POSITIVE TECHNOLOGIES

POSITIVE TECHNOLOGIES

Industrial protocols for pentesters

Timorin Alexander
Efanov Dmitry

Positive Technologies

PHDays III

Who We Are

Timorin Alexander

- Lead specialist of penetration testing team at Positive Technologies
- Main interests: penetration testing, SCADA systems, industrial protocols, password cracking
- atimorin@ptsecurity.ru

Who We Are

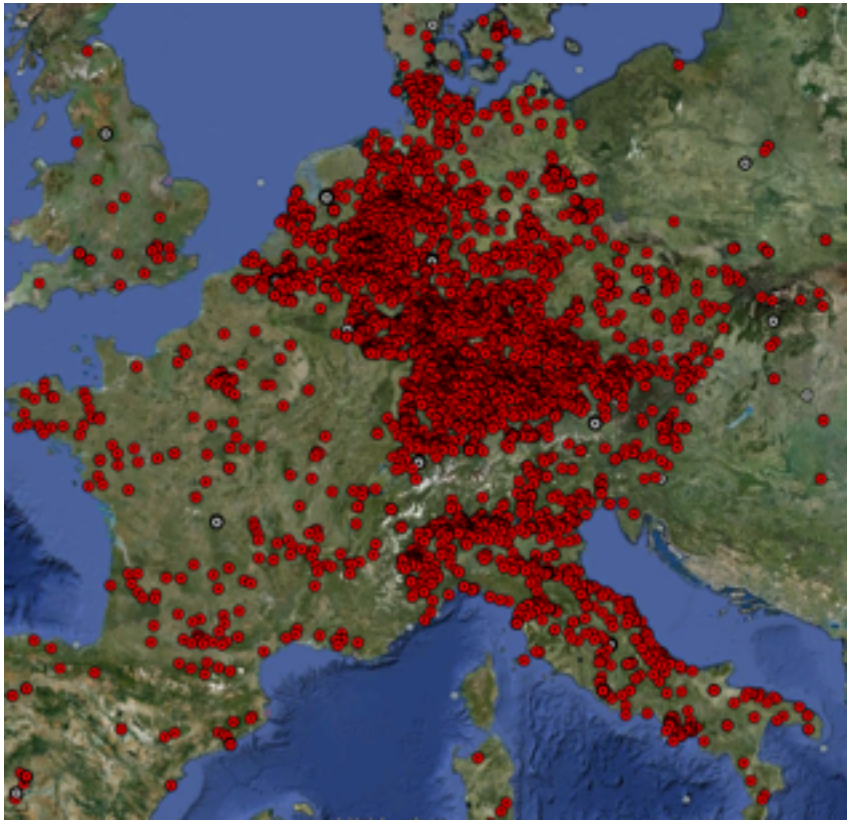
Efanov Dmitry

- Lead specialist of security development team at Positive Technologies

- Main interests: penetration testing, network protocols and hex-numbers

- defanov@ptsecurity.ru

ICS

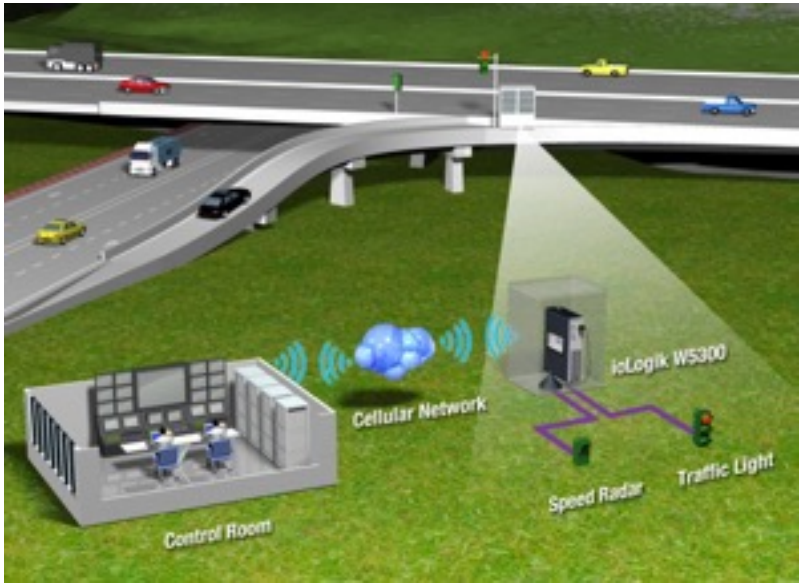


Industrial Control System

ICS in the World

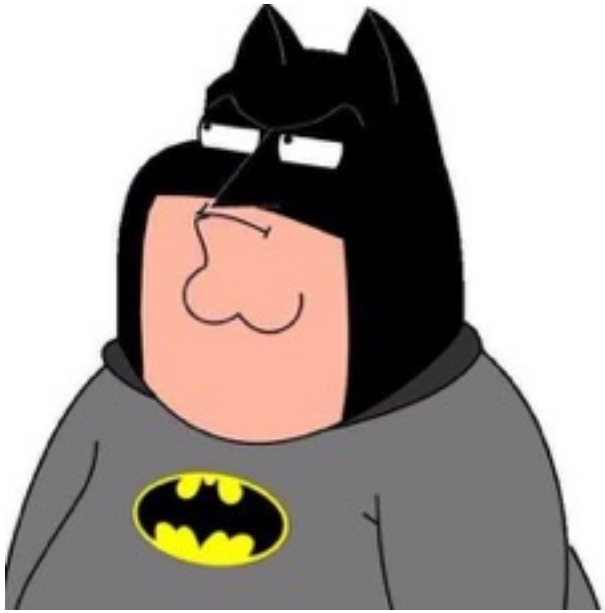
ICS in the World

ICS in the World



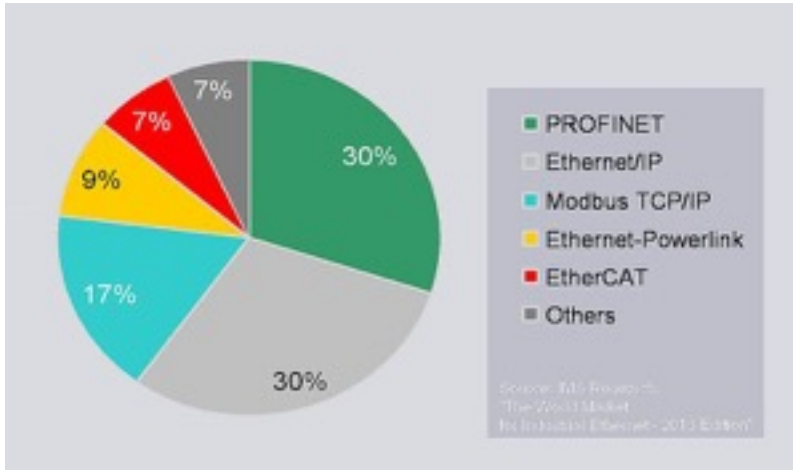
What we will talk about ?

