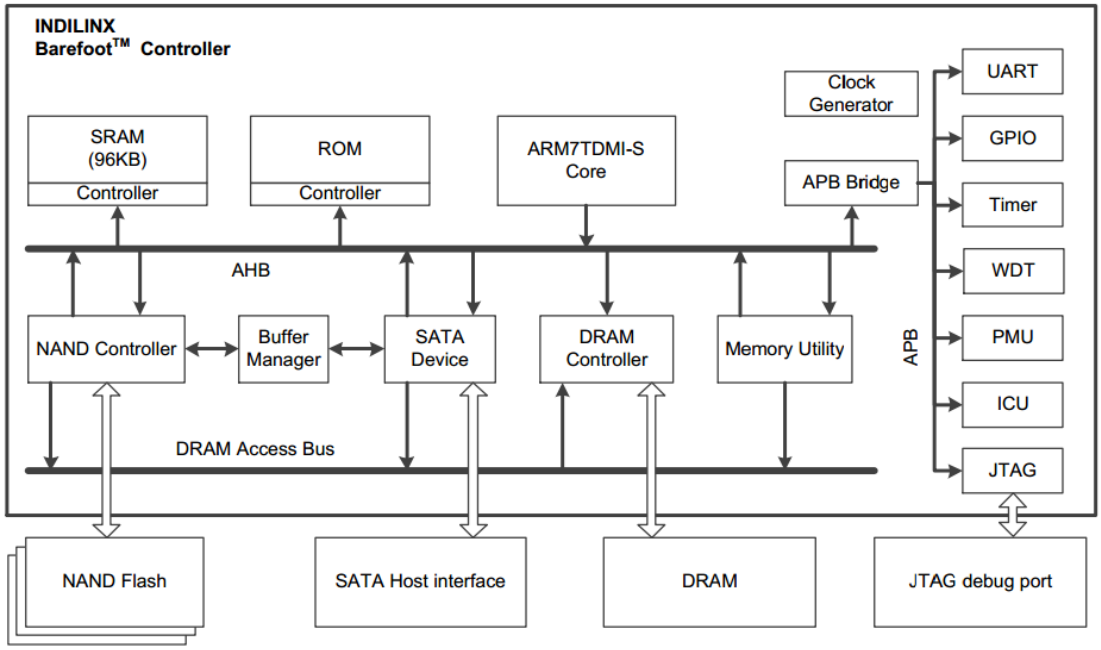


图 12 Jasmine 开发板结构框图

（http://www.openssd-project.org）

以 OpenSSD 项目（一个以研究性质的开源硬盘项目）中的 Jasmine 开发板为例，如果攻击者对某款硬盘的控制器的结构非常了解，包括片上的外设地址空间等信息。则改攻击者可以通过精心构造的修改版固件来对控制器的某些特定的行为进行拦截，并在数据传输途中对缓存 DRAM 中的数据进行修改。比如拦截 ATA 读取命令 20h，在读取指定扇区时篡改缓存中的数据，使计算机实际获得的数据与磁盘上存储的内容不符，进而实现自下而上的攻击效果。这种攻击方式对来自硬盘的数据流的改写完全绕过了计算机系统，因此可以在重装系统甚至低格硬盘之后仍然保持其危害性。

8 小结

时间总是让人觉得似曾相识，2 月 16 日，农历腊月二十八，还有两天就是春节了，关于方程式攻击组织的相关信息浮出水面。这让我们想起 2003 年农历小年的 Slammer 蠕虫、2004 五一节的震荡波蠕虫、以及去年国庆节前的“破壳”漏洞。但不同的是，此前事件冲击的是我们的应急速度，而方程式考验的是我们综合储备与能力深度，以及分析耐心。

对安天的分析团队来说，这是第一次在发布一篇分析报告时如此惴惴不安。在 2003 年发布 Dvldr（口令蠕虫）分析报告时，我们是那样急切，希望用户更快看到我们提供的解决方案；在发布震网分析报告时，我们是那样盲目，自以为我们的工作已经基本足够；在发布火焰系列组件分析报告时，我们是那样放任——既然庞大到分析不过来，就接力式的分析-发布好了。但这一次，完全不同，因为我们一度被卡住

