Stuxnet 0.5：缺失的环节

非官方中文译本• 安天实验室 译注

文档信息			
原文名称	Stuxnet 0.5:The Missing Link		
原文作者	Geoff McDonald, Liam O Murchu, Stephen Doherty, Eric Chien	**原文发布日期**	2013 年 2 月 26 日
作者简介	Liam O Murchu 是赛门铁克的高级开发经理，管理逆向工程团队，研究最新恶意攻击，分析前沿恶意软件。https://www.linkedin.com/profile/view?id=7067386&authType=NAME_SEARCH&authToken=JYVM&locale Stephen Doherty 是赛门铁克的质量管理经理。https://www.linkedin.com/profile/view?id=2380676&authType=NAME_SEARCH&authToken=lD5n&locale		
原文发布单位	赛门铁克		
原文出处	http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/stuxnet_0_5_the_missing_link.pdf		
译者	安天技术公益翻译组	**校对者**	安天技术公益翻译组
免责声明	• 本译文译者为安天实验室工程师，本文系出自个人兴趣在业余时间所译，本文原文来自互联网的公共方式，译者力图忠于所获得之电子版本进行翻译，但受翻译水平和技术水平所限，不能完全保证译文完全与原文含义一致，同时对所获得原文是否存在臆造、或者是否与其原始版本一致未进行可靠性验证和评价。 • 本译文对应原文所有观点亦不受本译文中任何打字、排版、印刷或翻译错误的影响。译者与安天实验室不对译文及原文中包含或引用的信息的真实性、准确性、可靠性、或完整性提供任何明示或暗示的保证。译者与安天实验室亦对原文和译文的任何内容不承担任何责任。翻译本文的行为不代表译者和安天实验室对原文立场持有任何立场和态度。 • 译者与安天实验室均与原作者与原始发布者没有联系，亦未获得相关的版权授权，鉴于译者及安天实验室出于学习参考之目的翻译本		

	文，而无出版、发售译文等任何商业利益意图，因此亦不对任何可能因此导致的版权问题承担责任。 • 本文为安天内部参考文献，主要用于安天实验室内部进行外语和技术学习使用，亦向中国大陆境内的网络安全领域的研究人士进行有限分享。望尊重译者的劳动和意愿，不得以任何方式修改本译文。译者和安天实验室并未授权任何人士和第三方二次分享本译文，因此第三方对本译文的全部或者部分所做的分享、传播、报道、张贴行为，及所带来的后果与译者和安天实验室无关。本译文亦不得用于任何商业目的，基于上述问题产生的法律责任，译者与安天实验室一律不予承担。

Symantec™
赛门铁克
安全响应

Stuxnet 0.5:
缺失的环节

Geoff McDonald,
Liam O Murchu,
Stephen Doherty,
Eric Chien

版本 1.0：2013年2月26日

目录

概述

2010年，赛门铁克发布关于一种新的、高级复杂的蠕虫的报告，并将其命名为Stuxnet。该蠕虫做为第一个被用作网络武器的计算机软件广为人知。它经专门设计，旨在控制工厂机械并使它们在危险条件下运行或异常运行，从而破坏进程。这在恶意代码史上还是第一次出现。

代码显示了还存在其他版本的蠕虫，它们可以潜在地执行各种行为，制造关于Stuxnet悬而未决的问题。对于缺失环节的等待现在结束了，因为赛门铁克发现了Stuxnet更早的版本，这足以能够回答Stuxnet进化过程中产生的问题。我们对发现的变种进行仔细地解剖和分析，主要发现如下：

- Stuxnet 0.5是被分析的最老的Stuxnet版本，早在2005年11月被开发，并在2007年11月开始盛行。
- Stuxnet 0.5没有Stuxnet 1.x版本那么具有攻击性，且只通过感染Step 7的项目文件进行传播。
- Stuxnet 0.5包含一个选择性攻击策略，关闭位于伊朗纳坦兹的铀浓缩设施的阀门，这将对离心机和铀浓缩系统造成严重的损害。

Stuxnet 0.5成功与否并不明确，但其后的版本是通过不同的框架开发的，更具攻击性；这些新版变种还采用改变离心机速度而不是暗示Stuxnet没有完全履行攻击者目标的策略进行攻击。

Stuxnet存在更多变种，但至今尚未发现。

演变

2007年11月，Stuxnet 0.5被提交到一个恶意软件扫描服务，并早在2005年11月开始执行恶意活动。设计该版本的目的在于在2009年7月4日停止攻击电脑，并阻止在同年1月11日之前的与其C&C服务器的通信。大部分代码中的时间戳看似并不可信，其大致范围在2001年期间。

表 1
Stuxet各版本的演变

版本	日期	说明
0.500	November 3, 2005	C&C 服务器注册
0.500	November 15, 2007	提交到公共扫描服务的日期
0.500	July 4, 2009	感染停止日期
1.001	June 22, 2009	主要的二进制编译时间戳
1.100	March 1, 2010	主要的二进制编译时间戳
1.101	April 14, 2010	主要的二进制编译时间戳
1.x	June 24, 2012	感染停止日期

