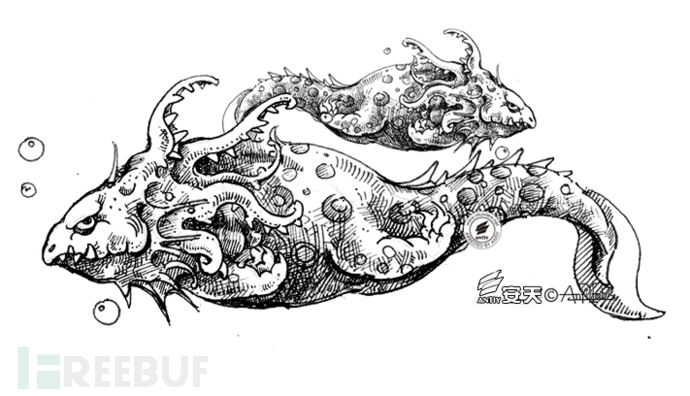
**针对工控恶意代码TRISIS的技术分析（安天）**

[antiylab](https://www.freebuf.com/author/antiylab)[https://image.3001.net/images/index/f2.png](http://www.freebuf.com/bufer)2018-09-14共106967人围观 ，发现 1 个不明物体[工控安全](https://www.freebuf.com/articles/ics-articles)

**前言**

**2017年8月，安天安全研究与应急处理中心（安天CERT）基于综合情报研判，将针对工业控制系统的恶意代码TRISIS（又名TRITON、HATMAN）列为需要重点分析关注的威胁，并将其命名为“海渊”。该恶意代码在中东某石油天然气厂的工业控制系统中被国外安全研究人员发现，根据各方信息判断，由于攻击者准备不充分，尚未对人员及财产造成重大损失。“海渊”（TRISIS）的目标为施耐德电气公司的安全仪表系统，通过植入固件更改最终控制元件的逻辑以达到攻击目的。其通过Tricon安全仪表系统所使用的TriStation通信协议进行攻击，因此运行此协议的所有安全控制器都可能受到影响。**

[](https://image.3001.net/images/20180911/1536667491_5b97af638bf85.png)

**一、概述**

“海渊”（TRISIS）所攻击的目标是工业控制系统（ICS）中的安全仪表系统（SIS）控制器，其主要瞄准施耐德电气的Tricon安全仪表系统，从而达到在最终控制元件中替换逻辑的目的。安全仪表系统（Safety Instrumented System），简称SIS，又称为安全联锁系统（Safety Interlocking System），主要为工厂控制系统中报警和联锁部分，对控制系统中检测的结果实施报警动作或调节或停机控制，是工厂企业自动控制中的重要组成部分。其包括传感器、逻辑运算器和最终执行元件，即检测单元、控制单元和执行单元。SIS系统可以监测生产过程中出现的或者潜伏的危险，发出告警信息或直接执行预定程序，立即进入操作，以防止事故的发生，同时降低事故带来的危害及影响。

安天CERT针对该恶意代码的攻击原理及样本展开了技术分析，发现该恶意代码的攻击流程为利用社工技巧伪装成安全仪表系统的日志软件进入目标网络，之后通过特殊ping包发现安全仪表系统，在确定安全仪表系统可被入侵后，会上传组合后的二进制代码，以改变安全仪表系统的梯形图（即安全仪表系统逻辑），一旦攻击成功，将有可能对工业生产设备、工厂人身安全造成巨大危害，对关键信息基础设施安全、社会安全造成巨大影响。

**二、“海渊”（TRISIS）攻击原理**

**2.1 攻击原理简述**

“海渊”（TRISIS）和“震网”（Stuxnet）、“Industroyer/CrashOverride”等恶意代码一样具备从工业控制系统中发现特定目标装置的能力。但同时，其更进一步具有直接交互、远程控制和危害安全系统的能力。

“海渊”（TRISIS）采用Python脚本进行编写，并使用Py2EXE伪编译为PE可执行程序，以便于在未安装Python的系统环境下执行。攻击者充分了解安全仪表系统处理过程及环境的具体细节，才能构造有效载荷利用“海渊”（TRISIS）进行攻击，其主要通过修改和部署新的PLC（可编程逻辑控制器）梯形图，以对目标产生预期的影响。

