Ghost in the PLC
Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack

ALI ABBASI
SYSSEC GROUP,
RUHR UNIVERSITY BOCHUM, GERMANY
& SCS GROUP
UNIVERSITY OF TWENTE, NETHERLANDS

MAJID HASHEMI
QUARKSLAB, PARIS
Who we are

- Ali Abbasi, visiting researcher at chair of system security of Ruhr University Bochum and PhD student at Distributed and Embedded Systems Security Group, University of Twente, The Netherlands.

- Majid Hashemi, R&D researcher at Quarkslab, involved in this research as an independent researcher.
Plan of talk

• Background on existing attacks and defenses for embedded systems
• Applicable Defenses for PLCs
• Background on Pin Control
• The Problem
• Rootkit variant
• Non-rootkit variant
• Demo
• Discussions
What this talk is not about?

• The talk is trying to uncover existing design flaw in PLCs.
• The attack can be used in future by attackers.
• Today, there are far easier attacking techniques but it is trivial to defeat them.
• We are not unveiling fully functional rootkit for PLCs.
• NO exploitation techniques, no 0day leak
• We are not going to mention any vendor name.
What is Critical Infrastructure

- Electrical
What is Critical Infrastructure

- Water
What is Critical Infrastructure

- Gas
What is Critical Infrastructure

• Military

Steuerten Hacker Raketenstationen der Bundeswehr?

Hacker haben womöglich das Flugabwehrrsystem Patriot geknackt: In der Türkei stationierte Raketenstationen der Bundeswehr hätten "unerklärliche" Befehle ausgeführt, berichtet eine Fachpublikation.
What is Critical Infrastructure

• Elements of infrastructure that, if lost, could pose a significant threat to needed supplies, services, and communication\(^1\).

Who are the attackers

- Mostly State Sponsored
- Use 0days
- Use Sophisticated evasion techniques
- Best Example? Stux...
How to attack critical infrastructures?

• Get into the network without being detected.
  • Defeating Emulation Based Network Intrusion Detection Systems
    • APTs Way: Evading Your EBNIDS, Black Hat Europe, 2014.

• Manipulate the PLCs
  • The PLC logic

• Manipulate the process
  • Damn vulnerable chemical processes²

What is a PLC?

• An Embedded System with RTOS running logic.
What is Logic

• Logic is a program PLC executes
How PLC executes the Logic

- Logic
- Physical I/O
Chapter One

Existing Attacks and Defenses for Embedded Systems
Applicable to the PLCs
Current attacks against embedded systems

- Firmware modification attacks
  - Attacker upload new firmware to the PLC
- Configuration manipulation attacks
  - Attacker modify the logic
- Control Flow attacks
  - Attacker find a buffer overflow or RCE in the PLC
- Authentication bypass
  - Attacker find a backdoor password in the PLC.
- Hooking functions for ICS malwares (e.g. Stuxnet)
Current defenses for embedded systems

• Attestation
  • memory attestation

• Firmware integrity verification
  • Verify the integrity of firmware before its being uploaded

• Hook detection
  • Code hooking detection
    • Detect code hooking
  • Data hooking detection
    • Detect data hooking
Applicable Defenses for PLCs

- Designed for embedded devices running modern OS.
- No hardware modifications.
- Limited CPU overhead.
- No virtualization support required.
Non-trivial System-level protection for PLCs

- Trivial Defenses:
  - Logic Checksum
  - Firmware integrity verification

- Non-trivial software-based HIDS applicable to PLCs
  - Doppelganger (Symbiote Defense): an implementation for software symbiotes for embedded devices

- Autoscopy JR: A host based intrusion detection which is designed to detect kernel rootkits for embedded control systems
How Doppelganger Works

• Scan the firmware of the device for live code regions and insert symbiotes randomly.
How Autoscopy Jr works

- Tries to Detects function hooking by learning
- Verifies the destination function address and returns with the values and addresses in TLL (Trusted Location List)
Function hooking

**Code Hook**

- Code: `call doWriteOperation()`
  - Diverted Control Flow
  - `HOOK`
  - `Function doWriteOperation() ...