表2
Stuxnet漏洞的演变

漏洞	0.500	1.001	1.100	1.101	说明
CVE-2010-3888			X	X	任务调度程序EOP
CVE-2010-2743			X	X	LoadKeyboardLayout EOP
CVE-2010-2729		X	X	X	打印服务 RCE
CVE-2008-4250		X	X	X	Windows Server Service RPC RCE
CVE-2012-3015	X	X	X	X	Step7 安全库加载
CVE-2010-2772		X	X	X	WinCC 默认口令
CVE-2010-2568			X	X	Shortcut .lnk RCE
MS09-025		X			NtUserRegisterClassExWow/NtUserMessageCall EOP

根据内部版本号判断，该版本为Stuxnet 0.5，是已知的该蠕虫家族最早的版本。

Stuxnet 0.5复制的唯一方法是感染西门子Step7项目文件。与之后的1.x 版本不同，Stuxnet 0.5没有利用任何微软漏洞。

Stuxnet蠕虫家族各版本的漏洞利用和传播方式有所不同。

表3
Stuxnet复制的演变

复制技术	0.500	1.001	1.100	1.101
Step 7 项目文件	X	X	X	X
通过USB执行 Step 7 项目文件	X			
通过USB实现自启动		X		
通过USB触发CVE-2010-2568			X	X
网络共享		X	X	X
Windows Server RPC		X	X	X
Printer spooler		X	X	X
WinCC 服务器		X	X	X
通过邮槽更新对等网络	X			
通过RPC更新对等网络		X	X	X

Stuxnet 0.5部分基于Flamer平台，而1.x版本主要基于Tilded平台。随着时间的推移，开发者们好像已经更倾向于Tilded平台，他们在后续的版本中使用Tilded平台重新实现Flamer平台。

无论是Flamer平台还是Tilded平台的代码库的不同，都足以说明这其中包含不同的开发者。

Stuxnet 0.5还包含攻击铀浓缩设施阀门系统的代码，但不具备1.x版本修改离心机速度的功能。

安装和加载点

Stuxnet 0.5首先感染了Step7项目存档文件，包含s7hkimdb.dll和XR000001.MDX文件。借助西门子公司SIMATIC产品DLL加载任意代码执行漏洞（CVE-2012-3015），S7hkimdb.dll文件会被执行，随后，该文件解密主XR00001.MDX Stuxnet二进制文件，并将其注入到services.exe进程。Stuxnet正在系统内执行。

一旦注入到services.exe进程中，主Stuxnet二进制文件的副本和一个实现有效载荷的配套DLL文件经过加密后，连同一个MRXCLS.SYS加载点驱动器被保存到磁盘里。主Stuxnet二进制文件自称是一个outbreak.dll，以oem7a.pnf文件名保存在磁盘里。当系统启动时，MRXCLS.SYS加载点驱动器对存储在注册表中的配置数据进行解密，解密主Stuxnet二进制文件并将其注入到Explorer和Step 7进程中。同时有效的DLL也会被解密并注入到Explorer进程。当加载DLL资源时，Stuxnet利用模仿LoadLibrary的模块而不是直接调用LoadLibrary。这一技术的应用很可能是为了躲避安全软件查杀，且没有在Stuxnet 1.x版本中出现过。

PCIBUS.SYS，作为第二个驱动器，在安装的20后天通过创建系统蓝屏强制机器重启。

USBACC11.SYS，作为第三个驱动器，随后被安装。该驱动器与MRXCLS.SYS相似，但它为了执行svchost.exe和Internet Explorer 进程与对等网络和C&C通信而被解密并注入DLL文件。

结构和组织、资源及每一个组件的导出列表详见附录D。

此外，Stuxnet 0.5还检查大量代码路径中的当前日期，并在2009年7月4日之后没有继续传播。如果当前存在安全软件，那么某些模块可能就无法创建或加载。详见附录B。

大量的额外文件被创建，包括日志文件和配置文件。详见附录A。

复制

Stuxnet 0.5通过Step 7项目存档文件进行复制。当可移动驱动器插入到被感染的系统时，Stuxnet 0.5会感染驱动器中具有.s7p或.zip扩展名的Step 7项目存档文件。此外，本次磁盘存储的Step 7项目存档文件也会被感染。

因此，Stuxnet 0.5能够通过可移动驱动器或人为感染Step 7项目存档文件的方式传播到其他机器，例如借助电子邮件传播。

Stuxnet 0.5感染Step 7项目存档文件的方式与Stuxnet 1.x版本相同（详见Step 7 Project File Infections中

W32. Stuxnet Dossier部分）。具体例子如下：

ApiLog/Types—修改触发DLL加载漏洞

XUTILS/links/S7P00001.DBF—配置文件

XUTILS/listen/S7000001.MDX—有效载荷 DLL（installation.dll）

(installation.dll) XUTILS/listen/XR000000.MDX—主Stuxnet二进制文件(outbreak.dll)

hOmSave7/subfolder/s7hkimdb.dll—加载器

C&C

与Stuxnet 1.x版本相似，Stuxnet 0.5的C＆C能力也很有限。尤其是，Stuxnet 0.5不受其作者细粒度的控制。相反地，Stuxnet 0.5只能下载新代码和自我更新。Stuxnet最终需要在没有访问接口的独立的网络上传播，因此它采取独立的设计方式以降低对健全的和细粒度C＆C的需求。Stuxnet 0.5借助一个二次对等机制传播这些代码更新到那些同一级别但无法访问更广互联网的网络上。

C＆C由inetpsp.dll文件实现，而点对点通信则由netsimp32.dll文件实现。这两个文件均由usbacc11.sys驱动器加载，随后被注入到svchost.exe和iexplore.exe进程。

