SCADASTRANGELOVE.ORG

# TECHNIQUES OF ATTACKING REAL SCADA & ICS SYSTEMS

All pictures are taken from
Dr StrangeLove movie

# SCADAStrangeLove.org

- Group of security researchers focused on ICS/SCADA

to save Humanity from industrial disaster and to keep Purity Of Essence

Sergey Gordeychik
Ilya Karpov
Artem Chaykin
Dmitry Efanov
Andrey Medov
Alexander Zaitsev
Dmitry Sklyarov
Tlyapov

Gleb Gritsai
Sergey Bobrov
Yuriy Dyachenko
Yuri Goltsev
Sergey Scherbel
Dmitry Serebryannikov
Alexander Timorin

Denis Baranov

Sergey Drozdov
Vladimir Kochetkov
Timur Yunusov
Dmitry Nagibin
Alexander

POSITIVE TECHNOLOGIES

# Our goals (for porfit)

- Goals

  to automate security assessment of ICS platforms and environment

- Objectives

  to understand system

  to assess built-in security features

  to create security audit/hardening guides

  to automate process

Vulnerabilities – waste production

# Our goals (for fun)

- Goal

  to create  PoC of Stuxnet-style attack

- Initial conditions

  common ICS components and configuration

  common ICS security tools

  only ICS components weakness

  vulnerabilities by SCADA StrangeLove team

# Agenda

- Tilting at windmills: ICS pentest project management

- Playing with networks

- Rooting the PLC: don't even try

- OS/DB/Application

- I'm the Lord of the SCADA

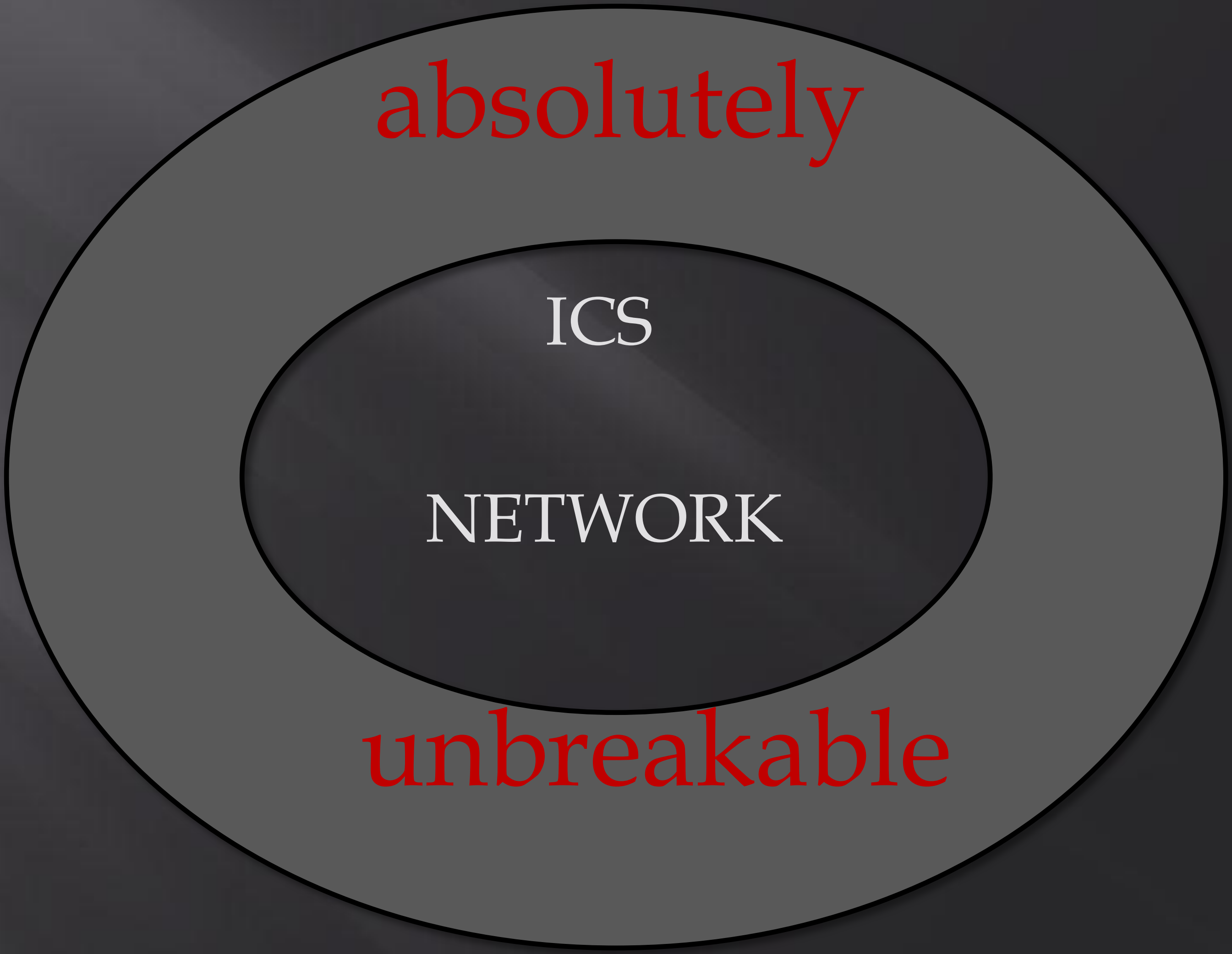
- Hunting the operator: ICS network "forensic“

