DRAGOS

Analyzing PIPEDREAM:
Challenges in testing an ICS attack toolkit

Speaker

Jimmy Wylie Principal Malware Analyst II, @mayahustle

Agenda

- Background
- PIPEDREAM Capabilities
- Hardware & Lab Setup
- Assessing the Impact
- Results from Lab Testing
- Real World Usage
- Conclusion

It’s a Group Effort

Carolyn Ahlers
Logan Carpenter
Sam Hanson
Reid Wightman
Kate Vajda
Kyle O’Meara
Casey Brooks
Conor McLaren
Kevin Woolf
Maritza Dubec
Cathy Clarke
Matt Pahl
Austin Scott
Raahul Mareddy
Michael Logoyda
Jess O’Bryan
Brian Warehime
Sergio Caltagirone
Thomas Winston
Gus Serino
John Burns
Gregory Pollman
Julian Gutmanis
Rob Lee
Marissa Costa
Grant Freter
Andrue Coombes
Camille Stauffer
Monserrat Thomason
Danielle Gauthier
Avril Adams
Megan Pingatore
Mark Urban
Peter Vescuso
Gloria Cedillo
Mike Hoffman

Where’d you find it?

Required Legalese

Dragos identified and analyzed PIPEDREAM’s capabilities through our normal business, independent research, and collaboration with various partners in early 2022.

PIPEDREAM Components

EVILSCHOLAR

Designed to discover, access, manipulate, and disable Schneider Electric PLCs. Can target additional hardware through CODESYS library.

BADOMEN

Designed to scan, identify, interact, and manipulate Omron PLCs

MOUSEHOLE

Tool for interacting with OPC UA servers. Designed to read and write node attribute data, enumerate the Server Namespace and associated NodeIds, and brute force credentials.

Windows Components

DUSTTUNNEL

Remote operational implant to perform host reconnaissance and command-and-control.

LAZYCARGO

User-mode Windows executable that drops and exploits a vulnerable ASRock driver to load an unsigned driver.

Lab Setup & Acquisition

Impacted Devices, Abused Protocols & Vulns

OMRON

NX1P2 Compact Machine Controller
NX-SL3300 Safety Controller
NJ501-1300 Automation Controller
NX-ECC EtherCAT Coupler
NX-EIC202 Ethernet/IP Coupler
NX-ECC203 EtherCAT Coupler
S8VK Power Supply
1S-series Servo Drives

Schneider Electric

TM251 PLC
TM241 PLC
TM221 PLC
TM258 PLC
TM238 PLC
LMC058 Motion Controller
LMC078 Motion Controller

ICS Protocols

CODESYS
Schneider Discovery (NetManage)
Modbus
Omron FINS
OPC UA

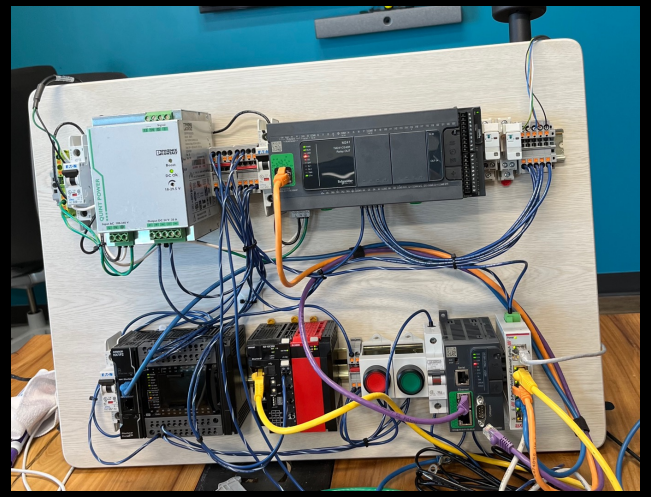
Vulnerabilities, Exposures, and Susceptibilities

CVE-2020-15368 – LAZYCARGO utilizes this CVE to load an unsigned driver.

Undisclosed Vulnerabilities in Schneider Electric.

CVE-2022-34151 - Hardcoded Creds in Omron devices.

Lab Set-up



Omron Devices

NX1P2 Controller

NX-EIC202 Ethernet IP Coupler

NX-ECC201 EtherCat Coupler

SL3300 Safety Controller

R88D-1SN10F-ECT Servo Drive

R88D-1SN01L-ECT Servo Drive

R88M-1M10030S-S2 AC Servo Motor

Schneider Electric Devices

Modicon TM241 PLC (4.0.X firmware)

Modicon TM251 PLC (5.0 firmware)

OPC UA (not shown)

Kepware OPC UA server

Various Rockwell devices controlling a small pipeline process.

Other hardware

Phoenix Contact power supply

Sixnet Industrial ethernet switch

Stack light, buttons

Device Acquisition is a Pain

TM251 running 5.0, need 4.0.X

- Downgrading == bricked PLC ☹

Ask the vendor

- Company attitude
- CYA, NDAs, reporting deadlines

<u>eBay or ask a friend</u>

Easy to purchase,
stay independent

Assessing Impact & Approach

Analysis Approach

Looking at 5 malware samples in parallel

- Focus on PLCs and malware RE.
- Test on representative devices.

Analysis Process & Goals

1. Static analysis while waiting for hardware.
2. Release of initial details and mitigation advice.
3. Runtime analysis with PCAP collection.
4. Which techniques work, and on what device.
5. Release updated details and specific mitigation advice.
6. Hypothesize and test what next versions could look like.

Analysis Questions

IMPACT OF CURRENT CAPABILITY	• Does it work? • Is the CODESYS implementation specific to Schneider or applicable to all CODESYSv3 devices? • Does BADOMEN's Servo module work on other servos? • More generally: can these tools manipulate PLC logic?
ASSESSMENT OF FUTURE CAPABILITY	• How easy is it to automate an attack? • What functionality could be added with minimal research? What will it look like in 6 months?
VICTIMOLOGY	• Can we narrow down a likely target?

Lab Testing Results

Multiplatform toolkit to interact with OPC UA servers.

FORMAT:
Python framework

TARGETS:
OPC-UA servers

- Scan for OPC UA Servers on a local network (default: TCP/4840).
  - Port can be changed and can scan for OPC UA Servers anywhere.
- Brute force OPC UA server based on password list supplied by operator.
  - Can use a default password or compromised passwords.
- Read OPC UA structure from the server and change specific attributes.
- Better implementation of CRASHOVERRIDE OPC-DA attack methodology.

