基于 Snort 的 PROFINET 入侵检测系统设计

Design of PROFINET Intrusion Detection System Based on Snort

刘红阳 （北方工业大学现场总线技术及自动化重点实验室，北京 100144）

摘要：近年来工业化与信息化的不断融合使工厂的管理更加透明化，企业的效率和效益也得到很大提升。随着工控行业信息化建设的不断推进，产生的工控安全问题也愈发突出。以 Snort 入侵检测系统和 PROFINET 协议为研究对象，设计了一款预处理器插件扩展到原有 Snort 系统中。实现了 Snort 对 PROFINET 协议的解析与识别，并通过设计模拟攻击实验验证了改进后的 Snort 系统能够对异常 PROFINET 协议进行实时检测与报警，从而保护了 PROFINET IO 通信系统的安全。

关键词：入侵检测；Snort；PROFINET IO；预处理器

Abstract:This paper takes Snort intrusion detection system and PROFINET protocol as the research object,and designs a preprocessor plugin to extend into the original Snort system.The Snort analysis and identification of the PROFINET protocol is realized,and the simulated Snaut system can verify the real-time detection and alarm of the abnormal PROFINET protocol by designing the simulation attack experiment,thus protecting the security of the PROFINET IO communication system.

Keywords:intrusion detection,Snort,PROFINET IO,preprocessor

目前，工业防火墙是工控安全领域常用的一种有效网络防护方法，但它属于被动防御，通常对于来自内部的攻击无能为力。而入侵检测技术作为一种主动的信息安全保障技术，有效地弥补了防火墙等传统安全防护技术的缺陷[1]。

Snort 作为一款免费的轻量级开源入侵检测系统，能够实时捕获网络流量，并对协议进行分析，对捕获的非法流量和可疑数据的特征与规则库匹配，并记录到日志文件并实时报警。通过对 Snort 系统架构和处理数据包流程的分析可知，Snort 能够调用不同的解码函数对捕获数据包的协议类型进行解析，从而根据不同的检测规则对不同类型的异常协议进行识别与报警。但对于一些特殊的工业通信协议，例如基于数据链路层的 PROFINET 协议，Snort 还无法对其解析与识别。ROFINET 协议同其他工业通信协议一样存在缺乏安全认证和授权保护等安全隐患[2]。为了使 Snort 实现对 PROFINET IO 通信系统的保护，本文设计了一款预处理器扩展到 Snort 系统中，Snort 实现了对 PROFINET 协议的解析与识别。

1 Snort 预处理器设计与实现

本文对 Snort 系统的功能扩展主要是通过设计的预处理器插件实现的，下面对其实现过程进行介绍。

在 Snort 源码的 templates 文件中存放着用于开发预处理器的模板 spp_template.c 和 spp_template.h。将 spp_template.c 拷贝到 preprocess 文件中，并重命名为 spp_profinet.c。通常情况下，新增加的预处理器源文件（即 spp_profinet.c）主要由以下三个函数结构组成[3]：

1）插件安装函数：一般是在 Snort 进行初始化时启动，主要作用是调用注册预处理器插件函数，本文中将该函数命名为 SetupProfinNet()，并将设计的预处理器名称命名为“Profinet”；

2）插件初始化函数：此函数主要作用是完成预处理器的初始化和注册预处理器功能执行函数等，本文将该函数命名为 SetupProfiNetInit()；

3）功能执行函数：该函数是整个 PROFINET 预处理器工作的核心。将其命名为 DetectProfiNetPackets()，主要功能是对 PROFINET DCP 协议和 RT 协议进行解析并存储。PROFINET 预处理器程序从启动到报文解析结束的具体工作流程如图 1 所示。