“海渊”（TRISIS）可以改变目标安全仪表的基本逻辑，通过可执行文件将其目标作为在执行时传递给它的命令行参数，其核心功能是通过将四个二进制文件组合到目标上进行工作，其中植入Python脚本中的两个二进制载荷，主要用于准备和加载包含替换逻辑的外部模块（参见图2-1、图2-2）；附加的两个外部的二进制文件，由脚本中的名称专门引用，但位于单独的文件中，其中imain.bin为主要功能载荷（参见图2-3）。

[](https://image.3001.net/images/20180911/1536667516_5b97af7cb6214.png)

图2-1 植入脚本中的二进制载荷

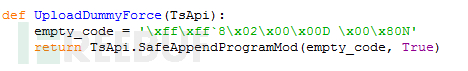
[](https://image.3001.net/images/20180911/1536667530_5b97af8a16b55.png)

图2-2 脚本中用于覆盖核心载荷的傀儡程序

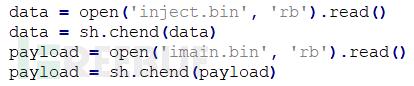
[](https://image.3001.net/images/20180911/1536667542_5b97af965db12.png)

图2-3 获取外部独立二进制文件

**2.2 攻击流程分析**

trilog.exe是“海渊”（TRISIS）的主程序，其原本是Tricon安全仪表系统中的一个模块程序，攻击者利用其进行实施攻击的主要流程参见图2-4：

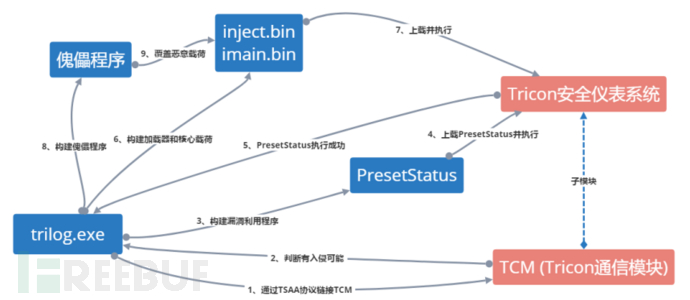
[](https://image.3001.net/images/20180911/1536667554_5b97afa2dbcda.png)

图2-4 TRISIS攻击流程

攻击流程说明：

trilog.exe通过TSAA协议链接TCM（Tricon通信模块），识别并获得一个能够与安全仪表系统通信的系统，并判断是否满足入侵的条件；

确认可入侵后，识别目标安全仪表系统类型，并用替换逻辑和加载器开发“海渊”（TRISIS）功能代码，构建漏洞利用程序PresentStatus；

上载PresentStatus到Tricon安全仪表系统中，并执行确保“海渊”（TRISIS）在预期环境下工作；

构建加载器和核心载荷inject.bin、imain.bin，将“海渊”（TRISIS）传输到包含装载器模块的目标上；

“海渊”（TRISIS）可执行文件运行时，伪装成用于分析日志的软件，利用嵌入的二进制文件来识别控制器上存储器中的适当位置以进行逻辑替换，并上传“初始化代码”（4字节序列）；

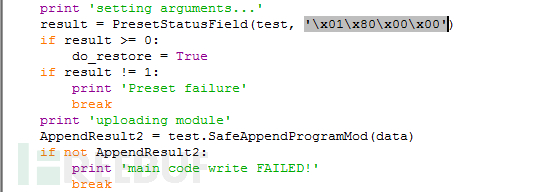
[](https://image.3001.net/images/20180911/1536667565_5b97afada05aa.png)

图2-5 上传初始化代码

验证前一步是否成功，然后上传新的PLC梯形图到安全仪表系统；

上传傀儡程序覆盖核心载荷。

[上传傀儡程序](https://image.3001.net/images/20180911/1536667576_5b97afb8a4e1c.png)

图2-6 上传傀儡程序

**三、”海渊”（TRISIS）样本分析**

**3.1 “海渊”（TRISIS）与Triconex的通信过程分析**

“海渊”（TRISIS）与Triconex的通信主要依赖于TsHi、TsBase、TsLow、TS\_cnames等模块，这些模块提供了极为强大的远程连接控制Triconex的代码。