- Modbus
- Mystical S7
- Authentication and protection
- Profinet



Industrial protocols

- CIP
- BACnet
- CC-Link
- Ethernet/IP
- Modbus
- Profinet
- S3 / S5 / S7
- DNP3



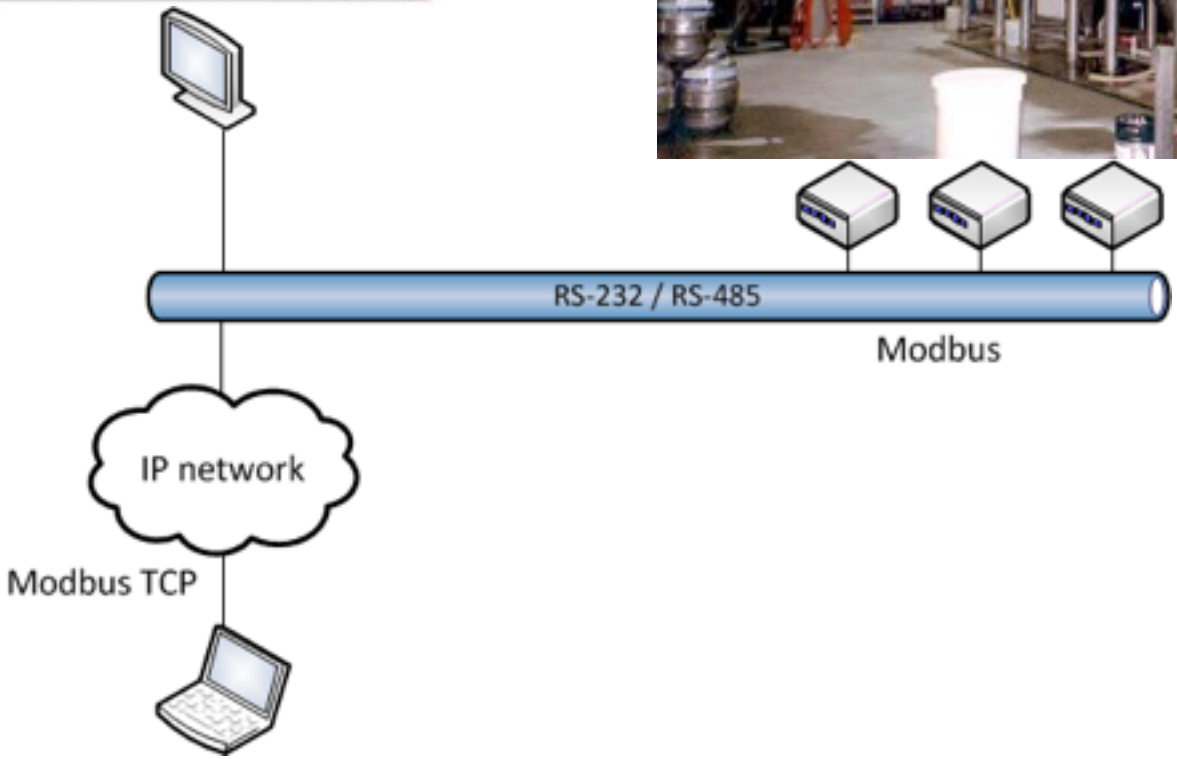
Old Modbus

- Published by Modicon (now Schneider Electric) in 1979.

- Widely used for connecting industrial electronic devices
  - Schneider Electric
  - Advanced Micro Controls
  - ABB
  - Emerson
  - Chinese NONAME
  - and all other vendors

Modbus in XX

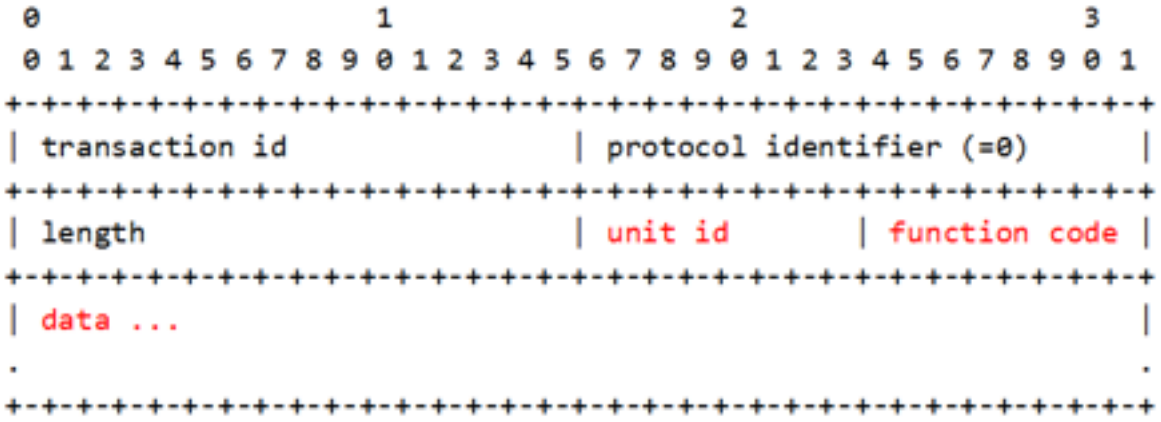
Modbus in XXI



Modbus TCP

Standard port – 502/tcp

Modbus Request packet:



- No authentication
- No encryption
- No security

Modbus Functions

- Data access
  - Read/Write Coils and Registers
  - Read/Write File Records
- Diagnostics
  - Device Identification
  - …
- + User Defined Functions

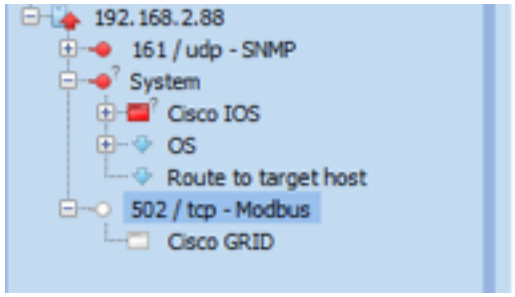
Modbus Device Identification

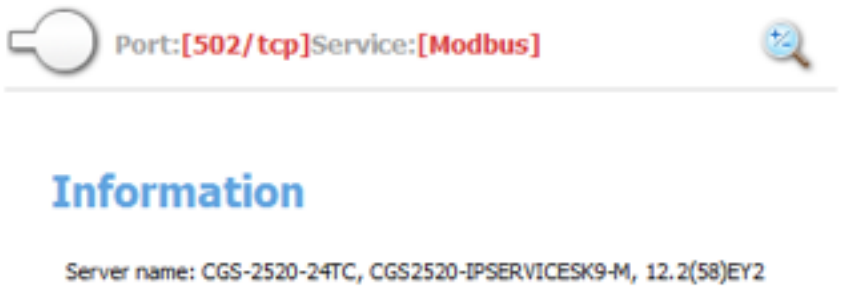
Standard Function (opcode 0x2B, subcode 0x0E)

- **VendorName**
- **ProductCode**
- **MajorMinorRevision**
- VendorUrl
- ProductName
- ModelName
- UserApplicationName

Modbus Device Identification

192.168.2.6	Modicon M340: BMX NOE 0110 v2.0
192.168.2.10	Modicon M340: BMX NOE 0110 v4.50
192.168.2.81	Modicon M340: BMX NOE 0110 v4.50
192.168.2.11	Modicon M340: BMX NOE 0110 v5.50
192.168.2.83	Modicon M340: BMX NOE 0110 v5.50
192.168.2.2	Modicon M340: BMX P34 2020 v2.0
192.168.2.7	Modicon M340: BMX P34 2020 v2.0
192.168.2.12	Modicon M340: BMX P34 2020 v2.0
192.168.2.62	Modicon M340: BMX P34 2020 v2.0
192.168.2.72	Modicon M340: BMX P34 2020 v2.0
192.168.2.4	Modicon M340: BMX P34 2020 v2.2
192.168.2.76	Modicon M340: BMX P34 2020 v2.2
192.168.2.65	Modicon Premium: TSX ETY 4103 v4.3
192.168.2.86	Modicon Premium: TSX ETY 4103 v4.3
192.168.2.85	Modicon Premium: TSX ETY 4103 v4.4