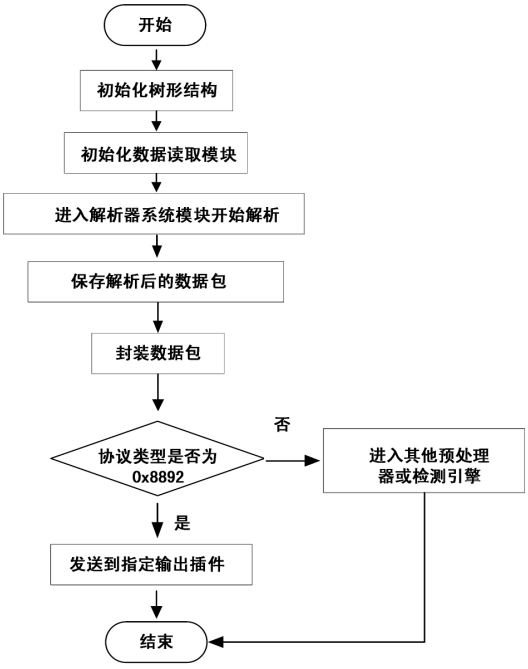

图 1 预处理器功能执行流程

由图 1 可知，本文设计的预处理功能的执行主要是通过三个模块来实现的：解析器系统模块、数据读取模块和数据存储模块。

（1）解析器系统模块

在这一模块，本文根据 PROFINET 协议类型标志 0x8892 和 PROFINET RT、DCP 协议帧结构中 Frame ID 字段的取值范围，设计了一个包含五种具有特定解析功能的解析器系统。解析器系统以及内部关系结构如图 2 所示。

解析器系统的设计是基于三种结构体来实现的。为了更清楚地阐述各个特定解析器之间的调用关系，把定义的五种解析器（结构体）暂时称为特定解析器，把提供实际解析函数入口的结构体称为基本解析器，把能够注册和获取特定解析器的结构体称为解析器“寄存器”。在每个特定解析器结构体内部都包含着一个基本解析器结构体，基本解析器结构体内部又包含一个解析器“寄存器”结构体。通过这种结构体嵌套的方法，就使得每种特定解析器在能够解析不同协议字段的同时还能够调用下一

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

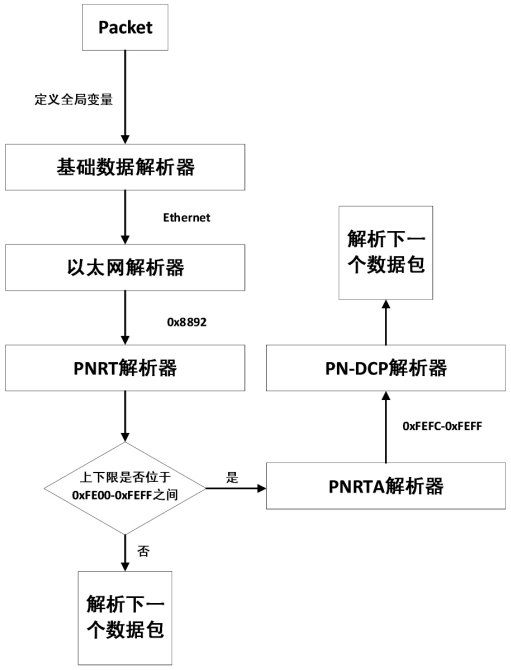

图 2 解析器系统结构图

个特定解析器。各个特定解析器调用下一个特定解析器的方法是在解析器“寄存器”内部定义了可以存放 8 个字节大小空间的上下限数组。从基础数据解析器到以太网解析器,上下限手动设定为 ETHERNET。从以太网解析器到 PN-RT 解析器,同样将上下限手动设定为 0x8892。从 PN-RT 解析器到 PN-RTA 解析器,则需要根据解析的 Frame ID 协议字段值来进行判定。

(2)数据读取模块

该模块的主要工作是对协议报文字段值的读取,字段值读取的过程也是各个特定解析器解析的过程。该过程是在定义的双向链表中完成的,链表中的每个节点结构体都由指针域和数据域组成。在指针域中包含指向 Snort 的 Packet 结构体指针变量 p 和指向包含读取协议字段值方法的结构体指针变量 ops;在数据域中定义的整形变量 position 表示协议字段偏移量,通过此偏移量,才能实现对帧结构字段的动态读取。具体的读取方法是通过定义在链表节点中的数据读取函数,数据读取函数原型及作用见表 1。

表 1 数据读取函数

函数原型	作用
Buffy_getBitsWalk8	读取此后 1 个字节大小的字段,并将偏移量+1
Buffy_getBitsWalk16	读取此后 2 个字节大小的字段,并将偏移量+2
Buffy_getBitsWalk32	读取此后 4 个字节大小的字段,并将偏移量+4
Buffy_getBitsWalk64	读取此后 8 个字节大小的字段,并将偏移量+8

在网络分析领域,关于偏移量的计算主要有基于数据包偏移和基于协议偏移的方法。由于本文所要解析的 PROFINET RT 和 DCP 协议帧格式都是固定的且均是基于以太网报头的,因此本文采用基于数据包的偏移量计算方法来对 PROFINET 协议字段值进行读取。