- Jumping to business level

# Playing with networks

# What is a typical ICS network by design ?

# You think it looks like this ?


absolutely

ICS

NETWORK

unbreakable

# NO WAY !

- 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 happened )
- Low physical security
- … and
- Industrial protocols

# NO WAY !

- ~~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 happened )~~
- ~~Low physical security~~
- … and
- Industrial protocols

# “Sir! I have a plan…”

# How ICS protocols live in the network ?

- Full expanse
- Not blocked by firewalls/switches
- Accessible between LAN segments
- Works from data link to application layers
- Easy for detecting
- Easy for intercepting and analyzing
  ( but not all! )

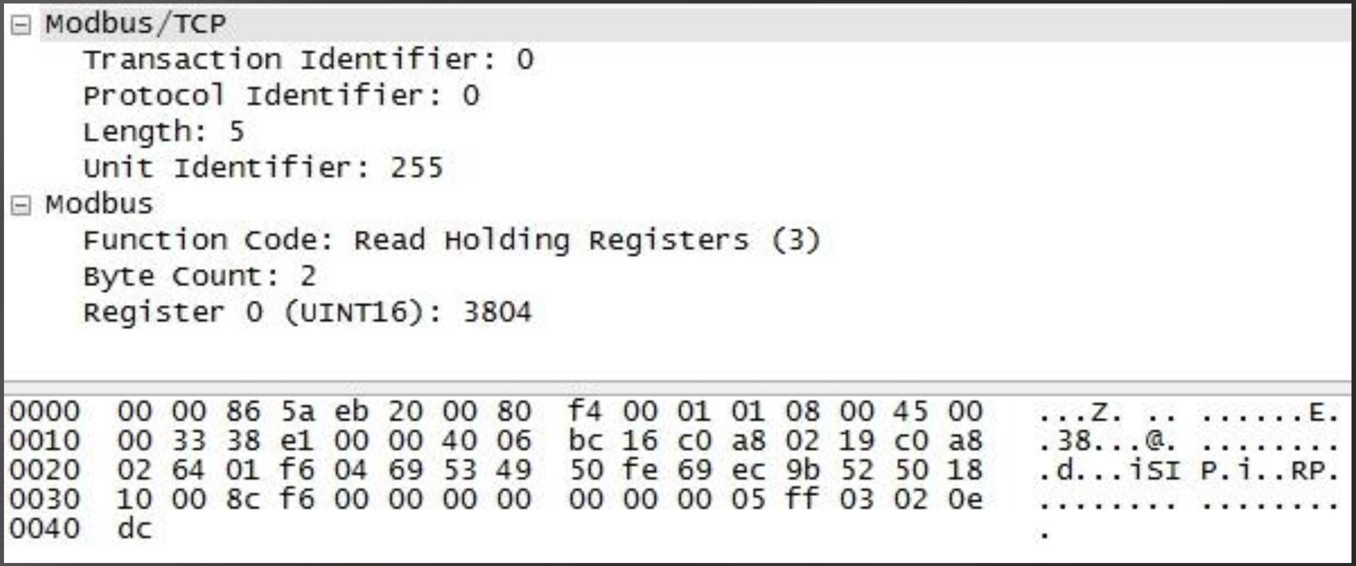
And what we know about protocols ?

# Popular industrial protocols

- Modbus
- Profinet family
- DNP3
- IEC 61850-8-1 ( MMS )
- IEC 60870-5-104 ( IEC 104 )
- Siemens S7
- … and much more

And most of them INSECURE BY DESIGN

# Modbus

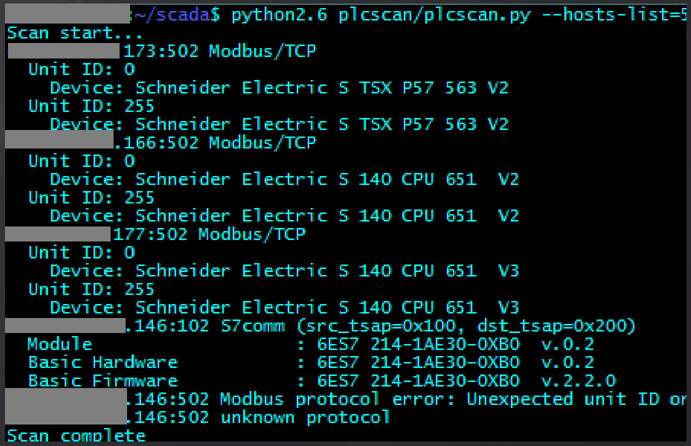


```
⊟ Modbus/TCP
    Transaction Identifier: 0
    Protocol Identifier: 0
    Length: 5
    Unit Identifier: 255
⊟ Modbus
    Function Code: Read Holding Registers (3)
    Byte Count: 2
    Register 0 (UINT16): 3804
```

```
0000   00 00 86 5a eb 20 00 80  f4 00 01 01 08 00 45 00   ...Z. .. ......E.
0010   00 33 38 e1 00 00 40 06  bc 16 c0 a8 02 19 c0 a8   .38...@. ........
0020   02 64 01 f6 04 69 53 49  50 fe 69 ec 9b 52 50 18   .d...iSI P.i..RP.
0030   10 00 8c f6 00 00 00 00  00 00 00 05 ff 03 02 0e   ........ ........
0040   dc                                                 .
```

