# Reversing the Reversing of the TriStation Network Protocol

An **ICSsec** newb learns about ICS things:

**TRITON**

**Triconex**

**TriStation**

Steve Miller

@stvemillertime

# Steve Miller

## @stvemillertime

- FireEye/Mandiant researcher
- Focusing on network & ICS
- Former incident responder
- Reformed USG SIGINT analyst





Origin ID: phosphorus1


Optus AU
10:39 AM
83%
Chats
Mom Miller
online
8:17 AM
Today
Fyi: mail from Office of personnel management says your background investigation  records were target of a malicious cyber attack.
10:38 AM
Lol nice
10:39 AM

# Topic

- For 28 minutes, the **T** in SEC-**T** stands for:
  - **T**RITON – Malware framework (not a threat actor)
  - **T**riconex – Industrial Control Systems Controller
  - **T**riStation – Proprietary network protocol

- A story of: curiosity, obsession, laziness

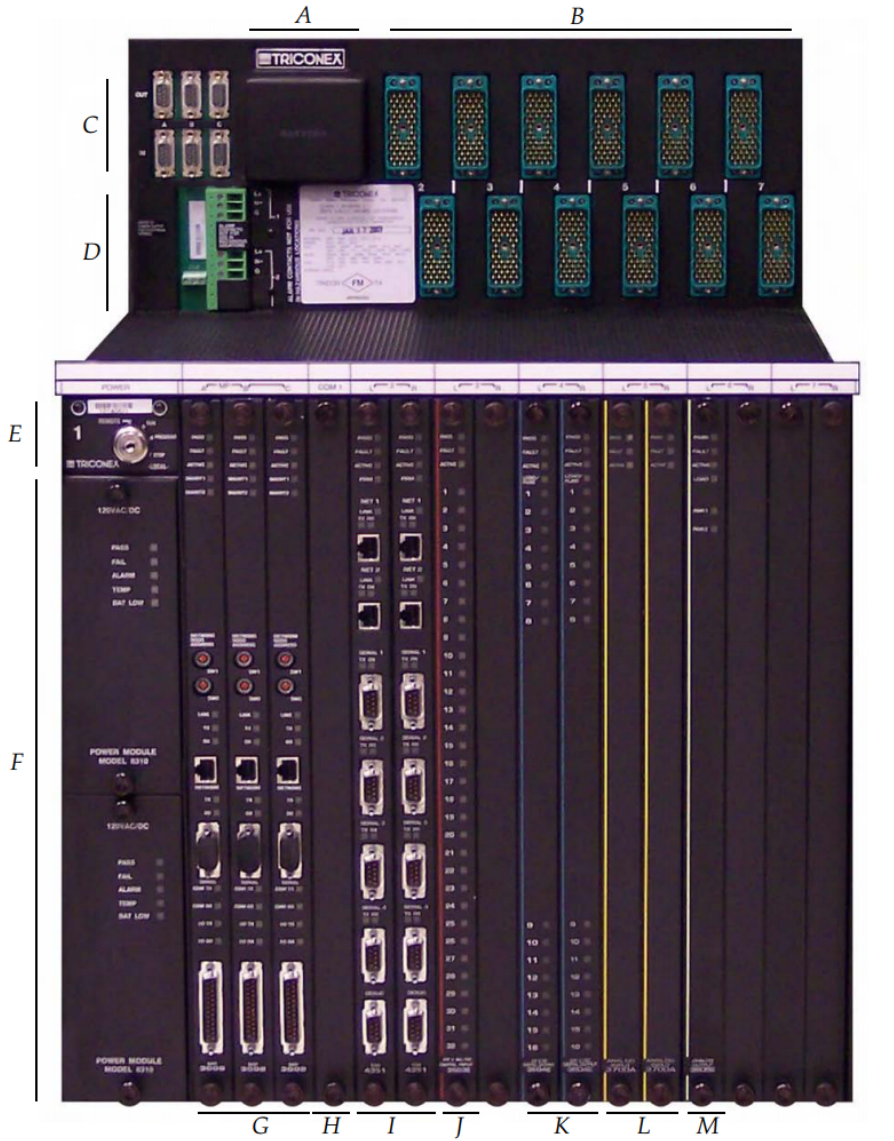
- You will learn:  <something>?

# TRITON

- Malware attack framework
- Written in python, compiled w/ py2exe
- Designed to “interact” with Triconex controllers

GENERIC PICTURE, DEFINTELY NOT THE ACTUAL VICTIM SITE, SOURCE: ZORAZHUANG/GETTY IMAGES

# Triconex Tricon Controller




A. Memory backup battery
B. Connectors for terminations
C. I/O expansion ports
D. Power terminals
E. Keyswitch
F. Redundant Power Modules
G. Three Main Processors
H. COM slot (empty)
I. Two TCMs
J. DI module without spare
K. DO module with hot-spa
L. AI module with hot-spare
M. AO module without spare

**Sample Layout of a Tricon Chassis**

**PLC** = Programmable Logic Controller

**SIS** = Safety Instrumented Systems

This one is an SIS or “safety PLC” – meant to control valves, sensors, actuators, widgets, gizmos and other physical doodads

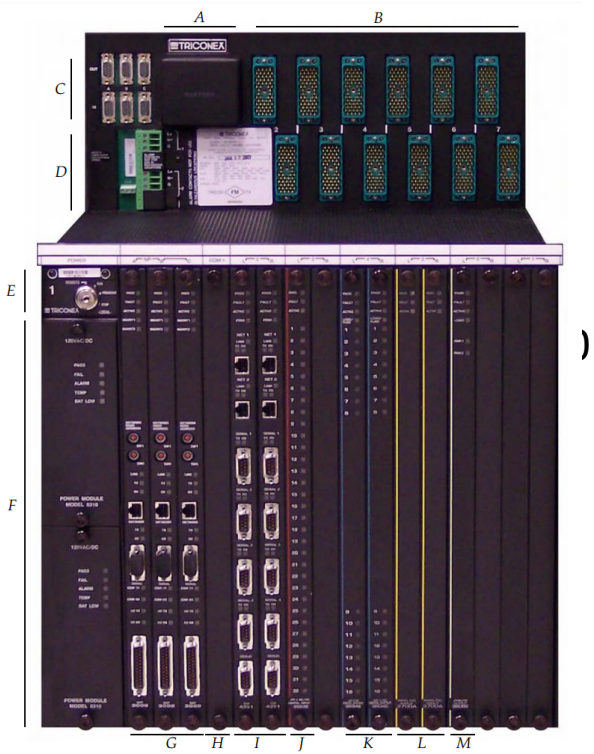
**Safety and critical control applications**

- Emergency shutdown
- Fire and gas
- Burner management system
- High integrity pressure protection system
- Turbomachinery control and protection

**Industries that might use it**

- Refining and petrochemicals
- Upstream and midstream oil and gas
- Chemicals and specialty chemicals
- Power generation
- Pharmaceuticals

# TriStation



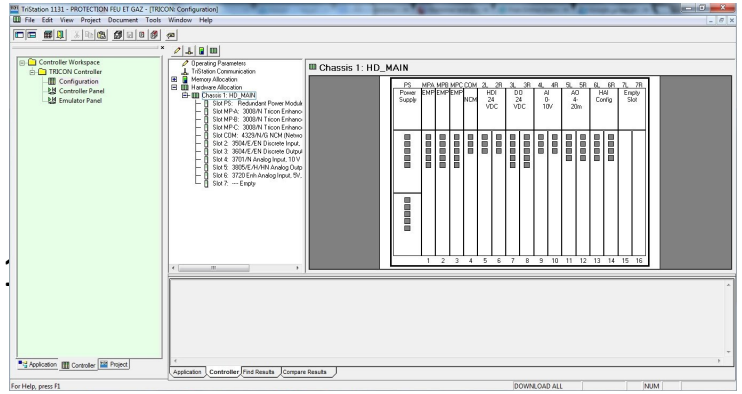

A. Memory backup battery
B. Connectors for terminations
C. I/O expansion ports
D. Power terminals
E. Keyswitch
F. Redundant Power Modules
G. Three Main Processors
H. COM slot (empty)
I. Two TCMs
J. DI module without spare
K. DO module with hot-spa
L. AI module with hot-spare
M. AO module without spare

**Sample Layout of a Tricon Chassis**

# Tricon Controller

)10100100101001010010100101000010101001011

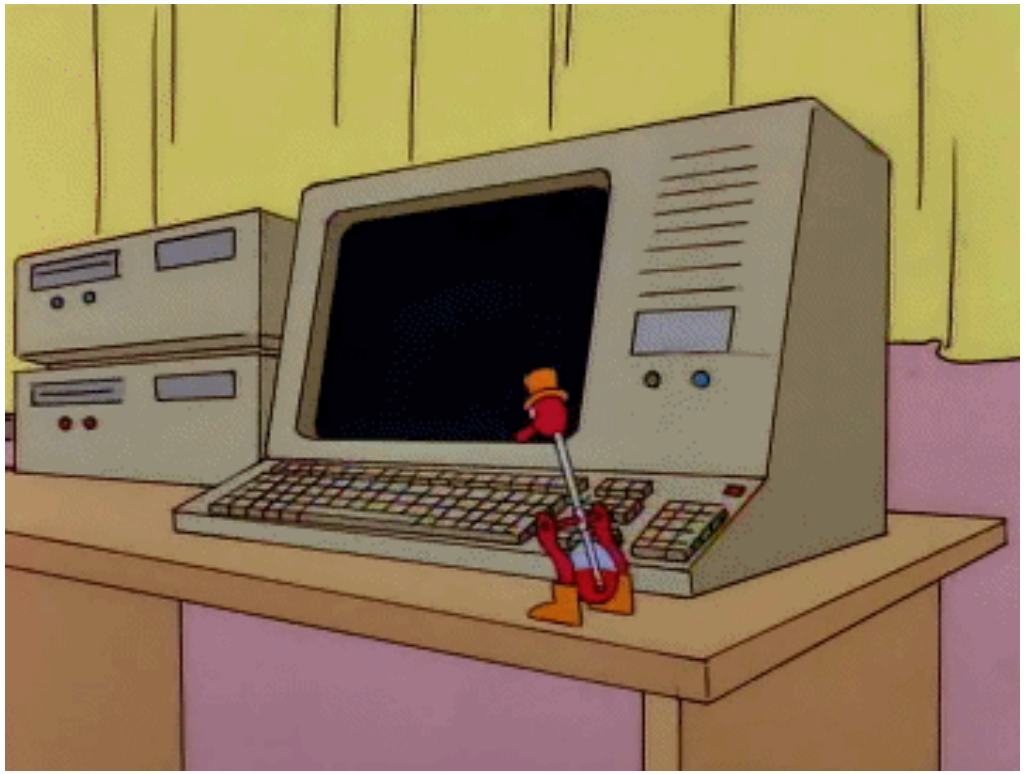
**TriStation Network Protocol**
**UDP 1502**




# TS1131 Software

# What happened?

- I became interested in the **TRITON malware framework**
- Someone said “it uses an proprietary ICS protocol”
- I said “let’s see if we can figure it out *and do as little work as possible*”

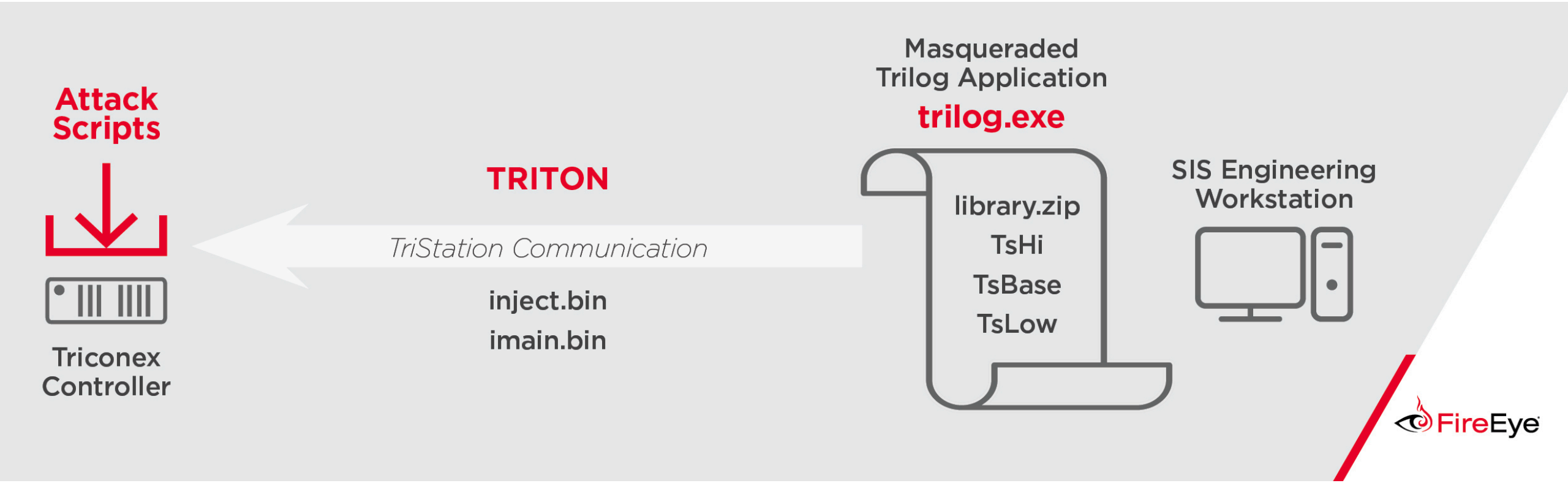


# Big questions – how was TRITON developed?

- Wtf?
- How does the TriStation network protocol work?
- How did the they reverse engineer the TriStation protocol?
- To what extent did they implement TriStation into TRITON?

