A Diet of Poisoned Fruit: Designing Implants & OT Payloads for ICS Embedded Devices

Jos Wetzels, Marina Krotofil
Marina Krotofil
@marmusha

• Senior Security Engineer

• Specializing on offensive security of Critical Infrastructures

• **Focus:** Physical Damage or how to make somethings go bad, crash or blow up by means of cyber-attacks
Jos Wetzels  
@s4mvartaka

- Principal Consultant & Security Researcher

- **Focus**: Embedded Systems Security (ICS, Automotive, IoT, ...)

- (previously) Security Researcher @ University of Twente on protection of critical infrastructure
AGENDA

1. Introduction
2. Cyber-Physical Attack Lifecycle
3. Implants
4. OT Payloads
5. Conclusion
Here is a Plant. What is Your Plan?
Two Common View on Cyber-Physical Attacks

• “Trivial! Look at the state of ICS security!”

• “Borderline impossible! These processes are extremely complex & engineered for safety!”

Secura
Typical Expectation: MAGIC BUTTON

(does not exist!)
Attacks with Strategic and Long Lasting Effect

- Attacks with strategic, lasting damage will be process specific & require good process comprehension
- Will require attacker to develop detailed ‘damage scenario’
  - What causes a pipeline to explode?
  - What causes the right pipeline to explode?
  - What causes the right pipeline to explode at the right moment?
IT Security vs. OT Security

ICS security

IT security
(cyber-security -> taking over the infrastructure)

OT security
(causing impact on the operations -> process and equipment)

Attack payload
Marina & Jos
Industrial Plants Work on Control Loop Concept

Process Optimization Applications

HMI

Control system

SET POINT

Sensors

Actuators

© 2019

https://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Leitstand_2.jpg/327px-Leitstand_2.jpg
Industrial Network Architecture

Definition of Real Time

Planning and management

Optimization Applications

HMI (Supervisory control)

Controllers (Regulatory control)

Field Instrumentation
Physical Process and Control Equipment
Physical Process and Control Equipment
Security vs. Safety

Threats

Layers of security protections

Hazards

Layers of safety protections

Incident

Time
Hazards and Layers of Safety Protections

DEFENSE IN DEPTH

Mitigation

Emergency response

Passive protection (e.g. Bund/Dike)

Active protection (e.g. Relief valve/Rupture disk)

Incident

Safety Instrumented System

Trip

Operator Intervention

Alarm

Process Control

Loop

Process value

Process Design


© 2019

Designing Cyber-Physical Payload

Evil Motivation

Cyber-physical Payload

https://cdn5.vectorstock.com/i/1000x1000/32/14/skull-and-crossbones-with-binary-code-vector-20603214.jpg
AGENDA

1. Introduction
2. Cyber-Physical Attack Lifecycle
3. Implants
4. OT Payloads
5. Conclusion
Cyber-Physical Attack Development Lifecycle

• If you know how attackers work, you can figure out how to stop them
• Attack lifecycle is a common method to describe a process of conducting cyber attacks
Cyber-Physical Attack Development Lifecycle

- Access
- Discovery
- Control
- Damage
- Cleanup

Obtaining Feedback
Preventing Response

© 2019 Secura
How Does This Fit into Other Attack Frameworks?
Lockheed Martin, the Cyber Kill Chain®

1. Reconnaissance
   - Harvesting email addresses, conference information, etc.

2. Weaponization
   - Coupling exploit with backdoor into deliverable payload

3. Delivery
   - Delivering weaponized bundle to the victim via email, web, USB, etc.

4. Exploitation
   - Exploiting a vulnerability to execute code on victim's system

5. Installation
   - Installing malware on the victim

6. Command & Control (C2)
   - Command channel for remote manipulation of victim

7. Actions on Objectives
   - With 'Hands on Keyboard' access and 'on-demand' intruder, accomplish their objectives

You are here

SANS Industrial Control System Cyber Kill Chain

Stage 1 mimics a targeted and structured attack campaign.

