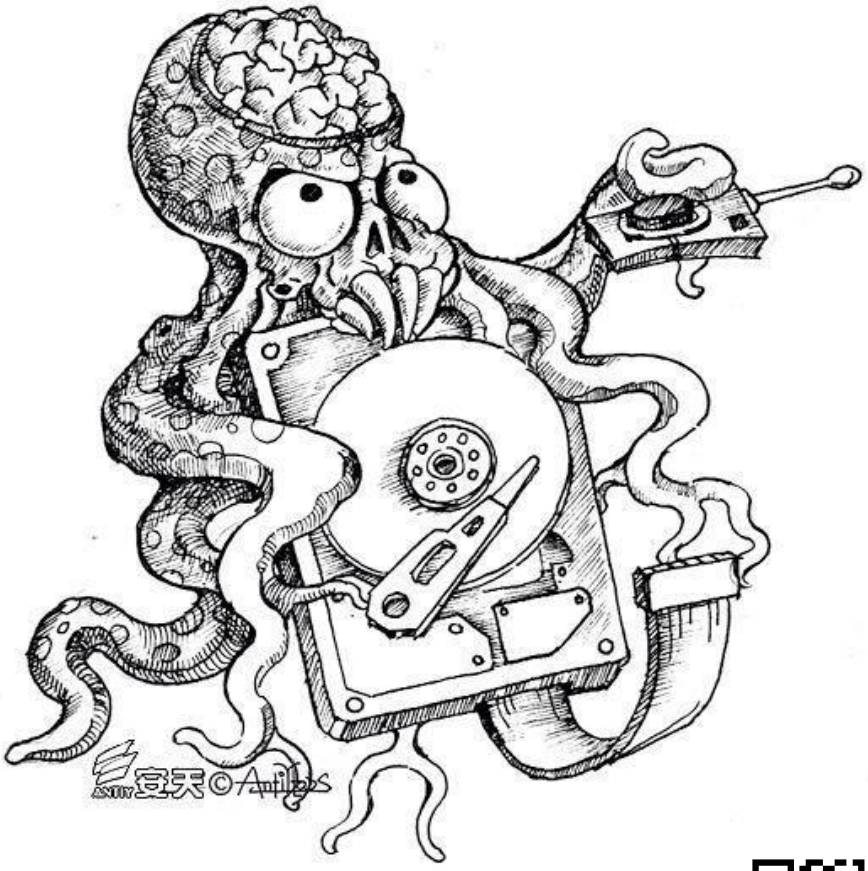
从“方程式”到“方程组”

EQUATION 攻击组织高级恶意代码的全平台能力解析

安天安全研究与应急处理中心（Antiy CERT）



初稿完成时间：2014 年 1 月 15 日 16 时 43 分

首次发布时间：2016 年 10 月 4 日 10 时 00 分

本版更新时间：2016 年 10 月 4 日 13 时 00 分

目 录

1 背景

安天从 2015 年 2 月起，陆续公布了两篇针对方程式攻击组织的分析报告，分析了其针对 Windows 平台的恶意代码组件构成、对硬盘的持久化能力和对加密算法的使用。本报告则将首次公布安天对方程式攻击组织针对 Solaris 平台和 Linux 平台的部分样本分析，我们也可以自豪的说，这是业内首次正式证实这些“恶灵”真实存在的公开分析。事实上，安天的相关工作完成于数年前。安天的分析工程师们从 2012 年起，已经关注到超级攻击组织，力图将其载荷能力覆盖一切可以达成入侵和持久化的场景，在这些场景中，各种服务器操作系统，如 Linux、Solaris、FreeBSD 等是其高度关心的目标。这些载荷不是寻常的脚本木马，而是**组件化、具备 Rootkit 能力、具有超强加密抗分析能力、严格进行加密通讯的二进制组件**。在安天工程师一直将类似超级攻击组织发起的攻击称为 A^2PT，并把恶意代码载荷的全平台覆盖能力作为 A^2PT 组织的重要标志。

安天将长期跟踪分析高级威胁和高级恶意代码的经验转化为产品能力，为用户探海威胁检测系统协助用户在网络中捕获载荷投放与横向移动，利用智甲终端防御系统为传统 Windows 主机和国产操作系统提供全面的保护，协助用户使用追影安全分析平台进行多种平台的恶意代码分析。这些产品的部署也使安天能够在用户支持下获取更多的威胁线索。同时安天也积极关注开源情报和公开信息，关注相关组织的有关信息与动向。

在去年年初卡巴斯基和安天先后对方程式组织使用的恶意代码进行分析曝光后，方程式组织又在一系列“爆料”事件中浮出水面。在 2016 年 8 月所外泄的方程式组织针对多种防火墙和网络设备的攻击代码中[1]，公众第一次把方程式组织和名为“ANT”的攻击装备体系联系起来，并以此看到其针对 Cisco、Juniper、Fortinet 等防火墙产品达成注入和持久化的能力。而在 2016 年 10 月 31 日，The Hacker News 发布文章“Shadow Brokers reveals list of Servers Hacked by the NSA”[2]，文章披露了“影子经纪人”公开的更多文件，其中包括部分方程式组织入侵的外国服务器列表。相关文件声称，大部分被感染的服务器运行的是 Solaris，Oracle-owned Unix 等版本的操作系统，有些运行的是 FreeBSD 或 Linux 系统。而随着这些信息和安天的捕获分析工作相互印证，一个关于这个超级攻击组织的**几乎无死角的、全平台化攻击能力**已经日趋清晰。

我们的分析工作不断验证着这些信息，在过去数年，这种分析如此漫长、复杂和艰难，超出了我们之前对“震网”、“火焰”的分析和复现中所面临的挑战。这种高度复杂、隐蔽的全能高级恶意代码，无论是对受害者，还是分析者来说，都是一个巨大的挑战。特别是当其打击范围几乎覆盖所有体系结构与操作系统的时候，相对更擅长 Windows、Linux 和 Android 等主流操作系统平台下恶意代码分析的传统安全分析团

队感受到了巨大的压力和挑战。如果用这个组织的名称“方程式”做一个关于分析难度的比喻的话，**我们需要破解的已经并不只是一个“方程式”，而是更为复杂的多元多次的“方程组”**。

2 方程式组织的多平台作业能力

方程式组织采用了**工业水准的制式化攻击武器库**，安天在此前报告中已经对其 6 件恶意代码组件“装备”进行了分析，他们分别是：EquationLaser、EquationDrug、DoubleFantasy、TripleFantasy、Fanny 和 GrayFish，其中 EquationDrug、DoubleFantasy 安天均已发现其他平台的样本。方程式武器库信息见下表：

组件名称	多平台特性	组件说明	使用时间
EquationLaser	尚未发现	方程式组织早期使用的植入程序，大约在 2001 至 2003 年间被使用。兼容 Windows 95/98 系统。	2001-2003
EquationDrug	**部分插件已经发现**	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者动态上传和卸载的模块插件系统。怀疑是 EquationLaser 的升级版。	2003-2013
DoubleFantasy	**已经证实**	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	**推测存在**	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	尚未发现	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投放 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	尚未发现	方程式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

读者可以通过阅读下列报告，自己完成了方程式攻击组织针对多平台操作系统的拼图：

信息	Windows	Linux	Solaris	Oracle-owned Unix	FreeBSD	Mac OS
安天：修改硬盘固件的木马 探索方程式（EQUATION）组织的攻击组件[3]	分析样本载荷和硬盘持久化能力					

安天：方程式（EQUATION）部分组件中的加密技巧分析[4]	分析加密算法					
安天：EQUATION 攻击组织的全平台载荷能力解析（本报告）		曝光存在，分析相关载荷	分析相关载荷			
The Hacker News：《Shadow Brokers reveals list of Servers Hacked by the NSA》			曝光存在	曝光存在	曝光存在	
卡巴斯基：Equation: The Death Star of Malware Galaxy[5]	揭秘方程式攻击组织					
卡巴斯基：A Fanny Equation: "I am your father, Stuxnet"[6]	Fanny 组件分析					
卡巴斯基：Equation Group: from Houston with love[[7]	Doublefantasy 分析					
卡巴斯基：《EQUATION GROUP: QUESTIONS AND ANSWERS》[8]	方程式组织问与答					根据网络特征提出猜测

注：安天在 Solaris 样本中分析出的 User Agent 具有 Solaris 标识，而卡巴斯基在“EQUATION GROUP: QUESTIONS AND ANSWERS”[8]中披露出曾捕获到 Mac OS X 的 User Agent 的信息，由此来看，尽管安天和卡巴斯基厂商目前都尚未捕获 Mac OS X 的样本，但方程式组织针对 MAC OS X 的攻击载荷是真实存在的。

3 X86 Linux 部分载荷分析

安天已经捕获分析了 Linux 下的 DoubleFantasy 组件。该组件是方程式组织在 Linux 平台上用于前期侦查、探测预期目标的攻击样本。由于是 Linux 平台下的样本，在具体功能实现的技术细节上与我们之前的曝光的 Windows 样本有所区别。

3.1 侦查、探测的前导模块——DoubleFantasy

3.1.1 文件标签

病毒名称	Trojan/Linux.DoubleFantasy
原始文件名	[redacted]

MD5	■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
处理器架构	X86（32 位平台）
文件大小	■■■■■■■■■
文件格式	BinExecute/ELF
时间戳	N/A
数字签名	无
加壳类型	无
编译语言	C 语言

3.1.2 运行流程

Trojan/Linux.DoubleFantasy 样本执行时分为有参数和无参数两种情况，当有参数‘-c’时候，仅仅是用于获取系统信息，可以视为场景侦查功能，其流程如下：