*If they messed up at all, we could detect that…*

# Makeup and use of TRITON in the wild…

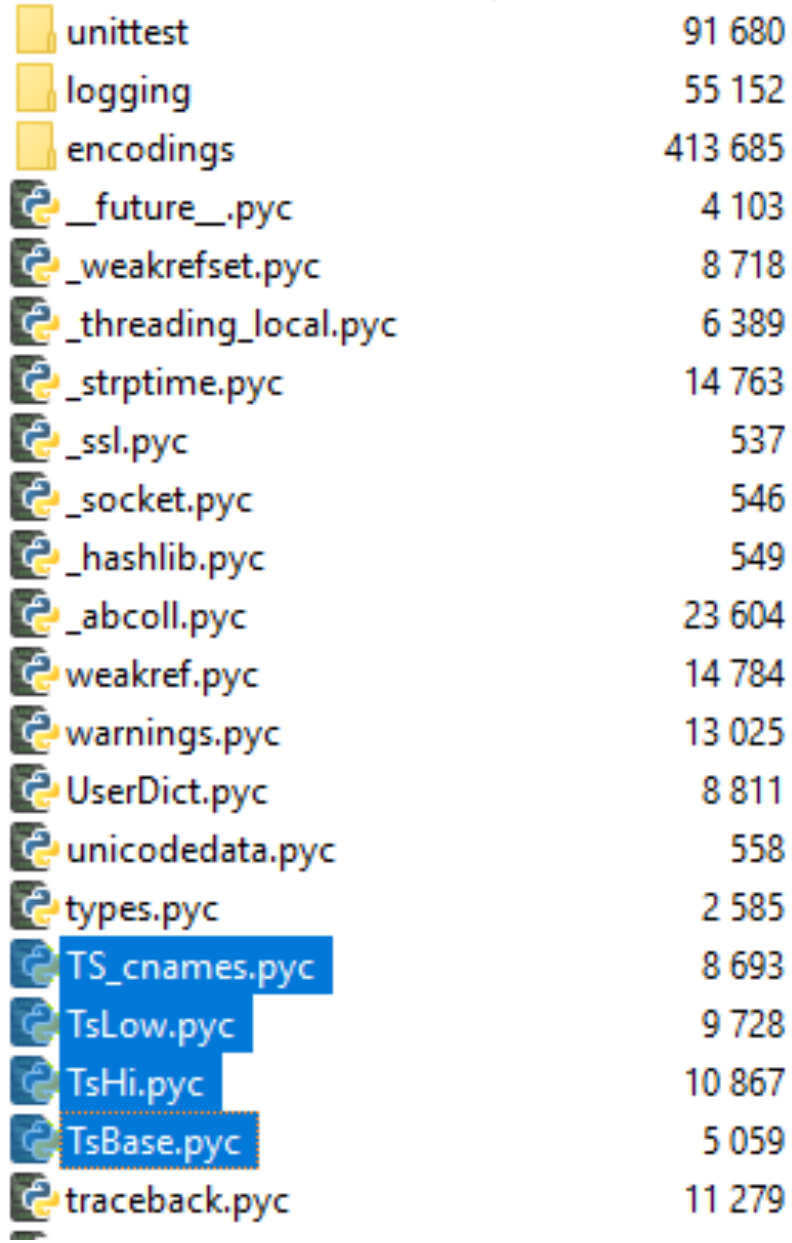


# TRITON main executable “trilog.exe”

- Compiled with py2exe

- Use of default with option “zipfile”
  - *name of shared zipfile to create; defaults to 'library.zip’.*
  - *if  set to None, the files will be bundled within the exe instead of 'library.zip’.*

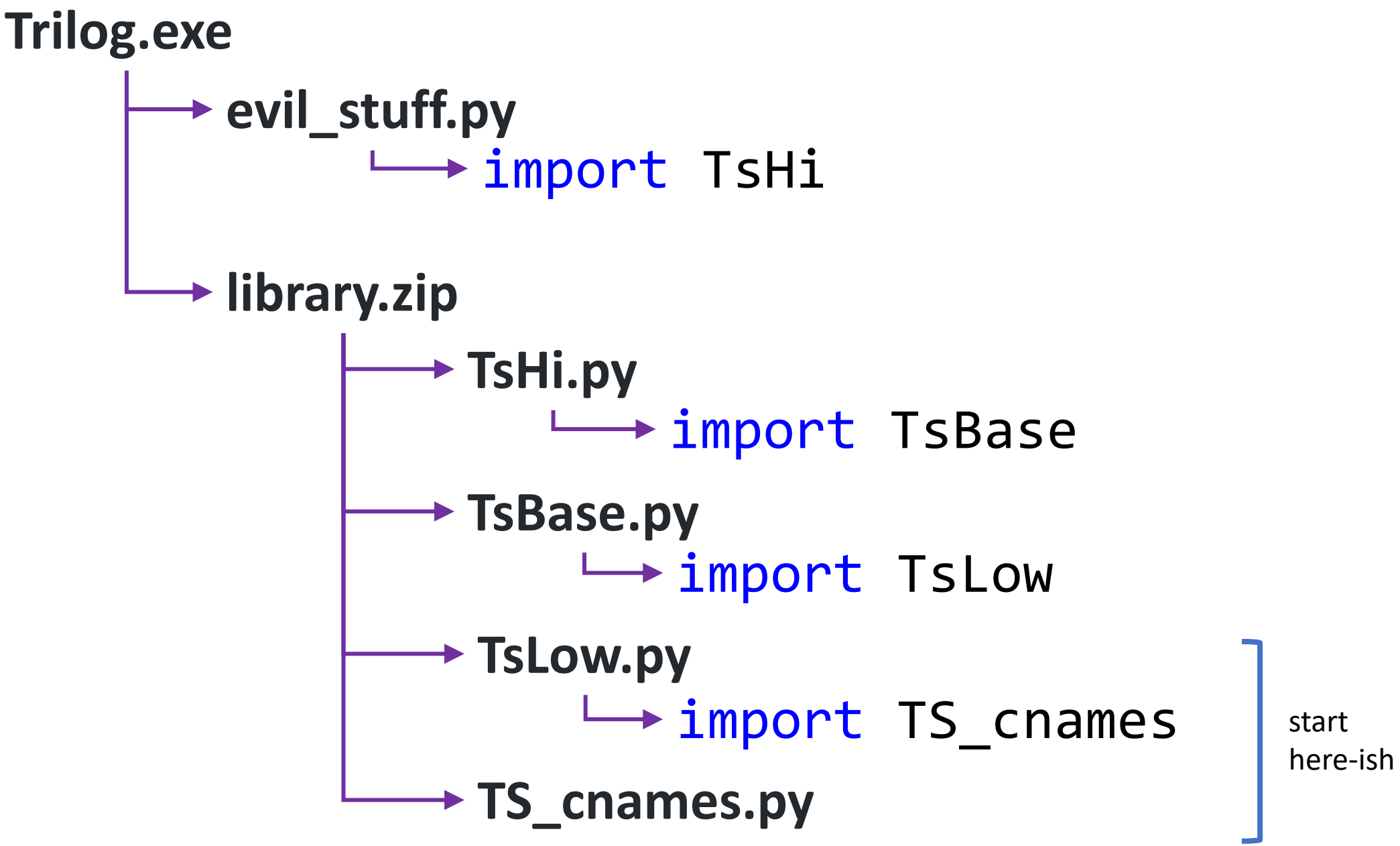
- Library.zip contains all the supporting/required/imported py scripts




Py2exe -> unpy2exe

Looking in library.zip

.pyc -> uncompyle6


Trilog.exe
evil_stuff.py
import TsHi
library.zip
TsHi.py
import TsBase
TsBase.py
import TsLow
TsLow.py
import TS_cnames
TS_cnames.py
start
here-ish

# Undocumented “binary” protocols

- There are a lot of things to look for, common elements

- Magic values (constants)
- Type values
- Counters
- Timestamps
- Size values
- Padding
- Header vs non-header

# TS_cnames.py TS_cst

TS_cnames.py

```
7  TS_cst = {1:  'CONNECT REQUEST',
8      2:  'CONNECT REPLY',
9      3:  'DISCONN REPLY',
10     4:  'DISCONN REQUEST',
11     5:  'COMMAND REPLY',
12     6:  'PING',
13     7:  'CONN LIMIT REACHED',
14     8:  'NOT CONNECTED',
15     9:  'MPS ARE DEAD',
16     10:  'ACCESS DENIED',
17     11:  'CONNECTION FAILED'
18     }
```

# TsLow.py print_last_error

```
def print_last_error(self):
    if self._uerror != 0:
        pdict('UDP Error:', self._uerror, NvTs_err)
    tcm_err = self.tcm_result()[0]
    if tcm_err >= 7:
        pdict('TCM Error ', tcm_err, TS_cnames.TS_cst)
    ts_err = self.ts_result()[0]
    if ts_err != 0:
        pdict('TS Error ', ts_err, TS_cnames.TS_names)
    if self._perror != 0:
        pdict('Parse Error:', self._perror, NvTs_err)
```

# TsLow.py
tcm_result

```
def tcm_result(self):
    if self._tcm_result != None:
        return self._tcm_result
    self._perror = -1
    data_received = self.udp_result()
    while True:
        self._tcm_result = (0, None)
        if data_received == None or len(data_received) < 6:
            print 'bad tcm size'
            self._perror = 10
            break
        type, size = struct.unpack('<HH', data_received[0:4])
        packet = data_received[4:-2]
        if len(packet) != size:
            print 'bad tcm size'
            self._perror = 10
            break
        checksum = struct.unpack('<H', data_received[-2:])[0]
        test_cksum = crc.crc16(data_received[:-2])
        if checksum != test_cksum:
            print 'bad tcm crc'
            self._perror = 11
            break
```

# TriStation packet structure…?

```
struct gen_packet {
    uint16_t  type;  // 0x00
    uint16_t  size;  // 0x02
    uint16_t  crc;   // -2:
};
```

# TsLow.py ts_result

```
def ts_result(self):
    if self._ts_result != None:
        return self._ts_result
    self._ts_result = (-1, None, 0)
    self._perror = -1
    while True:
        tcm_reply = self.tcm_result()
        if tcm_reply[0] != 5:
            self._perror = 0
            break
        ts_packet = tcm_reply[1]
        if len(ts_packet) < 10:
            print 'bad ts size'
            self._perror = 7
            break
        dir, cid, cmd, cnt, unk, cks, siz = struct.unpack('<BBBBHHH', ts_packet[0:10])
        if cnt != self._lcnt:
            print 'bad ts cnt'
```

**B = unsigned char**

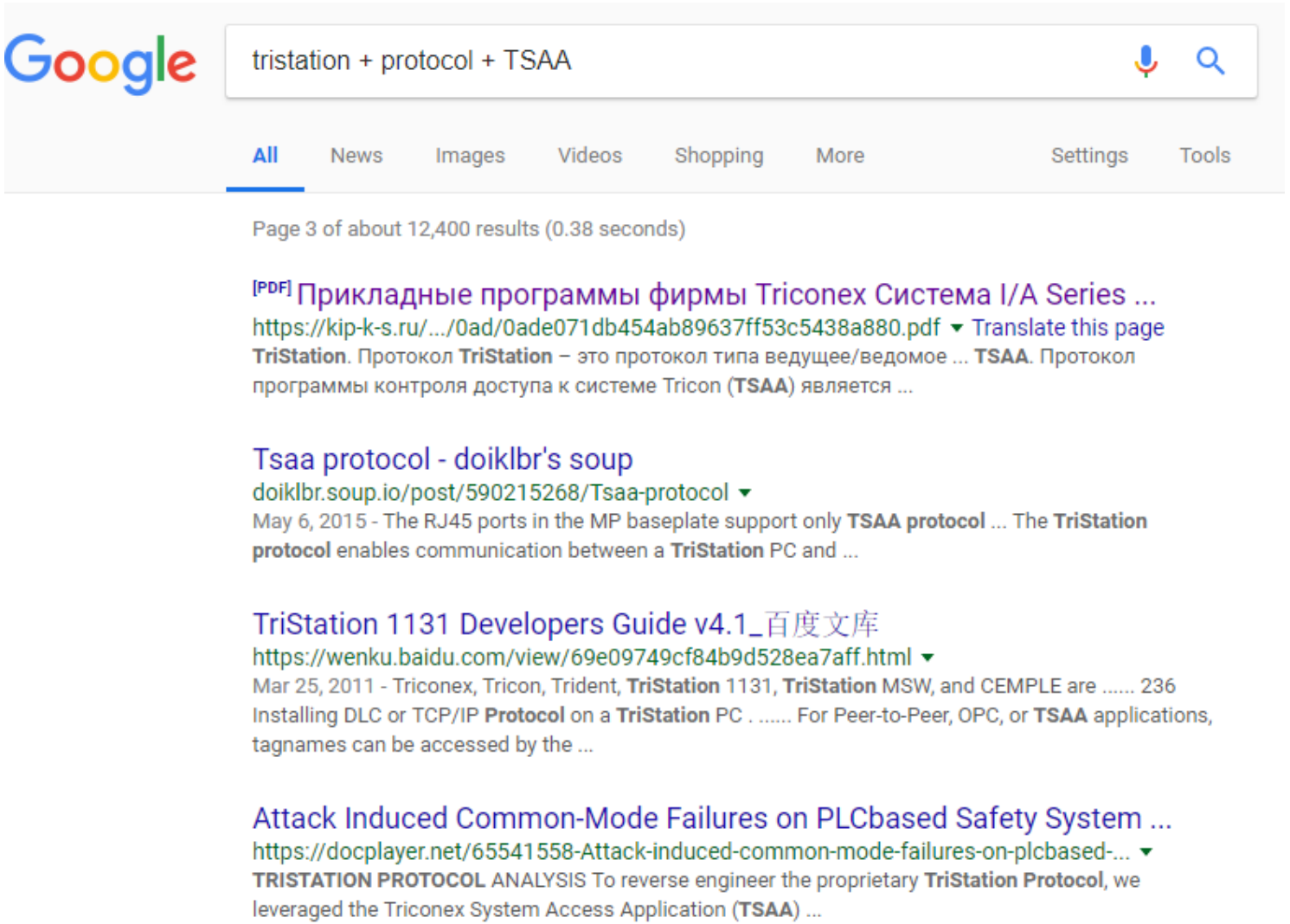
**H = unsigned short**

# TriStation cmd_packet structure…?

```
struct type_5_packet {
    uint16_t type;      // 0x00
    uint16_t size;      // 0x02
    uint8_t  dir;       // 0x04
    uint8_t  cid;       // 0x05
    uint8_t  cmd;       // 0x06
    uint8_t  cnt;       // 0x07
    uint16_t unk;       // 0x08
    uint16_t cks;       // 0x0A
    uint16_t siz;       // 0x0C
    data     data;      // 0x0E
    uint16_t crc;       // -2:
} packet;
```

```
30  TS_names = {-1: 'Not set',
31     0: 'Start download all',
32     1: 'Start download change',
33     2: 'Update configuration',
34     3: 'Upload configuration',
35     4: 'Set I/O addresses',
36     5: 'Allocate network',
37     6: 'Load vector table',
38     7: 'Set calendar',
39     8: 'Get calendar',
40     9: 'Set scan time',
41     10: 'End download all',
42     11: 'End download change',
43     12: 'Cancel download change',
44     13: 'Attach TRICON',
```

