Siemens Simatic S7 PLC Exploitation

S7-Fu (功夫) with Rapid7 Metasploit

Black Hat USA+2011

Dillon Beresford

d1n@nsslabs.com

Dillon Beresford

狄龙

Security Researcher

at

With assistance from…

NSS Labs, Project Funding
Brian Meixell, Engineering Support.
Ian Parker, Lab Setup and QA.
Tim Otto, Hardware Procurement.
Dale Peterson (DigitalBond) Blog
Bob Radvanovsky (SCADASEC) List
INL/ICS-CERT, Disclosure Process
PLCTrainer.NET (PLC Trainers)

Introduction

- PLCs are computers used to automate mechanical device processes.
- PLCs are used in the nuclear, oil and gas refineries, coal, water and waste treatment, transportation, aerospace, defense and commercial factories, among many other things…
- The S7-300, and S7-400 are currently the most common PLCs in use, however the S7-1200 is gaining more traction .
- Simatic S7 PLCs offer state of the art PROFIBUS and PROFINET communication protocols.
- From an attacker perspective, each of the S7 PLCs have one thing in common. They communicate over ISO-TSAP (RFC-1006) on TCP port 102.
- When TSAP was layered on Top of TCP, security wasn't factored in.
- The Simatic S7 PLCs run on a 32-bit Linux operating system.
- S7 PLC Firmware images are encrypted and hex encoded, some are using simple rotating shift sequences to obfuscate the strings in the firmware.
- The S7-300 we are going to exploit has a TELNET daemon and HTTP server running as background process which are used by the Siemens developers for debugging.
- The S7-1200 also has a web server included in it for diagnostics and HMI.

Testing

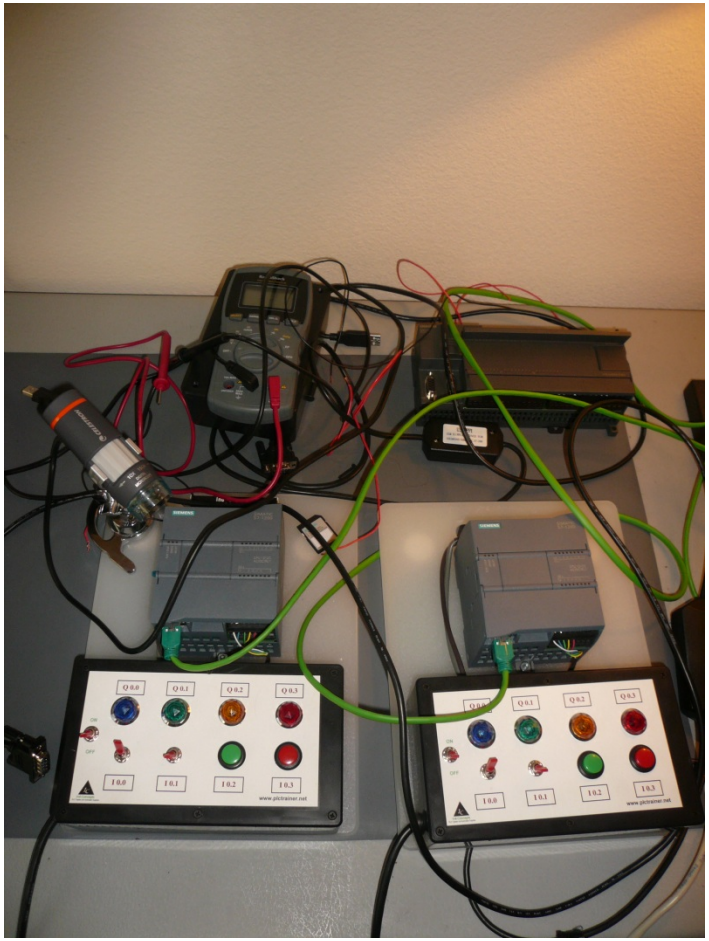
Devices Under Test:

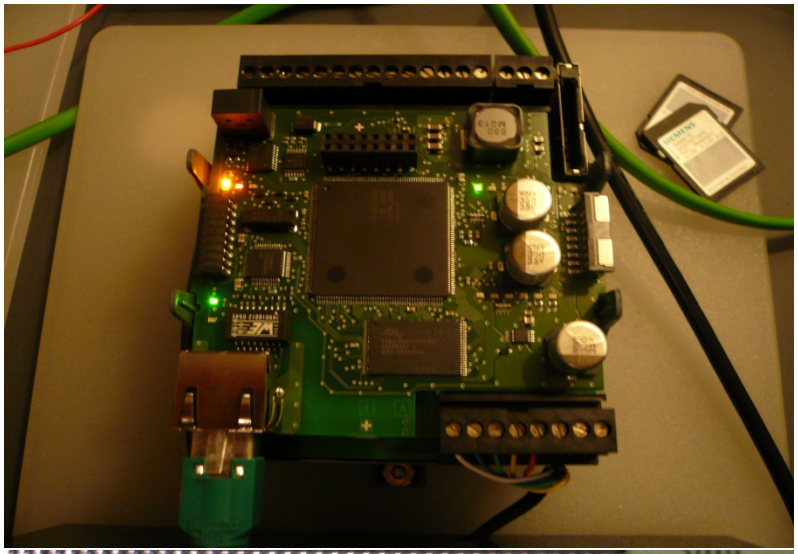
- PLC1 6ES7 212-1BD30-0XB0 AC/DC => Siemens Simatic S7-1200
- PLC2 6ES7 212-1BD30-0XB0 AC/DC => Siemens Simatic S7-1200
- PLC3 317-2EJ10-0AB0 => Siemens Simatic S7-300
- PLC4 317-2EJ10-0AB0 => Siemens Simatic S7-300

PLC Firmware Versions:

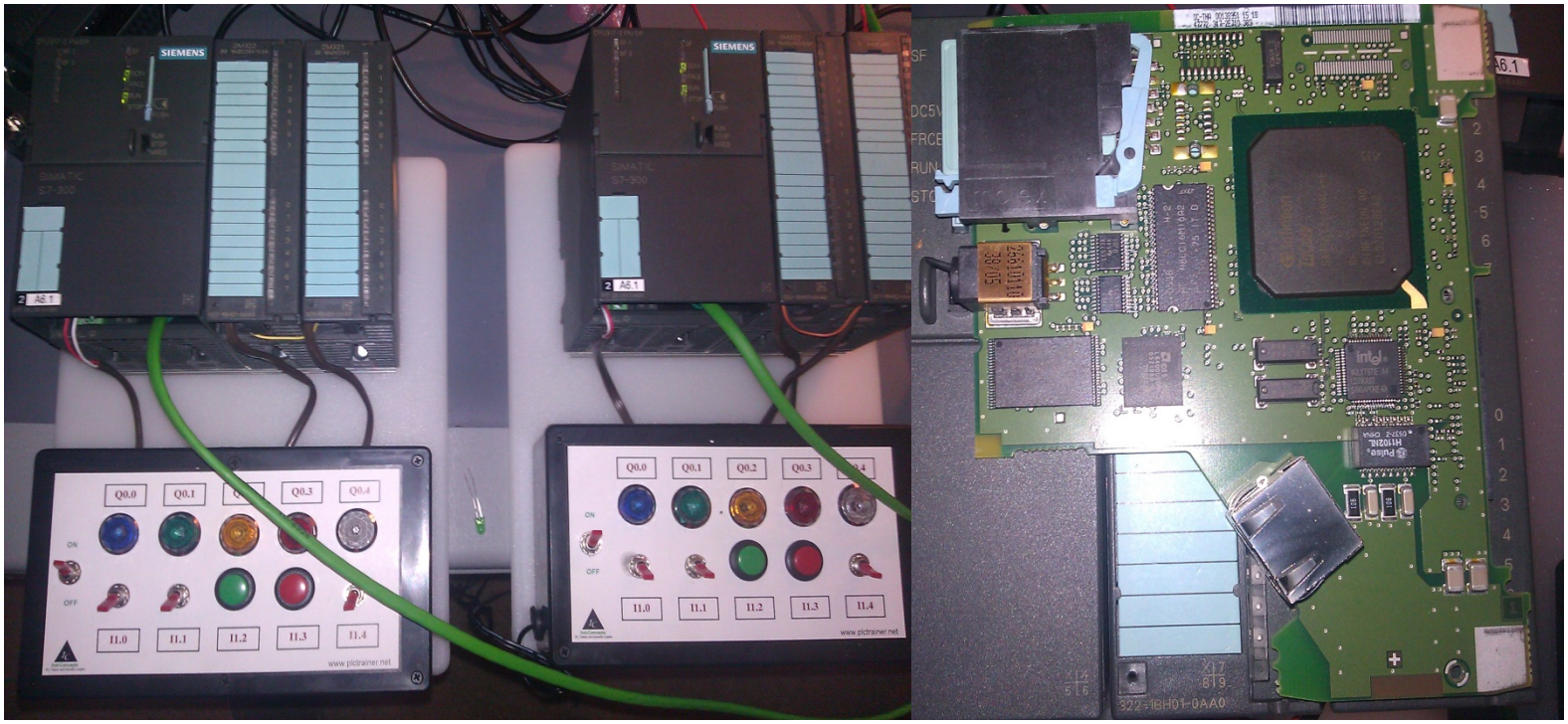
- Version 2.2 => Siemens Simatic S7-1200
- Version 2.3.4 => Siemens Simatic S7-300

Simatic S7-1200



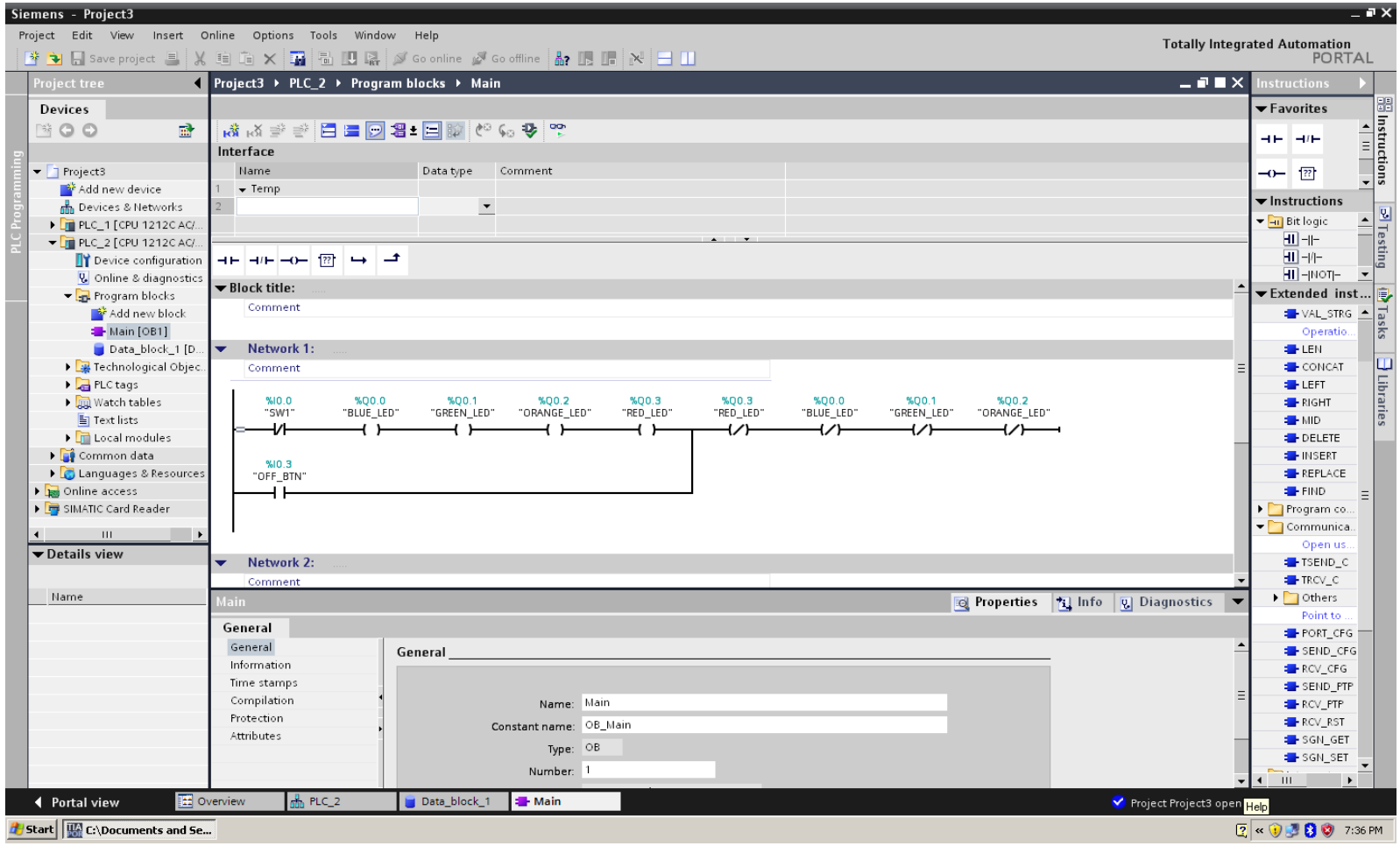


Simatic S7-300



"The Big Boys"