Stuxnet 0.5共有4个C＆C服务器，目前这些服务器有的已不可用，有的是被不相关的人士注册：

- smartclick.org
- best-advertising.net
- internetadvertising4u.com
- ad-marketing.net

有趣的是，设计Stuxnet 0.5的目的在于在2009年1月11日后停止连接C&C服务器，尽管这一威胁的目的仅是在2009年7月4日之后停止继续传播。

图1

Stuxnet C&C 服务器互联网广告机构的主页

C&C服务器的域名创建于2005年，全部显示在互联网广告机构Media Suffix相同的首页，还附有标语“相信心灵能够成就梦想”。

这些服务器托管在美国、加拿大、法国和泰国的商业托管服务提供商手里。而这些域名自2005年起开始应用的事实表明，Stuxnet项目至少开始于7年前。

Stuxnet 0.5利用以下形式发出的第一个请求：

http://<domain>/cgi/link.php?site=xx

这表明一个活跃成功感染的C&C服务器。接下来，Stuxnet 0.5发送下列请求：

http://<domain>/cgi/click.php?xite=xx&num=yy&c=1&j=%x&k=%x&l=%x

如果更新可用，还可能会下载并执行一个文件。

Stuxnet 0.5的最终目标很可能是与互联网隔离的。为使更新到达这些机器，Stuxnet 0.5还借助了点对点机制。只要一个更新版本被引入到网络，例如通过一个被感染的USB密钥，那么处于同一网络上的其他被感染的机器都会收到这些更新或新的代码模块。

Stuxnet 0.5利用Windows邮槽进行点对点通信。邮槽允许进程向远程机器上的个人传递信息。Stuxnet 0.5枚举网络上所有的机器并尝试连接到名为\\[REMOTE MACHINE NAME]\mailslot\svchost的邮槽。随后，它提供一个名为\\[LOCAL MACHINE NAME]\mailslot\imnotify的回调邮槽。

Stuxnet 0.5利用这些邮槽进行点对点通信并传递代码更新。此外，Stuxnet 0.5可以配置系统以允许匿名登录，进而提供下列文件共享：

- temp$
- msagent$
- SYSADMIN$
- WebFiles$

这允许通过对等感染进行文件检索。共享的文件如下：

%WinDir%\msagent\agentsb.dll

%WinDir%\msagent\intl\agt0f2e.dll

%WinDir%\system32\complnd.dll

%WinDir%\system32\dllcache\datacprs.dll

%WinDir%\system32\wbem\perfnws.dll

%WinDir%\Installer\{6F716D8C-398F-11D3-85E1-005004838609}\places.dat

有效载荷

中间人攻击

为了获取指纹目标系统并劫持恶意的可编程逻辑控制器代码，Stuxnet 0.5替换了两个Step 7 DLL文件以便劫持与PLC的通信。

第一个DLL为s7otbxdx.dll，被劫持的目的在于插入恶意的PLC代码。Stuxnet 1.x版本也是用了这种技术（W32.Stuxnet Dossier, Modifying PLCs中对此有所描述）。Stuxnet 0.5会hook几个导出函数并验证CPU为417 PLC而不是315 PLC，除此之外其他的行为基本相同。

第二个DLL为s7aaapix.dll，用于获取指纹目标系统并创建DB8061——指导攻击所需的PLC数据块。导出函数AUTDoVerb被劫持后，恶意的s7otbxdx.dll文件能够通过魔法值（0x91E55A3D, 0x996AB716, 0x4A5CBB03）调用该函数，这样就可以创建或提供之前已经创建好的DB8061数据块执行感染。Stuxnet劫持AUTDoVerb以便监控一切“DOWNLOAD”行为，这表示必须再次获取指纹和创建DB8061才能确保目标系统的正确配置。

获取指纹并生成DB8061

DB8061数据块的生成是一个复杂而又漫长的过程。

通过劫持导出函数，Stuxnet 0.5将获得一个指向最近使用过的数据块的指针（PLC包括代码和数据块）。随后，Stuxnet 0.5会遍历项目结构查找S7项目使用的符号。这些符号是人为设计的表示每一个被PLC控制的设备的标签，大致遵循《管道及仪表流程图》采用的《ANSI/ISA S5.1仪表符号和识别标准》。

Stuxnet 0.5借助这些标签获取指纹并确定每一个设备的地址，便于修改这些设备的行为。

表 4
Stuxnet 0.5钩挂的几个导出函数

Stuxnet v0.500	Stuxnet v1.xxx
s7_event	s7_event
	s7ag _bub_cycl_read_create
	s7ag _bub_read_var
	s7ag _bub_read_var_seg
	s7ag _bub_write_var
	s7ag _bub_write_var_seg
	s7ag _link_in
s7ag _read_szl	s7ag _read_szl
s7ag _test	s7ag _test
s7blk_delete	s7blk_delete
s7blk_findfirst	s7blk_findfirst
s7blk_findnext	s7blk_findnext
s7blk_read	s7blk_read
s7blk_write	s7blk_write
	s7db_close
s7db_open	s7db_open

符号标签解析

目标系统必须为使用417 PLC的SIMATIC400站（0x14109A）或SIMATIC H站（0x141342）。
这些符号标签必须与格式相匹配：

<delimiter><FunctionIdentifier><delimiter><CascadeModule><delimiter><CascadeNumber><DeviceNumber>