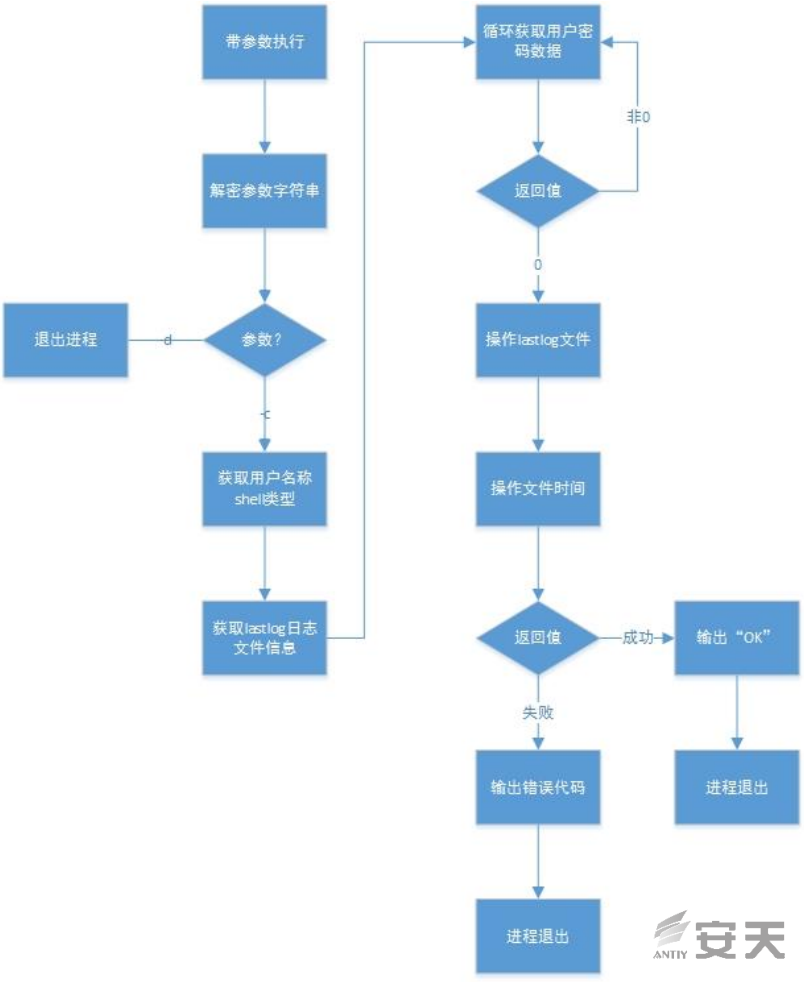


图 1 Trojan/Linux.DoubleFantasy－c 参数流程

如果样本以无参数运行会具有网络通信行为，流程如下：

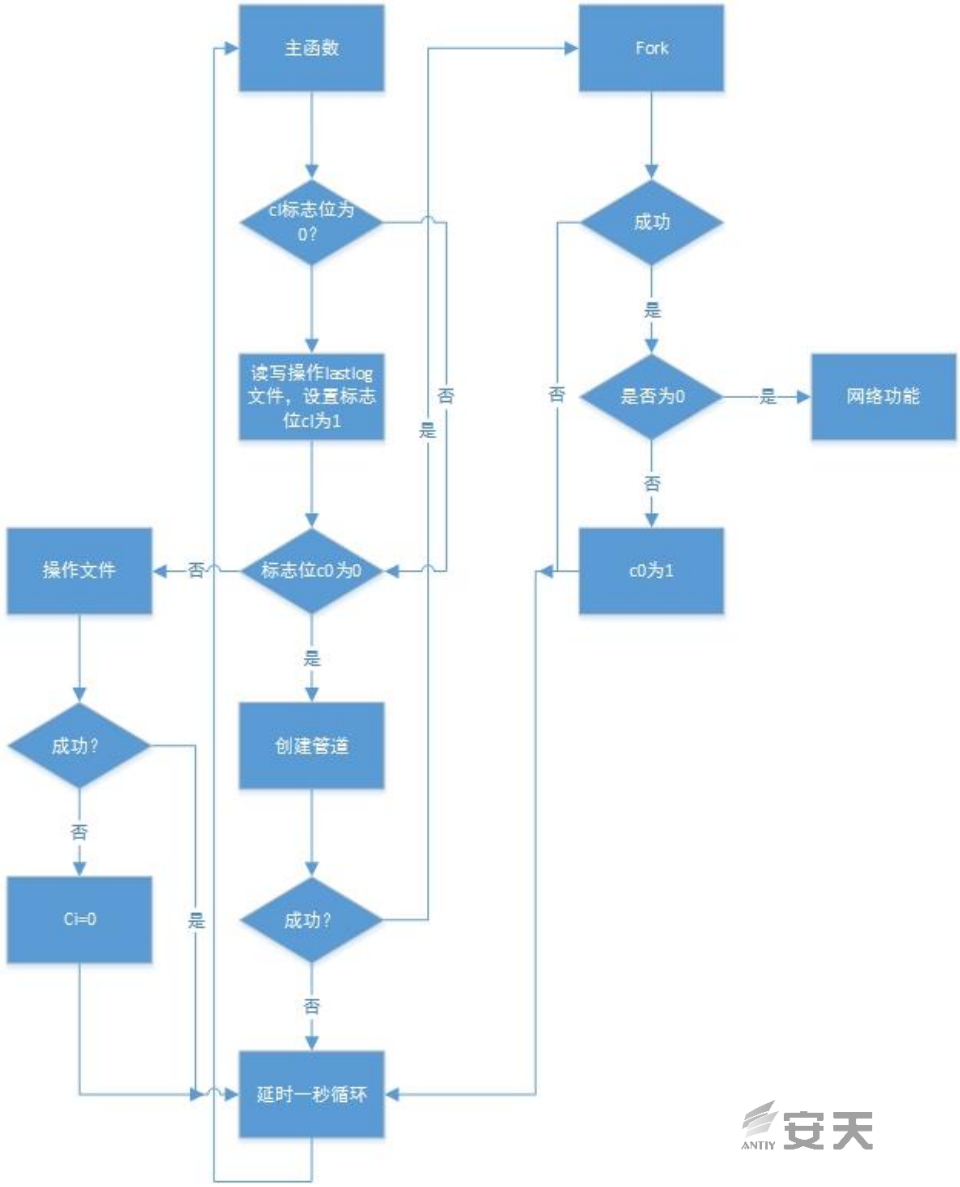


图 2 Trojan/Linux.DoubleFantasy 无参数运行流程

3.1.3 基本功能

- 遍历系统文件、清除/var/log/lastlog 记录、获取系统账户密码信息。
- 连接 Google 判断网络连通状态。
- 连接远程服务器，并根据远程控制指令进行不同的操作。
- 样本中同样存在多处信息加密算法和网络通讯加密算法。
- 样本会利用一个链接文件启动自身，proc/%d/exe 文件指向样本自身文件。
- 样本运行后会开启三个连续的 PID 线程。

- 随后样本收集被感染机器的信息包括系统目录、文件扩展名等信息。如下图：

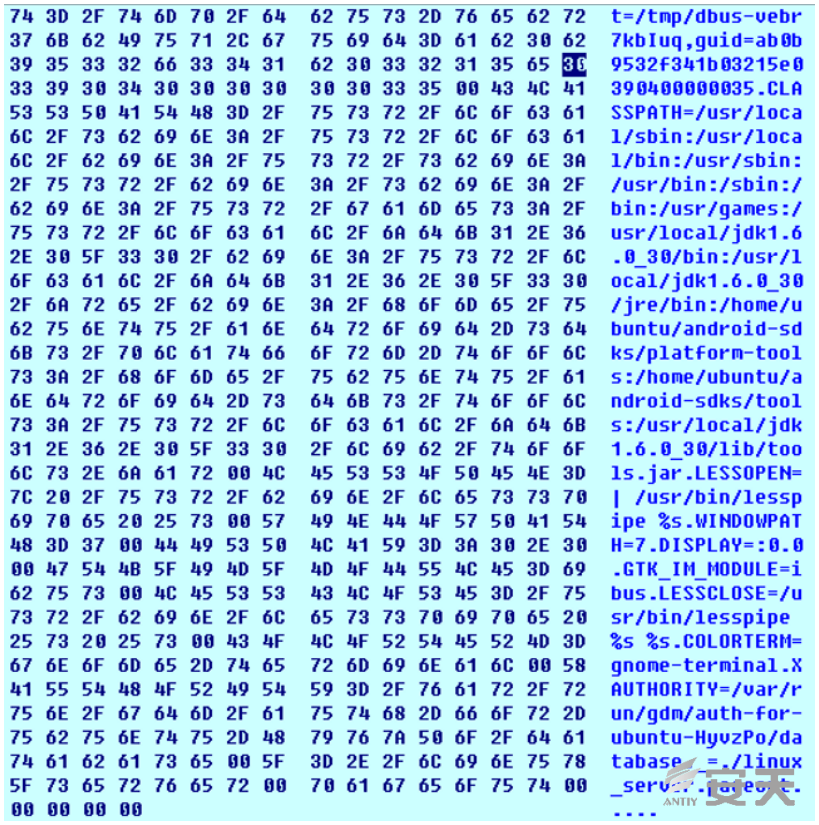


图 3 收集常规系统信息

- 恶意代码开始 fork( )进程，并判断 fork( )的子进程的 PID 号，判断是否执行成功，如果执行成功则主进程退出，无法调试，影响调试过程如下图所示：

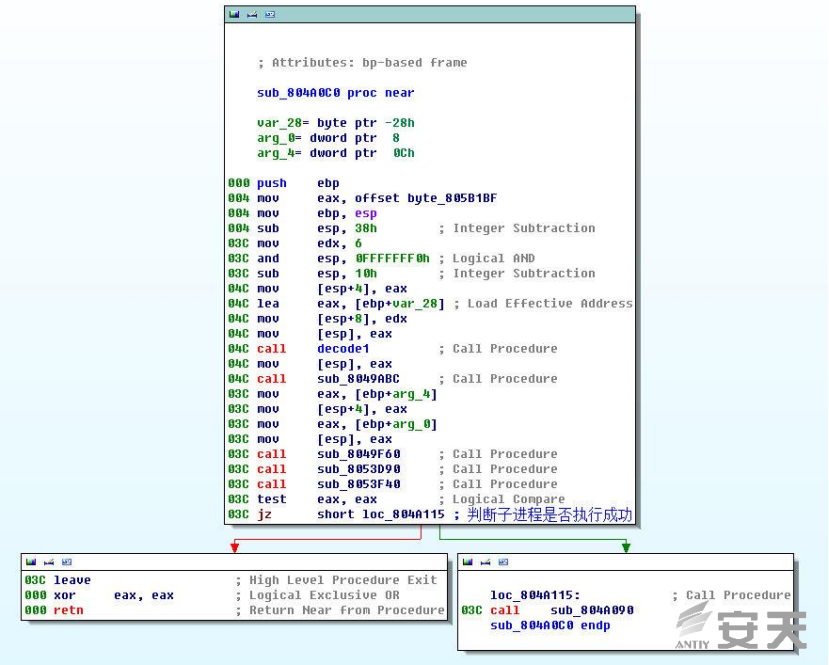
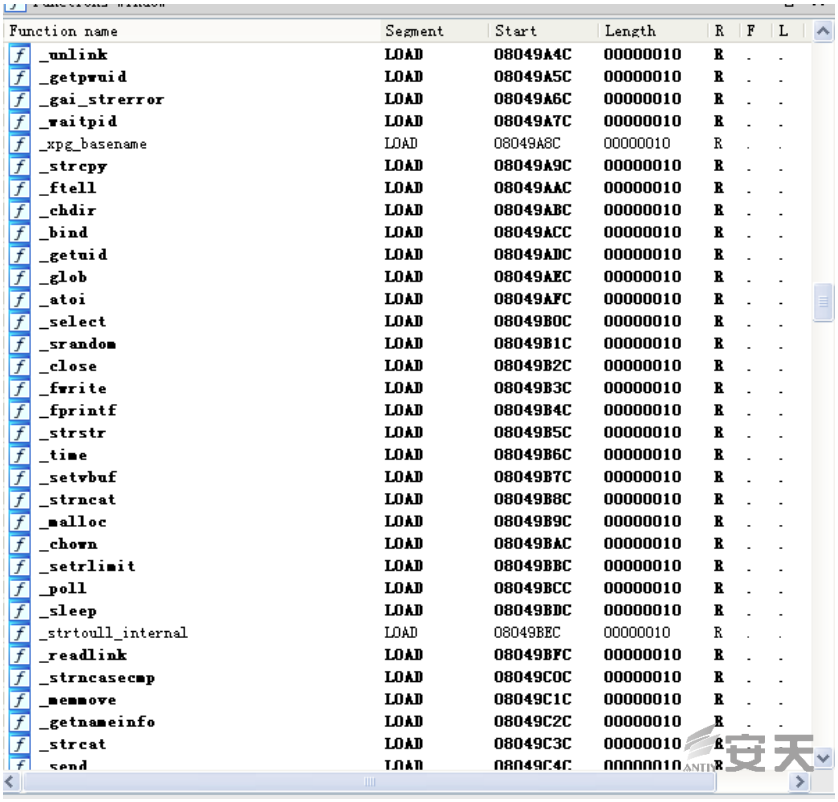