以以太网报头为基准,可得出 PROFINET RT 协议的以太网报头偏移量为 14,Frame ID 字段偏移量为 16(忽略 Vlan tag 信息字段)。同理,PROFINET DCP 协议的以太网报头偏移量为 14,Frame ID 字段偏移量为 16,Service ID 为 17,Service Type 为 18 等等。通过协议字段与偏移量的对应关系,就可以把 Snort 的 Packet 结构体中的数值读取到自定义的相应数值结构体中。以读取 PNOFINET RT 协议报文字段值为例,流程如下:

Step1,数据包先进入基础数据解析器,读取 Packet 结构体中的帧长度、时间戳等信息,并赋给定义的相应帧长度、时间戳结构体,接着进入以太网解析器;

Step2,在以太网解析器中,先定义 dest 结构体,之后调用 memcpy 函数将 Snort 数据包的以太网帧的目的地址字段信息复制到定义的结构体中。用同样的方法也可以将源地址与以太网类型字段信息读取到定义的相应结构体当中去。最后根据读取到的以太网类型 0x8892 来调用 PN-RT 解析器,解析完以太网字段后创建一个新的链表节点,新的节点从给定的偏移量 14 的位置开始读取下个字段;

Step3,在 PN-RT 解析器中,同样定义 Frame ID 结构体,调用 Buffy_getBitsWalk16()读取 Frame ID 字段值。之后读取到的 Frame ID 值如果取值在 0x8000-0xBFFF 之间,则此协议类型 PROFINET RT 协议;如果在 0xFE00-0xFEFF 之间,则说明此协议类型为 PROFINET DCP 协议,进入 DCP 协议的读取过程。在 DCP 解析器中,也是用同样的方法读取协议字段值。

(3)数据存储模块

该模块的主要作用是存储解析后的协议字段值,以方便后续对报文的实时演示与分析。本文通过构造一种树形结构来保存解析后的数据。在每个特定解析器解析协议的过程中,调用两种不同的结点操作函数来创建分支结点,并根据新创建的分支结点结构体中定义的布尔型变量 isImportant 的取值来判断该结点是否要存储协议字段值。因此树形结构中的节点可分为标识结点和存储结点。标识结点是上个标识结点的一个分支,每个标识结点可以创建多个存储结点。在每个标识结点只有一个存储结点的情况下,树形结构结点间的逻辑关系如图 3 所示。

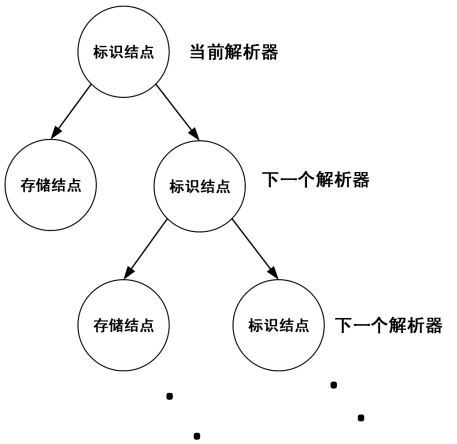

图 3 协议结点逻辑关系图

在协议树中,每个结点都包含指向父结点和子结点的指针。标识结点只用作创建分支存储结点,存储结点用来存储字段值。为了能够实现对数据的快速处理,本文选用 HashMap 数据结构来对解析后的协议字段值进行存取。在每个特定解析器中,以给定的字符串作为 key,以保存字段值的结构体作为 value 进行存储。用除留取余法构造哈希函数,从而确定关键字与存储地址的关系。最终使解析器解析后的数据保存在当前标识结点的存储结点中。

2 实时界面实现

实时界面的实现主要是为了对解析后的协议报文字段值进行可视化演示和对异常 PROFINET 协议的形象分析。实时界面是借助开源网络工具 Truffle-Hog 实现的,Truffle-Hog 支持在 Linux 系统上运行,能够接收指定类型的以太网数据包,并将接收到的数据包展现在其运行后的分析界面中。因此,为了实现对 Snort 解析后的 PROFINET 流量的实时演示,本文将树形结构中的协议封装成固定的数据结构,并实现 Snort 与 Truffle-Hog 之间的通信,使

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Truffle–Hog 能够接收以太网类型为 0x8892 的协议报文。