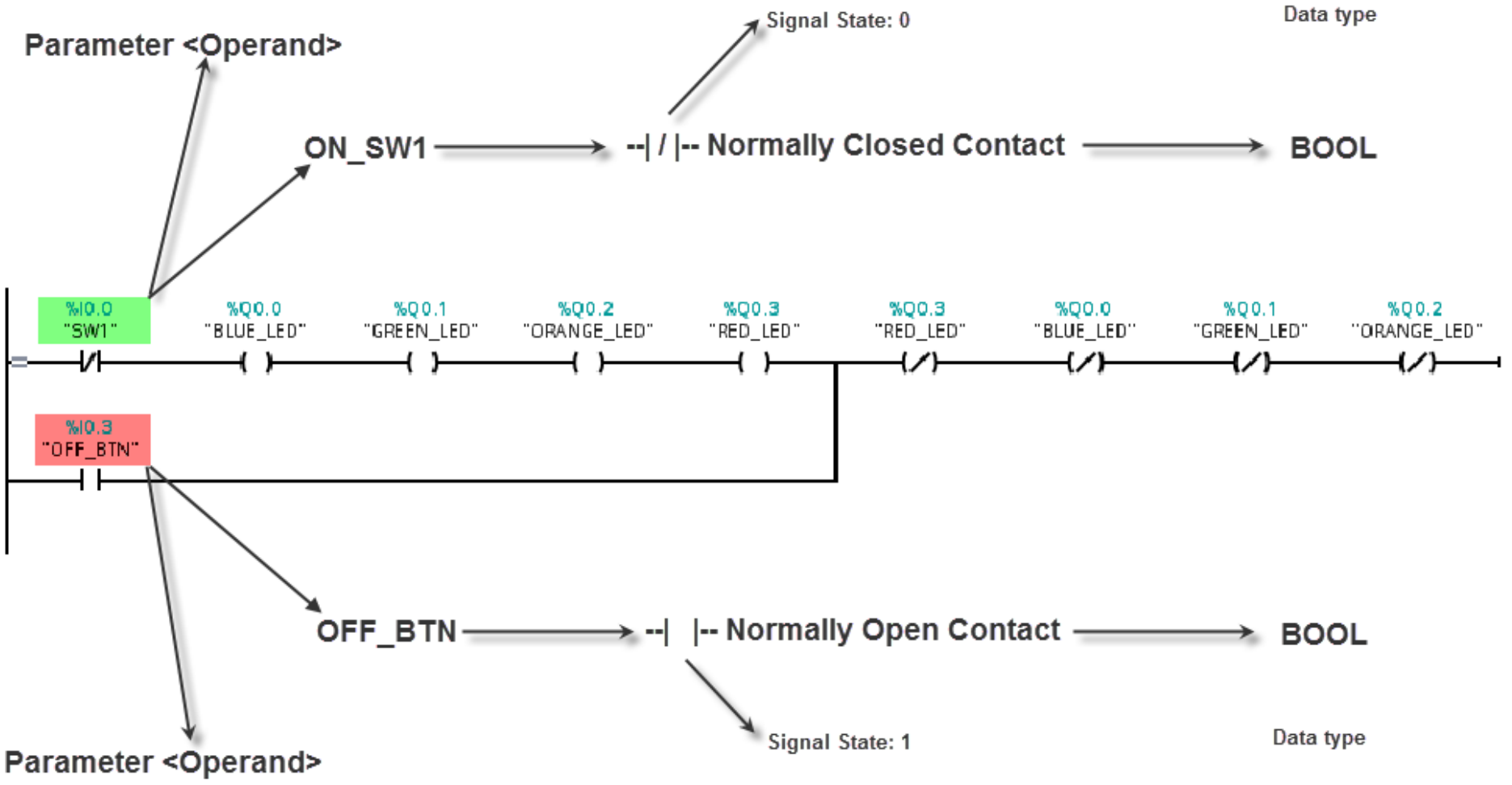
Step 7 Basic


Siemens - Project3
Project Edit View Insert Online Options Tools Window Help
Totally Integrated Automation PORTAL
Project tree
Devices
Project3
Add new device
Devices & Networks
PLC_1 [CPU 1212C AC/...
PLC_2 [CPU 1212C AC/...
Device configuration
Online & diagnostics
Program blocks
Add new block
Main [OB1]
Data_block_1 [D...
Technological Objec...
PLC tags
Watch tables
Text lists
Local modules
Common data
Languages & Resources
Online access
SIMATIC Card Reader
Details view
Name
Project3 ▸ PLC_2 ▸ Program blocks ▸ Main
Interface
Name
Data type
Comment
1 Temp
2
Block title:
Comment
Network 1:
Comment
%I0.0 "SW1"
%Q0.0 "BLUE_LED"
%Q0.1 "GREEN_LED"
%Q0.2 "ORANGE_LED"
%Q0.3 "RED_LED"
%Q0.3 "RED_LED"
%Q0.0 "BLUE_LED"
%Q0.1 "GREEN_LED"
%Q0.2 "ORANGE_LED"
%I0.3 "OFF_BTN"
Network 2:
Comment
Main
Properties
Info
Diagnostics
General
General
Information
Time stamps
Compilation
Protection
Attributes
Name: Main
Constant name: OB_Main
Type: OB
Number: 1
Instructions
Favorites
Instructions
Bit logic
Extended inst...
VAL_STRG
LEN
CONCAT
LEFT
RIGHT
MID
DELETE
INSERT
REPLACE
FIND
Program co...
Communica...
TSEND_C
TRCV_C
Others
PORT_CFG
SEND_CFG
RCV_CFG
SEND_PTP
RCV_PTP
RCV_RST
SGN_GET
SGN_SET
Portal view
Overview
PLC_2
Data_block_1
Main
Project Project3 open
Start
C:\Documents and Se...
7:36 PM

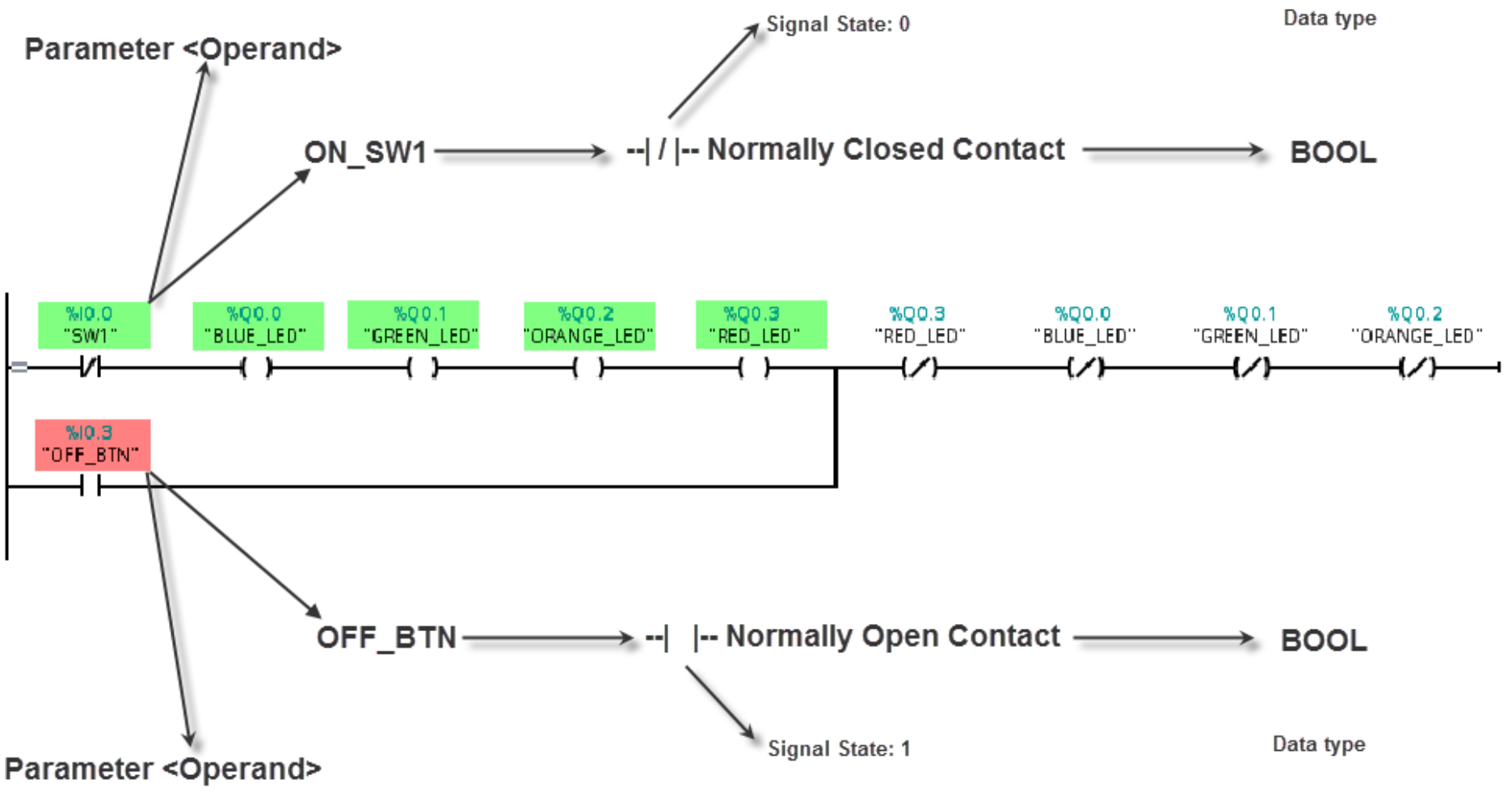
What do those panels attached to the PLCs do and how are they controlled?