# Validating the struct



# Attack Induced Common-Mode Failures on PLC-Based Safety System in a Nuclear Power Plant: Practical Experience Report



5 Author(s)

Bernard Lim ; Daniel Chen ; Yongkyu An ; Zbigniew Kalbarczyk ; Ravishankar Iyer

View All Authors



**Abstract:**
This paper demonstrates attack induced common-mode failures on an industrial-grade (Tricon) Triple-Modular-Redundant PLC (programmable logic controller) and its impact in a Nuclear Power Plant settings. The attack exploits the fact that during the configuration phase the same control logic is downloaded to all three redundant modules. We describe how an attacker can exploit this vulnerability to embed malicious control logic and how to trigger the attack. The feasibility and the attack impact are evaluated on a testbed, which includes the Tricon PLC as part of a safety protection system in a simulated nuclear power plant.

**Published in:** 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)

**Date of Conference:** 22-25 Jan. 2017

**INSPEC Accession Number:** 16871743

**Date Added to IEEE *Xplore*:** 08 May 2017

**DOI:** 10.1109/PRDC.2017.34

**ISBN Information:**

**Publisher:** IEEE

**Electronic ISSN:** 2473-3105

**Conference Location:** Christchurch, New Zealand

# Knowledge nuggets from this paper

- TriStation (undocumented) is SIMILAR TO “TSAA” (documented)
- TriStation basic structure
- Some examples

| Byte Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | Type Constant | | Constant | | Length | | | |
| 8 | Constant (Tristation: 0 / Tricon: 268) | | | | Command | | Counter | |
| 16 | Constant | | | | Sum-of-Bytes Checksum | | | |
| 24 | Length | | | | Command Counter | | | |
| 32 | Rest of DATA (variable) | | | | | | | |
| 40 | CRC (little endian) | | | | | | | |

Fig. 4. General TriStation packet structure.

# What do you do with a protocol structure??

- Dissect

- Take pcap and figure out is going on in comms

changes by the same amount (a byte-value) in most cases. This pattern was difficult to see because this field is in the middle of the header (see Fig. 5) but the sum includes the header and data fields except the first two bytes, the CRC at the end, and the checksum itself. Furthermore, if the Sum-of-Bytes overflows the two-bytes, the sum is increased by an offset value determined by the length of the packet.

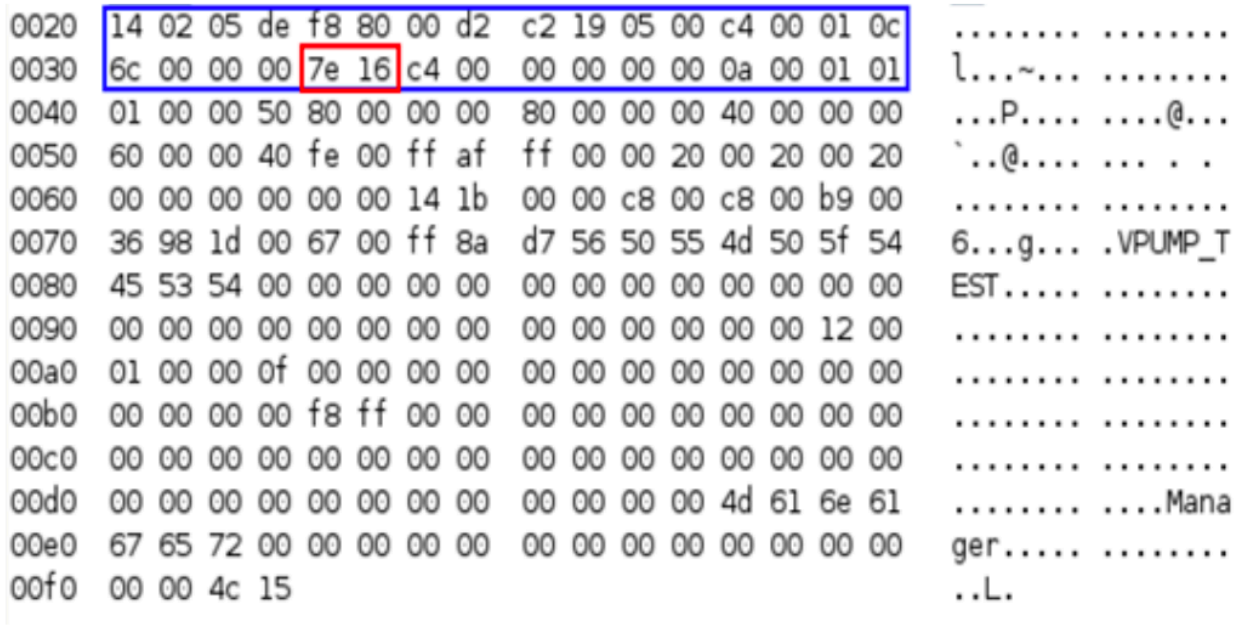



Fig. 5. Sample TriStation packet with the header highlighted in blue and the Sum-of-Bytes checksum highlighted in red.

## VII. Configuration Download Phase

the only one with such characteristics. This led us
the third Command-37 packet contains payload (th
in Fig. 4) that represents the change in the control

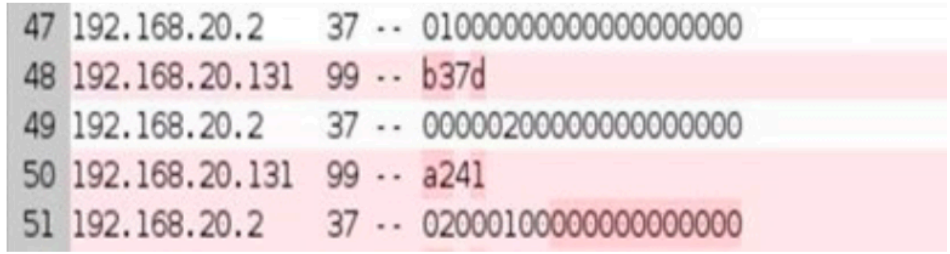



Fig. 6. Download Changes with no actual change.

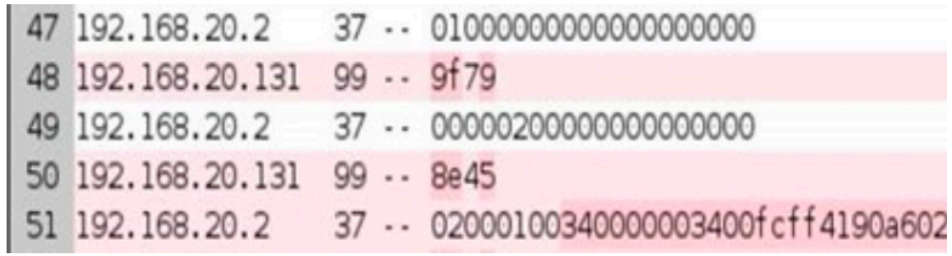



Fig. 7. Download Changes with actual change.

### B. *Identification of Parts of the Packet to Chang*

With further analysis on Command-37 packets to map portions of its DATA field to specific parts logic downloaded with relatively high confidence. Fig. 9 shows a sample program in TriStation 113

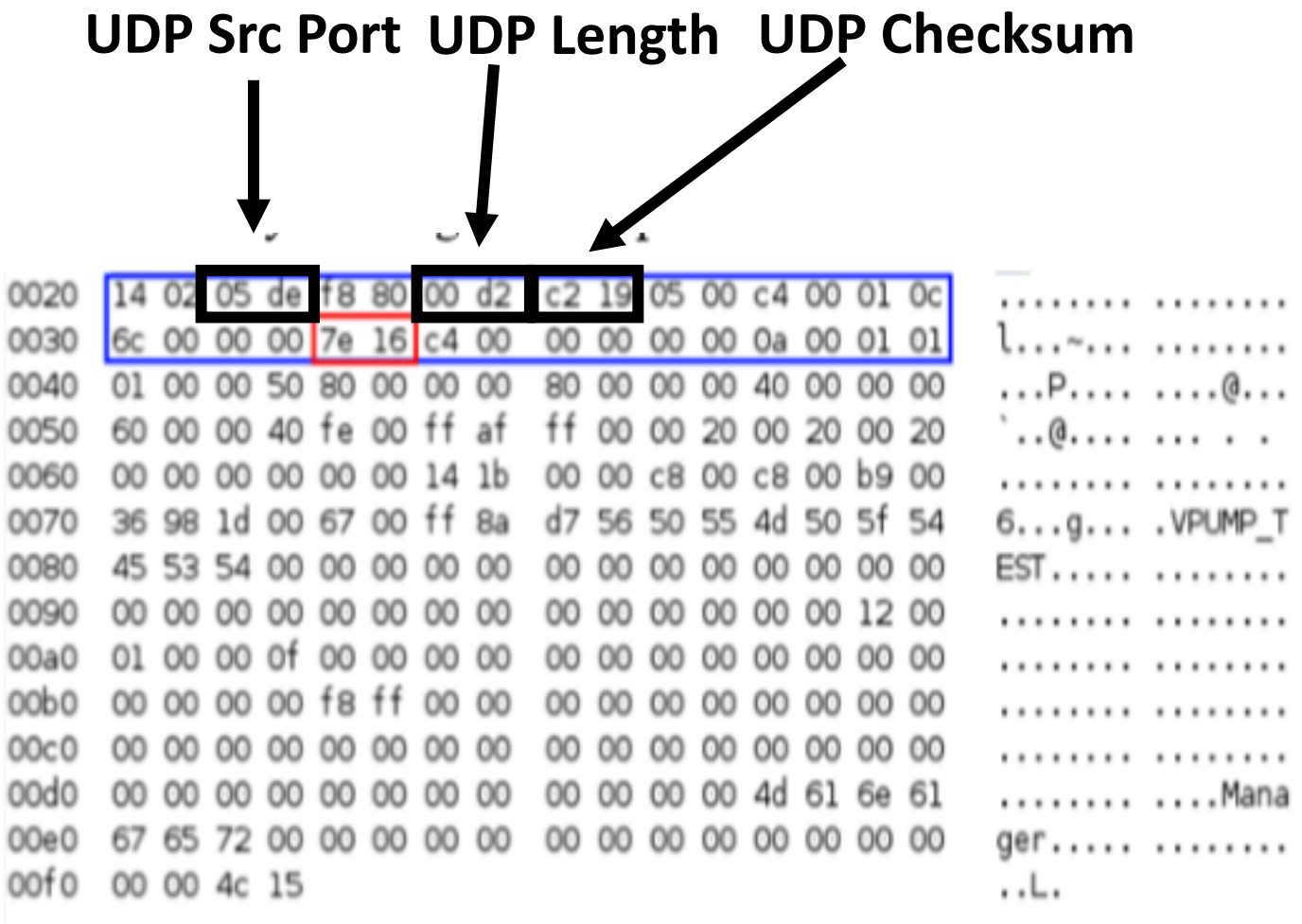
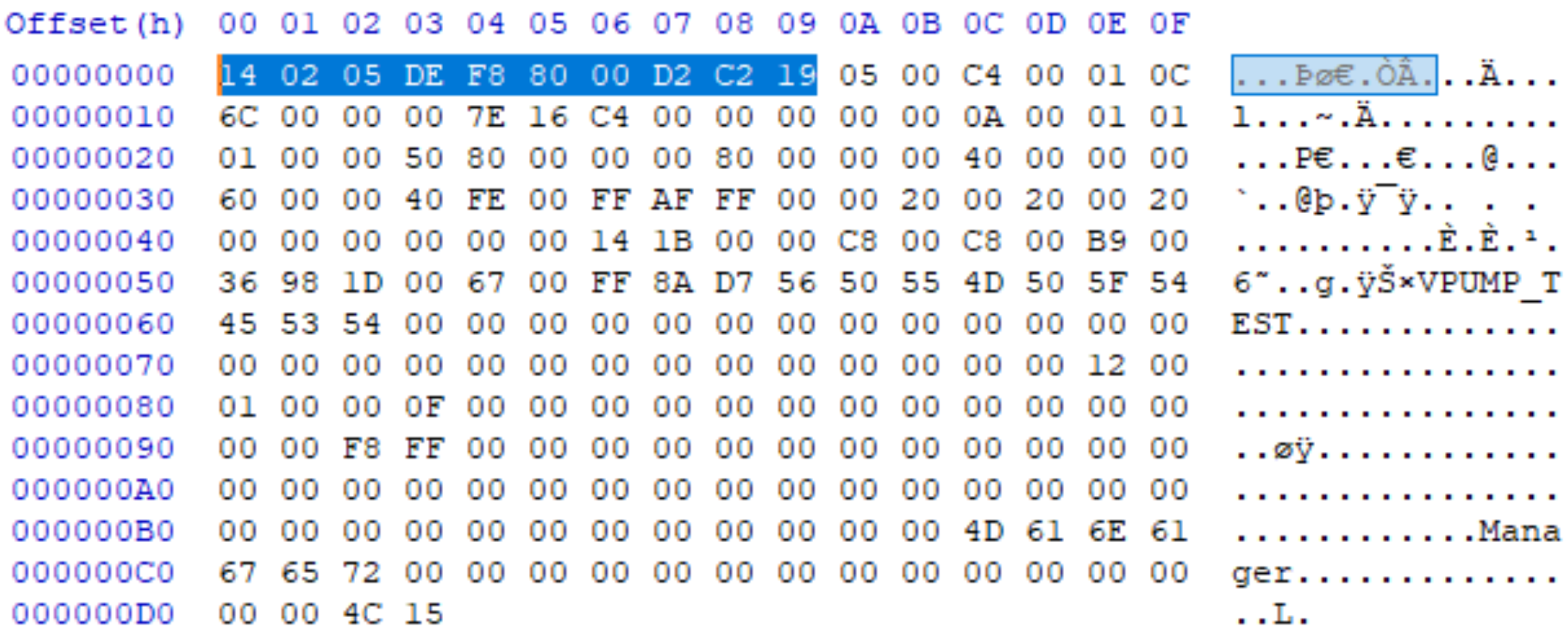