MOUSEHOLE: Remote Interaction to Attack

- MOUSEHOLE works primarily due to the open-source OPC UA library it runs on.
- Easy to get it to work and manipulate a process.
- We tested on a Kepware OPC server connected to Rockwell PLCs controlling a mock pipeline process.
- After verifying it worked, we automated an attack using MOUSEHOLE to produce a more real-world attack scenario.

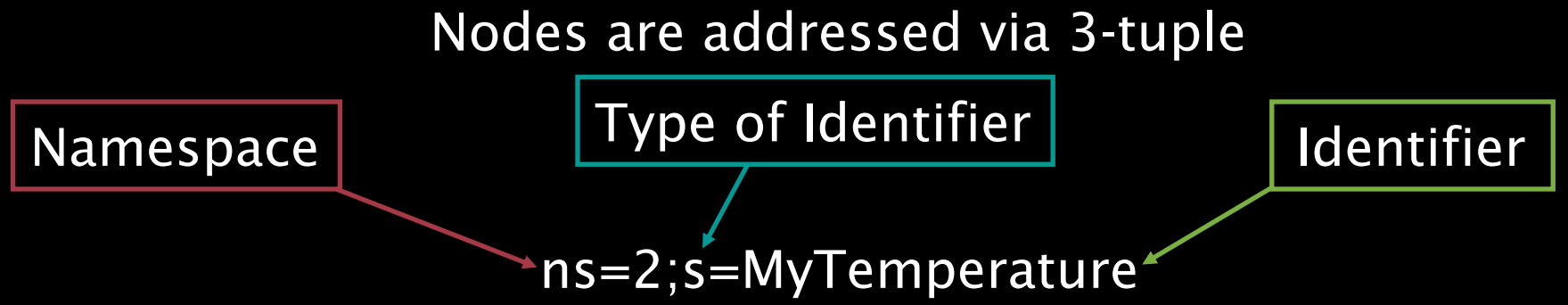
MOUSEHOLE OPERATOR VIEW

DRAGOS

MOUSEHOLE
SAFE SHUTDOWN

DRAGOS

MOUSEHOLE: Attack scenario

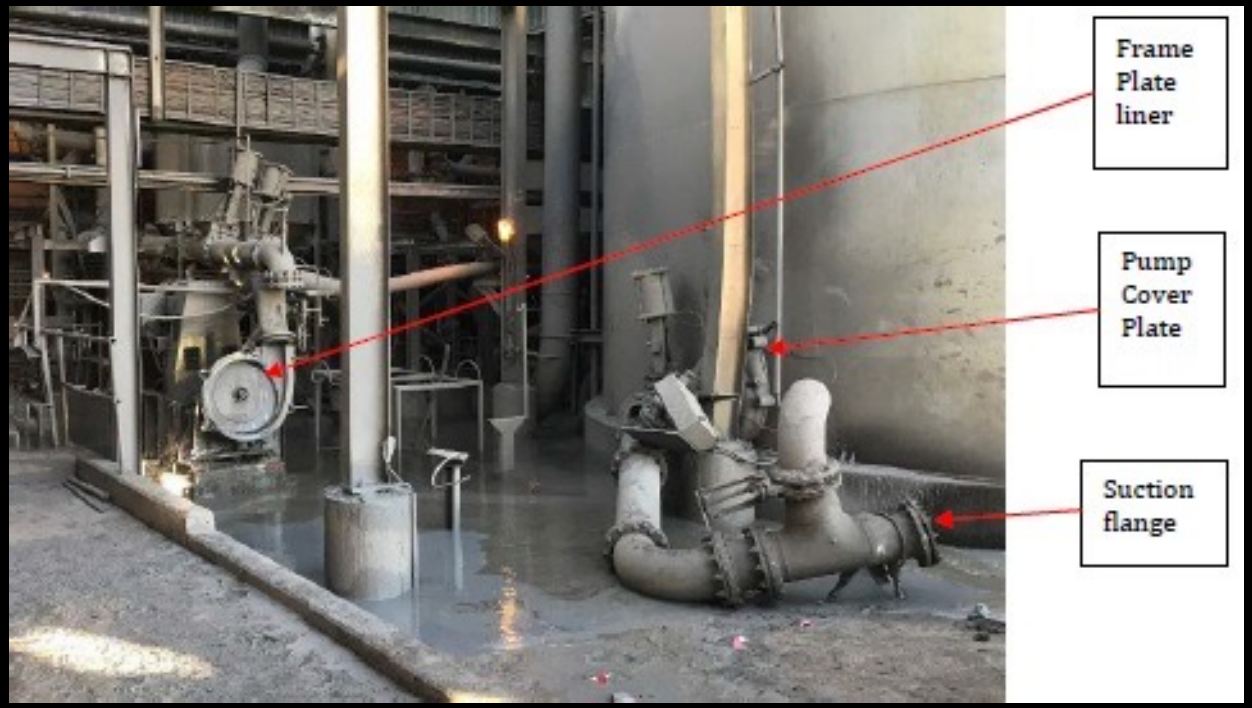


- Run MOUSEHOLE via CLI to gather Server and Node information for a process.
  - ns=2;s=Channel1.Device2.high pressure setpoint
  - ns=2;s=Channel1.Device2.Level_PID
  - ns=2;s=Channel1.Device2.solenoid energize
- Remove MOUSEHOLE and deploy an automated utility to manipulate those values.
  - High pressure setpoint 15 psi -> 100 psi
  - Level_PID: 80 -> 100 (pump speed)
  - Solenoid energize: True -> False (closes the valve)

MOUSEHOLE UNSAFE CONDITIONS

MOUSEHOLE: Potential Repercussions

Deadheading is when a pump operates with no flow through the pump due to a closed or blocked discharge valve.



Queensland Mines Inspectorate
Resources Safety & Health
Queensland

ALERT | ALERT | ALERT | ALERT | ALERT

Mines safety alert no. 346 | 04 August 2017 | Version 1

Slurry pump explosion

- A modification to the PLC logic incorrectly allowed the pump to keep running outside of designated start/stop sequence with the suction and discharge valves closed.
- Slurry inside the pump became superheated.
- This resulted in pump explosion.

EVILSCHOLAR

Framework to interact with Schneider Electric controllers via CODESYS and Modbus libraries

FORMAT:
Python + Linux ELF Library

TARGETS:
Schneider Electric Controllers

- Rapid scan that identifies all Schneider PLCs on the local network from a device that has already been compromised via User Datagram Protocol (UDP) multicast with a destination port of 27127.
- Brute force Schneider Electric PLC passwords using UDP port 1740.
- CODESYS denial-of-service attack to prevent network communications from reaching the PLC.
- Sever CODESYS connections, likely to facilitate either credential capture or to prep for DOS.
- ‘Packet of death’ attack.
- Proxy Modbus traffic through a target PLC.
- ”Maintenance” actions like logging in/out, uploading/downloading files, etc.