Stage 2 shows the steps associated with a material attack that requires high confidence.

https://www.sans.org/reading-room/whitepapers/ICS/paper/36297
<table>
<thead>
<tr>
<th>Persistence</th>
<th>Privilege Evasion</th>
<th>Defense Evasion</th>
<th>Operator Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>Execution</th>
<th>Command and Control</th>
<th>Disruption</th>
<th>Destruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid Accounts</td>
<td>Rootkit</td>
<td>Network Sniffing</td>
<td>Exploitation of Vulnerability</td>
<td>Connection Proxy</td>
<td>Module Firmware</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module Firmware</td>
<td>Exploitation of Vulnerability</td>
<td>File Deletion</td>
<td>Block Serial Comm Port</td>
<td>Brute Force</td>
<td>Device Information</td>
<td>Default Credentials</td>
<td>Scripting</td>
<td>Commonly Used Port</td>
<td>Module Firmware</td>
<td></td>
</tr>
<tr>
<td>Modify Control Logic</td>
<td>Alternate Modes of Operation</td>
<td>Modify Reporting Settings</td>
<td>Exploitation of Vulnerability</td>
<td>Role Identification</td>
<td>External Remote Service</td>
<td>Command Line Interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify System Settings</td>
<td>Masquerading</td>
<td>Modify Reporting Message</td>
<td>Credential Dumping</td>
<td>Location Identification</td>
<td>Modify Control Logic</td>
<td>Modify System Settings</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory Residence</td>
<td>Modify System Settings</td>
<td>Block Reporting Message</td>
<td>Network Connection Enumeration</td>
<td>Man in the Middle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Firmware</td>
<td>Spoofer Reporting Message</td>
<td>Serial Connection Enumeration</td>
<td>Alternate Modes of Operation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Tag</td>
<td>I/O Module Enumeration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Control Logic</td>
<td>Remote System Discovery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Physical Device Display</td>
<td>Network Service Scanning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify HMI/Historian Reporting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Modify Parameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We don’t know where we are in this model just yet :-)
Overview of Stages
Access

• Target facility
  • Discovery
  • Access to needed assets
  • Attack execution

• Trusted 3\textsuperscript{rd} party (staging target)
  • Access to target facility
  • Access to needed assets
  • Process comprehension

• Non-targeted/Opportunistic
Targeting

• There are few known cases of strategic targeting
• Target might be also selected as best suitable certain criteria
• Collateral victim
• Opportunistic
Venezuela, 2019

• Suspected cyber-attack on Guri hydroelectric power plant
• Produces 80% of country’s electricity
• Details of plant’s upgrade are publicly available, including possible remote access
Venezuela, 2019

ABB's 800 kV substations strengthen Venezuelan power grid

2007-10-16 - ABB has added another impressive customer reference to its all-round capability in bulk power transmission - two 800 kilovolt (kV) turnkey substations that will strengthen and enable Venezuela's power grid in order to supply an ever-increasing level of power to the Francis turbines in the Guri hydroelectric power plant.

In May 2009, Alstom was awarded a contract for the refurbishment of four powerful turbines. ABB is now supplying the necessary equipment.

ABB supplies critical systems for giant power plant

2007-03-12 - ABB is upgrading the 20 generating units of the 10,000 megawatt Guri hydropower plant in Venezuela – the second largest hydro-electric plant on earth – with new control, protection and instrumentation systems.
Ukraine, 2016

• INDUSTROYER malware was deployed to shutdown electricity distribution at Pivnichna substation
• There is no strong indications that victim substation was strategic target
• Details of substation upgrade are publicly available
Ukraine, 2016

INDUSTROYER malware work was used to shutdown electricity distribution at Pivnichna substation.

There is no strong indication that the victim substation was a strategic target.

Details of the substation upgrade are publicly available:


Targeted by malware
Saudi Arabia, 2017

- TRITON malware targeted Safety Instrumented Systems at petrochemical plant
- There is no strong indication that TRITON victim was strategic target
- Affected site could have been used as live drill and testing platform before attacking strategic target

16.02.2003 · Triconex, a supplier of products, **systems** and services for safety, has received contracts from Jubail United Petrochemical (JUPC) of **Saudi Arabia**, to provide critical safety and turbomachinery control.

**NEWS**

*Invensys wins Qatar, Iraq contracts*

July 2006

Invensys has won two major contracts in the Middle East, one to supply steam turbine control systems for a Qatar LNG project and the other for the supply of Foxboro and Eurotherm control equipment for use in Iraqi oilfields.