Fig. 5. Sample TriStation packet with the header highlighted in blue and the Sum-of-Bytes checksum highlighted in red.



| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 14 | 02 | 05 | DE | F8 | 80 | 00 | D2 | C2 | 19 | 05 | 00 | C4 | 00 | 01 | 0C | ...Þø€.ÒÂ..Ä... |
| 00000010 | 6C | 00 | 00 | 00 | 7E | 16 | C4 | 00 | 00 | 00 | 00 | 00 | 0A | 00 | 01 | 01 | l...~.Ä......... |
| 00000020 | 01 | 00 | 00 | 50 | 80 | 00 | 00 | 00 | 80 | 00 | 00 | 00 | 40 | 00 | 00 | 00 | ...P€...€...@... |
| 00000030 | 60 | 00 | 00 | 40 | FE | 00 | FF | AF | FF | 00 | 00 | 20 | 00 | 20 | 00 | 20 | `..@þ.ÿ¯ÿ.. . . |
| 00000040 | 00 | 00 | 00 | 00 | 00 | 00 | 14 | 1B | 00 | 00 | C8 | 00 | C8 | 00 | B9 | 00 | .........È.È.¹. |
| 00000050 | 36 | 98 | 1D | 00 | 67 | 00 | FF | 8A | D7 | 56 | 50 | 55 | 4D | 50 | 5F | 54 | 6˜..g.ÿŠ×VPUMP_T |
| 00000060 | 45 | 53 | 54 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | EST............. |
| 00000070 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 12 | 00 | ................ |
| 00000080 | 01 | 00 | 00 | 0F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ................ |
| 00000090 | 00 | 00 | F8 | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ..øÿ............ |
| 000000A0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ................ |
| 000000B0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 4D | 61 | 6E | 61 | ............Mana |
| 000000C0 | 67 | 65 | 72 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ger............. |
| 000000D0 | 00 | 00 | 4C | 15 | | | | | | | | | | | | | ..L. |

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   05 00 C4 00 01 0C 6C 00 00 00 7E 16 C4 00 00 00  ..Ä...l...~.Ä...
00000010   00 00 0A 00 01 01 01 00 00 50 80 00 00 00 80 00  .........P€...€.
00000020   00 00 40 00 00 00 60 00 00 40 FE 00 FF AF FF 00  ..@...`..@þ.ÿ¯ÿ.
00000030   00 20 00 20 00 20 00 00 00 00 00 00 14 1B 00 00  . . . ..........
00000040   C8 00 C8 00 B9 00 36 98 1D 00 67 00 FF 8A D7 56  È.È.¹.6˜..g.ÿŠ×V
00000050   50 55 4D 50 5F 54 45 53 54 00 00 00 00 00 00 00  PUMP_TEST.......
00000060   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070   00 00 00 00 12 00 01 00 00 0F 00 00 00 00 00 00  ................
00000080   00 00 00 00 00 00 00 00 F8 FF 00 00 00 00 00 00  ........øÿ......
00000090   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0   00 00 4D 61 6E 61 67 65 72 00 00 00 00 00 00 00  ..Manager.......
000000C0   00 00 00 00 00 00 00 00 4C 15                    ........L.
```

## Recipe

### From Hex

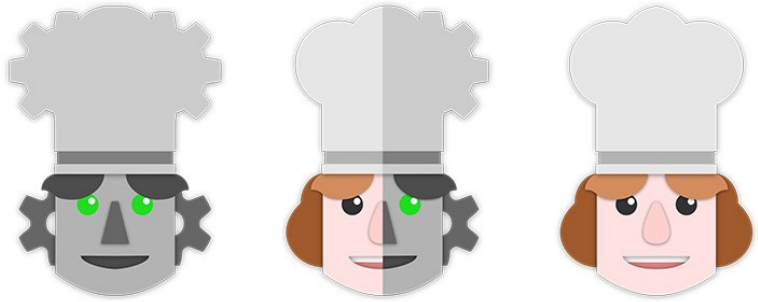
Delimiter: Auto

### To Hexdump

Width: 16

Upper case hex

Include final length



CyberChef

## Input

start: 605
end: 605
length: 0

length: 605
lines: 1

```
05 00 C4 00 01 0C 6C 00 00 00 7E 16 C4 00 00 00 00 00 0A 00 01 01 01 00 00 50
80 00 00 00 80 00 00 00 40 00 00 00 60 00 00 40 FE 00 FF AF FF 00 00 20 00 20
00 20 00 00 00 00 00 00 14 1B 00 00 C8 00 C8 00 B9 00 36 98 1D 00 67 00 FF 8A
D7 56 50 55 4D 50 5F 54 45 53 54 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 12 00 01 00 00 0F 00 00 00 00 00 00 00 00
00 00 00 00 00 00 F8 FF 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 00 4D 61 6E 61
67 65 72 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4C 15
```

## Output

time: 2ms
length: 1007
lines: 13

```
00000000  05 00 c4 00 01 0c 6c 00 00 00 7e 16 c4 00 00 00  |..Ä...l...~.Ä...|
00000010  00 00 0a 00 01 01 01 00 00 50 80 00 00 00 80 00  |.........P......|
00000020  00 00 40 00 00 00 60 00 00 40 fe 00 ff af ff 00  |..@...`..@þ.ÿ¯ÿ.|
00000030  00 20 00 20 00 20 00 00 00 00 00 00 14 1b 00 00  |. . . ..........|
00000040  c8 00 c8 00 b9 00 36 98 1d 00 67 00 ff 8a d7 56  |È.È.¹.6...g.ÿ.×V|
00000050  50 55 4d 50 5f 54 45 53 54 00 00 00 00 00 00 00  |PUMP_TEST.......|
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
00000070  00 00 00 00 12 00 01 00 00 0f 00 00 00 00 00 00  |................|
00000080  00 00 00 00 00 00 00 00 f8 ff 00 00 00 00 00 00  |........øÿ......|
00000090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
000000a0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  |................|
000000b0  00 00 4d 61 6e 61 67 65 72 00 00 00 00 00 00 00  |..Manager.......|
000000c0  00 00 00 00 00 00 00 00 4c 15                    |........L.|
```

# Text2pcap it

```
PS C:\Users\smiller> & 'C:\text2pcap.exe’
      -i 17 -4 192.168.1.99,192.168.1.33 -u 1502,58714
hexdump.txt out.pcap
Input from: hexdump.txt
Output to: out.pcap
Output format: PCAP
Generate dummy Ethernet header: Protocol: 0x800
Generate dummy IP header: Protocol: 17
Generate dummy UDP header: Source port: 1502. Dest port:
58714
Wrote packet of 244 bytes.
Read 1 potential packet, wrote 1 packet (284 bytes).
```

| No. | Time | Protocol | Length | Info |
|---|---|---|---|---|
| 1 | 0.000000 | UDP | 244 | 1502 → 58714 Len=202 |

```
> Frame 1: 244 bytes on wire (1952 bits), 244 bytes captured (1952 bits)
> Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
> Internet Protocol Version 4, Src: 192.168.1.99, Dst: 192.168.1.33
v User Datagram Protocol, Src Port: 1502, Dst Port: 58714
    Source Port: 1502
    Destination Port: 58714
    Length: 210
    Checksum: 0xfb40 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
v Data (202 bytes)
    Data: 0500c400010c6c0000007e16c400000000000a0001010100...
    [Length: 202]
```

```
0000  0a 02 02 02 02 02 0a 01  01 01 01 01 08 00 45 00   ........ ......E.
0010  00 e6 12 34 00 00 ff 11  24 fe c0 a8 01 63 c0 a8   ...4.... $....c..
0020  01 21 05 de e5 5a 00 d2  fb 40 05 00 c4 00 01 0c   .!...Z.. .@......
0030  6c 00 00 00 7e 16 c4 00  00 00 00 00 0a 00 01 01   l...~... ........
0040  01 00 00 50 80 00 00 00  80 00 00 00 40 00 00 00   ...P.... ....@...
0050  60 00 00 40 fe 00 ff af  ff 00 00 20 00 20 00 20   `..@.... ... . .
0060  00 00 00 00 00 00 14 1b  00 00 c8 00 c8 00 b9 00   ........ ........
0070  36 98 1d 00 67 00 ff 8a  d7 56 50 55 4d 50 5f 54   6...g... .VPUMP_T
0080  45 53 54 00 00 00 00 00  00 00 00 00 00 00 00 00   EST..... ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 12 00   ........ ........
00a0  01 00 00 0f 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 f8 ff 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 4d 61 6e 61   ........ ....Mana
00e0  67 65 72 00 00 00 00 00  00 00 00 00 00 00 00 00   ger..... ........
```