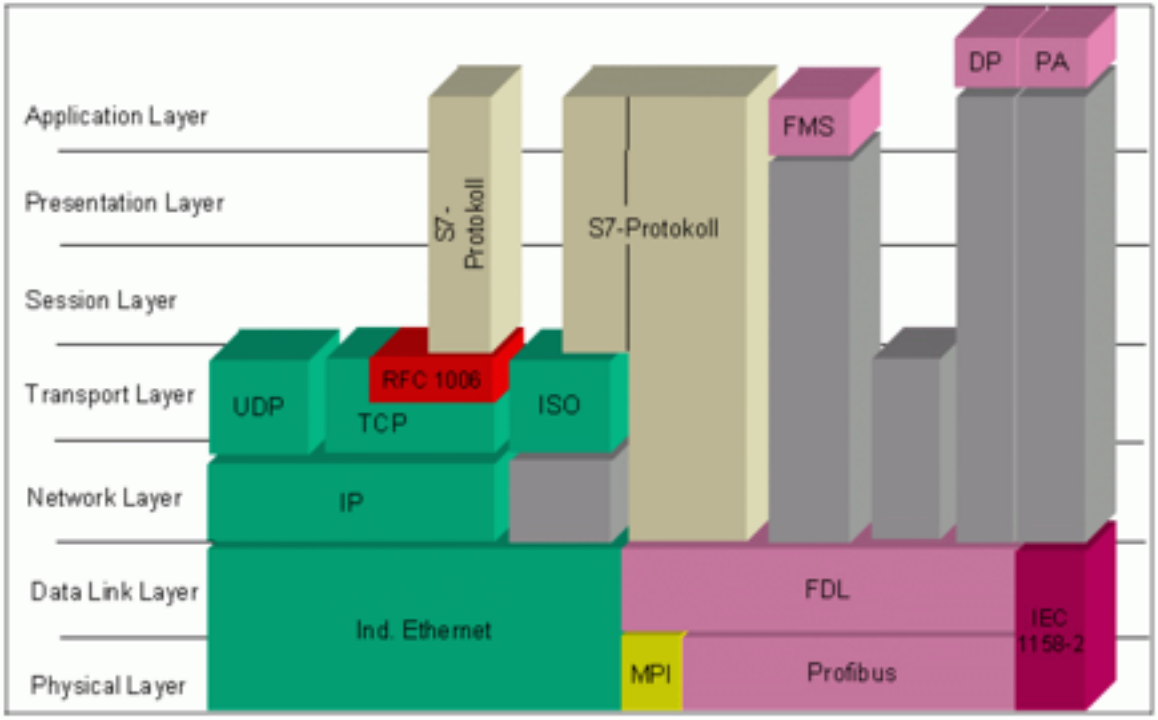


Modbus Tools

- Emulators:
  - http://www.modbustools.com/download.asp
- Device Discovery:
  - https://code.google.com/p/plcscan/
  - https://code.google.com/p/modscan/
- …
- Wireshark
- python

Modbus Demo

Mystic S7



Standard port – 102/tcp

In Siemens docs - iso-on-tcp, rfc 1006

S7 materials

- Exploiting Siemens Simatic S7 PLCs (by Dillon Beresford)

  http://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_Slides.pdf

- Wireshark dissector

  http://sourceforge.net/projects/s7commwireshark/

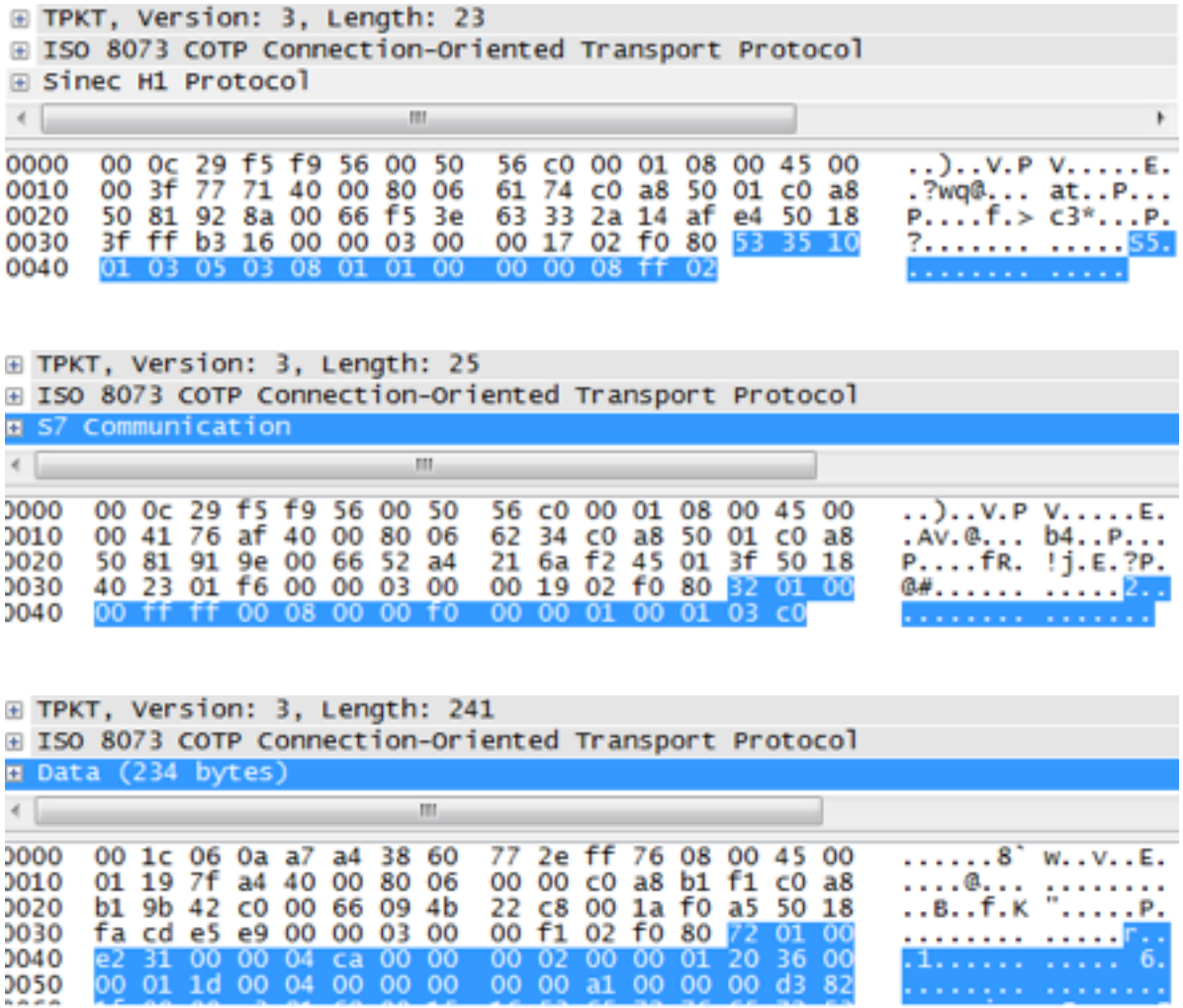
- Libnodave – free communication library

  http://sourceforge.net/projects/libnodave/

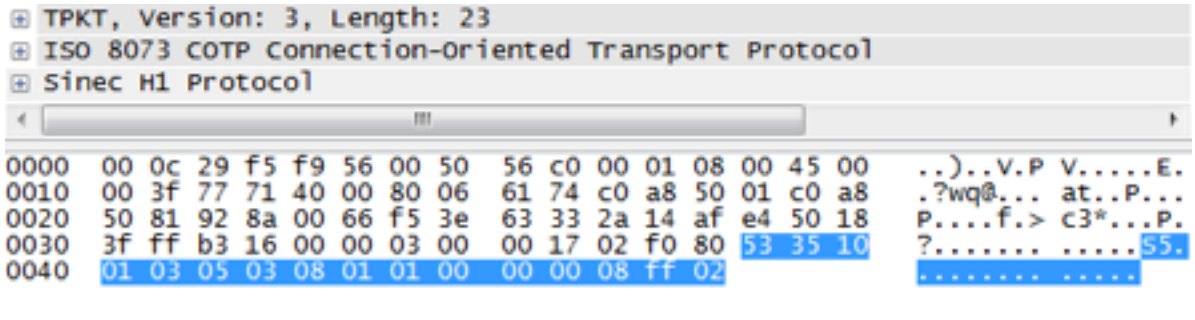
ISO-on-TCP (RFC 1006)

- Transport layer only
- Require source and destination TSAP (Transport Service Access Point) for connection
- TSAP (2 bytes)
  - Connection type (PG – 0x01, OP – 0x02)
  - Rack/Slot Id

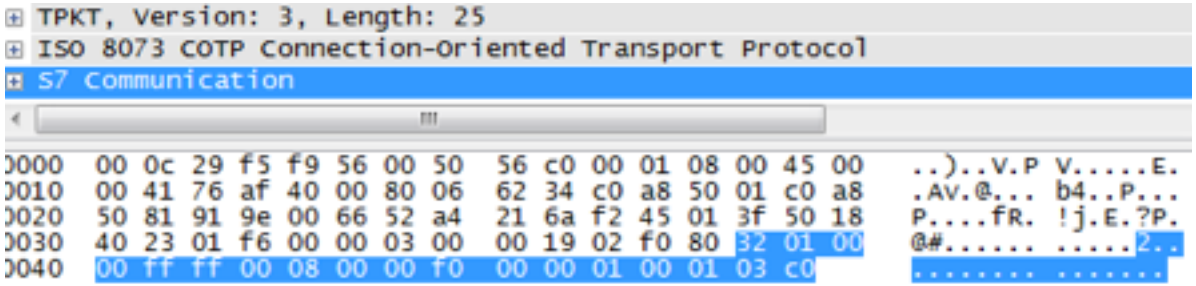
What is under ISO-on-TCP?