EVILSCHOLAR: CODESYS

- CODESYS is a device management protocol used by 100s of vendors.
- Wide usage == natural target.
- Layered protocol: Services, Channel, Datagram, Block Driver layer.
- Essentially a custom TCP implementation on top of UDP with Application layer support.

EVILSCHOLAR: Lab Set Up Issues

Devices Tested: TM241, TM251

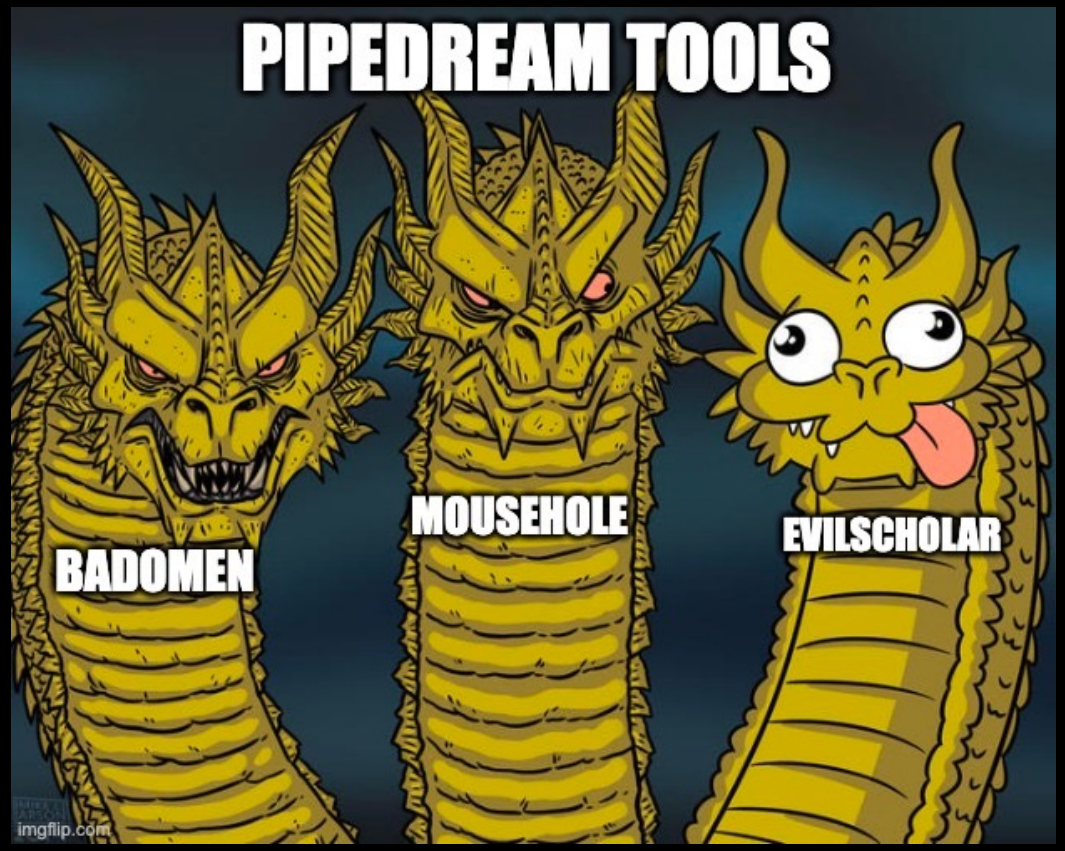
Custom CODESYS implementation is a nightmare to deal with.

CODESYS is pseudo-routable

Setting up lab connections == No VM with NAT or you won't receive packets

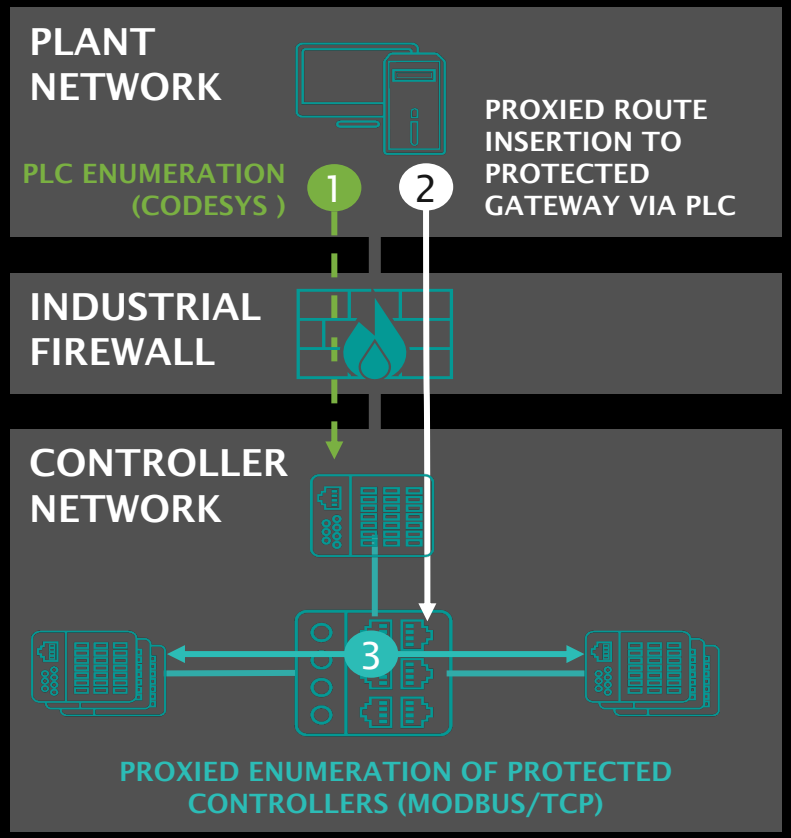
- We had a hell of a time connecting to the devices.
- In one lab, we couldn't establish any connections.
- In another lab, connecting to these devices wasn't an issue. (?!? 😡)
- We thought it was a firmware issue. (We were wrong.)

EVILSCHOLAR: Bad Assumptions



- Multiple Parts of EVILSCHOLAR's CODESYS implementation were broken, due to the developer's invalid assumptions.
- Could connect to the TM241/TM251 with no fixes if the devices were on a network of the right size.
- Once fixed, we could connect to the TM241, TM251, Raspberry Pi, and Hitachi EHV+, and start testing plugins without worrying about network sizing. (no big endian)

EVILSCHOLAR: PLCProxy Results



Originally, we thought PLCProxy worked like this.