```
268
269      message_type = ProtoField.uint16 ("tris.message_type", "messageType", base.DEC,
270                         header_message_type_codes)
271     message_length = ProtoField.uint16 ("tris.message_lenth", "messageLength", base.DEC)
272     message_src_id = ProtoField.uint16 ("tris.message_src_id", "messageSrcId", base.DEC)
273      cmd_func_code = ProtoField.uint8  ("tris.cmd_func_code", "commandFunctionCode", base.DEC,
274                         command_function_codes_TS_cnames)
275        cmd_counter = ProtoField.uint8  ("tris.cmd_counter", "commandCounter", base.DEC)
276      unknown_const = ProtoField.uint16 ("tris.unknown_const", "unknownConstant", base.DEC)
277       sob_checksum = ProtoField.uint16 ("tris.sob_checksum", "sumByteChecksum", base.DEC)
278     message_length2 = ProtoField.uint16 ("tris.message_length2", "messageLength", base.DEC)
279       payload_data = ProtoField.new    ("tris.payload_data", "payloadData", base.DEC)
280               crc_val = ProtoField.uint16 ("tris.crc_val", "crc_val", base.DEC)
281
282 WARNING: THIS IS MY EXTREMELY BAD LUA CODE
283
284
285  function tris.dissector(buffer, pinfo, tree)
286    length = buffer:len()
287    if length == 0 then return end
288
289    pinfo.cols.protocol = tris.name
290
291    if string.find(tostring(pinfo.cols.info), "^TriStation") == nil then
292          pinfo.cols.info:set("TriStation" .. " Detail: ")
293    end
294
295    local subtree = tree:add(tris, buffer(), "TriStation Protocol Data")
296    subtree:add_le(message_type, buffer(0,2))
297    subtree:add_le(message_length, buffer(2,2))
```

# Nozomi Networks

The leader of industrial cybersecurity. Delivering real-time visibility to manage cyber risk & improve resilience for ICS and industrial operations at scale

San Francisco, CA  https://www.nozominetworks.com

Repositories 18  People 3  Projects 0

Search repositories...  Type: All

EXTREMELY GOOD LUA CODE

## tricotools

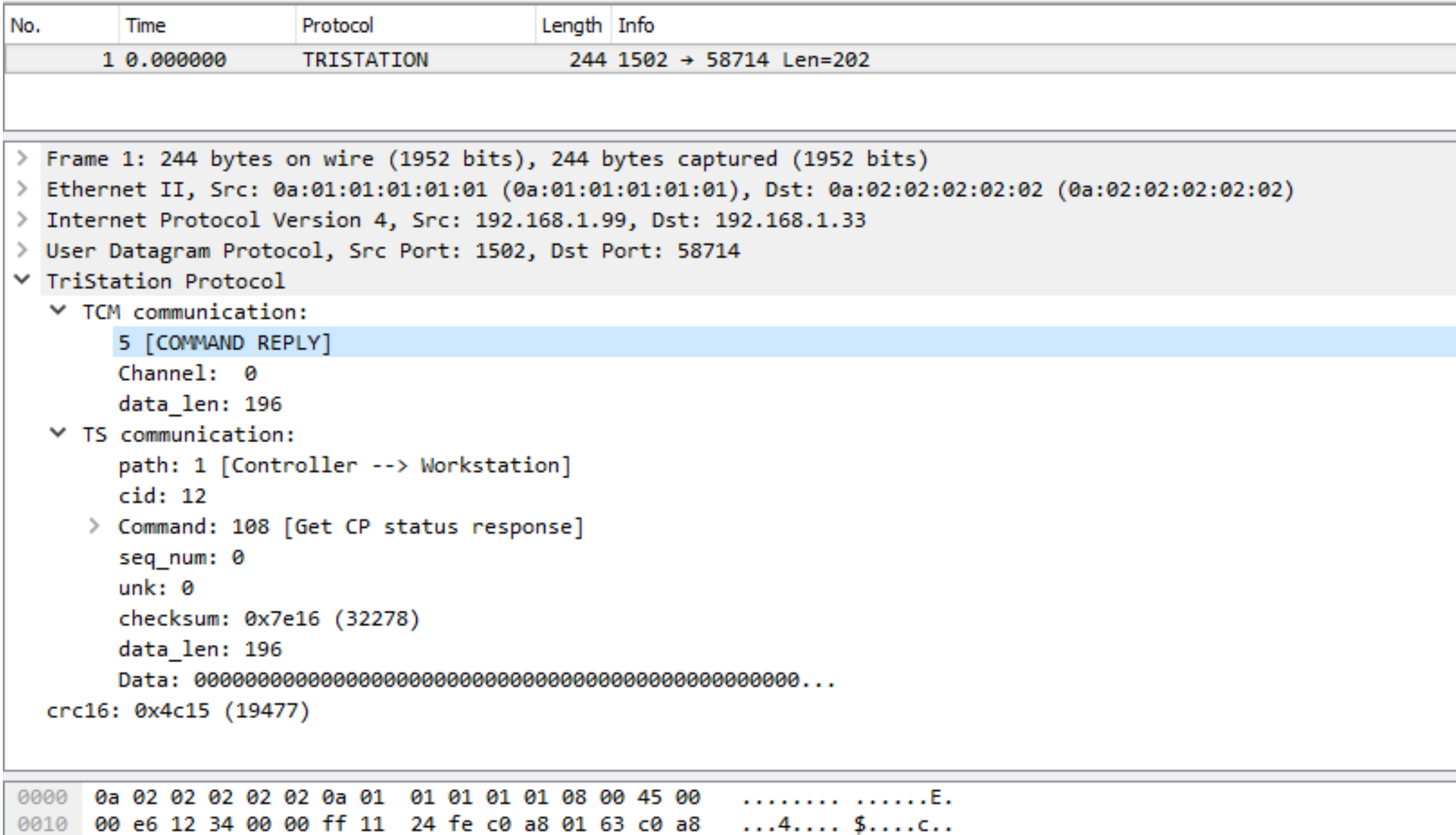
Triconex TriStation utilities and tools

Lua  ★ 21  3  BSD-3-Clause  Updated on Aug 14

Top languages

Python  JavaScript  Ruby

C++  HTML


No. Time Protocol Length Info
1 0.000000 TRISTATION 244 1502 → 58714 Len=202
Frame 1: 244 bytes on wire (1952 bits), 244 bytes captured (1952 bits)
Ethernet II, Src: 0a:01:01:01:01:01 (0a:01:01:01:01:01), Dst: 0a:02:02:02:02:02 (0a:02:02:02:02:02)
Internet Protocol Version 4, Src: 192.168.1.99, Dst: 192.168.1.33
User Datagram Protocol, Src Port: 1502, Dst Port: 58714
TriStation Protocol
TCM communication:
5 [COMMAND REPLY]
Channel: 0
data_len: 196
TS communication:
path: 1 [Controller --> Workstation]
cid: 12
Command: 108 [Get CP status response]
seq_num: 0
unk: 0
checksum: 0x7e16 (32278)
data_len: 196
Data: 0000000000000000000000000000000000000000000000000...
crc16: 0x4c15 (19477)
0000 0a 02 02 02 02 02 0a 01 01 01 01 01 01 08 00 45 00 ..... .....E.
0010 00 e6 12 34 00 00 ff 11 24 fe c0 a8 01 63 c0 a8 ...4.... $....c..

# What else can you do with proto struct?

- Observe traffic in real-ish time
- Don’t think about Snort rules as necessarily providing “alerts” for just “evil” stuff
- Context, yo

```
# Embedded file name: TS_
# Compiled at: 2017-08-03
TS_cst = {1: 'CONNECT REQ
    2: 'CONNECT REPLY',
    3: 'DISCONN REPLY',
```

```
def tcm_result(self):
    if self._tcm_result != None:
        return self._tcm_result
    self._perror = -1
    data_received = self.udp_result()
    while True:
        self._tcm_result = (0, None)
        if data_received == None or len(data_received) < 6:
            print 'bad tcm size'
            self._perror = 10
            break
        type, size = struct.unpack('<HH', data_received[0:4])
        packet = data_received[4:-2]
        if len(packet) != size:
            print 'bad tcm size'
            self._perror = 10
            break
        checksum = struct.unpack('<H', data_received[-2:])[0]
        test_cksum = crc.crc16(data_received[:-2])
        if checksum != test_cksum:
            print 'bad tcm crc'
            self._perror = 11
            break
```

| | type | size | crc16 |
|---|---|---|---|
| content:"\|01 00 | 00 00 | ?? ??\|"; | |


alert udp any any -> any 1502 (msg: "TriStation CR"; content:"|01 00 00 00 01 fc|"; sid:1;)

# Bundesamt für Sicherheit in der Informationstechnik

- BSI's ICS-SEC team
- Thomas Schmidt
- Snort IDS ruleset for TriStation
- Visibility where none existed b4

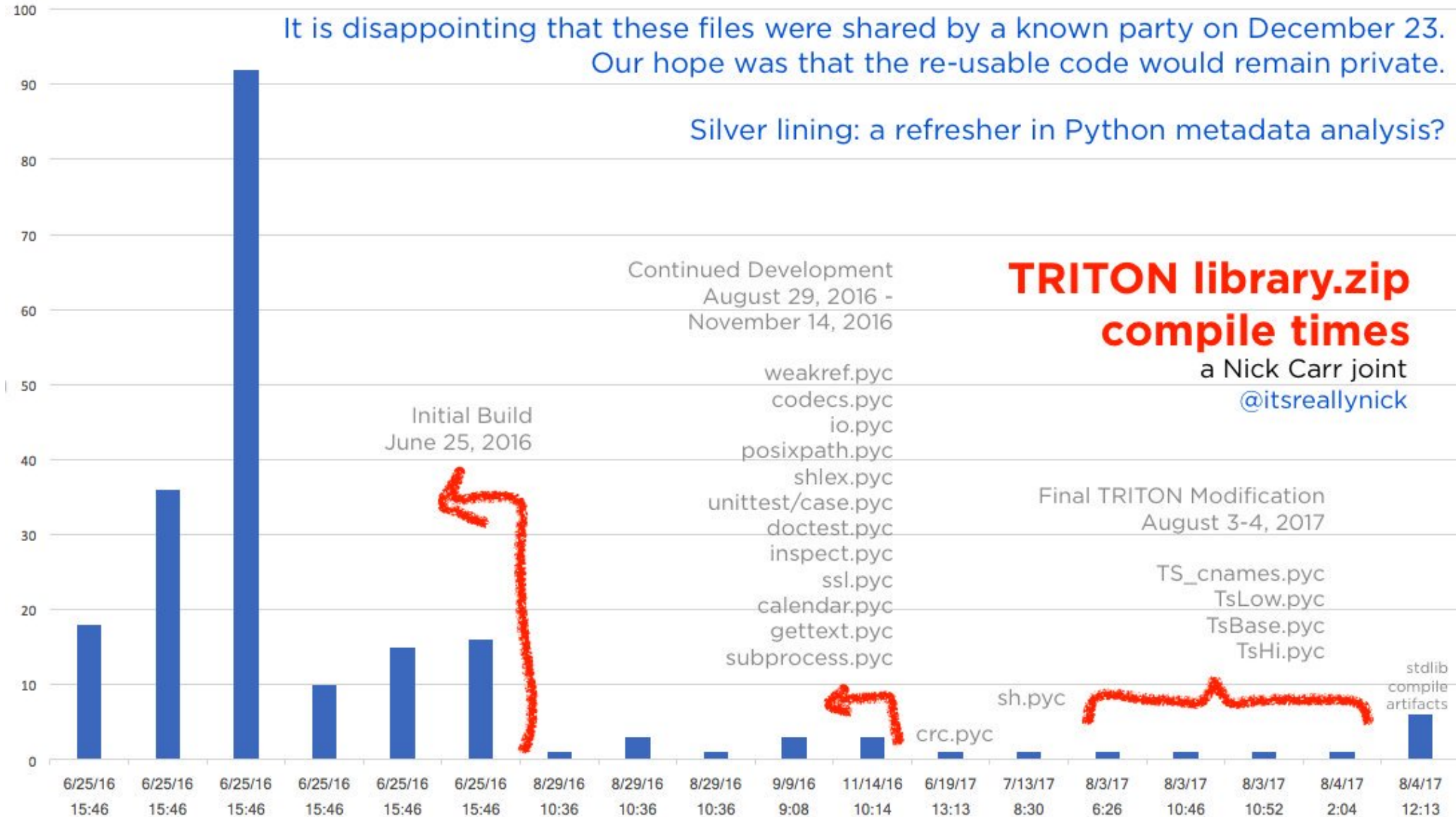
```
#==== Alert Rules (Simple) ====
# Alert on commands if they weren't sent by the $TS_EWS to the $TS_CONTROLLER (since that's obviously
suspicious traffic)
# Note: This won't catch packets where the adversary set the source IP to $TS_EWS

# Alert on any Connection Request that is sent to a SPLC on UDP/$TS_PORT unauthorized
alert udp !$TS_EWS any -> $TS_CONTROLLER $TS_PORT (msg:"TriStation Connection Request to SPLC attempt
Non Authorized Host"; sid:851750010; rev:3; content:"|01 00 00 00 01 FC|"; offset:0; depth:6;
classtype:bad-unknown;)

# Alert on any Connection Reply that is sent from a SPLC on UDP/$TS_PORT to an unauthorized host
alert udp $TS_CONTROLLER $TS_PORT -> !$TS_EWS any (msg:"TriStation Connection Reply from SPLC attempt
Non Authorized Host"; sid:851750020; rev:3; content:"|02 00 00 00 01 B8|"; offset:0; depth:6;
classtype:bad-unknown;)

# Alert on any Disconnection Reply that is sent from a SPLC on UDP/$TS_PORT to an unauthorized host
alert udp $TS_CONTROLLER $TS_PORT -> !$TS_EWS any (msg:"TriStation Disconnection Reply from SPLC attem
To Non Authorized Host"; sid:851750030; rev:3; content:"|03 00 00 00 00 44|"; offset:0; depth:6;
classtype:bad-unknown;)