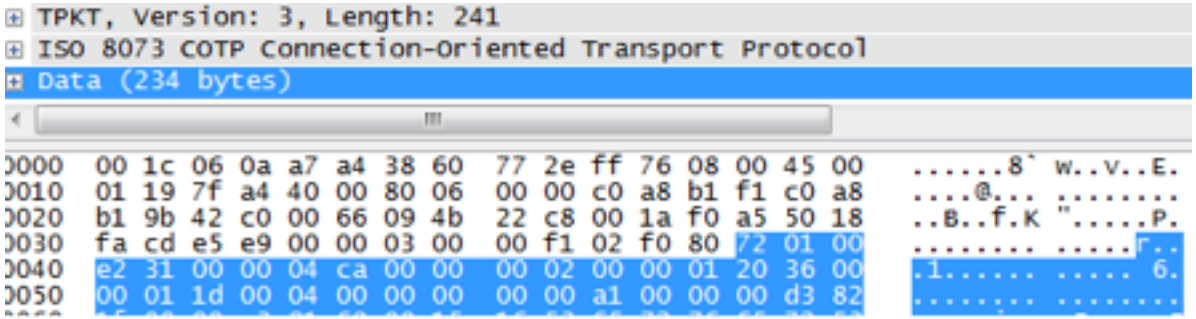
What is under ISO-on-TCP?



S5 Communication
aka
FETCH / WRITE
aka
Sinec H1



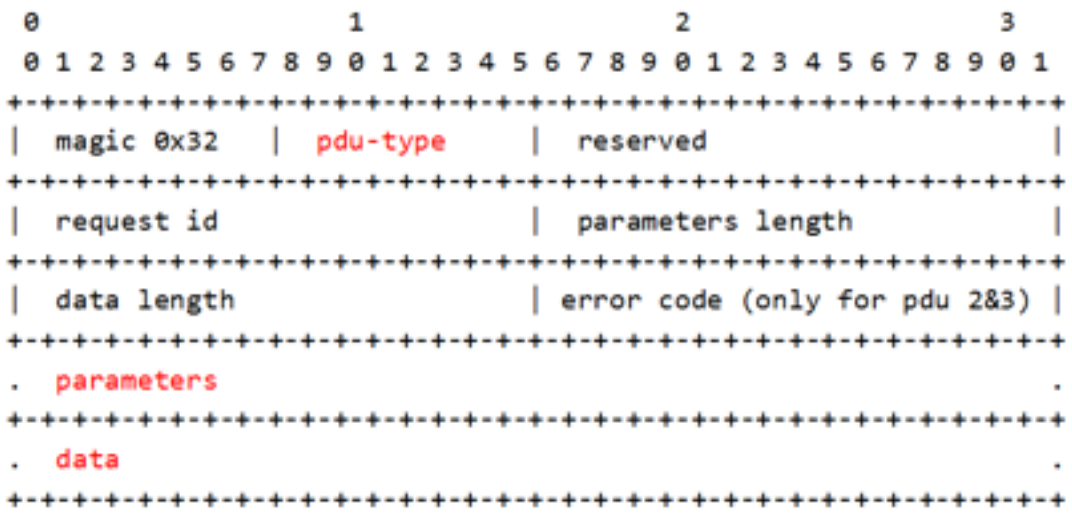
S7 Communication



Another
S7 Communication

S7 communication

S7 packet:



PDU-types:

- 0x01 – Request
- 0x02 – Acknowledgement
- 0x03 – Response
- 0x07 – User Data

What we can do

- Read / Write data
- Start / Stop CPU
- Upload / Download Blocks

- List blocks
- Get blocks info
- Read SZL (System Status List)
    - Module Identification
    - Component Identification
    - LED’s status

Device Identification

- PLC scan (https://code.google.com/p/plcscan/)

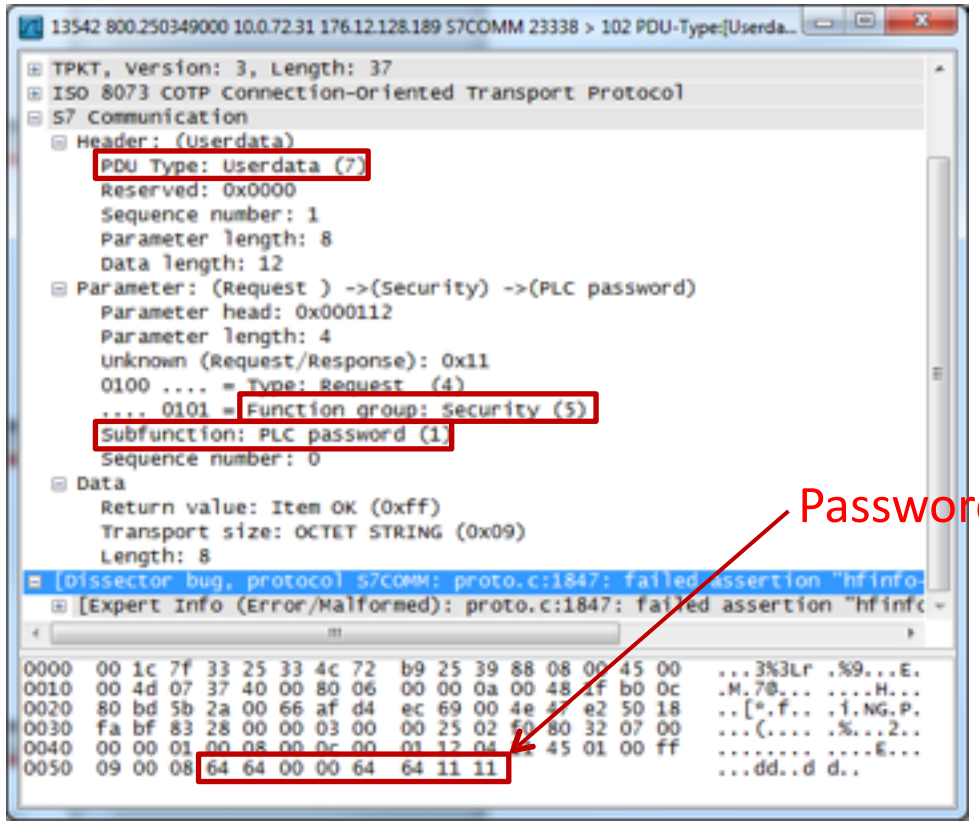
- For s7-300:

```
Module :                6ES7 151-8AB01-0AB0 v.2
Basic Hardware :        6ES7 151-8AB01-0AB0 v.2
Basic Firmware :        v.3.2.6
PLC Name :              SIMATIC 300(Bla_bla_name)
Module Name :           IM151-8 PN/DP CPU
Plant ID :
Copyright :             Original Siemens Equipment
Module Serial number : S C-BOUV49xxxxx1
Module type name :      IM151-8 PN/DP CPU
Memory card Serial number : MMC 6CAxxxx0
Module OEM ID :
Module Location :
```