**请求连接**

“海渊”（TRISIS）通过在Script\_test.py中调用TsLow中的connect函数进行连接，我们以此作为入口点，对其协议进行简单分析。

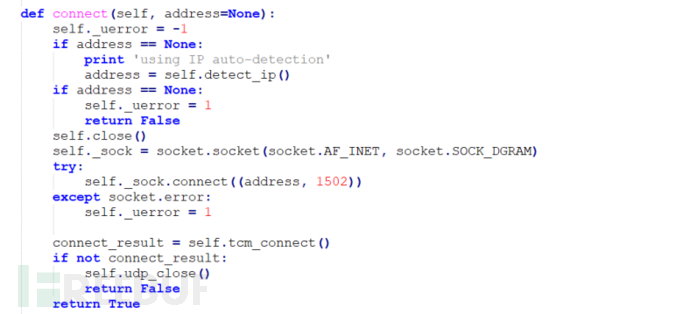
[](https://image.3001.net/images/20180911/1536667590_5b97afc679aec.png)

图3-1 connect函数

在connect函数中detect\_ip和tcm\_connect为关键函数。

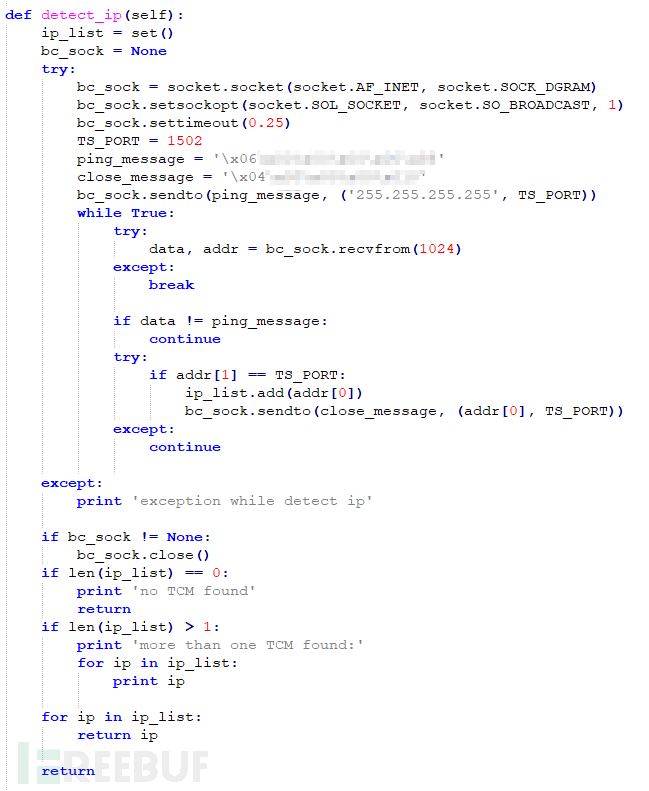
[](https://image.3001.net/images/20180911/1536667608_5b97afd808403.png)

图3-2 detect\_ip函数

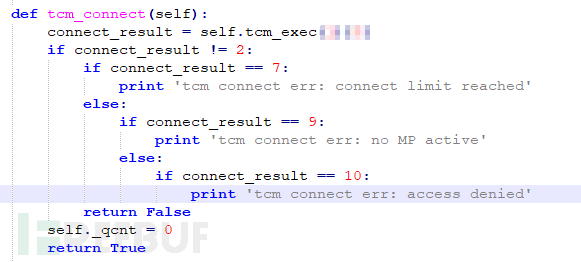
[](https://image.3001.net/images/20180911/1536667710_5b97b03e331a2.png)

图3-3 tcm\_connect函数

在detect\_ip函数中使用1502端口，用变量TS\_PORT定义；另外，对ping数据包和close数据包定义，分别采用0×06和0×04为两者标识码，如下图3-4。

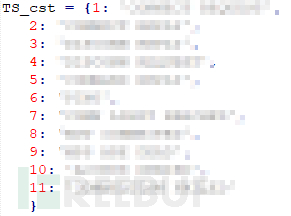
[](https://image.3001.net/images/20180911/1536667724_5b97b04c4a50d.png)

图3-4 数据包类型标识码定义

在tcm\_connect函数中connect\_result同样为数据包类型标识码，具体定义见上图3-4，其中关键函数为tcm\_exec（type参数的值为1）。

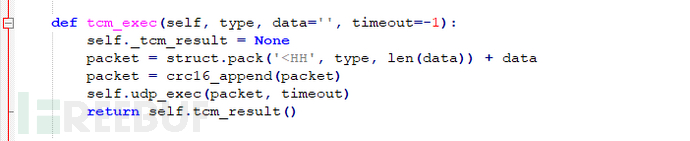
[](https://image.3001.net/images/20180911/1536667738_5b97b05a397ec.png)