图 4 子进程判断

- 分别解密各种字符串，获取用户信息，包括系统版本等。
- 获取用户登陆信息 getpwnam。
- 查看文件 /bin/fast /sbin/login /usr/sbin/nologin 。
- 获取用户登陆密码 getpwuid。
- 读取用户日志 var/log /lastlog。

3.1.4 函数、数据动态加载

此样本所调用的函数和数据都是动态加载调用，在分析中需要动态调试，经过分析我们把函数调用地址通过动态分析解密出来如下图：



Function name	Segment	Start	Length	R	F	L
_unlink	LOAD	08049A4C	00000010	R	.	.
_getpwuid	LOAD	08049A5C	00000010	R	.	.
_gai_strerror	LOAD	08049A6C	00000010	R	.	.
_waitpid	LOAD	08049A7C	00000010	R	.	.
_xpg_basename	LOAD	08049A8C	00000010	R	.	.
_strcpy	LOAD	08049A9C	00000010	R	.	.
_ftell	LOAD	08049AAC	00000010	R	.	.
_chdir	LOAD	08049ABC	00000010	R	.	.
_bind	LOAD	08049ACC	00000010	R	.	.
_getuid	LOAD	08049ADC	00000010	R	.	.
_glob	LOAD	08049AEC	00000010	R	.	.
_atoi	LOAD	08049AFC	00000010	R	.	.
_select	LOAD	08049B0C	00000010	R	.	.
_srandom	LOAD	08049B1C	00000010	R	.	.
_close	LOAD	08049B2C	00000010	R	.	.
_fwrite	LOAD	08049B3C	00000010	R	.	.
_fprintf	LOAD	08049B4C	00000010	R	.	.
_strstr	LOAD	08049B5C	00000010	R	.	.
_time	LOAD	08049B6C	00000010	R	.	.
_setvbuf	LOAD	08049B7C	00000010	R	.	.
_strncat	LOAD	08049B8C	00000010	R	.	.
_malloc	LOAD	08049B9C	00000010	R	.	.
_chown	LOAD	08049BAC	00000010	R	.	.
_setrlimit	LOAD	08049BBC	00000010	R	.	.
_poll	LOAD	08049BCC	00000010	R	.	.
_sleep	LOAD	08049BDC	00000010	R	.	.
_strtoull_internal	LOAD	08049BEC	00000010	R	.	.
_readlink	LOAD	08049BFC	00000010	R	.	.
_strncasecmp	LOAD	08049C0C	00000010	R	.	.
_memmove	LOAD	08049C1C	00000010	R	.	.
_getnameinfo	LOAD	08049C2C	00000010	R	.	.
_strcat	LOAD	08049C3C	00000010	R	.	.
_send	LOAD	08049C4C	00000010	R	.	.

图 5 函数调用地址

3.1.5 字符串解密分析

样本内部采用了一种自定义的加密算法，用于加密内部要用到的字符串信息，该算法共被调用了 115 次，加密算法如下：

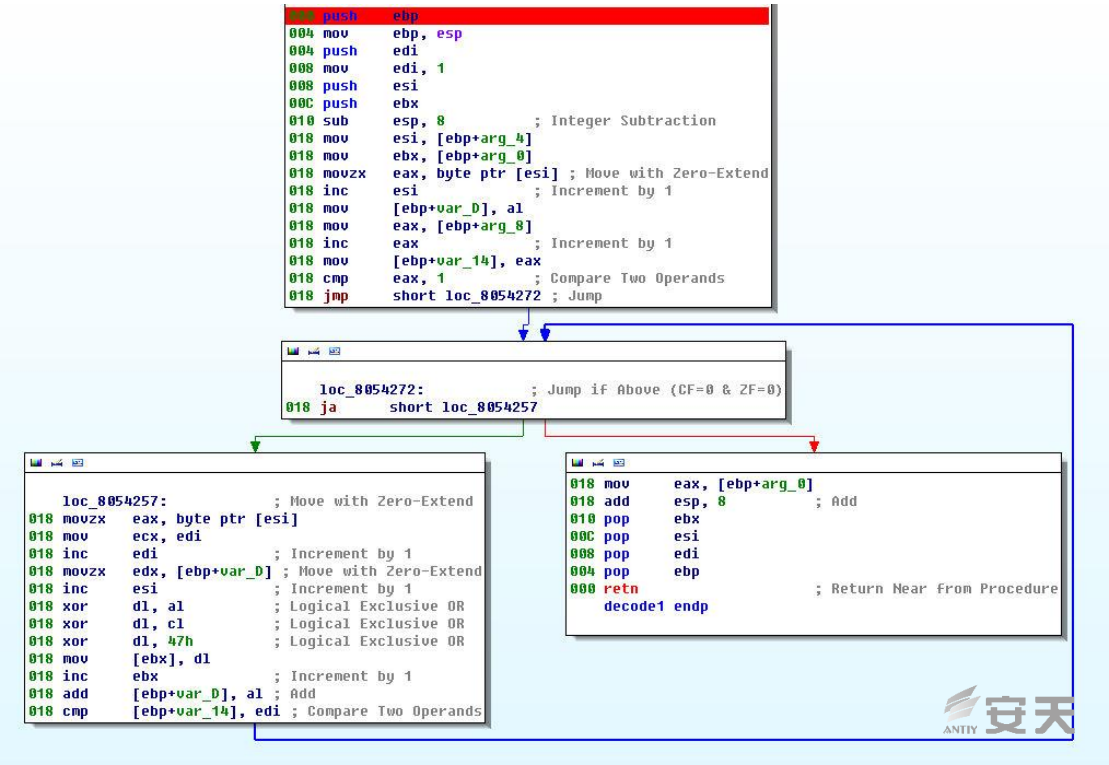


图 6　Linux 样本字符串加密算法

3.1.6　网络通信加密

DoubleFantasy 的 Linux 样本在网络通信时，硬编码在样本中的 16 位密钥与 DoubleFantasy 在 Windows 平台样本中加密注册表相关数据的 16 位密钥相同：

66 39 71 3c 0f 85 99 81 20 19 35 43 fe 9a 84 11

经计算后生成的子密钥为：

E9 BE CD E0 A8 9F 4D DB C3 42 AC 2B 24 77 AB CB 5A C1 52 F8 5B 3E F0 78 CB 01 0A 69 29 8F 85 8C 03 9C 7C EF 5E 36 0E 8B C0 40 76 28 9C 9C F2 24 81 9D 02 72 4F 6A BB B5 5B 42 73 14 88 F2 73 75 8B F9 37 98 3B 9F 64 2B A3 C4 FF C7 8A 40 67 C1 25 9F 65 54 45 36 48 FF E2 86 05 1A F4 94 AC 2B 08 D5 E5 83 BE 2C AD EE D0 A6 98 CB 8D 35 ED EE C4 F0 8C F2 CD BA 87 03 54 27 3D 13 A7 9B 6A 05 C7 02 30 21 05 67 58 3B E6 A1 44 0A 37 16 3C 86 E9 BC 8B 20 1A 98 7E 28 E6 7F F7 CA F7 9E 38 31 7F F0 2F 93 11 2B 28 F0 FF 11 B7 FC 1C 63 86 CB

Linux 样本的自定义算法与 Windows 下的样本相同，而使用的加密密钥只有一个（因为 Linux 系统没有注册表，所以就没有注册表加密这功能），该密钥与 Windows 平台下注册表加密数据的 Key 相同（Windows 平台有两组 key，一组注册表 key 一组网络通讯 key），从下图中可以看出两个平台的二级密钥变化算法是相同的（具体算法可以参照 Windows 加密算法分析部分）。