例如，在模块A21里、位于级联8，并与离心机160关联的阀门的符号标签为A21-8-160。
各字段定义如下：

定界符

或者是空格(“ ”)，连字符，下划线（ ）（“_”），或者什么都没有。

功能标识符

一个与大致遵循《ANSI/ISA S5.1仪表符号和识别标准》的一系列字符串（见附录C）相匹配的字符串。如果该字符串为PIA（压力指示报警器），那么可以预计它后边有一个一位数。这些字符串代表设备类型（如阀门、换能器或状态灯等）。

级联模块

必须包含字符串“A21”至“A28”。这些字符串与伊朗纳坦兹的级联模块相匹配，公开描述成“A24”、“A26”和“A28”。

级联数

单字符的范围是字母A到R。如果不在这个范围内，那么它会检查是否为00到18之间的两位数。这个两位数字是字母A至R的数字表示。

设备号

这被一种更复杂的、取决于函数标识符确定的设备类型的方式解析，同时使用三种可能的级联排列。映射函数标识符的设备类型在附录C中可见。

设备类型 0

数字串：如果数字长度小于3，那么设备类型则转换至设备类型6。如果数字长度大于或等于3，那么设备类型转换至设备类型7。

设备类型 1、 2、 3

“##”：在1到25之间的一个两位数。

设备类型4、5或 7

设备类型4、5或7有三种不同的格式：

格式1

“####”：解码为两个独立的两位数，分别表示阶段数和该阶段内的离心机数。阶段数必须是1到15之间的整数，与已知的位于纳坦兹的配置相匹配。15个阶段中的每一个阶段，离心机的最大数量见下表。

表 5
阶段数与预期的离心机数量

阶段	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
最大数量	2	2	4	6	8	10	12	16	20	24	20	16	12	8	4

例如，阶段3预计有另一个两位数等于或小于4。这种要求符合一个级联中的离心机排列。

格式 2

“###”：一个必须小于164的三位数，而且是一个级联中的离心机数。

格式3

“##L”：后跟一个字母的两位数。该字母必须在A到D之间，这个数字必须是1到43之间的整数。这种排列将每个阶段下分为四个子群聚。

设备类型 6

“##”：在1到30之间的一个两位数。

设备类型 8、 9、 B或 C

“##”：1到3之间的一个两位数。

设备类型 A

“##<delimiter><string>”：1到3之间的一个有分隔符的可选字符串的两位数。这个字符串必须是字母S开头并且包含字母P。如果这样的字符串存在，那么设备会被修改成设备类型0xB（流量变送器控制器输出，而不是设备类型0xA流量变送器控制器输入）。

基于符号指纹，下表总结了Stuxnet在符号表中查找的设备和标签。

表6
Stuxnet 符号标签解析

设备类型	P&ID功能标识符	每个级联的设备数	
		最少	最多
辅助阀门	{HS, HV, PV, EP}, {ZLO,ZO},{ZLC,ZC}	2	25
离心机阀门	{MVS, RVS, VS}, {MV,RV,SV,YV}	163	164
阶段压力传感器	PT, PCV, PIA#, PIT, PIC, PI, PS	3	30
离心机压力传感器	PT, PCV, PIA#, PIT, PIC, PI, PS	0	164
流量传感器	{FIA}, {FIT}, {FITC},FIC,FT,MFC,MFM}	0	3

符号地址解析

每一个符号标签都有两个相应的地址：过程映像区中的地址和符号表示的设备的直接外围地址。修改这些地址的内存允许PLC控制和读取相关设备的行为。例如，这个值可能表示开启或关闭开关的布尔值，或一个16位的值表示系统当前的温度。这些地址可以是输出（PLC设置值修改设备行为），也可以是输入（PLC读取值判断设备当前的状态）。

设备类型0、1、5和B必须是输出地址，设备类型2、3、4、5、6、7、8、9、A和C必须是输入地址。

V设备类型0、1、2、3、4和5的地址值必须是位值。而设备6、7、8、9、A、B和C的地址值必须是16位值。

级联评级和DB8061生成

解析每一个级联的符号和地址后，代码会检查每一个级联的配置。根据配置的不同，对级联进行评级。具有特定配置的特定设备将会有更高的评级。全部评级完成后，只有六级级联的数据写入到DB8061。