图3-5 tcm\_exec函数

Struct.pack函数作用为按照给定的格式(fmt)，把数据转换成字符串(字节流)，并将该字符串返回，即packet是把type和data长度按照16进制解释，再加上data和crc16校验的数据包。Struct.pack数据包结构如下：

[](https://image.3001.net/images/20180911/1536667761_5b97b071e66e9.png)

**上传PresetStatus**

“海渊”（TRISIS）通过使用SafeAppendProgramMod函数上传PresetStatus。

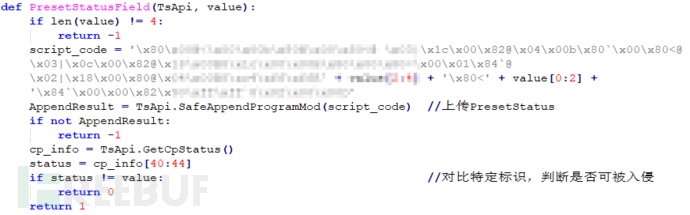
[](https://image.3001.net/images/20180911/1536667777_5b97b0810ca20.png)

图3-6 PresetStatusField函数

script\_code为需要上传的执行代码。该函数为对AppendProgramMin函数的封装，AppendProgramMin函数调用WriteProgram函数（对WriteFunctionOrProgram的封装）。

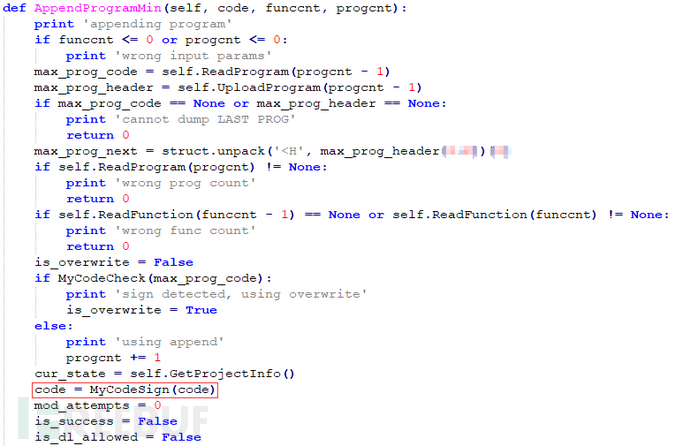
[](https://image.3001.net/images/20180911/1536667793_5b97b0910949f.png)

图3-7 AppendProgramMin函数（部分）

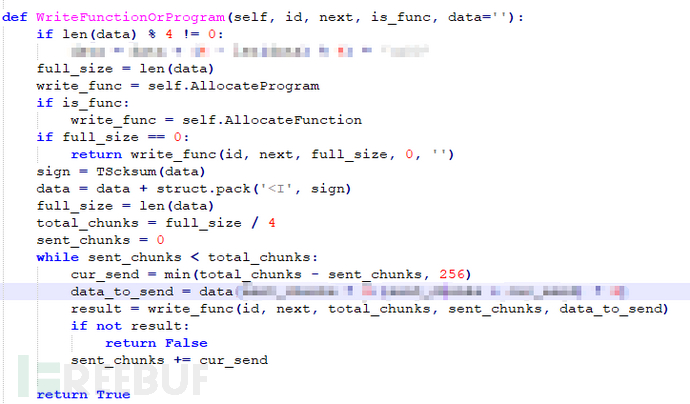
[](https://image.3001.net/images/20180911/1536667812_5b97b0a4969cd.png)

图3-8 WriteFunctionOrProgram函数

AppendProgramMin函数在代码尾部加上CRC校验：

[MyCodeSign函数](https://image.3001.net/images/20180911/1536667831_5b97b0b738cba.png)

图3-9 MyCodeSign函数

在WriteFunctionOrProgram函数中调用AllocateProgram函数向仪表写入程序：

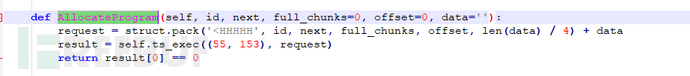
[](https://image.3001.net/images/20180911/1536667851_5b97b0cb7d97f.png)