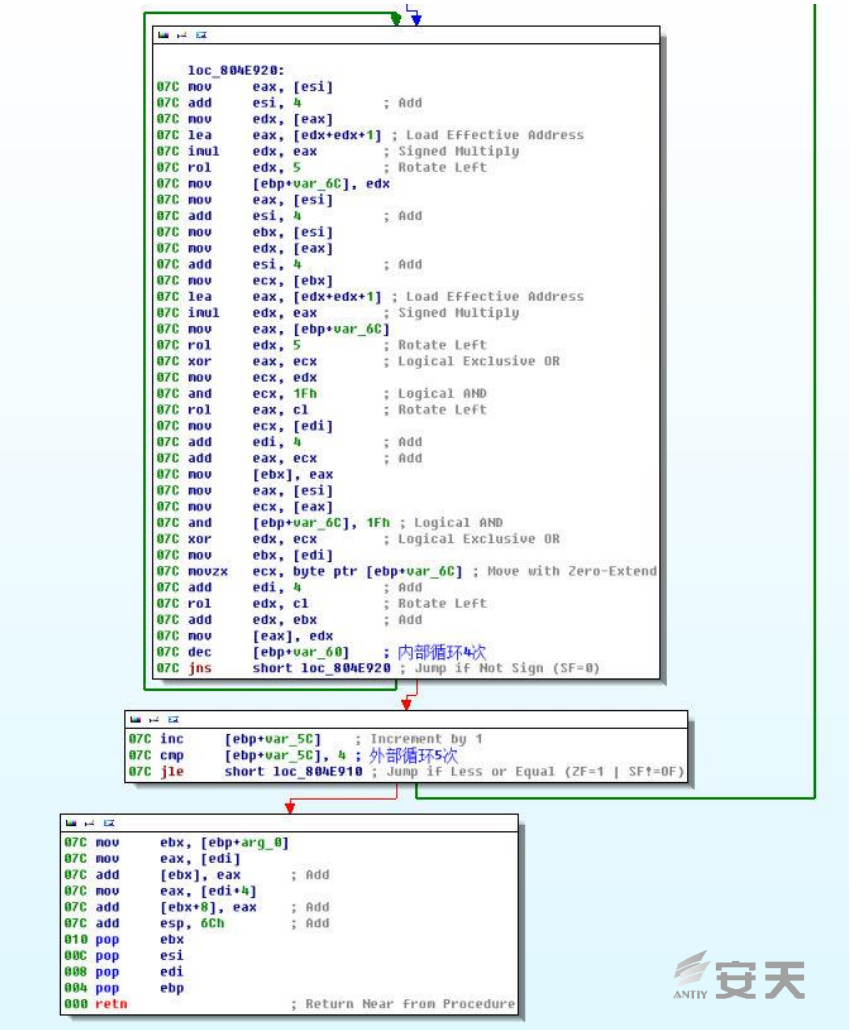


图 7 二级密钥变化算法

3.1.7 网络控制指令

Linux 样本的指令分支部分与安天此前所发布的报告中分析的 Windows 部分基本相同，Linux 样本共有 9 个分支指令，功能也大致相同，指令代码分别为：0x4A、0x4B、0x60、0x70、0x75、0x76、0x78、0x79、0x80。

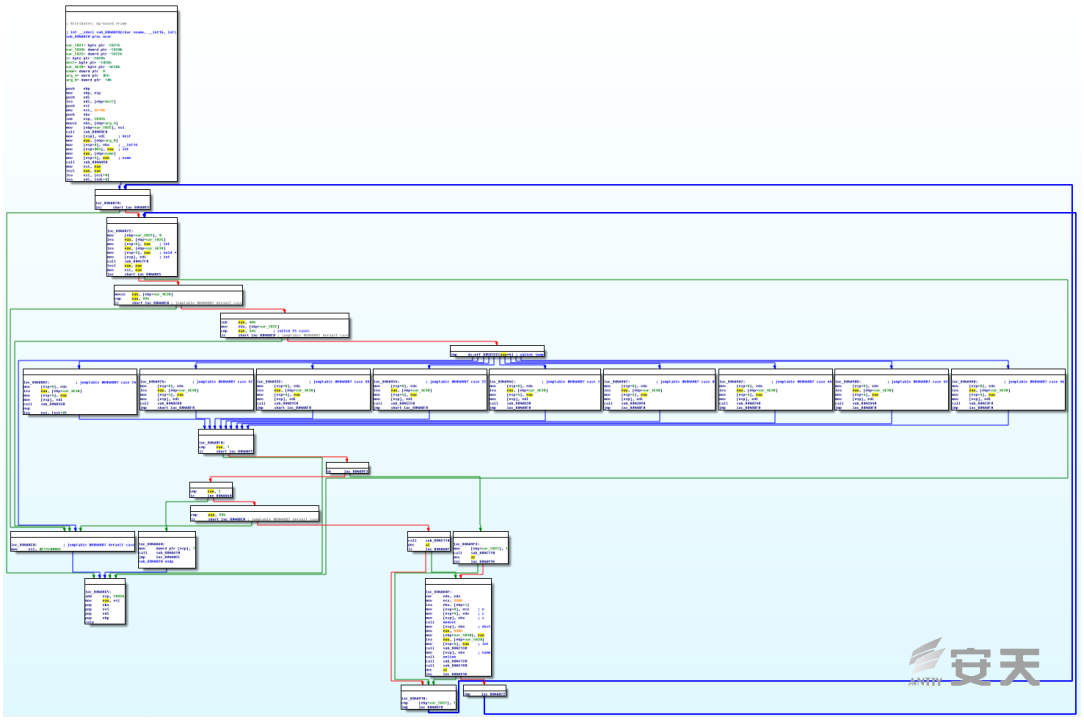
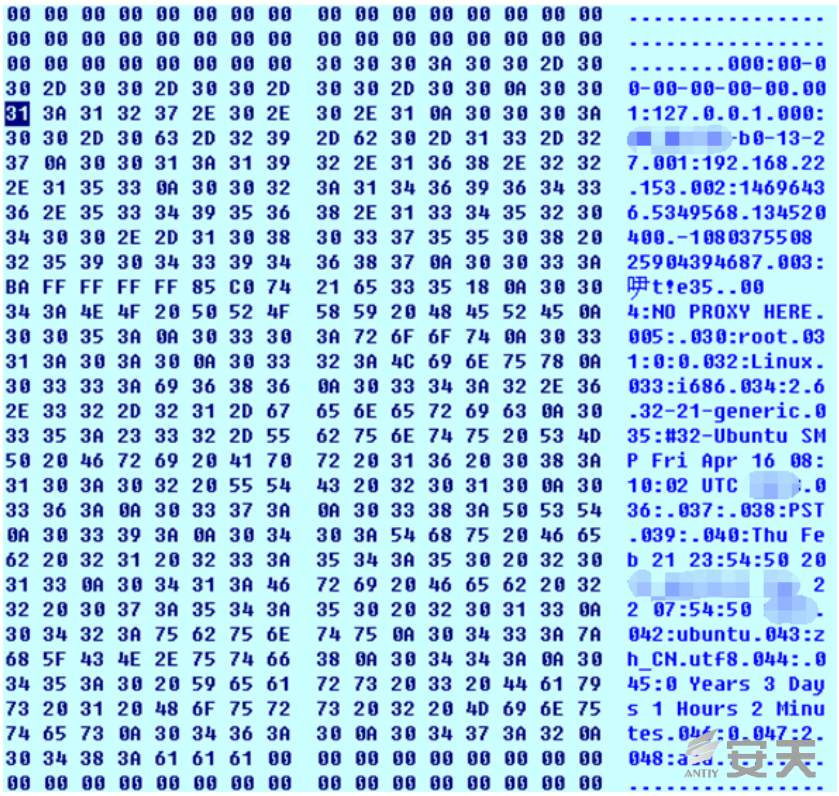