最后，设置标记表示DB8061已经生成。每一次“DOWNLOAD”执行该标记都会被重新设置为0。

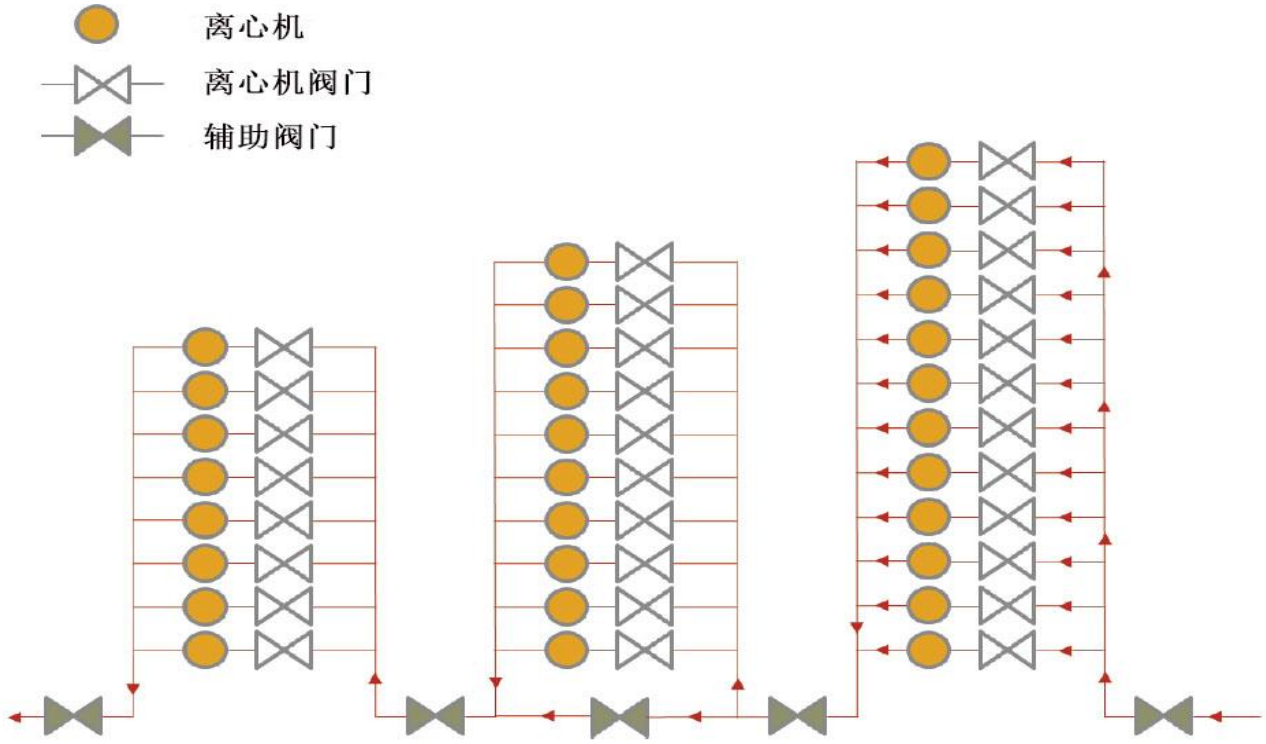
PLC设备攻击代码

代码通过关闭18个级联中的6个顶级级联的阀门执行攻击。其中两个型号的阀门状态被修改：

- 离心机阀门——一组三个阀门（供料、产品和尾线），每个阶段都有一个阀门控制UF6流
- 辅助阀门——控制输入和输出每个阶段（阶段阀门）或全部级联的UF6流的阀门

图 2

一个级联中三个阶段的两种类型阀门的配置



与Stuxnet 1.x版本相似，PLC设备攻击代码包含一个具有八种可能状态的机器：

状态0—等待：完成系统识别，等待铀浓缩过程达到稳定状态，之后进行攻击（大约30天）。

状态1—记录：进行快照和创建虚假输入信息数据块，以备后续使用。

状态2—攻击离心机阀门：开始回放虚假输入信息，关闭大部分离心机上的阀门（最初阶段离心机的阀门除外）。

装态3—读取二级压力：打开处于最后阶段级联的阀门，获取较低压力读数。

状态4—等待改变压力：等待所需的压力变化或时间限制。这一阶段大概需要2小时。

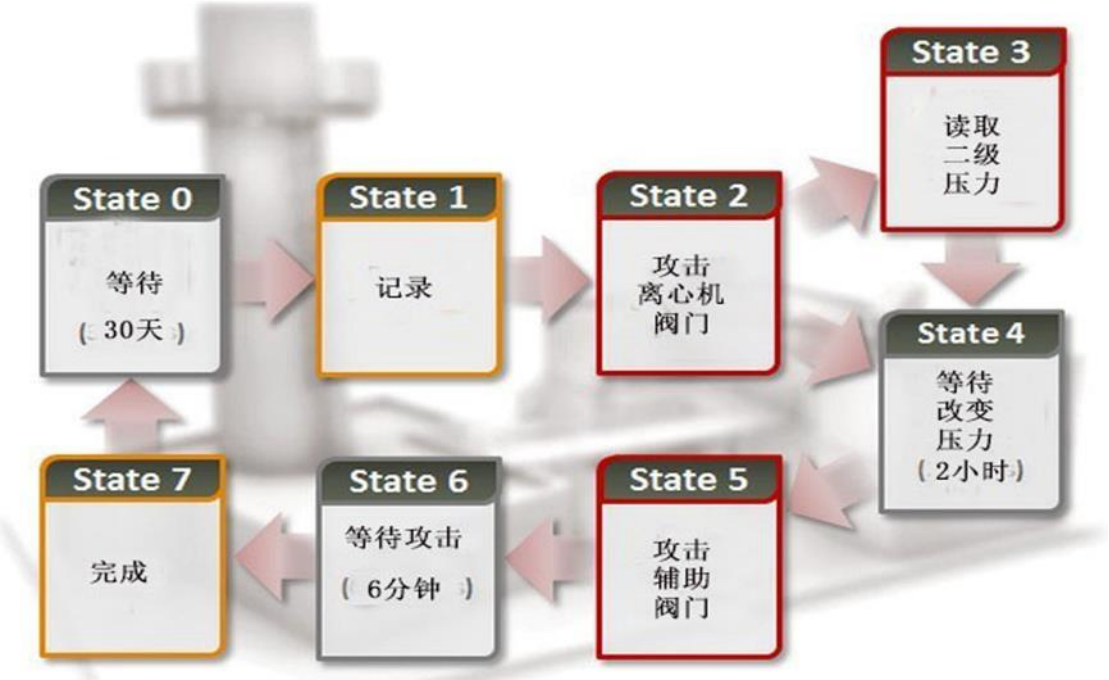
状态5—攻击辅助阀门：打开除了接近阶段1（或阶段10）的所有阀门，此过程需要持续3分钟。

