Through the Eyes of the Attacker:

Designing Embedded Systems Exploits for Industrial Control Systems

Midnight Blue



Jos Wetzels, Marina Krotofil

DEF CON 26, August 11, 2018

Who Are We?

Midnight Blue

Jos Wetzels

Independent Security Researcher @ **Midnight Blue**
Embedded Systems (ICS, Automotive, IoT, …)

(Previously) Security Researcher @ **UTwente**
Critical Infrastructure / ICS

@s4mvartaka
http://www.midnightbluelabs.com
http://samvartaka.github.io

Marina Krotofil

ICS/SCADA security professional, focusing on offensive and defensive cyber-physical security

Previously Principal Analyst at FireEye and Lead Cyber Security Researcher at Honeywell

@marmusha
marmusha@gmail.com
https://www.linkedin.com/in/marina-krotofil

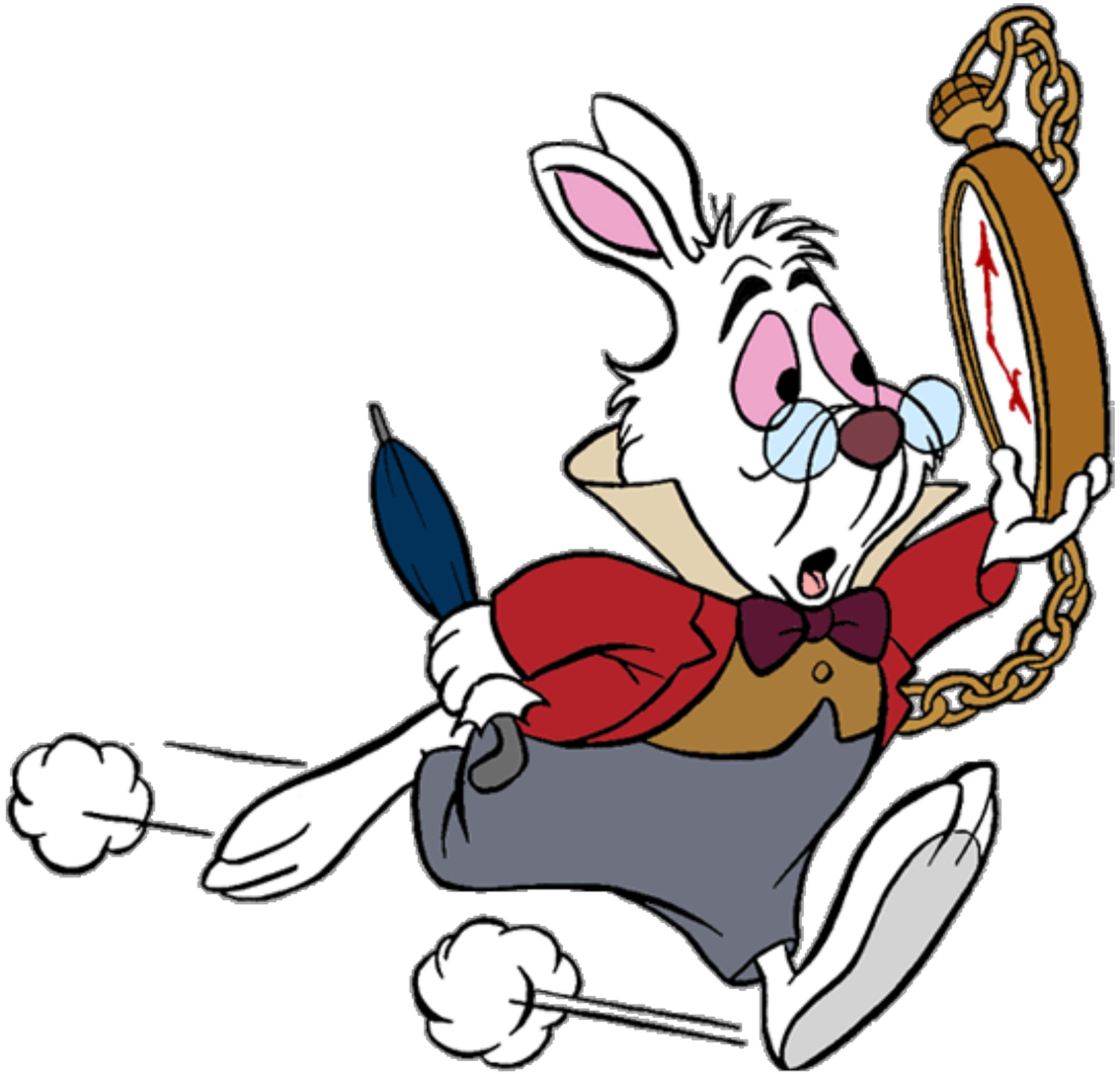
ROADMAP

Midnight Blue

- Introduction
- ICS Device Exploitation
- Developing ICS Device Implants & OT Payloads
- Conclusions

WARNING: FAST PACED TALK

Midnight Blue



https://www.disneyclips.com/imagesnewb/alice4.html

INTRODUCTION

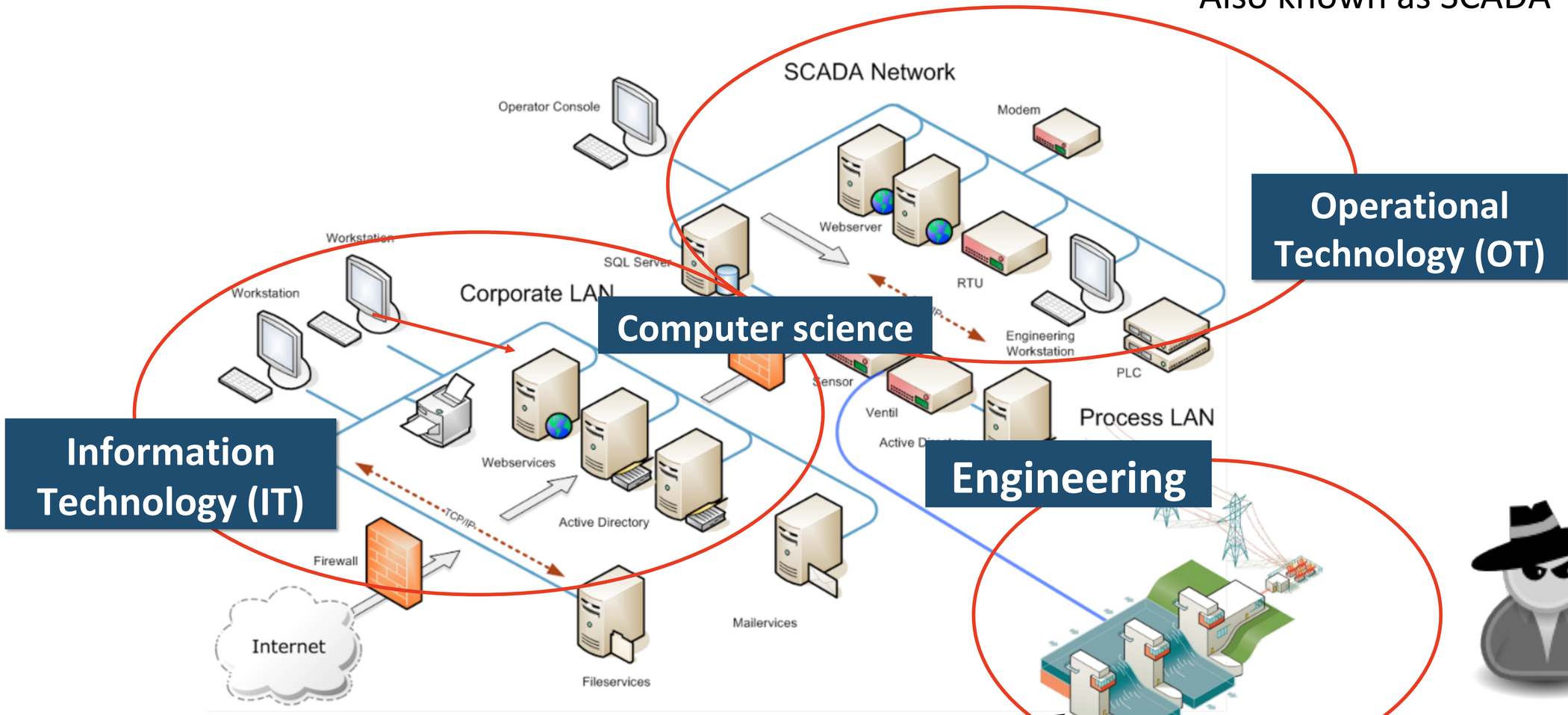


http://ats-transporttechnieken.nl/wp-content/uploads/photo-gallery/Draadloze%20shuttle%20voor%20zwembaden/2H8_016.JPG

Industrial Control Systems (ICS)

Also known as SCADA

Midnight Blue



Physical process and control equipment

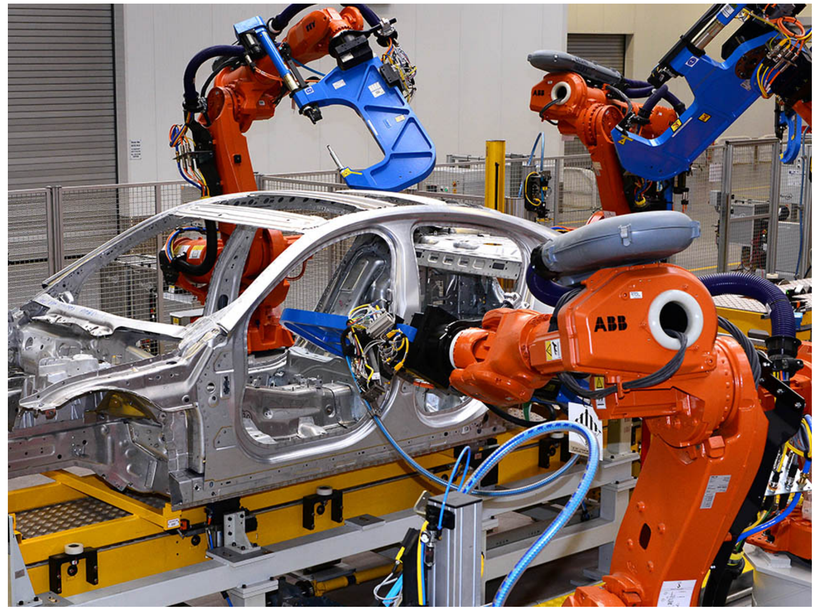
Midnight Blue



https://vecer.mk/files/article/2017/05/02/485749-saudiska-arabija-ja-kupi-najgolemata-naftena-rafinerija-vo-sad.jpg



http://www.jfwhite.com/Collateral/Images/English-US/Galleries/middleboro9115kvbreakers.jpg



https://www.roboticsbusinessreview.com/wp-content/uploads/2016/05/jaguar-factory.jpg



https://www.oilandgasproductnews.com/files/slides/locale_image/medium/0089/22183_en_16f9d_8738_honeywell-process-solutions-rtu2020-process-controller.jpg



https://selinc.com/uploadedImages/Web/Videos/Playlists/Playlist_RTAC_1280x720.png?n=63584758126000



http://www02.abb.com/global/seitp/seitp202.nsf/0/0601d25ed243cfb0c1257d7e0043e50e/$file/7184_lvl2.jpg

Cyber-Physical Attacks

Midnight Blue







PHYSICAL



CYBER

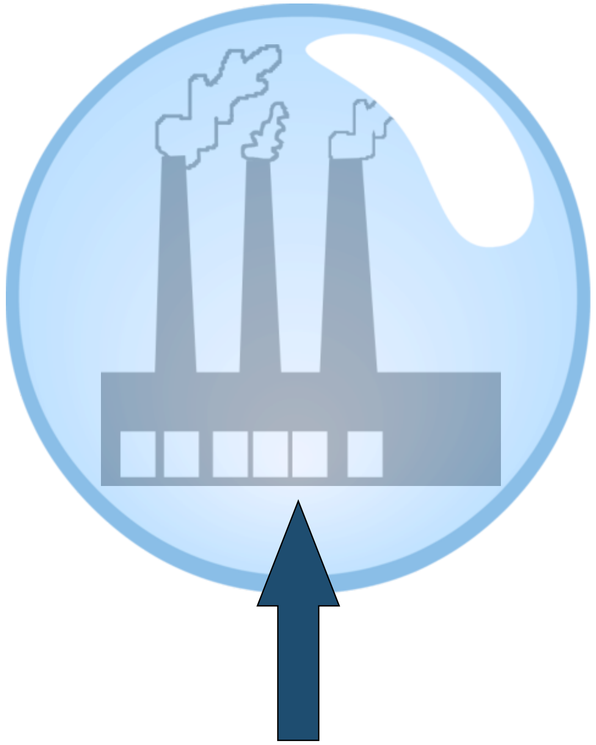


https://www.oilandgasproductnews.com/files/slides/locale_image/medium/0089/22183_en_16f9d_8738_honeywell-process-solutions-rtu2020-process-controller.jpg

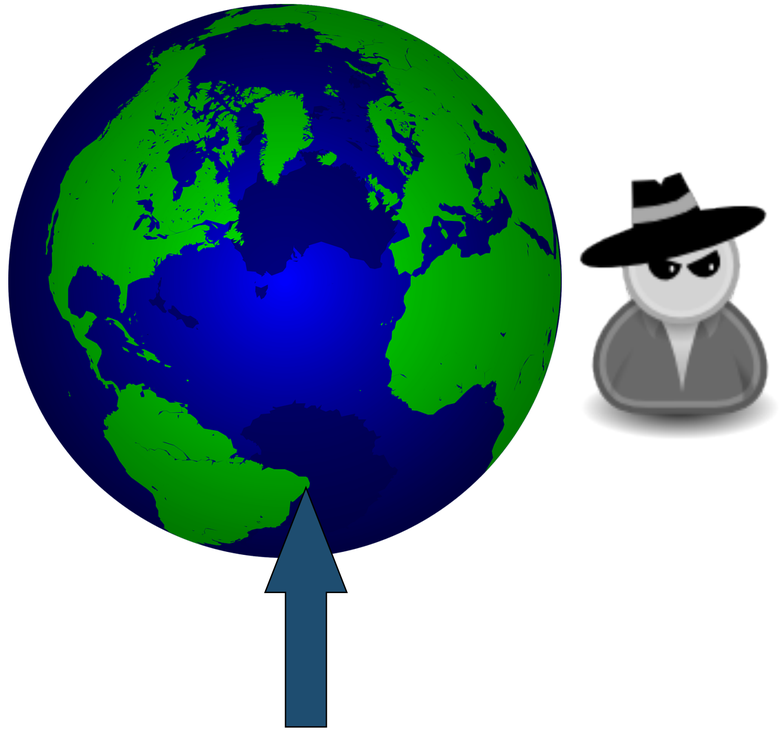
http://magazine.scientificmalaysian.com/wp-content/uploads/2013/12/Oil-refinery-explosion.png

ICS threat landscape Has Changed

Midnight Blue



Nobody even knows about our existence



Crazy amount of hacking on a daily basis

ICS Threat Landscape Has Changed

Midnight Blue

Planned operation to hinder Iran's nuclear program

Recon and weaponization of capabilities

It's happening: Publicly known cyber-physical attacks



https://qph.fs.quoracdn.net/main-qimg-f741c6e5db32b87f282e54448a2129ce

2010

STUXNET

2013

First publicly known OT recon activities (HAVEX)

2015

Ukr. power grid attack (BlackEnergy)

2016

Ukr. power grid attack (Industroyer)

2017

TRITON



https://www.schneider-electric.com/ww/en/images/tricon-IC-654x654.jpg

Seems to be a trigger and a tipping point

TRITON Attack

Midnight Blue

THE WALL STREET JOURNAL.

TECH

New Type of Cyberattack Targets Factory Safety Systems

Malicious software Triton was able to manipulate Schneider Electric devices' memory and run unauthorized programs by leveraging a previously unknown bug

The Washington Times

Industrial safety systems targeted by Triton malware meant to cause 'physical consequences': Reports

WIRED

ANDY GREENBERG SECURITY 12.14.17 10:00 AM

UNPRECEDENTED MALWARE TARGETS INDUSTRIAL SAFETY SYSTEMS IN THE MIDDLE EAST

Hackers use Triton malware to shut down plant, industrial systems

The malware has been designed to target industrial systems and critical infrastructure.

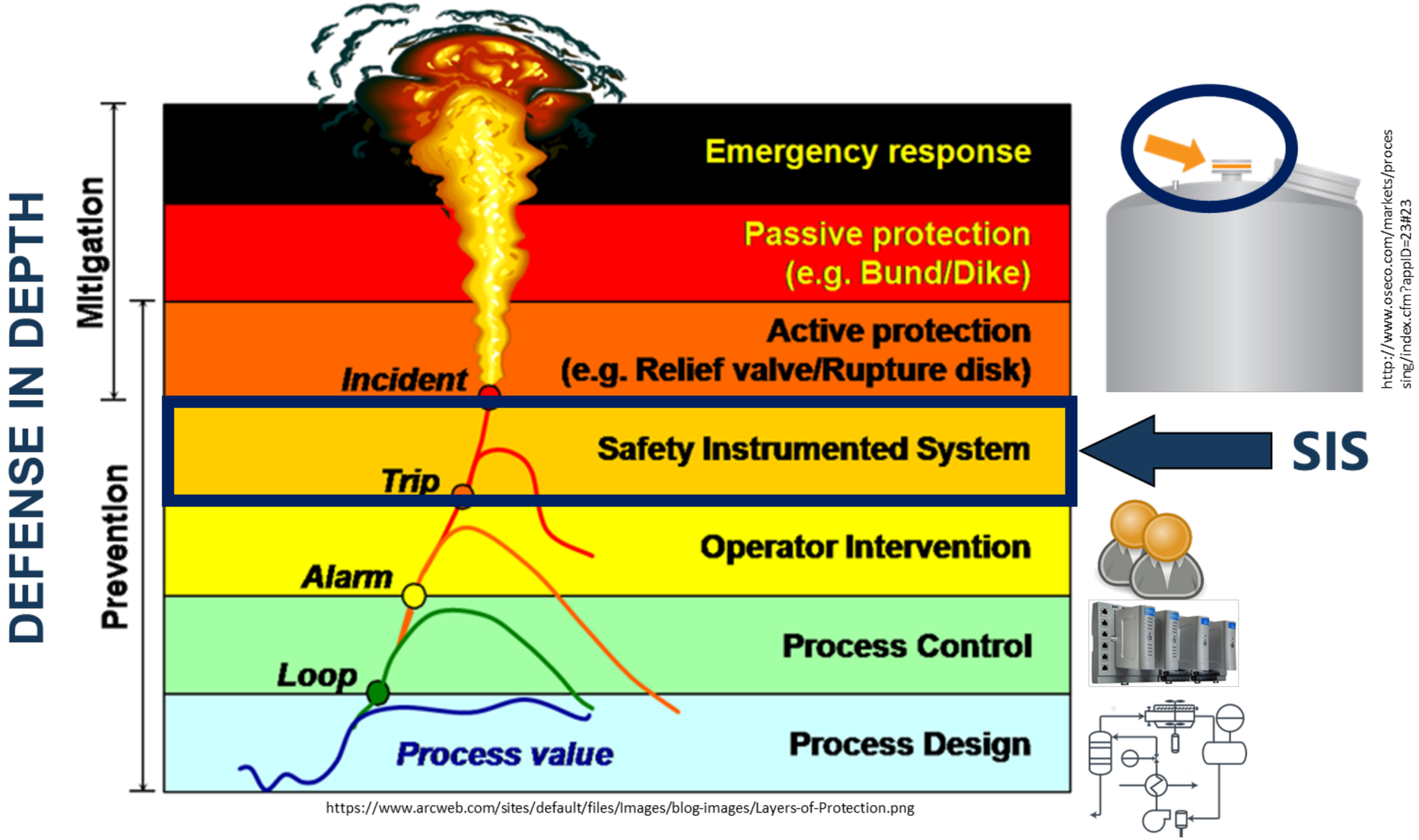


By Charlie Osborne for Zero Day | December 15, 2017 -- 09:54 GMT (01:54 PST) | Topic: Security

ZDNet

Hazards and Layers of Protection

Midnight Blue

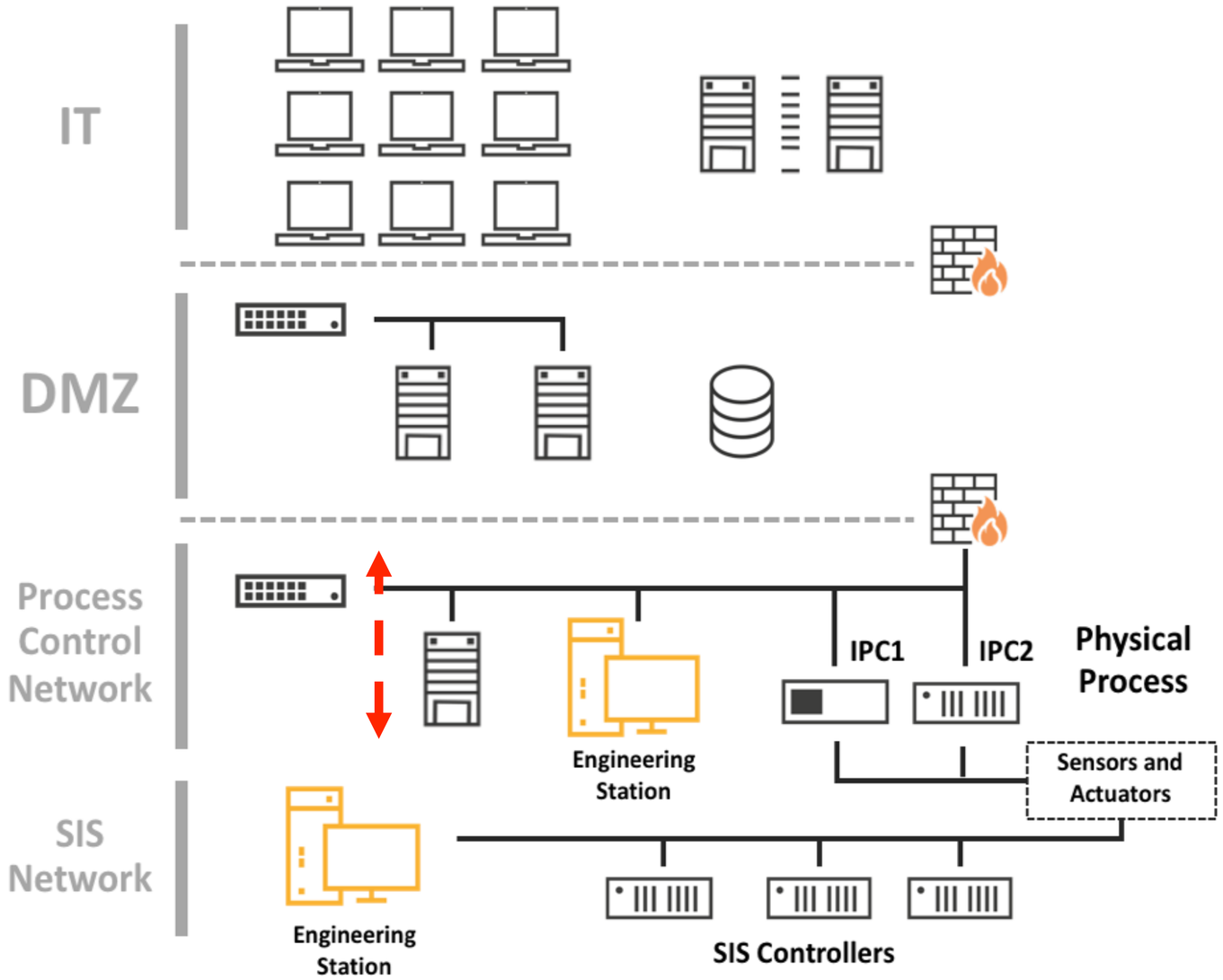


https://www.arcweb.com/sites/default/files/Images/blog-images/Layers-of-Protection.png

DCS / SIS Network

Midnight Blue

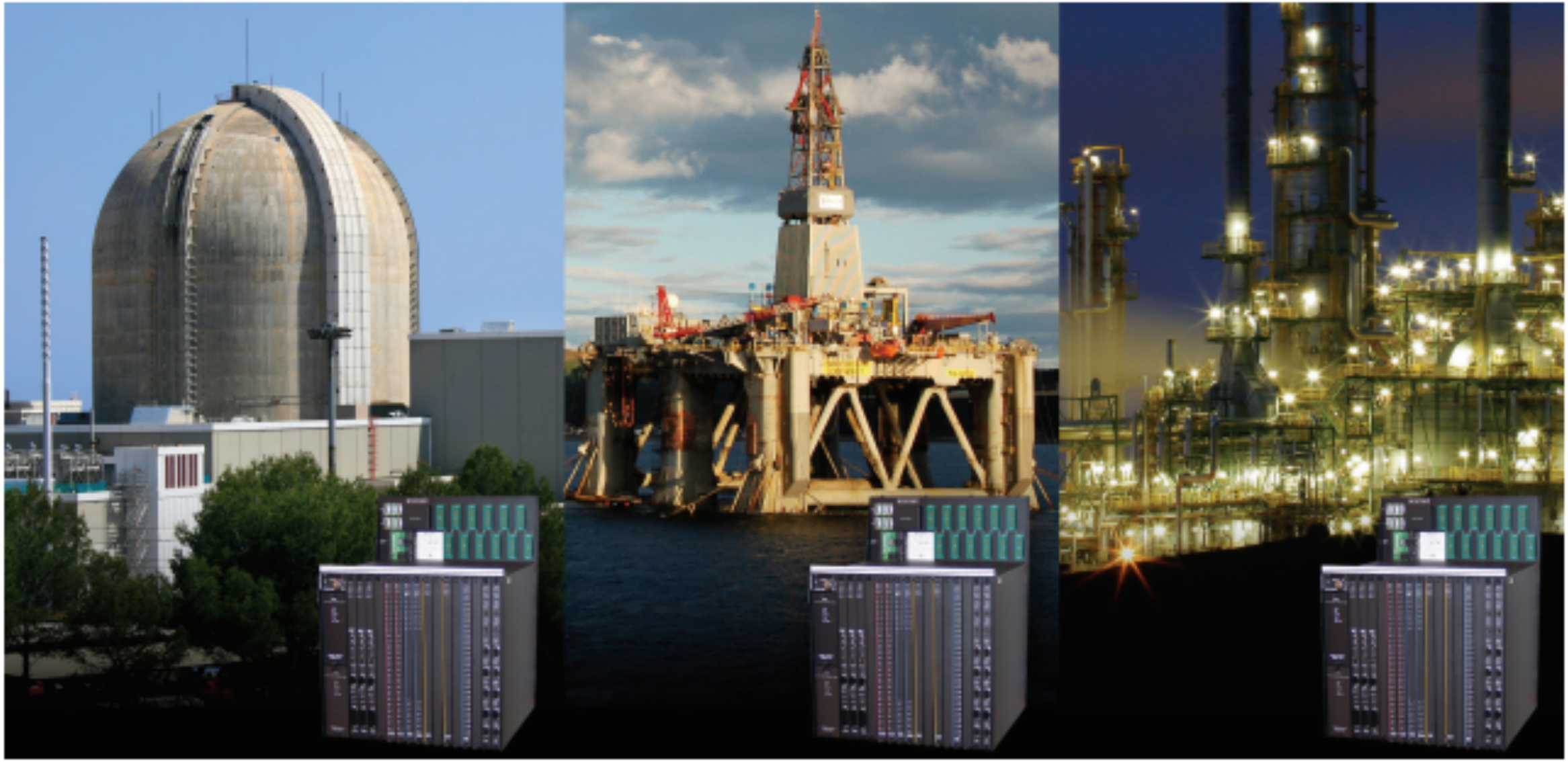
An attack on a safety system can cause the MOST DAMAGING outcome of a cyber-physical attack



https://www.cyberark.com/threat-research-blog/anatomy-triton-malware-attack/

Triconex (SIL3)