**Data Hook**

- System Call Table
  - `Function 1 Pointer`
  - `MODIFIED Pointer`
  - `Function 3 Pointer`
  - `Function 4 Pointer`

- `HOOK`
  - `Function 3`
  - `Function 2`
  - `Function 1`
Solution Found?

- Looks like if we deploy both Doppelganger and Autoscopy Jr, the problem will be solved.

- Autoscopy Jr detects code hooking and doppelganger verify static parts and detect data hooking.
Dynamic Memory (Heap)

- Doppelganger cannot verify dynamic memory.
Dynamic Memory (Heap)

- Doppelganger cannot verify dynamic memory.

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a; // goes on stack
    int *p;
    p = (int*)malloc(sizeof(int));
}
```
Limitations

• Static Referencing
  • Both Autoscopy Jr and Doppelganger use static references, similar to signature-based approaches.

• Dynamic Memory
  • Doppelganger only verify static memory

• Function Hooking Definition
  • The function hooking definition of Autoscopy Jr is incomplete
Chapter Two

Background on Pin Control
Background on Pin Control

Pin Control Subsystem

- Pin Configuration
- Pin Multiplexing
BCM2835 and available I/O Functions
Pin Configuration

• **Input Pin**
  - readable but not writeable

• **Output Pin**
  - readable and writeable
Security concerns regarding Pin Control in PLCs

1. No Interrupt for Pin Configuration
   • How the OS knows about the modification of pin configuration?

2. No Interrupt for Pin Multiplexing
   • What if somebody multiplex a Pin at runtime?
Introducing Pin Control Attack: A Memory Illusion

Operating System/Kernel

PLC Runtime

Physical I/O Memory

State Register

Write register

Read register
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

PLC Runtime

Operating System/Kernel

Physical I/O Memory

State Register

Write register

Read register

<table>
<thead>
<tr>
<th>Processor</th>
<th>Type</th>
<th>Connect.</th>
<th>Polarity</th>
<th>Value</th>
<th>Default</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPIO04</td>
<td>Disconnection</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>configuration of GPIO04</td>
</tr>
<tr>
<td>GPIO07</td>
<td>Disconnection</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>configuration of GPIO07</td>
</tr>
<tr>
<td>GPIO10</td>
<td>Disconnection</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>configuration of GPIO10</td>
</tr>
<tr>
<td>GPIO12</td>
<td>Disconnection</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>configuration of GPIO12</td>
</tr>
<tr>
<td>GPIO14</td>
<td>Disconnection</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>configuration of GPIO14</td>
</tr>
</tbody>
</table>

black hat EUROPE 2016
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

Operating System/Kernel

map via MMU

PLC Runtime

Physical I/O Memory

State Register

Write register

Read register

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Default</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G0[0]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[0]</td>
</tr>
<tr>
<td>G0[1]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[1]</td>
</tr>
<tr>
<td>G0[2]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[2]</td>
</tr>
<tr>
<td>G0[3]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[3]</td>
</tr>
<tr>
<td>G0[6]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[6]</td>
</tr>
<tr>
<td>G0[7]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[7]</td>
</tr>
<tr>
<td>G0[8]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[8]</td>
</tr>
<tr>
<td>G0[9]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[9]</td>
</tr>
<tr>
<td>G0[10]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[10]</td>
</tr>
<tr>
<td>G0[12]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[12]</td>
</tr>
<tr>
<td>G0[13]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[13]</td>
</tr>
<tr>
<td>G0[14]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[14]</td>
</tr>
<tr>
<td>G0[15]</td>
<td>Discharge of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>configuration of G0[15]</td>
</tr>
</tbody>
</table>

black hat EUROPE 2016
Introducing Pin Control Attack: A Memory Illusion

Operating System/Kernel