首先定义统一的结构体变量 Truffle，此结构体成员为包含以太网报头的结构体 etherheader 和包含 PROFINRT RT、DCP 帧字段的结构体 frame；其次为了保证 PROFINET 协议报文传输的效率，本文选取 Unix 域套接字的方式来实现 Snort 和 Truffle–Hog 之间的通信。在建立通信之前，需要将包含 Truffle 结构体信息和通信连接建立标识的头文件都加入 Truffle–Hog 源码中，进行重新编译。对于本地套接字来说，与网络套接字通信的工作流程完全相同，只是在用法上有些不同。例如套接字的创建，对于本地通信，第一个参数为协议族，在这里必须为 AF_UNIX 或者 AF_LOCAL；第二个参数为 Socket 类型，本文选用的是 SOCK_SEQPACKET；第三个参数设为 0。另外，在调用 bind()绑定地址时，sun_path 代表的是一个路径名；最后，通信建立之后，在运行 Truffle–Hog 时，便可以看到 Snort 解析后的 PROFINET 协议报文已经成功传输到 Truffle–Hog 中。

在实时界面中，MAC 地址为 00–0E–8C–CA–99–19 的节点代表 IO 控制器，MAC 地址为 28–63–36–47–90–88 的节点代表 IO 设备，此时正在建立通信过程中。以太网类型 34962 代表十六进制的 0x8892。

3 Snort 系统性能测试

通过对 Snort 预处理器插件的设计和实时界面的实现，已经完成了对 Snort 的功能扩展。为了验证改进后 Snort 系统的有效性，本文搭建了如图 4 所示的模拟攻击实验环境，并设计两种入侵攻击场景来实现 Snort 对异常 PROFINET 协议报文的检测与报警。

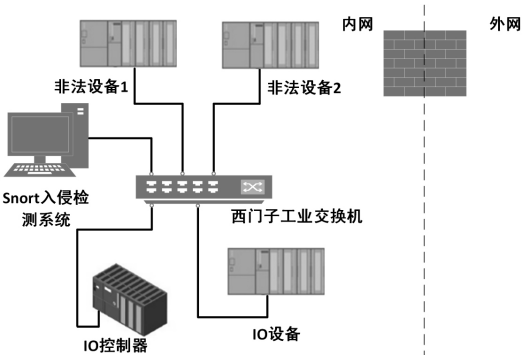

图 4 模拟攻击实验环境

此环境以工业以太网交换机为桥梁，并利用其镜像功能实现了数据的互相传送。包含改进后的 Snort 入侵检测系统和正常通信的 IO 控制器和 IO 设备，以及模拟的两台能够发送异常 PROFINET 协议报文的非法设备，实验是在内部网络中进行的。

3.1 PROFINET IO 系统安全性分析

尽管 PROFINET IO 通信系统基本都是建立在内部网络的环境中，受到外部攻击的可能性比较小，但这并不能代表绝对的安全。因此本文从两个方面来分析 PROFINET IO 通信系统可能面临的安全问题。

(1)针对 PROFINET DCP 协议的“拒绝服务”攻击

PROFINET IO 系统建立周期通信前负责对 IO 设备的 IP 和名称分配，之后便是建立周期通信的过程。而在此过程中，如果在 IO 设备的名称和 IP 被分配之前，非法设备向 IO 设备发送同样的配置信息，则有可能导致原有控制器无法连接 IO 设备的故障。此情况发生后，IO 设备便与 IO 控制器失去了连接，并被非法设备所控制，原有正常通信连接建立失败。

(2)针对 PROFINET RT 协议的 “中间人”攻击

还有一种可能产生的攻击情况就是攻击者直接根据 IO 设备的 MAC 地址试图与其建立连接，发送非法 connect 请求。这种情况也可能对原有通信系统造成不利影响，请求次数过多时会影响 IO 设备的稳定运行，造成系统故障。

3.2 模拟攻击与系统验证

针对以上两种可能发送的入侵场景，可发现其共同特征：无论是在系统启动之初还是启动之后，与 IO 设备试图取得连接的非法设备必定与正常 IO 控制器和 IO 设备的 MAC 地址不同。因此，可以以此为依据，建立检测机制。首先，在设计的预处理器插件源文件中加入针对这两种攻击场景的检测模板，使 Snort 能够对异于正常通信的 IO 控制器与 IO 设备的 MAC 地址组合产生报警信息；其次，在 Snort 的配置文件中的 Step #8 中使预处理器报警规则生效，便可以模拟攻击实验。