*The contract for Qatar involves the supply of four Triconex centrifugal pump steam turbine speed and overspeed control systems for use on the world's largest liquefied natural gas (LNG) project.*

Known as Qatargas II, this 9.5 billion euro project involves expanding the LNG liquefaction plant at the Ras Laffan Industrial City in Qatar. The project will further develop the large gas reserves in the country's North Field. These are estimated to be in the region of 200 trillion cubic feet, with projected production of 9.5 billion cubic feet per day.

**Saudi Aramco Southern Area Gas Oil Separation Plant Control System Upgrade Project**

*process control systems, each consisting of a DCS (CENTUM CS 3000), emergency shutdown system (Triconex), vibration monitoring system (Bently Nevada), and field instrumentation.*

Role of OSINT in Targeting

- The Internet is full of proprietary and confidential industrial documentation.
- Discovering helpful information about certain industrial facility may provoke targeting.
Role of OSINT in Targeting

The Internet is full of proprietary and confidential industrial documentation. Discovering helpful information about certain industrial facilities may provoke targeting.
Targeting 3rd parties (supply chain)

• Getting access to into target facilities
• Getting access to needed assets/equipment,
  – E.g. through maintenance support contracts
• Obtaining information related to target or potential victims
  – Engineering/networking/config documentation
  – User application (control logic), etc.
National Advisories on the Threat

Alert (TA18-074A)
Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors

Original release date: March 15, 2018 | Last revised: March 16, 2018

This campaign comprises two distinct categories of victims: staging and intended targets. The initial victims are peripheral organizations such as trusted third-party suppliers with less secure networks, referred to as “staging targets” throughout this alert. The threat actors used the staging targets’ networks as pivot points and malware repositories when targeting their final intended victims. NCCIC and FBI judge the ultimate objective of the actors is to compromise organizational networks, also referred to as the “intended target.”

https://www.us-cert.gov/ncas/alerts/TA18-074A

Advisory: Hostile state actors compromising UK organisations with focus on engineering and industrial control companies

The NCSC is aware of an ongoing attack campaign against multiple companies involved in the CNI supply chain. These attacks have been ongoing since at least March 2017. The targeting is focused on

The NCSC is aware of an ongoing attack campaign against multiple companies involved in the CNI supply chain. These attacks have been ongoing since at least March 2017. The targeting is focused on engineering and industrial control companies.
Data Exposure is Penalizable in Regulated Facilities

• NERC CIP-003-3 standard
• Sensitive utility’s network infrastructure data were exposed via server of third-party service provider

DATA EXPOSURE BY VENDOR LEADS TO $2.7 MILLION NERC PENALTY FOR UTILITY
March 09, 2018

A seven-figure penalty reported by the North American Electric Reliability Corporation demonstrates the potentially severe consequences for electric utilities related to improper data handling practices and underscores the challenges in preventing and resolving unauthorized disclosures.

A public filing by the North American Electric Reliability Corporation (NERC) on February 28 reported that an unidentified electric utility agreed to pay a $2.7 million penalty to resolve violations of the Critical Infrastructure Protection (CIP) reliability standards related to the exposure of sensitive data. While settlement agreements...
Role of Access Stage

• Access stage largely defines the selection of damage scenario
  • Access driven
    – E.g., obtained access to specific equipment via 3\textsuperscript{rd} party remote maintenance contract
    – Did not manage to access Safety Systems
  • Information driven
    – E.g., obtained specific information about unhealthy state or repairs of equipment
Discovery

• Network reconnaissance
  • Majority of this stage is similar to traditional IT recon process/attack life cycle, tools may differ
  • Information enumeration

• Process comprehension
  • Understanding exactly what the process is doing, how it is built, configured and programmed
Discovery

• Network reconnaissance
  • Majority of this stage is similar to traditional IT recon process/attack life cycle, tools may differ
• Information enumeration

Process comprehension

On the Significance of Process Comprehension for Conducting Targeted ICS Attacks

Benjamin Green
Lancaster University
Lancaster, United Kingdom
b.green2@lancaster.ac.uk

Marina Krotofil
Hamburg University of Technology
Hamburg, Germany
marina.krotofil@tuhh.de

Ali Abbasi
University of Twente
Enschede, Netherlands
a.abbasi@utwente.nl

http://eprints.lancs.ac.uk/88089/1/sample_sigconf.pdf
Control

• Least understood and studied stage among all
• It is about discovering:
  • Dynamic model of the process and its limits
  • Ability to control process
  • Attack effect propagation
  • Active stage in live environment
Case Study: Water Treatment Plant
Use Case: Killing UF Filter in Water Treatment Facility