状态6—等待实施攻击：等待6分钟，同时阻止任何状态的变化。

状态7—攻击完成：重启回至状态0。

图 3

417PLC攻击代码状态流程图



状态0：

代码通过监控每一个辅助阀门的状态和运行的时间总量验证系统是否达到稳定状态。

- 在完成300个快照的时间内不允许阀门改变状态。此外，由代码判定是否有最多的离心机阀门处于开启或关闭状态。
- 所有的级联必须运行三天或者更久，或者不可用状态。
- 必须至少有一个级联已经运行35天以上，或所有级联已经运行的总天数超过297天。
- 第一批的21个辅助阀门中的3到7必须在最后两天保持开启状态。

•大部分与辅助阀门相关的压力读数必须在预期的范围内。

只有在这些条件全部符合的条件下，才会进入到状态1阶段。

状态1：

有21个外围I/O值的快照，这些值存储下来用于攻击中重新播放。这样可以组织系统和技术人员实现系统不再按照预期进行运行。

状态2：

首先，正常的运行压力被获取并保存用于稍后的重播。除了供料阶段（阶段10），其他每个阶段都会关闭一定数量的离心机阀门。供料阶段的离心机阀门保持完全开启，而产品和尾线阶段的离心机阀门则保持完全关闭状态。

每个阶段都随机选择关闭特定的离心机阀门。代码随机选择一个开启的离心机阀门，然后按照顺序关闭下一个阀门，直到剩下该阶段的最后一个离心机阀门。如果关闭的阀门总量没有达到预期，代码则继续从该阶段的第一个阀门开始，直到关闭最多的阀门为止。

表7
处理阶段和阀门状态

阶段	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
离心机	2	2	4	6	8	10	12	16	20	24	20	16	12	8	4
待关闭的离心机阀门	2	2	2	4	6	8	10	13	14	0	14	13	10	8	4
关闭的百分比	100%	100%	50%	67%	75%	80%	83%	81%	70%	0%	70%	81%	83%	100%	100%

状态3：

此状态下，在单一的级联里，阶段1的两个离心机阀门都被开启，而且阶段1的阶段阀门很可能也被开启。随后，代码获取此阶段的压力读数。该阶段的压力值应该是比较低的。此值用于后续阶段的重新播放。如果在阶段2没有获取恰当的正常运行压力，状态3会被跳过，使用硬编译的默认值取而代之。

状态4：

状态4等待所需的压力变化或继续状态5前的预定时间限制。只要满足以下任何一个条件，代码就会继续到状态5：

- 阶段10或11的传感器显示的绝对值大于280单位的预期值，且比预期值大5倍。
- 辅助阀门从状态1记录的原始状态被修改的46分钟后，除了很可能是接近产品端的阶段阀门的17号辅助阀门。
- 在攻击开始后的2小时3分钟内没有任何离心机阀门状态发生改变。
- 至少四个离心机阀门的状态从状态1记录的原始状态被修改用时2小时3分钟。

状态5：

在状态5下，除了17、18和20号外，所有的辅助阀门都处于开启状态。在开始状态6 前，代码会等待至少2分53秒。

状态6：

在状态6中，伪造的值被继续重放，任何改变设备值的尝试都会被阻断6分58秒。

状态7：

数据被重置，代码返回到状态0。

关闭除了供料阶段的所有阀门，UF6还可以继续在系统中流动。单单这种行为可能会损坏离心机本身。然而，攻击预期这一压力达到正常运行压力的5倍。在此压力下，铀浓缩系统可能被严重损害，UF6甚至重新凝固。