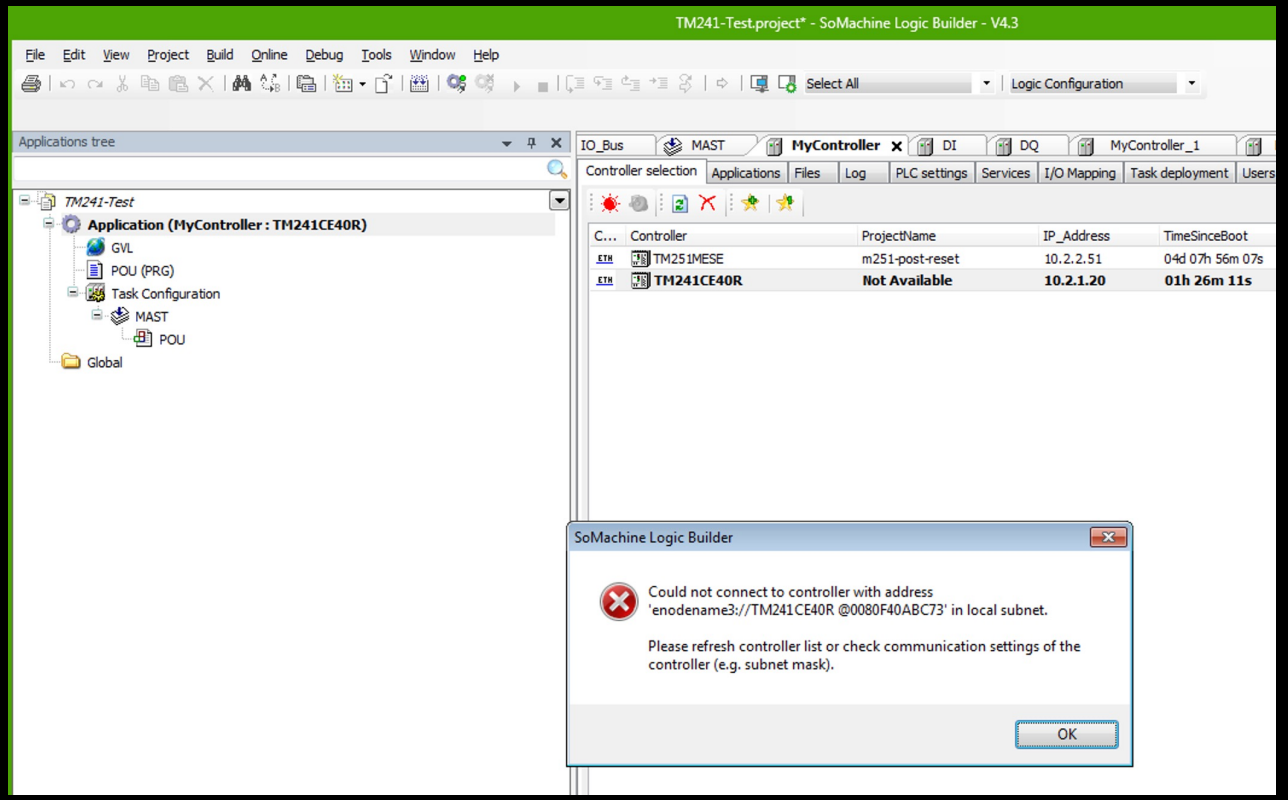
AFTER DYNAMIC ANALYSIS

- Original finding that it creates a route to gateway IP was incorrect.
  1. What it does is create a route to the internal network where gateway is located.
  2. The target network cannot be on the same network as the malware.
- While the proxy plugin only uses Modbus, it turns out that the TM251 will route any protocol it receives (SSH, HTTP, etc.).

EVILSCHOLAR: Logic Corruption

- EVILSCHOLAR allows an operator to transfer files to the device.
  - Pull logic, modify it, and put it back on the controller.
- This should be trivial provided the logic compiles to native assembly, and not an unknown bytecode.

SE Logic Corruption Result



SE Logic Corruption Takeaways

- We can crash and manipulate logic on the controller, creating various error and output states.
- In both cases, comms to the controller from the EWS are not possible without a power cycle.
- If you connect to the PLC before the crash, or before the code starts, the PLC won't let you retrieve the logic, only let you overwrite it. (EVILSCHOLAR can though!)
- Outputs are not asserted to 'FAIL SAFE' values if the crash is triggered on the *first* execution cycle.

Wait where are the deets?

- Reported the vulnerabilities to Schneider Electric and CODESYS Group on June 22.
- Following responsible disclosure process.
- Waiting on CVEs.

BADOMEN

Remote shell to interact with Omron controllers via Omron HTTP API and FINS protocol

FORMAT:
Python framework

TARGETS:
Omron equipment

- Log into a PLC with a variety of methods.
- Exploit telnet connections to the PLC to load a malware implant.
- List directories of the PLC.
- Upload, download, delete and execute files on the PLC.
- Denial-of-service (DoS) attack against a PLC.
- Terminate active PLC connections.
- Scan and identify Omron devices using FINS (Factory Interface Network Service) protocol.
- Interpret Omron device responses.
- Collect PCAP on the OT network via uploaded TCPDUMP.
- Manipulate Servos via EtherCat.
- Creating, restoring, and decoding of system process and configuration files (possible ladder logic theft).
- Change Operating Mode (Program -> Run).
- Wipe the controller's memory.

BADOMEN: Console

- Console takes advantage of CVE-2022-34151(Hard-coded Credentials) to interact with an HTTP Server on the NX1P2 and other NJ-series devices
- The server has various CGI endpoints (also used by SYSMAC studio) to manipulate and administer the device: “cpu.fcgi” and “ecat.fcgi”

```
POST /cgi-bin/cpu.fcgi HTTP/1.1
Authorization: Digest username="[redacted]",realm="CPU-Unit Interface",nonce="[redacted]",uri="/cgi-bin/
cpu.fcgi",cnonce="[redacted]",nc=00000a59,qop="auth",response="[redacted]"
Host: 192.168.56.17
Cookie: ID=1585757116
Content-Length: 14319

File_download 44f4c5794ef4d2683de1c96754ebc97845f8f55874d5ae6d64cbec,2489D786231DB9289831DC235BC8BC06 .,.+...+..5.......o6..`(.
$.G....v.............g....J='.>.......x.2.../u......,..Y..k....Z...H~.?....x..>H........m..xy.B>...5...O. @2.P!...>lK...^..|
e...O..H_..^a...wrn.z....1tk....Q.e.a..<.......P?PdYR.I}L..... ^&YCff......T.BN.>7K>,.....J
EP.~..dR....d..._......,.....T.Yg..Ik.>`.D.........3..G..g'h&.}..."U..O=..........m.!m....
```

SYSMAC logic transfer

BADOMEN: Logic Corruption

Backup & Transfer modules allow for retrieving & repackaging new logic.