Acknowledgement: Sridhar Adepu and Prof. Aditya Mathur, SUTD, Singapore for conducting an experiment for this talk

https://itrust.sutd.edu.sg/testbeds/secure-water-treatment-swat/
Use Case: Killing UF Filter in Water Treatment Facility

• Water treatment process consists of multiple stages, including several stages of filtering
  • Water filters are expensive
  • When broken, water supply is interrupted

https://en.wikipedia.org/wiki/Ultrafiltration
https://en.wikipedia.org/wiki/Reverse_osmosis
UF Filtering: HMI Screen
UF Filtering: PI&D Diagram
UF Backwash: HMI and PI&D Diagram
How Do We Pull This off?

• There are tree conditions which can trigger backwash process, each guided by a state machine
  • Preset timer (every 30 minutes)
  • UF filter differential pressure (DP) ≥ 40 kPa
  • Plant shutdown
How Do We Pull This off?

- There are three conditions which can trigger the backwash process, each guided by a state machine:
  - Preset timer (every 30 minutes)
  - UF filter differential pressure (DP) ≥ 40 kPa
  - Plant shutdown
How Do We Pull This off?

- There are three conditions which can trigger the backwash process, each guided by a state machine.
- Preset timer (every 30 minutes)
- UF filter differential pressure (DP) ≥ 40 kPa
- Plant shutdown
One Possible Attack Execution Scenario

Stage 6

1.1 MV303 OPEN

1.2 P602 ON

Pressure in UF membrane will increase. But HOW MUCH?

Stage 3

ON

Tank T301

Pump P301

Stage 4

UF is active

Valve MV303

Pump P602

UF

LIT301

© 2019 Secura
Control Stage of Process Comprehension

• Average UF filter DP is ≈ 12-13 kPa
• Max DP is 98 kPa, reached in 8 sec
• Process recovery (return to normal) is 5 sec
• Note, this data still does not tell us whether this pressure kills the UF filter and how quickly
Control Stage of Process Comprehension

- Average UF filter DP is ≈ 12-13 kPa
- Max DP is 98 kPa, reached in 8 sec
- Process recovery (return to normal) is 5 sec
- Note, this data still does not tell us whether this pressure kills the UF filter and how quickly
Damage

• Requires subject-matter knowledge (engineering)
• Cant take several forms
  • Explosions (of course!)
  • Equipment breakage
  • Pollution
  • Product Out of Specification
  • Increased production costs, etc.
Attack Design != Implementation Success

Attckr

FIT401
Spoofed value

1.1 FIT401
Spoofing to 0.4

1.1 UV401
OFF

ORP meter
AIT502

Attckr

2.1 AIT502
Spoof to Low

Reverse Osmosis filtering

Stage 4

Stage 5
Cleanup

• In traditional hacking it is possible to execute the entire attack without being ever detected
  • In process control it is not an option because of physical effect

• Create forensic footprint of what the investigators should identify as cause of the incident/accident
  – E.g. time attack to process troubleshooting
Why Implant?
Implant

“Hardware or software modification designed to gain unauthorized control over specific system functionality.”
OT Payload

“Digital implementation of (part of) a cyber-physical attack”
Why Implant

• Why not just modify control logic / change setpoints / send malicious command?

• For more complicated attacks
  • Coordination, Feedback, Speed, Low-level functionality access

• Many scenarios possible without implants
  • Eg. Ukraine 2015 & 2016
Where to Implant?
Where to Implant?
Network Equipment

Manipulating OT traffic

Observing & learning OT traffic

Dropping traffic to cause loss of control / view by suppressing alarm or signal
Process & Safety Controllers

Suppress condition monitoring alerts

Prevent Safety Response

Measure attack progress

Manipulate IO
Field Devices

Spoofing sensor data at high speed

Overriding digital safety mechanisms
How to Implant?
We want *smooth* native code execution

• Need access to low-level, privileged functionality
  • Memory-/Port-Mapped IO (MMIO/PMIO)
  • Kernel memory objects
  • Logic runtime memory
  • Persistence mechanisms

• Ideally via silent hot-patching
  • No reboots, no service restarts, *no process upsets*
Implant Delivery Vectors
PLC 101 - Architecture

Standalone

Modular
Power Supply, CPU, I/O, Comms, ...
PLC 101 - Backplane

Inter-Module Databus
Multibus, P-Bus, VMEbus, X-Bus, STD-32, PCIe, …
PLC 101 – CPU Module Internals
PLC 101 – Boot Sequence

Diagram of boot sequence:
- Flash
  - Bootloader
  - Firmware
  - Logic Program(s)
- Processor
- Boot ROM
- RAM
- Bootloader
- RTOS / Executive
- Tasks
  - Runtime
  - Httpd
  - ...