图3-10 AllocateProgram函数

此时对数据包进行一次封装，现在数据包为：

[https://image.3001.net/images/20180911/1536667874_5b97b0e25efed.png!small](https://image.3001.net/images/20180911/1536667874_5b97b0e25efed.png)

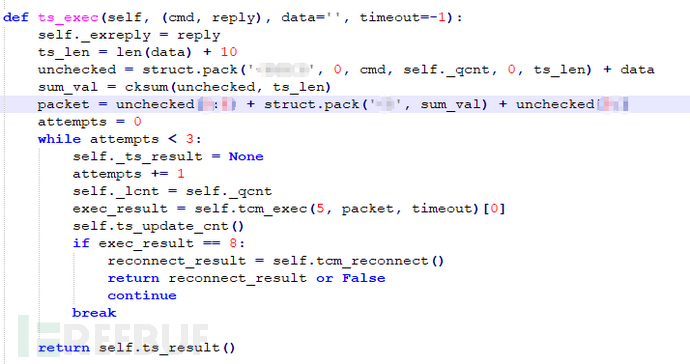
[](https://image.3001.net/images/20180911/1536667895_5b97b0f7c4e05.png)

图3-11 Ts\_exec函数

Ts\_exec函数负责一层数据包封装，此时数据包为：

[](https://image.3001.net/images/20180911/1536667924_5b97b11420529.png)

在Ts\_exec函数中调用tcm\_exec函数（参见图3-5），此时数据包为：

[](https://image.3001.net/images/20180911/1536667941_5b97b125ae98b.png)

Tcm\_exec调用udp\_exec函数，最终调用sock.send函数通信。

**上传有效载荷**

上传有效载荷与上传PresetStatus流程相同，在此不再赘述，有效载荷的数据包结构为：

[](https://image.3001.net/images/20180911/1536667964_5b97b13c62be8.png)

**3.2 模块分析**

“海渊”（TRISIS）构建了精简的TriStation通信框架，框架包含模块TsHi.py、TsBase.py、TsLow.py、TS\_cnames.py。除框架外，“海渊”（TRISIS）还包含一个Script\_test.py脚本，此脚本使用TriStation通信框架连接到Tricon，并注入载荷。

**Script\_test.py分析**

Script\_test.py是使“海渊”（TRISIS）真正实现功能的模块，Script\_test.py文件小巧，其通信功能的实现主要依赖TriStation协议支持库。由于TriStation协议目前为止仍是闭源协议，这些支持库极有可能为“海渊”（TRISIS）背后攻击者花费精力逆向得来，如果要对TriStation协议进行深入了解，对涉及的库文件进行分析是很好的选择。

Script\_test.py与通信有关代码主要实现了代码上传、寻找仪表系统和利用库函数进行通信的功能（参见图3-12、3-13、3-14）。

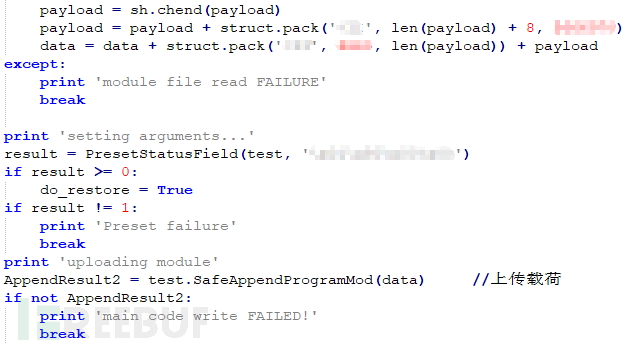
[](https://image.3001.net/images/20180911/1536667986_5b97b15262609.png)

图3-12 代码上传

[](https://image.3001.net/images/20180911/1536668002_5b97b1628fa65.png)

图3-13 寻找仪表系统

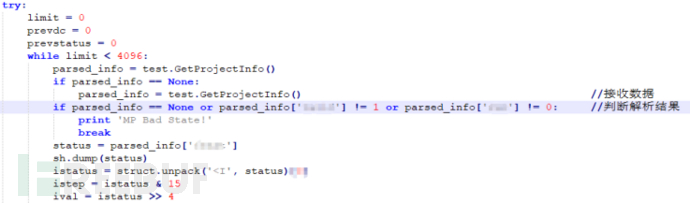
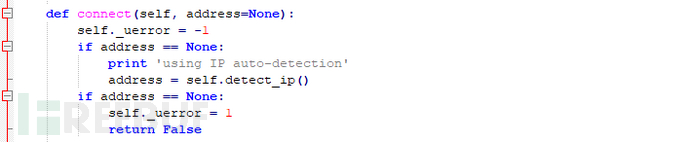
[](https://image.3001.net/images/20180911/1536668029_5b97b17d7de0f.png)