- For s7-1200:

```
Module :                6ES7 212-1BD30-0XB0 v.2
Basic Hardware :        6ES7 212-1BD30-0XB0 v.2
Basic Firmware :        6ES7 212-1BD30-0XB0 v.2.2.0
```

S7-300 password protection



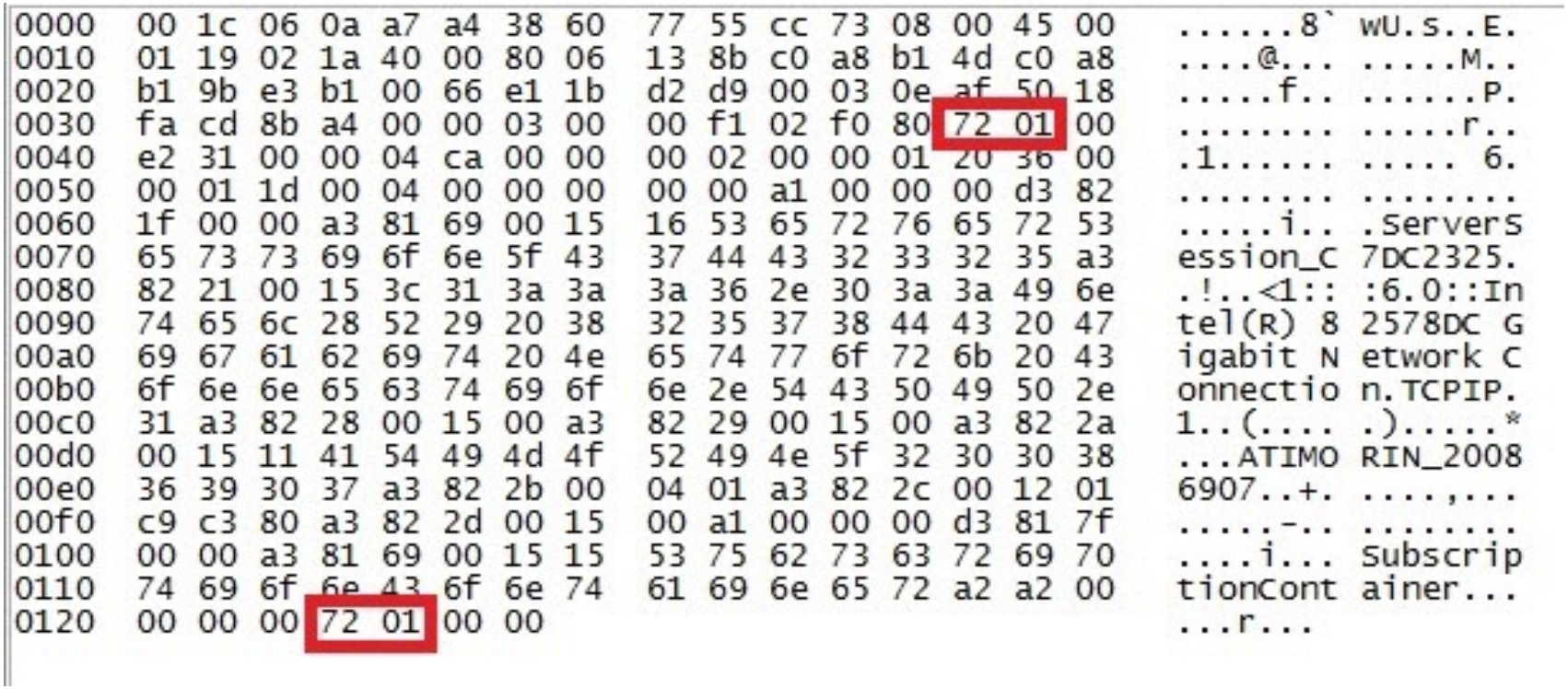
«Encryption»:

```
s[0] = pwd[0] ^ 0x55
s[1] = pwd[1] ^ 0x55
for (int i=2; i<8; i++)
    s[i] = pwd[i] ^ s[i-2] ^ 0x55
```

S7comm on S7-1200

	S7-300	S7-1200
Read/Write Vars	+	+
Device Identification	+	+/-
Start/Stop CPU	+	-
Upload/Download Blocks	+	-
Blocks Info	+	-
LED's status	+	-

«Another S7 communication»



```
0000  00 1c 06 0a a7 a4 38 60  77 55 cc 73 08 00 45 00   ......8` wU.s..E.
0010  01 19 02 1a 40 00 80 06  13 8b c0 a8 b1 4d c0 a8   ....@... .....M..
0020  b1 9b e3 b1 00 66 e1 1b  d2 d9 00 03 0e af 50 18   .....f.. ......P.
0030  fa cd 8b a4 00 00 03 00  00 f1 02 f0 80 72 01 00   ........ .....r..
0040  e2 31 00 00 04 ca 00 00  00 02 00 00 01 20 36 00   .1...... ..... 6.
0050  00 01 1d 00 04 00 00 00  00 00 a1 00 00 00 d3 82   ........ ........
0060  1f 00 00 a3 81 69 00 15  16 53 65 72 76 65 72 53   .....i.. .ServerS
0070  65 73 73 69 6f 6e 5f 43  37 44 43 32 33 32 35 a3   ession_C 7DC2325.
0080  82 21 00 15 3c 31 3a 3a  3a 36 2e 30 3a 3a 49 6e   .!..<1:: :6.0::In
0090  74 65 6c 28 52 29 20 38  32 35 37 38 44 43 20 47   tel(R) 8 2578DC G
00a0  69 67 61 62 69 74 20 4e  65 74 77 6f 72 6b 20 43   igabit N etwork C
00b0  6f 6e 6e 65 63 74 69 6f  6e 2e 54 43 50 49 50 2e   onnectio n.TCPIP.
00c0  31 a3 82 28 00 15 00 a3  82 29 00 15 00 a3 82 2a   1..(.... .).....*
00d0  00 15 11 41 54 49 4d 4f  52 49 4e 5f 32 30 30 38   ...ATIMO RIN_2008
00e0  36 39 30 37 a3 82 2b 00  04 01 a3 82 2c 00 12 01   6907..+. ....,...
00f0  c9 c3 80 a3 82 2d 00 15  00 a1 00 00 00 d3 81 7f   .....-.. ........
0100  00 00 a3 81 69 00 15 15  53 75 62 73 63 72 69 70   ....i... Subscrip
0110  74 69 6f 6e 43 6f 6e 74  61 69 6e 65 72 a2 a2 00   tionCont ainer...
0120  00 00 00 72 01 00 00                              ...r...
```

Simple S7 packet (connection establishment)