Midnight Blue



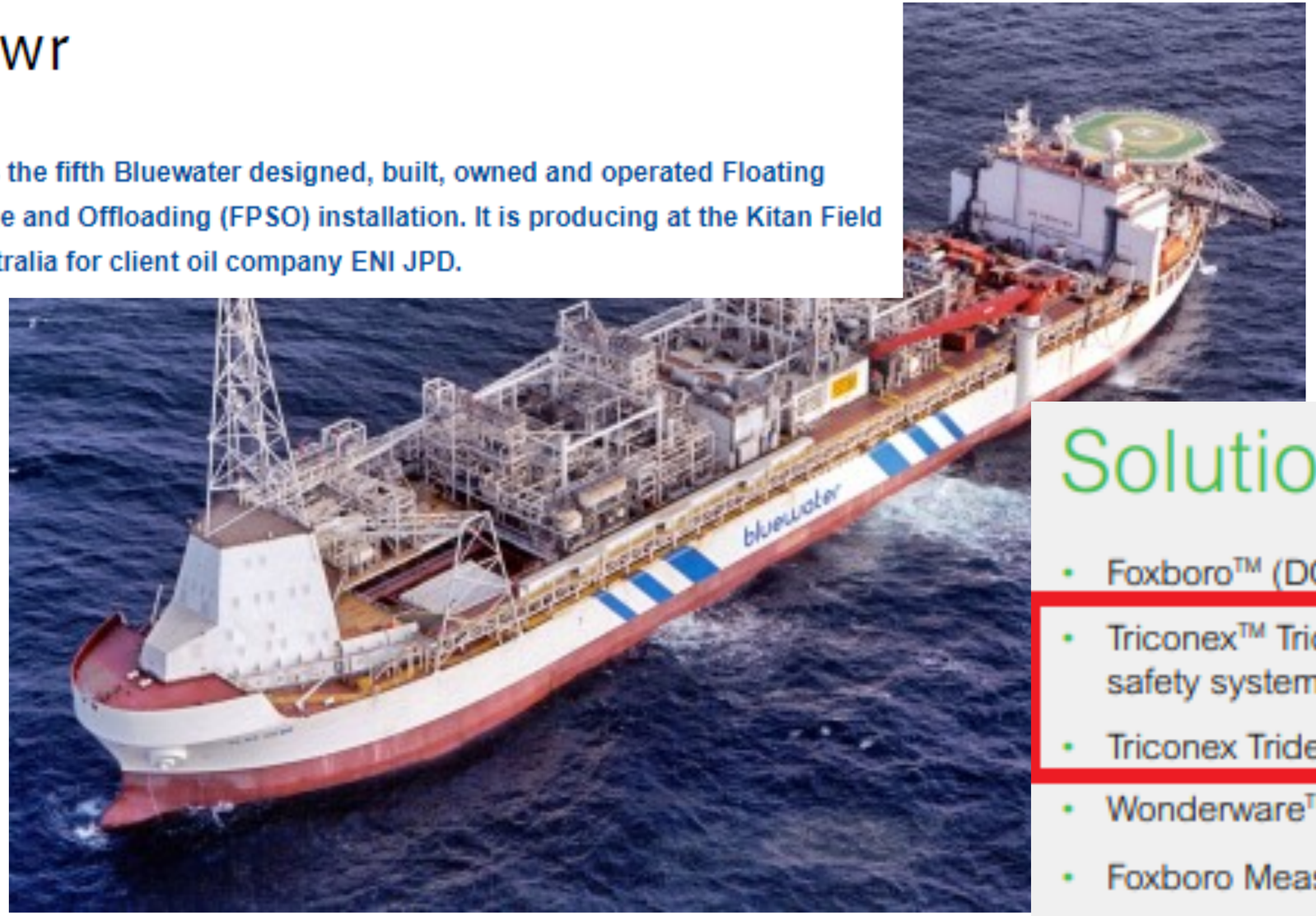
http://iom.invensys.com/EN/pdfLibrary/Datasheet_Triconex_TriconSIL3_06-11.pdf

Triconex is everywhere…[OSINT]

Midnight Blue

Glas Dowr

The 'Glas Dowr' is the fifth Bluewater designed, built, owned and operated Floating Production Storage and Offloading (FPSO) installation. It is producing at the Kitan Field off northwest Australia for client oil company ENI JPD.



https://www.bluewater.com/fleet-operations/our-fpso-fleet/glas-dowr/

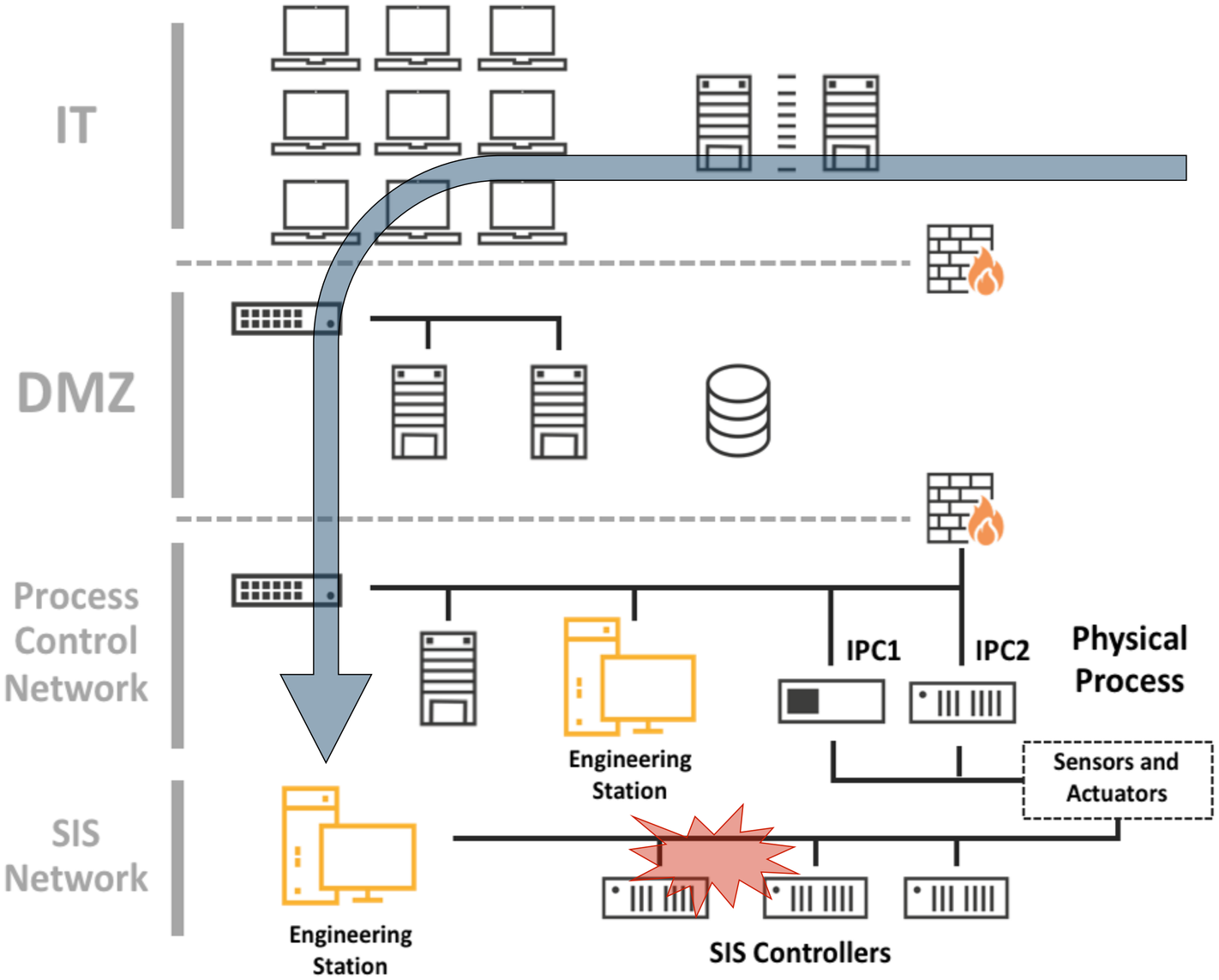
Solutions

- Foxboro™ (DCS)
- Triconex™ Tricon™ triple redundant safety system
- Triconex Trident™ safety system
- Wonderware™ Historian
- Foxboro Measurement and Instruments

http://software.schneider-electric.com/about-us/success-stories/listing-content/bluewater/

TRITON Attack: Overview

Midnight Blue





Attacker obtained remote access to SIS work station

https://www.cyberark.com/threat-research-blog/anatomy-triton-malware-attack/

TRITON Payload: Overview

Midnight Blue

Attacker attempted to inject passive backdoor/remote access trojan into industrial safety controller

- Read arbitrary memory
- Write into memory
- Execute arbitrary code

“Your wish is my command”



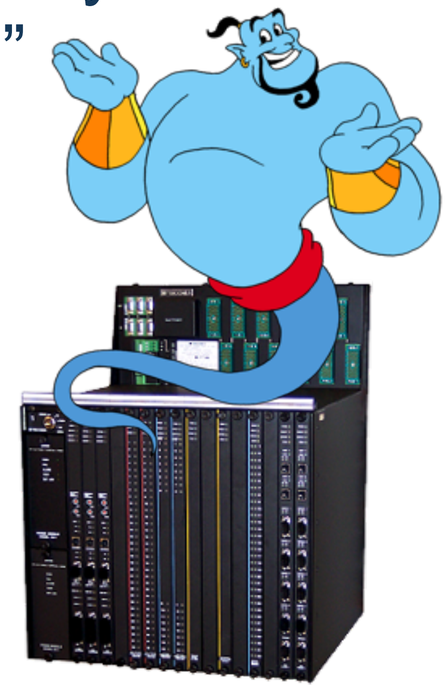
Eng. Workstation

trilog.exe

- script_test.py
- library.zip
- inject.bin
- imain.bin

TriStation protocol

imain.bin + inject.bin



Attacks on Industrial Systems

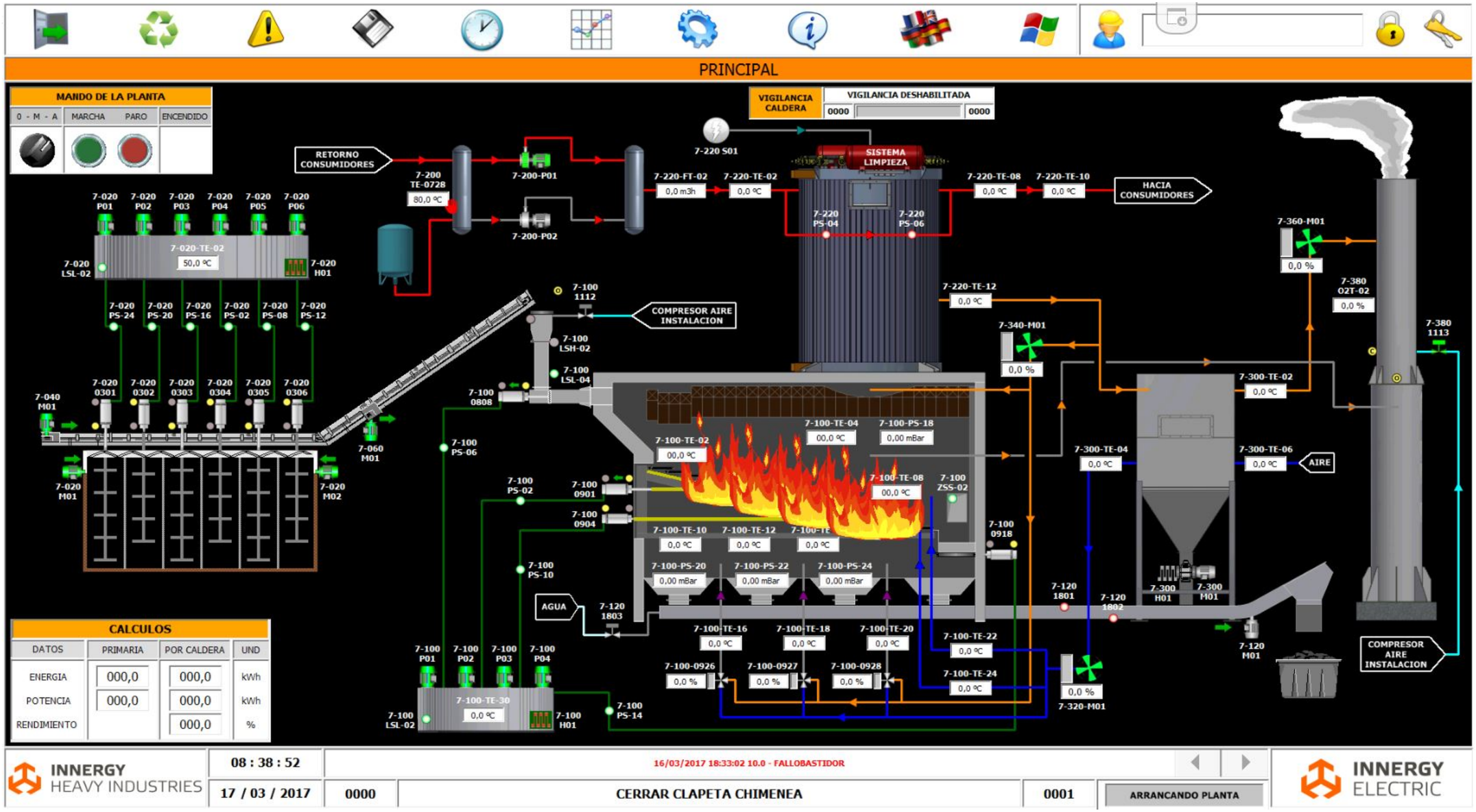
Midnight Blue

- **Attack scenario depends on attacker goal**
- **Sometimes this means explosions**
- **Sometimes it doesn't**
  - No need to hit someone with a baseball bat if you want to slap their wrist
- **Simple process shutdown / economic disruption might do fine**
  - Doubles as **testing round** for offensive ICS toolkit + TTPs …

M. Krotofil, J. Larsen "*Rocking the Pocket Book: Hacking Chemical Plants for Competition and Extortion*", Def Con, 2015

Simple ‘Do Not Press’ HMI Attack

Midnight Blue

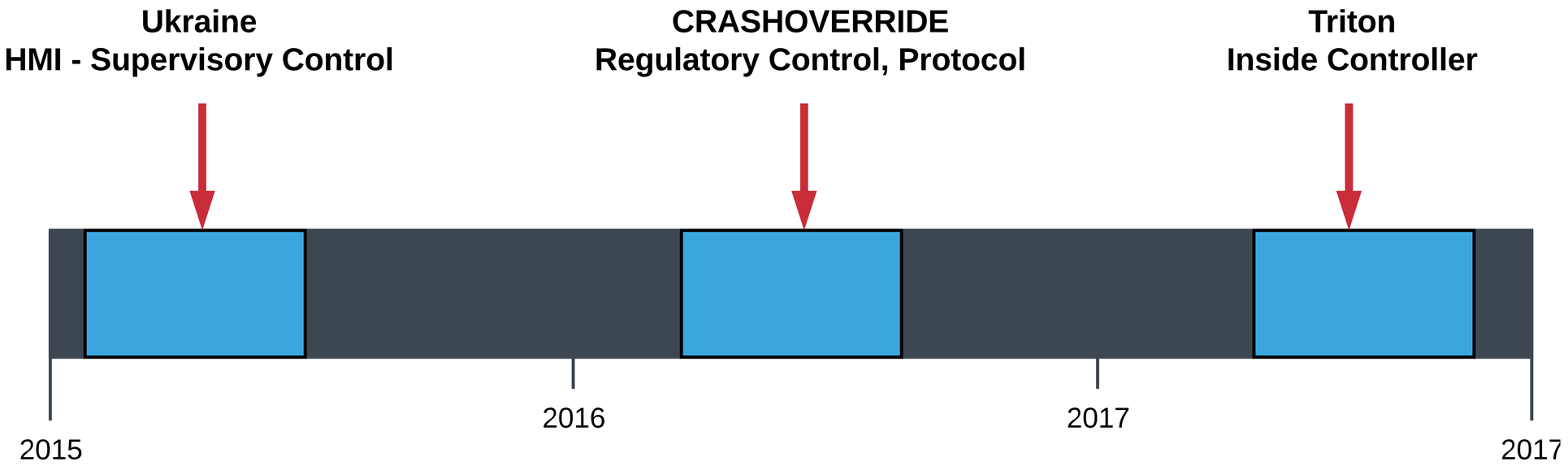


http://innergy-global.com/en

Source: Innergy

More Complex Attacks

Midnight Blue



TRITON used an **implant** on Triconex SIS
Process shutdown could've been achieved *much* easier

What's going on here?

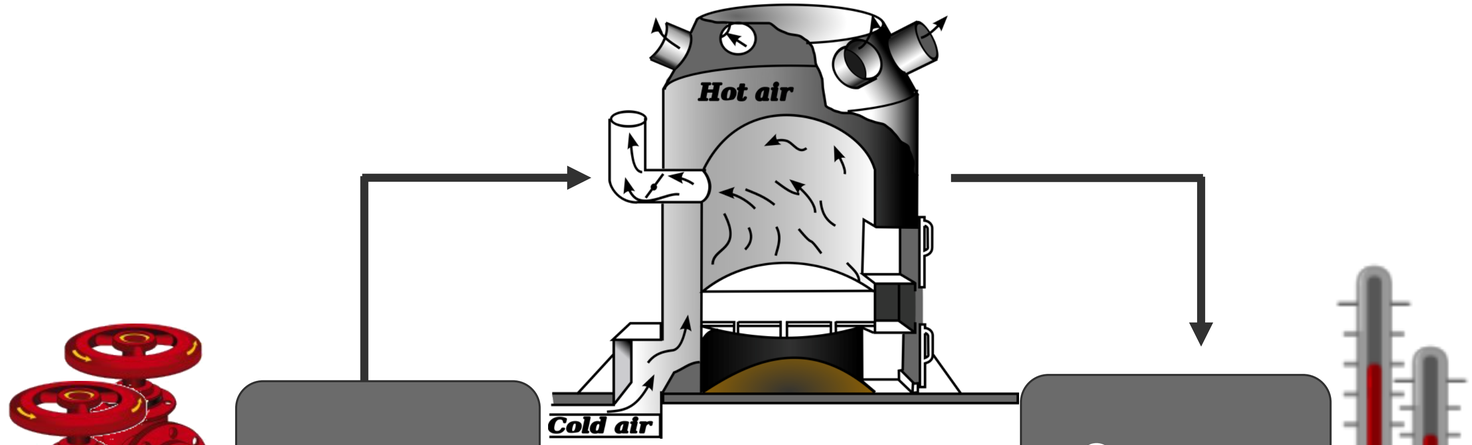
More Complex Attacks

Midnight Blue

- Industrial processes are complicated and are built to be **robust** & **recoverable**
- More complex attacks with significant, 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?

Industrial Attacks Are About Control Loops

Midnight Blue

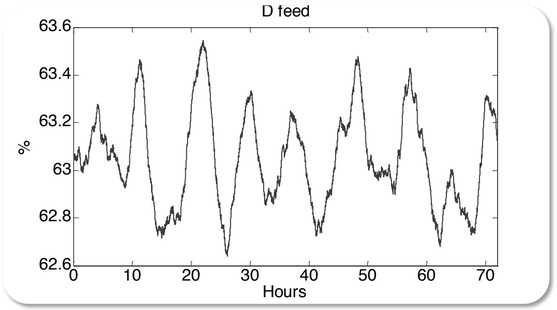


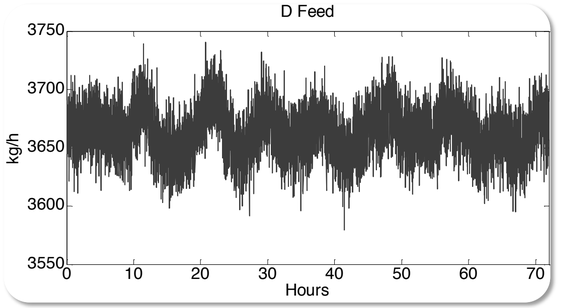


Adjusted to influence process behavior

Measure process state

Computes control commands for actuators





Industrial Attack Approach

Midnight Blue



1

Manipulate the process

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

2

Obtain Feedback

Direct or Derived (e.g., via proxy sensors / calculations)

Often hardest to achieve

3

Prevent response



- **Operators**
  - 
  - Blind
  - Mislead
- **Control / Safety System**
  - Modify operational / safety limits
  - Blind about process state

OT Payload

Where does **TRITON** fit into all this?

Midnight Blue



1

Manipulate the process

- **Direct**
  - Manipulation of actuators



- **Indirect**
  - Deceive controller/ operator about process state (e.g. spoof sensor)



2

Obtain Feedback

- Direct or Derived (e.g., via proxy sensors / calculations)

3

Prevent response



- **Operators**
  - Blind
  - Mislead
- **Control / Safety System**
  - Modify operational / safety limits
  - Blind about process state



OT Payload

Clandestine Control Loops

Midnight Blue

- Cyber-Physical Attack is collection of '***clandestine control loops***'
  - Cycle of process observation & manipulation to achieve **unsafe state**

- **Attack Timing is Crucial**
  - Processes aren't vulnerable all the time
  - Many damage scenarios take time to execute

- **Attack Coordination is Crucial**
  - Observation of state A in component B needs to trigger payloads X, Y, Z

- Requires **granular control** across process
  - Manage task quantity & timing

Timing & State Diagrams (TSD)

Midnight Blue



* Jason Larsen – Hacking Critical Infrastructure like You're not a n00b – RSA, 2016

Mapping TSD to Devices

Midnight Blue



* Jason Larsen – Hacking Critical Infrastructure like You're not a n00b – RSA, 2016

Mapping Devices to Implants

Midnight Blue

- In order to coordinate all this we will need **implants**
  1. For **executing** OT payloads
  2. For **monitoring** attack progress & **activating** OT payloads

- Carrying out attack at device level via implant has additional benefits
  - **Autonomy** in control zone with own TSD logic
  - **Arbitrary control** over device rather than what's dictated by protocol
  - **More stealthy**: limited network traffic, limited introspection

- Before we can implant a device we have to **exploit** it

Mapping Devices to Implants

Midnight Blue



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

Y(XTREME y
p

Will need to fit implant in there

- Signals processing?
- Malicious logic?
- Comms?