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

PLC Runtime

Virtual I/O Memory (mapped)

State Register
Write register
Read register

map via MMU

Physical I/O Memory

State Register
Write register
Read register
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

map via MMU

Physical I/O Memory

State Register

Write register

Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

map via MMU

CPU
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

PLC Runtime

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/Kernel

Virtual I/O Memory (mapped)

State Register
Write register
Read register

map via MMU

Physical I/O Memory

State Register
Write register
Read register

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Write register
Virtual I/O Memory (mapped)

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Write register
Virtual I/O Memory (mapped)

Physical I/O Memory
Introducing Pin Control Attack: A Memory Illusion

Operating System/Kernel

Virtual I/O Memory (mapped)

State Register

Read register

Write register

Physical I/O Memory

Map (I/O Memory, +16 bytes)

Request for mapping the physical I/O Memory

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

0 for bit 24
1 for bit 22

0
1

State Register

Write register

Read register

map via MMU

Full black background
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

PLC Runtime

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Write register
Virtual I/O Memory (mapped)

Read register
State Register

0 for bit 24
1 for bit 22

Write register
State Register

Physical I/O Memory

map via MMU

0 for bit 24
1 for bit 22

Write register
State Register

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

Virtual I/O Memory (mapped)

State Register

Write register

Read register

Physical I/O Memory

State Register

Write register

Read register

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

0 for bit 24
1 for bit 22

map via MMU

40
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/Kernel

Map via MMU

Virtual I/O Memory (mapped)

Read register

Write register

State Register

0 for bit 24
1 for bit 22

Physical I/O Memory

State Register

0 for bit 24
1 for bit 22

Write register

Read register

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

PLC Runtime

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Introducing Pin Control Attack: A Memory Illusion

Read register

Write register

State Register

0 for bit 24
1 for bit 22

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/Kernel

Map via MMU

Virtual I/O Memory (mapped)

Read register

Write register

State Register

0 for bit 24
1 for bit 22

Physical I/O Memory

State Register

0 for bit 24
1 for bit 22

Write register

Read register

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register

Write register

Read register

Virtual I/O Memory (mapped)

0 for bit 24
1 for bit 22

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Read register

Write register

0 for bit 24
1 for bit 22

map via MMU
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

map via MMU

Virtual I/O Memory (mapped)

State Register

Write register

Read register

0 for bit 24
1 for bit 22

1

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

0 for bit 24
1 for bit 22

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Logic

PLC Runtime

Pin 24

Pin 22

0 for bit 24
1 for bit 22

Read Pin 24

Write register

Read register

0

1
Introducing Pin Control Attack: A Memory Illusion

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

1

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

map via MMU

0 for bit 24
1 for bit 22

Write register

Read register

0

1

0

1

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

- Request for mapping the physical I/O Memory
- Map (I/O Memory, +16 bytes)

**PLC Runtime**

**Logic**

- Blink LED every 5 sec in Pin 22 if Pin 24 is True

- Pin 24 == Input (bit == 0)
- Pin 22 == Output (bit == 1)

- Write 0/1 every 5 sec
- Read Pin 24

**Operating System/Kernel**

- Virtual I/O Memory (mapped)
- State Register: 0 for bit 24, 1 for bit 22
- Write register
- Read register

**Physical I/O Memory**

- State Register: 0 for bit 24, 1 for bit 22