首先将改进后的 Snort 系统通过工业以太网交换机部署在内网中，以便能够检测内网中所有流量；其次借助开源工具 ProfinetExplorer 模拟非法设备对 IO 设备进行攻击，分别在系统不同启动阶段对 IO 设备发送非法请求，达到攻击的目的。

模拟“拒绝服务”攻击场景：当系统网络中存在未分配设备名称或者未与 IO 控制器建立通信的 IO 设备时，在 ProfinetExplorer 中选择设置分配设备名称等参数的选项，实行对 IO 设备的非法连接或对原有设备名称进行篡改。

执行以上操作时，在 Snort 所在系统中执行指令：tail –f /var/log/snort/alert，便可以发现在 Snort 的 Alert 中已经产生报警信息，报警界面如图 5 所示。

```
[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:46:07.041664

[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:46:18.726250

[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:46:18.742223
```

图 5 “拒绝服务”的报警日志

同时，对应的在 Truffle–Hog 中显示的实时界面如图 6 所示，图中已经显示了模拟非法设备的 PC 机的 MAC 地址，可以看到此时 IO 设备的名称已经被篡改，原 IO 控制器 S7–300 已经失去与 IO 设备的连接。

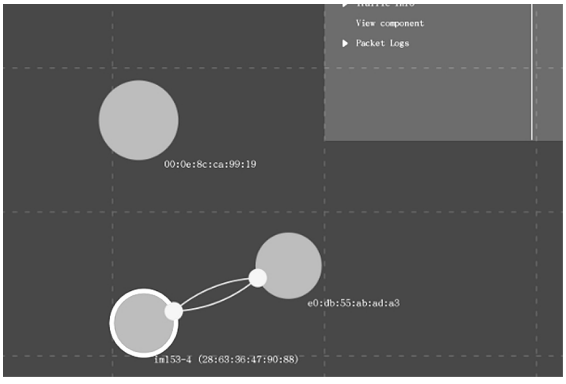

图 6 “拒绝服务”的 Truffle–Hog 界面

以上是模拟”拒绝服务”攻击场景的实验，下面进行模拟“中间人”攻击的入侵检测。当 IO 控制器与 IO 设备正常通信时，在模拟非法设备的两台 PC 中的 ProfinetExplorer 中选择“Connect IO”选项，工具便会自动向 IO 设备发送协议类型为 PN–IO 的周期性 IO 设备请求。此时，在 Snort 的 Alert 同样发出了报警，报警界面如图 7 所示。

（下转第 15 页）

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

踪中。在实验中规定相机频率为 200fps，对三条轨迹运动分别添加其对应的最优水平噪声，得到每条轨迹的误差水平，如图 4。

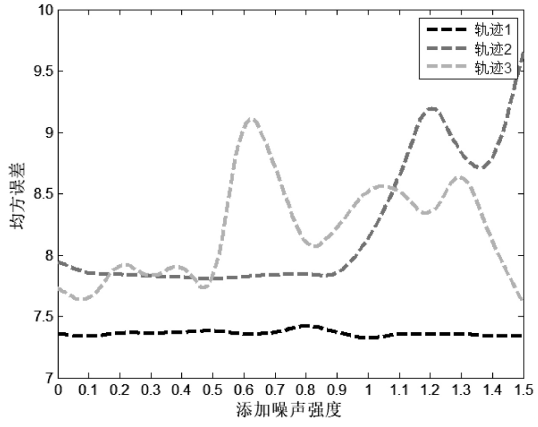

图 4 不同噪声水平下，三种运动轨迹的误差情况

可以看到，在添加不同的噪声水平后，不同的轨迹将产生不同的误差变化。对每一种噪声水平下的均方误差进行求平均得到$\bar{d}=\frac{\sum_i d_i}{3}$，得出柱状图 5。

从图 5 可以看出，随着噪声水平的增大，在特定范围内，存在最优噪声和最优输出。此时系统发生随机共振，消除了部分系统因素造成的误差。当噪声水平过大时，输出的随机性上升，造成了误差增大。另外，过大的噪声使不同轨迹标准差增大，造成系统输出的不稳定。