图 8 Linux 样本的指令分支代码

Linux 系统下的样本在指令上与 Windows 样本功能一致，仅在获取系统信息上有细微差别，Linux 样本获取信息格式如图：



```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00  30 30 30 3A 30 30 2D 30  ........000:00-0
30 2D 30 30 2D 30 30 2D  30 30 2D 30 30 0A 30 30  0-00-00-00-00.00
31 3A 31 32 37 2E 30 2E  30 2E 31 0A 30 30 30 3A  1:127.0.0.1.000:
30 30 2D 30 63 2D 32 39  2D 62 30 2D 31 33 2D 32  00-0c-29-b0-13-2
37 0A 30 30 31 3A 31 39  32 2E 31 36 38 2E 32 32  7.001:192.168.22
2E 31 35 33 0A 30 30 32  3A 31 34 36 39 36 34 33  .153.002:1469643
36 2E 35 33 34 39 35 36  38 2E 31 33 34 35 32 30  6.5349568.134520
34 30 30 2E 2D 31 30 38  30 33 37 35 35 30 38 20  400.-1080375508
32 35 39 30 34 33 39 34  36 38 37 0A 30 30 33 3A  25904394687.003:
BA FF FF FF FF 85 C0 74  21 65 33 35 18 0A 30 30  ºÿÿÿÿ.Àt!e35..00
34 3A 4E 4F 20 50 52 4F  58 59 20 48 45 52 45 0A  4:NO PROXY HERE.
30 30 35 3A 0A 30 33 30  3A 72 6F 6F 74 0A 30 33  005:.030:root.03
31 3A 30 3A 30 0A 30 33  32 3A 4C 69 6E 75 78 0A  1:0:0.032:Linux.
30 33 33 3A 69 36 38 36  0A 30 33 34 3A 32 2E 36  033:i686.034:2.6
2E 33 32 2D 32 31 2D 67  65 6E 65 72 69 63 0A 30  .32-21-generic.0
33 35 3A 23 33 32 2D 55  62 75 6E 74 75 20 53 4D  35:#32-Ubuntu SM
50 20 46 72 69 20 41 70  72 20 31 36 20 30 38 3A  P Fri Apr 16 08:
31 30 3A 30 32 20 55 54  43 20 32 30 31 30 0A 30  10:02 UTC 2010.0
33 36 3A 0A 30 33 37 3A  0A 30 33 38 3A 50 53 54  36:.037:.038:PST
0A 30 33 39 3A 0A 30 34  30 3A 54 68 75 20 46 65  .039:.040:Thu Fe
62 20 32 31 20 32 33 3A  35 34 3A 35 30 20 32 30  b 21 23:54:50 20
31 33 0A 30 34 31 3A 46  72 69 20 46 65 62 20 32  13.041:Fri Feb 2
32 20 30 37 3A 35 34 3A  35 30 20 32 30 31 33 0A  2 07:54:50 2013.
30 34 32 3A 75 62 75 6E  74 75 0A 30 34 33 3A 7A  042:ubuntu.043:z
68 5F 43 4E 2E 75 74 66  38 0A 30 34 34 3A 0A 30  h_CN.utf8.044:.0
34 35 3A 30 20 59 65 61  72 73 20 33 20 44 61 79  45:0 Years 3 Day
73 20 31 20 48 6F 75 72  73 20 32 20 4D 69 6E 75  s 1 Hours 2 Minu
74 65 73 0A 30 34 36 3A  30 0A 30 34 37 3A 32 0A  tes.046:0.047:2.
30 34 38 3A 61 61 61 00  00 00 00 00 00 00 00 00  048:aaa.........
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

图 9 Linux 样本获取信息格式

获取信息格式说明：

标号	说明	标号	说明	标号	说明
000	MAC 地址	**033**	平台类型如（i386\i686）	**042**	操作系统（Ubuntu）
001	IP 地址	**034**	系统内核版本	**043**	区域语言（zh_cn.utf8）
002	样本版本号	**035**	操作系统类型时间	**044**	未知
003	样本 clsid	**036**	未知	**045**	系统运行时间
004	代理设置信息	**037**	未知	**046**	未知
005	未知	**038**	PST	**047**	未知
030	用户名	**039**	未知	**048**	样本名称
031	密码	**040**	时间		
032	操作系统类型如（Linux）	**041**	时间		

4 SPARC 架构 Solaris 场景能力

方程式组织可能制造了第一个 SPARC 架构[9]下的具有 Rootkit 属性的恶意代码，并为 DoubleFantasy 的 Solaris[10] 版本来提供掩护。

4.1 Solaris 系统及 SPARC 架构

Solaris 是 Sun Microsystems 研发的计算机操作系统，采用 SPARC 架构或 X86 架构，主要用于工作站、服务器上的操作系统。Solaris 平台下的恶意代码比较罕见，从安天统计来看，即使加上之前的 SUN OS 时期，二进制编译形态的恶意代码变种数也不超过 60 种，而且几乎都是基于 X86 平台的。

SPARC 全称为“可扩充处理器架构”（Scalable Processor ARChitecture），是 RISC 微处理器架构之一，其指令集和 X86 有显著区别，并且有自己独有的窗口、延迟槽、过程调用特点。

SPARC 架构的计算机一般用于工业、航天相关领域，其在类似 IDC 和一般 IT 场景的使用极为罕见。

4.2 Rootkit 隐藏模块

该模块是 SPACR 架构的 Solaris 平台下的一个 Rootkit 程序，同其他 Rootkit 程序一样，它主要负责隐藏主功能样本文件、以及相关衍生的文件和其自身，包括进程、文件、和服务信息。它首先在目标机器上运行，侦查目标机器的系统环境、配置信息、网络状态，并隐藏指定的文件和进程。

4.2.1 文件标签

病毒名称	Trojan[Rootkit]/Solaris. Equation
原始文件名	[redacted]
MD5	[redacted]
处理器架构	SPARC（32 位平台）
文件大小	[redacted]
文件格式	BinExecute/ELF
时间戳	N/A
数字签名	无
加壳类型	无
编译语言	C 语言

4.2.2 样本主函数

样本共有 249 个函数，如图所示为样本主函数流程，部分函数也相对复杂，且样本内部同样存在多种加密数据。

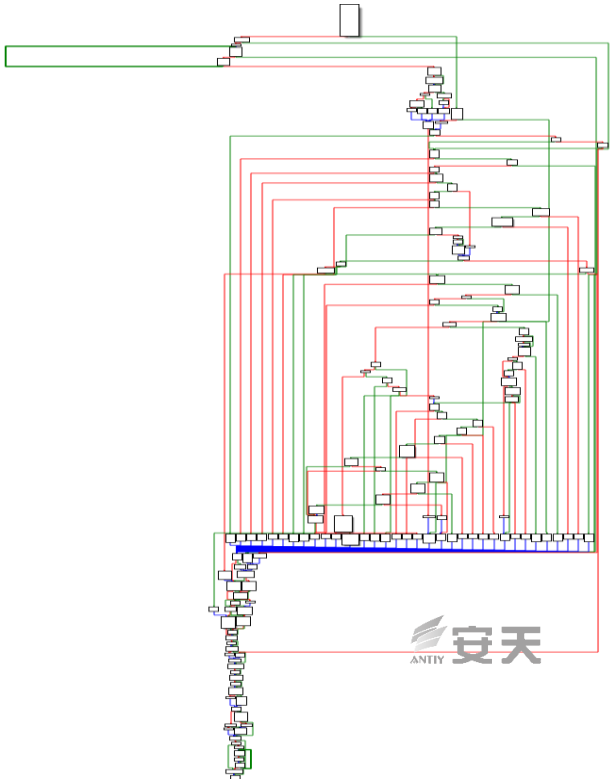


图 10 样本主函数

4.2.3 衍生文件名及路径

样本运行后根据内部配置的两组字符串组合生成文件名，作为自身的新文件名，并将自身复制到/sbin/目录下。

字符串 1	字符串 2
audit	admr
boot	agent
cache	conf
core	client
cron	info
init	mgr
inet	statd
filesys	serv

通过上表可以发现，这些单词都是系统文件、系统命令中使用的高频单词或前后缀。因此样本的文件名称是经过精心构造的，文件名极具迷惑性，换在系统文件中，一般的管理员也难以察觉异样。

4.2.4 启动脚本

样本使用服务的方式实现开机启动，在 etc/rc.d/目录下创建脚本（S85s%），此脚本会作为开机要执行的服务以 start 参数运行。

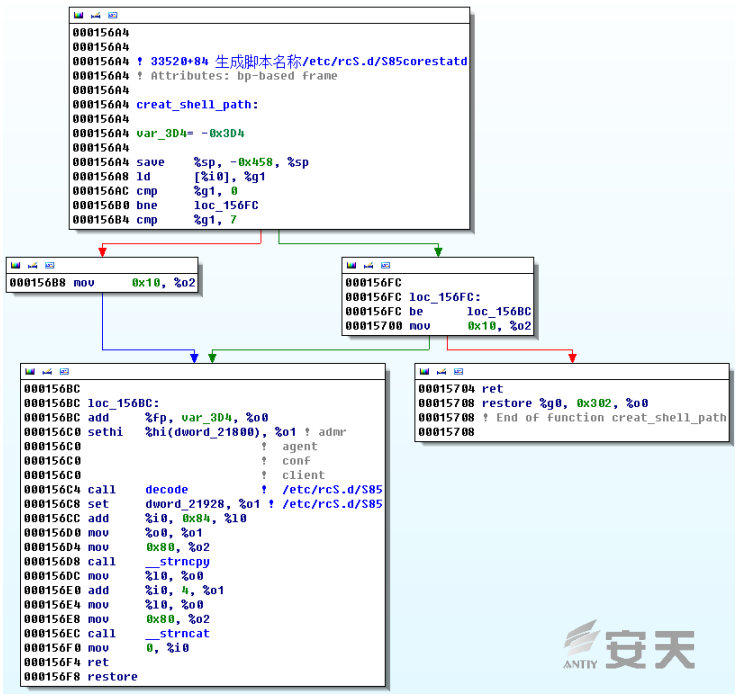


图 11 服务脚本

S85s%文件的内容是加密的，样本运行时调用自身函数解密，并修改其中文件名的变量，再将其写入到/etc/rc.d/目录（下图%E 处会修改为样本自身路径）。

```
0001541C add     %fp, var_801, %o0
00015420 sethi   %hi(byte_21400), %o1
00015424 call    decode          ! #!/sbin/sh  脚本
00015428 set     dword_21710, %o1 ! #!/sbin/sh
00015428                          ! #
00015428                          ! # Copyright (c) 1995, 1997 by Sun Microsystems, Inc.
00015428                          ! # All rights reserved.
00015428                          ! #
00015428                          ! #ident  "@(#)%N 1.2     97/12/08 SMI"
00015428                          !
00015428                          ! case "$1" in
00015428                          ! 'start')
00015428                          !         %E
00015428                          !         ;;
00015428                          !
00015428                          ! 'stop')
00015428                          !         ;;
00015428                          !
00015428                          ! *)
00015428                          !         echo "Usage: $0 { start | stop }"
00015428                          !         exit 1
00015428                          !         ;;
00015428                          ! esac
00015428                          ! exit 0
0001542C call    modify_shell     ! 完善脚本变量
00015430 mov     %l0, %o1         ! 33520
00015434 orcc    %o0, 0, %i0
00015438 bne     locret_15454
0001543C nop
```

图 12 解密后脚本内容

4.2.5 隐藏目录、文件

样本会根据目标机器的 HOSTID 生成 MD5，然后再将 MD5 进行一个类 base64 的算法计算，最后取前 6 位，将.tmp 与这 6 位字符拼接成文件夹名称，然后创建该文件夹。

```
! 创建 tmp%6s文件夹
! Attributes: bp-based frame

create_MD5Path:                         ! CODE XREF: path:loc_1390C↑p

var_30          = -0x30

                save    %sp, -0x90, %sp
                orcc    %i0, 0, %l0
                sethi   %hi(-0x10000000), %i0
                be      locret_15268
                set     -0xFFFFFF7, %i0
                add     %fp, var_30, %l1
                mov     %l0, %o0
                call    get_arg_stat    ! 取文件夹参数1stat 放入var
                                        !
                mov     %l1, %o1
                mov     0x1C0, %o1
                mov     %o0, %i0
                call    __mkdir
                mov     %l0, %o0        ! %s.tmp%6s 目录
                cmp     %o0, 0
                bne     locret_15270
                cmp     %i0, 0
                bne     locret_15268
                mov     %l1, %o1        ! 创建目录成功
                call    sub_1A378
                mov     %l0, %o0
                mov     %l0, %o0
                call    sub_1A4B0
                mov     %l1, %o1
```

图 13 样本创建的文件夹名

样本还会根据运行参数，将其他文件复制到此文件夹下执行，并负责隐藏此文件夹下的所有文件。

4.2.6 版本判断

样本通过 uname 函数确定系统不是 sun4m、sun4d 版本，通过读取/dev/ksyms 文件判断系统架构：i386、ia64、sparc、sparcv9，确定是 SPARC 架构，确定 release 版本必须是 5.1。

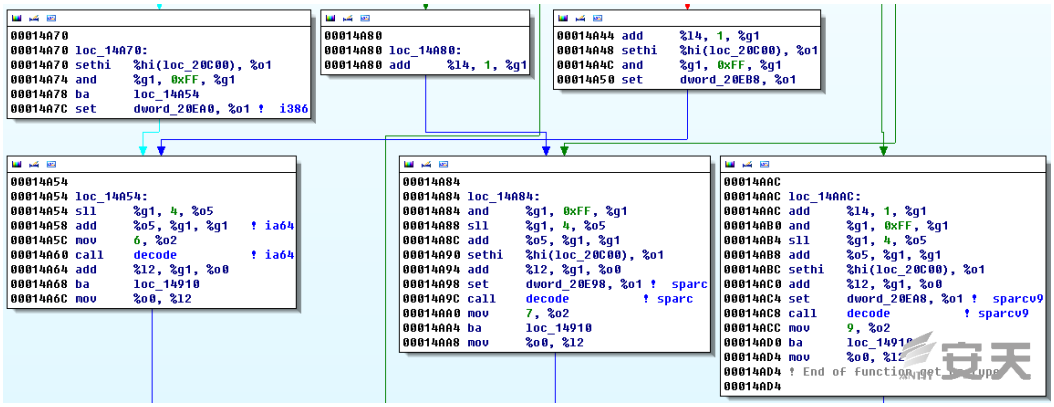


图 14 版本判定

4.2.7 加密配置数据

样本内部存在多处加密算法，其中一个调用多次，我们分析并解密出其数据。