Often jam-packed by normal functionality already



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

Comparing Attack Strategies

Midnight Blue

	Start Conditions	Learning	Control	Spoof	Success Feedback	Failure Check	Control Out	Spoof Out
vlvFermenter	82							
vlvEvaporator	87		86	86			82	
vlvCondensor	93						85	
vlvTransfer	84		97	92				
lvlEvaporator		85		84				87
lvlCondensor		81		86				83
lvlCooling		91		80				85
lvlAlcohol	90			97	97			80
lvlCatch		81		92				93
tmpEvaporator	96	90		89				93
tmpCondensor		86		82	82	82		80
tmpCatch		91		95	96	87		
tmpIgnitor				81				83
optPlateStrikePos				98				81
pmpCondensorOn			85	85			80	
pmpCondensorSpeed		80	85	85			85	84
pmpTransferOn			92	92				
pmpTransferSpeed			87	87				
pmpFountainOn	82		84			84		
pmpFountainSpeed	98	82	93	93		93	81	85
ignitorOn	86					92		
setpntTmpEvaporator		97	86	86				95
setpntTmpCondensor		82	96	96				91
setpntPssTransfer			86					
setpntPssFountain			89	89		89	97	85

Implants	
Implant PLC1	90
Implant PLC2	81

PoolOfDeath	7102
InfernoOfDeath	7020
SteamCollapse	13781

* Jason Larsen – Hacking Critical Infrastructure like You're not a n00b – RSA, 2016

Ah, so that's why everything isn't blowing up all the time …

Midnight Blue

- **This is complicated, expensive stuff**
  - Ton of engineering know-how, RE, vuln research, exploit & implant dev, testing, …
  - High chance of messing up
  - Offsets terrible IT / OT security

- Check out '*Hacking Critical Infrastructure Like You're Not a N00b*' @ RSAConf 2016 by Jason Larsen

- Let's walk through the process required for **developing a single exploit / implant / payload combo** (eg. TRITON)



ICS DEVICE EXPLOITATION

The Process

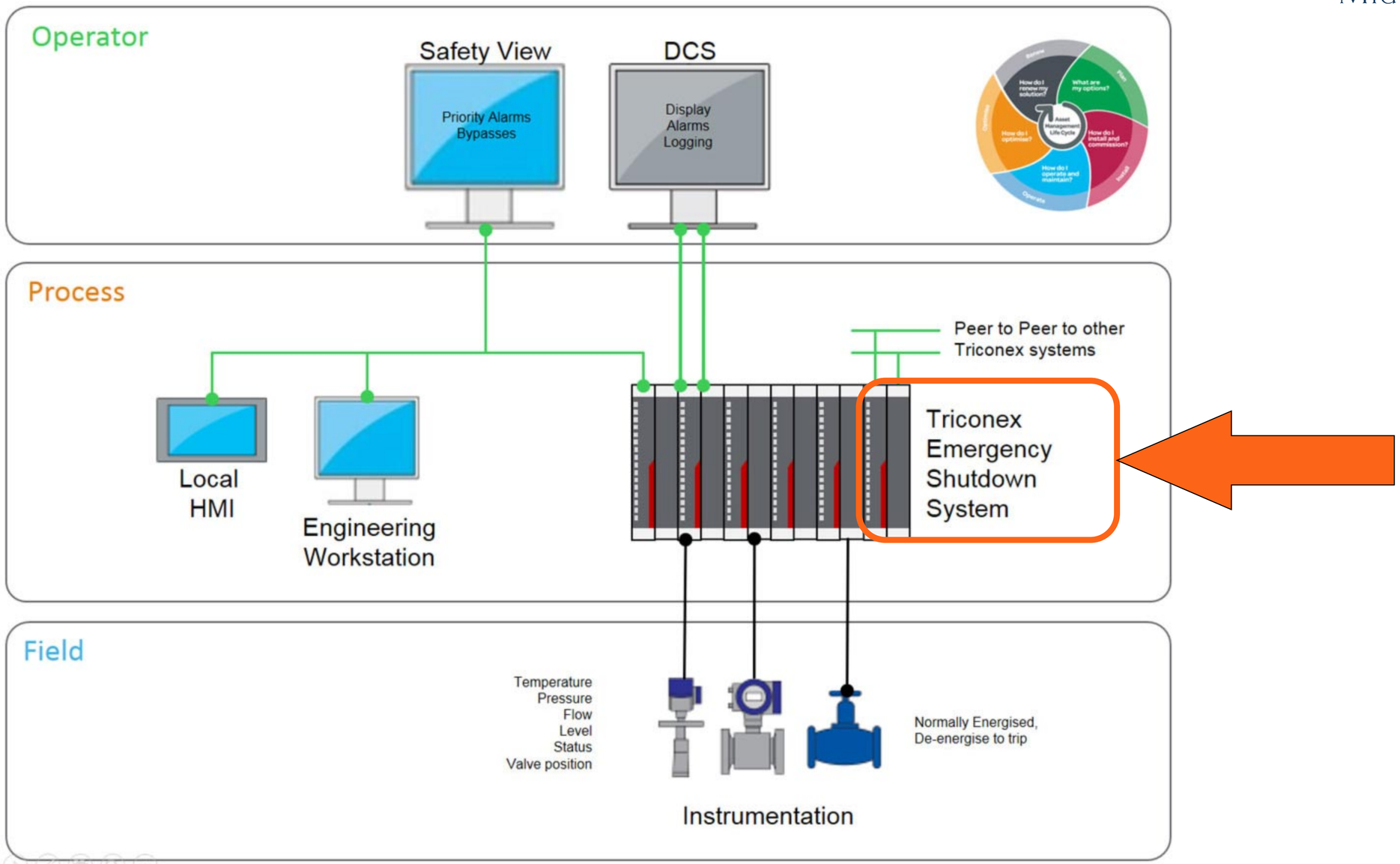
Midnight Blue

1. Obtaining the Necessary Materials
2. Device Teardown & PCB Analysis
3. RE of Engineering Software, Firmware & Protocols
4. Vulnerability Discovery
5. Exploit Development

Obtaining the Necessary Materials

Midnight Blue

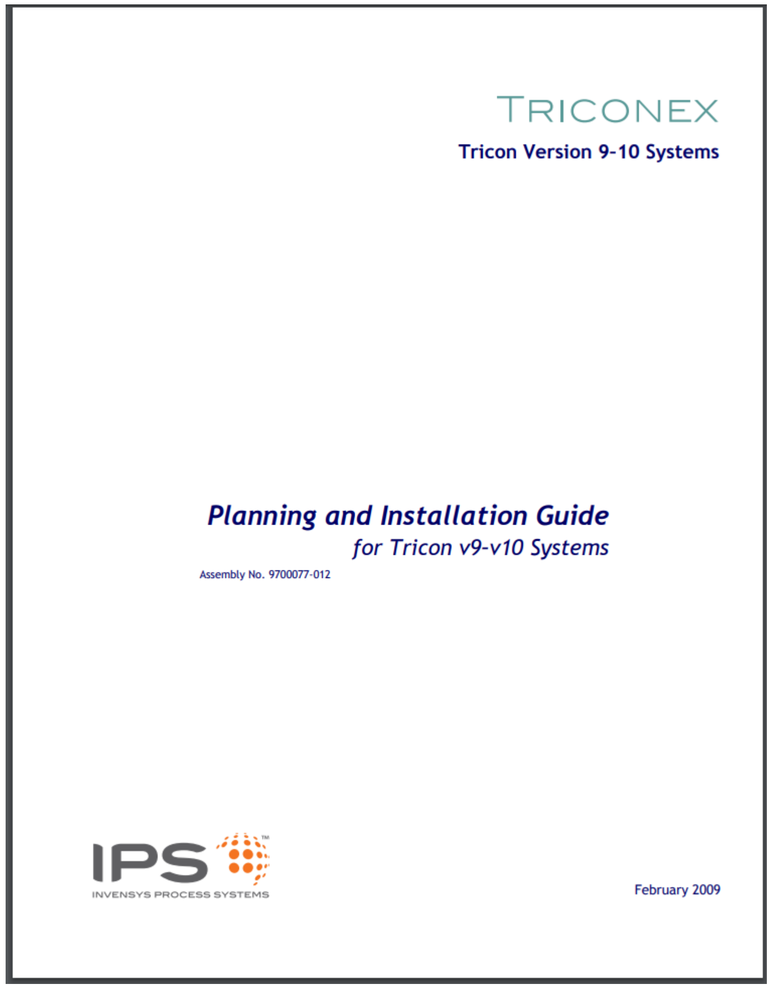
1



* Schneider Electric

Obtaining Documentation

Midnight Blue



TRICONEX

Tricon Version 9-10 Systems

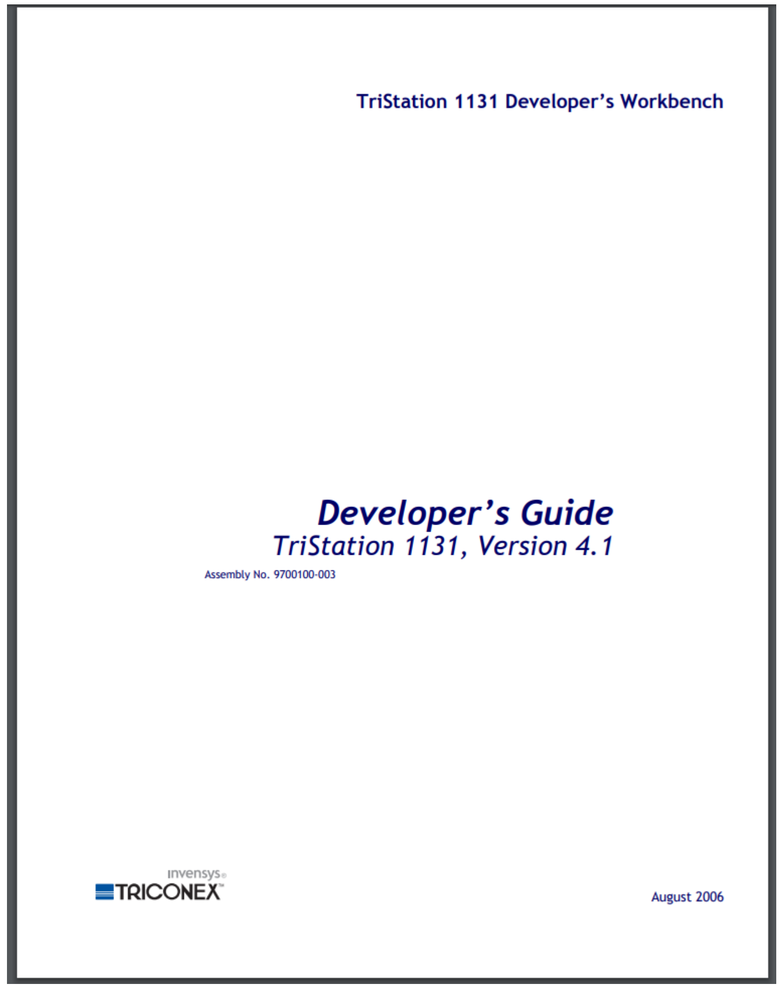
Planning and Installation Guide

for Tricon v9-v10 Systems

Assembly No. 9700077-012

IPS INVENSYS PROCESS SYSTEMS

February 2009



TriStation 1131 Developer's Workbench

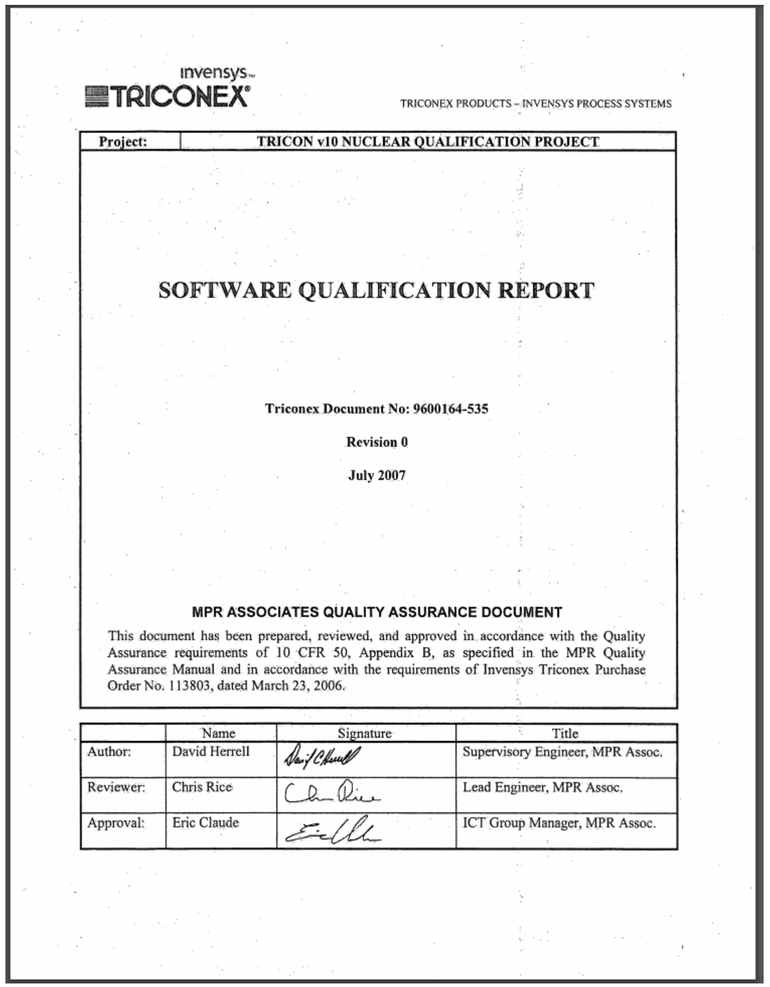
Developer's Guide

TriStation 1131, Version 4.1

Assembly No. 9700100-003

invensys TRICONEX

August 2006



invensys TRICONEX

TRICONEX PRODUCTS – INVENSYS PROCESS SYSTEMS

Project:	TRICON v10 NUCLEAR QUALIFICATION PROJECT

SOFTWARE QUALIFICATION REPORT

Triconex Document No: 9600164-535

Revision 0

July 2007

MPR ASSOCIATES QUALITY ASSURANCE DOCUMENT

This document has been prepared, reviewed, and approved in accordance with the Quality Assurance requirements of 10 CFR 50, Appendix B, as specified in the MPR Quality Assurance Manual and in accordance with the requirements of Invensys Triconex Purchase Order No. 113803, dated March 23, 2006.

	Name	Signature	Title
Author:	David Herrell		Supervisory Engineer, MPR Assoc.
Reviewer:	Chris Rice		Lead Engineer, MPR Assoc.
Approval:	Eric Claude		ICT Group Manager, MPR Assoc.

Obtaining Documentation

Midnight Blue

TRICONEX

Tricon Version 9-10 Systems

Planning and Installation Guide

for Tricon v9-v10 Systems

Assembly No. 9700077-012

IPS INVENSYS PROCESS SYSTEMS

February 2009



invensys

TRICONEX

TRICONEX PRODUCTS – INVENSYS PROCESS SYSTEMS

Project:	TRICON v10 NUCLEAR QUALIFICATION PROJECT

SOFTWARE QUALIFICATION REPORT

Triconex Document No: 9600164-535

Revision 0

July 2007

MPR ASSOCIATES QUALITY ASSURANCE DOCUMENT

document has been prepared, reviewed, and approved in accordance with the Quality urance requirements of 10 CFR 50, Appendix B, as specified in the MPR Quality surance Manual and in accordance with the requirements of Invensys Triconex Purchase der No. 113803, dated March 23, 2006.

	Name	Signature	Title
uthor:	David Herrell		Supervisory Engineer, MPR Assoc.
Reviewer:	Chris Rice		Lead Engineer, MPR Assoc.
Approval:	Eric Claude		ICT Group Manager, MPR Assoc.

Obtaining Engineering Software

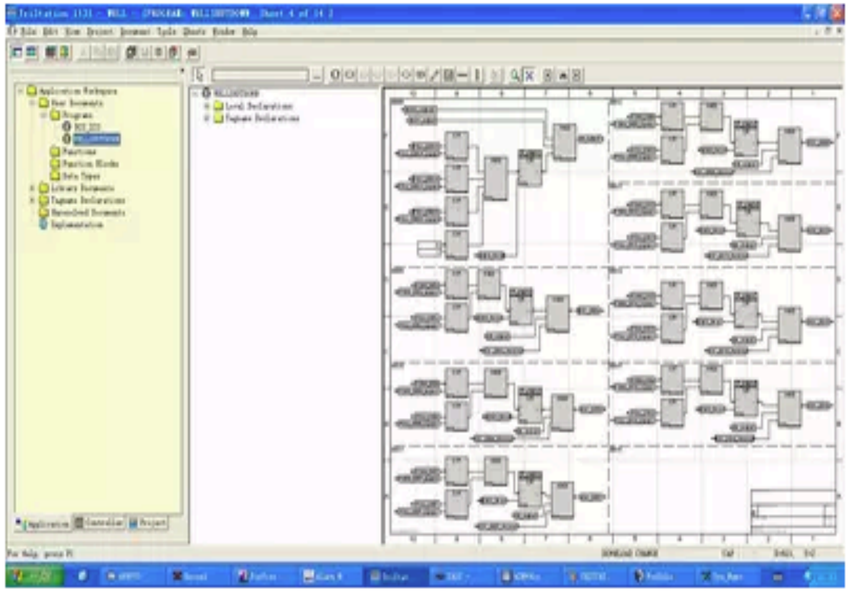
Midnight Blue

- **From vendor website (or by asking nicely)**
- **Through asset owners**
  - If you're already in their IT / OT network might as well grab a souvenir
- **Via sketchy sources on the internet**
  - Installation CDs sold on Ebay or Alibaba
  - Loose executable & archives drifting on the web
  - Open directories, FTP servers, etc.

Obtaining Engineering Software

Midnight Blue

- **From vendor website (or by asking nicely)**
- **Thro**
  - If y
- **Via s**
  - Ins
  - Lo
  - Op





Obtaining the Device

Midnight Blue



TRICONEX 3008 MODULE Used Working %100

Pre-Owned

$1,600.00

or Best Offer

+$60.82 shipping

12 Sold

Customs services a



TRICONEX 3008 MODULE Tricon

Pre-Owned

$1,850.00

or Best Offer

+$122.00 shipping

Free Returns

From Chi



TRICONEX Crop 3008 Main processor MOD

New (Other)

$3,500.00

or Best Offer

+$20.00 shipping

Free Returns

From Chi



2 Photos

TRICONEX Main Processor Module 3008

US $1000.00-$5000.00 / Piece

1 Piece (Min. Order)

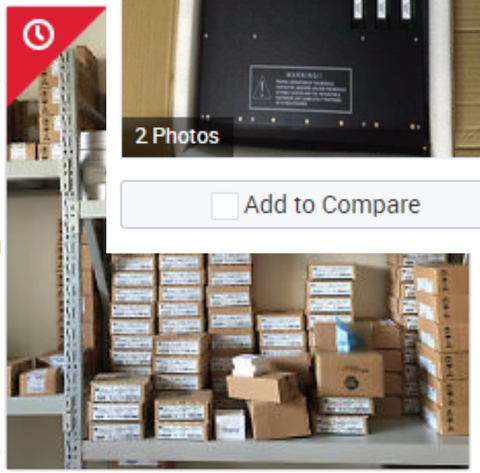
Place of Origin: US

Brand Name: TRICONEX

Model Number: 3008

Add to Compare

Similar Products



Only 1 left!



Triconex 3008 Processor Module - New Surplus Open

New (Other)

$6,298.93

or Best Offer

Free Shipping

See more like this

From Australia

Obtaining Device Firmware



- **Various Options**
  - Download from Vendor Website
  - Extract from Firmware Update Utility
  - Extract from Device Flash

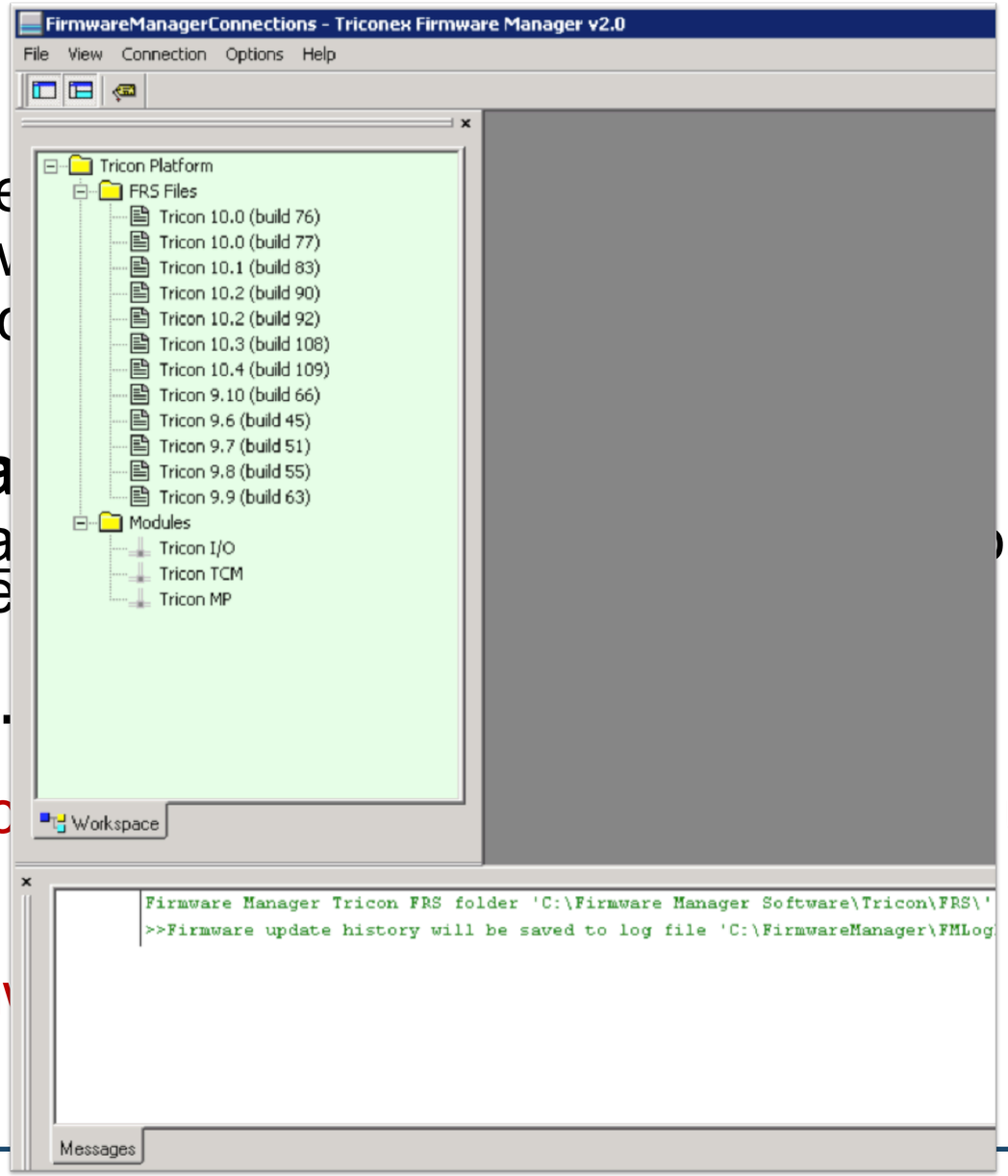
- **Obtaining firmware can be complicated**
  - Worst-case scenario: encrypted firmware + chip readout protection requiring bypass & invasive or side-channel attacks

- **Not so much for Triconex**
  - No readout protection on flash. Desolder -> adapter + universal programmer does the trick

Obtaining Device Firmware

Midnight Blue

- **Various Options**
  - Download from Ve
  - Extract from Firmv
  - Extract from Devic
- **Obtaining firmwa**
  - Worst-case scena ut protection requiring bypass & invasive
- **Not so much for**
  - No readout protec universal programmer does the trick
  - Or get it from firmv



2 Device Teardown & PCB Analysis

Midnight Blue

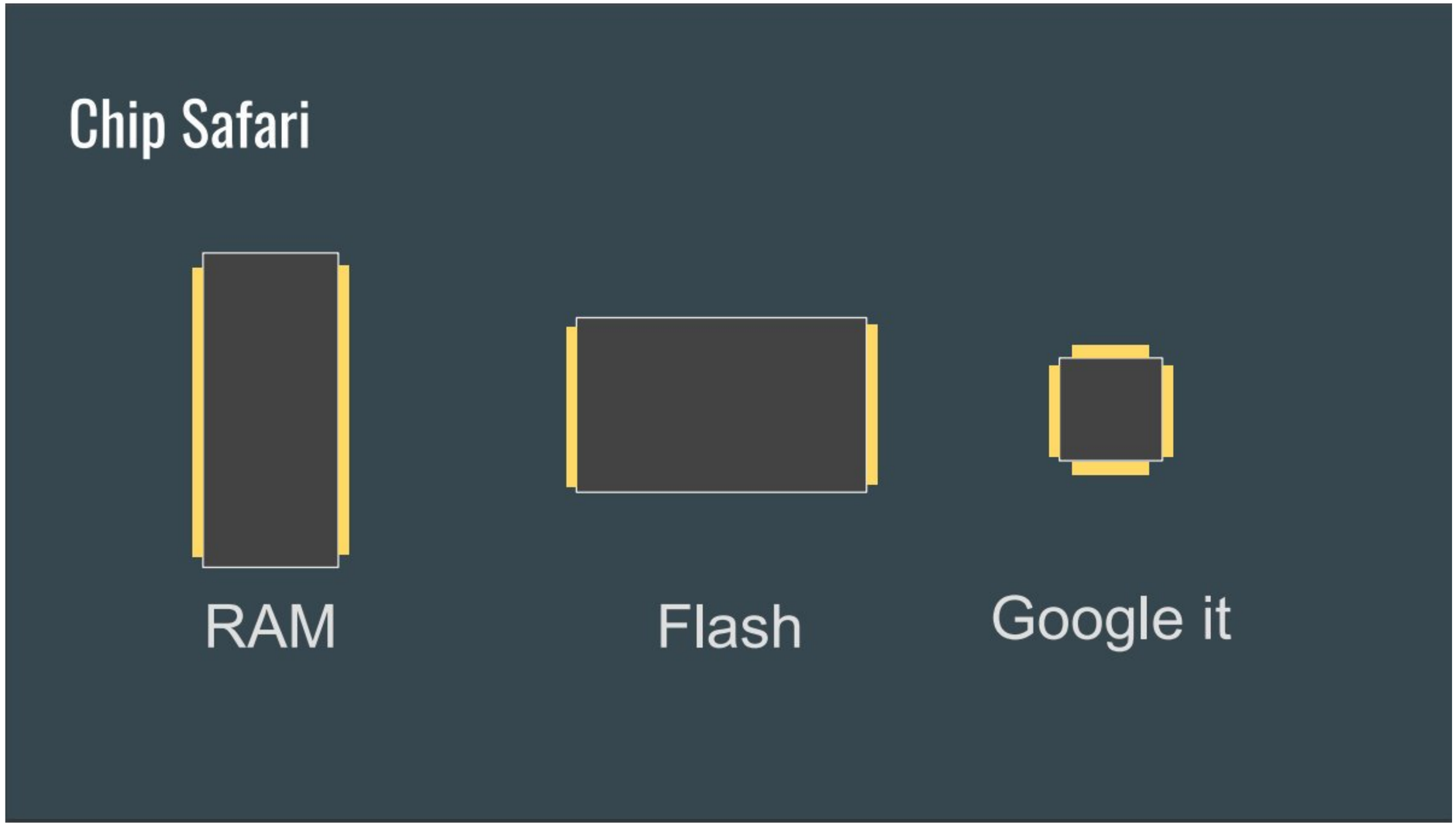
- **We need info on**
  - Microcontroller / SoC used
  - Device Functional Domain Divisions (where does what happen?)
  - Interesting interfaces like UART / JTAG

- **Sometimes we're lucky**
  - FCC IDs, public teardowns, etc.
  - Triconex: Planning & Installation Guide has block diagrams!