- Write register
- Read register

**Map via MMU**

**Physical I/O Memory**

- Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

State Register
0 for bit 24
1 for bit 22

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU

State Register
0

Pin 22 == Input (bit == 1)

PLC Runtime

Pin 24 == Output (bit == 0)

Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU

Pin 22 == Output (bit == 1)

Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec
Read Pin 24

Write register
Read register

0 for bit 24
1 for bit 22

map via MMU
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/Kernel

Virtual I/O Memory (mapped)

0 for bit 24
1 for bit 22

0/1

Write register

Read register

State Register

0 for bit 24
1 for bit 22

Physical I/O Memory

Write register

Read register

0 for bit 24
1 for bit 22

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

map via MMU

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec
Read Pin 24

Virtual I/O Memory (mapped)

0 for bit 24
1 for bit 22
0/1

State Register

Write register
Read register

Physical I/O Memory

State Register

Write register
Read register

0 for bit 24
1 for bit 22
0/1

map via MMU

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Write register
Read register

0/1

1

map via MMU

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion
Introducing Pin Control Attack: A Memory Illusion

Operating System/Kernel

PLC Runtime

Physical I/O Memory

State Register

Write register

Read register

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Current Use</th>
<th>Value</th>
<th>Default Use</th>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP004</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP004</td>
<td></td>
</tr>
<tr>
<td>GP007</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP007</td>
<td></td>
</tr>
<tr>
<td>GP011</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP011</td>
<td></td>
</tr>
<tr>
<td>GP020</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP020</td>
<td></td>
</tr>
<tr>
<td>GP022</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP022</td>
<td></td>
</tr>
<tr>
<td>GP023</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP023</td>
<td></td>
</tr>
<tr>
<td>GP024</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP024</td>
<td></td>
</tr>
<tr>
<td>GP025</td>
<td>Discontinuation of BYTE</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td>config of GP025</td>
<td></td>
</tr>
</tbody>
</table>
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

PLC Runtime

Physical I/O Memory

State Register

Write register

Read register

CPU

black hat EUROPE 2016
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Operating System/Kernel

map via MMU

Physical I/O Memory

State Register

Write register

Read register

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Current Value</th>
<th>Bit Value</th>
<th>Default Value</th>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP0104</td>
<td>Input</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0104</td>
</tr>
<tr>
<td>GP0107</td>
<td>Input</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0107</td>
</tr>
<tr>
<td>GP0109</td>
<td>Input</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0109</td>
</tr>
<tr>
<td>GP0102</td>
<td>Output</td>
<td>Output</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0102</td>
</tr>
<tr>
<td>GP0103</td>
<td>Input</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0103</td>
</tr>
<tr>
<td>GP0105</td>
<td>Input</td>
<td>not used</td>
<td>not used</td>
<td>not used</td>
<td></td>
<td>configuration of GP0105</td>
</tr>
</tbody>
</table>
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

State Register

Write register

Read register

map via MMU

Physical I/O Memory

State Register

Write register

Read register
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16bytes)

PLC Runtime

Operating System/Kernel

map via MMU

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
Write register
Read register

Logic
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

Operating System/Kernel

Map via MMU

PLC Runtime

Virtual I/O Memory (mapped)