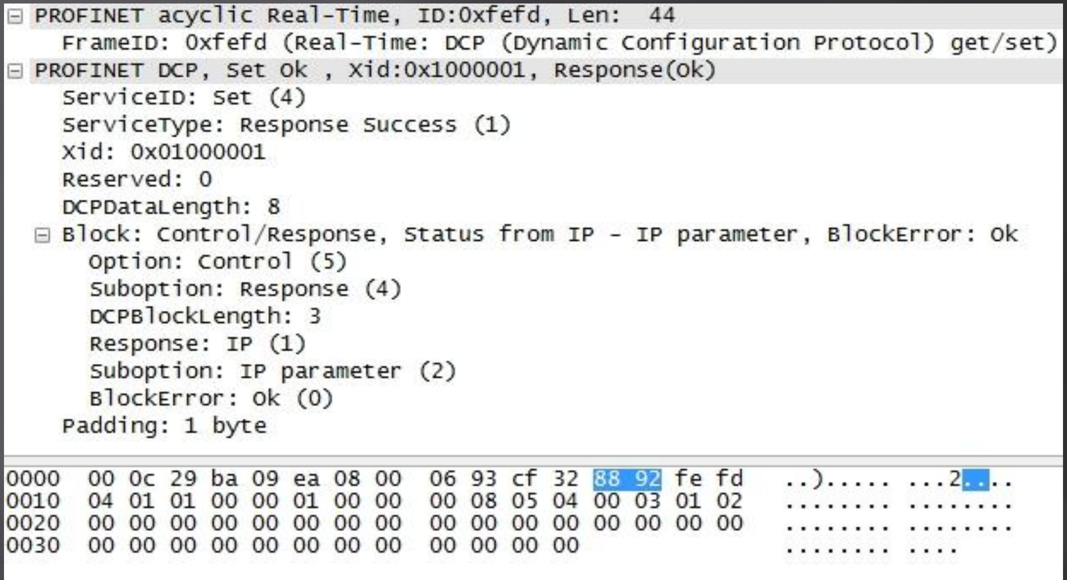
# Modbus

- http://www.modbus.org/
- Diagnostic functions
- Read/Write data/registers/tags
- Read/Write files
- Toolkit: PLCSCAN by Dmitry Efanov http://code.google.com/p/plcscan/



```
[redacted]:~/scada$ python2.6 plcscan/plcscan.py --hosts-list=5
Scan start...
[redacted]173:502 Modbus/TCP
  Unit ID: 0
    Device: Schneider Electric S TSX P57 563 V2
  Unit ID: 255
    Device: Schneider Electric S TSX P57 563 V2
[redacted].166:502 Modbus/TCP
  Unit ID: 0
    Device: Schneider Electric S 140 CPU 651  V2
  Unit ID: 255
    Device: Schneider Electric S 140 CPU 651  V2
[redacted]177:502 Modbus/TCP
  Unit ID: 0
    Device: Schneider Electric S 140 CPU 651  V3
  Unit ID: 255
    Device: Schneider Electric S 140 CPU 651  V3
[redacted].146:102 S7comm (src_tsap=0x100, dst_tsap=0x200)
  Module                        : 6ES7 214-1AE30-0XB0  v.0.2
  Basic Hardware                : 6ES7 214-1AE30-0XB0  v.0.2
  Basic Firmware                : 6ES7 214-1AE30-0XB0  v.2.2.0
[redacted].146:502 Modbus protocol error: Unexpected unit ID or
           .146:502 unknown protocol
Scan complete
```