Q 0.0
Q 0.1
Q 0.2
Q 0.3
ON
OFF
I 0.0
I 0.1
I 0.2
I 0.3
Ind-Concepts
www.plctrainer.net

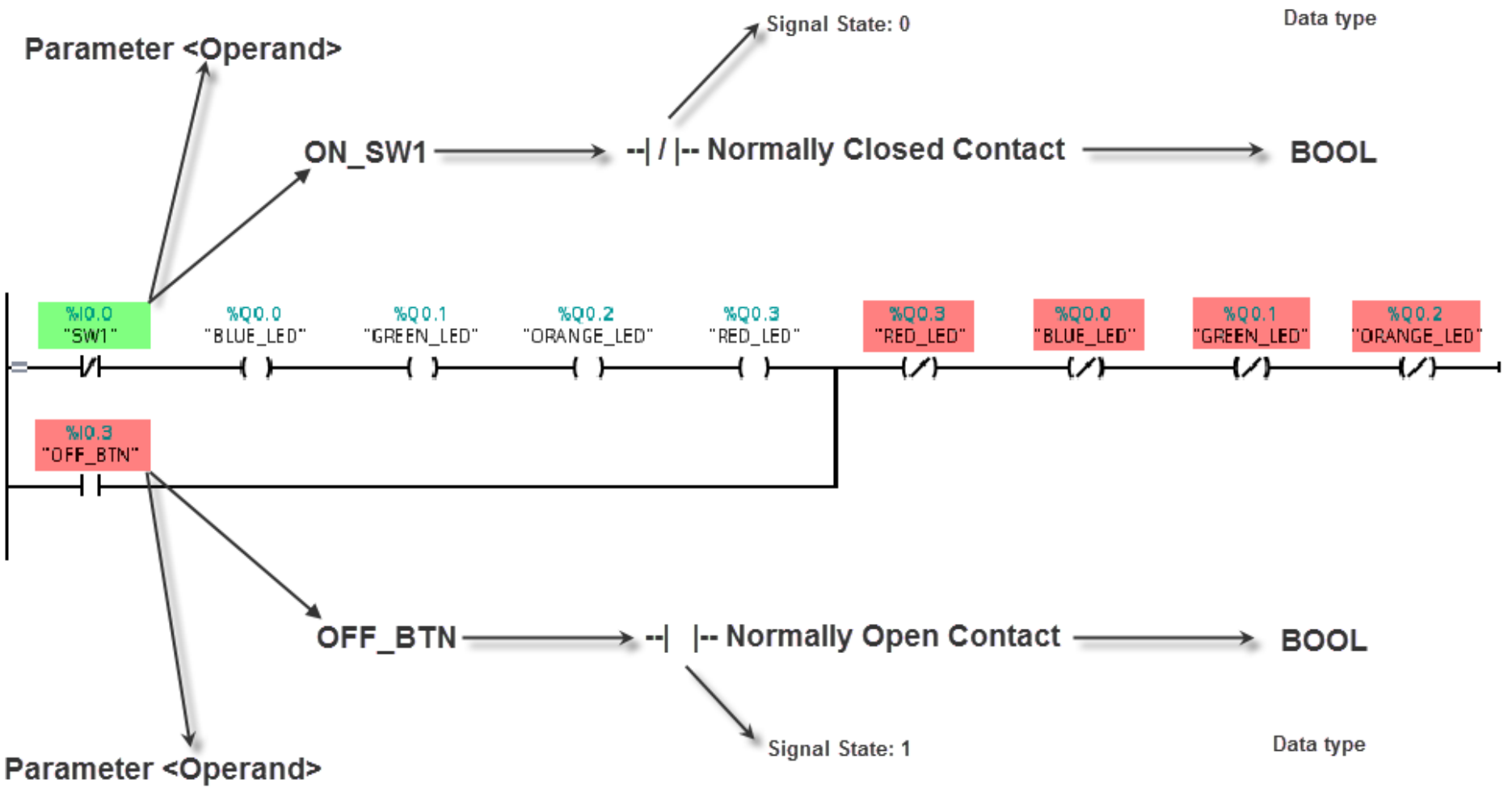
PLC Trainer LAD Network



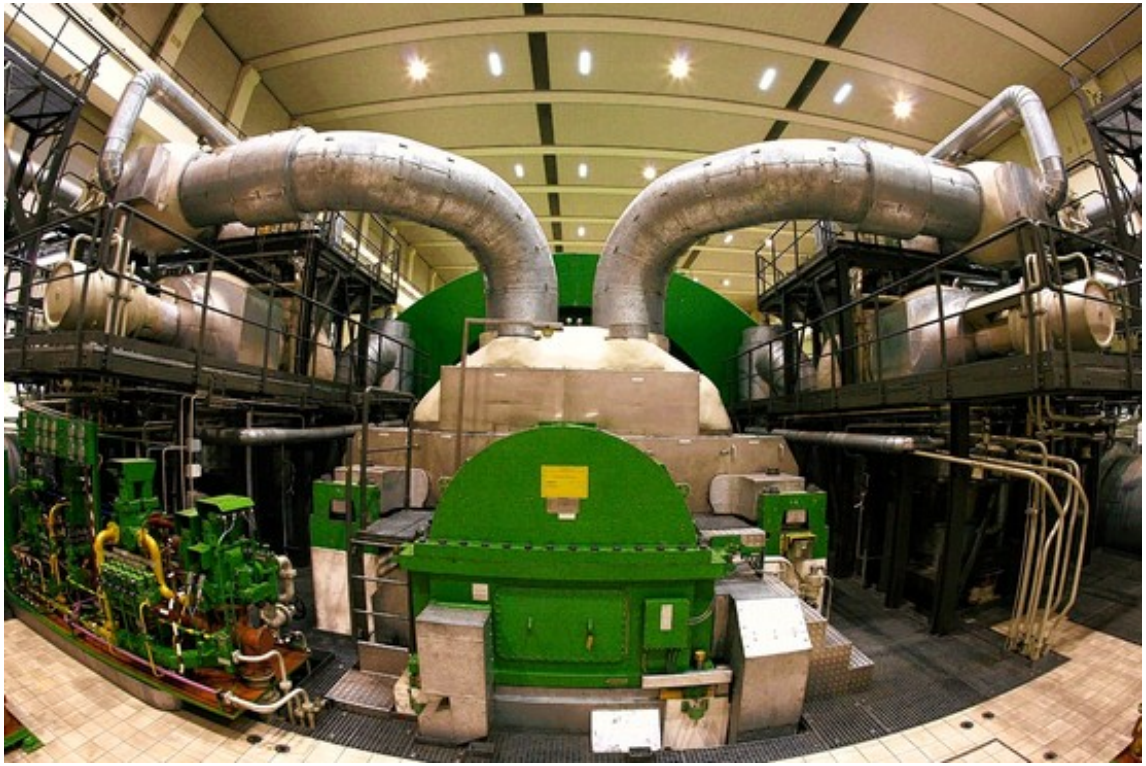
PLC Trainer LAD Network



PLC Trainer LAD Network

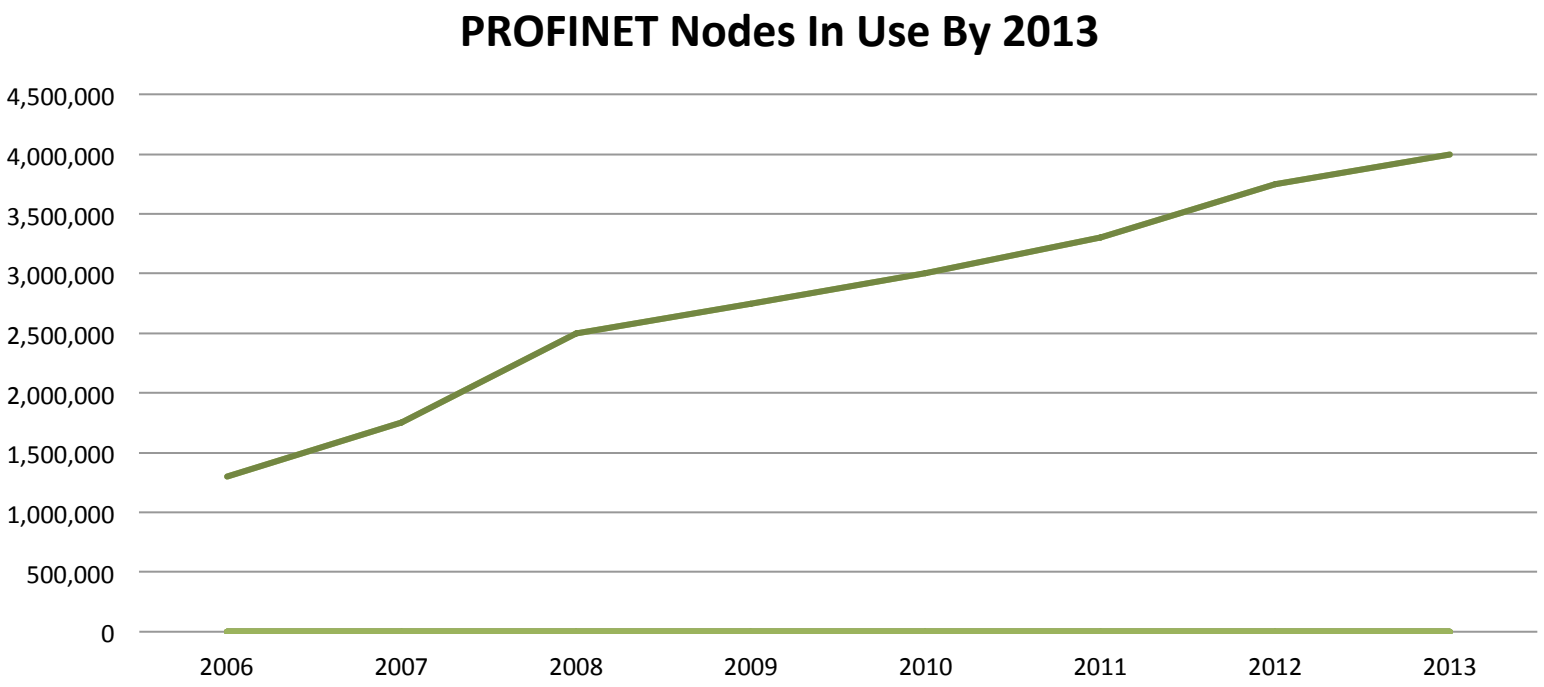


The Siemens Simatic S7-300, S7-400 and S7-1200 rely on the PROFINET IEEE 802.3 Ethernet standard, for industrial grade connectivity in environments where Manufacturing Execution Systems (MES) are critical.



PROFINET/ETHERNET

Today there are over 3.5 million PROFINET enabled devices actively deployed.



ISO-TSAP and Simatic PLCs

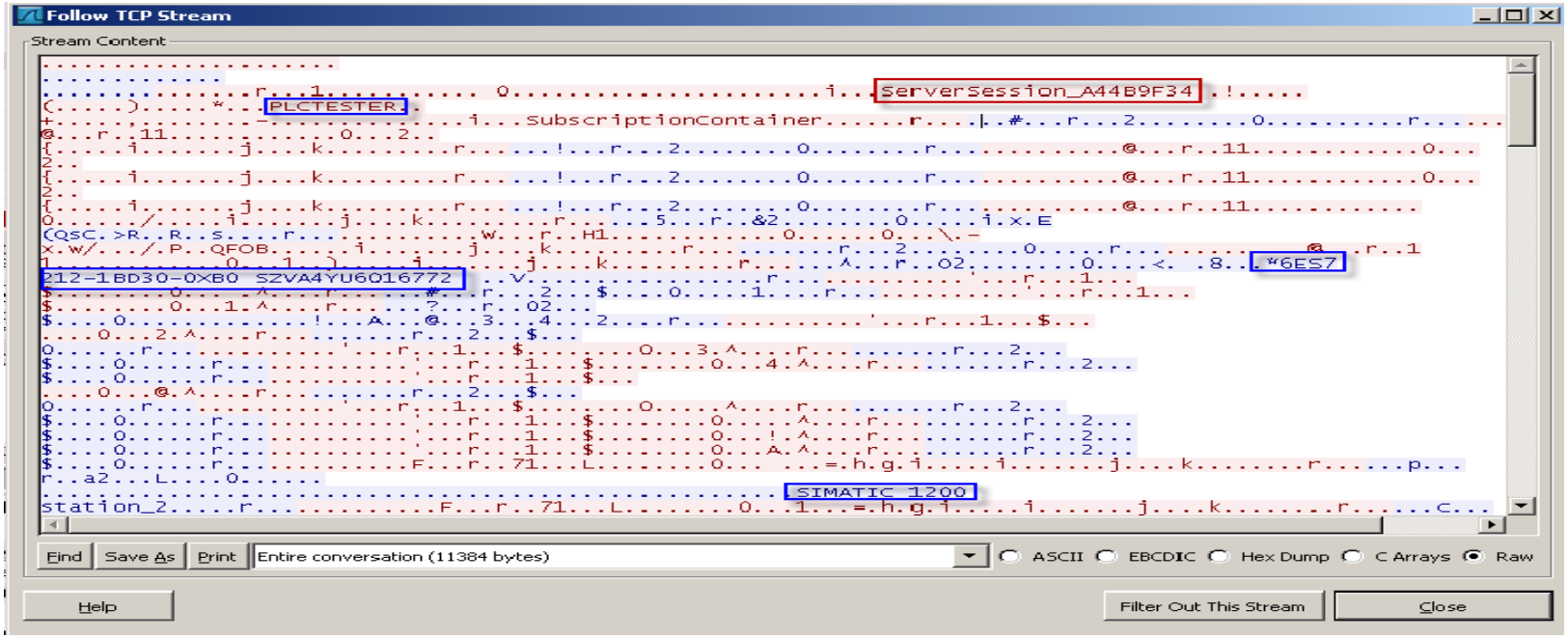
- S7 PLC listens on TCP Port 102 for connections.
- Communicates w/Step 7 software over ISO-TSAP.
- ISO-TSAP is layered on Top of TCP connections.
- S7 PLC also accepts remote commands over 102.
- ISO-TSAP was designed with special-purpose interfaces in mind to be useful in the short term.
- Expedites the process for development of TSAP based applications that need TCP.