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Physical I/O Memory

State Register
Write register
Read register

Write register
Virtual I/O Memory (mapped)

State Register
Read register

PLC Runtime

Blink LED

in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

Operating System/Kernel
map via MMU

PLC Runtime

Logic
Blink LED
every 5 sec
in Pin 22 if
Pin 24 is True

Virtual I/O Memory (mapped)

State Register
0 for bit 24
1 for bit 22

Write register

Read register

Physical I/O Memory

State Register
Write register
Read register

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Write register
Virtual I/O Memory (mapped)

0 for bit 24
1 for bit 22

Read register

Blink LED
Write register
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Operating System/ Kernel

map via MMU

Virtual I/O Memory (mapped)

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

State Register

0 for bit 24
1 for bit 22

Write register

Read register

Physical I/O Memory

State Register

Write register

Read register

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

PLC Runtime

Logic
Blink LED every 5 sec in Pin 22 if Pin 24 is True

Operating System/ Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register
0 for bit 24
1 for bit 22

Write register

Read register

map via MMU

0 for bit 24
1 for bit 22

Write register

Read register

Logic: Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Virtual I/O Memory (mapped)

PLC Runtime

Physical I/O Memory
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

map via MMU

PLC Runtime

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Read Pin 24

Read register

0 for bit 24
1 for bit 22

Write register

Virtual I/O Memory (mapped)

State Register

Physical I/O Memory

Pin 24 == True
Pin 22 == False

Read register

Write register

Blink LED

every 5 sec

in Pin 22 if Pin 24 is True

0 for bit 24
1 for bit 22

State Register

PLC Runtime

Operating System/Kernel

map via MMU

Physical I/O Memory
Introducing Pin Control Attack: A Memory Illusion

- Request for mapping the physical I/O Memory
- Map (I/O Memory, +16 bytes)
- Operating System/Kernel
  - Map via MMU

Logic
- Blink LED every 5 sec in Pin 22 if Pin 24 is True

PLC Runtime
- Pin 24 == Input (bit == 0)
- Pin 22 == Output (bit == 1)

Virtual I/O Memory (mapped)
- State Register
  - 0 for bit 24
  - 1 for bit 22
- Write register
- Read register

Physical I/O Memory
- State Register
  - 0 for bit 24
  - 1 for bit 22
- Write register
- Read register

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

PLC Runtime

Map (I/O Memory, +16 bytes)

Request for mapping the physical I/O Memory

Operating System/Kernel

map via MMU

Virtual I/O Memory (mapped)

0 for bit 24
1 for bit 22

State Register

Read register

Write register

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Logix

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

Read register
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Operating System/Kernel

map via MMU

PLC Runtime

Virtual I/O Memory (mapped)

Physical I/O Memory

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Read Pin 24

State Register

Write register

Read register

0 for bit 24
1 for bit 22

Write register

Read register

0 for bit 24
1 for bit 22

Blink LED every 5 sec in Pin 22 if Pin 24 is True

- PLC Runtime
- Operating System/Kernel
- Virtual I/O Memory (mapped)
- Physical I/O Memory
- Logic
- Blink LED every 5 sec in Pin 22 if Pin 24 is True

Map (I/O Memory, +16bytes)

State Register

Write register

Read register

0 for bit 24
1 for bit 22
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

PLC Runtime

Operating System/Kernel

map via MMU

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

1

Read register

Write register

Read Pin 24

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec

Pin 24	==	Input	(bit	==	0)
Pin 22	==	Output	(bit	==	1)
Write register

0 for bit 24
1 for bit 22

1

map via MMU

Logic

Blink LED
every 5 sec
in Pin 22 if
Pin 24 is True

Read Pin 24

Write 0/1 every 5 sec
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

map via MMU

PLC Runtime

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Virtual I/O Memory (mapped)