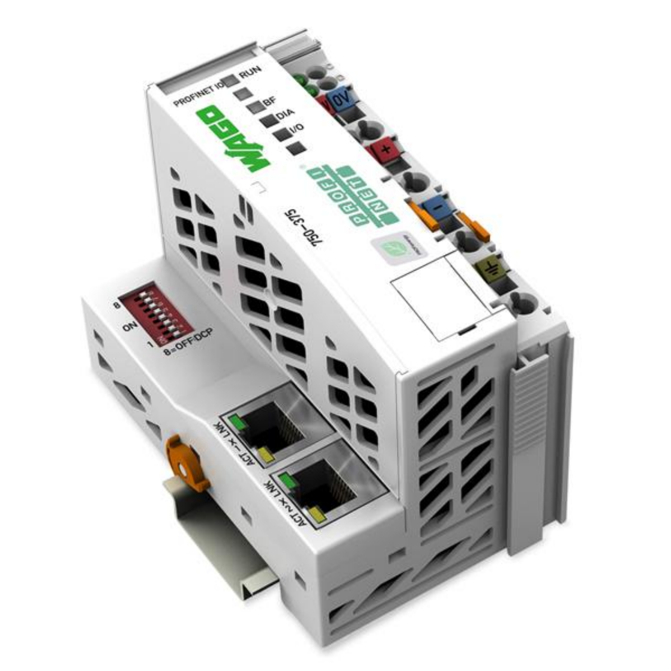
# Profinet family



```
⊟ PROFINET acyclic Real-Time, ID:0xfefd, Len:  44
    FrameID: 0xfefd (Real-Time: DCP (Dynamic Configuration Protocol) get/set)
⊟ PROFINET DCP, Set Ok , Xid:0x1000001, Response(Ok)
    ServiceID: Set (4)
    ServiceType: Response Success (1)
    Xid: 0x01000001
    Reserved: 0
    DCPDataLength: 8
  ⊟ Block: Control/Response, Status from IP - IP parameter, BlockError: Ok
      Option: Control (5)
      Suboption: Response (4)
      DCPBlockLength: 3
      Response: IP (1)
      Suboption: IP parameter (2)
      BlockError: Ok (0)
    Padding: 1 byte

0000  00 0c 29 ba 09 ea 08 00  06 93 cf 32 88 92 fe fd   ..)..... ...2....
0010  04 01 01 00 00 01 00 00  00 08 05 04 00 03 01 02   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

IEC 61158, IEC 61784



# Profinet family

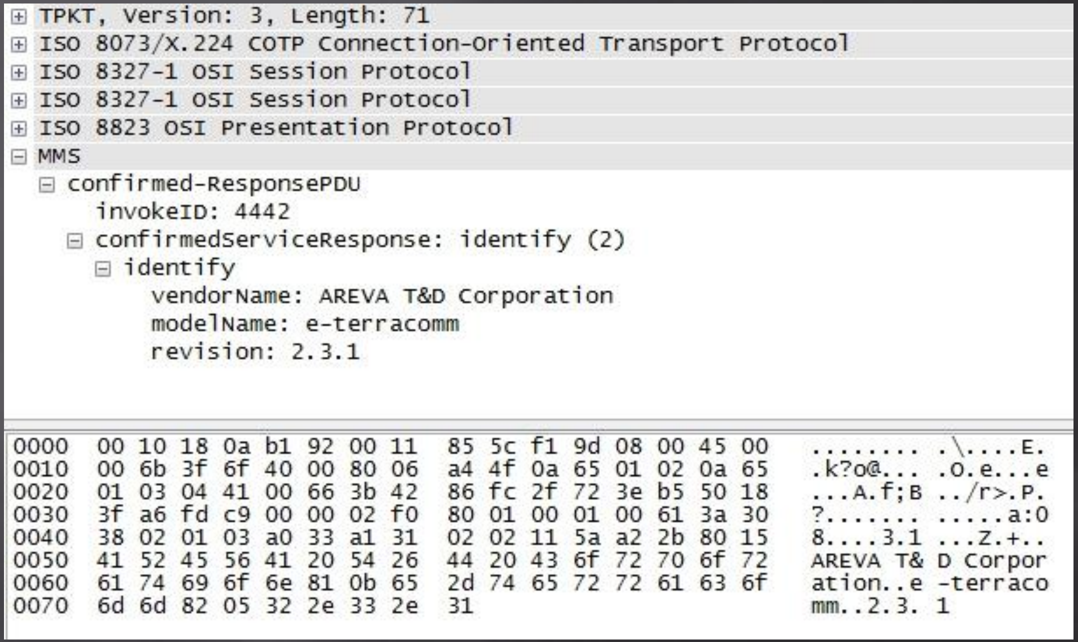
- Profinet CBA/IO/PTCP/DCP
- Ethernet type **0x8892**
- Exchange data in real-time cycles
- Multicast discovery devices and stations
- No encryption, no auth, no security
- We can change settings: name of the station, ip, netmask, gateway
- We can simulate and real DoS of PLC, HMI
- Toolkit: WWW

# DNP3

- http://www.dnp.org
- Spread and popular
- Useful info: http://www.digitalbond.com/scadapedia/protocols/dnp3/

  http://blog.iec61850.com/search/label/DNP3
- Secure DNP3 specification
- Toolkit: coming soon ….

# IEC 61850-8-1 ( MMS )



```
⊞ TPKT, Version: 3, Length: 71
⊞ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
⊞ ISO 8327-1 OSI Session Protocol
⊞ ISO 8327-1 OSI Session Protocol
⊞ ISO 8823 OSI Presentation Protocol
⊟ MMS
  ⊟ confirmed-ResponsePDU
      invokeID: 4442
    ⊟ confirmedServiceResponse: identify (2)
      ⊟ identify
          vendorName: AREVA T&D Corporation
          modelName: e-terracomm
          revision: 2.3.1
```