Software logos:
- ThreadX
- Real-Time Linux
- QNX
- VxWorks
- CODESYS
- ProConOS
PLC 101 – Logic Program Execution
PLC 101 - Scan Cycle

Scan Cycle

1. Read Inputs
2. Execute Program
3. Diagnostics / Comms
4. Write Outputs

Input Image Table

Input Buffer

Input Images

Discrete Inputs

Analog Inputs

Output Image Table

Output Buffer

Output Images

Discrete Outputs

Analog Outputs
Implant Installation

- Escalate Privileges*
- Disable Diagnostics
- Relocate Implant
- Ensure Persistence*
- Set Hooks
- Go Resident

Implant stability

Eg. modify firmware or stored logic in flash

* Optional
# Implant Design Considerations

<table>
<thead>
<tr>
<th>Active Implant</th>
<th>Dormant Implant</th>
<th>Persistence</th>
<th>Memory Residence</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Includes OT payload</td>
<td>• OT payload delivered later</td>
<td>• Complicated by code signing</td>
<td>• No reboot survival</td>
</tr>
<tr>
<td>• Limits detection / network</td>
<td>• Limits forensics exposure</td>
<td>• Need ability write to flash</td>
<td>• Limits forensics exposure</td>
</tr>
<tr>
<td>forensics exposure</td>
<td></td>
<td>&amp; enough space</td>
<td></td>
</tr>
</tbody>
</table>
We want scalability

- Target different vendors’ systems with similar implant functionality

- But limited number of players out there
  - Eg. construct arsenal of generic templates for key DCS & safety controllers

- One-time upfront investment, no huge turnover
Complication: Heterogeneity

Processor

OS

Runtime

IO

Interaction

Memory

Organization

Security

Features
Complication: In-House vs Commercial

Proprietary SoC / ASIC*

Proprietary OS / Executive

Proprietary Runtime

Example: Triconex SIS

- In-House OS + Runtime, different processors & OS variants between versions of same product

Triconex MP 9 (3006)

Triconex MP 10 (3008)

Triconex MP 11 (3009)
Counter-Example: Rise of Commercial RTOSes & Runtimes
Complication: Resource Constraints

- MPC860, 50 MHz
- 6 MB Flash
- 16 MB DRAM
- 32 KB SRAM

You better enjoy programming...

- ARM9, 14 MHz
- 512 KB Boot Flash
- 8 MB RW Flash
- 2 MB SRAM

Will need to fit implant in there
- Signals processing? Malicious logic? Comms?
  Often stretched by normal functionality already
Complication: Security Engineering

- Domain & Privilege Separation
- Firmware & Logic Signing
- Sandboxing
- Exploit Mitigations
- Programming Key-locks

Case Study: TRITON
TRITON / Trisis / HatMan (2017)
TRITON Attack Overview

IT

DMZ

Process Control Network

SIS Network

Physical Process

Sensors and Actuators

Engineering Station

SIS Controllers
TRITON injects ‘dormant’ implant into Triconex controller memory

"Your wish is my command"

Eng. Workstation

**trilog.exe**
- script_test.py
- library.zip
- inject.bin
- imain.bin

TriStation Engineering Protocol

Logic Download
(compiled for PPC, executed on CPU)

"Execute my shellcode please"
Why not just modify firmware?

Firmware Download
(FC 0x50: unauthenticated, unsigned)

Controller reboots into download mode,
logic execution interrupted!