# Alert on any Disconnection Request that is sent to a SPLC on UDP/$TS_PORT unauthorized
alert udp !$TS_EWS any -> $TS_CONTROLLER $TS_PORT (msg:"TriStation Disconnection Request to SPLC attem
From Non Authorized Host"; sid:851750040; rev:3; content:"|04 00 00 00 01 30|"; offset:0; depth:6;
classtype:bad-unknown;)
```

# How was TRITON developed?

- It took these developers/operators a long time, maybe ~ a year?

242 lines (242 sloc) | 6.96 KB

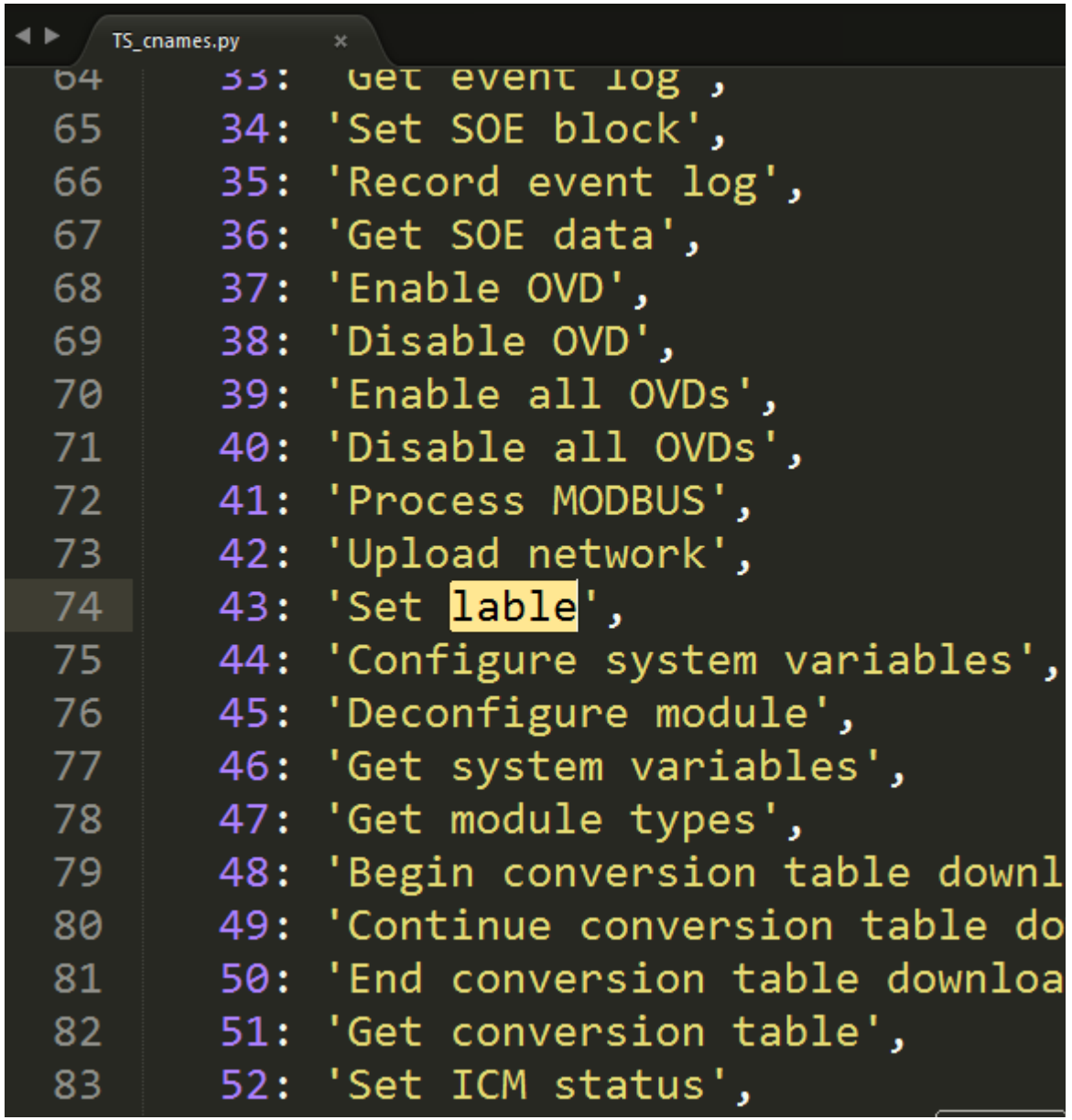
```
1   # uncompyle6 version 2.14.1
2   # Python bytecode 2.7 (62211)
3   # Decompiled from: Python 2.7.12 (default, Nov 19 2016, 06:48:10)
4   # [GCC 5.4.0 20160609]
5   # Embedded file name: TS_cnames.pyc
6   # Compiled at: 2017-08-03 12:26:36
7   TS_cst = {1: 'CONNECT REQUEST',
8      2: 'CONNECT REPLY',
```

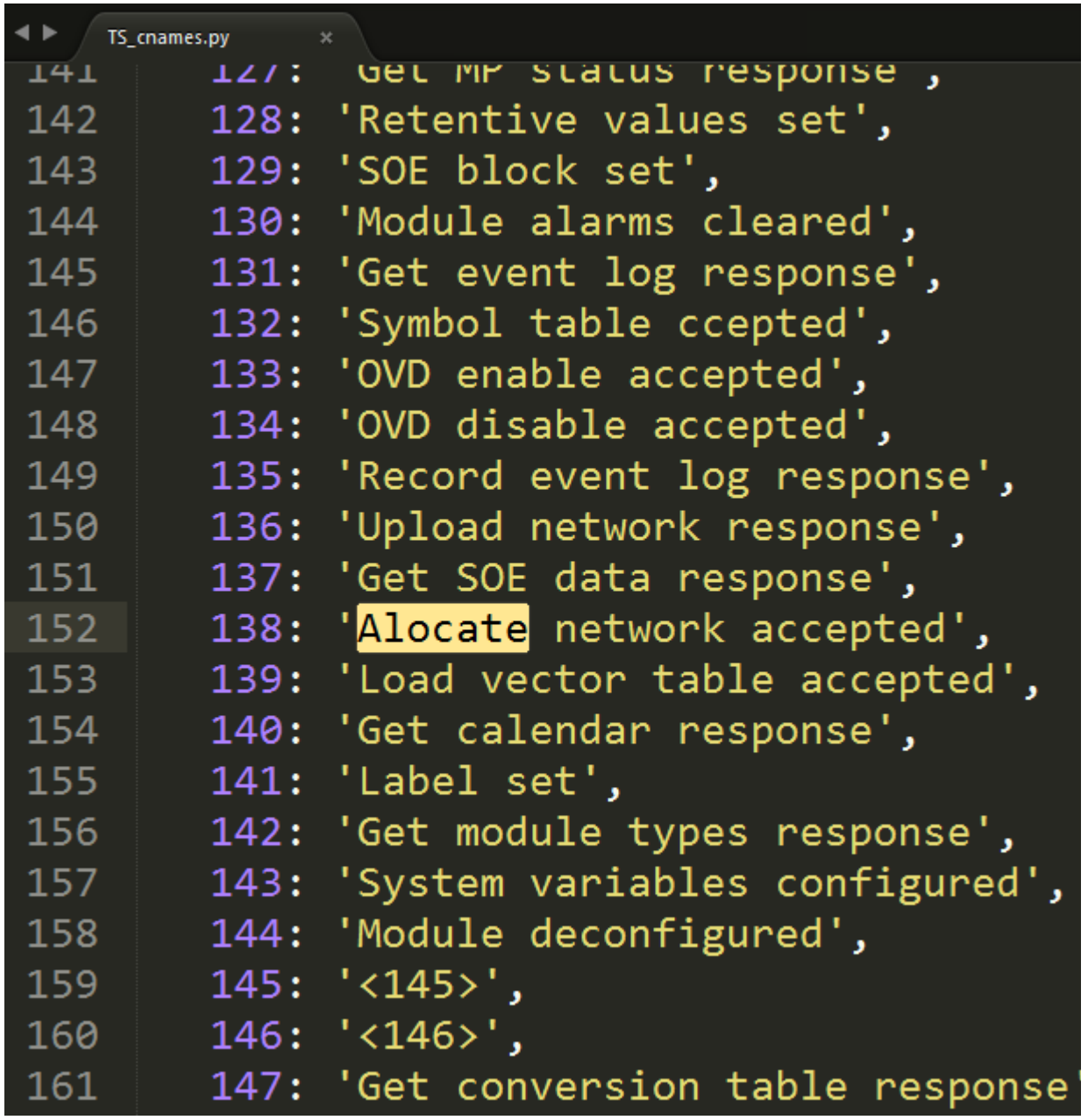

It is disappointing that these files were shared by a known party on December 23.
Our hope was that the re-usable code would remain private.
Silver lining: a refresher in Python metadata analysis?
TRITON library.zip compile times
a Nick Carr joint
@itsreallynick
Initial Build
June 25, 2016
Continued Development
August 29, 2016 -
November 14, 2016
weakref.pyc
codecs.pyc
io.pyc
posixpath.pyc
shlex.pyc
unittest/case.pyc
doctest.pyc
inspect.pyc
ssl.pyc
calendar.pyc
gettext.pyc
subprocess.pyc
crc.pyc
sh.pyc
Final TRITON Modification
August 3-4, 2017
TS_cnames.pyc
TsLow.pyc
TsBase.pyc
TsHi.pyc
stdlib compile artifacts
100
90
80
70
60
50
40
30
20
10
0
6/25/16 15:46
6/25/16 15:46
6/25/16 15:46
6/25/16 15:46
6/25/16 15:46
6/25/16 15:46
8/29/16 10:36
8/29/16 10:36
8/29/16 10:36
9/9/16 9:08
11/14/16 10:14
6/19/17 13:13
7/13/17 8:30
8/3/17 6:26
8/3/17 10:46
8/3/17 10:52
8/4/17 2:04
8/4/17 12:13

# Discrepancies

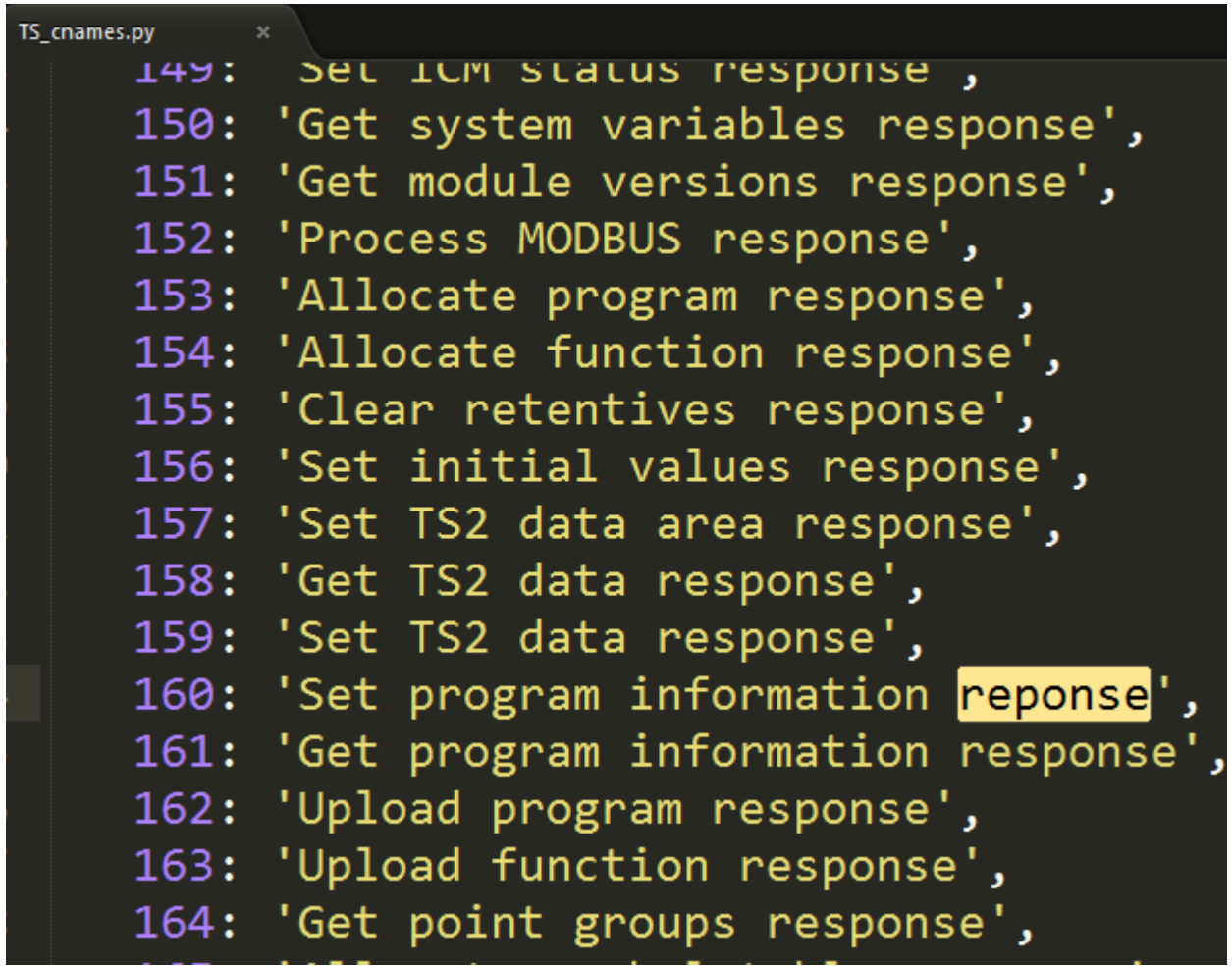
- TS_cnames.py TS_names
  - Misspellings
  - Inconsistent capitalization
  - “type” value (is it 1 or is it 2 bytes? Does it even matter?)
    - TRITON code + Source A call it 2 bytes
    - Source B + Source C think its 1 byte
- Most of the TRITON code is decent
- Annotations, comments
- Focus on proto checking & error info
- Sloppiness in TS_cnames.py library stands out as **INCONSISTENT**

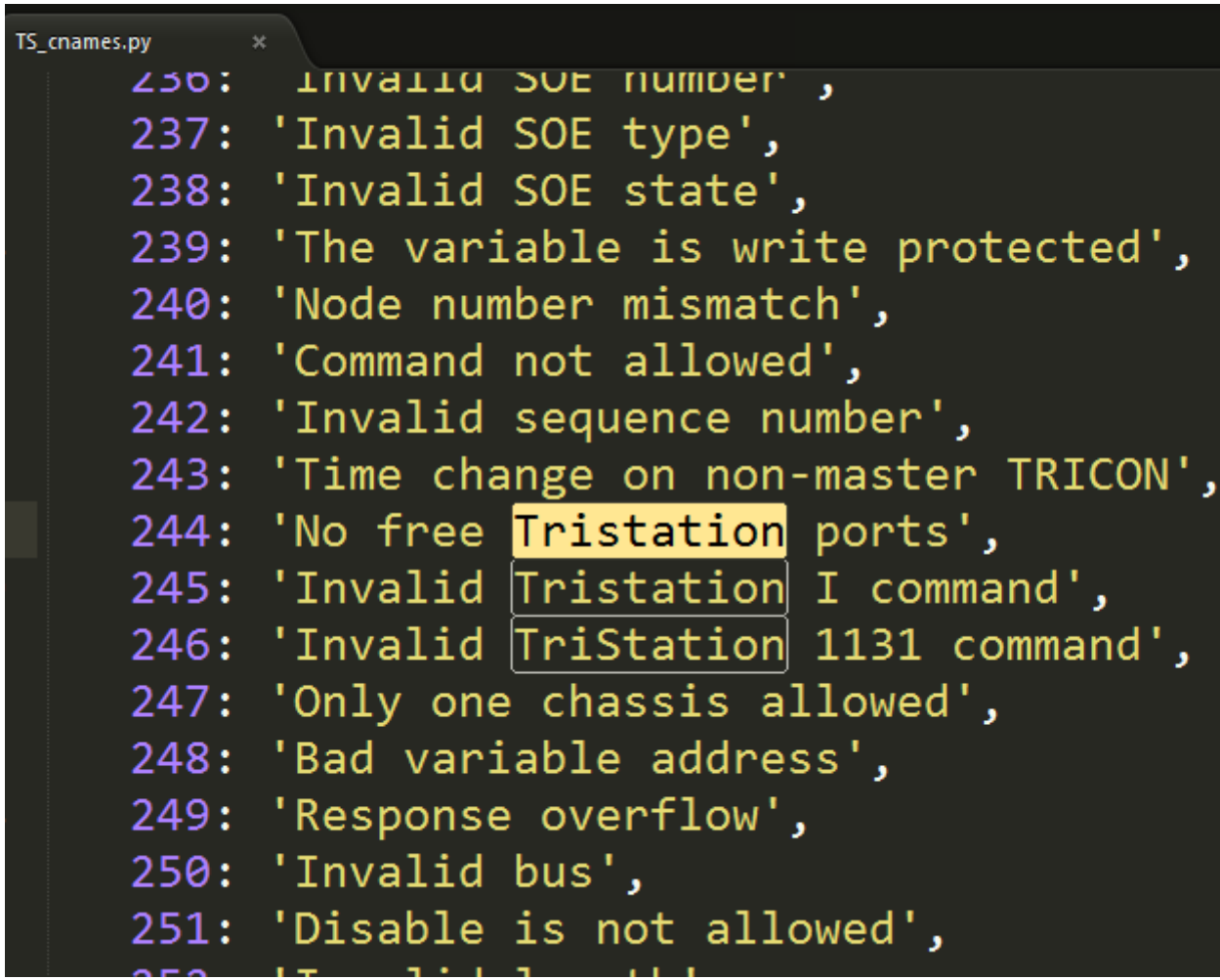
# TS_cnames.py TS_names



# TS_cnames.py


TS_cnames.py
'Get system variables response',
'Get module versions response',
'Process MODBUS response',
'Allocate program response',
'Allocate function response',
'Clear retentives response',
'Set initial values response',
'Set TS2 data area response',
'Get TS2 data response',
'Set TS2 data response',
'Set program information reponse',
'Get program information response',
'Upload program response',
'Upload function response',
'Get point groups response',


TS_cnames.py
'Invalid SOE type',
'Invalid SOE state',
'The variable is write protected',
'Node number mismatch',
'Command not allowed',
'Invalid sequence number',
'Time change on non-master TRICON',
'No free Tristation ports',
'Invalid Tristation I command',
'Invalid TriStation 1131 command',
'Only one chassis allowed',
'Bad variable address',
'Response overflow',
'Invalid bus',
'Disable is not allowed',

# VirusTotal Intelligence

virustotal intelligence

Rulesets | Notifications | Scan file | Retrohunt

| | |
|---|---|
| Job status | Finished |
| Rules | /* Paste your rules here. Ma |
| Creation time | Aug. 24, 2018, 1:59 a.m. |
| Start time | Aug. 24, 2018, 4:25 a.m. |
| Finish time | Aug. 24, 2018, 7:34 a.m. |
| Scanned data | 108.0 TB |
| Scanning speed | 4.8 GB/s |
| Matches | 0 |

- Internal FireEye tool
- Index all VT submission metadata:
  - Source UUID
  - “Location”
  - If_first_submitter
  - # of submissions etc
  - Submission name! woop woop

Awesome Possum ☐ Search ☐ Profile ☐ Docs steven.miller Logout

# Awesome Possum

Submissions ts1131.exe ? Go!

**Success!** Your search yielded **6** results!

☐ Submissions

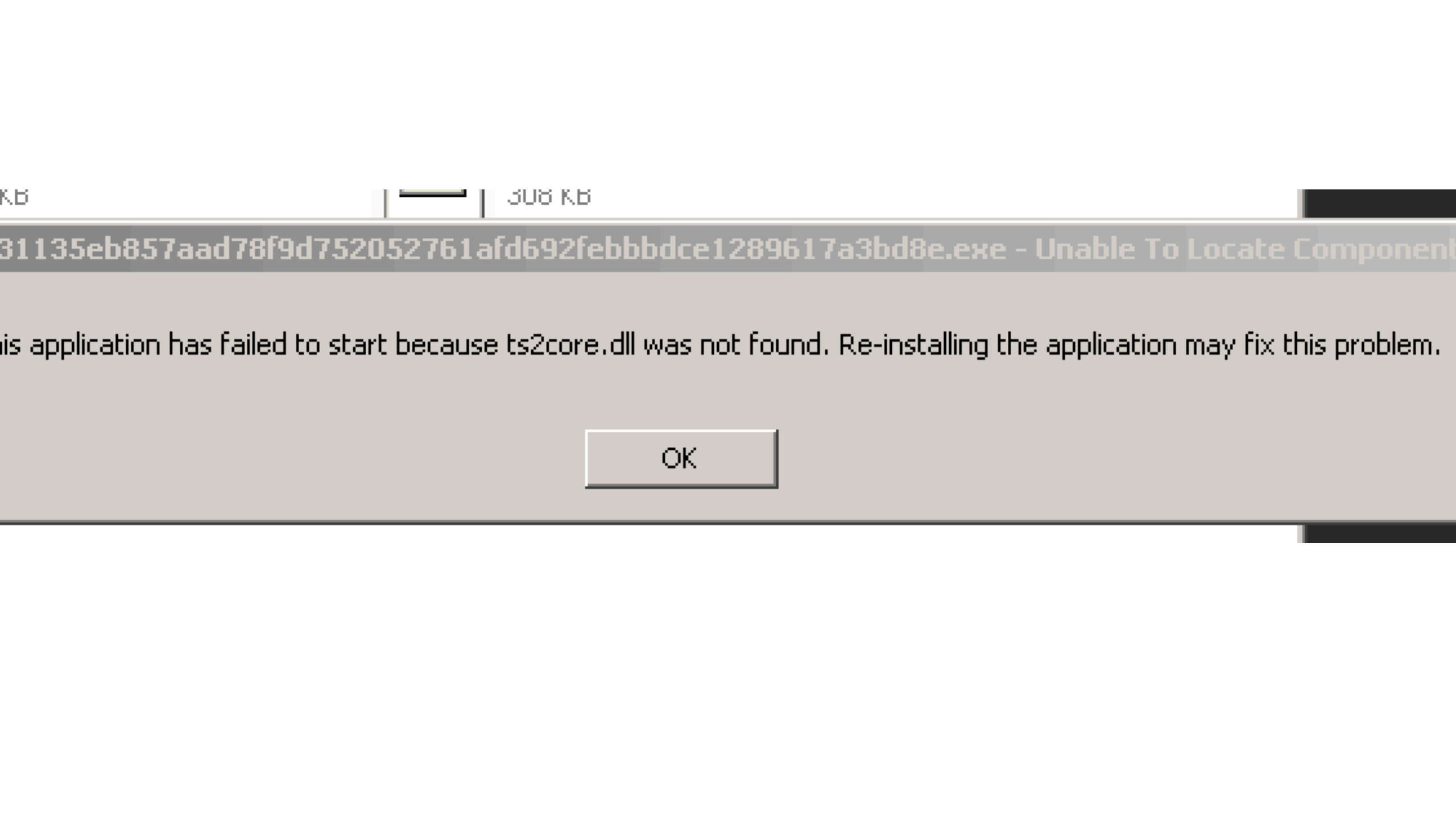
@e34e3b22 (api) US - 1 year ago
ts1131.exe (Win32 EXE)
MD5: 50cb81ae7bb274e33798bbb83f2b080f Size: 712.3 KB VT Score: 0/61
peexe signed overlay

@e34e3b22 (api) US - 1 year ago
ts1131.exe (Win32 EXE)
MD5: e85614a0eb0a40fa5161081bb680f536 Size: 702.8 KB VT Score: 0/62

@e34e3b22 (api) US - 1 year ago
ts1131.exe (Win32 EXE)
MD5: 83ada0e8cb42bd35a9abd66081472d01 Size: 712.3 KB VT Score: 0/26

@e34e3b22 (api) US - 1 year ago
ts1131.exe (Win32 EXE)
MD5: 85a1d183b5d2c0dc777083c99822eca4 Size: 308.0 KB VT Score: 0/56

@e34e3b22 (api) US - 2 years ago


KB
306 KB
31135eb857aad78f9d752052761afd692febbbdce1289617a3bd8e.exe - Unable To Locate Componen
is application has failed to start because ts2core.dll was not found. Re-installing the application may fix this problem.
OK

003315dec031135eb857aad7...
Application
713 KB

45e39e851d4a962812ef449c...
File
1,040 KB

4f7b04647ff77645a32cc8131...
Application
703 KB

6daff91c8641db042efc291f8f...
File
308 KB

cd9c0810d61075795456a721...
Application
713 KB

ts2core.dll
Application Extension
3,712 KB

ea5dbd8ff8f3dfa25b8424800...
File
312 KB

ts2iec.dll
Application Extension
359 KB

15dec031135eb857aad78f9d752052761afd692febbbdce1289617a3bd8e.exe - Unable To Locate Component

This application has failed to start because ts2ass.dll was not found. Re-installing the application may fix this problem.

OK

# File information

Identification | Details | Content | Analyses | Submissions | ITW | Comments

## Prevalence metrics

| | |
|---|---|
| **First submission** | 2017-06-04 14:49:09 |
| **Last submission** | 2017-06-04 14:49:09 |
| **Number of submissions** | 1 |
| **Distinct source submissions** | 1 |

## In-the-wild file names

LAGEVN40

LAGEVN40.DLL

LagEvn40.dll

## File bundles

This file was sent to VirusTotal in one or more file bundles, these are the sha256 hashes of those bundles.

f96edd1f421b182772886f1e4b2ea88e2f246dc4551fe869d1ddda6a15a0722e

Download file | Re-scan file | Close

# File information

Identification | Content | Analyses | Submissions | ITW | Comments

## Prevalence metrics

| | |
|---|---|
| **First submission** | 2017-06-04 14:45:27 |
| **Last submission** | 2018-06-08 13:06:08 |
| **Number of submissions** | 2 |
| **Distinct source submissions** | 2 |

## In-the-wild file names

Data1.cab

## File bundles

This file was sent to VirusTotal in one or more file bundles, these are the sha256 hashes of those bundles.

c1a9d4b967ec1a978383e0087a46f4c7222d9bcb792e6a88e2a73718633f06a4

Download file | Re-scan file | Close

# File information

Identification | Details | Content | Analyses | Submissions | ITW | Comments

**The file being studied is a compressed stream!** More specifically, it is a RAR file.

## Contained files

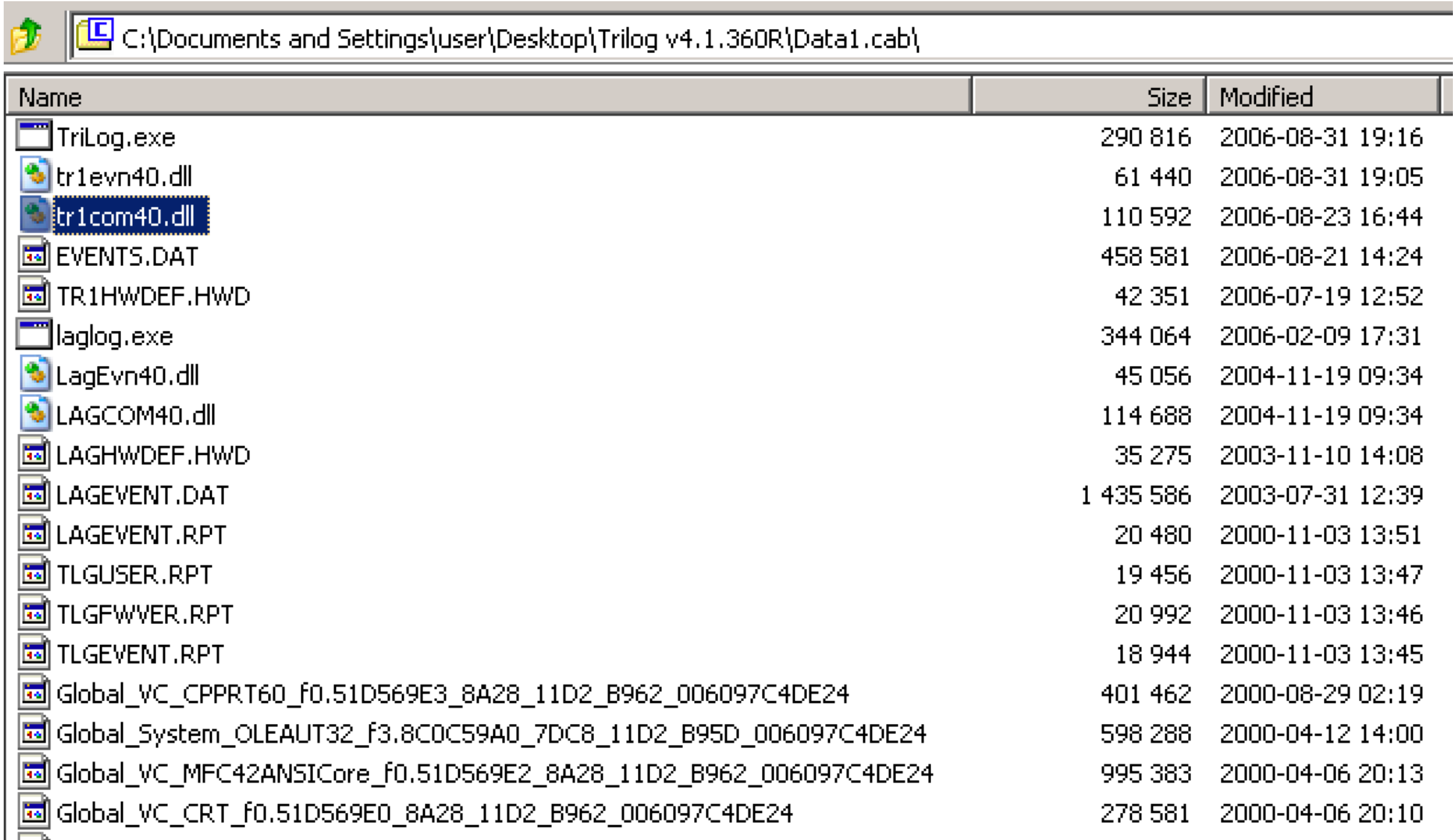
This file is a compressed stream containing 4 files.

| | | |
|---|---|---|
| [+] Trilog v4.1.360R\\Data1.cab | unknown | 1820611 Bytes |
| [+] Trilog v4.1.360R\\setup.exe | unknown | 2883380 Bytes |
| [+] Trilog v4.1.360R\\Triconex System Utilities.msi | unknown | 935128 Bytes |
| [+] Trilog v4.1.360R | directory | 0 Bytes |

## Compression metadata

| | |
|---|---|
| **Contained files** | 4 |
| **Uncompressed size** | 5639119 |
| **Highest datetime** | 2016-02-21 18:28:47 |
| **Lowest datetime** | 2006-11-14 00:00:00 |

Download file | Re-scan file | Close


C:\Documents and Settings\user\Desktop\Trilog v4.1.360R\Data1.cab\
Name
Size
Modified
TriLog.exe 290 816 2006-08-31 19:16
tr1evn40.dll 61 440 2006-08-31 19:05
tr1com40.dll 110 592 2006-08-23 16:44
EVENTS.DAT 458 581 2006-08-21 14:24
TR1HWDEF.HWD 42 351 2006-07-19 12:52
laglog.exe 344 064 2006-02-09 17:31
LagEvn40.dll 45 056 2004-11-19 09:34
LAGCOM40.dll 114 688 2004-11-19 09:34
LAGHWDEF.HWD 35 275 2003-11-10 14:08
LAGEVENT.DAT 1 435 586 2003-07-31 12:39
LAGEVENT.RPT 20 480 2000-11-03 13:51
TLGUSER.RPT 19 456 2000-11-03 13:47
TLGFWVER.RPT 20 992 2000-11-03 13:46
TLGEVENT.RPT 18 944 2000-11-03 13:45
Global_VC_CPPRT60_f0.51D569E3_8A28_11D2_B962_006097C4DE24 401 462 2000-08-29 02:19
Global_System_OLEAUT32_f3.8C0C59A0_7DC8_11D2_B95D_006097C4DE24 598 288 2000-04-12 14:00
Global_VC_MFC42ANSICore_f0.51D569E2_8A28_11D2_B962_006097C4DE24 995 383 2000-04-06 20:13
Global_VC_CRT_f0.51D569E0_8A28_11D2_B962_006097C4DE24 278 581 2000-04-06 20:10

- Export Directory
- Import Directory
- Resource Directory
- Relocation Directory

**Address Converter**
**Dependency Walker**
**Hex Editor**
**Identifier**
**Import Adder**
**Quick Disassembler**
**Rebuilder**
**Resource Editor**
**UPX Utility**

| | |
|---|---|
| Created | Wednesday 12 September 2018, 05.04.05 |
| Modified | Wednesday 23 August 2006, 16.44.46 |
| Accessed | Wednesday 12 September 2018, 05.24.51 |
| MD5 | 069247DF527A96A0E048732CA57E7D3D |
| SHA-1 | 5B41975B509D985D103B47EED798256617ACD9E5 |

| Property | Value |
|---|---|
| Comments | Tricon Support Module |
| CompanyName | Triconex Corporation |
| FileDescription | Tricon Communications Interface |
| FileVersion | 4, 2, 441, 0 |
| InternalName | TR1COM |
| LegalCopyright | Copyright © 1993-2006 Triconex Corporation |
| LegalTrademarks | |
| OriginalFilename | TricCom.DLL |
| PrivateBuild | |
| ProductName | TricCom Dynamic Link Library |
| ProductVersion | 4.2.441 |

# Tr1com.dll

# Obligatory IDA screenshot 1