这种攻击方式是否成功目前还不明确。即使攻击是成功的，攻击者也决定在Stuxnet 1.x版本中使用不同的策略——攻击离心机的速度。

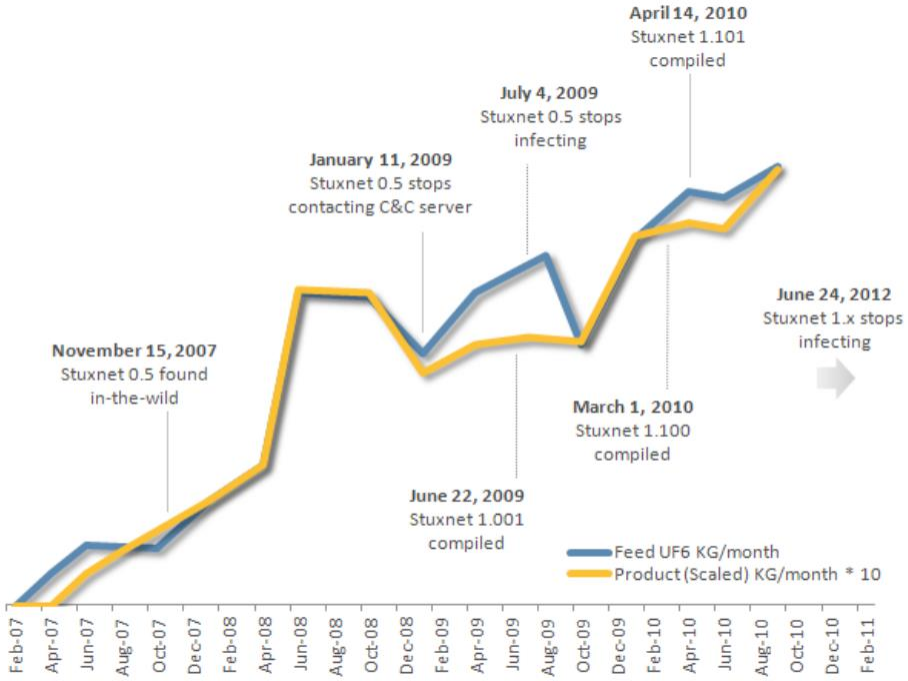
总结

Stuxnet 0.5阐明了Stuxnet蠕虫的演变及历史。很显然，随着时间的推移，Stuxnet变得更具攻击性，而且开发平台也从0.5版本转移到1.x版本。

1.x版本中消失的417攻击代码的主要部分在0.5版本中得到全面的体现。这证明417攻击代码是Stuxnet所用的第一个攻击策略。原始的417攻击代码修改位于伊朗、纳坦兹铀浓缩过程中的阀门状态，给离心机及整个系统造成严重损害。

图 4

低浓缩铀产品 (源自ISIS)



Stuxnet 0.5的攻击成功与否还没有定论。然而，图4中纳坦兹铀浓缩生产却是Stuxnet发展的主要里程碑。有趣的是，生产总量虽然下降但却需要同样或更多的供给量（如图两条曲线之间的距离所示）。

虽然Stuxnet0.5的发现有助于我们进一步全面理解Stuxnet及其目的，但仍未发现其他版本。如果定位了这些版本，那么将可能发现此行动背后的更多信息，但进一步获取其他样本看似是无法实现的。

附录A

以下注册表项为威胁信标：

- HKEY _ LOCAL _ MACHINE\SYSTEM\CurrentControlSet\Services\MRxCls
- HKEY _ LOCAL _ MACHINE\SYSTEM\CurrentControlSet\Services\usbacc11
- HKEY _ USERS\<SID>\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellRecoveryState

以下文件为威胁信标：

- %WinDir%\inf\mdmcpq3.PNF – 加密的installation.dll
- %WinDir%\inf\mdmeric3.PNF – P2P 配置文件
- %WinDir%\inf\oem6C.PNF – 日志文件
- %WinDir%\inf\oem7A.PNF – 主 Stuxnet 组件 (outbreak.dll)
- %WinDir%\inf\oem7F.pnf
- %WinDir%\inf\oem7w.pnf – 加密的 installation.dll
- %WinDir%\inf\~67.tmp – 加密的 installation.dll
- %System%\drivers\mrxcls.sys – 加载点驱动器
- %System%\drivers\usbacc11.sys – C&C服务器模块的加载点驱动器
- %System%\drivers\PCIBUS.SYS – 导致蓝屏的定时器驱动程序
- %System%\comuid.dat
- %System%\netsimp32.dll – P2P 通信
- %System%\inetpsp.dll – C&C 服务器通信
- %System%\perfg009.dat
- %WinDir%\msagent\agentsb.dll
- %WinDir%\msagent\intl\agt0f2e.dll
- %System%\complnd.dll
- %System%\dllcache\datacprs.dll
- %System%\wbem\perfnws.dll
- %WinDir%\Installer\{6F716D8C-398F-11D3-85E1-005004838609}\places.dat
- %System%\dssbase.dat – Log file
- %AllUsersProfile%\Application Data\Microsoft\HTML Help\hhorcslt.dat
- %Temp%/DF419a.tmp
- %WinDir%\help\winmic.fts – Step 7感染的配置文件

附录 B

检查的进程及其相关的安全产品：

- umxagent, Tiny Personal Firewall
- cfgintpr, Tiny Personal Firewall
- umxldra, Tiny Personal Firewall
- amon, Tiny Activity Monitor
- UmxCfg, Tiny Personal Firewall
- UmxPol, Tiny Personal Firewall
- UmxTray, Tiny Personal Firewall
- vsmon, ZoneAlarm Personal Firewall
- zapro, ZoneAlarm Personal Firewall
- zlclient, ZoneAlarm Personal Firewall
- tds-3,TDS3 Trojan Defense Suite
- avp, Kaspersky
- avpcc, Kaspersky
- avpm, Kaspersky
- kavpf, Kaspersky
- kavi, Kaspersky
- safensec,SafenSoft
- snsmcon, SafenSoft
- filemon, Sysinternals Filemon
- regmon, Sysinternals Filemon
- FrameworkService, McAfee
- UpdaterUI, McAfee
- shstat, McAfee
- naPrdMgr, McAfee
- rapapp.exe, Blackice Firewall
- blackice.exe, Blackice Firewall
- blackd.exe, Blackice Firewall
- rcfgsvc.exe
- pfwcfgsurrogate.exe, Tiny Personal Firewall
- pfwadmin.exe, Tiny Personal Firewall
- persfw.exe, Kerio Personal Firewall
- agentw.exe, Kerio Personal Firewall
- agenta.exe, Kerio Personal Firewall
- msascui.exe, Windows Defender
- msmpeng.exe, Windows Defender
- fssm32.exe, F-Secure
- fsgk32st.exe, F-Secure
- fsdfwd.exe, F-Secure
- fsaw.exe, F-Secure
- fsavgui.exe, F-Secure
- fsav32.exe, F-Secure
- fsav.exe, F-Secure
- fsma32.exe, F-Secure
- fsm32.exe, F-Secure
- fsgk32.exe, F-Secure