Test:

1. Use the Backup Module to retrieve logic.
2. Identify binary shared object and disassemble.
3. Change entrypoint function code with a branch to a bad address.
4. Repackage code.
5. Use Transfer Module to replace files on the controller.

Results – Major Fault

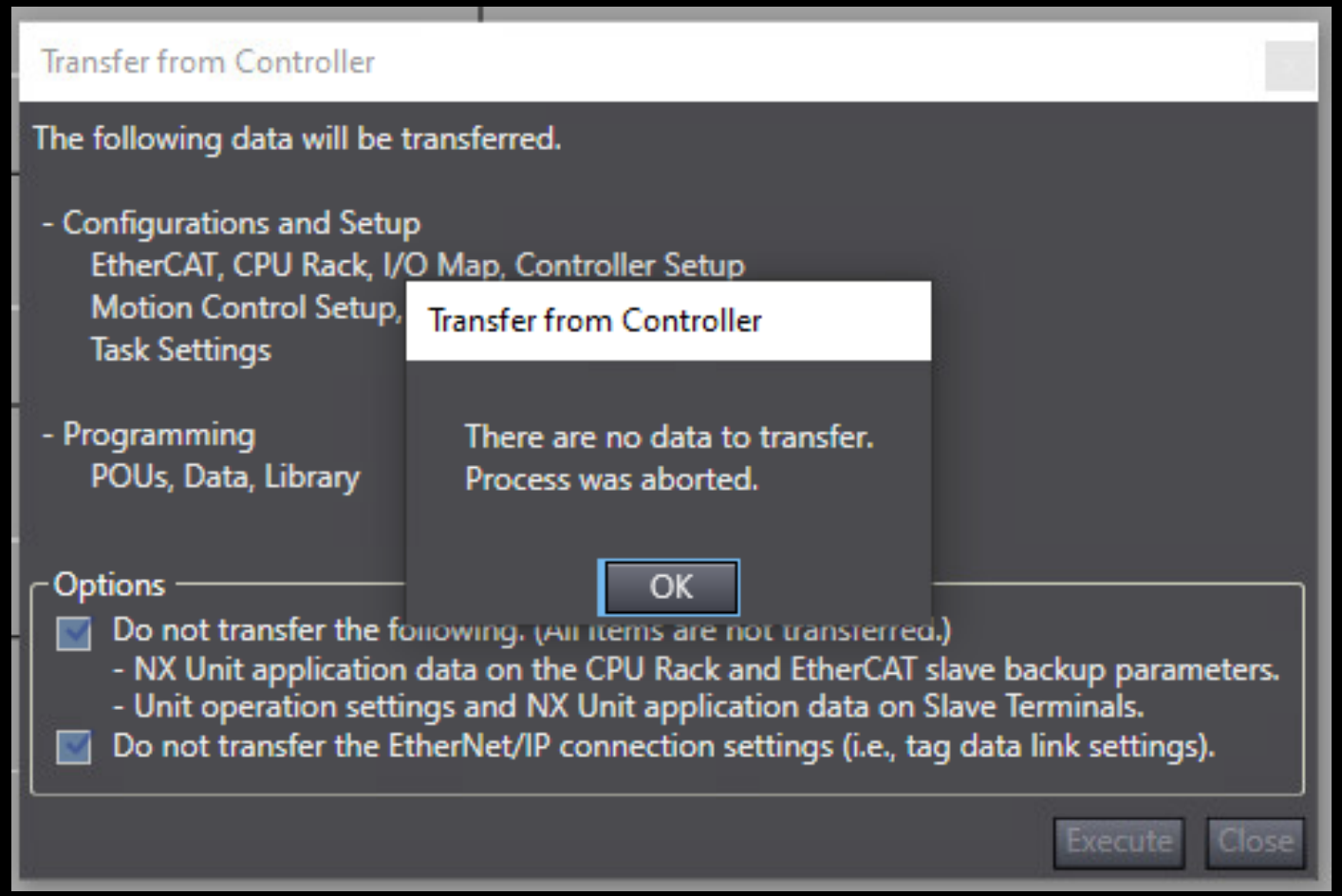
Level	Source	Source Details	Event Name	Event Code
Major fault	PLC		PLC Function Processing Error	0x40110000
Major fault	PLC		Task Execution Timeout	0x60020000

Details	A fatal error was detected in the PLC Function Module. [Cause] An error occurred in the software. [Attached information 1] System information [Attached Information 2] System information [Attached information 3] System information [Attached information 4] System information
Attached information 1	PRG:Program0

Task execution exceeded the timeout detection time.
[Cause]
(1) The timeout detection time setting is too short.
(2) The task period setting is too short.
(3) A user program is too large.
(4) The number of times that processing is repeated is larger than expected.
(5) Task Priority Error
(6) Frequent Event Task Execution
[Attached Information 1]
Name of task where error occurred.