```
.data:10015A30                  dd offset aWorkfileTr1str ; "$Workfile:   TR1STRS.CPP  $ $Modtime:  "...
.data:10015A34 off_10015A34     dd offset aStartDownloadA
.data:10015A34                                            ; DATA XREF: sub_10006ED0:loc_10006EE2↑r
.data:10015A34                                            ; "Start download all"
.data:10015A38                  dd offset aStartDownloadC ; "Start download change"
.data:10015A3C                  dd offset aUpdateConfigur ; "Update configuration"
.data:10015A40                  dd offset aUploadConfigur ; "Upload configuration"
.data:10015A44                  dd offset aSetIOAddresses ; "Set I/O addresses"
.data:10015A48                  dd offset aAllocateNetwor ; "Allocate network"
.data:10015A4C                  dd offset aLoadVectorTabl ; "Load vector table"
.data:10015A50                  dd offset aSetCalendar  ; "Set calendar"
.data:10015A54                  dd offset aGetCalendar  ; "Get calendar"
.data:10015A58                  dd offset aSetScanTime  ; "Set scan time"
.data:10015A5C                  dd offset aEndDownloadAll ; "End download all"
.data:10015A60                  dd offset aEndDownloadCha ; "End download change"
.data:10015A64                  dd offset aCancelDownload ; "Cancel download change"
.data:10015A68                  dd offset aAttachTricon ; "Attach TRICON"
.data:10015A6C                  dd offset aSetIOAddressLi ; "Set I/O address limits"
.data:10015A70                  dd offset aConfigureModul ; "Configure module"
.data:10015A74                  dd offset aSetMultiplePoi ; "Set multiple point values"
.data:10015A78                  dd offset aEnableAllPoint ; "Enable all points"
.data:10015A7C                  dd offset aUploadVectorTa ; "Upload vector table"
.data:10015A80                  dd offset aGetCpStatus  ; "Get CP status "
.data:10015A84                  dd offset aRunProgram   ; "Run program"
```

# Obligatory IDA screenshot 2