72 01 – S7 data delimiter

TIA Portal read/write protection

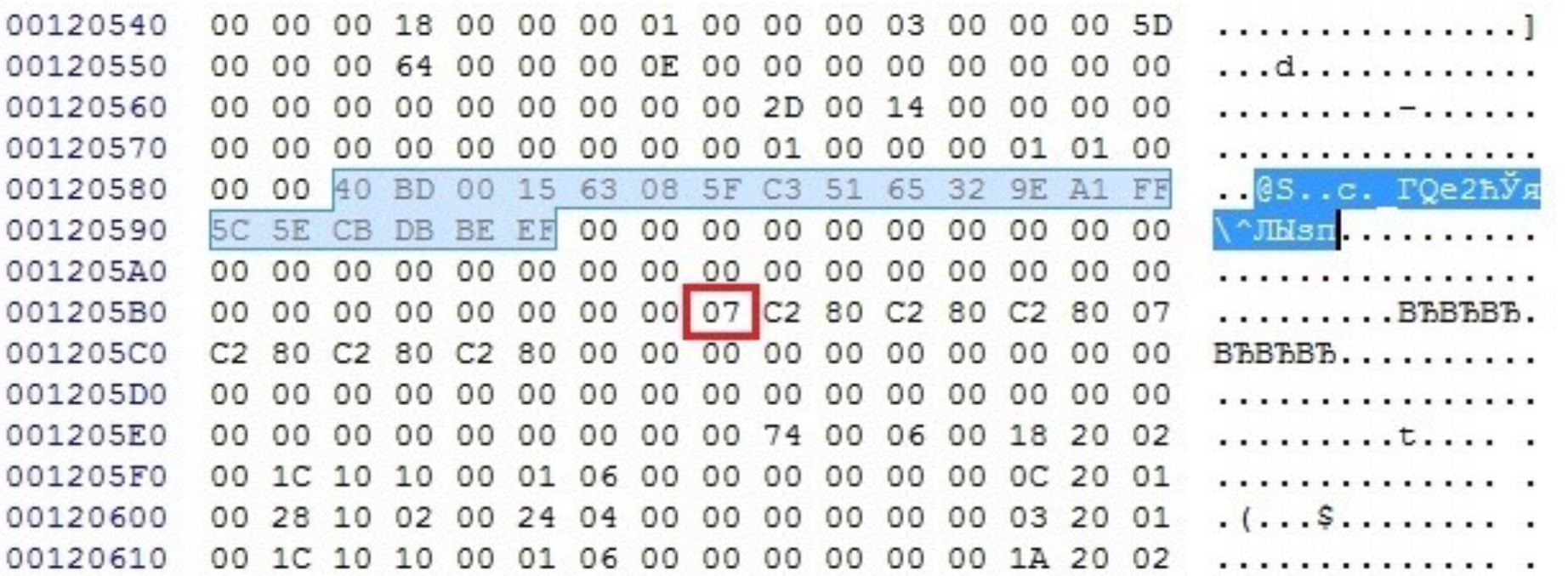
PLC read/write password protection for main operations:

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

TIA Portal PEData.plf passwords history

Simple SHA-1 passwords:

456e6372797074656450617373776f72[a-f0-9]{240,360}000101000000[a-f0-9]{40}



redbox value: password_length * 2 + 1

S7 password hashes extractor

source: http://code.google.com/p/scada-tools/source/browse/s7_password_hashes_extractor.py

```
root@pc:/home/johndoe/siemens/phdays2013/PEData# python s7_password_hashes_extractor.py
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 history list:
        hash 1: 40bd001563085fc35165329ea1ff5c5ecbdbbeef
        hash 2: ef56ad1362587b5302461ca1c03df022d61b0a1e
        hash 3: c74bd9bc4ef69048126c62055741985a16aa7a83
        hash 4: 3e9ba8eb61b1f8b0335a2cdfac6fc2f0fc5a825c        (current)
root@pc:/home/johndoe/siemens/phdays2013/PEData#
```

extracting all password sha1 hashes from TIA Portal project file and simple bruteforce.

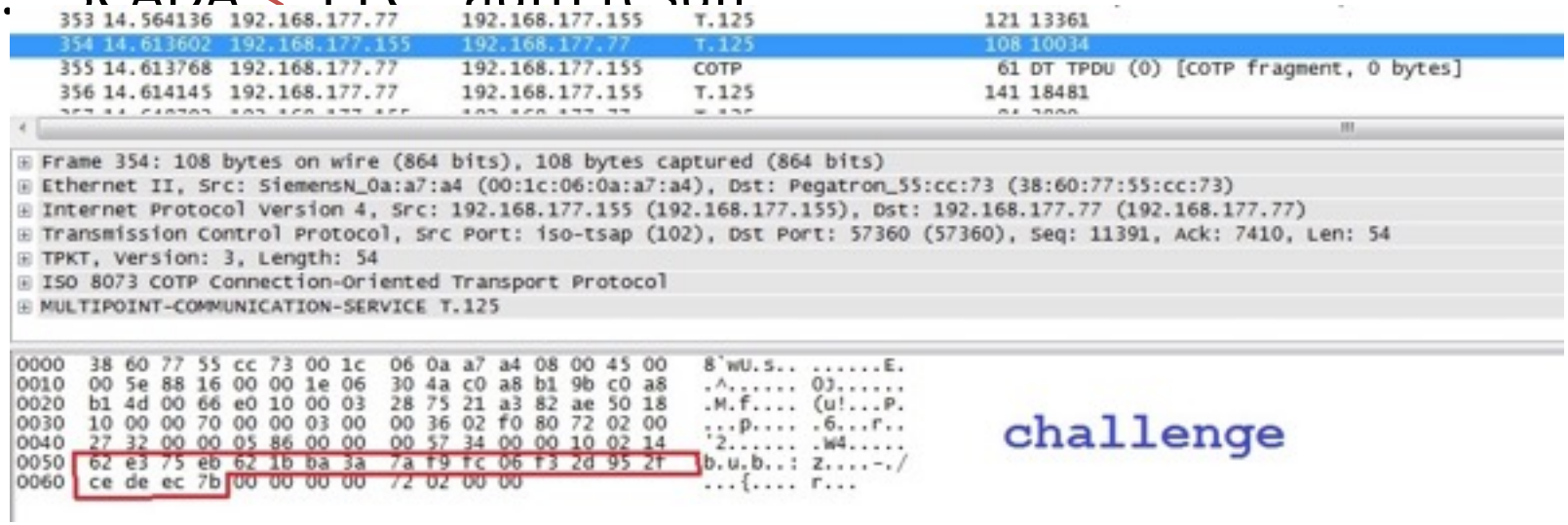
Also possible to intercept password hash when uploading new project to PLC. It's easy.

Know-how protection:

- prevent code blocks (OB, FB, FC, DB) from unauthorized access
- base64(sha1(password-in-unicode))

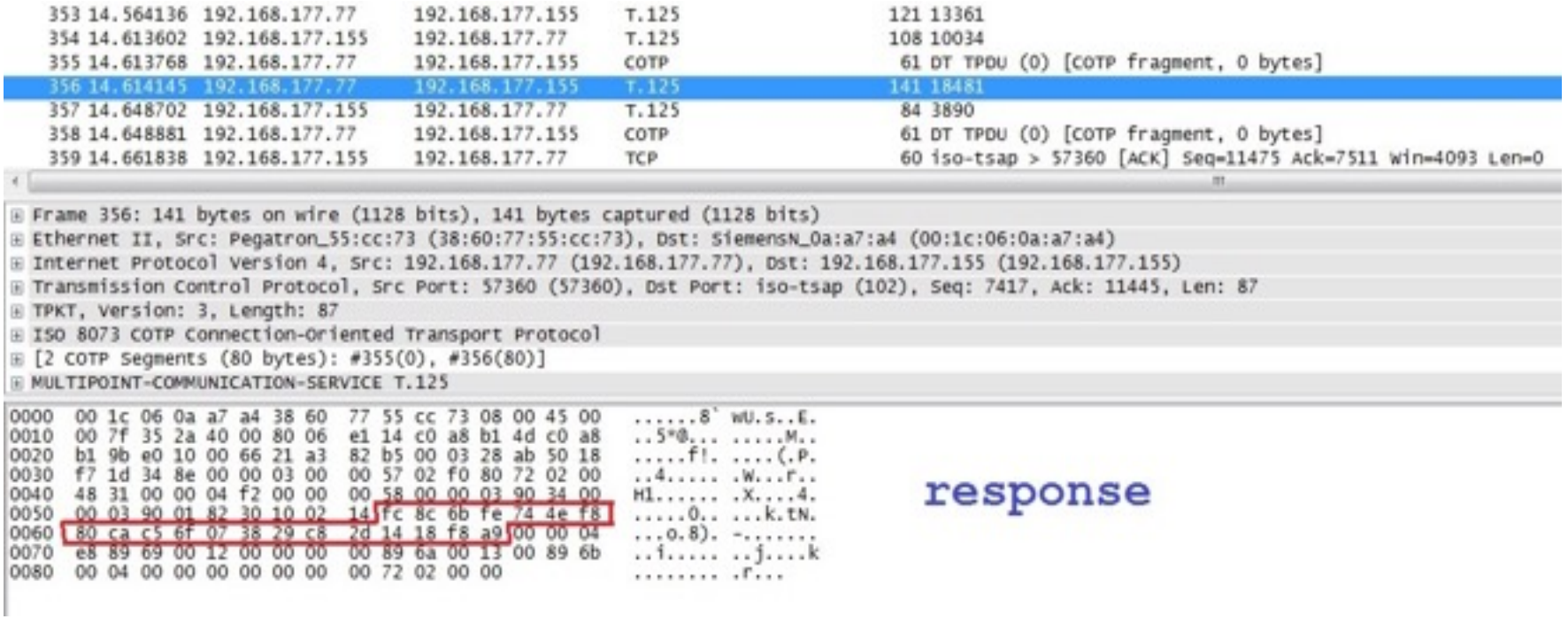
SCADA <-> PLC S7 authentication

1. SCADA -> PLC : auth request
2. SCADA <- PLC : challenge
3. SCADA -> PLC : response = HMAC(SHA1(password), challenge)
4. SCADA <- PLC : auth result