图3-14 利用库函数进行通信

攻击步骤说明：

Script\_test.py脚本首先尝试连接Tricon仪表系统，当不带参数启动时，直接广播搜索可识别TriStation协议的设备——Tricon仪表系统：

[](https://image.3001.net/images/20180911/1536668062_5b97b19ee97fb.png)

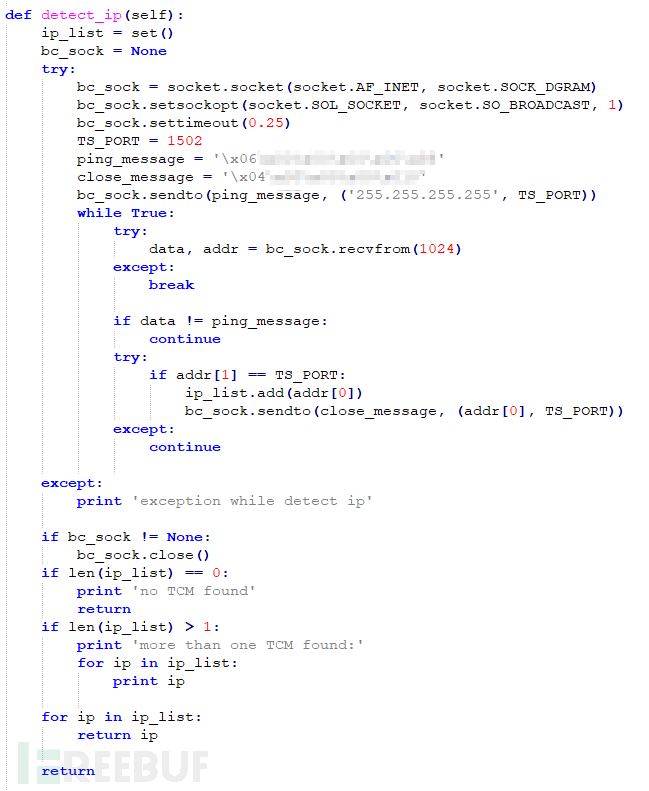
[](https://image.3001.net/images/20180911/1536668072_5b97b1a8885a8.png)

图3-15 广播搜索Tricon仪表系统

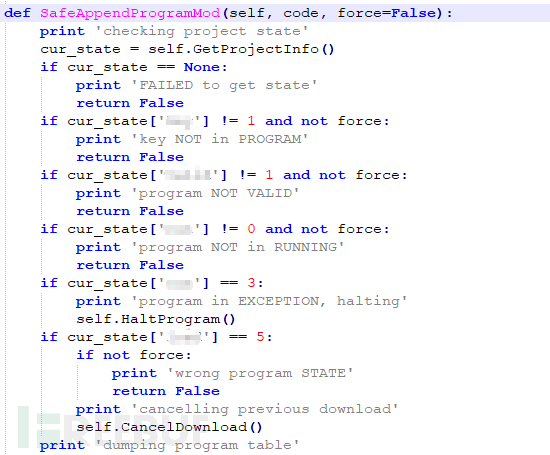
一旦寻找到目标，将上传PresetStatus程序确定此目标是否可被利用；确定可被利用后，上传inject.bin和imain.bin两个主要载荷，篡改图表，以达到破坏或监控目的；最后，上传无用代码覆盖载荷消除痕迹。

**TsHi.py分析**

TsHi.py是框架的高级接口，允许读写函数和程序，以及检索项目信息和与植入有效负载的交互（如下所述），其包括SafeAppendProgramMod函数，该函数可获取程序表，读取程序和函数，并将提供的shellcode附加到现有的控制程序，它还在必要时处理CRC32校验和。

[](https://image.3001.net/images/20180911/1536668199_5b97b22717e0f.png)

从函数名我们可以轻易猜出每个函数的作用，“海渊”（TRISIS）只使用了其中的SafeAppendProgramMod函数来上传其载荷。

[](https://image.3001.net/images/20180911/1536668249_5b97b2593aa97.png)