```
.data:10015ED6                  align 4
.data:10015ED8 aBadVariableAdd  db 'Bad variable address',0 ; DATA XREF: .data:10015D44↑o
.data:10015EED                  align 10h
.data:10015EF0 aOnlyOneChassis  db 'Only one chassis allowed',0 ; DATA XREF: .data:10015D40↑o
.data:10015F09                  align 4
.data:10015F0C aInvalidTrist_0  db 'Invalid TriStation 1131 command',0
.data:10015F0C                                          ; DATA XREF: .data:10015D3C↑o
.data:10015F2C aInvalidTristat  db 'Invalid Tristation I command',0
.data:10015F2C                                          ; DATA XREF: .data:10015D38↑o
.data:10015F49                  align 4
.data:10015F4C aNoFreeTristati  db 'No free Tristation ports',0 ; DATA XREF: .data:10015D34↑o
.data:10015F65                  align 4
.data:10015F68 aTimeChangeOnNo  db 'Time change on non-master TRICON',0
.data:10015F68                                          ; DATA XREF: .data:10015D30↑o
.data:10015F89                  align 4
.data:10015F8C aInvalidSequenc  db 'Invalid sequence number',0 ; DATA XREF: .data:10015D2C↑o
.data:10015FA4 aCommandNotAllo  db 'Command not allowed',0 ; DATA XREF: .data:10015D28↑o
.data:10015FB8 aNodeNumberMism  db 'Node number mismatch',0 ; DATA XREF: .data:10015D24↑o
.data:10015FCD                  align 10h
.data:10015FD0 aTheVariableIsW  db 'The variable is write protected',0
.data:10015FD0                                          ; DATA XREF: .data:10015D20↑o
.data:10015FF0 aInvalidSoeStat  db 'Invalid SOE state',0 ; DATA XREF: .data:10015D1C↑o
.data:10016002                  align 4
.data:10016004 aInvalidSoeType  db 'Invalid SOE type',0 ; DATA XREF: .data:10015D18↑o
.data:10016015                  align 4
```

] https://github.com/ICSrepo/TRISIS-TRITON-HATMAN/blob/master/decompiled_code/library/TS_cnames.py

```
30  TS_names = {-1: 'Not set',
31     0: 'Start download all',
32     1: 'Start download change',
33     2: 'Update configuration',
34     3: 'Upload configuration',
35     4: 'Set I/O addresses',
36     5: 'Allocate network',
37     6: 'Load vector table',
38     7: 'Set calendar',
39     8: 'Get calendar',
40     9: 'Set scan time',
41     10: 'End download all',
42     11: 'End download change',
43     12: 'Cancel download change',
44     13: 'Attach TRICON',
45     14: 'Set I/O address limits',
46     15: 'Configure module',
47     16: 'Set multiple point values',
48     17: 'Enable all points',
49     18: 'Upload vector table',
50     19: 'Get CP status ',
51     20: 'Run program',
52     21: 'Halt program',
53     22: 'Pause program',
```

```
67 6C  status..Do singl
70 72  e scan..Pause pr
72 6F  ogram...Halt pro
6F 67  gram....Run prog
74 75  ram.Get CP statu
74 6F  s ..Upload vecto
20 61  r table.Enable a
74 20  ll points...Set
20 76  multiple point v
75 72  alues...Configur
74 20  e module....Set
6D 69  I/O address limi
43 4F  ts..Attach TRICO
6E 6C  N...Cancel downl
64 20  oad change..End
65 00  download change.
6C 6C  End download all
69 6D  ....Set scan tim
61 72  e...Get calendar
61 72  ....Set calendar
72 20  ....Load vector
74 65  table...Allocate
74 20   network....Set
00 00  I/O addresses...
72 61  Upload configura
20 63  tion....Update c
00 00  onfiguration....
20 63  Start download c
64 6F  hange...Start do
```

Trilog v4.1.360R

Firmware Managerv2.1.215 (Tricon10.5.4)

TriStation 1131 v4.9.0 (build 117)

| PE DLL Name | Compile Date | Imphash | Reference CPP Strings Code |
|---|---|---|---|
| Lagcom40.dll | 11/19/04 | 50e[illegible]3be9c0da787a553[illegible]17ceee5aa1 | LAGSTRS.CPP $ $Modtime: Jul 21 1999 17:17:26 $ $Revision: 1.0 |
| Tr1com40.dll | 8/23/0[illegible] | e130020b087[illegible]9393fe3fb3ec8f79df1 | TR1STRS.CPP $ $Modtime: May 16 2006 09:55:20 $ $Revision: 1.4 |
| | | | |
| Triccom.dll | 7/23/08 | 63a7188c63035e985c162f64132d2da2 | TR1STRS.CPP $ $Modtime: May 16 2006 09:55:20 $ $Revision: 1.4 |
| Tr1com.dll | 4/27/11 | 63a7188c63035e985c162f64132d2da2 | TR1STRS.CPP $ $Modtime: May 16 2006 09:55:20 $ $Revision: 1.4 |
| Triccom.dll | 4/27/11 | 63a7188c63035e985c162f64132d2da2 | TR1STRS.CPP $ $Modtime: May 16 2006 09:55:20 $ $Revision: 1.4 |
| | | | |
| Tridcom.dll | 7/23/08 | 8493239c5bafe58979fd5baaff57d19d | LAGSTRS.CPP $ $Modtime: Jul 21 1999 17:17:26 $ $Revision: 1.0 |
| Tridcom.dll | 9/29/10 | 8493239c5bafe58979fd5baaff57d19d | LAGSTRS.CPP $ $Modtime: Jul 21 1999 17:17:26 $ $Revision: 1.0 |
| Lagcom.dll | 4/27/11 | 8493239c5bafe58979fd5baaff57d19d | LAGSTRS.CPP $ $Modtime: Jul 21 1999 17:17:26 $ $Revision: 1.0 |
| Tridcom.dll | 4/27/11 | 8493239c5bafe58979fd5baaff57d19d | LAGSTRS.CPP $ $Modtime: Jul 21 1999 17:17:26 $ $Revision: 1.0 |

LAGSTRS.CPP – probably for **Trident** controller (older? Version of Tricon?)

TR1STRS.CPP – probably for **Tricon** controller

# TS_cnames.py developer annotations (?)

```
215: '<215>',
216: 'Bad Index for a module',
217: 'Module address is invalid',
218: '<218>',
219: '<219>',
220: 'Bad offset for an I/O point',
221: 'Invalid point type',
222: 'Invalid Point Location',
223: 'Program name is invalid',
224: '<224>',
225: '<225>',
226: '<226>',
```

```
72: 'Allocate multiple functi
73: 'Get node number',
74: 'Get symbol table',
75: 'Unk75',
76: 'Unk76',
77: 'Unk77',
78: 'Unk78',
79: 'Unk79',
80: 'Go to DOWNLOAD mode',
81: 'Unk81',
83: 'Unk83',
100: 'Command rejected',
101: 'Download all permitted'
```

# So what?

- Hand wavey suspicion of RE/DEV -> *conclusive evidence of what was RE'd*
- No vis into TriStation -> *some viz into protocol*
- TriStation implementation in TRITON was INCOMPLETE
- TRITON operators had a tough time using it ITW
  - Firmware, ref: UsnJrnl, USN "Change Journal"
  - Communications, ref: ping pong ping pong
- Experimentation is probs what got them caught

# Acking Public TRITON research

- A lot of excellent research by a variety of folks, notably:
  - FireEye
  - Nozomi Networks
  - Dragos
  - Accenture
  - Midnight Blue Labs
  - ICS-CERT
  - Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
  - https://github.com/ICSrepo/TRISIS-TRITON-HATMAN
  - https://www.shodan.io/search?query=screenshot.label%3Aics

# Thank you!
# Qs?

smiller@fireeye.com

@stvemillertime