- **Sometimes we're not**
  - Teardown time

Don’t be afraid of teardowns

Midnight Blue



* Serge Bazanski, Michal Kowalczyk

ICS Devices are not Magic

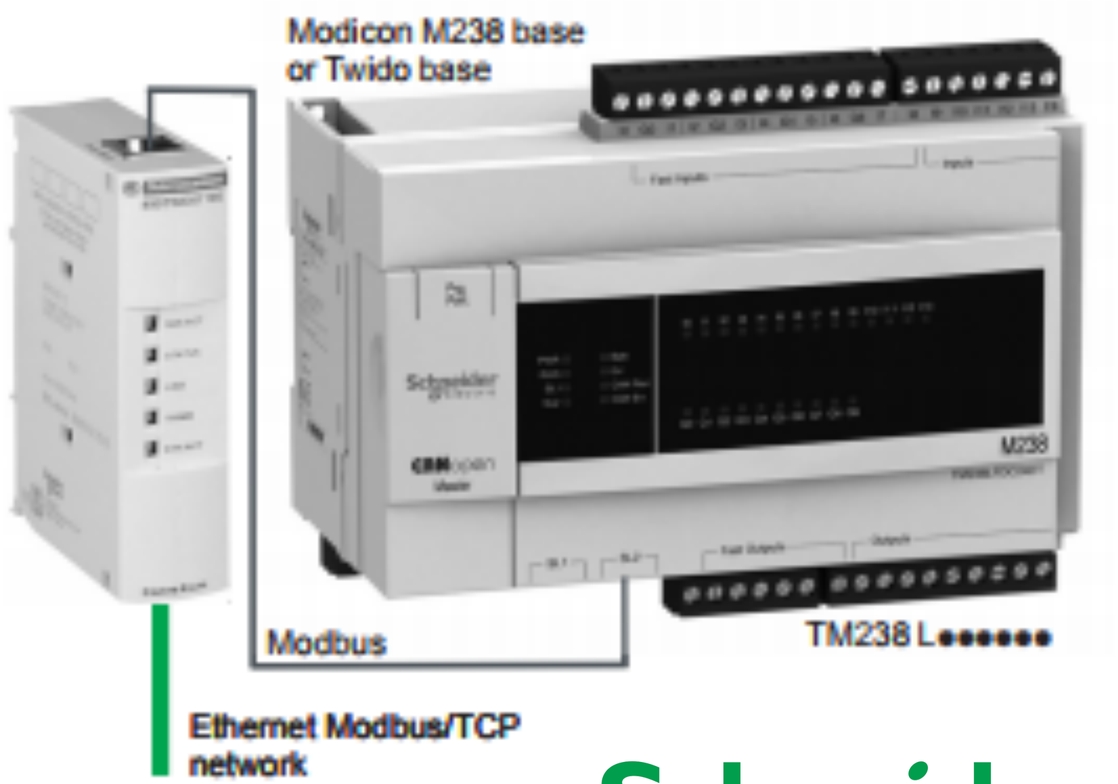
Midnight Blue



* Stephen A. Ridley, Senrio Inc., 2016

Example:
Modicon M238 & M340

Midnight Blue

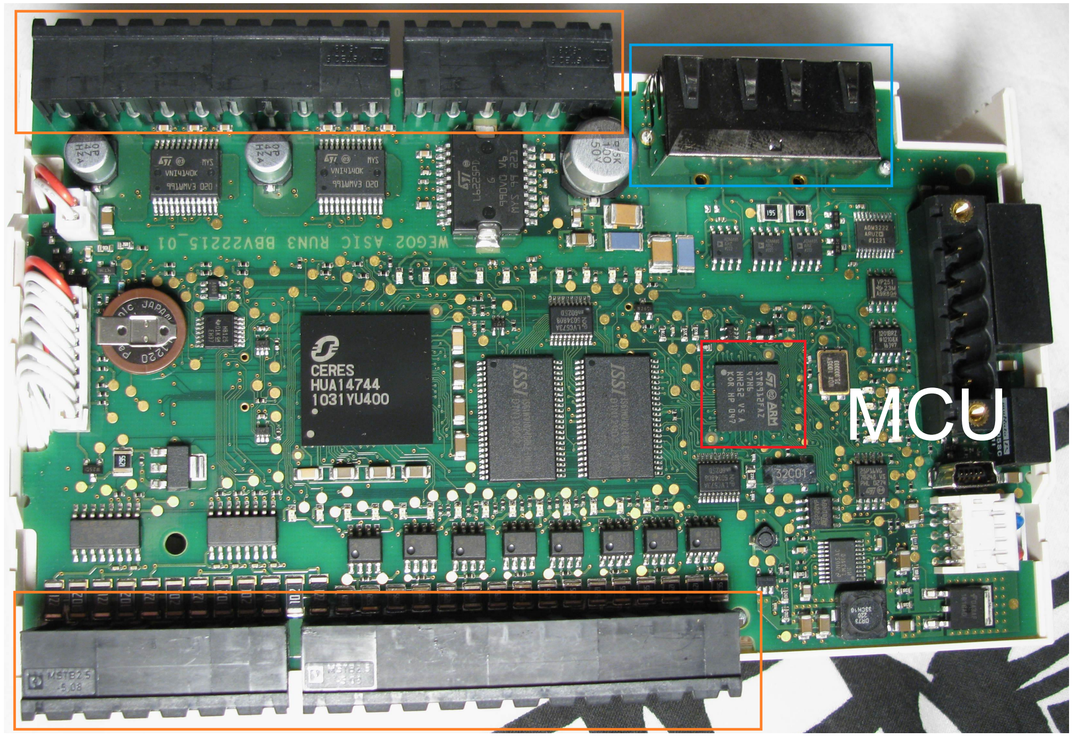


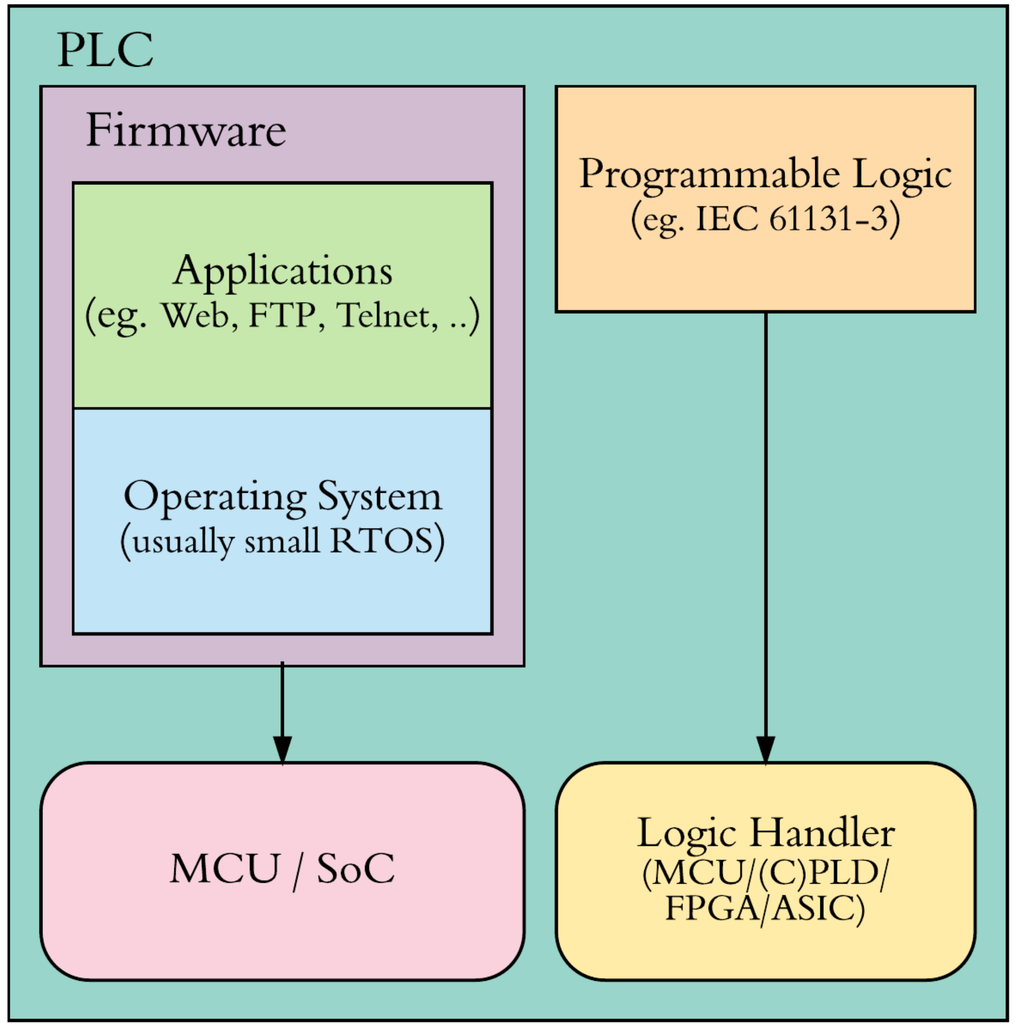
Schneider Electric



Typical (simplified) PLC Architecture

Midnight Blue





Source: www.eevblog.com

Triconex TMR Architecture

Midnight Blue

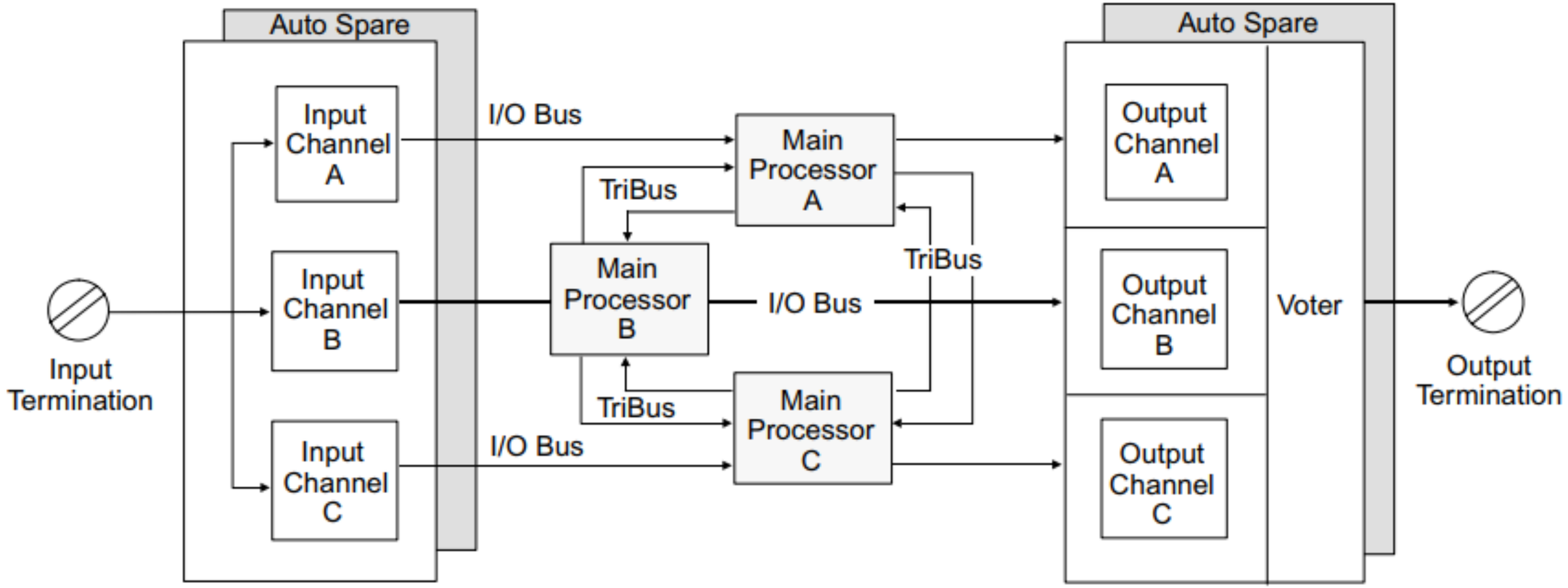


Figure 2 Triplicated Architecture of the Tricon Controller

https://www.nrc.gov/docs/ML0932/ML093290420.pdf

Triconex 3008 MP

Midnight Blue

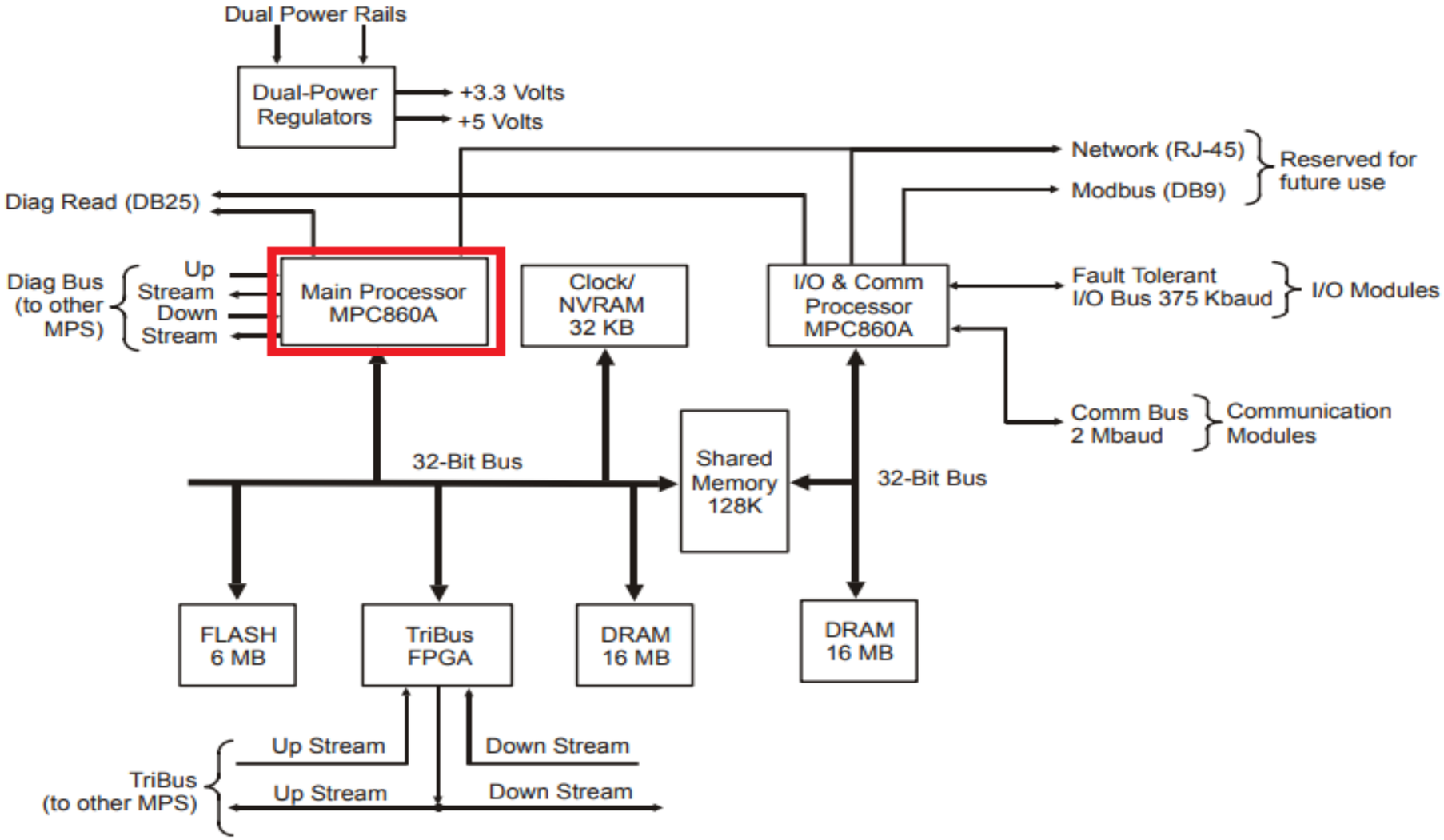


Figure 3 Architecture of a Model 3008 Main Processor

3 Reverse-Engineering Protocols

Midnight Blue

- **One of first areas-of-interest in ICS devices**
- **Often legacy, proprietary protocols**
  - Ports of old serial protocols retrofitted onto Ethernet
- **Control very sensitive functionality**
  - PLC start / stop, firmware update, control logic download, …
- **Might present way into device itself**
  - RCE!
- **We want to know packet structure & semantics**

Reverse-Engineering Protocols



- **Comparison to functionally similar documented protocols**
- **Testing for common encodings & fields**
  - TLV, sequential identifiers, checksums, entropic analysis, …
- **Differential analysis of functional batches of packets**

“*Believe it or not, if you stare at the hex dumps long enough, you start to see the patterns*”

– Rob Savoye, FOSDEM 2009

PCAP-Only Analysis

Midnight Blue



Rob Savoye, FOSDEM 2009

https://s-media-cache-ak0.pinimg.com/originals/1c/28/bb/1c28bbba04e46f4fe517c9b5309f8386.jpg

Ideally we assist analysis with binary RE



- **Want reconstruction to be *complete* & *sound***
  - Want to write ***reliable*** exploits

- **PCAP-Only can be incomplete, inaccurate or opaque**
  - Undocumented / rare behavior, inferred semantics, encryption / compression

- **PCAP-Only can damage your sanity**

tr1com40.dll

Midnight Blue

- TriStation (UDP/1502) Comms DLL. Has all the juicy stuff!
- Debug Symbols Present ☺
- Easy semantic mapping of function codes

CAPLTricon::ReceiveMessage(AplRecvEvents,uchar *,int)	100029E0	96
CTMIAPLTricon::ReceiveMessage(CTMIAPLTricon::TMIAplRecvEvents,uchar *,int)	10007C10	97
CAPLTricon::ResetSequenceNumber(void)	100039A0	98
CTMIAPLTricon::ResetSequenceNumber(void)	10007C00	99
CAPLTricon::Run(void)	10001A80	100
CAPLTricon::SendDnldBlock(Dnld_Block *)	10003A70	101
CAPLTricon::SendDnldEnd(Dnld_End *)	10003B30	102
CAPLTricon::SendDnldGetBlock(int,ulong,ulong)	10003B60	103
CAPLTricon::SendDnldGetModuleInfo(uchar)	10003BD0	104
CAPLTricon::SendDnldGetVersion(int,int)	10003BA0	105
CAPLTricon::SendDnldMessage(int,uchar *,int)	10003CE0	106
CAPLTricon::SendDnldModuleSelect(Dnld_Module_Select *)	10003AE0	107
CAPLTricon::SendDnldMsg(int,int,uchar *,int)	10003A20	108
CAPLTricon::SendDnldReboot(ulong)	10003AB0	109
CAPLTricon::SendDnldRequest(int,int,uchar *,int)	10003C00	110

```
int __thiscall CAPLTricon::GetCpStatusCmd(CAPLTricon *this)
{
  *((_BYTE *)this + 592) = 108;
  return CAPLTricon::SendMessageA(this, 19, 0, 0);
}
```

```
int __thiscall CAPLTricon::Run(CAPLTricon *this)
{
  return CAPLTricon::SendRequest(this, 20u, 0x6Du, 0, 0);
}
```

-HATMAN/blob/master/decompiled_code/library/TS_cnames.py

```
19: 'Get CP status ',
20: 'Run program',
21: 'Halt program',
22: 'Pause program',
23: 'Do single scan',
```

Protocol RE for Attackers

Midnight Blue

- Don’t need full RE, only need to understand a few interesting packet types fully

- Attacker cares about crafting an exploit not a full protocol parser

```
int __thiscall CAPLTricon::SendMessageA(CAPLTricon *this, int function_code,
{
  CAPLTricon *message_obj; // ebx
  char *v5; // eax
  int v6; // eax
  char *v7; // ST10_4
  char *v8; // eax
  int result; // eax
  int v10; // eax
  _BYTE *v11; // ebp
  unsigned __int16 v12; // cx
  int v13; // edx
  int v14; // ecx

  message_obj = this;
  if ( *((_WORD *)this + 274) >= 1u )
  {
    v5 = sub_10006ED0(function_code);
    CAPLTricon::DisplayDebugMessage(message_obj, 1u, aReqS, v5);
  }
  v6 = *((_DWORD *)message_obj + 150);
  ++*((_DWORD *)message_obj + 151);
  if ( v6 == 2 )
  {
    v10 = 0;
    v11 = (char *)message_obj + 1828;
    if ( a4 )
    {
      qmemcpy(v11, a3, a4);
      v10 = 0;
      *((_WORD *)message_obj + 918) = a4;
    }
    else
    {
      *((_WORD *)message_obj + 918) = 10;
    }
    *((_BYTE *)message_obj + 1831) = *((_BYTE *)message_obj + 595);
    v12 = *((_WORD *)message_obj + 918);
    *((_BYTE *)message_obj + 1830) = function_code;
    v13 = v12;
    *((_WORD *)message_obj + 916) = 0;
    *((_WORD *)message_obj + 917) = 0;
```

Vulnerability Discovery

Midnight Blue

- **The next step is getting code exec on the device**
- **Ideally pre-auth vulnerability but**
  - Pre-auth is a relative concept here…
- **ICS Vulns are often byproduct of RE**
  - Insecure by default, ancient legacy shit, …
  - Shake a stick at it & vulns fall out



http://www.fao.org/docrep/006/AD226E/AD226E12.gif

Example:
Moxa NPort W2150A*

Midnight Blue

- **Serial-to-Ethernet/WiFi Converter**
- **Web Interface**
- **Broken auth (hashing on client side)**
- **CMD injection in ping test form**



* Thomas Roth, 2017

Example:
Opto 22 OPTEMU-SNR-DR2*

Midnight Blue

- **Energy Monitoring & Control Device**
- **FTP + OptoMMP (unauth)**
- **Use OptoMMP to**
  - Disable IP filtering, enable FTP, fetch FTP credentials
- **Use FTP to upload firmware & reflash over FTP**
  - No firmware signing



* David Barksdale, Jeremy Brown, 2016

Example: Modicon Quantum PLC*

Midnight Blue

- **Large PLC for process applications**
- **FTP with hardcoded creds**
  - Read / Write configuration, firmware, passwords, …
- **Telnet with hardcoded backdoor**
  - Actually a C interpreter …
- **Unauthenticated Proprietary Modbus Extension**
  - Start / Stop PLC
  - Overwrite programmable logic
  - …
- **Gazillion ways to get code exec**

* K. Reid Wightman, Rubén Santamarta, 2011-2012



Example:
Advantech EKI-1522*

Midnight Blue

- **Serial-to-Ethernet Converter**
- **Web Interface**
- **Unlocking web interface on one PC disables auth for everyone …**
- **CMD injection in email alert setup**



* Thomas Roth, 2017

You get the idea …

Midnight Blue



https://i.redd.it/e5l1ngm7rzr01.jpg

TRITON Vulnerability: Execute My Packet Please!

Midnight Blue

- **Vulnerability is freebie of protocol RE: Safety program download functionality**
  - '*Start Download Change*' (FC: 0x01)
  - '*Allocate Program*' (FC: 0x37)
  - '*End Download Change*' (FC: 0x0B)

- **No authentication**
- **No control program secure signing**

- **Right …**



```
User Datagram Protocol, Src Port: 5812, Dst Port
TriStation PDU
    PDU Type: Command (5)
    Payload Length: 40
  Command
      Dir: 0
      CID: 0
      Command: Allocate program (55)
      Query Count: 190
      Unknown: 0x0000
      Checksum: 0x0973
      Command Length: 40
    Command Parameters
        ID: 3
        Next: 1
        Full Chunks: 5
        Offset: 0
        4 Byte Words Count: 5
        Data: 0xffff603802000044200080 4edfa1288ddc4e57a0
  CRC: 0xf949
```

```
0000  0a 02 02 02 02 02 0a 01  01 01 01 01 08 00 45 00   ........ ......E.
0010  00 4a 12 34 00 00 ff 11  95 6c 0a 00 00 01 0a 00   .J.4.... .l......
0020  00 02 16 b4 05 de 00 36  38 43 05 00 28 00 00 00   .......6 8C..(...
0030  37 be 00 00 73 09 28 00  03 00 01 00 05 00 00 00   7...s.(. ........
0040  05 00 ff ff 60 38 02 00  00 44 20 00 80 4e df a1   ....`8.. .D ..N..
0050  28 8d dc 4e 57 a0 49 f9                           (..NW.I.
```

Skip directly from RE to XDEV: neat!

Exploit Development

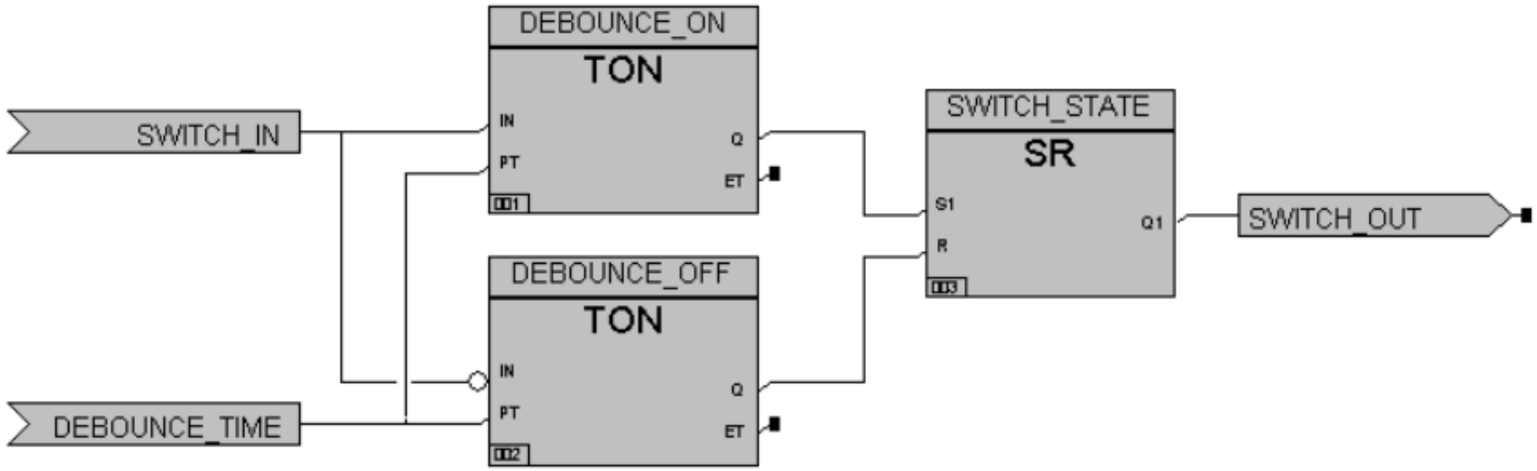
- **After finding a suitable vulnerability, we need to craft an exploit to gain code execution, eg.**
  - Inserting payload directly into unauthenticated firmware upload
  - Use buffer overflow to hijack control-flow to execute payload
  - Etc.