附录C

功能标识符、设备类型及相关的设备名称一览表。

表 8
功能标识符、设备类型及相关名称

功能标识符	设备类型	设备名称
PT	0	压力传送器
PCV	0	压力控制阀
PIA	0	压力指示警报器
PIT	0	压力指示发送器
PIC	0	压力指示控制器
PI	0	压力指示器
PS	0	压力开关
HS	1	手闸
HV	1	手动阀
PV	1	压力阀
EP	1	电压（测试）点
ZLO	2	指示灯状态开启 (状态指示灯)
ZO	2	状态开启
ZLC	3	指示灯状态关闭
ZC	3	状态关闭
MVS	4	手动阀开关
RVS	4	安全阀开关
VS	4	阀门开关
SHS	4	高频开关
MV	5	手动阀
RV	5	安全阀开关
SV	5	频率控制阀
YV	5	阀门状态指示器
FIA	8	流量指示警报器
FITC	A	流量指示器发送器控制器
FIT	9	流量指示发送器
FIC	C	流量指示控制器
FT	C	流量发送器
MFC	C	质量流量控制器
MFM	C	质量流量计

附录D

Stuxnet 0.5组件的组织架构及每个导出函数的行为。

图 5

Stuxnet 0.5组件的组织架构。

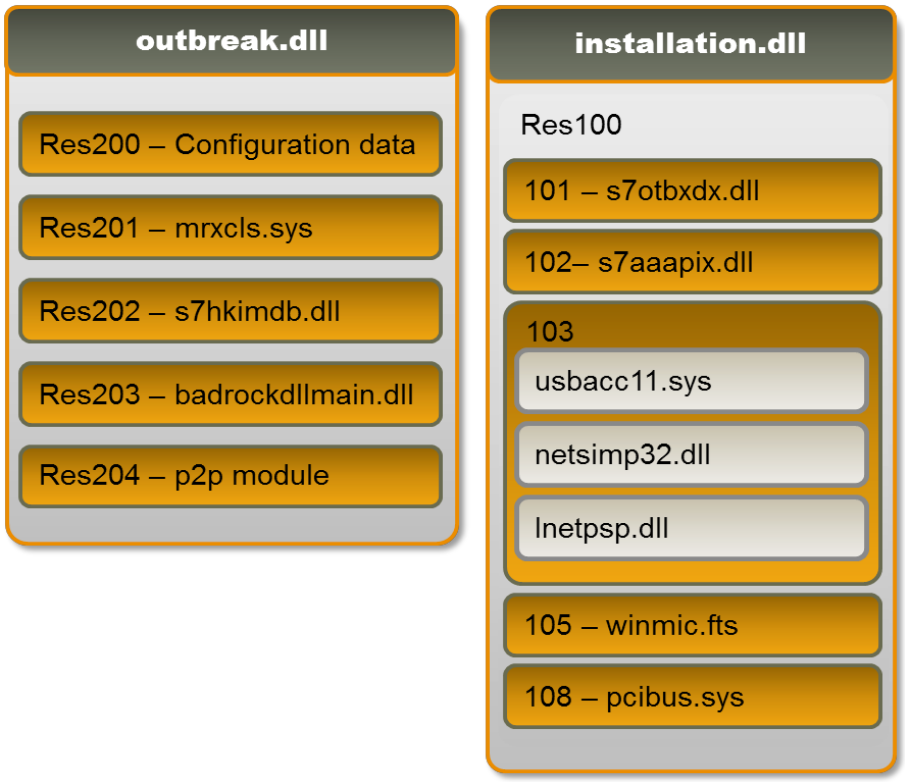


表 9

Outbreak.dll的有效载荷导出函数

有效载荷导出函数	说明
Export 1	通过插入可移动驱动器感染Step 7项目
Export 2	为执行 Step 7 项目感染 hook Step 7
Export 4	卸载程序
Export 5	验证安装
Export 6	返回版本号
Export 7	加载点对点通信数据文件
Export 8、 9和10	从被感染的 Step 7 项目文件中更新 Stuxnet
Export 11	向 services.exe 中注入模块
Export 12	安装程序
Export 13	调用 Export 1

相关资料

W32.Duqu

http://www.symantec.com/security_response/writeup.jsp?docid=2011-101814-1119-99

W32.Flamer

http://www.symantec.com/security_response/writeup.jsp?docid=2012-052811-0308-99

W32.Stuxnet

http://www.symantec.com/security_response/writeup.jsp?docid=2010-071400-3123-99

Multiple Siemens SIMATIC Products DLL Loading Arbitrary Code Execution Vulnerability (CVE-2012-3015)

http://www.securityfocus.com/bid/54651

Stuxnet 0.5: The Missing Link

http://www.symantec.com/connect/blogs/stuxnet-05-missing-link

Stuxnet 0.5: Disrupting Uranium Processing At Natanz

http://www.symantec.com/connect/blogs/stuxnet-05-disrupting-uranium-processing-natanz

Stuxnet 0.5: How it Evolved

http://www.symantec.com/connect/blogs/stuxnet-05-how-it-evolved

Stuxnet 0.5: Command-and-Control Capabilities

http://www.symantec.com/connect/blogs/stuxnet-05-command-and-control-capabilities