Program Upload Fails

This also prevents the engineer from enabling Program Mode



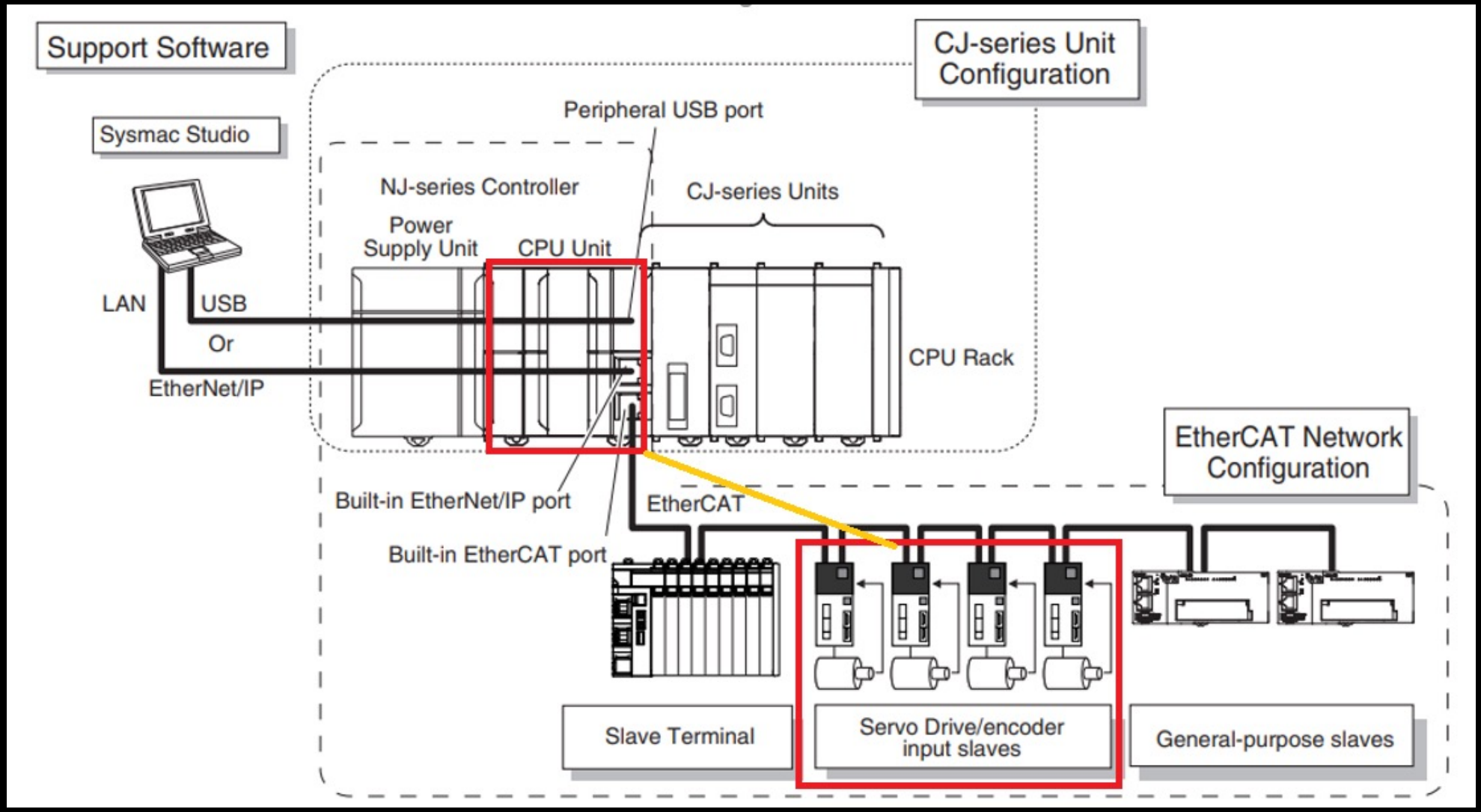
BADOMEN: Logic Corruption Confirmed

- Recovery?
  - Restore from SD Card fails, but allows you to enable Program mode
  - Then, factory reset.
  - Finally, you can restore logic to controller.
- Still under investigation:
  - Other methods of recovery.
  - What our code modification did to the controller.

BADOMEN: Servo Module Testing

Devices Tested: NX1P2, R88D-1SN10F-ECT & R88D-1SN01L-ECT Servo Drives, & R88M-1M10030S-S2 AC Servo Motor

- BADOMEN has a Servo Module for reading and writing Servo Drive parameters via EtherCat.
- Servo Motor spins a shaft.
- Servo Drive powers the motor, controls the motor, handles comms to the master PLC.
- 1SN10F is a 380VAC to 480VAC drive. 1SN01L is a 100-120VAC Drive.
- Verified that comms worked with the large drive, then switched to the small one.
  - No access to reliable source of three phase 480VAC
  - Using 120VAC greatly reduces risk of accidental electrocutions or arcing



ETHERCAT SETUP

BADOMEN: Troublesome Parameters

We disabled the following parameters:

Excessive Velocity Deviation Detection (3B60.05)

Warning Mask 1 selection (4020.01)

Warning Mask 3 selection (4020.03)

Position Detection Function -Following Error (3B50.05)

We manipulated these:

Excessive Speed Detection (3B60.04) ← Set to 1 stops the servo
Set to max means 1.2x max speed

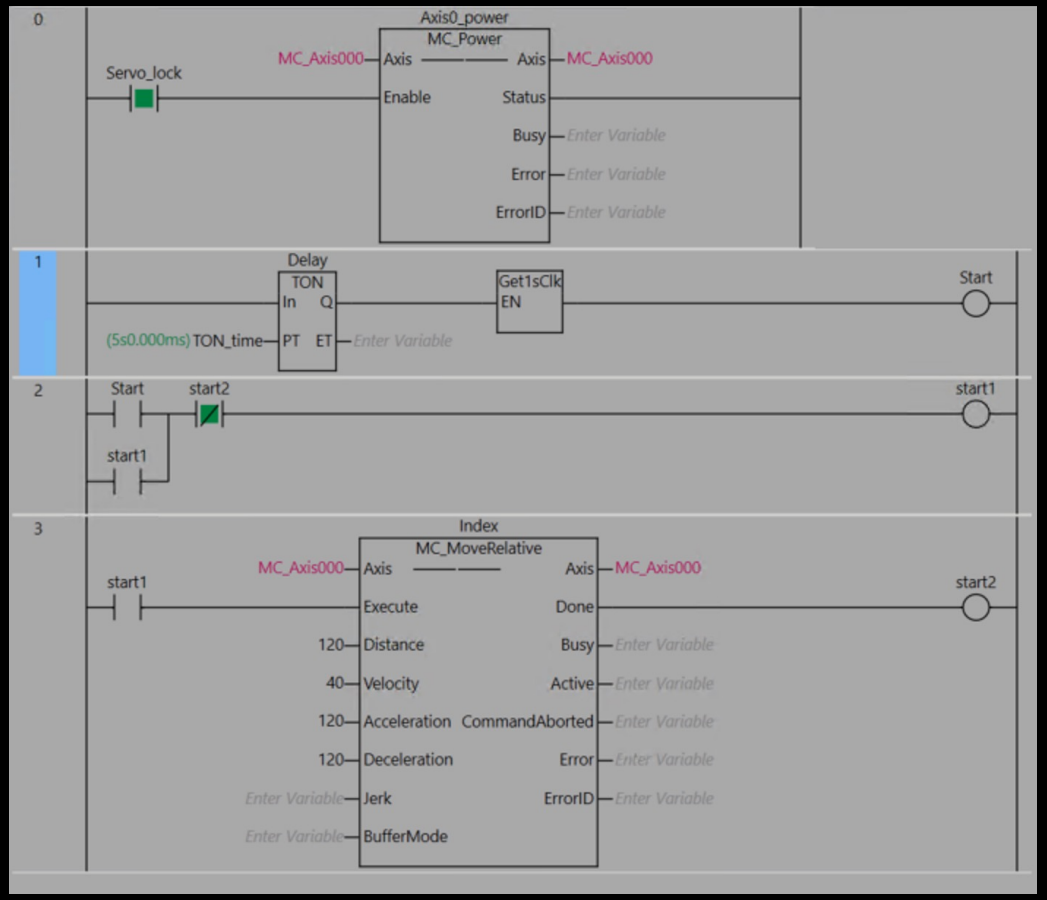
Set Vibration Detection (3B70.01) ← Set this to max 500%