ISO-TSAP Problems

- Security was never factored into the equation.
- Packets transmitted o/ISO-TSAP are sent in plain-text.
- The protocol was intended to be open and reliable.
- Uses a ‘layering principle’ which means that even if an encrypted bridge between the client and server on top of the TCP is implemented it can still be MITM
- The protocol hasn’t been revised or even looked at since the late 80s.

Don’t trust Simatic ‘ServerSession’ based connections.

An attacker, can disable or enable the CPU protection of a PLC by sending crafted packets over port 102. The attacker can also control the CPU’s internal operational state, change logical operations in the PLC’s OB1 portion of memory and or shutdown processes connected to the PLC.



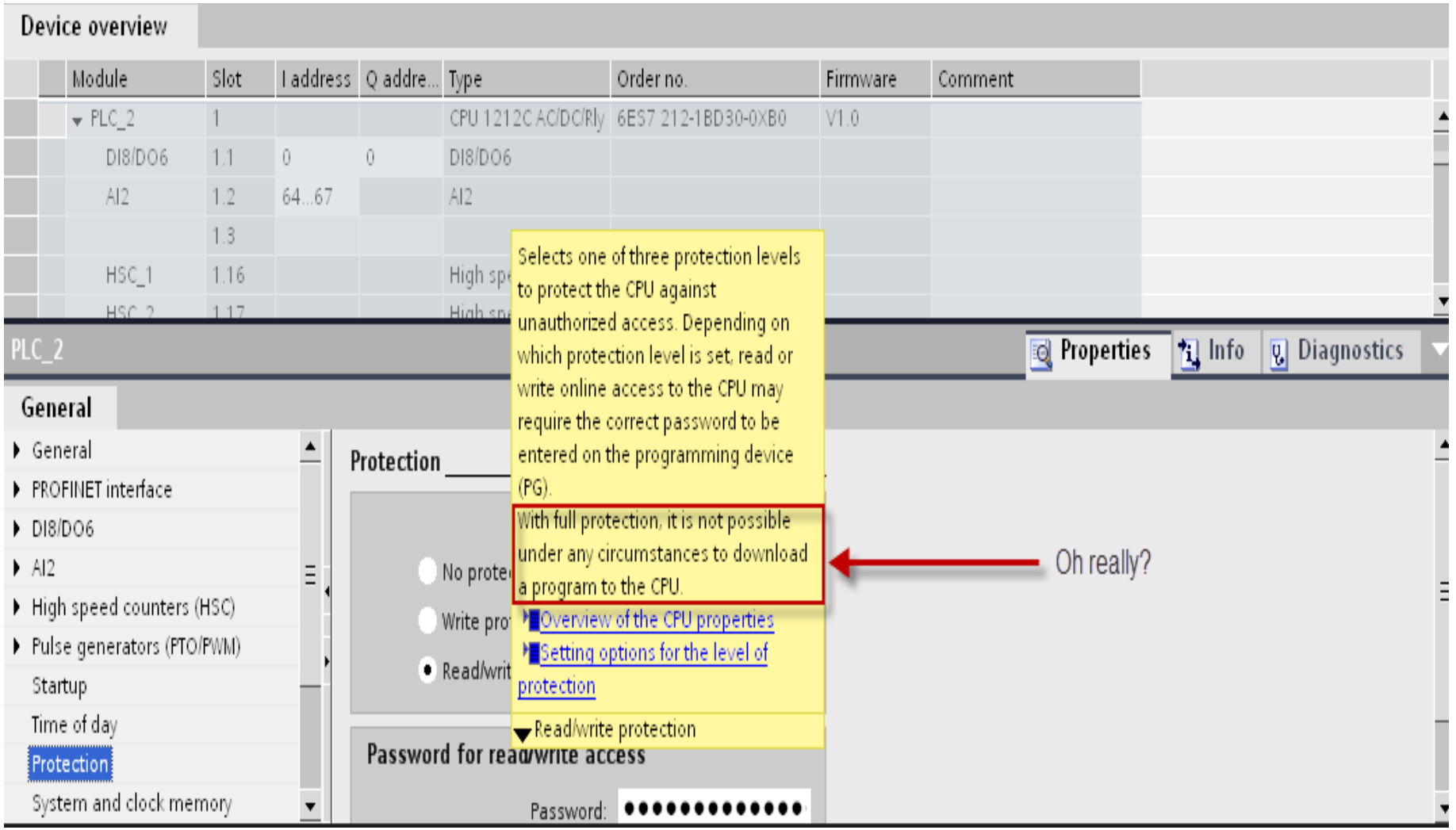
The packets in the red are being generated and sent from the Siemens Step 7 Basic software. The packets in the blue are being sent by the PLC. The entire TCP stream is 11384 bytes. Pay close attention to the ServerSession_ID in the red box, that little ID will play an important role later on.

Siemens claimed that they had a mitigation strategy in place, and they did, but it was flawed.

They call it their ‘password protection’ feature.

They also claim that its not possible, under any circumstances whatsoever, to read or write to memory when the feature is enabled.

S7 Memory Protection



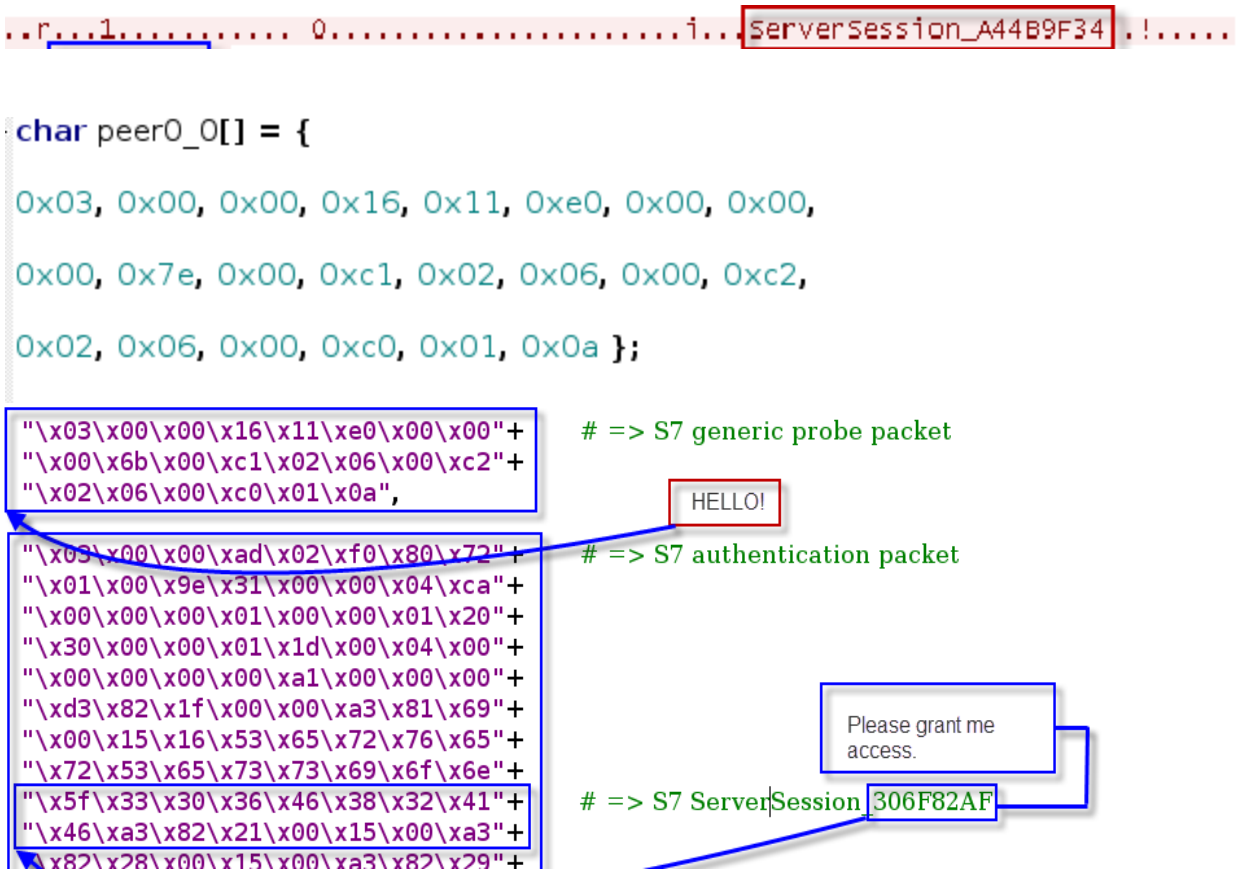
Selects one of three protection levels to protect the CPU against unauthorized access. Depending on which protection level is set, read or write online access to the CPU may require the correct password to be entered on the programming device (PG).
With full protection, it is not possible under any circumstances to download a program to the CPU.

- Overview of the CPU properties
- Setting options for the level of protection

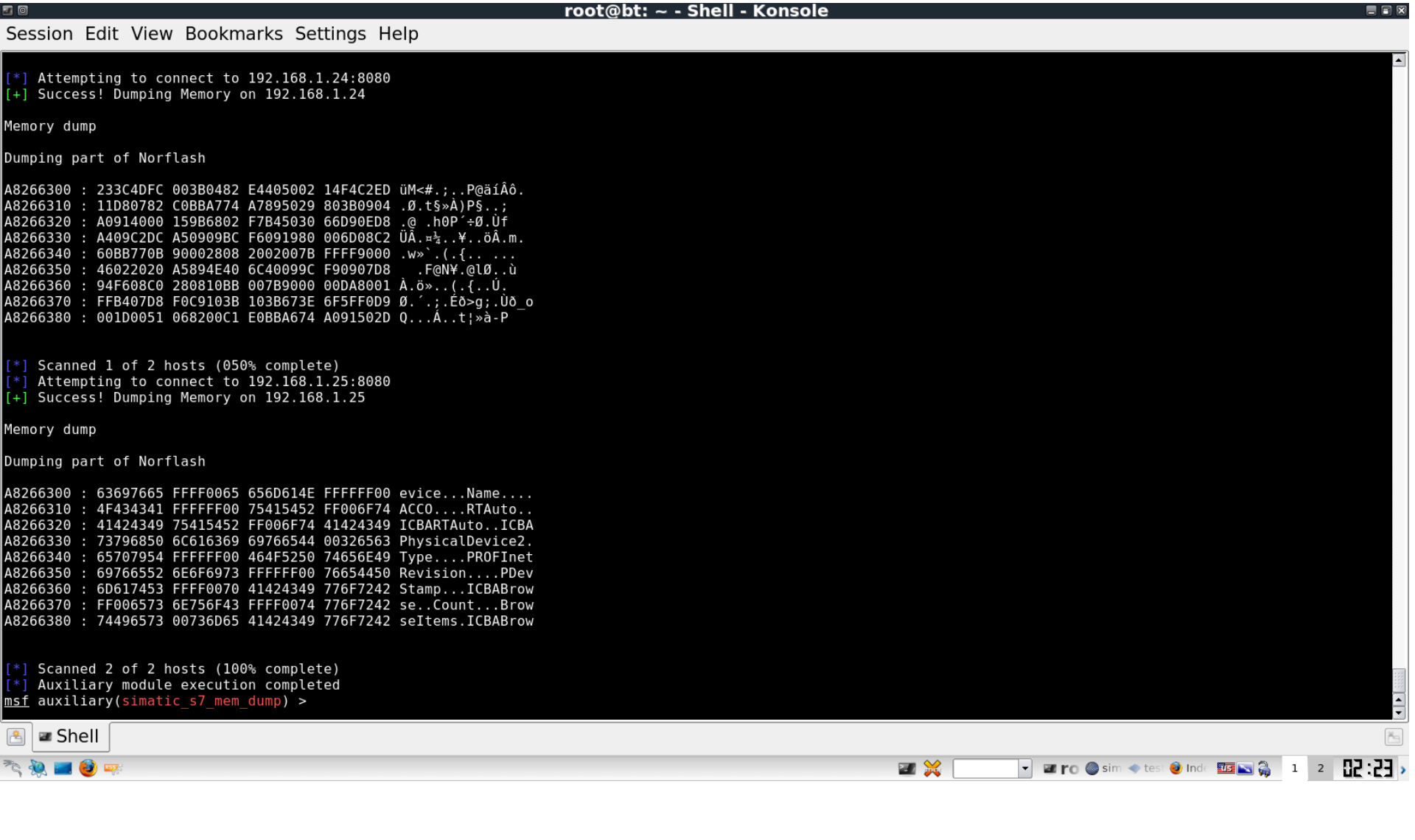
Read/write protection

We don’t need the ‘password’

Packet2Hex



Memory Protection?