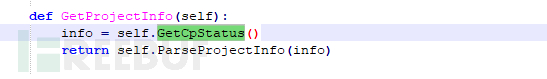
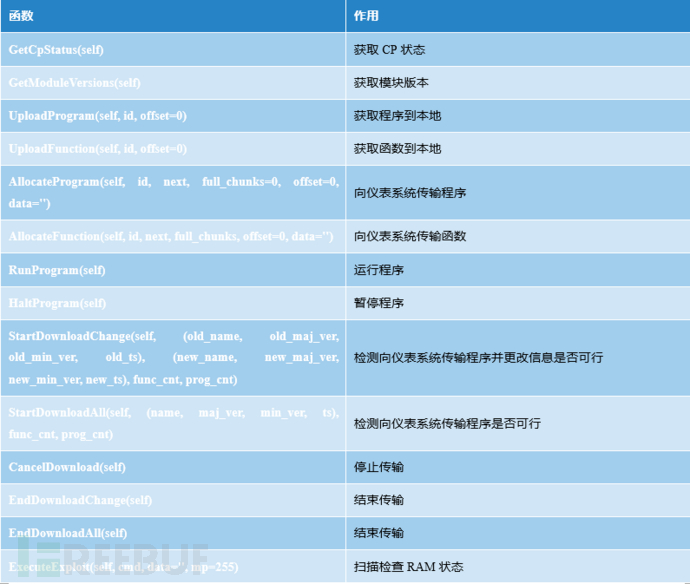
[](https://image.3001.net/images/20180911/1536668249_5b97b25968be2.png)

图3-16 SafeAppendProgramMod函数检查目标状态

之后，此函数获取目标系统中已上传的程序列表及函数数量。最后用AppendProgramMin函数上传载荷，并执行。

**TsBase.py分析**

TsBase.py主要充当高级接口和低级TriStation功能代码之间的转换层，以及用于上载和下载程序或获取控制程序状态和模块版本等功能的数据格式：

[](https://image.3001.net/images/20180911/1536668267_5b97b26bcaa6f.png)

**TsLow.py分析**

TsLow.py可实现将上层制作的TriStation数据包通过UDP发送到Tricon通信模块（TCM）的功能的最底层，还包括通过向1502端口发送UDP“ping”广播消息来自动发现Tricon控制器。

[](https://image.3001.net/images/20180911/1536668280_5b97b278ee4ab.png)

**TS\_cnames.py分析**

TS\_cnames.py包含TriStation协议功能和响应代码以及关键开关和控制程序状态的命名查找常量。

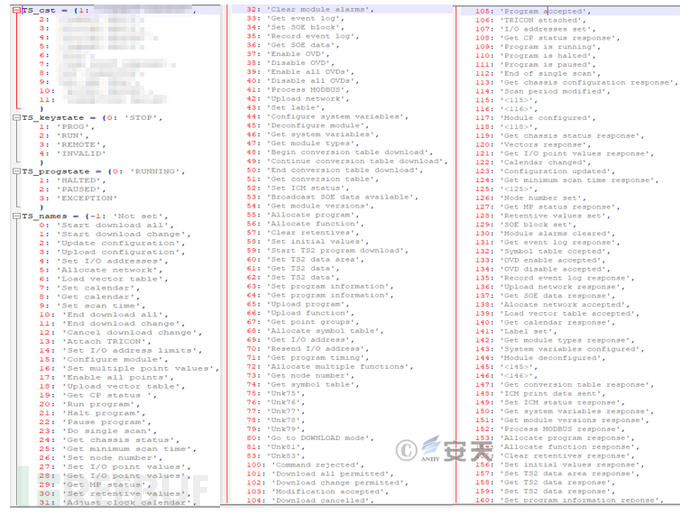
[](https://image.3001.net/images/20180911/1536668296_5b97b288283ab.png)

图3-17 状态码截图（部分）

**四、结语**

“海渊”（TRISIS）恶意代码呈现出了一些值得关注的特点，其开发者深入了解相关工控产品的控制协议，除了上载到PLC中的二进制模块外，其他框架和功能代码全部采用脚本编写，非常容易被改造和加工。而其打击点则在作为工业控制系统的生产安全监测单元的SIS上。