TRITON: How to go from downloading safety program to executing arbitrary code on the microcontroller?

Safety & Control Applications

Midnight Blue

- **Developed in IEC 61131-3 (ST, IL, LD, FBD) and CEMPLE**
  - Get compiled, downloaded & executed on main processor
- **Another freebie: no breaking out of sandboxes, exploiting runtimes or hopping across chip perimeters!**




```
lwz     r3, 0(r2)
lis     r4, 0x40
cmplw   r3, r4
bne     loc_1002C
```

TRITON Code Injection

Midnight Blue

- **TRITON does not overwrite original programming but appends to it**
  - '*Download Changes*' (0x01) instead of '*Download All*' (0x00)

Safety logic continues to run without interruption

Complication: Keyswitch

Midnight Blue



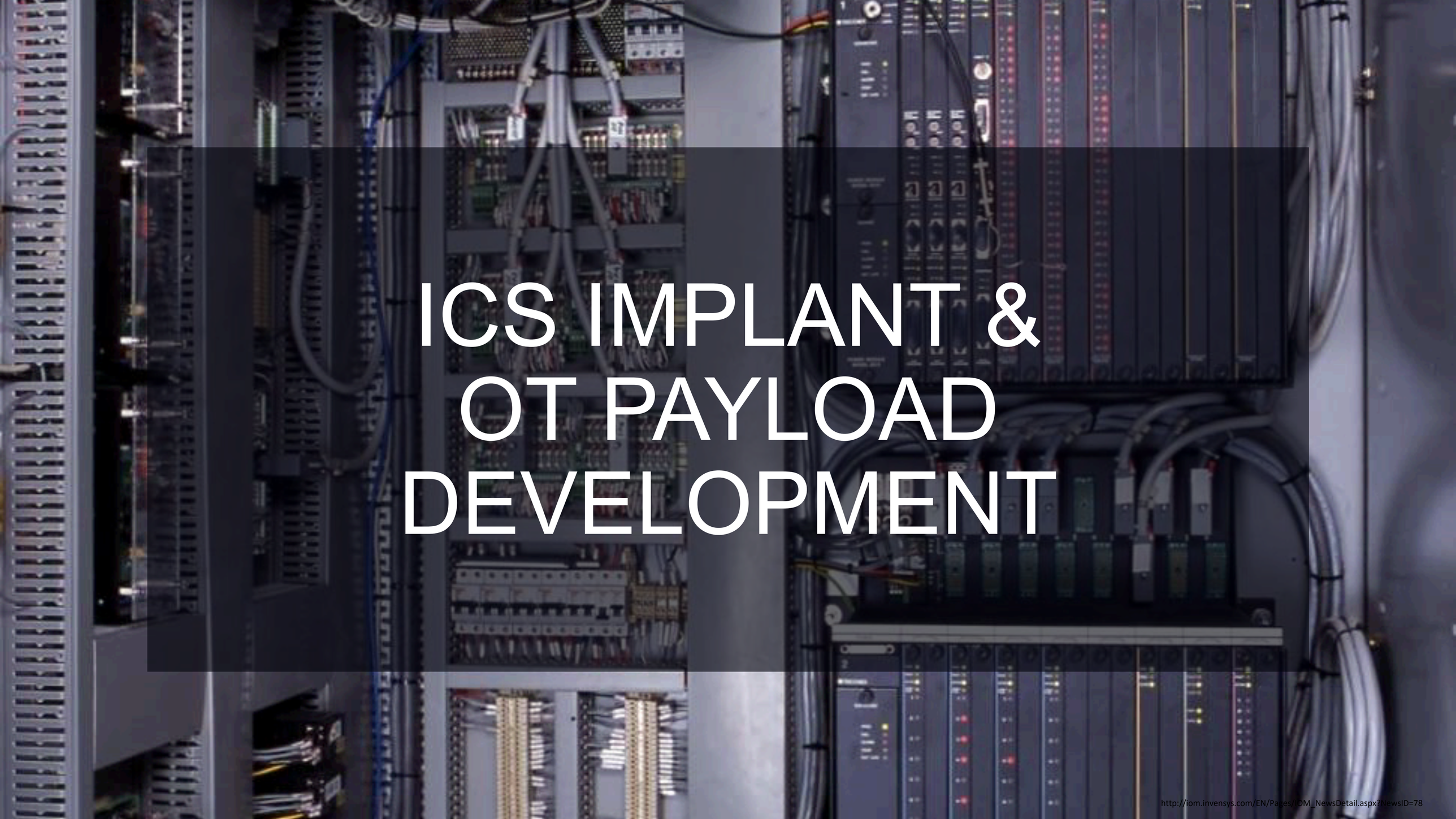
https://images-na.ssl-images-amazon.com/images/I/41jr93jKzML._SX466_.jpg

Complication: Heterogeneity

Midnight Blue

- **Embedded Devices are far more heterogeneous than general purpose ones**
  - Architectures: ARM, MIPS, PPC, 8086, AVR, V850, 68K, …
  - (RT)OSes: VxWorks, ThreadX, QNX, FreeRTOS, MQX, Nucleus, BSD & Linux flavors, custom executives, …
- **Triconex: different architectures & 'OS' per version**
  - Triconex 9 (3006): NS32GX32 + TSX
  - Triconex 10 (3008): MPC860 + ETSX
  - Triconex 11 (3009): Power Architecture e500mc + ETSX

Scaling the attack requires writing / modifying payloads & implants for each version



ICS IMPLANT & OT PAYLOAD DEVELOPMENT

http://iom.invensys.com/EN/Pages/IOM_NewsDetail.aspx?NewsID=78

ICS Implant & OT Payload Development



Alright we can run arbitrary PPC shellcode, now what?

1. Exploitation is just one step among many, for complicated OT payloads we will need to develop an **implant**.

2. After that we will have to craft an **OT payload** to do (part of) the actual damage.

1

ICS Implant Strategies

Midnight Blue

- **Directly implant OT payload or implant backdoor**
  - Keeps OT payload secret until Zero Hour (*'killswitch'*)
- **Cross-Boot Persistence**
  - Requires modifying flash / enough space
- **Memory-Residence**
  - Requires executable RAM
  - Reboot = implant gone (safety controller uptime …)
  - Complicates forensics

1

ICS Implant Scalability

Midnight Blue

- **Common Devices Throughout ICS (cross-facility)**
  - > 18000 Triconex systems in > 80 countries
- **Common Software Throughout ICS (cross-vendor)**
  - Protocol / Connectivity Stacks
  - Control Runtimes / RTOSes
- **Construct arsenal of exploits & implants against common devices & software stacks**
  - One time upfront investment, no huge turnover …

TRITON makes more sense as tool in such an arsenal rather than expensive one-off

Reverse-Engineering ICS Firmware

Midnight Blue

A Extract Firmware

B Preprocess the firmware

C Hunt for interesting functionality



Extracting Firmware

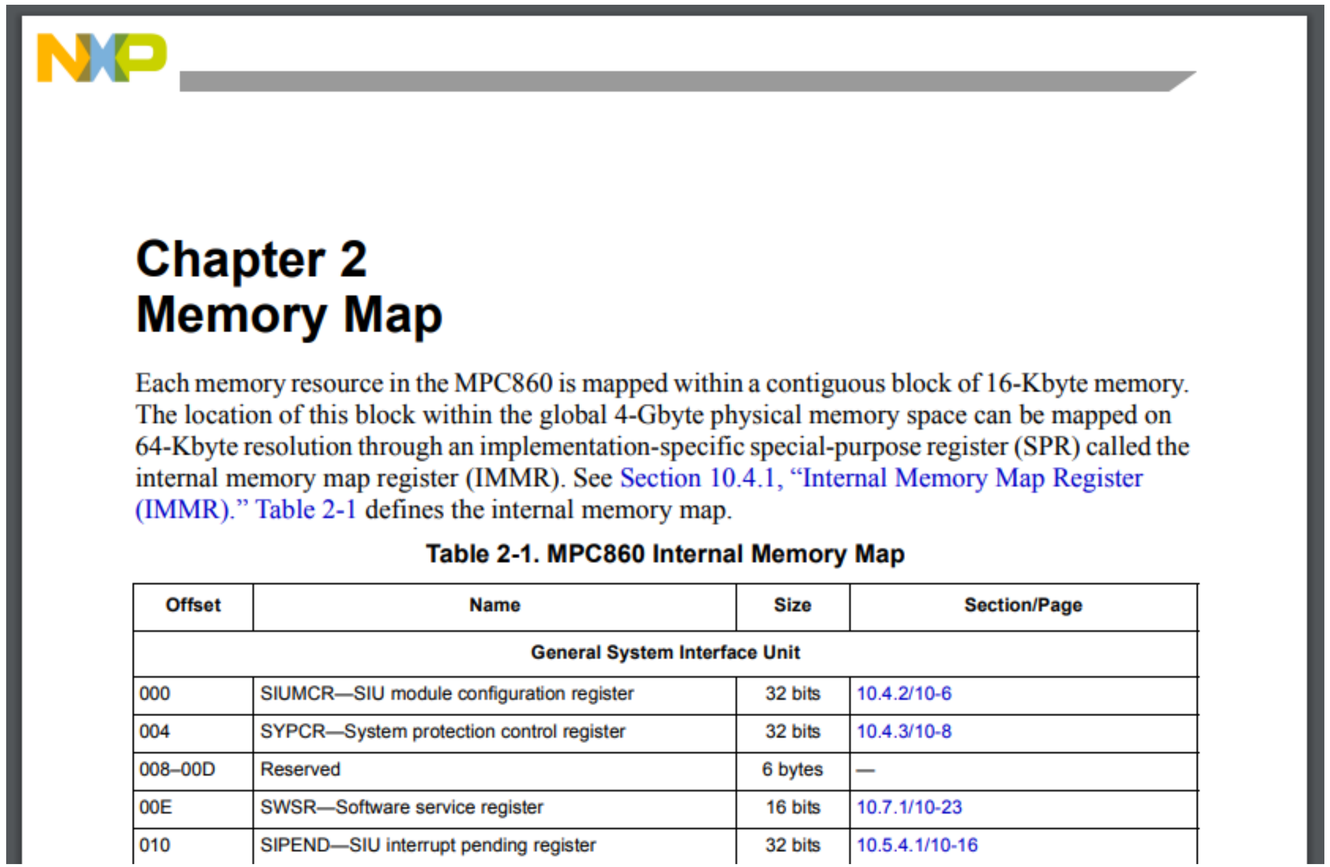
Midnight Blue

- **Determine firmware format & unpack**
  - Sometimes multiple chip firmwares / data blobs are glued together
- **Decompress / Decrypt**
  - Key might be in firmware util, bootloader, side-channel attacks, etc.

Triconex firmware unencrypted

Preprocessing Firmware

- **Obtain the memory map**
  - ROM
  - RAM
  - External Memory
  - Special Purpose Registers
- **Get this from the datasheet**



Chapter 2
Memory Map

Each memory resource in the MPC860 is mapped within a contiguous block of 16-Kbyte memory. The location of this block within the global 4-Gbyte physical memory space can be mapped on 64-Kbyte resolution through an implementation-specific special-purpose register (SPR) called the internal memory map register (IMMR). See Section 10.4.1, "Internal Memory Map Register (IMMR)." Table 2-1 defines the internal memory map.

Table 2-1. MPC860 Internal Memory Map

Offset	Name	Size	Section/Page
	General System Interface Unit		
000	SIUMCR—SIU module configuration register	32 bits	10.4.2/10-6
004	SYPCR—System protection control register	32 bits	10.4.3/10-8
008–00D	Reserved	6 bytes	—
00E	SWSR—Software service register	16 bits	10.7.1/10-23
010	SIPEND—SIU interrupt pending register	32 bits	10.5.4.1/10-16

Learn to love datasheets

Source: https://www.nxp.com/docs/en/reference-manual/MPC860UM.pdf



Identifying Base Address



- **Need to load blob at right address, how to find it?**
  - Many roads lead to Rome…
- **Chip-Fixed Address**
- **RE update utility**
- **Extract from IVT / bootloader**
- **Extract from self-relocating code, jump tables or string tables**

Example: RTU Firmware (ARM)

Midnight Blue

```
ROM:00000000 ; Segment type: Pure code
ROM:00000000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:00000000                 CODE32
ROM:00000000
ROM:00000000 loc_0                                   ; DATA XREF: sub_181E0+F4↓o
ROM:00000000                                         ; sub_3AD78+A8↓o ...
ROM:00000000                 LDR     PC, =0x10000100
ROM:00000004 ; ---------------------------------------------------------------
ROM:00000004
ROM:00000004 loc_4                                   ; DATA XREF: sub_13B28+1E0↓w
ROM:00000004                                         ; sub_13B28+1EC↓r ...
ROM:00000004                 LDR     PC, =0x10000244
ROM:00000008 ; ---------------------------------------------------------------
ROM:00000008
ROM:00000008 loc_8                                   ; DATA XREF: sub_13B28+B0↓r
ROM:00000008                                         ; sub_162FC+94↓r ...
ROM:00000008                 LDR     PC, =0x10000338
```

Example: RTU Firmware (ARM)

Midnight Blue

```
ROM:00000000 ; Segment type: Pure code
ROM:00000000                 AREA ROM, CODE, READWRIT
ROM:00000000                 CODE32
ROM:00000000
ROM:00000000 loc_0
ROM:00000000
ROM:00000000                 LDR      PC, =0x10000100
ROM:00000004 ; ---------------------------------------
ROM:00000004
ROM:00000004 loc_4
ROM:00000004
ROM:00000004                 LDR      PC, =0x10000244
ROM:00000008 ; ---------------------------------------
ROM:00000008
ROM:00000008 loc_8
ROM:00000008
ROM:00000008                 LDR      PC, =0x10000338
```

Address	Exception
0x00000000	Reset
0x00000004	Undefined instruction
0x00000008	Software interrupt
0x0000000C	Abort (prefetch)
0x00000010	Abort (data)
0x00000014	Reserved
0x00000018	IRQ
0x0000001C	FIQ

B Example: RTU Firmware (ARM)

Midnight Blue

```
ROM:00000100
ROM:00000104
ROM:00000108
ROM:0000010C
ROM:00000110
ROM:00000114
```

```
LDR     R0, =0xB000DD00
SUB     R0, R0, #0xC00
MOV     R3, #0xD7
MSR     CPSR_cxsf, R3
MOV     SP, R0
MOV     LR, #0
```

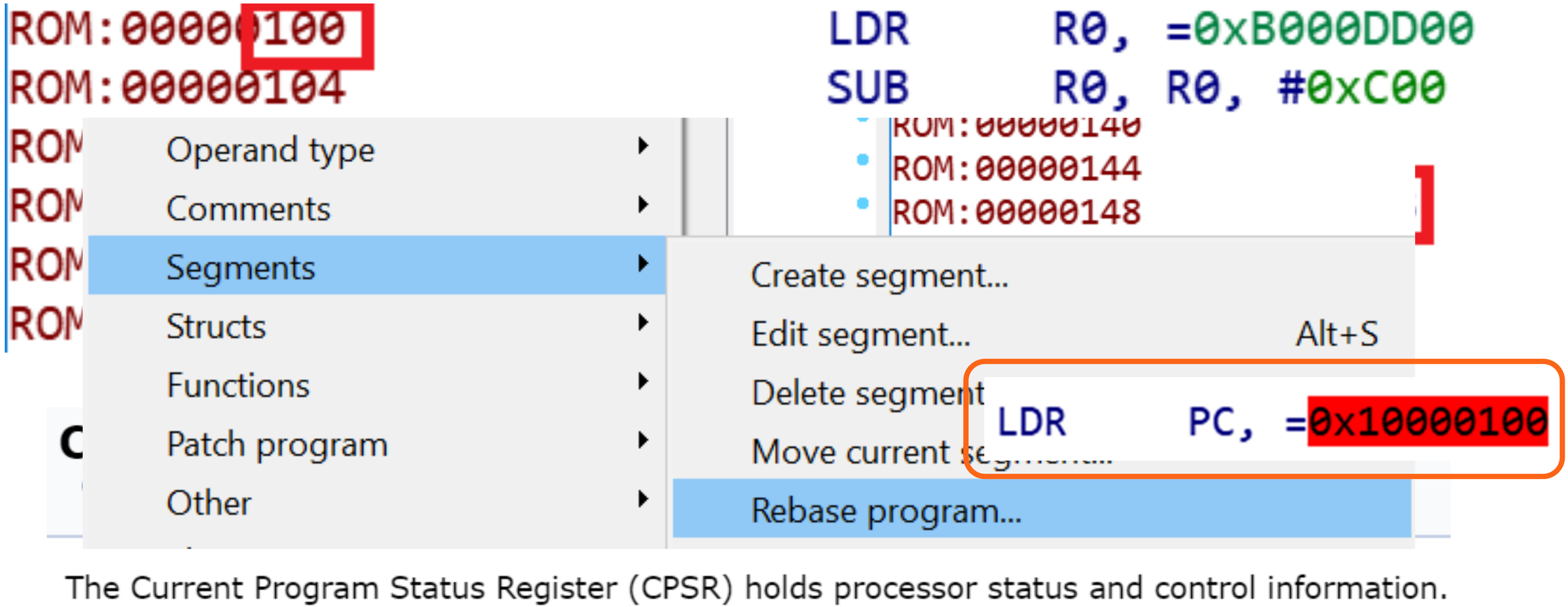
Current Program Status Register (CPSR)

Core Register Access

The Current Program Status Register (CPSR) holds processor status and control information.

Source: https://www.keil.com/pack/doc/CMSIS/Core_A/html/group__CMSIS__CPSR.html#details

Example: RTU Firmware (ARM)



The Current Program Status Register (CPSR) holds processor status and control information.

Source: https://www.keil.com/pack/doc/CMSIS/Core_A/html/group__CMSIS__CPSR.html#details

Example: RTU Firmware (ARM)

Midnight Blue

```
ROM:10000000 ; ===========================================================================
ROM:10000000
ROM:10000000 ; Segment type: Pure code
ROM:10000000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:10000000                 ; ORG 0x10000000
ROM:10000000                 CODE32
ROM:10000000
ROM:10000000 loc_10000000                            ; DATA XREF: sub_100029E0+8↓o
ROM:10000000                                         ; ROM:off_10002A68↓o ...
ROM:10000000                 LDR     PC, =sub_10000100
ROM:10000004 ; ---------------------------------------------------------------------------
ROM:10000004                 LDR     PC, =sub_10000244
ROM:10000004                                         ; DATA XREF: sub_1014A4C4+554↓r
ROM:10000008 ; ---------------------------------------------------------------------------
ROM:10000008                 LDR     PC, =sub_10000338
```



Reconstructing Code & Data Topology

Midnight Blue

- Firmware images usually not neat executable formats (PE, ELF, Mach-O)

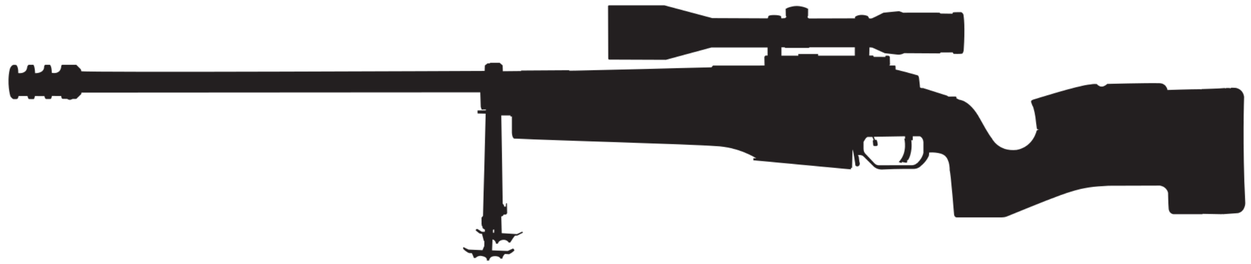
- Will have to heuristically identify functions, strings, jump tables, structs, etc.

Upside: you don’t need to fully RE firmware, only up until readiness for next step

Hunt for Interesting Functionality

Midnight Blue

- **Want to RE in sniper fashion**



- **Control runtime**
- **Protocol parsers**
- **Comms & Peripheral IO handlers**
- **Security / Safety / Sanity-Checking Functionality**

Reverse-Engineering ICS Firmware

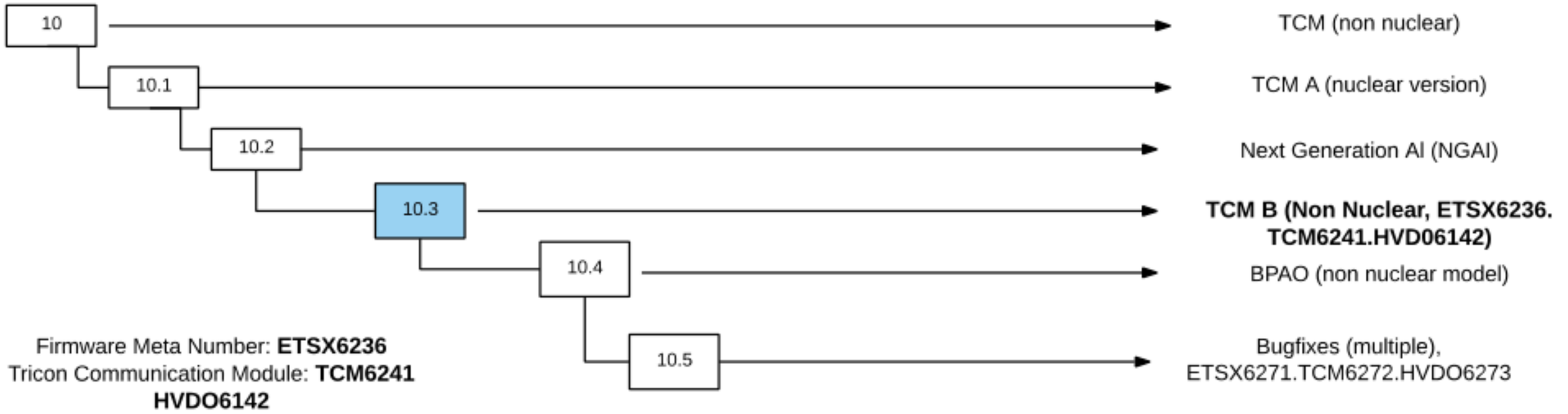


What does this look like for Triconex 3008?