```
0000   00 10 18 0a b1 92 00 11  85 5c f1 9d 08 00 45 00   ........ .\....E.
0010   00 6b 3f 6f 40 00 80 06  a4 4f 0a 65 01 02 0a 65   .k?o@... .O.e...e
0020   01 03 04 41 00 66 3b 42  86 fc 2f 72 3e b5 50 18   ...A.f;B ../r>.P.
0030   3f a6 fd c9 00 00 02 f0  80 01 00 01 00 61 3a 30   ?....... .....a:0
0040   38 02 01 03 a0 33 a1 31  02 02 11 5a a2 2b 80 15   8....3.1 ...Z.+..
0050   41 52 45 56 41 20 54 26  44 20 43 6f 72 70 6f 72   AREVA T& D Corpor
0060   61 74 69 6f 6e 81 0b 65  2d 74 65 72 72 61 63 6f   ation..e -terraco
0070   6d 6d 82 05 32 2e 33 2e  31                        mm..2.3. 1
```

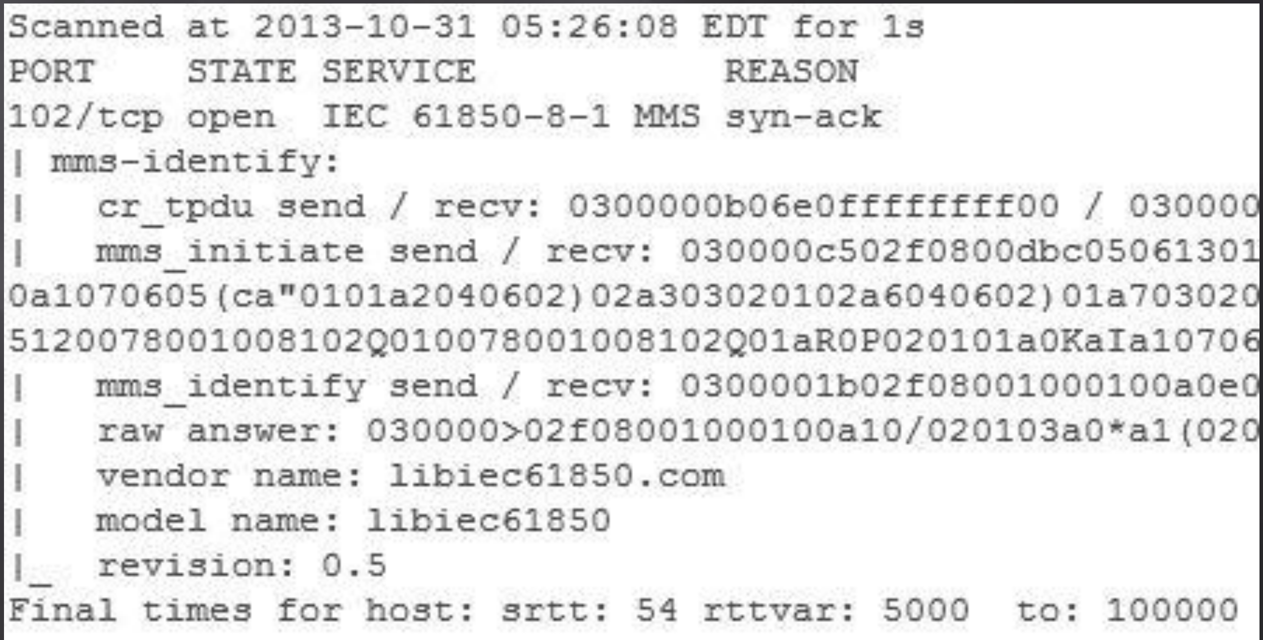
Manufacturing
Message
Specification

# IEC 61850-8-1 ( MMS )

- ISO 9506-1:2003
- Based on ISO-TSAP TCP/102
- Read/write PLC tags, variables, domains (large unstructured data, i.e. code)
- Start/Stop/Rewrite firmware of PLC
- Read/Write/Del files and dirs
- Poor security mechanism: simply methods whitelist
- No auth, no encryption
- Toolkit: python and nmap scripts

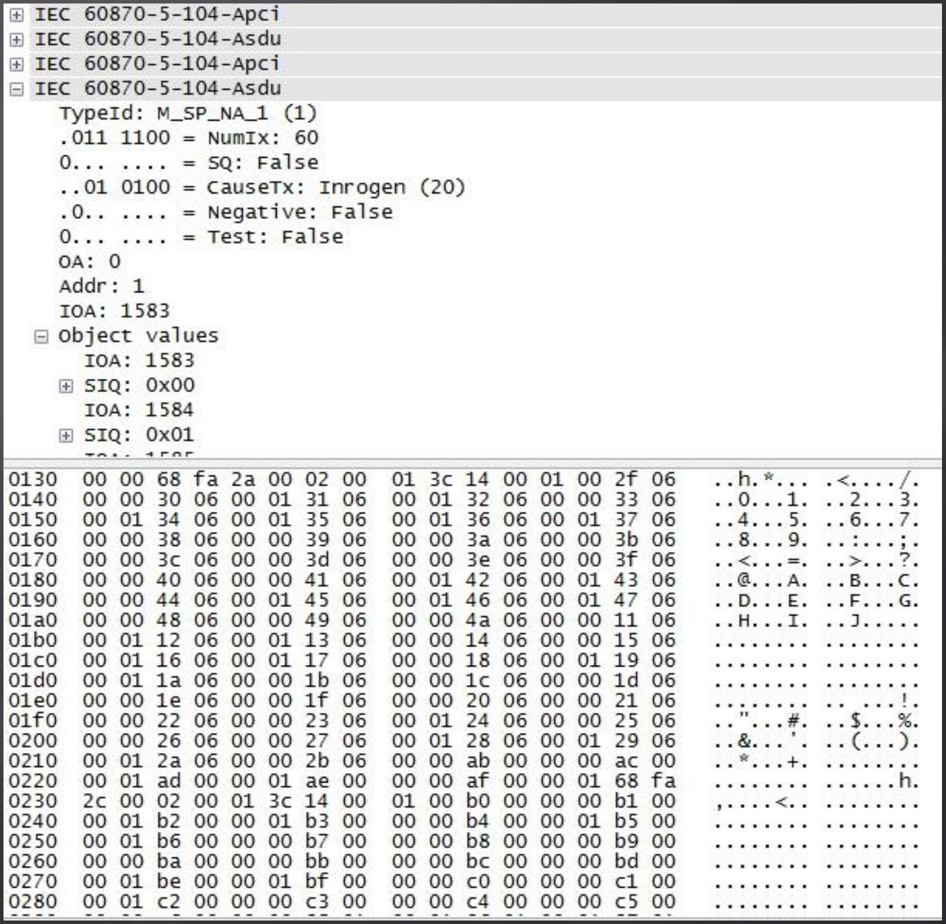
# IEC 61850-8-1 ( MMS )

- Python identify script: WWW
- Nmap identify script: WWW