S7 Packet-Fu over ISO-TSAP

S7-1200 PLC Memory Protection Delta

root@bt:~/cmp# diff 1234.txt 4321.txt
9,10c9,10
< "\x5f**x33**\x30\x36**\x46\x38\x32\x41**"+ (x41 = 'A')
< "\x46\xa3\x82\x21\x00\x15\x00\xa3"+

> "\x5f**x35**\x30\x36**\x43\x35\x36\x42**"+ (x42 = 'B')
> "\x37\xa3\x82\x21\x00\x15\x00\xa3"+

root@bt:~/cmp#

HEX to ASCII

Server Session

5f 33 30 36 46 38 32 41 46 => **_306F82AF** '1234' 0o,¯ Password => '1234'

5f 35 30 36 43 35 36 42 37 => **_506C56B7** '4321' PlV· Password => '4321'

Flawed S7-PLC Authentication Take-a-ways

If an attacker has captured packets containing the authenticated server session, from the automation network they can re-authenticate using the same packet and bypass that level of protection without ever needing any physical access to the engineering workstation or the PLC.

It is also possible to generate our own library of packets based on a pre-existing packet capture and either crack the password, or brute force our way in and since Siemens didn't enforce expired sessions we can simply use any ServerSession ID we like, against any S7 PLC.

Memory Protection Take-a-ways

- It is possible to read and write data to the PLC's memory even when the password protection is enabled.
- It is possible to retrieve sensitive information from the PLC through memory dumping.
- Its is also possible to disable the password protection feature on the PLC by flipping the security bit back to an OFF state.

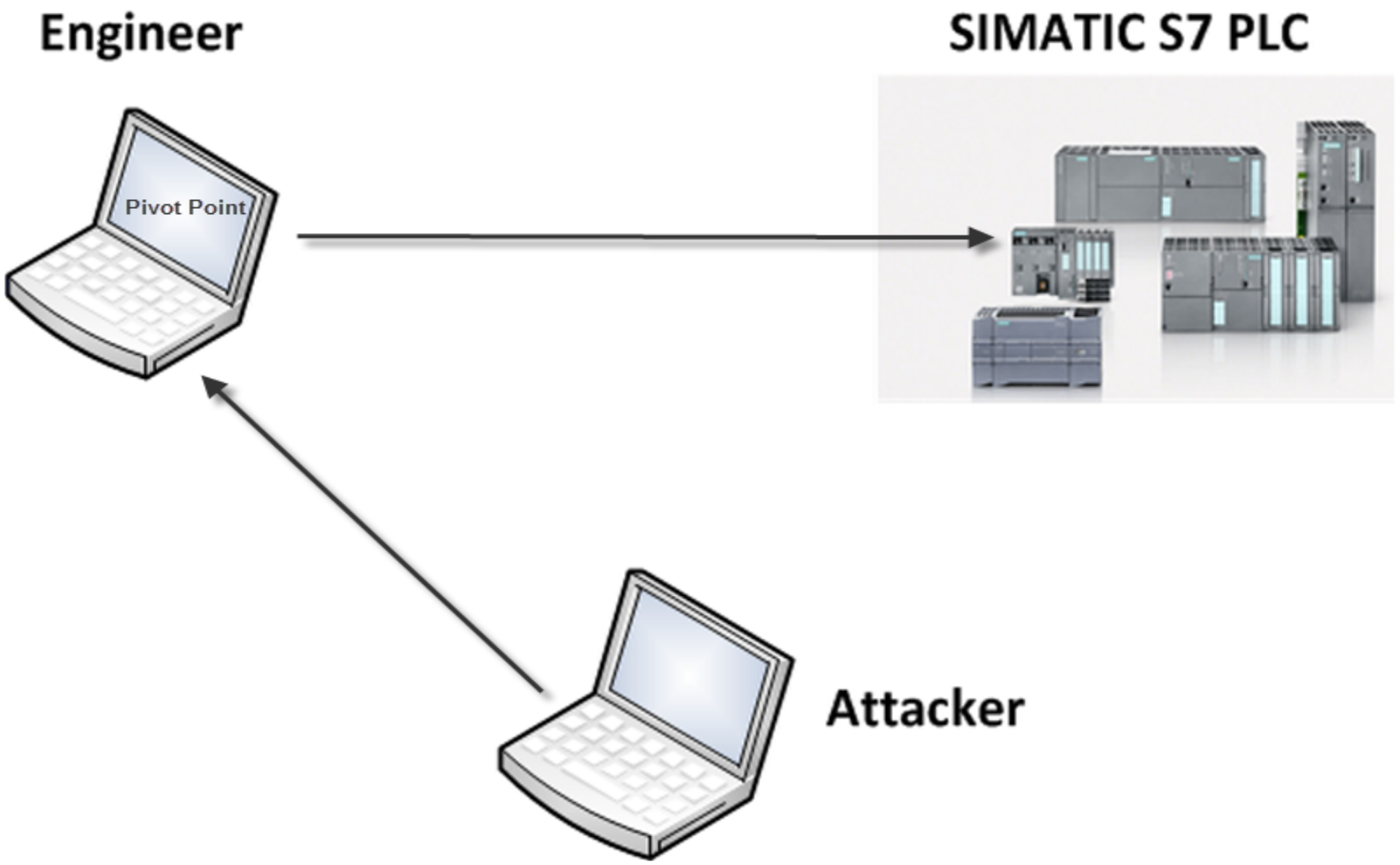
Attack Vector

1. We capture traffic going to and from the engineering workstation and the PLC.
2. We dissect the client portion of the TCP Stream.
3. We build our own packets based on the client portion.
4. We can replay those packets back to the PLC.

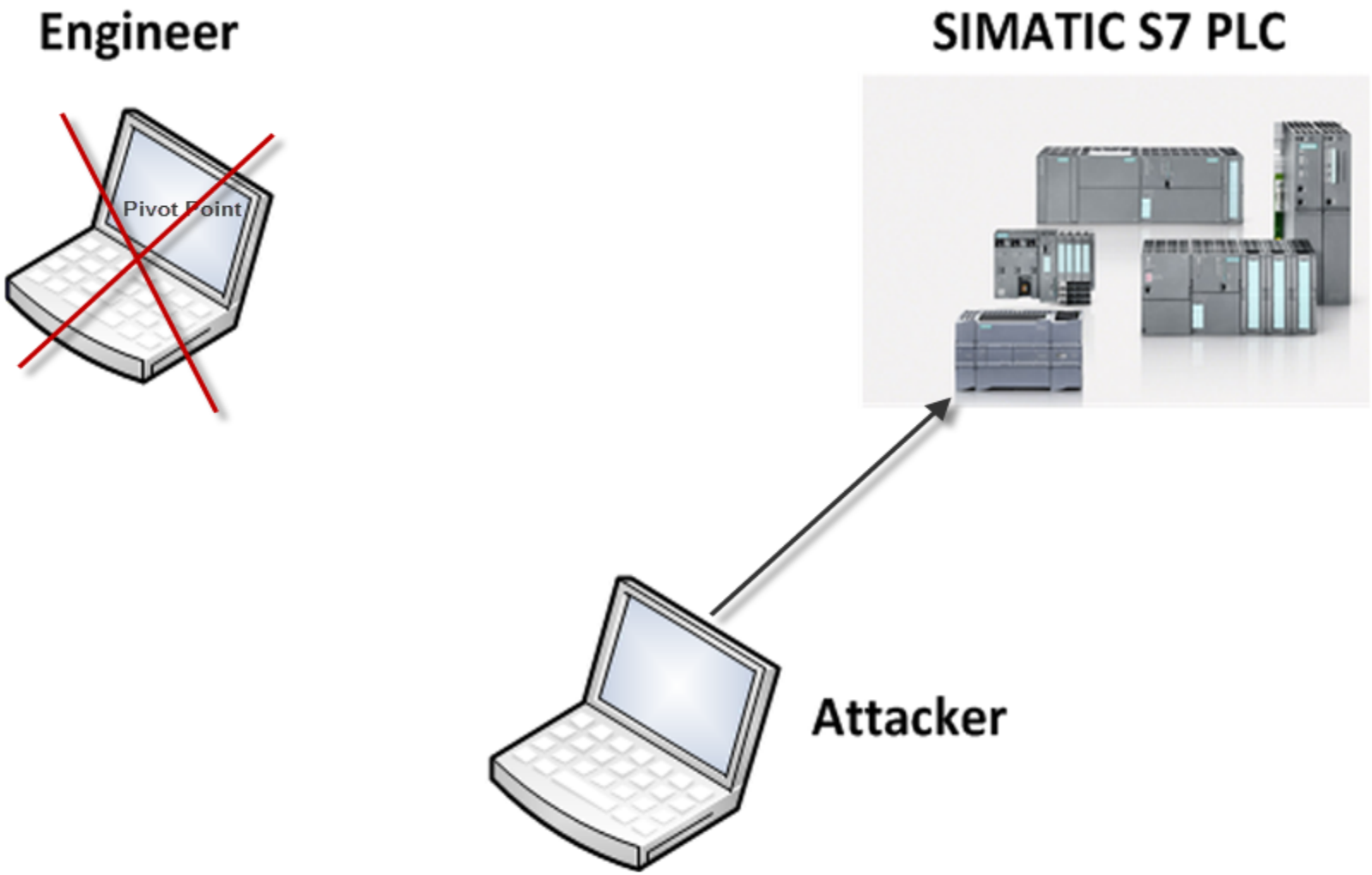
Process

- Scrape device information from memory.
- Change the results of ladder logical operations on the PLC, manipulate the logic to report false data to the operator. Devices in the field could explode or spin out of control!
- Put the CPU in START/STOP mode which could destroy process environments and damage productivity.
- Edit PLC device configuration, change MAC, IP address, device name, Time of Day, and even lock the operator out of their own PLC.
- Frag the PLC by triggering memory leaks in bugs.
- Execute arbitrary commands via command shell using the hardcoded credentials.