BADOMEN: Servo Logic Manipulation

- Manipulating Parameters is interesting and likely more of a precursor to an attack.
  - Although spamming Excessive Speed Detection would be a bad day.
- The NX1P2 stores the program that controls the servo drive.
- We already know we can modify code on the device.
- Can we modify code that controls the Servo's RPM?

Step 1: Download Logic to the Device

Step 0: Watch YouTube video on setting up Servo



Step 2: Grab Backup and Decompile

Function name	Segment	Start	Length	Locals
POU_runFUNCTION_BLOCK	.plt	00000548	0000000C	
LPOU_NX_Get1sClk	.plt	00000554	0000000C	
start	.text	00000560	00000008	
sub_568	.text	00000568	00000014	
sub_57C	.text	0000057C	0000018C	00000028
__imp_POU_runFUNCTION_BLOCK	extern	00001DD4	00000004	
__imp_LPOU_NX_Get1sClk	extern	00001DD8	00000004	

Step 3: Patch args
Step 4: Transfer to PLC

```
*(_BYTE *)(v17 + 16) = 0;
// Parameter settings before calling function block
// Distance,
// Velocity
// Acceleration
// Deceleration


// It appears that it may hvae just compiled it in order.
// Numbers are represented as IEE754 floats
**(_WORD **)(v16 + 8) = v14;
*(_QWORD *)*(_DWORD *)(v16 + 16) = 0x405E000000000000LL;// 120 units, one of distance, acceleration or deceleration
v18 = *(_DWORD **)(v16 + 24);
*v18 = 0;                               // 40 units - this is the Velocity
v18[1] = 0x40440000;
*(_QWORD *)*(_DWORD *)(v16 + 32) = 0x405E000000000000LL;// 120 units, one of distance, acceleration or deceleration
*(_QWORD *)*(_DWORD *)(v16 + 40) = 0x405E000000000000LL;// 120 units, one of distance, acceleration or deceleration
result = POU_runFUNCTION_BLOCK();
```

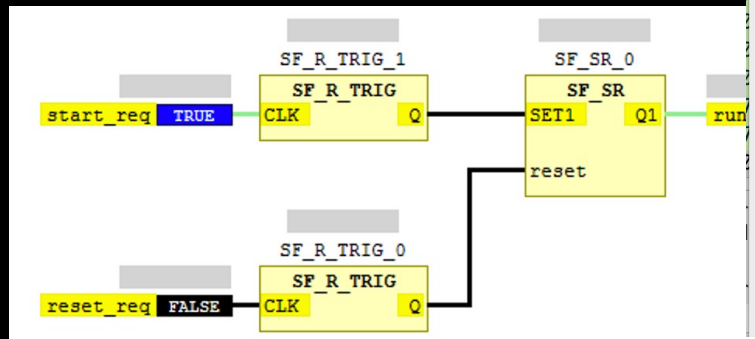
```
>>>
>>> from ieee754 import IEEE754
>>> conv = lambda x: print(IEEE754(x).str2hex())
>>> conv(120.0)
405e000000000000
>>> conv(40.0)
4044000000000000
>>>
```