```
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
5120078001008102Q0100780010081O2Q01aR0P020101a0KaIa10706
|   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 60870-5-104 ( IEC 104 )



```
⊞ IEC 60870-5-104-Apci
⊞ IEC 60870-5-104-Asdu
⊞ IEC 60870-5-104-Apci
⊟ IEC 60870-5-104-Asdu
    TypeId: M_SP_NA_1 (1)
    .011 1100 = NumIx: 60
    0... .... = SQ: False
    ..01 0100 = CauseTx: Inrogen (20)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 1
    IOA: 1583
  ⊟ Object values
      IOA: 1583
    ⊞ SIQ: 0x00
      IOA: 1584
    ⊞ SIQ: 0x01
```

```
0130  00 00 68 fa 2a 00 02 00  01 3c 14 00 01 00 2f 06   ..h.*... .<..../.
0140  00 00 30 06 00 01 31 06  00 01 32 06 00 00 33 06   ..0...1. ..2...3.
0150  00 01 34 06 00 01 35 06  00 01 36 06 00 01 37 06   ..4...5. ..6...7.
0160  00 00 38 06 00 00 39 06  00 00 3a 06 00 00 3b 06   ..8...9. ..:...;.
0170  00 00 3c 06 00 00 3d 06  00 00 3e 06 00 00 3f 06   ..<...=. ..>...?.
0180  00 00 40 06 00 00 41 06  00 01 42 06 00 01 43 06   ..@...A. ..B...C.
0190  00 00 44 06 00 01 45 06  00 01 46 06 00 01 47 06   ..D...E. ..F...G.
01a0  00 00 48 06 00 00 49 06  00 00 4a 06 00 00 11 06   ..H...I. ..J.....
01b0  00 01 12 06 00 01 13 06  00 00 14 06 00 00 15 06   ........ ........
01c0  00 01 16 06 00 01 17 06  00 00 18 06 00 01 19 06   ........ ........
01d0  00 01 1a 06 00 00 1b 06  00 00 1c 06 00 00 1d 06   ........ ........
01e0  00 00 1e 06 00 00 1f 06  00 00 20 06 00 00 21 06   ........ .. ...!.
01f0  00 00 22 06 00 00 23 06  00 01 24 06 00 00 25 06   .."...#. ..$...%.
0200  00 00 26 06 00 00 27 06  00 01 28 06 00 01 29 06   ..&...'. ..(...).
0210  00 01 2a 06 00 00 2b 06  00 00 ab 00 00 00 ac 00   ..*...+. ........
0220  00 01 ad 00 00 01 ae 00  00 00 af 00 00 01 68 fa   ........ ......h.
0230  2c 00 02 00 01 3c 14 00  01 00 b0 00 00 00 b1 00   ,....<.. ........
0240  00 01 b2 00 00 01 b3 00  00 00 b4 00 00 01 b5 00   ........ ........
0250  00 01 b6 00 00 00 b7 00  00 00 b8 00 00 00 b9 00   ........ ........
0260  00 00 ba 00 00 00 bb 00  00 00 bc 00 00 00 bd 00   ........ ........
0270  00 01 be 00 00 01 bf 00  00 00 c0 00 00 00 c1 00   ........ ........
0280  00 01 c2 00 00 00 c3 00  00 00 c4 00 00 00 c5 00   ........ ........
```

TCP/2404

HEADER:

1st byte: 0x68

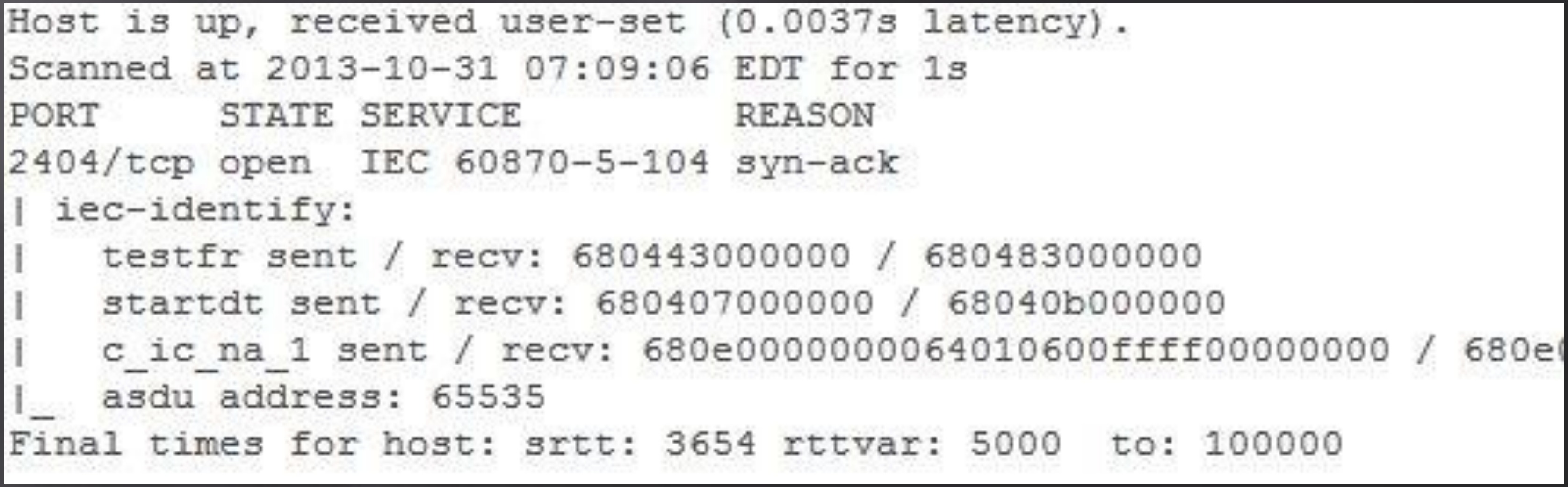
2nd byte: APDU len

# IEC 60870-5-104 ( IEC 104 )

- 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.
- No auth, no encryption
- Poor security mechanism: ip address whitelist
- Toolkit: python and nmap scripts

# IEC 60870-5-104 ( IEC 104 )

- Python identify script: WWW
- Nmap identify script: WWW



```
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
```

# Siemens S7

- I love this protocol!
- Proprietary communication protocol supported by Siemens SCADA Software, PLC, HMI
- We can: detect protocol, extract some useful info (device serial number, type of station, firmware info etc.), extract and  bruteforce (thanks to JtR community) authentication challenge-response hashes
- http://www.slideshare.net/phdays/timorin-alexander-efanov-dmitry

# Siemens S7

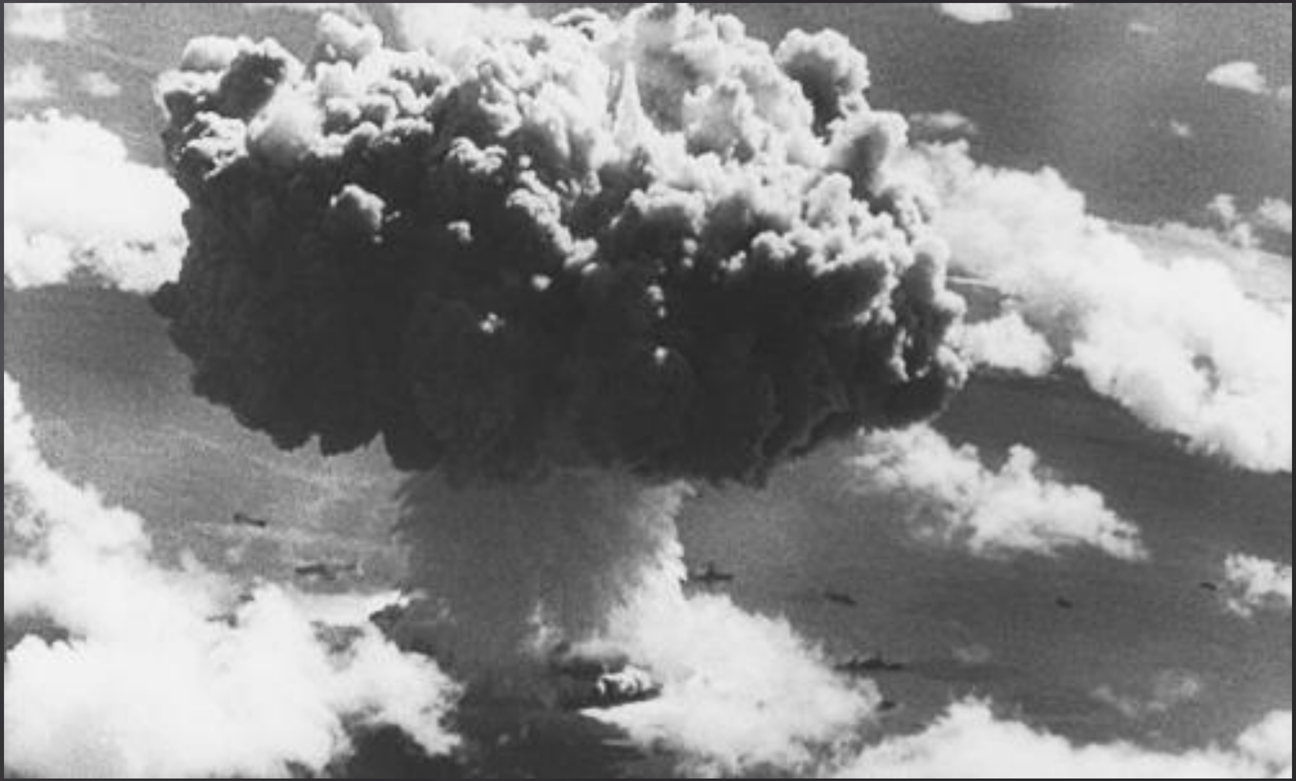
- Toolkit:

http://code.google.com/p/scada-tools/

https://code.google.com/p/plcscan/

# Wanna play with protocols ?

Welcome to our workshop!