6 结束语

在实际追踪高速物体运动时，实时添加对应的最优噪声，可以有效减小系统追踪误差，提高高速伺服系统的跟踪精确度。本文方法具有应用价值和发展前景。

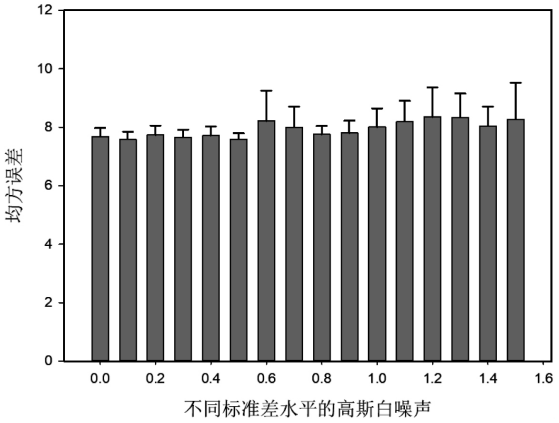

图 5 不同噪声水平下的平均追踪误差水平

参考文献

[1]Nakabo Y,Ishikawa M,Toyoda H,et al.1 ms column parallel vision system and its application of high speed target tracking[C] //Proceedings 2000 ICRA.Millennium Conference.IEEE International Conference on Robotics and Automation.Symposia Proceedings (Cat.No.00CH37065).IEEE,2000,1:650–655

[2]Dai N, Nakamura M, Komada S, et al. Tracking of moving object by manipulator using estimated image feature and its error correction on image planes [C]// IEEE International Workshop on Advanced Motion Control. 2004

[3]刘文芳，邵志刚，卢胜利，等.带时延补偿的图像雅可比矩阵在线估计方法[J].计算机工程与应用，2010，46(21)：181–184

[4]Benzi R, Sutera A, Vulpiani A. The mechanism of stochastic resonance[J]. Journal of Physics A: mathematical and general, 1981, 14(11): L453

[5]Fauve S,Heslot F. Stochastic resonance in a bistable system [J].Physics Letters A, 1983, 97(1–2): 5–7

[6]McNamara B,Wiesenfeld K.Theory of stochastic resonance[J]. Physical review A,1989,39(9):4854

[收稿日期：2019.5.5]

（上接第 12 页）

```
[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:46:18.790357

[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:46:18.804354

[**] [146:1:1] Refuse to serve or Man in the Middle attack.
 [**]
03/05-10:48:11.010125
```

图 7 “中间人攻击”的报警日志

对应此入侵场景的 Truffle–Hog 中显示的实时界面如图 8 所示。

综合以上实验，可以验证本次系统设计的有效性。系统不仅实现了对 PROFINET RT 和 DCP 协议的解析，还能检测内部局域网络中非法设备的入侵，从而降低了 PROFINET IO 通信系统被入侵的可能性，保护了通信系统的安全。

4 结束语

“安全可控”是工业化稳定快速发展的趋势之一，在工业 4.0、物联网、人工智能等新兴概念不断涌现的今天，入侵检测技术在工控系统中的应用已成为研究热点。本文设计的预处理器插件，使 Snort 入侵检测系统能够应用在 PROFINET IO 系统中，在一定程度上实现了对工业控制系统的防护。

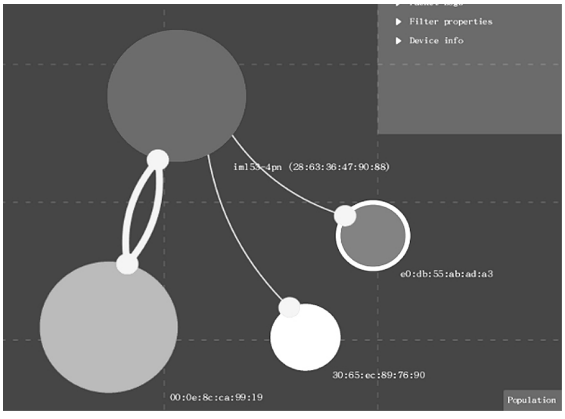

图 8 “中间人攻击”的 Truffle–Hog 界面

参考文献

[1]彭勇，江常青，谢丰，等.工业控制系统信息安全研究进展[J].清华大学学报(自然科学版)，2012，52(10)：1396–1408

[2]樊留群.实时以太网及运动控制总线技术[M].上海：同济大学出版社，2009

[3]侯向宁，高岭.基于 Snort 的 DHCP Flood 攻击检测[J].电脑开发与应用，2010，23(6)

[收稿日期：2019.3.25]

(C)1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net