Logic Append
(FC 0x01: unauthenticated, unsigned)

New logic appended to circular linked program list, logic continues running!
Implant Installation

• Safety program executed in *user* mode

• Need *supervisor* to flush icache & apply mods

• Privilege level set in PPC MSR register, NW for *user*

Requires Supervisor Privileges

* ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)
Stage 2: Privilege Escalation

- Exploit syscall 0x13 (SOE Status) to modify MSR while in *supervisor* mode, set saved MSR bit

- No memory permissions, can write anywhere in *user* mode, *including kernel globals*. Exploit write-what-where.
Stage 2: Disable RAM Check

```
    bge    loc_57EC
    lwz    r4, (dword_1D0890 - 0x1D0890)(r30)
    li     r5, 0x100
    bl     sub_611DC
    cmplwi r3, 0
    b      jump_over

    lwz    r4, 0(r29)
    lis    r3, aRamRomMismatch@ha
    lwz    r5, 0(r30)
    addi   r3, r3, aRamRomMismatch@l
    crclr  4*cr1+eq
    bl     sub_567BC
    li     r31, -1
```

- Originally conditional branch
- `# "Ram Rom Mismatch Rom(\%x) Ram(\%x)\r\n"
- `# CODE XREF: sub_5750+701j`

---

Escalate Privileges*
- Enable Diagnostics
- Relocate Implant
- Ensure Persistence*
- Set Hooks
- Go Resident

---

© 2019 Secura

* ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)
Stage 2: Relocate Implant

copy_payload_into_fw:
  mtctr  r5
  addi  r4, r4, -1
  addi  r3, r3, -1

loc_7B8:
  lbzu  r5, 1(r4)
  stbu  r5, 1(r3)
  bdnz  loc_7B8
  blr

Ensures Residence Even with full logic wipe

* ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)
Stage 2: Modify Network Command Handler

• Entry 0x1D (Get MP Status)
• Allows for network comms

```
li    r0, 0xCC   # Load Immediate
stw   r0, 0(r27) # Store Word
bl    patch_jump_table_entry # Branch
stw   r25, 0(r3) # Store Word
bl    patch_ram_check # Branch
li    r4, 0x4800 # Load Immediate
sth   r4, 0(r3)  # Store Half Word
bl    flush_instruction_cache # Branch
```

```c
default_handler, imain_bin_start_reloc, default_handler, libc8, loc_39C88, loc_39CE8, loc_39E38, loc_39D78, loc_39D78,
```

* ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)
Stage 3: Implant

- Execute
  - 249: Packet looks correct?
  - Address in firmware area?
  - Branch to address
    - Resp. length = 0x0

- Read Memory
  - 23: Packet looks correct?
  - Write Memory
    - 65: Packet looks correct?
      - Build Response
        - Copy response/length
        - Hand packet back
        - Branch to original code

- Subcommand value?
  - MP value matches?
    - No

* ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)
Stage 4: OT Payload

• Once implant is injected we have dormant ‘god mode’
  • Arbitrary *supervisor* RWX over network

• Deliver OT payload at later moment

• Not recovered from incident, but we can speculate …
AGENDA

1. Introduction
2. Cyber-Physical Attack Lifecycle
3. Implants
4. OT Payloads
5. Conclusion
Damage Stage

1. Manipulate the process
   - Direct Manipulation of actuators
   - Indirect Deceive controller/operator about process state (e.g., spoof sensor)

2. Prevent response
   - Operators Blind
     - Mislead
   - Control / Safety System
     - Modify operational/safety limits
     - Blind about process state

3. Obtain Feedback
   - Direct or Derived (e.g., via proxy sensors/calculations)
I/O Manipulation
I/O Manipulation

• Simple concept, non-trivial execution

• Many different approaches
  • Depends on how IO image tables are populated, how IO is wired to chip executing logic
  • Different technical ways to achieve same goal
I/O Manipulation

- Memory Breakpoint
- Patch Instructions
- Change Memory Permissions

* Ghost in the PLC – Ali Abbasi & Majid Hashemi, BlackHat EU 2016
I/O Manipulation

Hook accesses to IO portion of shared memory

Hook bus handler routines

Figure 3 Architecture of a Model 3008 Main Processor

© 2019

* https://www.nrc.gov/docs/ML0932/ML093290420.pdf
* www.amikonplc.com/
  www.kenosha-reuse.com
Complication: Field Device Limitations

• Cyber limitations might be placed on theoretically feasible functionality for protective reasons*
  • Valve closing speed
  • Non-digitally alterable VFD skip frequencies

• Prevents IO manipulation from achieving desired result
  • Overcoming this requires implanting field device
  • Patch out limitations / sanity checks

* Similar problem in automotive where certain diagnostic messages are disallowed above certain speeds
Alarm Suppression
Alarm Suppression