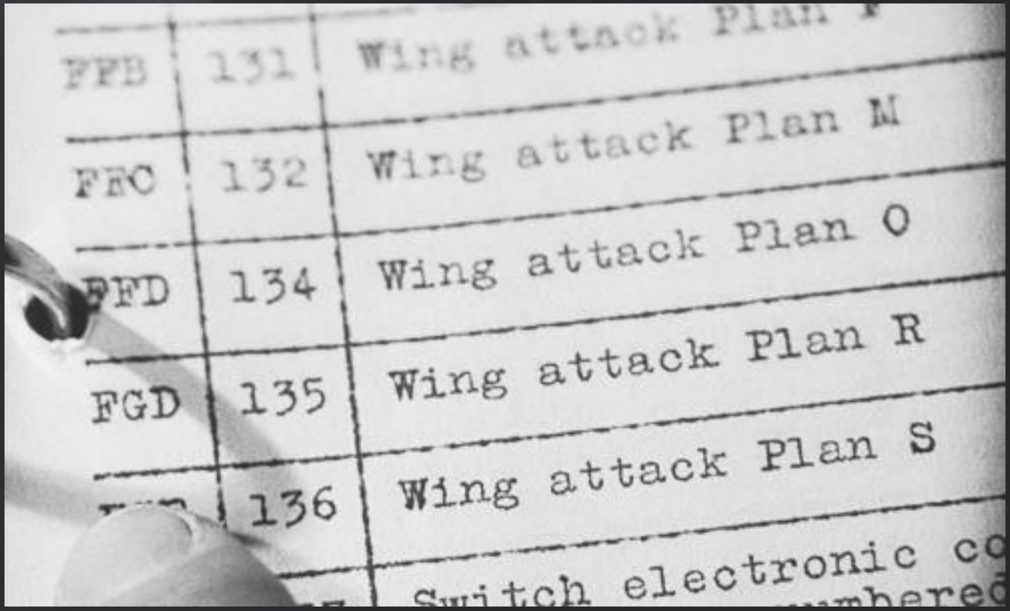


# Rooting the PLC: don't even try

# Ways to takeover PLC

- Pwn OS (often VxWorks, QNX)
- Reverse internal architecture
- Find bugs in services
- Snatch device

BUT FOR WHAT ?



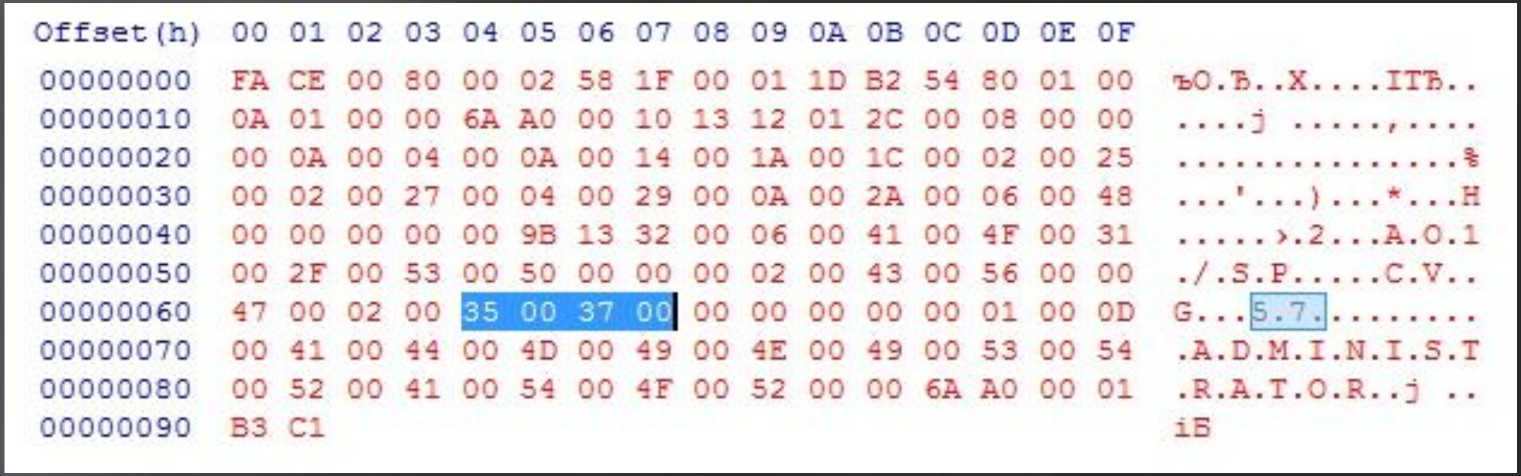
| | | |
|---|---|---|
| FFB | 131 | Wing attack Plan F |
| FFC | 132 | Wing attack Plan M |
| FFD | 134 | Wing attack Plan O |
| FGD | 135 | Wing attack Plan R |
| [illegible] | 136 | Wing attack Plan S |
| | | Switch electronic co<br>numbered |

# Use your knowledge about protocols

- It is a universal and complex approach
- You can:
  - detect devices and protocols
  - monitor state, commands, exchanging data
  - inject, modify, replay packets in real-time
- Because most of them INSECURE BY DESING

Real example ?

# 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  ъO.Ђ..X....IТЂ..
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                                            iБ
```

Simple UDP packet that set “speed” of turbine to 57 (min=1, max=100)

# What will happen if you send another packet ?

# Yes, you’re right