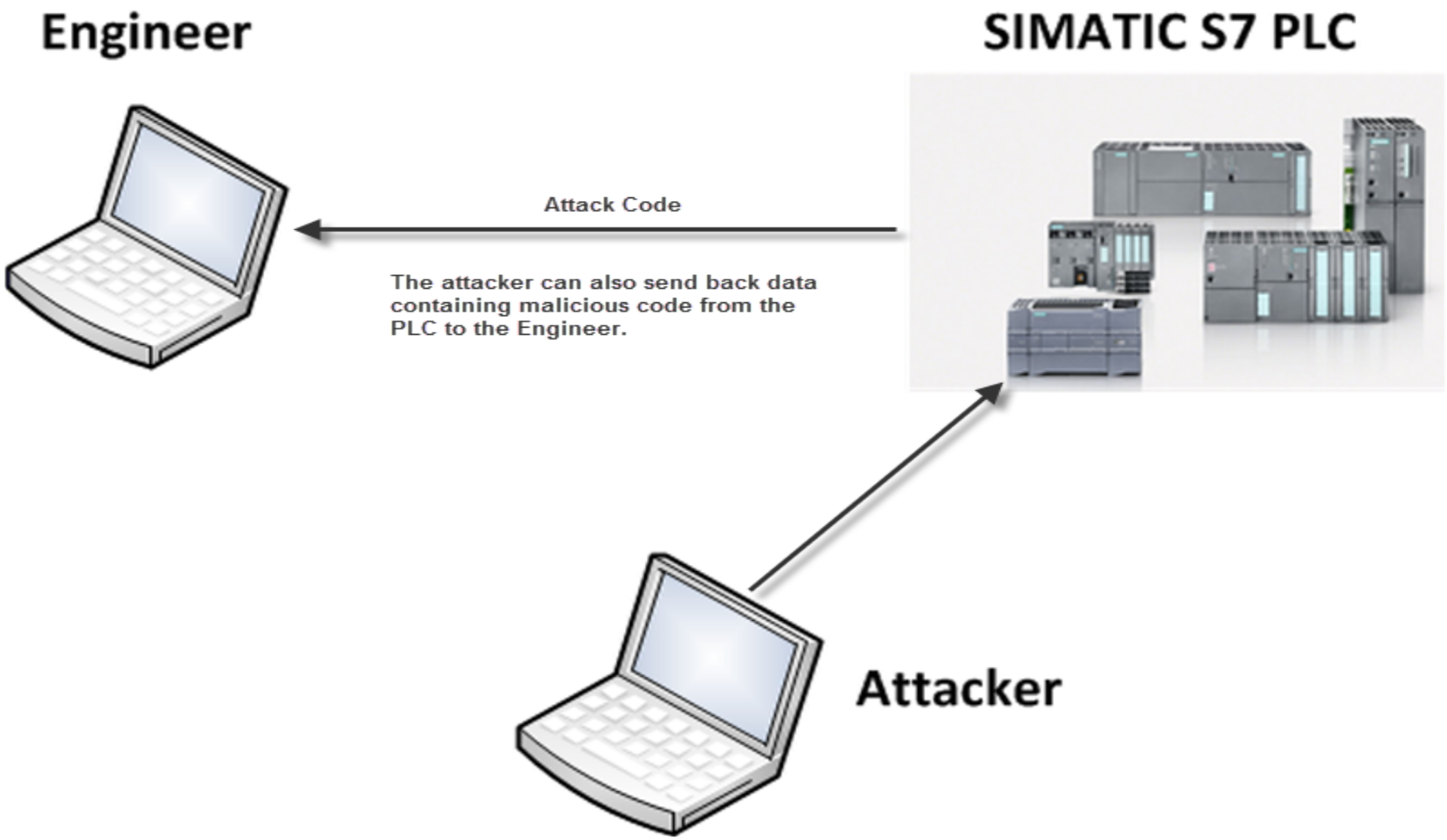
Stuxnet



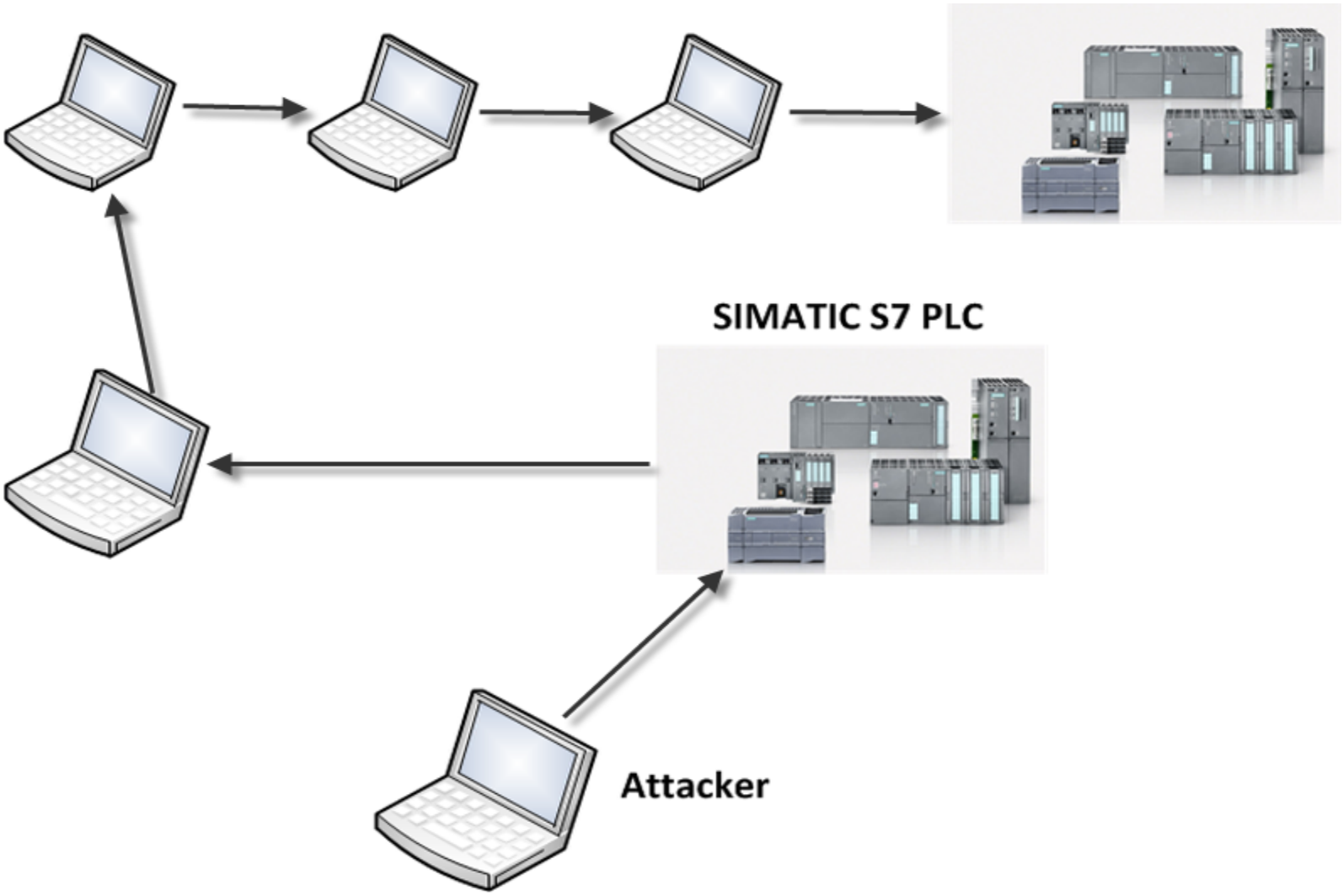
Metasploit



Worm


Engineer
SIMATIC S7 PLC
Attack Code
The attacker can also send back data containing malicious code from the PLC to the Engineer.
Attacker

Propagation

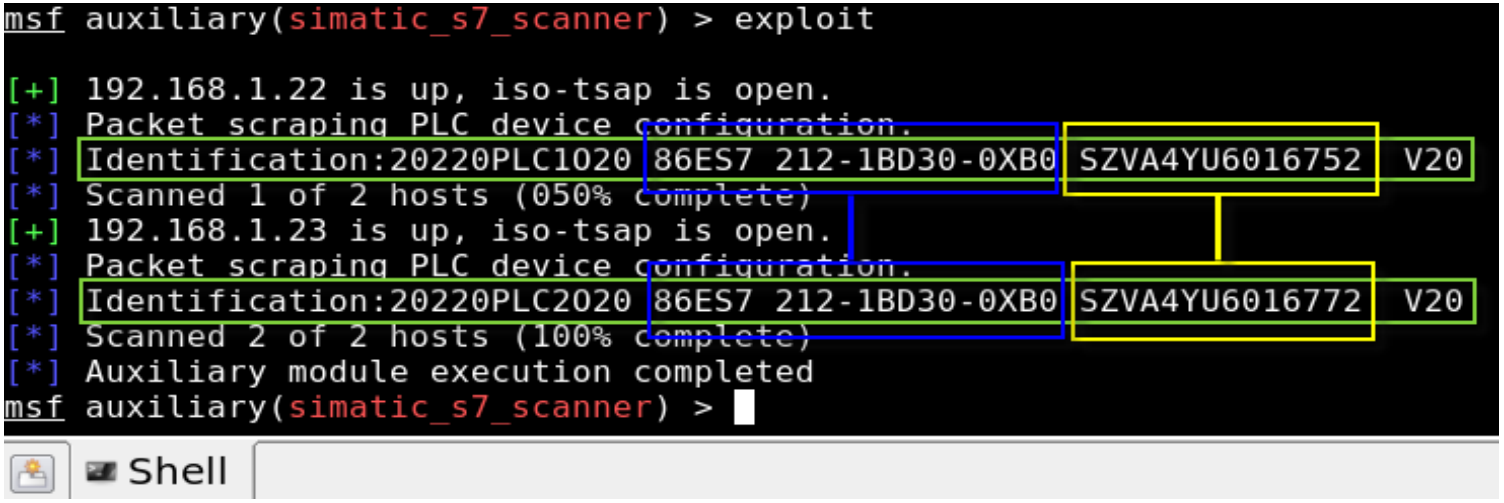


We can own everything on the automation network…

S7 Recon

Metasploit S7-1200 PLC Auxiliary Scanner Module

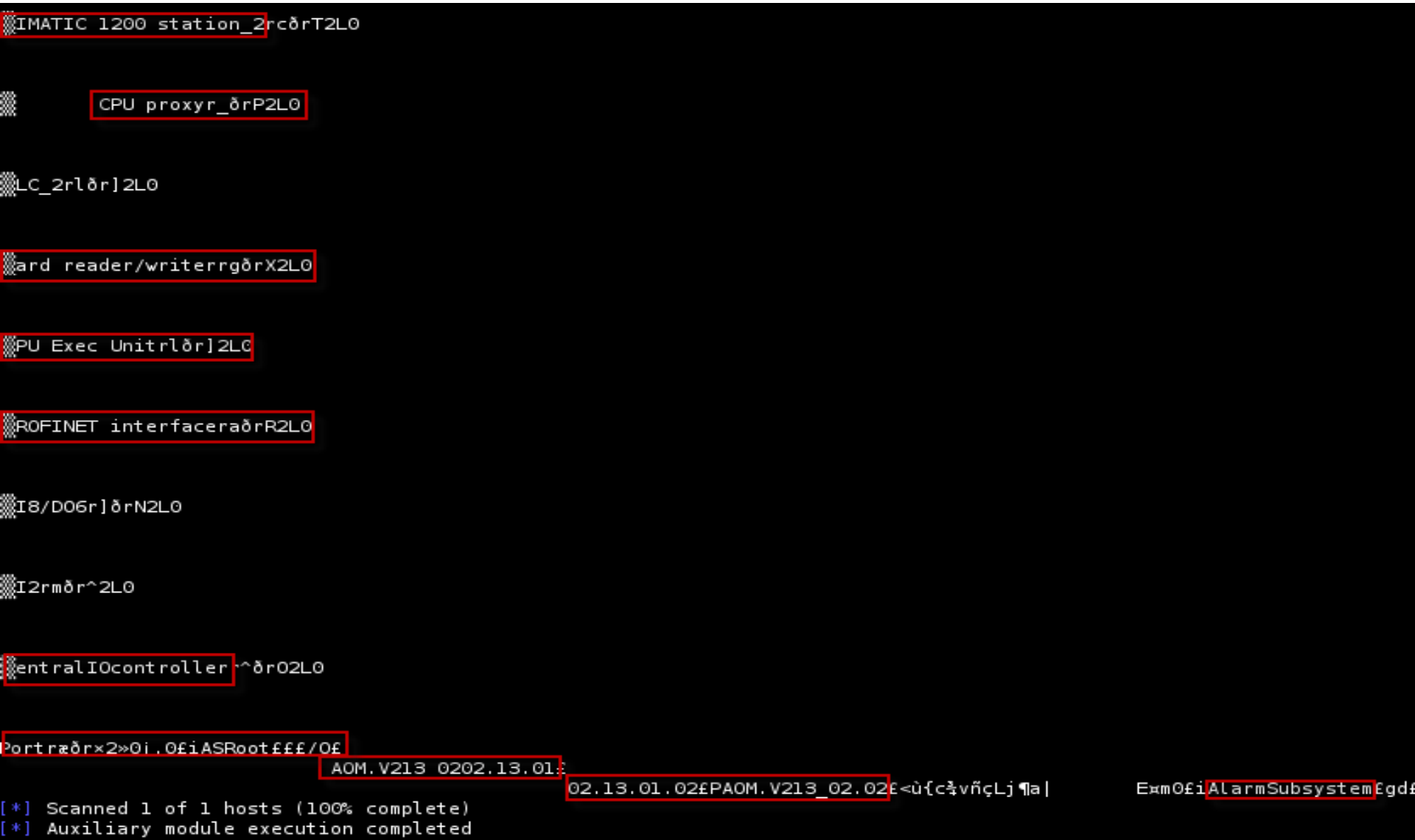
By sending a series of forged probe packet requests an attacker can fingerprint the S7-1200 across a network. This would enable the retrieval of sensitive information. You can grab information such as the serial number, firmware version, model number and PLC name.