Triconex 3008 V10.3 MP Firmware

Midnight Blue

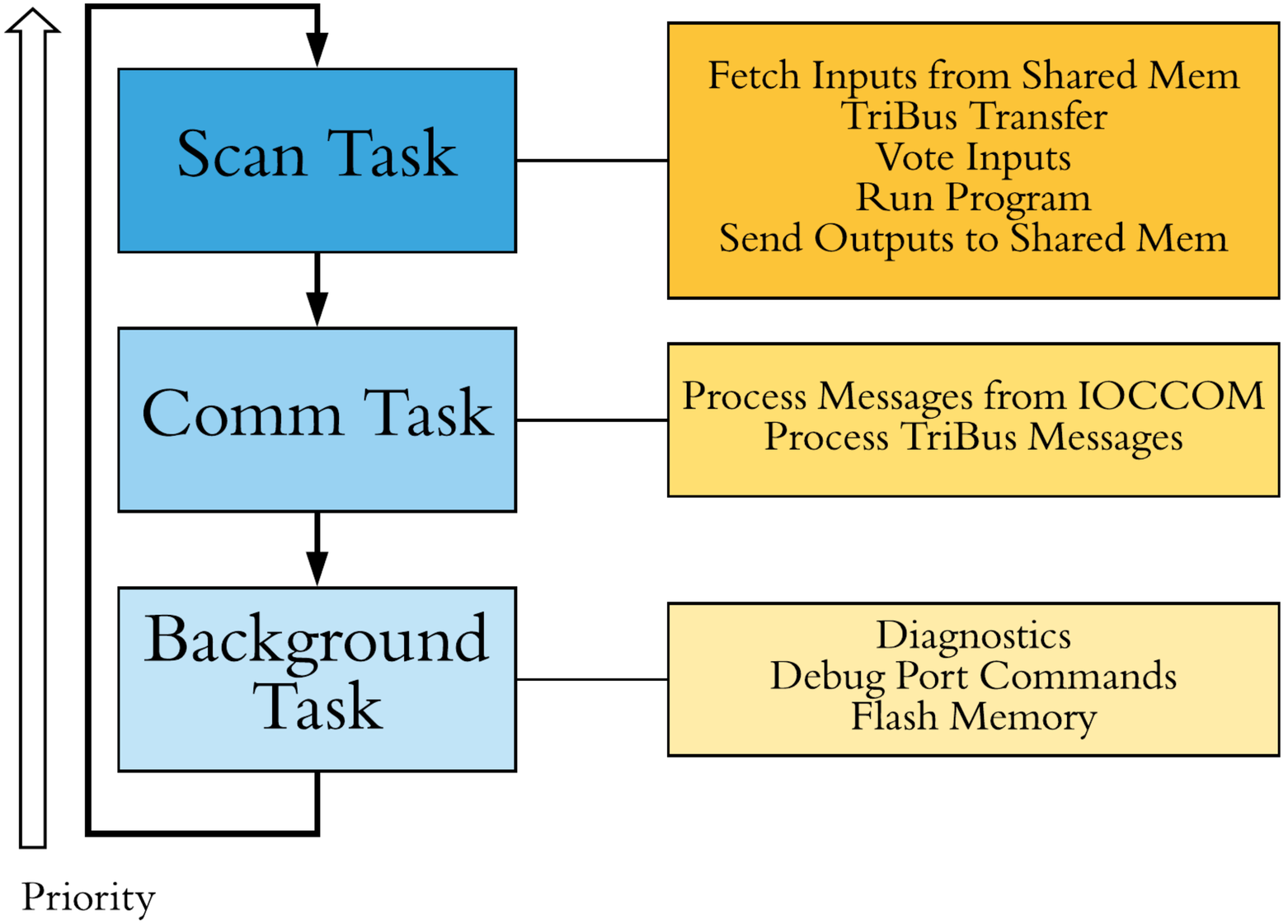
- PowerPC: nice, Hex-Rays decompiler available
  - Not substitute for reading disasm but eases navigation
- Uses ETSX 6236, tiny custom OS with 27 syscalls.
- Some sparse documentation exists (NRC)



Source: United States Nuclear Regulatory Commission , Document number NTX-SER-09-10, Page 96

Enhanced Tricon System Executive (ETSX)

Midnight Blue



Triconex Firmware RE Targets

Midnight Blue

- **Memory Layout & Management**
  - For memory residence
- **Consistency Checks & Diagnostics Functionality**
  - For implant stability
- **Network Command Dispatchers Functionality**
  - For implant comms
- **Privilege Mode Management**
  - For privilege escalation
- **(possibly) Scan Task I/O Transfers + Safety Program Storage / Running**
  - For OT payload

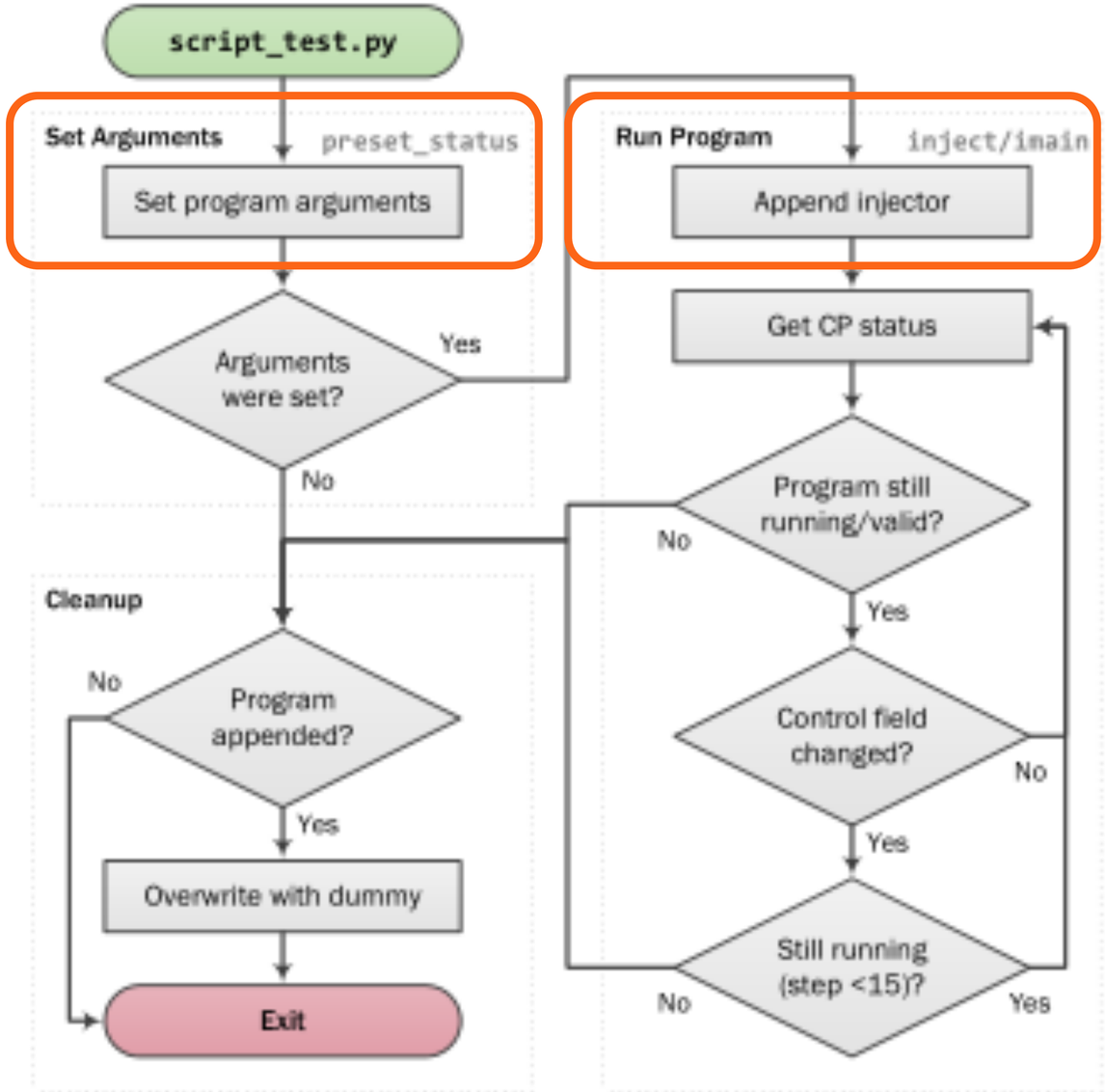


THE TRITON IMPLANT

```
loc_5F4:                                # CODE XREF: sub_3E4+1E4↑j
        lis     r27, 0x80               # Load Immediate Shifted
        ori     r27, r27, 0x6C # 0x80006C # OR Immediate
        li      r0, 0xBF                # Load Immediate
        stw     r0, 0(r27)              # Store Word
        mr      r3, r25                 # Move Register
        addi    r4, r28, 8              # Add Immediate
        lwz     r5, 4(r28)              # Load Word and Zero
        bl      copy_payload_into_fw # Branch
        lis     r27, 0x80               # Load Immediate Shifted
        ori     r27, r27, 0x6C # 0x80006C # OR Immediate
        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
```

TRITON Multi-Stage Payload

Midnight Blue



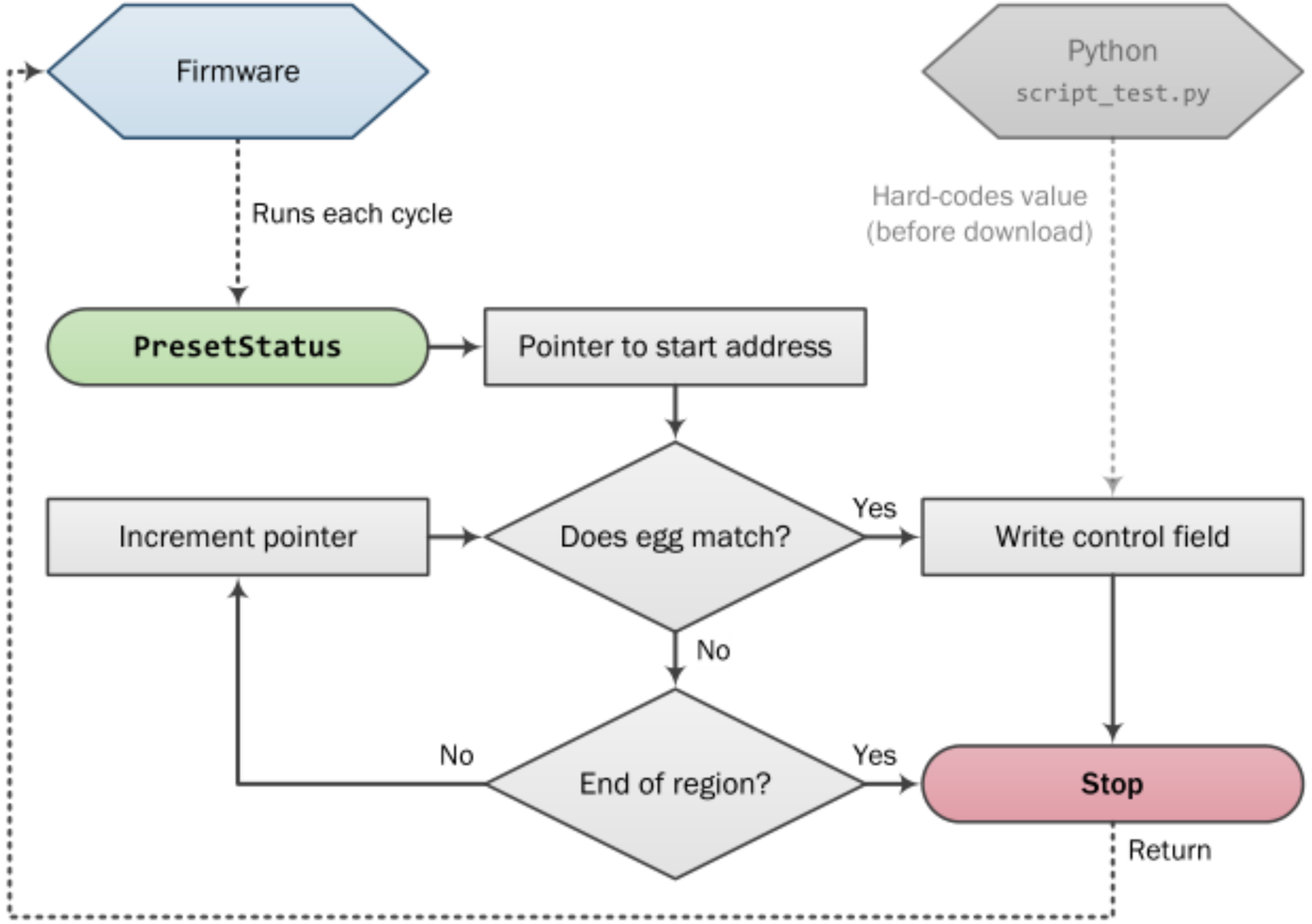
- Stage 1: Argument-Setter
- Stage 2: Implant Installer (**inject.bin**)
- Stage 3: Backdoor Implant (**imain.bin**)
- Stage 4: Missing OT Payload

* ICS-CERT Report on HATMAN

Payload Stage 1: Argument-Setter

Midnight Blue

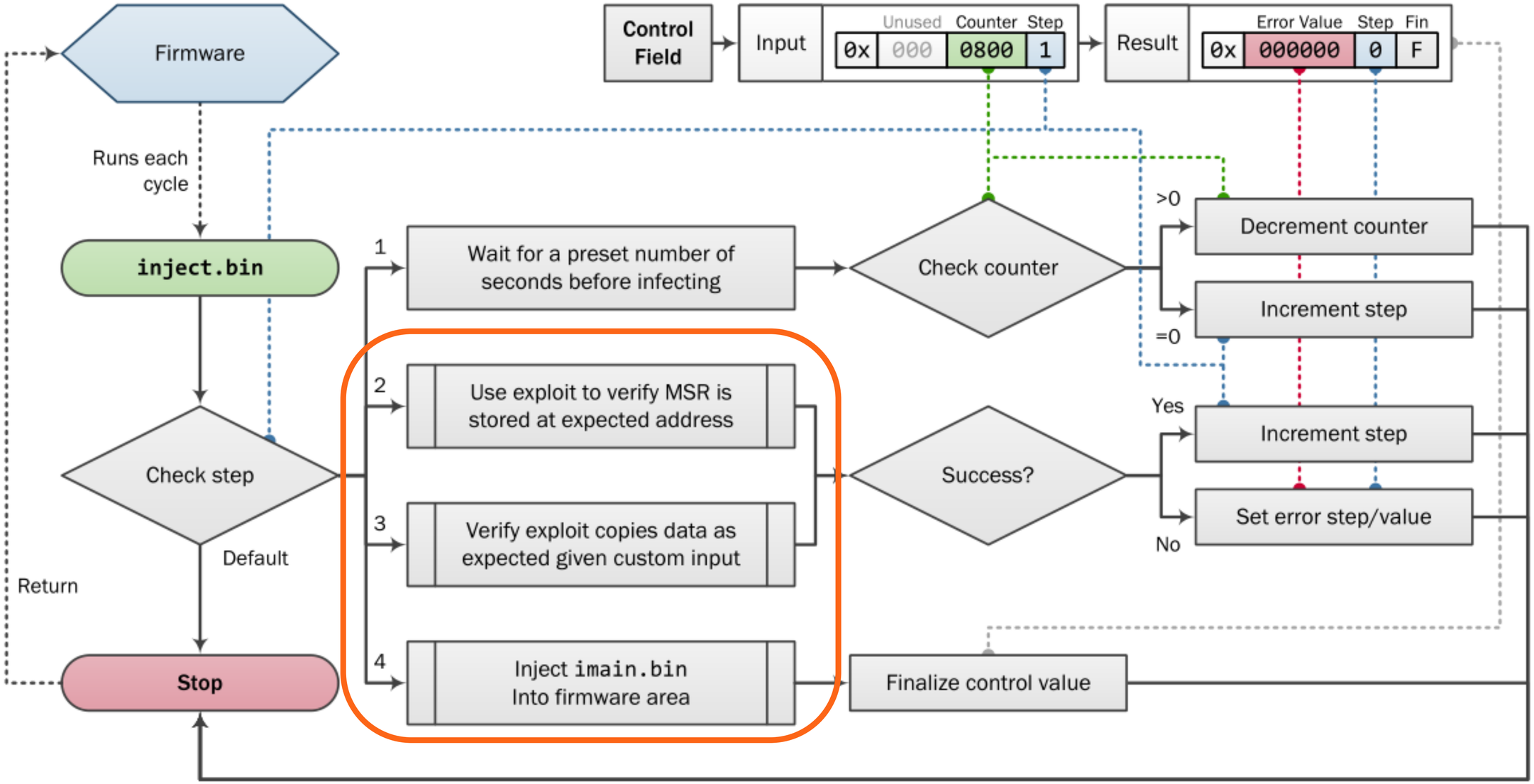
- Egghunt for Control Program (CP) *fstat* field, sanity tests writing, uses it for stage 2 FSM control



Source: ICS-CERT MAR-17-352-01 HatMan—Safety System Targeted Malware (Update A)

Payload Stage 2: Full FSM

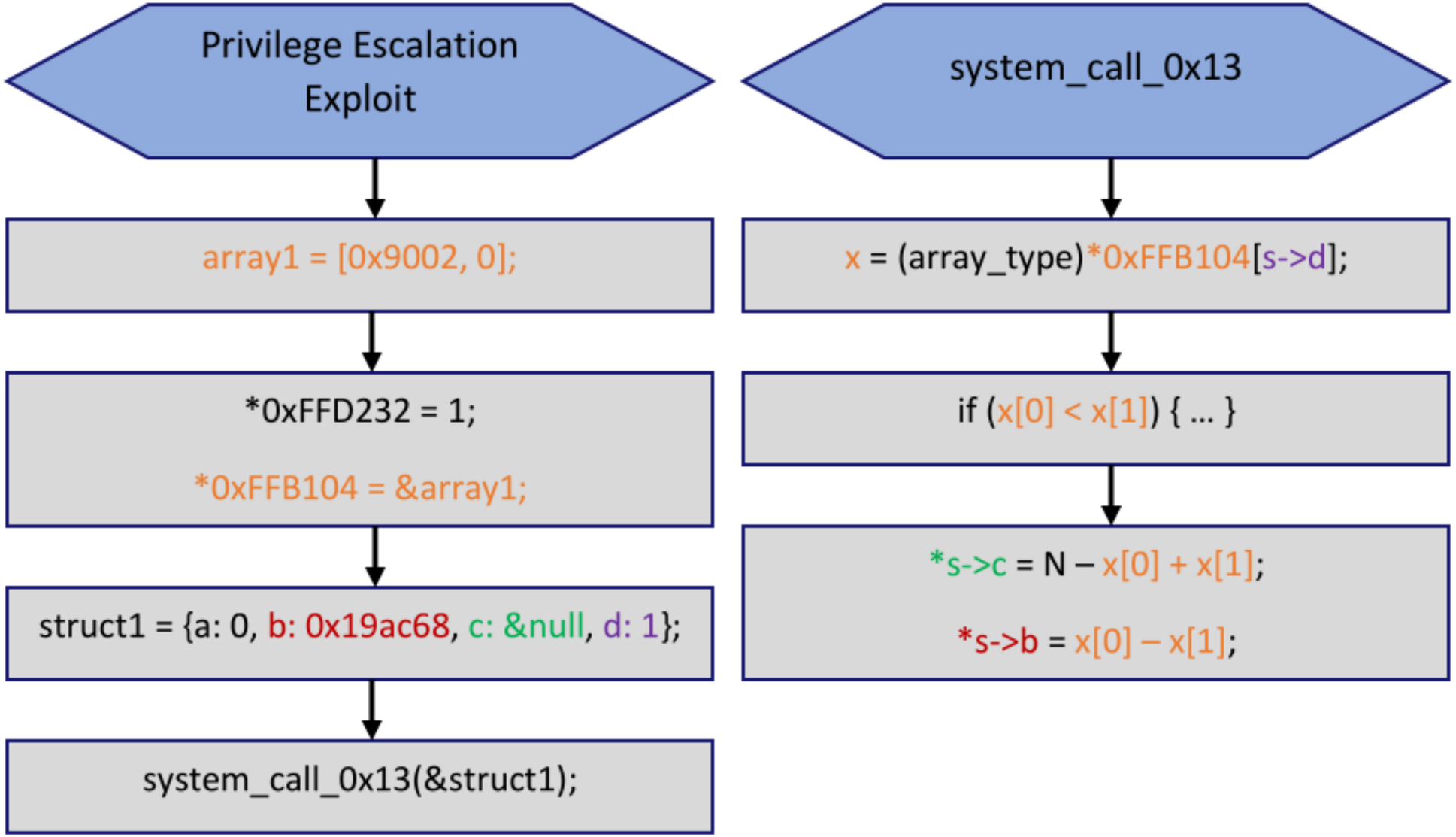
Midnight Blue



* ICS-CERT Report on HATMAN

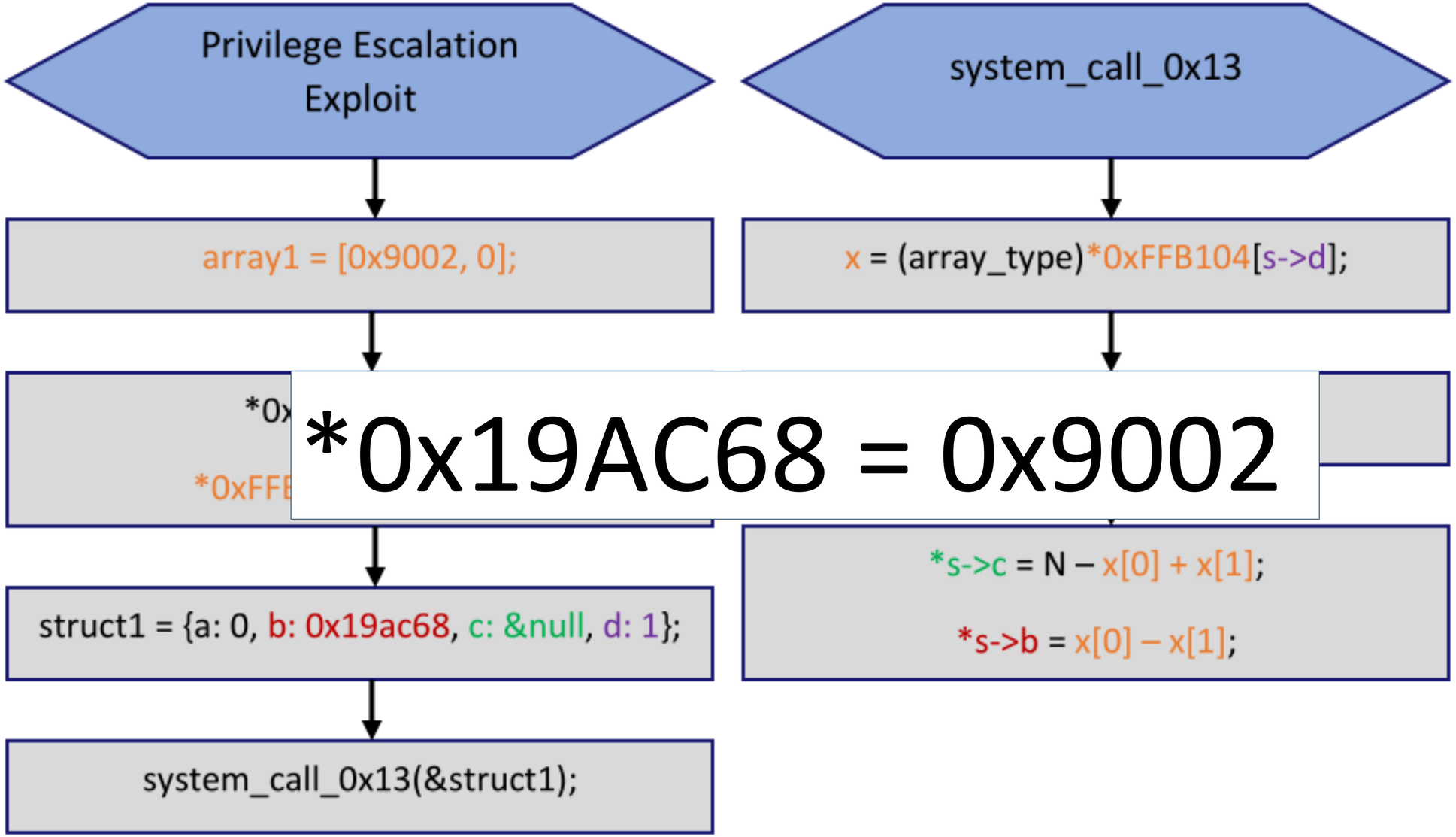
Payload Stage 2: PrivEsc Exploit

Midnight Blue



Payload Stage 2: PrivEsc Exploit

Midnight Blue



Payload Stage 2: PrivEsc Exploit

Midnight Blue

- **What's happening here? What lives at 0x19AC68?**
  - ETSX syscall invocation (re)stores SRR1 here (saved MSR)

```
mfctr     r30
mfxer     r31
stw       r26, (dword_19AC60 - 0x19ABDC)(r2)
stw       r27, (dword_19AC64 - 0x19ABDC)(r2)
stw       r28, (dword_19AC68 - 0x19ABDC)(r2)
```

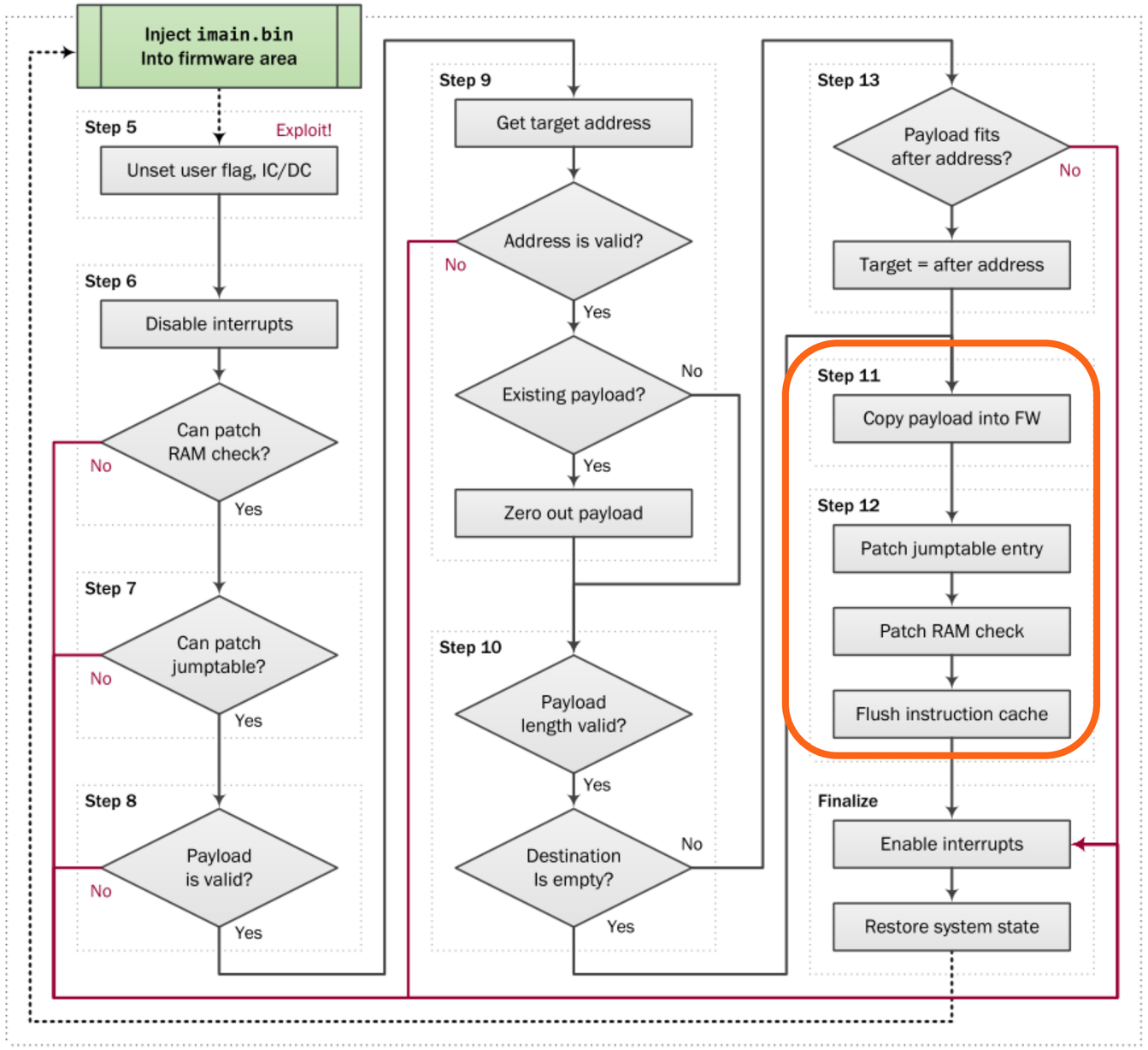
- **Why overwrite MSR with 0x9002?**
  - Bit 17 of MSR will be 0

17	PR[1]	Privilege level 0 The processor can execute both user- and supervisor-level instructions. 1 The processor can only execute user-level instructions.