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

State Register

Read register

Write register

0 for bit 24
1 for bit 22

0/1

1

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

0/1

1

map via MMU

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Virtual I/O Memory (mapped)

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

State Register

Read register

Write register

0 for bit 24
1 for bit 22

0/1

1

Physical I/O Memory

State Register

Write register

Read register

0 for bit 24
1 for bit 22

0/1

1
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

Operating System/Kernel

map via MMU

PLC Runtime

Virtual I/O Memory (mapped)

Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)
Write 0/1 every 5 sec

Read Pin 24
Write register
Read register

0 for bit 24
1 for bit 22

0/1
1

State Register

Physical I/O Memory

State Register

Write register
Read register

0 for bit 24
1 for bit 22

0/1
1

map via MMU

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory
Map (I/O Memory, +16 bytes)

Operating System/ Kernel

Pin 24 = Input (bit = 0)
Pin 22 = Output (bit = 1)

Virtual I/O Memory (mapped)

State Register
0 for bit 24
1 for bit 22

Read register
Write register

0/1

PLC Runtime
Pin 24 == Input (bit == 0)
Pin 22 == Output (bit == 1)

Write 0/1 every 5 sec
Read Pin 24

Pin 24	==	Input	(bit	==	0)
Pin 22	==	Output	(bit	==	1)

Write register
0/1

PLC Runtime

Logic
Blink LED
every 5 sec
in Pin 22 if
Pin 24 is True

Physical I/O Memory

State Register
Write register
0/1

Read register

0 for bit 24
1 for bit 22

CPU

map via MMU

Logic

Write register
0/1

Input (bit == 0)
Output (bit == 1)

Virtial I/O Memory (mapped)

0 for bit 22
1 for bit 24

0/1
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Operating System/ Kernel

map via MMU

Virtual I/O Memory (mapped)

Physical I/O Memory

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 22 == Input (bit == 0)

Write 0/1 every 5 sec

Read Pin 24

State Register

Write register

Read register

0 for bit 24
1 for bit 22

0/1

1

State Register

Write register

Read register

0 for bit 24
1 for bit 22

0

1

Pin 22

Pin 24

Input

(bit == 0)

Blink LED

every 5 sec

in Pin 22 if
Pin 24 is True

Pin 22

GPIO

Enable of GPIO

Input

Output

Configuration of GPIO

Configuration of GPIO

Configuration of GPIO

Configuration of GPIO

Configuration of GPIO
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

Physical I/O Memory

State Register

Write register

Read register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 22 == Input (bit == 0)

Write 0/1 every 5 sec

Read Pin 24

0 for bit 22

0/1

1

0 for bit 24

1 for bit 22

State Register

Write register

Read register

0/1

1

pin 24

Enable

Blink

0

dOut 22

Fault
Introducing Pin Control Attack: A Memory Illusion

- Request for mapping the physical I/O Memory
  - Map (I/O Memory, +16 bytes)

- PLC Runtime

- Operating System/Kernel

- Virtual I/O Memory (mapped)
  - State Register
    - Write register: 0/1
    - Read register: 0/1
  - Pin 22 == Input (bit == 0)
  - Write 0/1 every 5 sec
  - Read Pin 24

- Physical I/O Memory
  - State Register
    - Write register: 0 for bit 22
    - Read register: 0/1
  - Blink LED every 5 sec in Pin 22 if Pin 24 is True
  - Pin 22 == Input (bit == 0)
  - Write 0/1 every 5 sec
  - Read Pin 24

- Logic
  - Blink LED every 5 sec in Pin 22 if Pin 24 is True
  - 0 for bit 22
  - 1
Introducing Pin Control Attack: A Memory Illusion

Request for mapping the physical I/O Memory

Map (I/O Memory, +16bytes)

PLC Runtime

Operating System/Kernel

Virtual I/O Memory (mapped)

map via MMU

Physical I/O Memory

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 22 == Input (bit == 0)

Write 0/1 every 5 sec

Read Pin 24

Write register

Read register

0 for bit 22

0/1

1

State Register

Write register

Read register

0 for bit 22

0/1

1

State Register

Virtual I/O Memory (mapped)

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True

Pin 22 == Input (bit == 0)

Write 0/1 every 5 sec

Read Pin 24

Write register

Read register

0 for bit 22

0/1

1

State Register

Write register

Read register

0 for bit 22

0/1