了，不因加密、驱动、隐藏，而是因为“硬盘固件”，对于那些经过长期准备而施于一点的攻击来说，防御者搞清问题的关键往往去取决于愿意付出多少人力与时间。

我们自以为是敏锐的，我们很早就在关注嵌入式与固件，我们大讲威胁的泛化，但当威胁真正出现于面前之时，我们才发现对手更加先验和强大，而我们的所谓敏锐何其幼稚。

而同样令我们忧心忡忡的是，相关事件的报告正在不断的走形。很多用户向我们求证和询问“是不是所有硬盘都已经放入了后门木马”。

因此尽管我们的分析工作还在持续之中，我们依然要凭借经验给出下列结论或判断：

1. 硬件设备的固件可更新机制，是软硬件系统发展的必然结果，这种机制本身不能被称之为后门。同时对更多带有固件系统来说，如果没有更新机制，那么将导致有问题的版本不能得到补丁，即可能带来更大的运维成本，也反而可能是一个重要的安全隐患。
2. 综合安天和友商以及其他机构目前分析结果来推测，相关攻击中写入固件的行为发生于前导恶意代码回传主机信息，并被远端判定为有价值目标的情况下，即其并非一个普遍行为，而是一个高等级、有条件的入侵行为。
3. 通过长期的分析摸索，攻击者完全可以独立实现相关机理，并不一定需要入侵硬盘厂商获取技术文献，甚至靠硬盘厂商主动提供技术文献。
4. 其写入硬盘固件，关键是用于实现潜伏与长期存在，但上层的作业能力依然存在于主机系统中，而且可以通过网络灵活获取其他作业模块。
5. 鉴于相关国家此前的行为，我们也有理由怀疑，同样的组件，可能被用于物流链劫持，即在特定目标采购、返修主机或硬盘的过程中注入。但基于其作业手法和风险分析，我们有理由认为，对方程式这样的对手来说，这种 Bootkit+固件的作业手法，通常不会进行批量作业。
6. 但后续我们同样要警惕的是 Bruce Schneier 所警告的 “越来越多战争中的战术行为被应用于更广泛的网络空间环境中”，而且新的手法一经曝光就会对黑产产生强烈的启迪效应，从而使威胁泛滥。
7. 相关攻击确实体现了相关供应链的安全盲点，对于硬盘固件是否有有效的签名验证机制，而同时已经写入硬盘的固件，目前来看我们没有找到看到低成本的无条件读取的接口。类似的设计就给安全分析人员进行检测验证带来了困难。

防御阵地的规划，不能依赖于臆想对手。客观看待安全与发展的关系，深入具体的分析威胁，研判对手的策略和路径，永远是我们应对威胁的支点。

附录一：参考资料

[1] Equation: The Death Star of Malware Galaxy

http://securelist.com/blog/research/68750/equation-the-death-star-of-malware-galaxy/

[2] A Fanny Equation: "I am your father, Stuxnet"

http://securelist.com/blog/research/68787/a-fanny-equation-i-am-your-father-stuxnet/

[3] Equation Group: from Houston with love

http://securelist.com/blog/research/68877/equation-group-from-houston-with-love/

附录二：事件日志

更新日期	更新内容
2015-02-18	启动事件研判、验证
2015-02-21	安天 CERT 与微嵌入联合分析组成立
2015-02-25	启动全面分析
2015-03-02	开始编写初步分析报告
2015-03-04	形成分析报告第一版本
2015-03-05	安天 CERT 进行内容校对并版本更新到 V1.3

附录三：关于安天

安天是专业的下一代安全检测引擎研发企业，安天的检测引擎为网络安全产品和移动设备提供病毒和各种恶意代码的检测能力，并被超过十家以上的著名安全厂商所采用，全球有数万台防火墙和数千万部手机的安全软件内置有安天的引擎。安天获得了 2013 年度 AV-TEST 年度移动设备最佳保护奖。依托引擎、沙箱和后台体系的能力，安天进一步为行业企业提供有自身特色的基于流量的反 APT 解决方案。

关于反病毒引擎更多信息请访问：http://www.antiy.com（中文）

http://www.antiy.net（英文）

关于安天反 APT 相关产品更多信息请访问：http://www.antiy.cn