Source: https://www.nxp.com/docs/en/reference-manual/MPC860UM.pdf

Payload Stage 2: Implant Installer

Midnight Blue

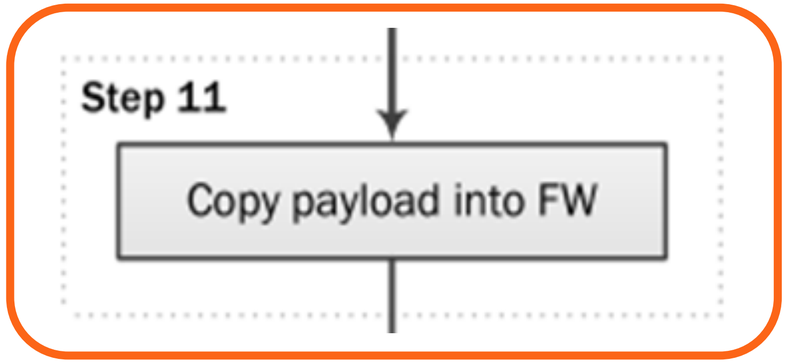


Requires Supervisor Privileges

* ICS-CERT Report on HATMAN

Payload Stage 2: Relocate Backdoor

Midnight Blue



Ensures Residence Even with full program wipe

```
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
```

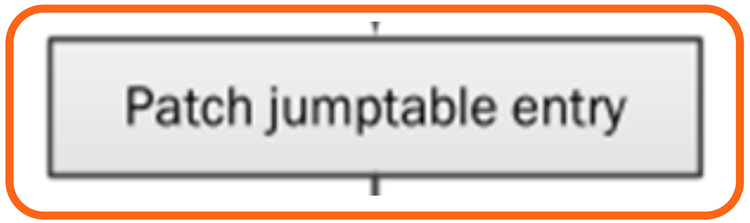
```
                .byte   0
                .byte   0
                .byte   0
                .byte   0
                .byte   0
# ---------------------------
# START OF FUNCTION CHUNK FOR su

imain_bin_start_reloc:

                bl      sub_9F
```

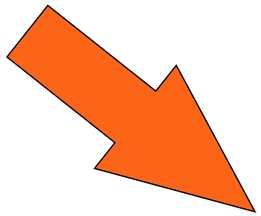
Payload Stage 2: Install TriStation Handler

Midnight Blue




```
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
```

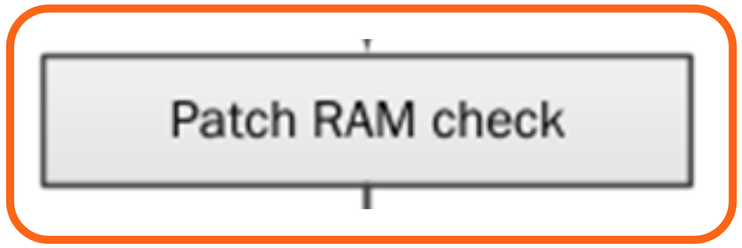
- Entry 0x1D (Get MP Status)
- Allows for network comms



```
default_handler, imain_bin_start_reloc, default_handler, d
CB8, loc_39C88, loc_39CE8, loc_39E38, loc_39D78, loc_39D78,
```

Payload Stage 2: Disable Diagnostics

Midnight Blue



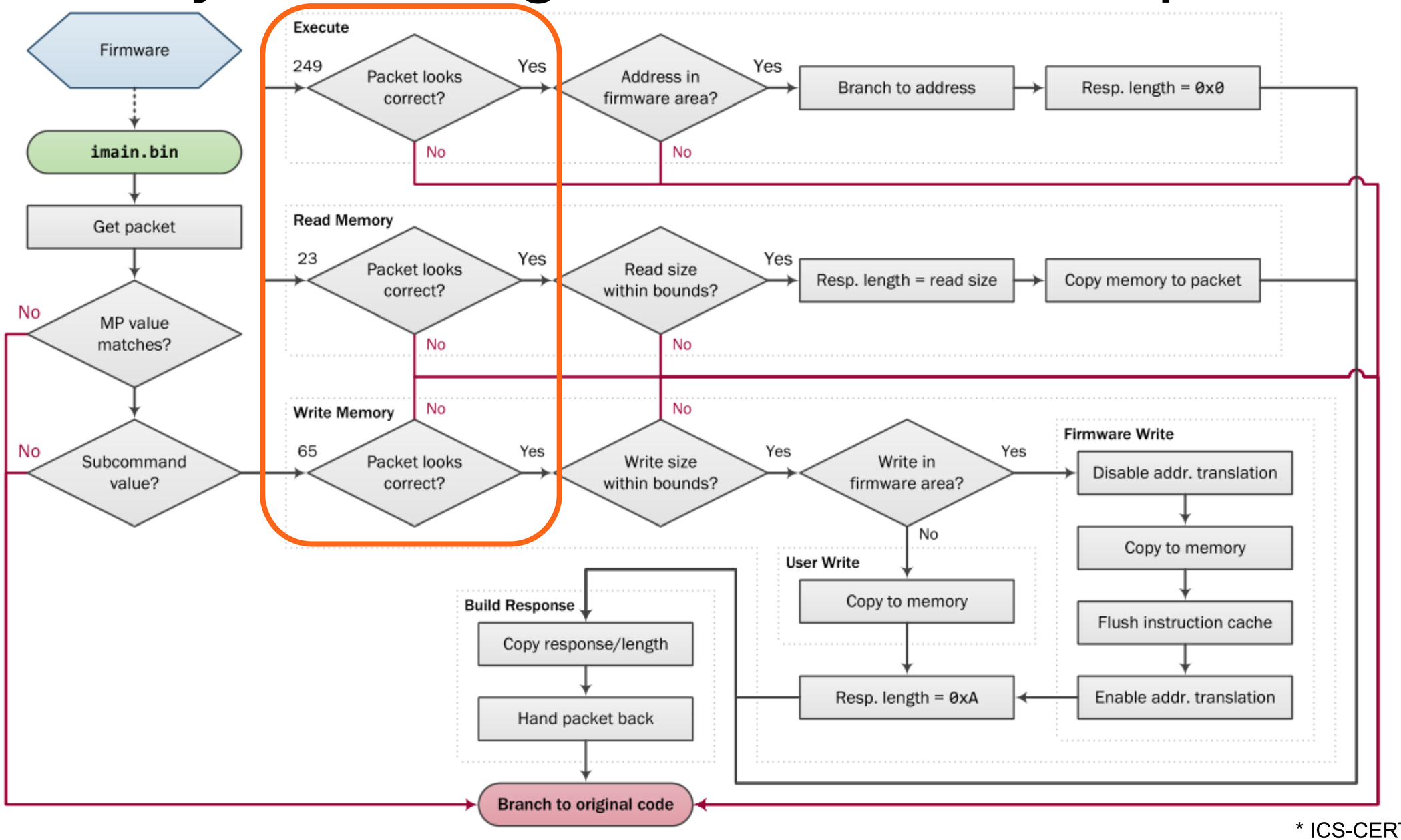

```
                bge         loc_57EC
                lwz         r4, (dword_1D0890 - 0x1D0890)(r30)
                li          r5, 0x100
                bl          sub_611DC
                cmplwi      r3, 0
                b           jump_over
# ---------------------------------------------------------------------------
                lwz         r4, 0(r29)
                lis         r3, aRamRomMismatch@ha
                lwz         r5, 0(r30)
                addi        r3, r3, aRamRomMismatch@l # "Ram Rom Mismatch Rom(%x) Ram(%x)\r\n"
                crclr       4*cr1+eq
                bl          sub_567BC
                li          r31, -1

jump_over:                              # CODE XREF: sub_5750+70↑j
                lwz         r7, (dword_1D0890 - 0x1D0890)(r30)
```

Originally conditional branch

Payload Stage 3: Backdoor Implant

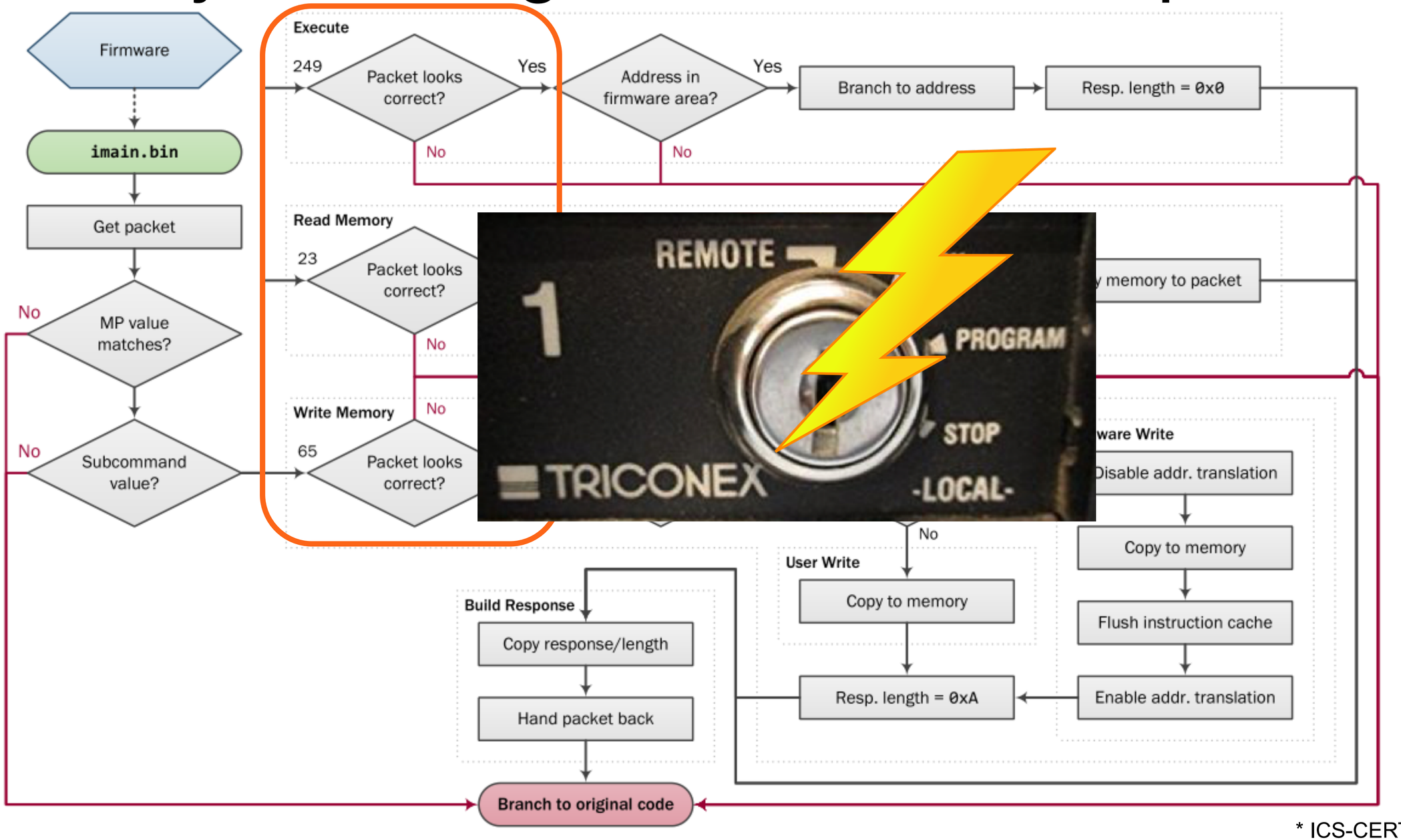
Midnight Blue



* ICS-CERT Report on HATMAN

Payload Stage 3: Backdoor Implant

Midnight Blue



* ICS-CERT Report on HATMAN

2 Payload Stage 4: OT Payload?

Midnight Blue

- Once backdoor is injected, we have god mode
- Still need OT payload to carry out ‘meat’ of the attack
- Not recovered from incident, hard to determine attack (sub) goal
- Asset owner can make educated guess, we can only speculate …
- **Which we will!**

2 Possible TRITON OT Payloads

Midnight Blue



3

Prevent response



Control / Safety System

Engineering

Modify operational / safety limits

Blind about process state



A I/O SPOOFING

Source: Wikimedia commons

OT PAYLOAD: I/O SPOOFING

Midnight Blue

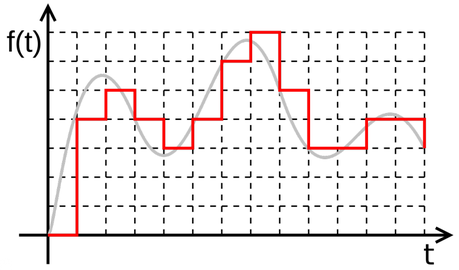
Measurement







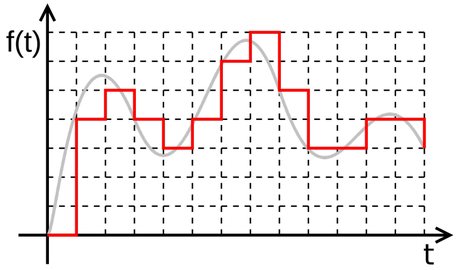
Input Signal



Controller



Output Signal





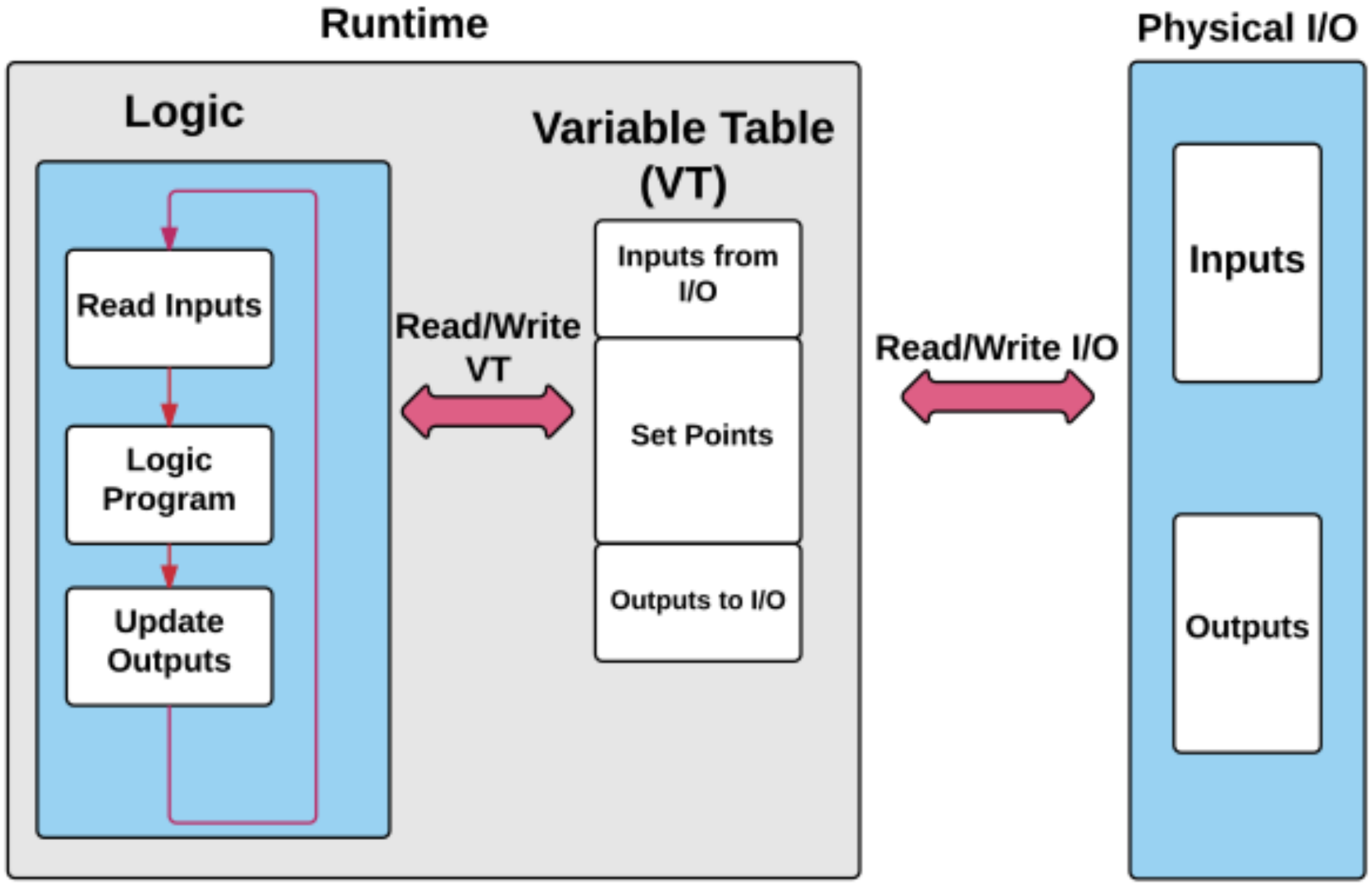
Instrumentation





I/O TRANSLATION

Midnight Blue



Source: Ghost in the PLC – Ali Abbasi, Majid Hashemi, BlackHat EU 2016

Triconex MP I/O

Midnight Blue

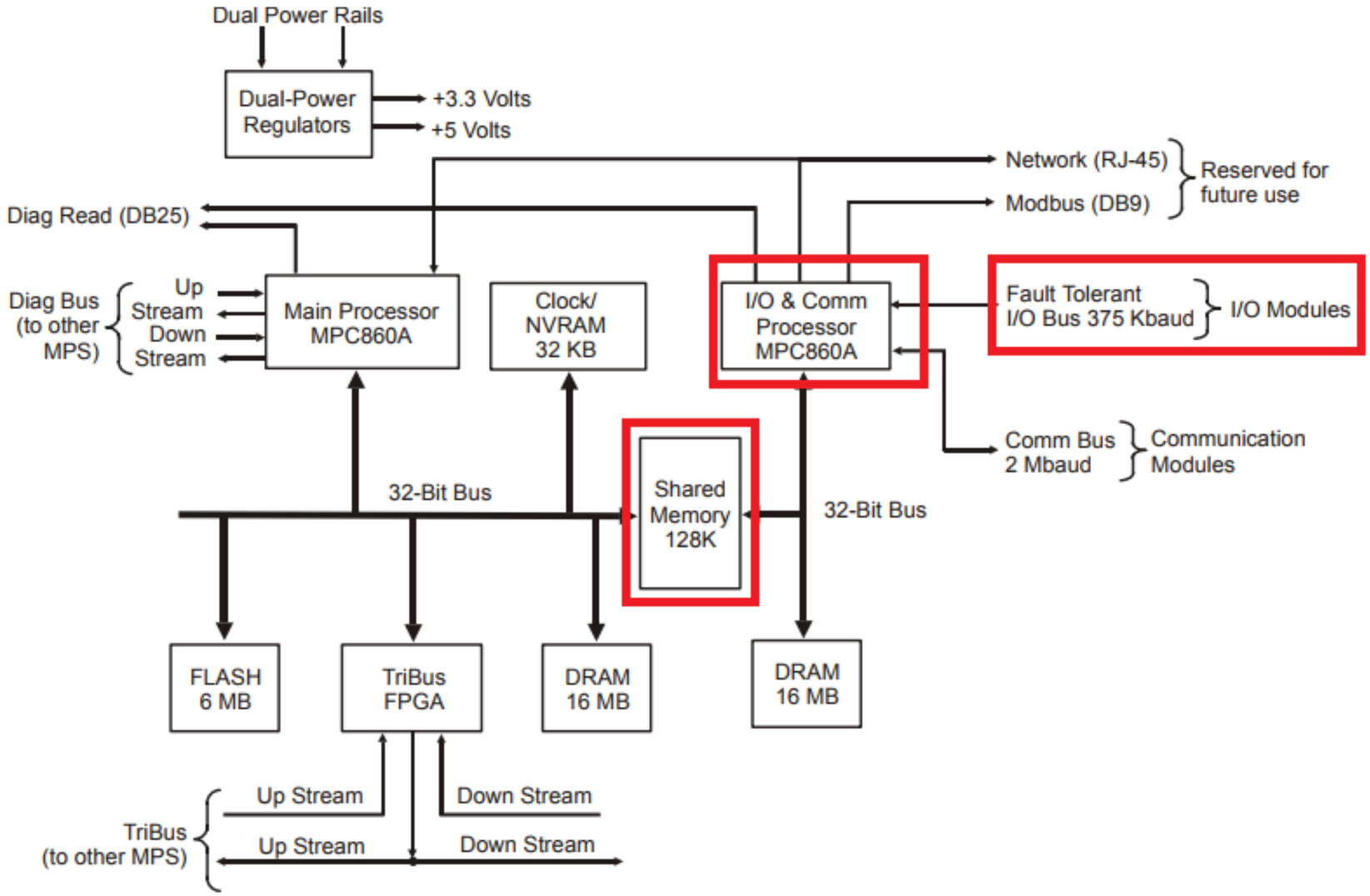
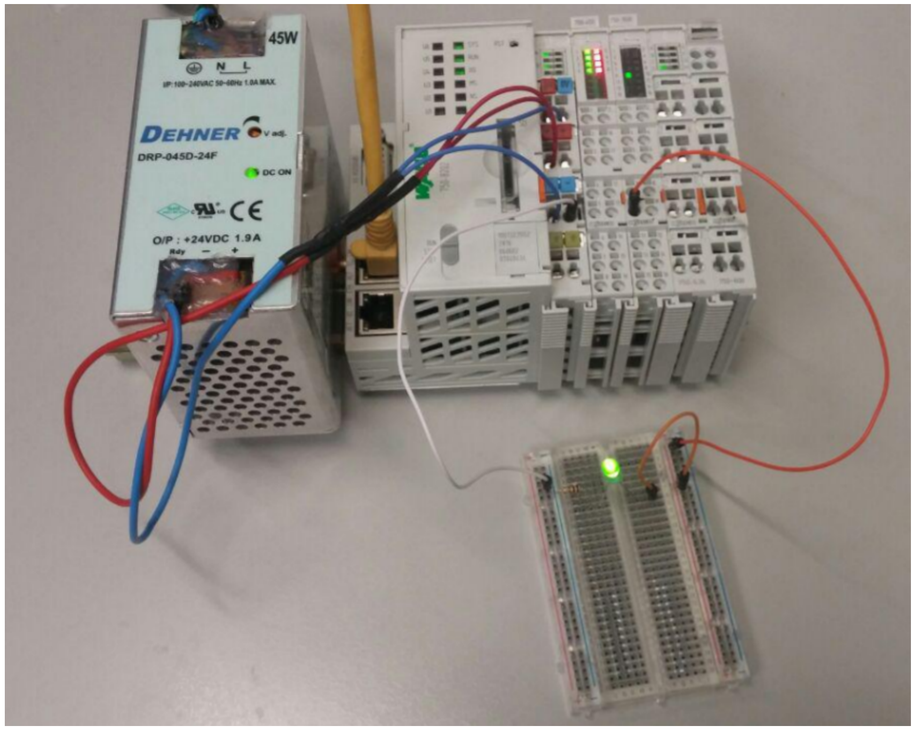


Figure 3 Architecture of a Model 3008 Main Processor

Example: Wago PFC 200 + 8CH DIO

Midnight Blue

- **ARM Cortex A8 + Real-Time Linux**
- **CODESYS runtime (TCP/2455)**
  - Runs as root
  - V2.4.7 has unauth arbitrary file read/write vuln*
- **Use CPU debug registers to catch access to memory mapped IO**
- **Custom exception handler changes pin mode from output to input to prevent outgoing writes to ‘actuator’**

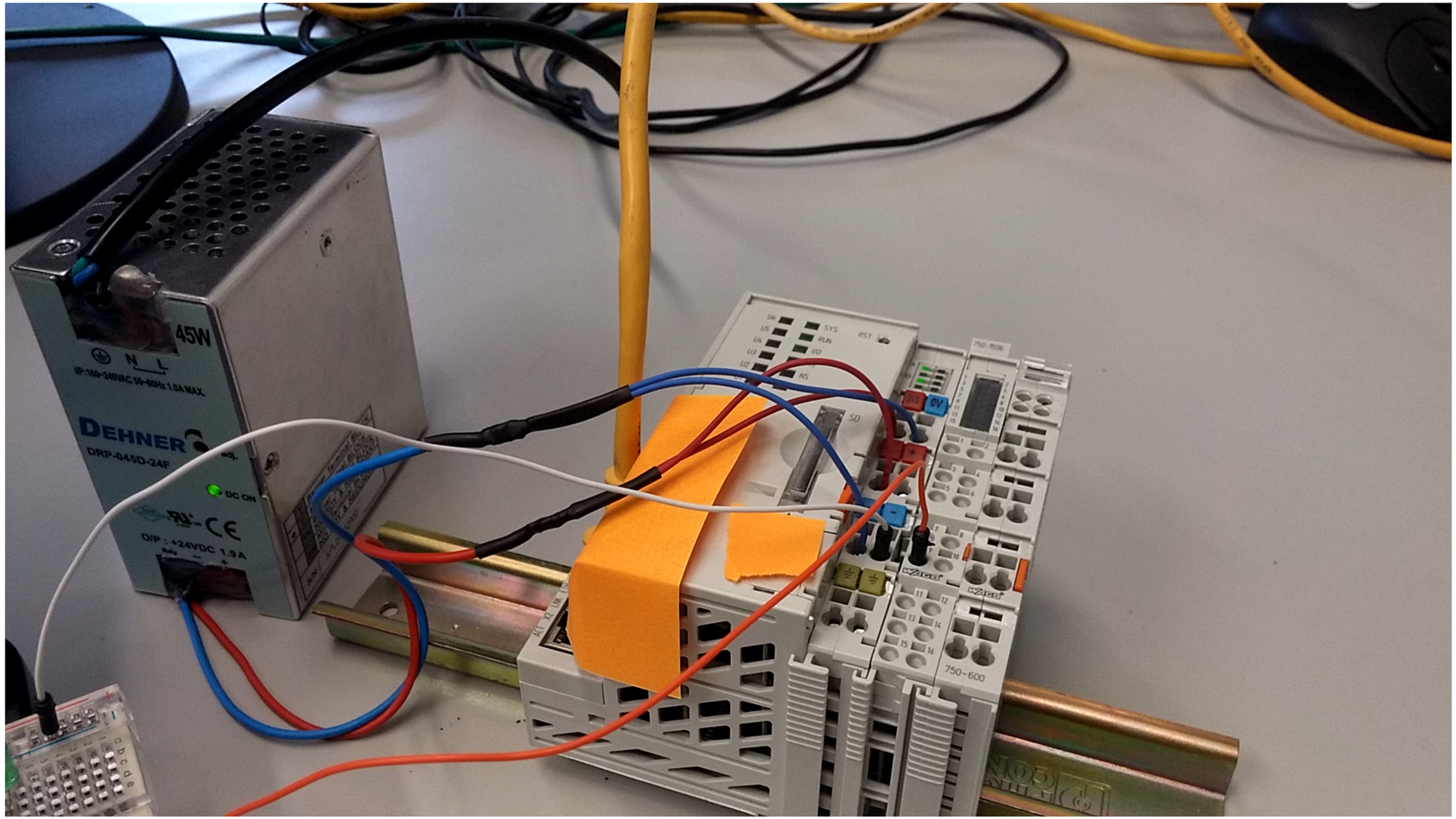


Source: Ghost in the PLC – Ali Abbasi, Majid Hashemi, BlackHat EU 2016

* https://www.sec-consult.com/en/blog/advisories/wago-pfc-200-series-critical-codesys-vulnerabilities/