1

State Register

Physical I/O Memory

0/1
Introducing Pin Control Attack: A Memory Illusion

- Request for mapping the physical I/O Memory
- Map (I/O Memory, +16 bytes)

**Operating System/Kernel**
- PLC Runtime
- Virtual I/O Memory (mapped)
  - Write register to State Register
  - Read register from State Register

**Physical I/O Memory**
- Write register to 0 for bit 22
- Read register to 1

**Logic**
- Blink LED every 5 sec in Pin 22 if Pin 24 is True
- Logic: Blink LED if Pin 24 is True, write 0/1 to register
Introducing Pin Control Attack: A Memory Illusion

Operating System/ Kernel

Request for mapping the physical I/O Memory

Map (I/O Memory, +16 bytes)

PLC Runtime

Virtual I/O Memory (mapped)

Physical I/O Memory

Pin 22 == Input (bit == 0)

Write 0/1 every 5 sec

Read Pin 24

Write register

Logical

Blink LED every 5 sec in Pin 22 if Pin 24 is True

0 for bit 22

Write register

0/1

Read register

0 for bit 22

Write Failure!!

Pin is in Input Mode

Read register

1

State Register

Write register

State Register

Physical I/O Memory

Write Failure!!

Pin is in Input Mode

Write register

0/1

Read register

1

State Register

Logic

Blink LED every 5 sec in Pin 22 if Pin 24 is True
Problem statement

• What if we create an attack using pin control that:
  • Do not do function hooking
  • Do not modify executable contents of the PLC runtime.
  • Do not change the logic file

• Obviously we consider other defenses available (e.g. logic checksum is also there)
Pin Control Attack

• Pin Control Attack:
  • manipulate the I/O configuration (Pin Configuration Attack)
  • manipulate the I/O multiplexing (Pin Multiplexing Attack)

• PLC OS never knows about it.
Two variants of the work and bypassing the protections

• 1. a rootkit which manipulates the Pin Configuration.

• 2. a malicious C code which can do the same.
What do we need for a Pin Control Attack

• First variant (rootkit)
  • Root privilege
  • Knowledge of SoC registers
  • Knowledge of mapping between I/O pins and the logic

• Second variant:
  • Equal privilege as PLC runtime
  • Knowledge of mapping between I/O pins and the logic
How actually a PLC Controls its I/O (very basic)

- PLC Prepares I/O for the logic:
  - Map physical I/O base addresses
  - Enable Input/Output Mode of the register and use it (write or read it)
First variant (rootkit), precise I/O manipulation

- Exploiting the I/O configuration.
- Utilizing Processor Debug Registers.
Debug Registers

• Designed for debugging purpose.
• Function hooking intercept the function call and manipulate the function argument.
• We use debug registers in ARM processors to intercept memory access (No function interception, no function argument manipulation)
Devil is in detail...

• Combination of Pin configuration registers and ARM Processor Debug register
  • Put the mapped I/O address to the debug register.
  • Manipulate the Pin Configuration or multiplexing upon I/O memory access.
How Pin Configuration Attack Works?

<table>
<thead>
<tr>
<th>Manipulate Read</th>
<th>Manipulate Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Put I/O Address into Debug register</td>
<td>1. Put I/O Address into Debug register</td>
</tr>
<tr>
<td>read(I/O, Pin)</td>
<td>write(I/O, Pin)</td>
</tr>
<tr>
<td>2. Intercept Read Operation from I/O</td>
<td>2. Intercept Write Operation to I/O</td>
</tr>
<tr>
<td>3. Set Pin to Output Mode</td>
<td>3. Set Pin to Input (write-ignore)</td>
</tr>
<tr>
<td>4. Write Desired Value to Output</td>
<td>write() continue...</td>
</tr>
<tr>
<td>read() continue...</td>
<td>Pin Control Attack actions</td>
</tr>
<tr>
<td></td>
<td>PLC runtime actions</td>
</tr>
</tbody>
</table>
Wait a minute!

- Shouldn’t the PLC runtime fail or get terminated because of I/O failure?
  - Nope!
  - Remember no interrupt therefore, everything looks fine to the PLC runtime!
Implementation of the attack

• Before implementation we have to know how the PLC runtime interact with the I/O in physical world!
• We need to know more about I/O registers
• We actually need to write the driver for target I/O in our PLC.
Simple Logic

Let's test it with a simple Function Block Language Logic.