```
                                ! CODE XREF: decode+38↓j
ldub    [%i1+%g1], %o4
add     %i0, %g1, %o3
xor     %o4, 0x47, %o2
btog    %o5, %o2
btog    %g1, %o2
stb     %o2, [%o3-1]
inc     %g1
cmp     %g1, %i2
bcs     loc_1FB24
add     %o5, %o4, %o5
```

图 15 加密算法

解密的加密数据：

偏移	明文	偏移	明文
0x10cd0	/platform/%s/kernel/sparcv9/unix	0x11830	mgr
0x10cf8	/var/sadm/i	0x11838	statd
0x10d28	SUNW	0x11840	serv
0x10d30	/var/sadm/patch/%s/README.%s	0x11848	svcd
0x10d50	var/sadm/pkg/%s/pkginfo	0x11851	\

0x10d70	PATCHLIST	0x11855	\W
0x10d80	/var/sadm/pkg/%s/pkginfo	0x11859	\O
0x10da0	PATCH_INFO	0x1185d	\G
0x10db8	Requires:	0x11861	\w
0x10dc8	Ob	0x11865	\o
0x10dcc	!8I 柞;	0x11869	\g
0x10dd8	Incompatibles:	0x1186d	\
0x10df0	module_main	0x11871	\
0x10e10	%s/%s	0x11878	audit
0x10e18	date	0x11880	boot
0x10e20	/etc/mnttab	0x11888	cache
0x10e38	swap	0x11890	core
0x10e40	tmpfs	0x11898	cron
0x10e48	ro	0x118a0	init
0x10e50	noexec	0x118a8	inet
0x10e60	D	0x118b0	filesys
0x10e68	sun4m	0x118c0	key
0x10e70	sun4d	0x118c8	ntp
0x10e78	sparc	0x118d0	root
0x10e80	/dev/ksyms	0x118d8	sys
0x10e98	sparc	0x118e0	rpcd
0x10ea0	i386	0x118e8	vol
0x10ea8	sparcv	0x11940	/
0x10eb8	ia64	0x11948	/usr/bin/
0x10ec0	sparc	0x11958	/bin/
0x10ec8	SunOS	0x11960	/sbin/
0x10ed0	Generic	0x11970	var/tmp/faipprep001
0x10ee0	boothowto	0x11990	init
0x10ef8	/dev/ksyms	0x11998	fini
0x10f08	/dev/kmem	0x119a0	minit
0x116d0	/var/tmp/	0x119a8	fini
0x116e0	/lib/	0x119b0	mdata
0x116e8	/dev/	0x119b8	priocntlsys

0x116f0	/etc/	0x119c8	/dev/ksyms
0x116f8	/	0x119d8	init
0x11700	%s.tmp%6s	0x119e0	/dev/kmem
0x11710	#!/sbin/sh	0x119f0	/dev/mem
0x11800	admr	0x11a00	/proc/self
0x1180a	NCk	0x11a10	.got
0x11810	conf	0x11a18	.got
0x11818	client	0x11a28	.got
0x11828	info	0x11a30	GLOBAL_OFFSET_TABLE_

4.2.8 解密执行其他代码/样本

样本在文件尾部添加加密数据，执行后通过末尾数据确定加密数据大小，通过定义的格式进行解析和读取，解密数据后猜测会加载执行。