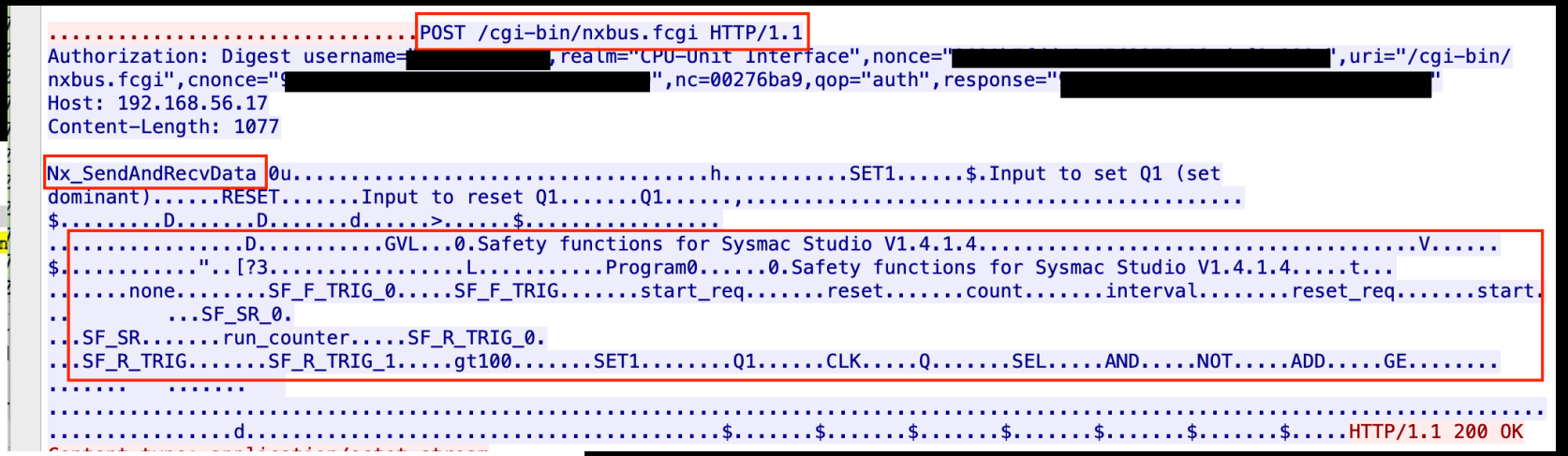
BADOMEN
MAX VELOCITY

BADOMEN: Safety System Corruption

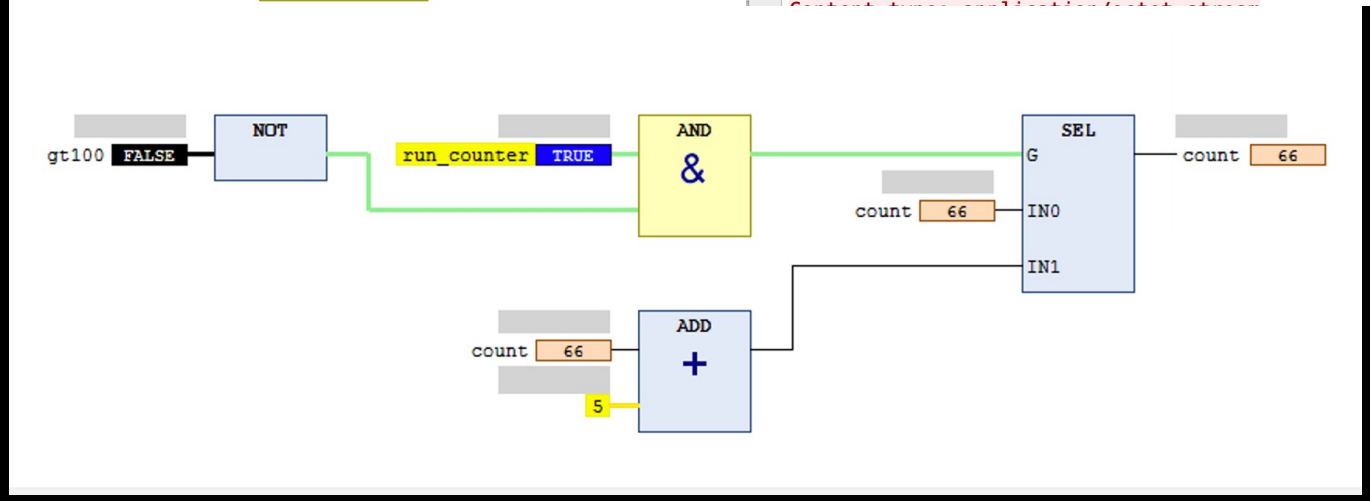
- Even from the UI, it was clear that safety programming was a very separate process:
  - Doesn’t transfer logic the same way.
  - Put the controller in program mode, then debug stop. This transfers logic.
  - Then transition to Debug Run, followed by a Safety Validation.
  - Finally, the safety controller can enter run mode.

BADOMEN: Safety Program Download





SYSMAC Safety Program Download



Safety Function

BADOMEN: No support for Safety Programs

- BADOMEN only supports the CPU and ECAT endpoints.
- Safety Programming uses the NxBus endpoint.
- There are no references to this endpoint in BADOMEN.
- This is likely the next step development-wise for a full-fledged attack against a Servo.
- (Unless the network is poorly configured.)

Real World Usage

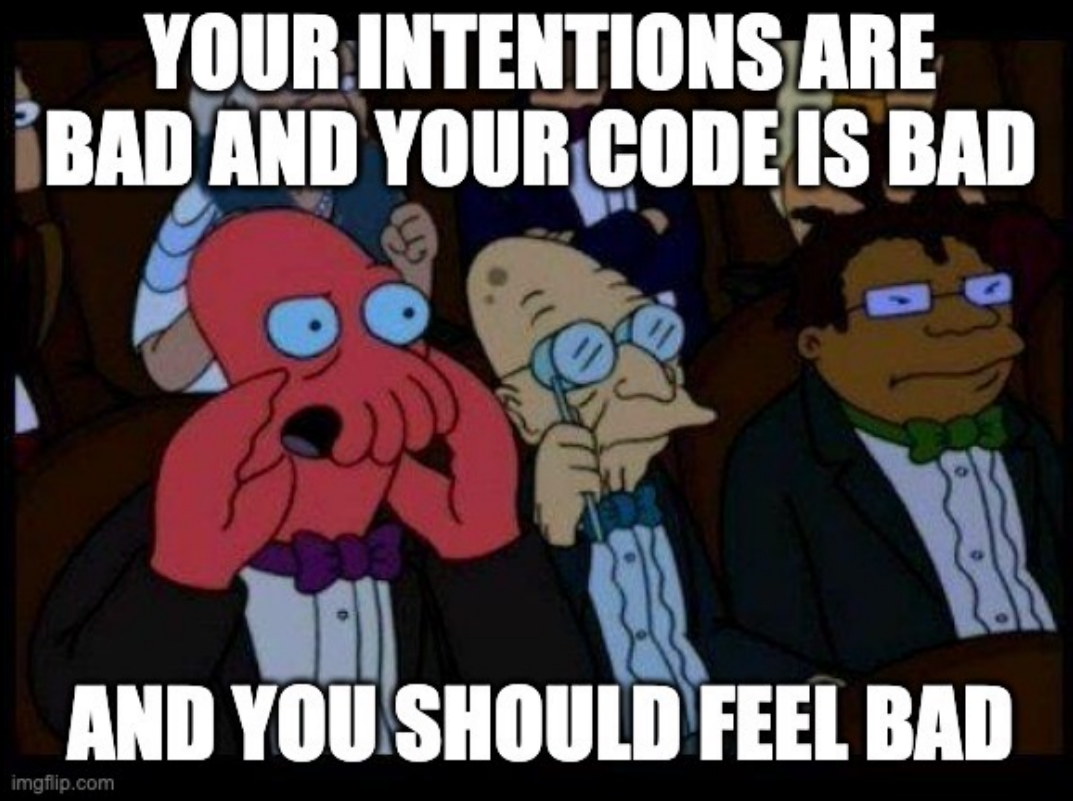
Real World Usage

- These utilities can cause problems now.
- But in our judgement, usage would look a lot like our MOUSEHOLE demo
  - IT intrusion to steal important process documentation
  - Use PIPEDREAM to recon plant network and find important PLCs that control key aspects of the target process
  - Use this knowledge to develop automated attack utilities to achieve a disruptive/destructive effect.

Conclusion

PIPEDREAM is a new escalation in mal dev

- First time we've seen malware devs attempt a plugin framework for specific PLCs/Protocols.
  - ICSsploit and Metasploit "SCADA" aren't quite the same.
- ICS process agnostic.
- Enables an operator to conduct recon and disruption.



Industry Improvements

- Open up Protocols and Operating Systems.
  - We're so behind vs IT Software and Internet Protocols.
- EWS or separate vendor distributed utilities need to support forensics for IR and post-mortems.
  - The fact that SoMachine and Sysmac can't pull a bad project file is an issue.
  - Vendors aren't the only experts anymore.
- Implement better integrity checking for running code. Don't assume code always comes from the EWS!

Thank You!

For mitigations:
dragos.com/pipedream

Jimmy Wylie
jimmy@dragos.com
@mayahustle