S7-1200 PLC Memory Read/Write

- Read device logic about the process connected to the PLC and create targeted attacks based on the information we receive.
- Read/Write boolean operations.
- Read/Write tag names from datablocks
- Disable protection, CPU operations, etc..
- Put the CPU into a perpetual STOP state.

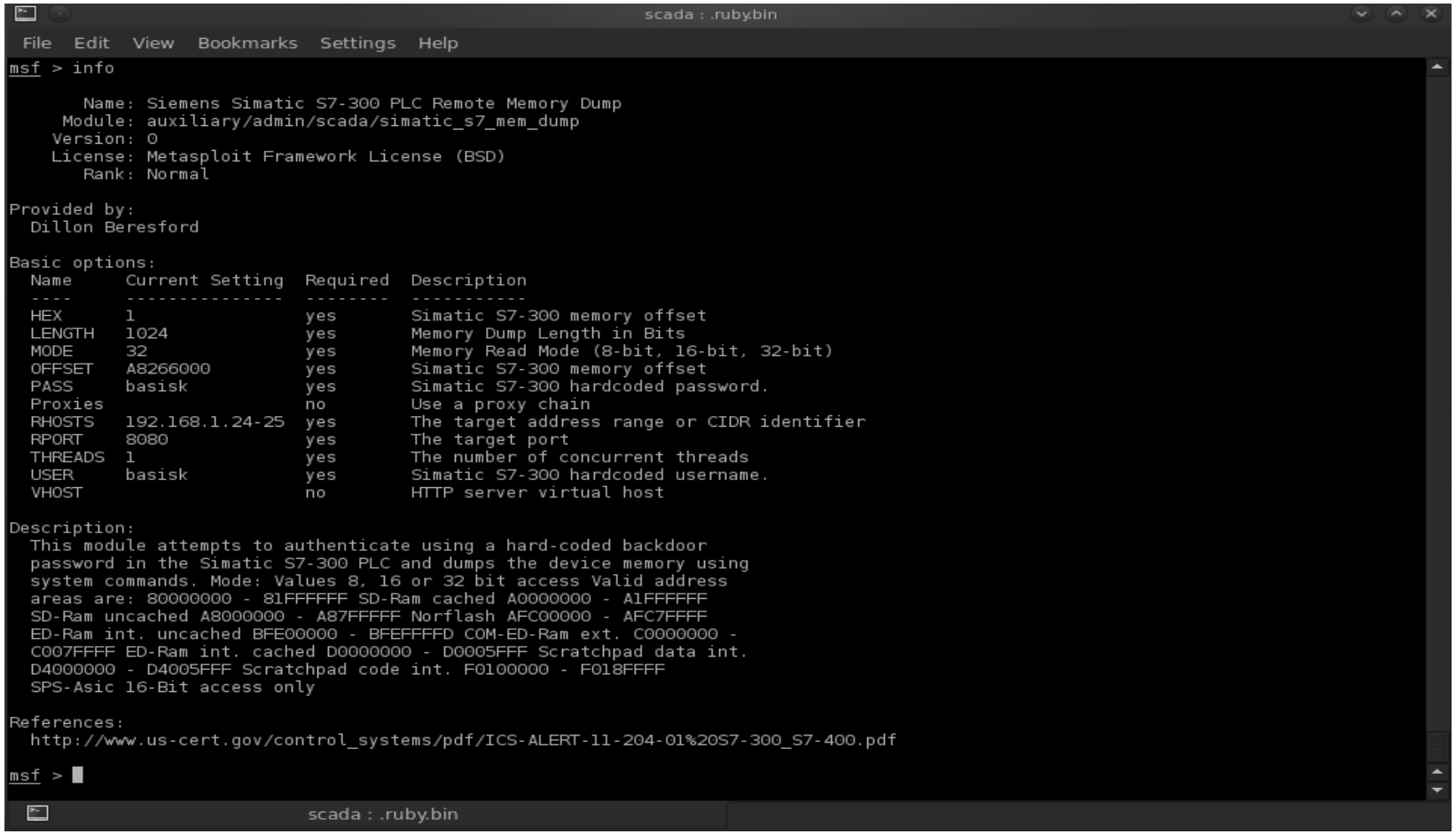
S7-1200 PLC Binary Data from PLC Memory



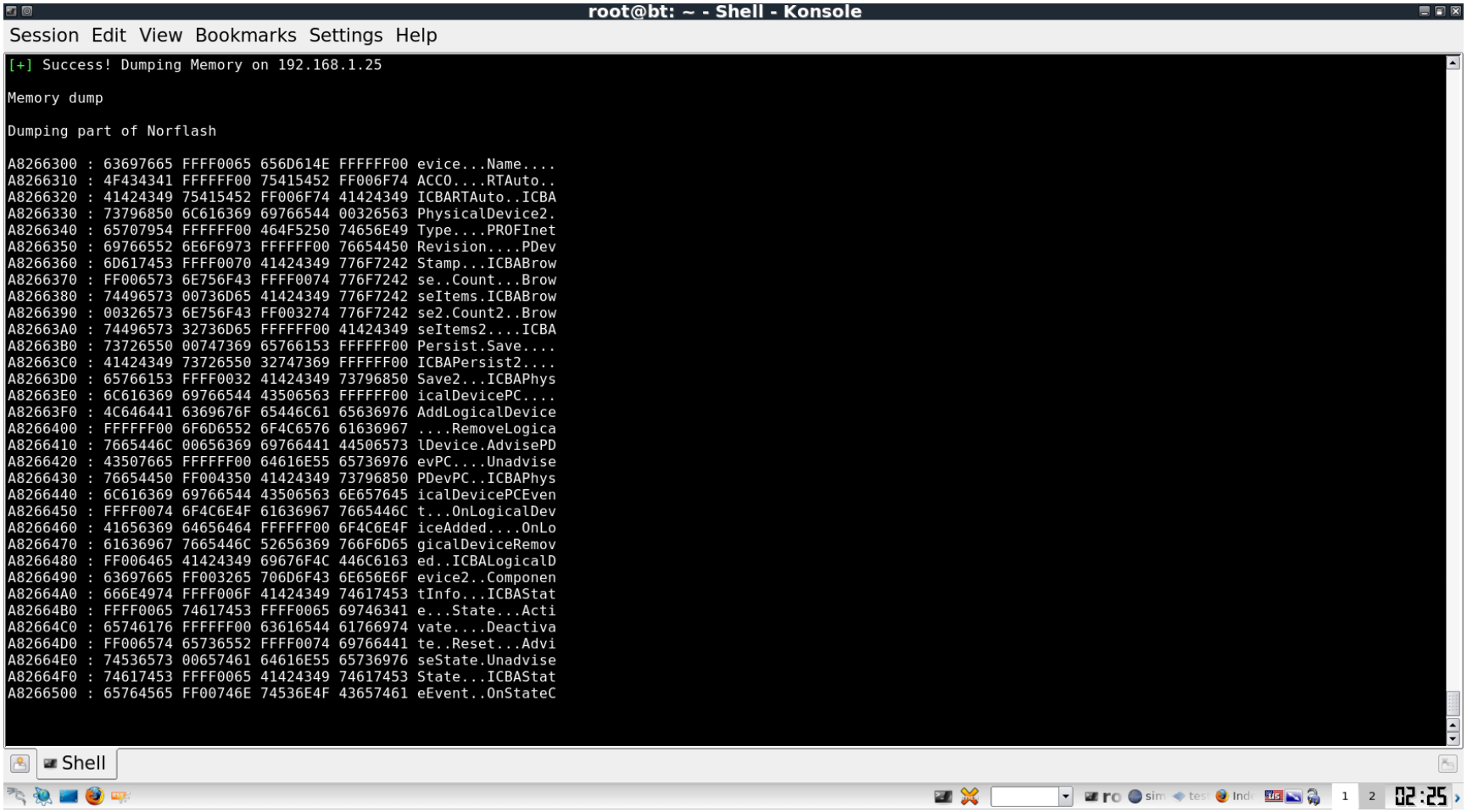
PLC Memory from Data_Block_1


Data_block_1
LED1
LED2
LED3
LED4
QArea
BLUE_LED
RED_LED

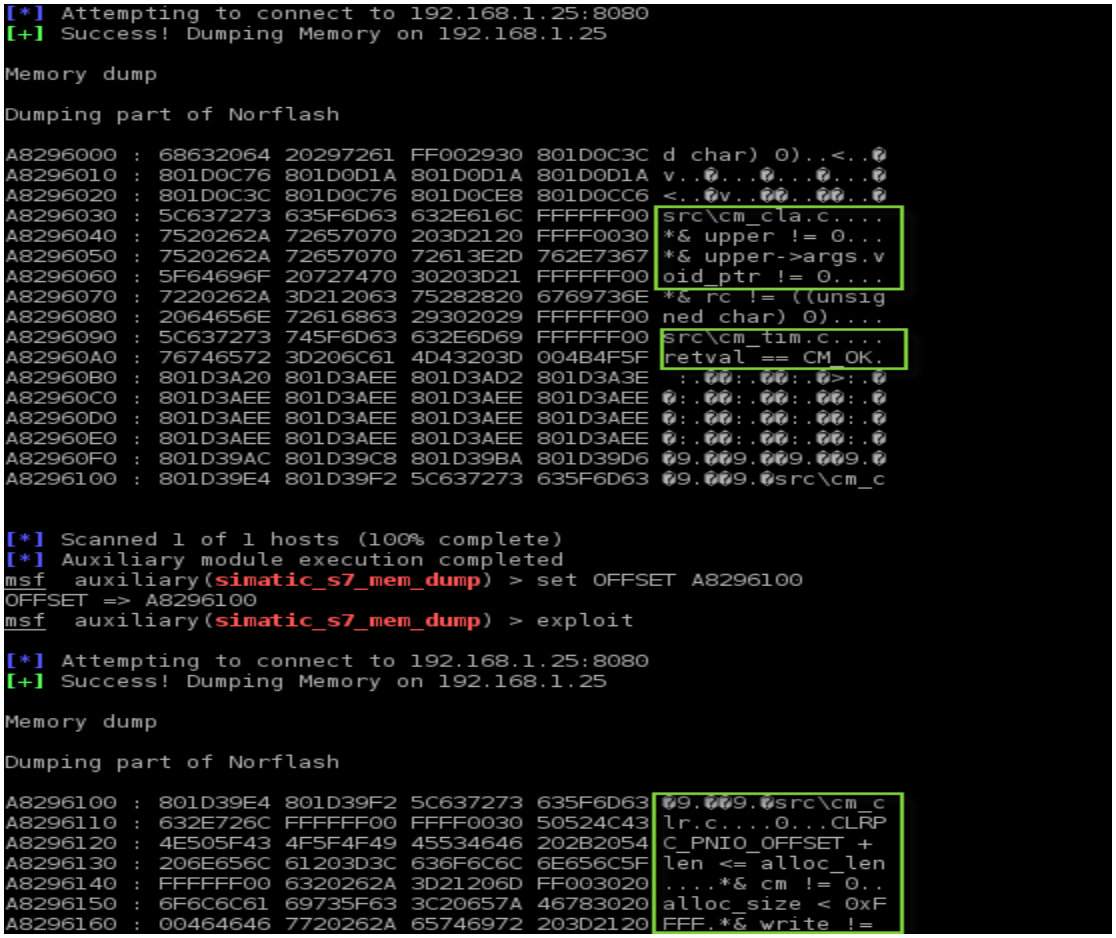
S7 Memory Dump Metasploit Module



Remote Memory Dump
S7-300 Norflash

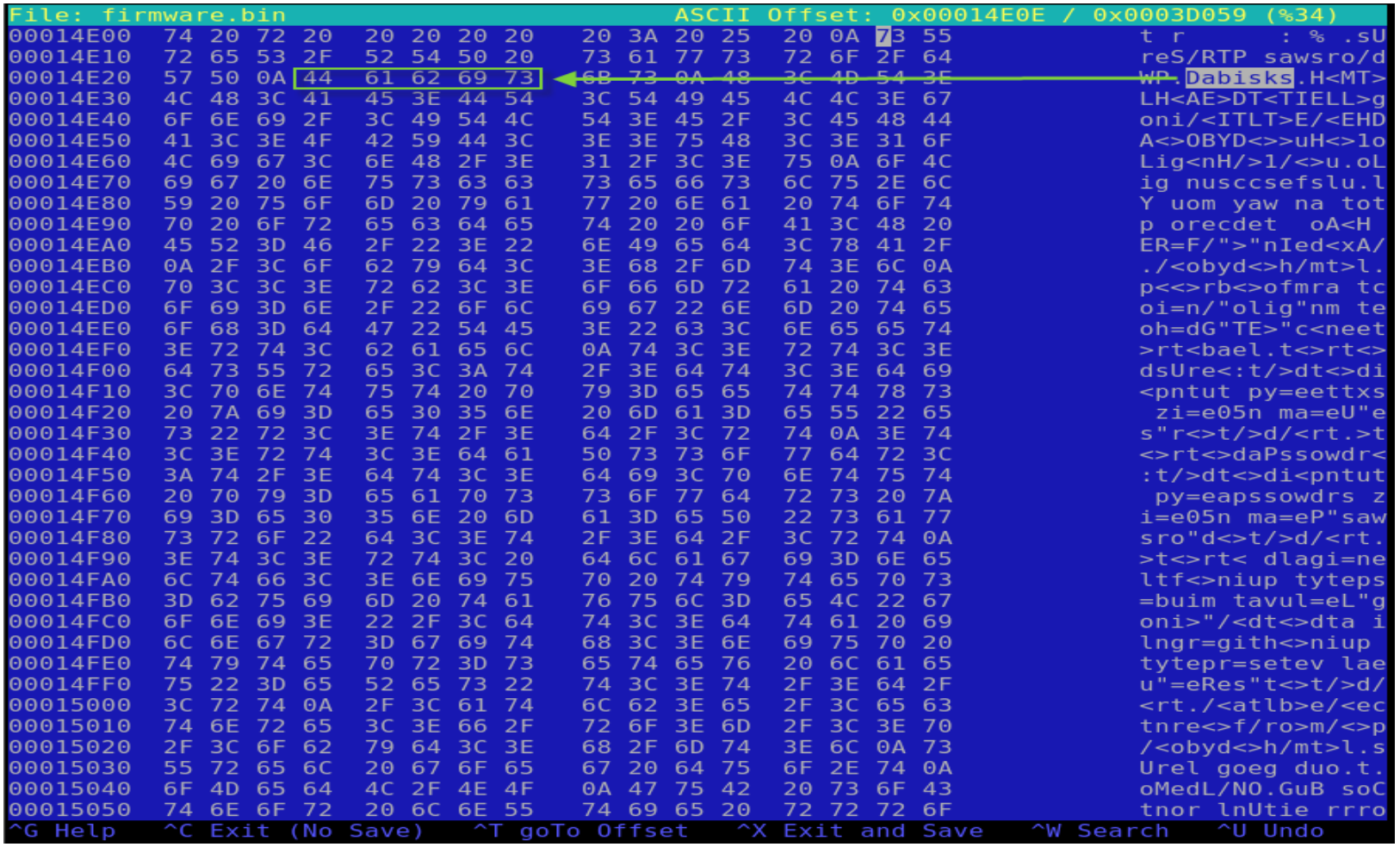


What can we find in dumps?

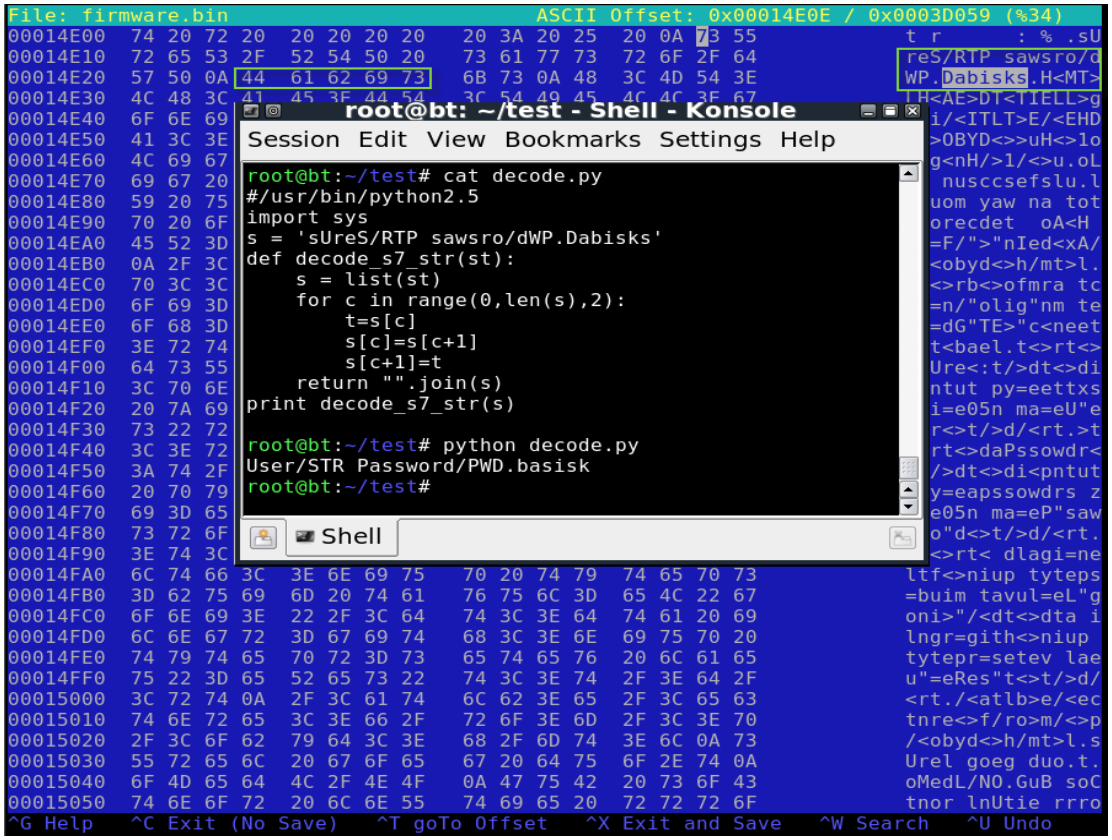


- Code paths
- Source code
- New Bugs

S7-300 Hardcoded Credentials



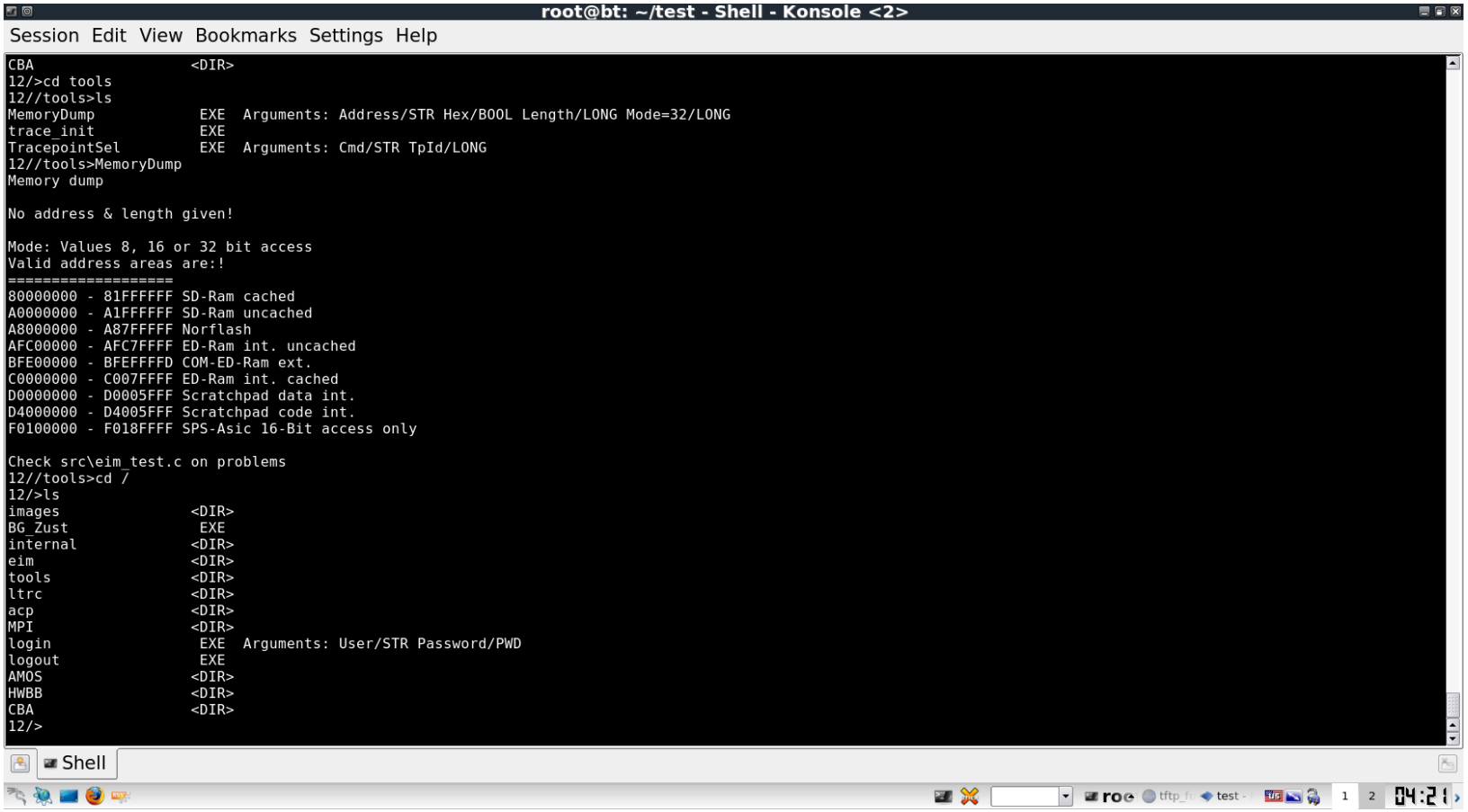
Cracking the S7 Password



User: basisk
Pass: basisk

- Scan thru firmware
- 12 lines of code
- Swap odd chars
- Login via Telnet
- Login via HTTP
- Dump Memory
- Delete Files
- Execute Commands

Owned


root@bt: ~/test - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
CBA <DIR>
12/>cd tools
12//tools>ls
MemoryDump EXE Arguments: Address/STR Hex/BOOL Length/LONG Mode=32/LONG
trace_init EXE
TracepointSel EXE Arguments: Cmd/STR TpId/LONG