```
00056890h: 1C AD 6C 49 E9 AB 37 4D 9D 43 EE A5 6B A6 F7 5C ; .璩I楂7M滴瞬k  \
000568a0h: FC 81 49 17 B2 FC AC E5 FF A1 3B A5 5A 18 AD A5 ; 蘧I.顫    ?  .
000568b0h: 9D 7D F8 F8 2B 6F 6A 47 90 4A 5D 15 A2 0C 9C 4A ; 顗  +ojG恇].?涮
000568c0h: 28 6B 74 90 4C AE 70 7D 1A 0C 02 67 E5 F0 99 10 ; (kt怏昀)...g屡?
000568d0h: AC 96 B1 17 36 9B 52 3E 77 F3 0B 48 48 E0 92 49 ; 壅?6汴>w?HH鄴I
000568e0h: 0C 24 1D CD F9 F1 A1 69 01 FC 03 04 F9 3B 6B 34 ; .$.往瘛i.?.?k4
000568f0h: 50 A3 D4 1D 8C D9 8F 00 B2 04 33 A9 67 D9 5C 43 ; P丌.屬?ゞ賊C
00056900h: 30 ED 01 80 00 00 00 00 00 00 00 0F 00 00 00 13 ; 0?€...
00056910h: 61 A7 08 5A 67 71 DC A4 53 D4 10 EF 41 90 65 9D ; a?Zgq埭S?蹩恊?
00056920h: 75 F9 82 03 02 10 03 00 04 38 A2                ; u鹃....8?□
```

4.3 DoubleFantasy 的 Sparc 架构模块

该样本与 Windows、Linux 平台样本的功能基本相同，主要区别在于 CPU 架构、汇编指令、配置信息存储位置、获取系统信息等。

4.3.1 文件标签

病毒名称	Trojan/Solaris.DoubleFantasy
原始文件名	[redacted]
MD5	[redacted]
处理器架构	SPARC（32 位平台）
文件大小	[redacted]
文件格式	BinExecute/ELF

时间戳	N/A
数字签名	无
加壳类型	无
编译语言	C 语言

4.3.2 基本功能

- 初始化字符串、动态数组，解密内部配置信息。
- 连接 Google 或 Yahoo 网址判断网络连通状态。
- 连接远程 URL 地址。样本会收集主机的信息回传至远程地址，并等待远程主机发送指令。
- 具有读取系统账户密码文件的功能，可以窃取用户及密码信息。
- 样本内部实现了以守护进程模式运行，可以达到自我保护防止被结束的功能。
- 该样本采用多种加密算法加密字符串信息。
- 获取系统大量信息并回传到服务器（如计算机名称、IP 地址、进程信息、账户信息等，详细内容可见本章节后面详细分析）。
- 网络指令部分，具有 7 条网络指令，功能上与 Windows 版本相同，可对计算机进行相对应的指令操作，对应指令详细功能见本章后面详细分析。

4.3.3 配置信息加密

由于 Solaris 系统没有 Windows 的注册表，因此该样本的配置数据会直接解密后使用，其中一个解密算法如下，该解密函数共调用 63 次。

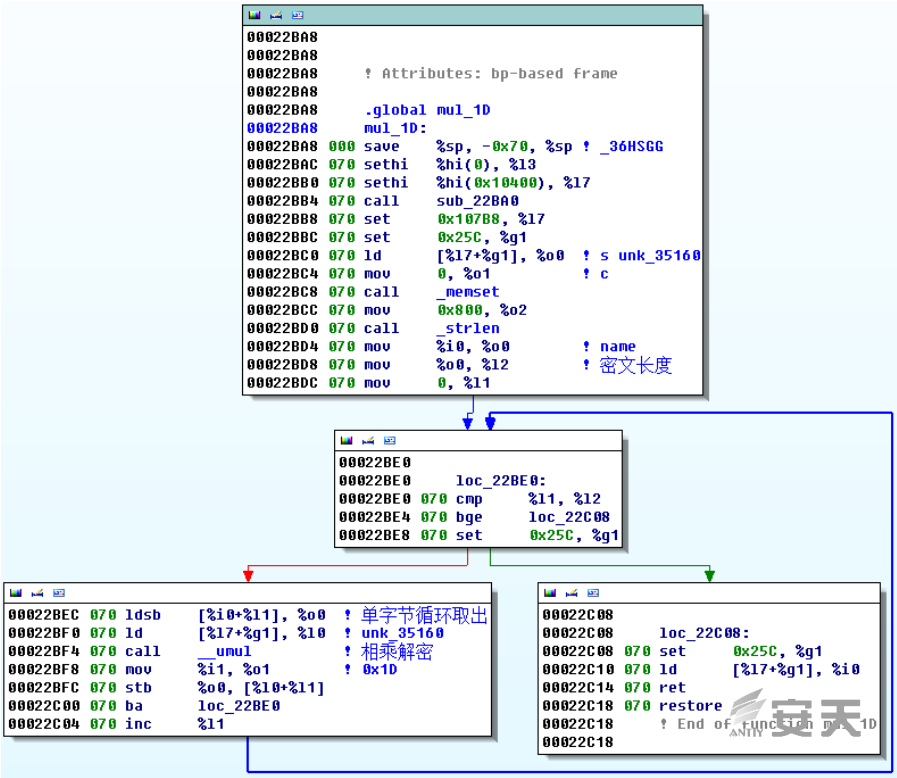


图 16 字符串解密

解密出字符串信息见下表：

偏移	明文	偏移	明文
0x1346c	' 200 Connection established'	0x13560	'Content-Length:'
0x13470	' 200 OK'	0x13458	'Content-Length: %d'
0x133fc	' days '	0x13460	'Content-length: %d'
0x13400	' hrs '	0x13488	'Cookie: %s'
0x13540	' HTTP/1.1\r\n'	0x13554	'GET '
0x1340c	' logged in'	0x13544	'Host: '
0x13404	' mins '	0x13474	'HTTP/'
0x13408	' total'	0x13478	'HTTP/1.0 200 OK'
0x133f4	' yrs '	0x13534	'http://'
0x133d8	""	0x13384	'I_MASK'
0x13584	'%02x-%02x-%02x-%02x-%02x-%02x'	0x133ec	'LANG'
0x13594	'%u.%u.%u.%u'	0x133f0	'LANGUAGE'
0x134dc	'/.mozilla/'	0x133c0	'LD_PRELOAD='
0x134e0	'/.mozilla/firefox/'	0x13418	'M_MASK'
0x134c4	'/.netscape'	0x133e4	'MACHTYPE'

0x13438	'/bin/false'	0x134b4	'network.proxy.http'
0x13520	'/bin/false'	0x134b8	'network.proxy.http_port'
0x1338c	'/dev/null'	0x134bc	'network.proxy.ssl'
0x133b0	'/dev/null'	0x134c0	'network.proxy.ssl_port'
0x134c8	'/preferences.js'	0x1348c	'p'
0x134cc	'/prefs.js'	0x133b8	'PATH'
0x13428	'/proc'	0x133bc	'PATH='
0x1343c	'/sbin/nologin'	0x1355c	'POST '
0x13524	'/sbin/nologin'	0x13410	'process info: '
0x13390	'/tmp/'	0x1354c	'Proxy-Connection: close\r\n'
0x133b4	'/tmp/'	0x134d8	'S'
0x135a0	'@C\xe3\xc0'	0x1353c	'S'
0x13558	'\r\n'	0x133c4	'sendmail'
0x13464	'\r\n\r\n'	0x13484	'SESSID="0%x%s%x:eac:%lu:%lu"\r\n'
0x13588	'0x%02x%02x%02x%02x%02x%02x'	0x1349c	'user_pref("'
0x1341c	'0xA857'	0x134a0	'user_pref("%s"
0x13388	'0xAA%llu'	0x134a8	'user_pref("%s%s'
0x13568	'CONNECT '	0x134f8	'v'
0x13550	'Connection: close\r\n'	0x13548	'y"'
0x13454	'Content-Length:'	0x1347c	'y"y"'
0x1345c	'Content-length:'		

样本使用的另一个加密字符串的算法，用于加密样本运行时需要使用的配置信息，解密算法如下：

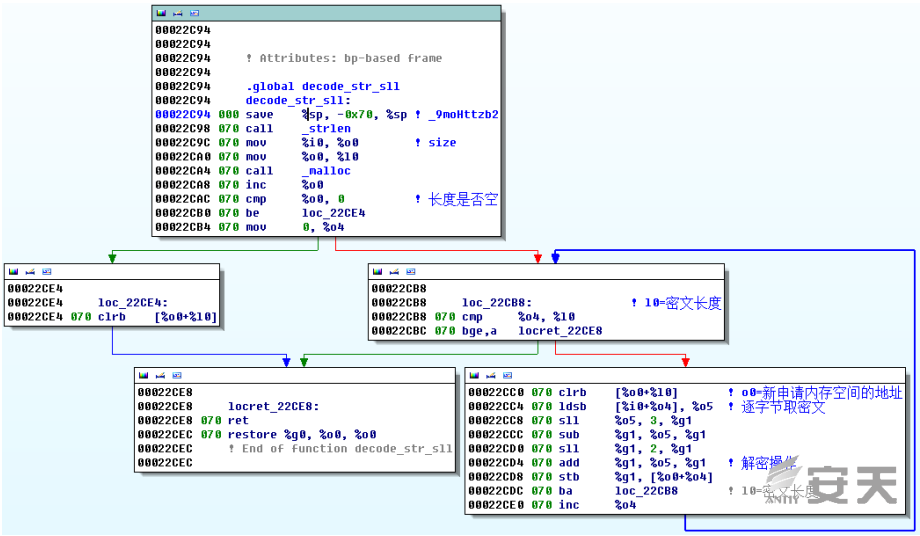


图 17 另一解密算法

解密内容见下表：

偏移	明文	偏移	明文
0x13c12	'www.google.com'	0x1439b	'ntp'
0x13d11	'www.yahoo.com'	0x143ac	'mail'
0x13e32	'\x91 xxx atech.com'	0x143bd	'mysql'
0x14034	'\\X'	0x143cd	'named'
0x1406c	'\\'	0x143db	'sys'
0x140e7	'\x91puX;\xc7;\xc7Xupp\x8dTq\x01{User-Agent: Mozilla/5.0 (X11; U; Solaris; en-US; rv:1.7.5) Gecko/20041111 Firefox/1.0\r\n'	0x143ec	'smtp'
0x1419d	'Accept: image/png	0x143fe	'nobody'
0x14314	'"'	0x1440c	'auth'
0x1437e	'daemon'	0x1441a	'LP'
0x1438b	'adm'	0x1442c	'UUCP'

4.3.4 网络通信加密

Solaris 样本的自定义算法与 Windows 下的样本相同，而使用的加密密钥只有一个（因为 Solaris 系统没有注册表，所以就没有注册表加密的功能），该密钥与 Windows 平台下注册表加密数据的 Key 相同，两个平台的自定义加密算法是相同的（具体算法可以参考 3.1.6 加密算法分析部分）。

通过安天 CERT 分析，得到原始 16 位密钥为：

66 39 71 3c 0f 85 99 81 20 19 35 43 fe 9a 84 11

长度为 16 字节，与 Windows 的原始 16 位密钥长度相同。

由于 Solaris 和 Windows 样本生成网络通信子密钥的算法相同，那么可生成子密钥：

E9 BE CD E0 A8 9F 4D DB C3 42 AC 2B 24 77 AB CB 5A C1 52 F8 5B 3E F0 78 CB 01 0A 69 29 8F 85 8C 03 9C 7C EF 5E 36 0E 8B C0 40 76 28 9C F2 24 81 9D 02 72 4F 6A BB B5 5B 42 73 14 88 F2 73 75 8B F9 37 98 3B 9F 64 2B A3 C4 FF C7 8A 40 67 C1 25 9F 65 54 45 36 48 FF E2 86 05 1A F4 94 AC 2B 08 D5 E5 83 BE 2C AD EE D0 A6 98 CB 8D 35 ED EE C4 F0 8C F2 CD BA 87 03 54 27 3D 13 A7 9B 6A 05 C7 02 30 21 05 67 58 3B E6 A1 44 0A 37 16 3C 86 E9 BC 8B 20 1A 98 7E 28 E6 7F F7 CA F7 9E 38 31 7F F0 2F 93 11 2B 28 F0 FF 11 B7 FC 1C 63 86 CB

此子密钥才是用于加解密发送、接收数据的。

4.3.5 网络控制指令

在对 Solaris 样本的分析中我们发现它的功能要比 Windows 样本的指令少一些，Solaris 平台下只有 7 个指令，功能上与 Windows 大致相同。下面为两个平台下 IDA 中的对比图，多图中可以看出 Solaris 平台下的样本指令比 Windows 平台上少很多结构图也很简单。

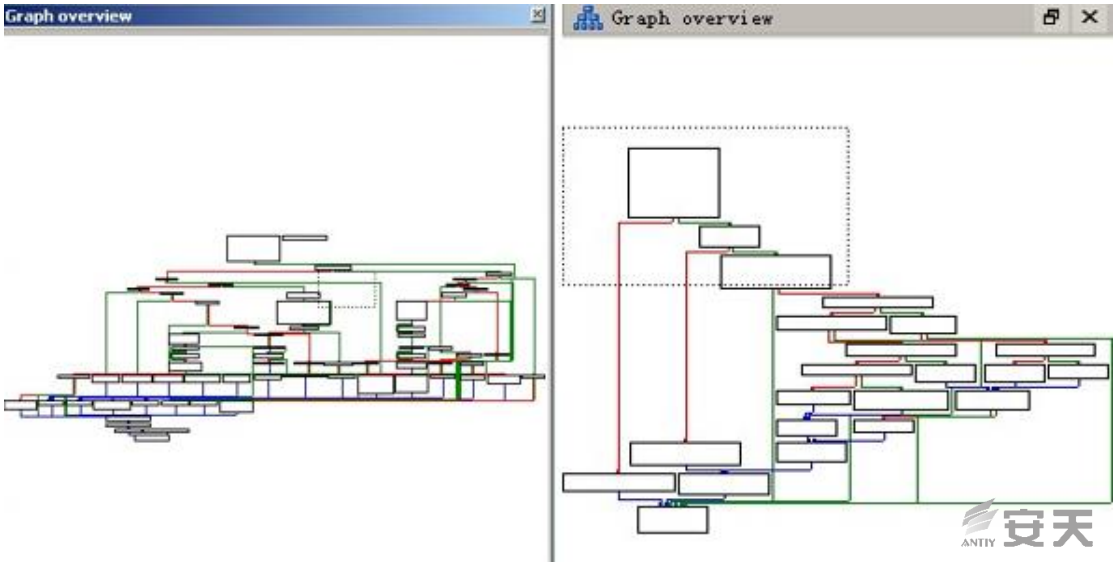


图 18 Windows 平台与 Solaris 平台下网络指令结构对比

Solaris 样本的指令功能并未在上图中实现，起初我们以为 Solaris 样本的指令功能还未完成，不过在进行进一步的分析后我们发现，Solaris 样本采用一种特殊的动态计算方式来跳转到不同的指令分支代码，下图中红色部分即为动态计算后跳转的指令分支。

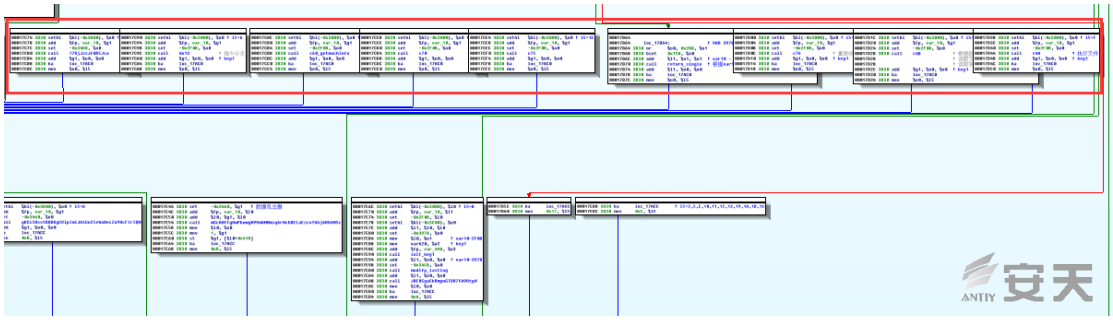


图 19 Solaris 指令分支函数

Solaris 样本各个指令功能简要描述如下，大体功能与 Windows 指令相同：

16 进制指令代码	指令功能
0x42	清理感染痕迹，删除自身
0x4A	创建文件
0x44	写入文件

0x56	执行文件
0x4B	读取文件回传
0x60	收集大量信息回传（具体格式见下表）
0x70	更新样本配置信息
0x75	更新样本 SLEEP 时间并重新收集信息回传
0x76	更新 C&C 服务器地址

其中下载执行部分样本与 Windows 一样，使用相同的指令标记，也是通过三步（创建、写入、执行）来完成下载执行的功能，只是在代码结构上有所不同，Solaris 把三条指令整合到一个函数中。

执行文件时，先给文件提权，然后使用 execle 函数带有参数执行文件：

- 参数 1：文件 B 路径
- 参数 2：文件名 B 或"sendmail"（猜测与 mail 有关）
- 参数 3：0
- 参数 4：PATH=%PATH%（环境变量）

例如：execle("/usr/bin/sample","sample", NULL, %envp%);

```
                        ! CODE XREF: execute_file+194↑j
set     0x58, %g1
mov     0x1D, %o1
call    mul_1D          ! 'sendmail'
ld      [%l7+%g1], %o0
mov     %o0, %o1        ! arg0
mov     %l4, %o0        ! status