• Again: simple concept, non-trivial execution
  • We want to prevent an outgoing alarm being raised or incoming alarm being acted upon

• Might require very different approaches
  • Alarm raised with dedicated protocol message
  • Alarm signal via IO
  • Alarm bit in flag accompanying read PV
Alarm Propagation

Goal: catalyst deactivation

Safety shutdown
Hiding Alarms
Suppressing Alarms
Example: Simple water tank level alarm

Safety program resides in memory as code, modify to set alarm to fixed false
Finding Instructions to Patch

```assembly
li    r28, 0
stw   r28, -4(r2)
lis   r27, _water_high@ha
lwz   r28, _water_high@l(r27)
clrlwi r28, r28, 31  # r28 := water_high
lis   r26, _water_low
lwz   r27, _water_low(r26)
clrlwi r27, r27, 31  # r27 := water_low
or    r26, r27, r28  # r26 := water_high OR water_low
addi  r27, r2, -4
lwz   r28, 0(r27)
insrwi r28, r26, 1,31
stw   r28, 0(r27)
lwz   r28, -4(r2)
clrlwi r28, r28, 31
lis   r26, _alarm
mr    r26, r26
lwz   r27, 0(r26)
insrwi r27, r28, 1,31
stw   r27, 0(r26)
```
Hot-Patching Safety Program

\begin{verbatim}
li r28, 0
stw r28, -4(r2)
lis r27, _water_high@ha
lwz r28, _water_high@l(r27)
cr1lw r28, r28, 31  # r28 := water_high
lis r26, _water_low
lwz r27, _water_low(r26)
cr1lw r27, r27, 31  # r27 := water_low
\textcolor{red}{li r26, 0}  # alarm := FALSE
addi r27, r2, -4
lwz r28, 0(r27)
inrwi r28, r26, 1,31
stw r28, 0(r27)
lwz r28, -4(r2)
cr1lw r28, r28, 31
lis r26, _alarm
mr r26, r26
lwz r27, 0(r26)
inrwi r27, r28, 1,31
stw r27, 0(r26)
\end{verbatim}
Alarm Suppression
Alarm Relaxation & Tightening
Why relax or tighten instead of suppress?

• Don’t prevent alarm from being raised but change conditions
  • Limits, deadband, priority

• Relax: Stealth during scheduled testing

• Tighten: Cause hard-to-resolve alarm storms
Hook functionality that decides whether to raise alarm

- Can be data (limit, priority, deadband): overwrite in RAM
  - Make sure to spoof values when queried!

- Or code (alarm logic): patch instructions
  
  ```assembly
  STR    R3, [SP,#0x60+var_40]
  ADD    R5, SP, #0x60+var_28
  MOV    R3, #0
  STR    R3, [R5,#-4]!
  MOV    R0, #0x18
  LDR    R1, =aRtalarmlistatt ; "RtAlarmListAttribute.cpp"
  LDR    R2, =0x19B
  ADD    R3, R3, #2
  BL     init_object
  ```
Implant Communication
Implants need to synchronize

1. Process state A
2. Change air / medium inflow
3. Process state B
4. Change agitator speed

Pressure, temperature, pH, moisture, ...

© 2019

https://en.wikipedia.org/wiki/Bioreactor
These can be in completely different parts of the process, on different networks

Might not see much electronic chatter after implanting
Process state change detection

Non-Parametric Cumulative Sum (NCUSUM)

\[
S_i^+ = \max(0, |X_{i-1} - X_i| + S_{i-1}^+ ) \\
S_i^- = \max(0, |X_i - X_{i-1}| + S_{i-1}^- )
\]

17640 bytes \(\approx 0.11\%\) of DRAM (unoptimized)

* CPS: Driving Cyber-Physical Systems to Unsafe Operating Conditions by Timing DoS Attacks on Sensor Signals – M. Krotofil et al.

© 2019

AGENDA

1. Introduction
2. Cyber-Physical Attack Lifecycle
3. Implants
4. OT Payloads
5. Conclusion
Conclusion

Marina

- Damage Scenario Design
- OT Payload Design
- Implant Design
- Exploit Implementation

Jos

- Attack Integration & Testing
- OT Payload Integration & Testing
- Implant Integration & Testing
- OT Payload Implementation
Appreciation

• Sridhar Adepu & Prof. Aditya Mathur

• Jason Larsen