sending authentication challenge from PLC to SCADA workstation

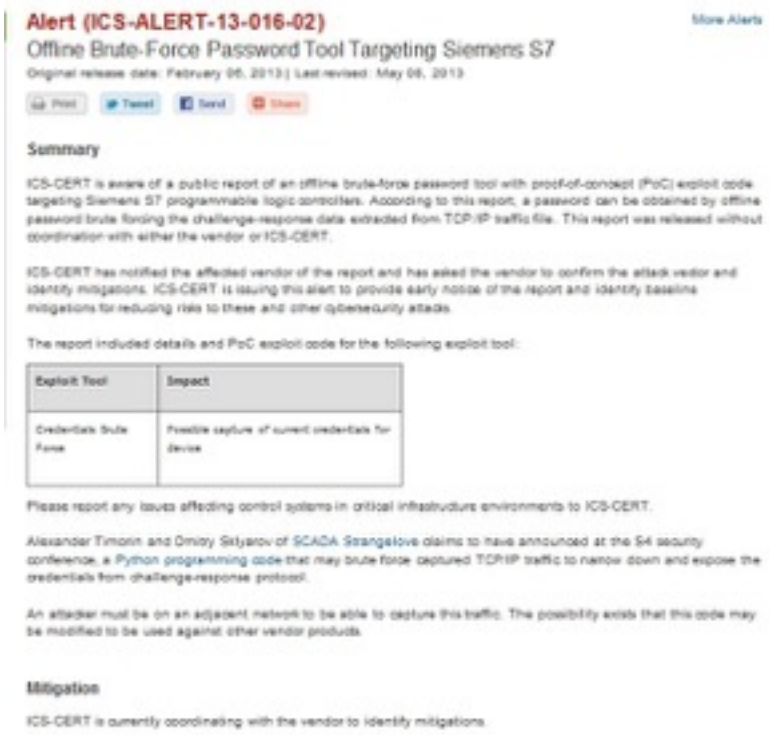
SCADA <-> PLC S7 authentication



sending authentication response from SCADA workstation to PLC

SCADA <-> PLC S7 authentication

- ICS-CERT alert: https://ics-cert.us-cert.gov/alerts/ICS-ALERT-13-016-02
- John the Ripper Jumbo patch: https://github.com/magnumripper/JohnTheRipper/pull/193
- http://www.digitalbond.com/blog/2013/05/10/john-the-ripper-s7-password-cracking/



Alert (ICS-ALERT-13-016-02) More Alerts

Offline Brute-Force Password Tool Targeting Siemens S7

Original release date: February 06, 2013 | Last revised: May 06, 2013

Print Tweet Send Share

Summary

ICS-CERT is aware of a public report of an offline brute-force password tool with proof-of-concept (PoC) exploit code targeting Siemens S7 programmable logic controllers. According to this report, a password can be obtained by offline password brute forcing the challenge-response data extracted from TCP/IP traffic file. This report was released without coordination with either the vendor or ICS-CERT.

ICS-CERT has notified the affected vendor of the report and has asked the vendor to confirm the attack vector and identify mitigations. ICS-CERT is issuing this alert to provide early notice of the report and identify baseline mitigations for reducing risks to these and other cybersecurity attacks.

The report included details and PoC exploit code for the following exploit tool:

Exploit Tool	Impact
Credentials Brute Force	Possible capture of current credentials for device

Please report any issues affecting control systems in critical infrastructure environments to ICS-CERT.

Alexander Timorin and Dmitry Sklyarov of SCADA Strangelove claims to have announced at the S4 security conference, a Python programming code that may brute force captured TCP/IP traffic to narrow down and expose the credentials from challenge-response protocol.

An attacker must be on an adjacent network to be able to capture this traffic. The possibility exists that this code may be modified to be used against other vendor products.

Mitigation

ICS-CERT is currently coordinating with the vendor to identify mitigations.

digital bond

Join

What's Hot: Project Basecamp S4 2013 Video Bandolier Training: Cyber Security For Power Generation

John The Ripper – S7 Password Cracking

Stephen Hill

Tweet 27 Share 7

At S4x13, Scadastrangelove (@scadasl) released a offline brute force password cracking script (http://pastebin.com/0G9Q2k6y). Shortly after the script was released the functionality from that script was added into John The Ripper. Documented in The Rack is how John The Ripper is capable of cracking S7 password hashes using the Scadastrangelove technique of offline password cracking from a packet capture.

John The Ripper has been around for many years, and is one of the most common password cracking utilities out there. With an add-on plugin and a script that is easy to run, the password hashes are extracted out of packet captures, and cracked using John The Ripper.

The use of John The Ripper outside of the normal workstations and servers inside of ICS environments is very limited, as most devices you can't get the information required to run the software against the password hashes.

With the rise of password complexity requirements inside of ICS environments, auditing the password complexity of PLC and like devices can be difficult and rely a lot of how much you trust the engineer. As an example there is nothing to say that the PLC configuration that you are looking at on the engineer workstation is the one that is truly pushed out to the PLC. With the ability to gather information from a packet capture and then verify the password complexity adds that much assurance to an assessment.

S7 challenge-response extractor

source: http://code.google.com/p/scada-tools/source/browse/s7_brute_offline.py

```
root@pc:/home/johndoe/siemens/phdays2013/s7-brute-offline# python s7-brute-offline.py
WARNING: No route found for IPv6 destination :: (no default route?)
using pcap file: stop_cpu_cmd_right_pass_123.pcap
found packets indeces: pckt_108=353, pckt_141=355, pckt_84=356, pckt_92=0
auth ok
found challenge: 62e375eb621bba3a7af9fc06f32d952fcedeec7b
found  response: fc8c6bfe744ef880cac56f073829c82d1418f8a9
start password bruteforsing  ...
found password: 123
root@pc:/home/johndoe/siemens/phdays2013/s7-brute-offline#
```

extracting challenge-response values from pcap file and simple bruteforce.

pckt_len+14 == 84 and hexlify(r[pckt_indx].load)[14:24] == '7202000f32‘ -> auth ok

pckt_len+14 == 92 and hexlify(r[pckt_indx].load)[14:24] == '7202001732‘ -> auth bad

Other researches/materials:

- Dillon Beresford: http://scadahacker.com/exploits/exploits-dillonbh2011.html

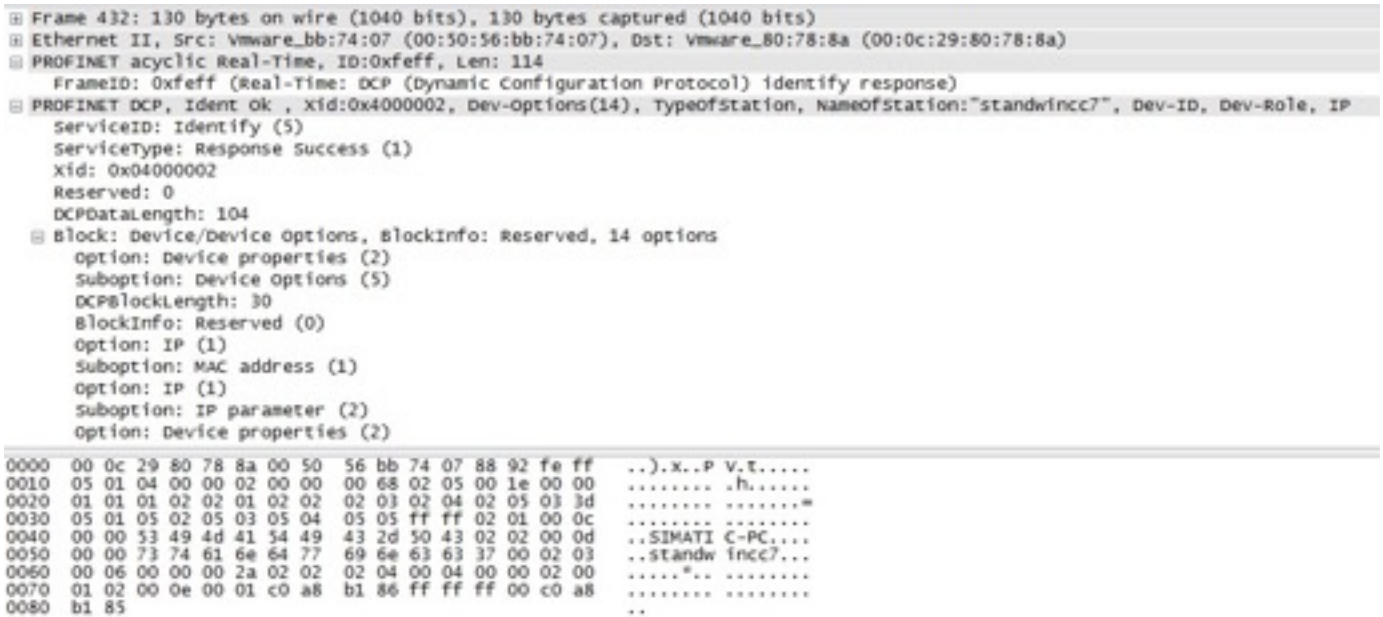
PROFINET family

2003: IEC 61158, IEC 61784

- PROFINET CBA (Component Based Automation)
- PROFINET IO

PROFINET IO

- master – slave communications
- RT (~ 10 ms), IRT (~ 1 ms)
- PROFINET PTCP (Precision Time Control Protocol)
- PROFINET DCP (Discovery and Basic Configuration Protocol)



PROFINET DCP scanner

source: http://code.google.com/p/scada-tools/source/browse/profinet_scanner.py

```
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      : winccv7sp3upd4  : 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
```

discovering all SCADA devices (PC, HMI, PLC) in subnet

PROFINET DCP scanner

payload = 'fefe05000401000200800004ffff0000'

pp = Ether(type=0x8892, src=src_mac, dst=01:0e:cf:00:00:00)/payload.decode('hex')

fefe	2b: DCP multicast header
05	1b: Identify service
00	1b: Request type
04010002	4b: Xid (request identificator)
0080	2b: Response delay
0004	2b: DCP data length
ffff0000	4b: dcp dataOption(All), Suboption(All)

Also we can:

- change name of station
- change ip, gateway
- request network info
- LED flashing: PLC, HMI (something wrong with PLC or devices ??)
- and much more …

profinet video demo

How to analyze protocols ?

- search-analyze-search-analyze-search …
- Rob Savoye: “Believe it or not, if you stare at the hex dumps long enough, you start to see the patterns”
- Rob Savoye: FOSDEM 2009 Reverse Engineering of Proprietary Protocols, Tools and Techniques : http://youtu.be/t3s-mG5yUjY
- Netzob: http://www.netzob.org
- Fuzzing
- wireshark

  tcpdump

  python

  scapy

  hex viewer

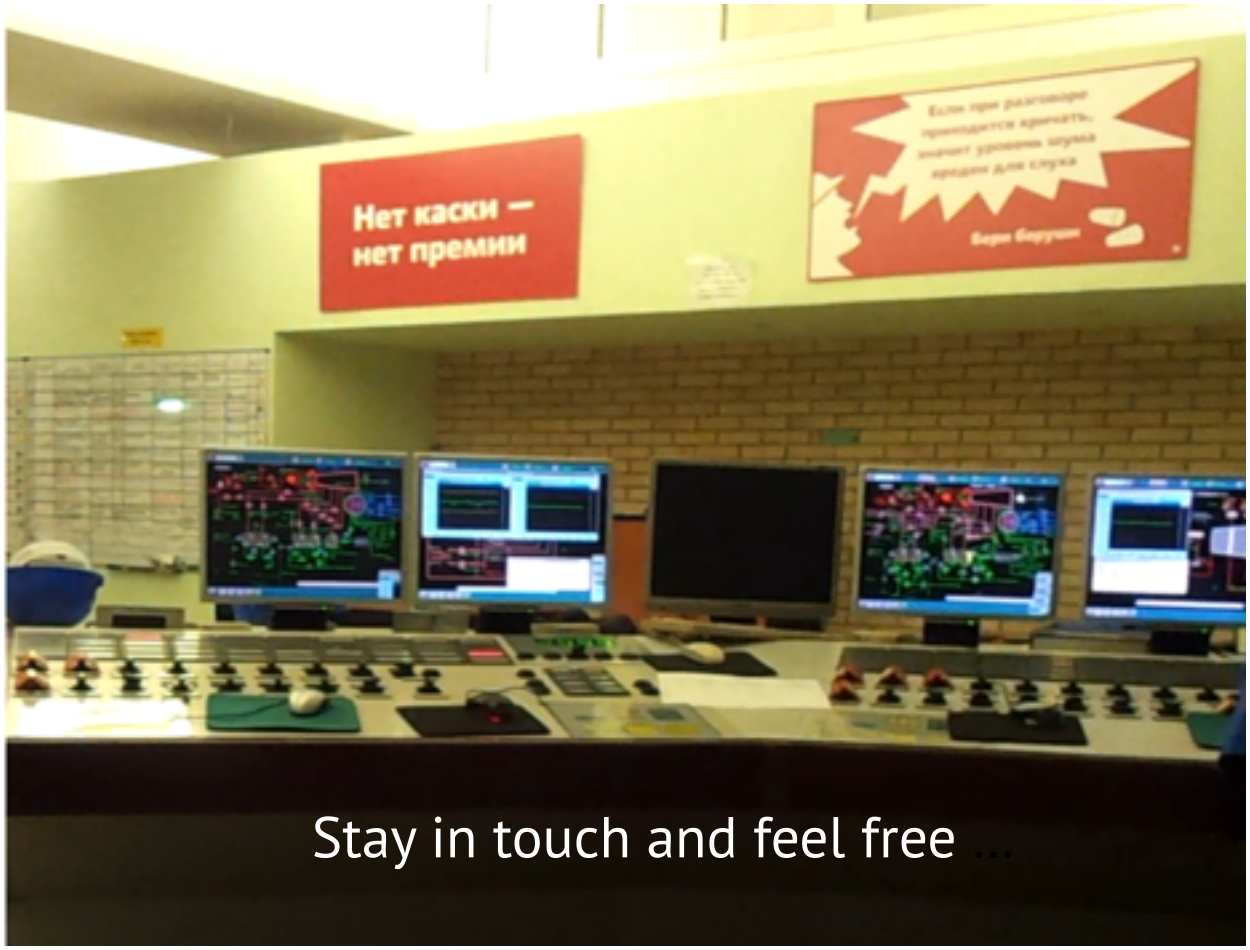
Outro

- Positive Technologies SCADA analytics:
  http://www.ptsecurity.com/download/SCADA_analytics_english.pdf
- Findings
- Recommendations:
  - http://scadastrangelove.org
  - http://www.scadahacker.com
  - http://www.digitalbond.com
  - http://ics-cert.us-cert.gov
- Releases:
  https://code.google.com/p/scada-tools/
  https://code.google.com/p/plcscan/
- Greetz to: SCADASTRANGELOVE TEAM
- QA
- And now …

S7-300. Live Demo

Thanks to all … to be continued

Timorin Alexander atimorin@ptsecurity.ru
Efanov Dmitry defanov@ptsecurity.ru



Stay in touch and feel free …

POSITIVE TECHNOLOGIES