                        ! CODE XREF: execute_file+1A0↑j
mov     %l0, %o3        ! PATH=%PATH%   LD_PRELOAD=
call    _execle
mov     0, %o2
```

图 20 执行文件参数

Solaris 样本指令功能、数据包格式与 Windows 样本相同，指令的详细功能、数据包格式说明可见 Windows 平台样本的指令分析。

Solaris 样本下收集的系统信息与 Windows 略有不同，具体如下：

计算机名	HostID	MAC 地址	IP 地址	用户名	Typically user's full name
用户 UID/GID	系统硬件架构信息	系统详细时间	系统的默认语言类型	当前程序运行路径	系统进程信息

5 总结

5.1 以真实威胁驱动我国信息防御能力的改进

安天希望用自己的工作告诉中国用户，那些关于超级攻击组织全平台覆盖能力的种种爆料，并非传说，而且是一种真实的威胁，是一种既定的事实。

在我国安全防御的实践中，有一种先入为主的观点，即认为由于各种规定和约束，暴露在互联网上的节点，乃至能够访问互联网的内网中，并不存放高价值的信息。“一切有价值的信息都存在于隔离网内”——这是一个美好的愿景和想象，但并非在这个信息大量产生、高速流动时代的真实情况。同时在大数据时代，高价值信息的定义和范围也在不断变化着。更多的信息资产已经不可避免地分布在公共网络体系中。而对这些资产的窥视和攻击也在持续增加着。而超级攻击组织则是类似攻击的始作俑者和长期实践者。

针对 DNS 服务器的入侵，可以辅助对其他网络目标实现恶意代码注入和信息劫持；针对邮件服务器的植入可以将用户所有的邮件通联一网打尽，针对运营商骨干节点的持久化，可以用来获取全方位的信息，包括收获类似 Camberdada[11]计划中说的那种“轻而易举的胜利”。

注：Camberdada 计划是斯诺登曝光的一份监听行动计划，相关机构通过在运营商的持久化节点，监听用户发放给杀毒厂商的邮件，以发现自己的攻击是否暴露，并实现对其他方投放的样本捕获和再利用。

而“物理隔离”的安全神话也已经到了应该破灭的时候，**习近平总书记在 4.19 讲话中已经提醒国内用户和网络安全工作者：“‘物理隔离’防线可被跨网入侵，电力调配指令可被恶意篡改，金融交易信息可被窃取，这些都是重大风险隐患。”**

而中国庞大又脆弱的信息化肌体则又面对着武装到牙齿的对手。攻击载荷的代码工程规模、作业链条的精密设计、全方位无死角的平台覆盖都已显示了方程式攻击组织这样的超级攻击组织空前的攻击能力。而根据相关曝光的信息，其所发动的面对大量的关键目标为期数年的攻击，也表明了这一组织极为坚定的攻击决心。安天在此前的研究中曾将类似的攻击能力组织称为 A^2PT，并从恶意代码载荷视角给出了 A^2PT 的若干评价标准。这些标准与方程式组织的行为与能力高度吻合。

安天所总结的 A^2PT 特点	方程式攻击装备和作业方式特点
有充足的 0day 储备	Fanny 利用 LNK 0day 漏洞、MS09-025 漏洞
载荷部分高度复杂，高度模块化	高度复杂、模块化的 EquationDrug、GrayFish 攻击组件
本地加密抗分析，网络严格加密通讯和伪装	配置数据资源加密 注册表、网络通信加密
多种植入方式	网络入侵 物流劫持（猜测）

	人员现场植入（猜测）
普遍使用无文件载体技术	Bootkit 启动 注册表存放样本、分段解密
持久化向深度扩展（固件），向广度扩展（防火墙、邮件网关、局网内横向移动）	硬盘固件修改 防火墙和其他网络安全设备植入 针对邮件服务器进行持久化
完整的覆盖所有操作系统平台（含移动）	Windows、Linux、Solaris、OS X 都存在样本

就像我们此前所概括的那样，相关超级攻击组织拥有“**成建制的网络攻击团队、庞大的支撑工程体系与制式化的攻击装备库、强大的漏洞采集和分析挖掘能力和关联资源储备以及系统化的作业规程和手册，具有装备体系覆盖全场景、漏洞利用工具和恶意代码载荷覆盖全平台、持久化能力覆盖全环节的特点**。面对这种体系化、既具备工业级水准又具有高度定向性的攻击，永动机注定停摆，银弹注定哑火。**想要达成防御效果，实现追踪溯源，唯有以清晰的战略、充分的成本投入，以体系化的防守对决体系化[12]的攻击，通过长期艰苦、扎实的工作和能力建设，才能逐渐取得主动。**

5.2 我们的努力和对能力型厂商深入协作的期待

从 2010 年开始，安天先后对“震网”、“毒曲”、“火焰”、“APT-TOCS(海莲花)”、“白象”、“乌克兰停电”、“方程式”等高级攻击行动或攻击组织进行了深入的分析，累计发布了数百页的分析报告。毫无疑问，高级威胁检测产品的能力，是依托扎实有效的分析过程来不断改进的。安天发布了面向高级威胁检测和态势感知的产品体系：安天的**探海威胁检测系统**改善了用户流量侧的威胁检测的深度和能力，安天的**智甲终端防御系统**为用户提供了包括“白名单+安全基线”在内的多种防御策略，安天的**追影威胁分析平台**则为用户提供了通过动静态手段深度分析威胁载荷的能力，安天也在多个行业和部门的**态势感知和通报预警平台**的建设中扮演了关键角色，提供整体的设计支持、开发集成以及供应关键的检测分析能力。

安天将**下一代威胁检测引擎、高定制化深度分析、面向资产和威胁的交互可视分析和知识与情报支撑，作为自身达成有效的、可落地的用户价值的产品基因**。

但安天也客观地看到，面对超级攻击组织的强大能力、坚定意志和难以想象的攻击成本，任何厂商的单打独斗，都难以有效地达成使命，因此安天一直与同业一起，倡导**能力型安全厂商间的积极协作和能力互认**。在此前针对来自南亚次大陆的网络攻击分析应对中，虽然安天将事件命名为“白象”，360 企业安全命名为“摩诃草”，但双方在报告形成中进行了有效地信息互通，以及对对方分析成果的引用互认，这是一个良好的开端，我们相信类似的能力型安全厂商的协作，将会越来越多。

5.3 期待一个更安全的网络世界

当前，超级攻击组织的全环节覆盖能力，已经引发了全球用户‘一切均不可信’的安全焦虑。”去年部分国内媒体对方程式攻击的报道中，将攻击者针对高价值目标节点硬盘固件实现攻击持久化的植入，解读为当前主流的硬盘都带有后门，这固然是一种误解，但也不能不说当一个超级攻击组织的能力强大到了只能猜测和想象的程度时，就不可能不引发恐慌，从而导致对超级大国产生“滥用供应链和信息流优势”的严重质疑。

而近期的方程式攻击代码泄露事件以及此前“ANT”装备体系的曝光，则又使我们看到了相关的 Exploit 储备和攻击思路流入到网络犯罪组织、甚至恐怖主义组织的可能性。鉴于网络攻击技术存在极低的复制成本的特点，当前已经存在严峻的网络军备扩散风险。因此，超级大国能否合理控制自身网络军备发展的速度和规模，并对因自身未有效履行责任而使网络领域发生可能的军备扩散，进行有效地干预和控制，是我们能达成一个更安全的网络世界的关键因素。

我们期待一个更安全的网络世界，我们将为之努力！

附录一：参考资料

[1] Equation Group Cyber Weapons Auction - Invitation

https://github.com/theshadowbrokers/EQGRP-AUCTION

[2] Shadow Brokers reveals list of Servers Hacked by the NSA

http://thehackernews.com/2016/10/nsa-shadow-brokers-hacking.html

[3] 安天：修改硬盘固件的木马 探索方程式（EQUATION）组织的攻击组件

http://www.antiy.com/response/EQUATION_ANTIY_REPORT.html

[4] 安天：方程式（EQUATION）部分组件中的加密技巧分析

http://www.antiy.com/response/Equation_part_of_the_component_analysis_of_cryptographic_techniques.html

[5] Kaspersky：Equation: The Death Star of Malware Galaxy

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

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

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

[7] Kaspersky：Equation Group: from Houston with love

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

[8] Kaspersky：Equation_group_questions_and_answers

https://securelist.com/files/2015/02/Equation_group_questions_and_answers.pdf

[9] SPARC 架构

https://en.wikipedia.org/wiki/SPARC

[10] Solaris 系统

https://en.wikipedia.org/wiki/Solaris_(operating_system)

[11] An Easy Win：Using SIGINT to Learn about New Viruses

[12] 黄晟：关于网络纵深防御的思考

http://www.antiy.com/wtc/2015/02_Joe.pdf

附录二：关于安天

安天从反病毒引擎研发团队起步，目前已发展成为以安天实验室为总部，以企业安全公司、移动安全公司为两翼的集团化安全企业。安天始终坚持以安全保障用户价值为企业信仰，崇尚自主研发创新，在安

全检测引擎、移动安全、网络协议分析还原、动态分析、终端防护、虚拟化安全等方面形成了全能力链布局。安天的监控预警能力覆盖全国、产品与服务辐射多个国家。安天将大数据分析、安全可视化等方面的技术与产品体系有效结合，以海量样本自动化分析平台延展工程师团队作业能力、缩短产品响应周期。结合多年积累的海量安全威胁知识库，综合应用大数据分析、安全可视化等方面经验，推出了应对高级持续性威胁（APT）和面向大规模网络与关键基础设施的态势感知与监控预警解决方案。

全球超过三十家以上的著名安全厂商、IT 厂商选择安天作为检测能力合作伙伴，安天的反病毒引擎得以为全球近十万台网络设备和网络安全设备、近两亿部手机提供安全防护。安天移动检测引擎是全球首个获得 AV-TEST 年度奖项的中国产品。安天技术实力得到行业管理机构、客户和伙伴的认可，安天已连续四届蝉联国家级安全应急支撑单位资质，亦是中国国家信息安全漏洞库六家首批一级支撑单位之一。安天是中国应急响应体系中重要的企业节点，在红色代码、口令蠕虫、震网、破壳、沙虫、方程式等重大安全事件中，安天提供了先发预警、深度分析或系统的解决方案。

安天实验室更多信息请访问： http://www.antiy.com（*中文*）

http://www.antiy.net （*英文*）

安天企业安全公司更多信息请访问： http://www.antiy.cn

安天移动安全公司（AVL TEAM）更多信息请访问： http://www.avlsec.com