12//tools>MemoryDump
Memory dump
No address & length given!
Mode: Values 8, 16 or 32 bit access
Valid address areas are:!
==========
80000000 - 81FFFFFF SD-Ram cached
A0000000 - A1FFFFFF SD-Ram uncached
A8000000 - A87FFFFF Norflash
AFC00000 - AFC7FFFF ED-Ram int. uncached
BFE00000 - BFEFFFFD COM-ED-Ram ext.
C0000000 - C007FFFF ED-Ram int. cached
D0000000 - D0005FFF Scratchpad data int.
D4000000 - D4005FFF Scratchpad code int.
F0100000 - F018FFFF SPS-Asic 16-Bit access only
Check src\eim_test.c on problems
12//tools>cd /
12/>ls
images <DIR>
BG_Zust EXE
internal <DIR>
eim <DIR>
tools <DIR>
ltrc <DIR>
acp <DIR>
MPI <DIR>
login EXE Arguments: User/STR Password/PWD
logout EXE
AMOS <DIR>
HWBB <DIR>
CBA <DIR>
12/>
Shell
1
2
04:21

Decoding the S7 password

- Took 1-2 hours to locate the actual password
- The result was command shell.

```
s = 'sUreS/RTP sawsro/dWP.Dabisks'
def decode_s7_str(st):
    s = list(st)
    for c in range(0,len(s),2):
        t=s[c]
        s[c]=s[c+1]
        s[c+1]=t
    return "".join(s)
print decode_s7_str(s)
```

Metasploit Modules

- simatic_s7_1200_cpu_cmd.rb
- simatic_s7_300_cpu_cmd.rb
- simatic_s7_mem_dump.rb
- simatic_s7_1200_cpu_mem_write.rb
- simatic_s7_disable_mem_protect.rb
- simatic_s7_cpu_cmd_protect_bypass.rb

Demo