作为针对工业系统攻击的恶意代码，“海渊”（TRISIS）很自然的会被与“震网”（Stuxnet）和“乌克兰停电”等攻击关键基础工业设施的事件相比较。

与“震网”庞大的恶意代码工程相比，“海渊”（TRISIS）看起来相对简单。震网攻击对于离心机整体控制机制的介入是极为深入的，这本身也源自铀离心工艺处理的复杂性，基于攻击者所要达成的复杂的攻击目的（铀无法达到武器级要求、大量损毁离心机）、攻击的隐蔽性攻击和攻击需要达成的阶段持续性。震网是一个支撑完整战役过程的恶意代码。相比之下，尽管“海渊”（TRISIS）小巧的令人可怕，但其更像一个灵巧的“战斗部”，其在攻击行动中，可能是与其他的攻击植入手段和恶意代码配合使用的。对“海渊”（TRISIS）的编写者来说，其核心资源和成本消耗，主要是对SIS系统达成深入分析了解。

“海渊”（TRISIS）的攻击方式与乌克兰电网遭遇攻击停电事件的明显差异是“海渊”（TRISIS）攻击的位置更加纵深。乌克兰停电的攻击效果是通过直接在SCADA控制界面上拉闸达成的，粗暴而有效，其并不依赖于深度解析和篡改控制指令。尽管乌克兰停电事件中，攻击者也篡改了远程变电站串口以太网关中的固件，但这一操作目的是为了导致已经被“拉闸”的远程变电站不能被远程合闸恢复。而“海渊”（TRISIS）的打击点，则是为PLC重置新的逻辑，而且其攻击的是安全仪表系统。

从防御工作来看，由于“海渊”（TRISIS）以通过伪装为SIS的日志软件获得被执行的机会，因此重要的防御点即在对软件供应链的管控上。应在采购阶段，严格落实供应链的安全管控，从源头遏制危害。在工业系统的运维中，针对工控系统环境的新设备安装上线、软件的发布升级、运维手段的接入等，都应进行全面的前置检查和移动介质接入管控。

与“震网”、“乌克兰停电”事件类似的是，“海渊”（TRISIS）攻击依然是以获得关键PC节点为攻击入口的，这一特点是具有普遍性的。而一旦关键PC节点沦陷，攻击已经针对生产系统实施了纵深影响，则极难防御。对于工业基础设施来说，做好生产网络和办公网络中的PC端点防御是一个必须做好的基础性工作，对于重要PC节点必须形成严格的依托白名单的主动防御机制。

从现状来看，大部分工业控制系统对效率性能的考虑远多于安全考虑，而安全考量中，更多依然是以传统的应对事故视角，而非应对攻击视角。做好工业系统的安全防御工作，必须按照三同步的原则进行，在系统规划、建设、运维的全周期考虑网络安全问题。这是一个复杂和系统的工作，在可管理网络的基础上，建设可防御的网络，推动从基础结构安全、纵深防御、态势感知与积极防御到威胁情报的整体叠加演进。这个过程需要大量基础扎实的工作和预算投入。对已有系统的安全改造，因为涉及到生产业务的连续性、稳定性，可能牵扯到更多的问题。

关于对工业系统的安全问题和防御，安天在“震网”、“乌克兰停电”等事件的分析中，已经有过很多的探讨，我们会为用户提供更系统的建议和解决方案。

**五、参考资料**

[1] Dragos ：TRISIS Malware——Analysis of Safety System Targeted Malware：<https://dragos.com/blog/trisis/TRISIS-01.pdf>

[2] Ics-cert :MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)：<https://ics-cert.us-cert.gov/sites/default/files/documents/MAR-17-352-01>HatMan – Safety System Targeted Malware (Update A)\_S508C.PDF

[3]《乌克兰电力系统遭受攻击事件综合分析报告》：<http://www.antiy.com/response/A_Comprehensive_Analysis_Report_on_Ukraine_Power_Grid_Outage/A_Comprehensive_Analysis_Report_on_Ukraine_Power_Grid_Outage.html>

[4]《对Stuxnet蠕虫攻击工业控制系统事件的综合分析报告》：<http://www.antiy.com/response/stuxnet/Report_on_the_Worm_Stuxnet_Attack.html>

**六、HASH**

[](https://image.3001.net/images/20180911/1536668306_5b97b292b5120.png)

**\*本文作者：安天实验室，转载请注明来自FreeBuf.COM**