```plaintext
input : State of In.24
output: State of Out.22

Main Logic:
while True do
    read input;
    while input True do
        switch_state(output, five seconds);
        //states are High or Low.
    end
    if input False then
        hold the state of the output;
    else
        go to first while;
    end
end
```
Simple Logic 2

- Second Logic for a real PLC
Let's look at it.

Demo
Codesys Dynamic and Static Analysis

- I/O Mapping
- Look for Base Addresses of I/O

```
[b6e47f54] open("/etc/35.dat", 0_RDONLY) = 8 <0.001979>
[b6df334c] close(8) = 0 <0.001878>
[b6e47f54] open("/proc/cpuinfo", 0_RDONLY) = 8 <0.001354>
[b6df334c] close(8) = 0 <0.001878>
[b6f2c7e4] open("/dev/mem", 0_RDWR) = 8 <0.001182>
[b6e1998] mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_SHARED, 8, 0x2028]
[b6f2b0d] close(8) = 0 <0.001246>
```

[Diagram of code analysis with highlighted lines]
I/O Attack: Rootkit

• Rootkit needs root user to install its code as a Loadable Kernel Module (LKM).
• `vmalloc()` allocates our LKM. It bypasses Doppelganger.
• Do not do any kind of function hooking, bypasses Autoscopy Jr.
• Can change the logic regardless of logic operation.
I/O Performance of rootkit variant
CPU Overhead

Write Manipulation: ~ 5%

Read Manipulation: ~ 23%
Real-Time Features

• Priority Inheritance Mutex support
  • no priority inversion problem in the rootkit!

• Implementation without using any Kernel API with generic spin-locks.

• Using IRQF_NODELAY for ARM debug registers.

• Works in Linux RT, QNX, Lynx, VxWorks (with MMU) RTOSes.
Second Variant of the Attack – No Rootkit! No Root!

- No need to have rootkit!
- We can do the same with the PLC runtime privilege.
- Overhead below 1%.
- We can either remap the I/O or use already mapped I/O address.
Second variant

The Malicious Code

Starting Time Calculation Loop

- Read the I/O Input every 4 millisecond
- Read the I/O Output every 4 millisecond

Write Manipulate
Read Manipulate

Reconfigure the I/O Pins

- /dev/mem
- Exported Kernel Object File System
- device driver

Logic

PLC Runtime

Outputs

Inputs

Physical I/O Pin
Second Variant

<table>
<thead>
<tr>
<th>Manipulate Read</th>
<th>Manipulate Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Find the Reference Starting Time</td>
<td>1. Find the Reference Starting Time</td>
</tr>
<tr>
<td>3. Set Pin to Output Mode (write-enable)</td>
<td>3. Set Pin to Input (write-ignore)</td>
</tr>
<tr>
<td>4. Write Desired Value to Output Pin</td>
<td>write() to I/O</td>
</tr>
<tr>
<td>read(I/O, Pin)</td>
<td>3. Set Pin to Output (write-enable)</td>
</tr>
<tr>
<td></td>
<td>Write desired value</td>
</tr>
</tbody>
</table>

Pin Control Attack actions

PLC runtime actions
What about Analog Control?

- Analog signals are basically aggregation of digital signals.
- Two ways to do it:
  1. If part of or entire analog memory can get multiplexed to digital pins attacker can multiplex the pin and write digital bits and basically control the values in the analog memory
  2. Using the technique which we can PC+1, we tell the interrupt handler to return the control to the next instruction within the PLC runtime, basically avoiding write operation occur
Analog I/O Manipulation
Let's look at it.

Demo

Analog
Other Future Possibilities!

• Attacking pull-up and pull-down resistors in I/O interfaces
• What if we disable them?
• Remotely manipulate the I/O via a powerful electromagnetic field!
Discussions

• For now attacker can:
  • Simply change the logic
  • Modify PLC Runtime executable

• Fixing these attacks are trivial:
  • Proper Authentication
  • Proper Logic Checksum
  • PLC Runtime integrity verification

• Next Step for attackers:
  • Achieve its goal without actually modifying the Logic or Runtime or hooking functions
Conclusions

• Need to focus on system level security of control devices In future more sophisticated techniques come that evade defenses.
  • Pin Control attack is an example of such attacks.

• Pin Control Attack:
  • lack of interrupt for I/O configuration registers
  • Significant consequences on protected PLCs and other control devices such as IEDs.

• Solution:
  • It is hard to handle I/O interrupts with existing real-time constraints.
  • Monitoring I/O Configuration Pins for anomalies.
  • User/Kernel space separation for I/O memory.
Questions?

Everything that has a beginning has an end.

The Matrix Revolutions.

Contact:

a.abbasi@utwente.nl
mhashemi@quarkslab.com