DEMO: I/O Spoofing Attack

Midnight Blue





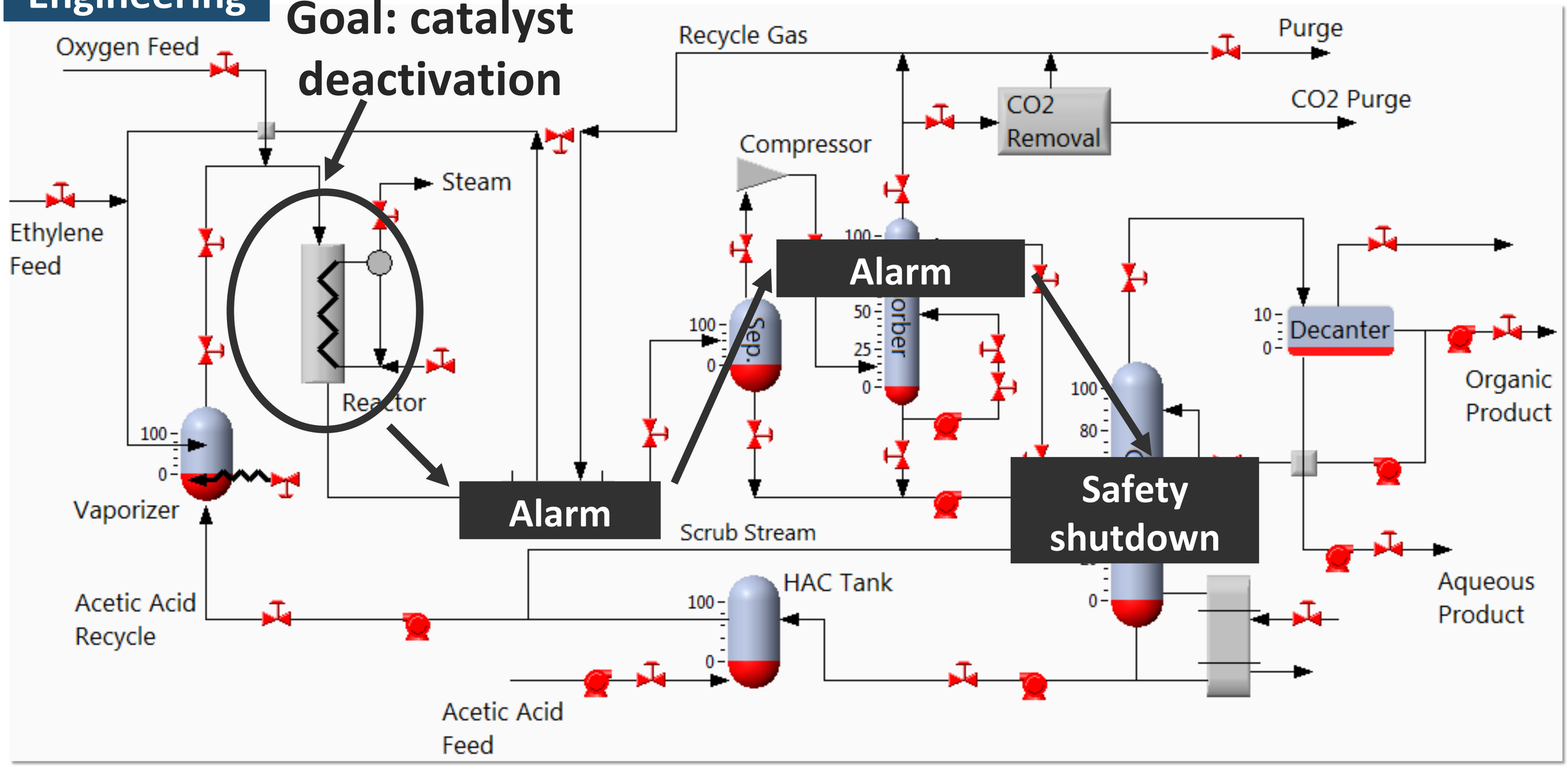
BLARM SUPPRESSION

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

OT PAYLOAD: ALARM SUPPRESSION

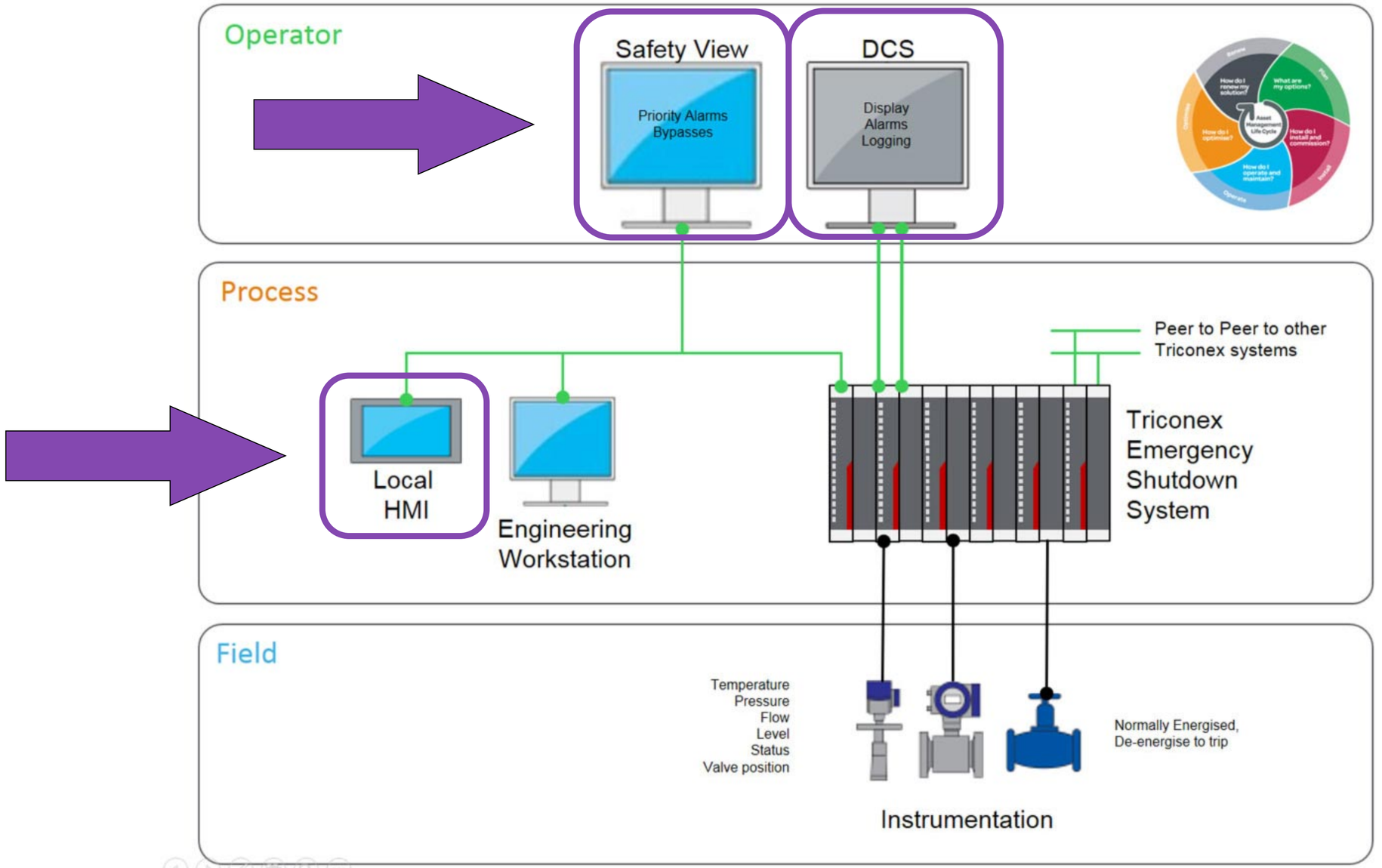
Midnight Blue

Engineering



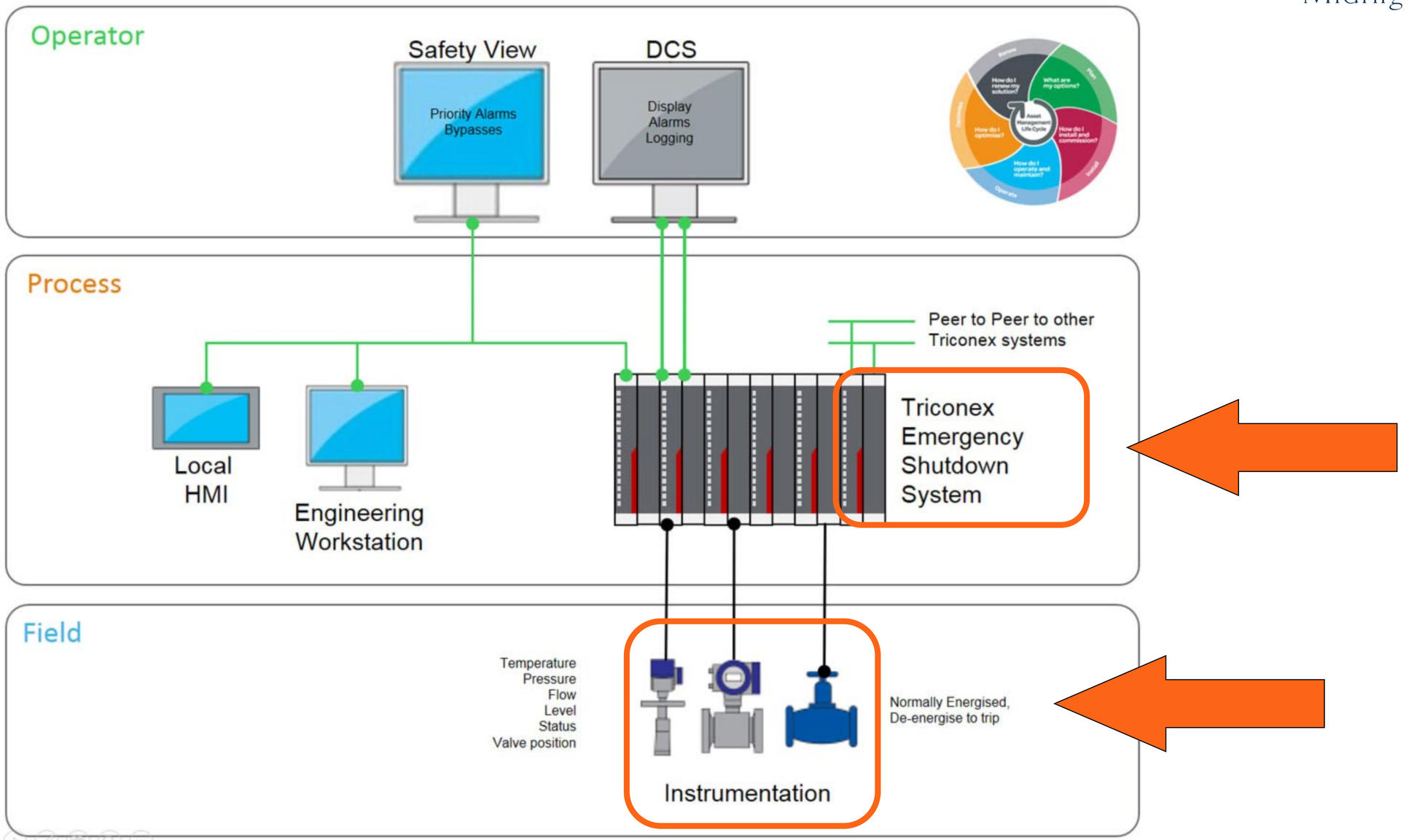
HIDING ALARMS

Midnight Blue



SUPPRESSING ALARMS

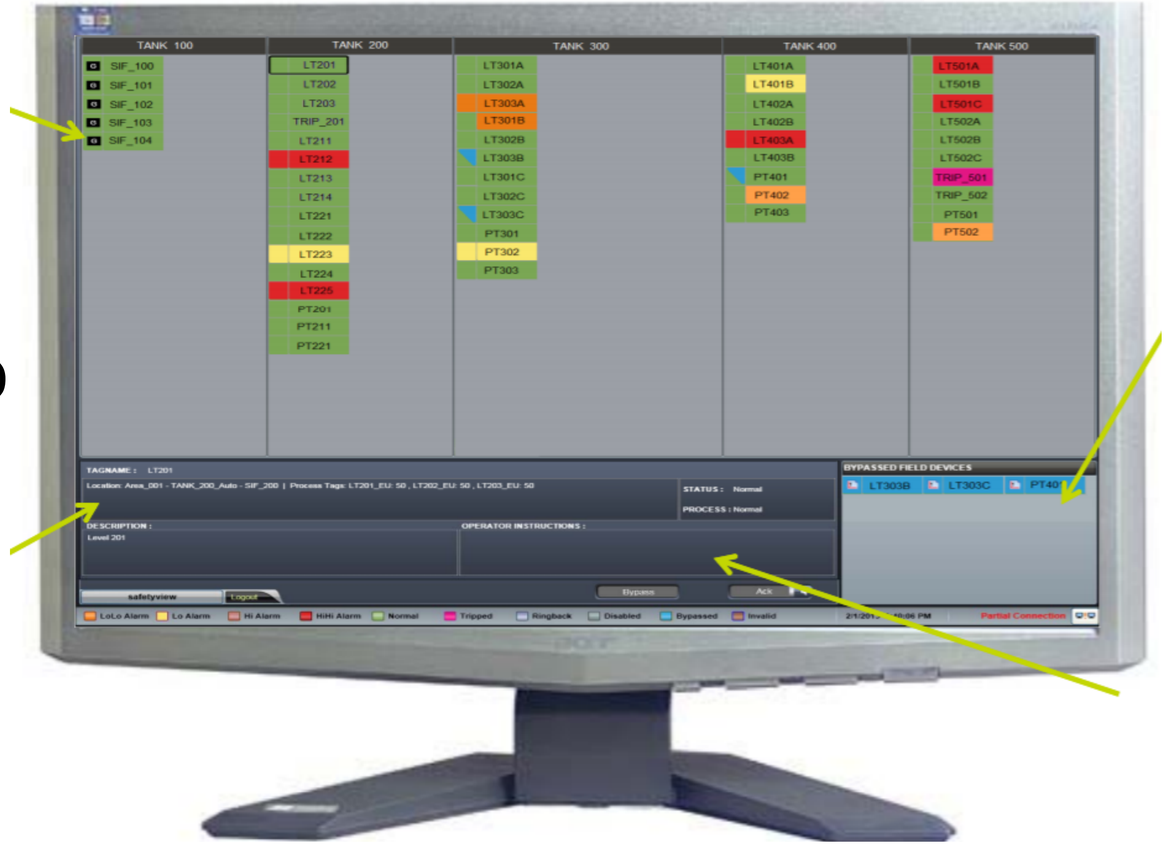
Midnight Blue



Example: Triconex Safety View

Midnight Blue

- PC-based HMI
- Management & bypass of priority 1 alarms
- Each HMI function is mapped to Triconex logic function blocks

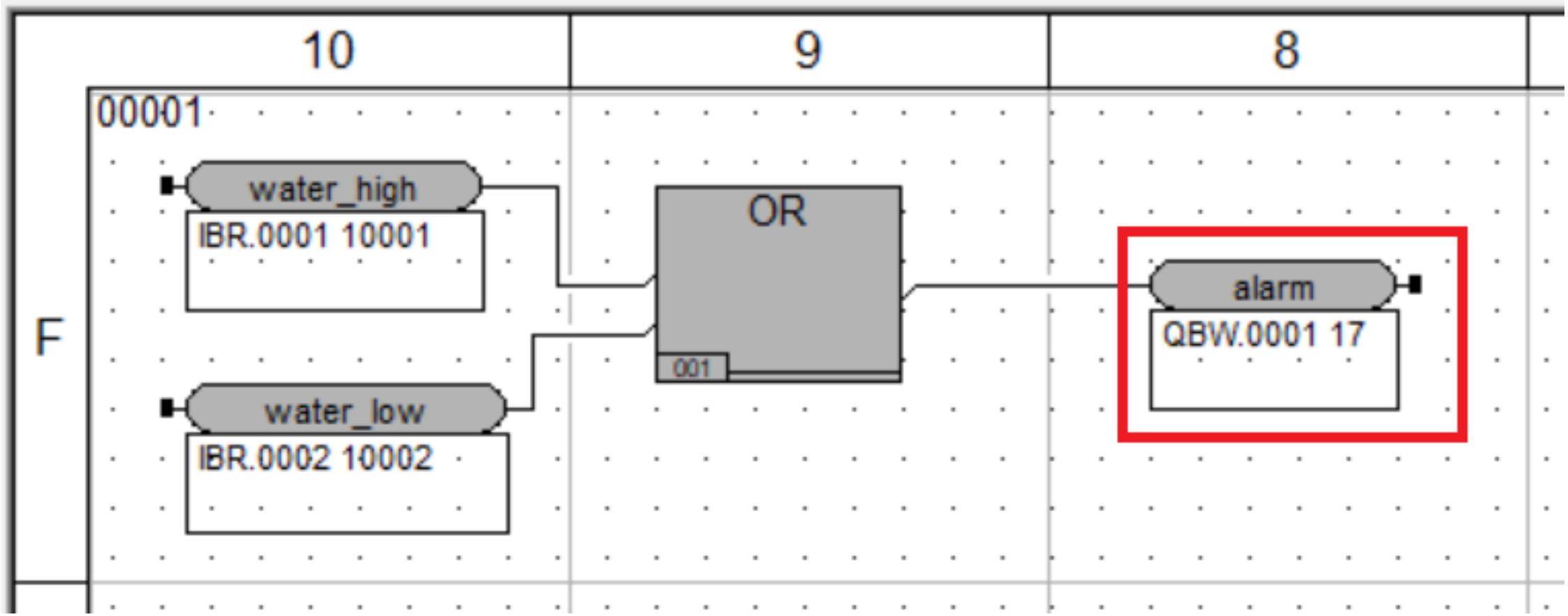


Source: Invensys / Schneider Electric

Example: Triconex Alarm FBs

Midnight Blue

- **Consider simple water tank level alarm**
  - OR of measurement DIs -> alarm DO



Example: Suppressing Alarms

Midnight Blue

- **Safety Program resides in-memory as code**
  - OT payload can modify instructions to set alarm to fixed FALSE
  - Stored program on flash remains untouched
- **Attacker needs to know**
  1. Where program lives in memory
  2. Which instructions of program to modify

Example: Finding Safety Program

Midnight Blue

- **Programs stored as circular linked list**

```
curr.flink:     .long next.flink
curr.blink:     .long null_entry
                .long 0xA04
```

```
# =============== S U B R O U T I N E =======================


inject_bin_start:                       # DATA XREF: RAM:00
                bl      inject_init
                li      r3, -1
                sc
# End of function inject_bin_start


# =============== S U B R O U T I N E =======================


do_syscall_0x13:                        # CODE XREF: sub_FE
                                        # sub_FE56D4+94↓p
```

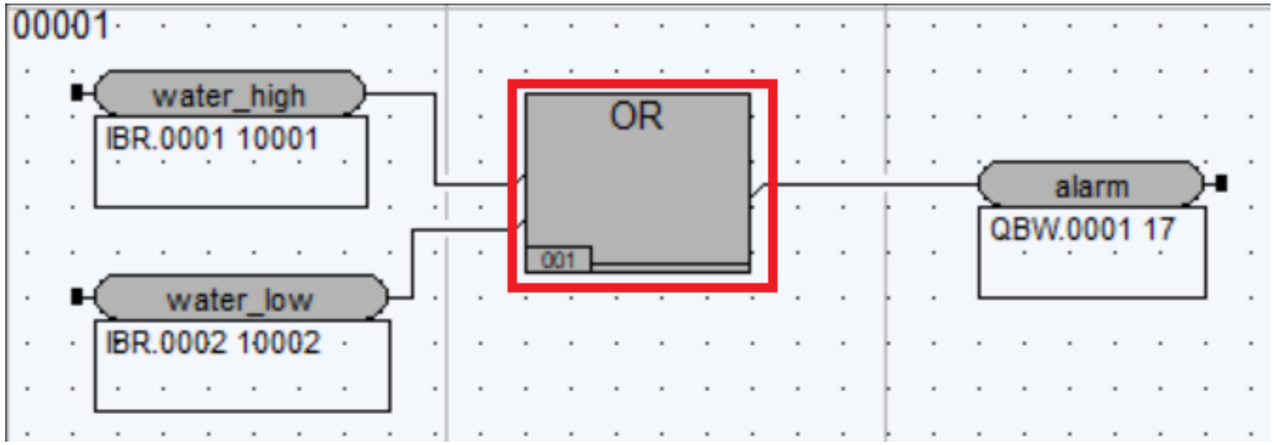
```
                .long inject_bin_start
                .long 0x96385E0B
next.flink:     .long null_entry

next.blink:     .long null_entry
                .long 0x60
# ---------------------------------
                |
some_other_program:
                lis     r2, 0x80

loc_FE5E24:
                lwz     r3, 0(r2)
```

Example: Analyzing Safety Program

Midnight Blue




```
                                # CODE XREF: end_loop+1C↓j
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)
```

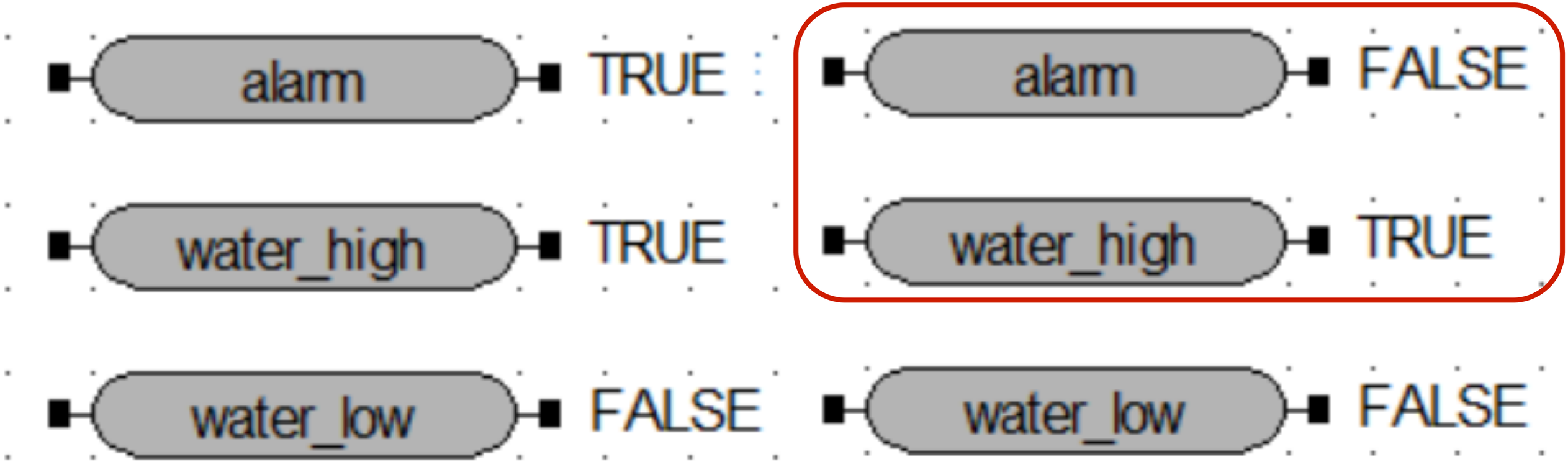
Example: Hot-Patching Safety Program




```
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
li          r26, 0        # alarm := FALSE
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)
```

Example: Alarm Suppression

Midnight Blue



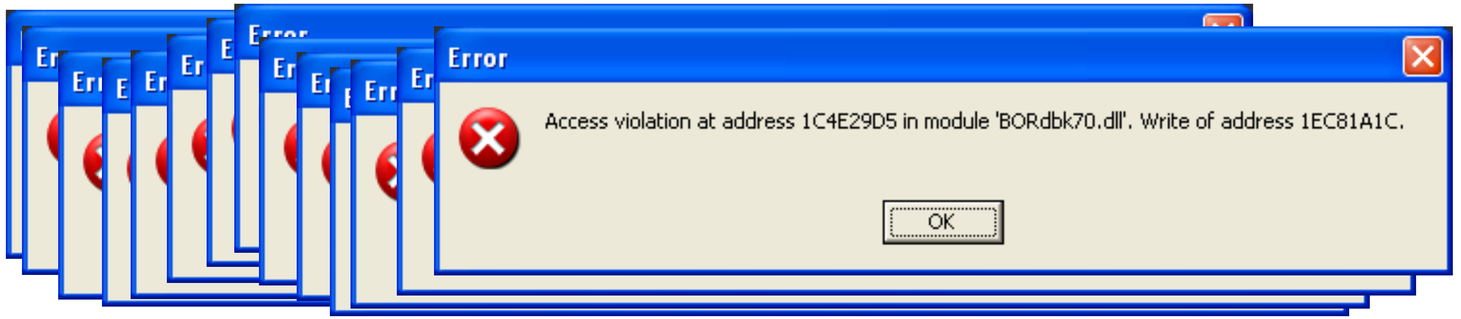


More Speculation Ahead: Why Did The Attack Fail?

Option A: b0rked payload?

Midnight Blue

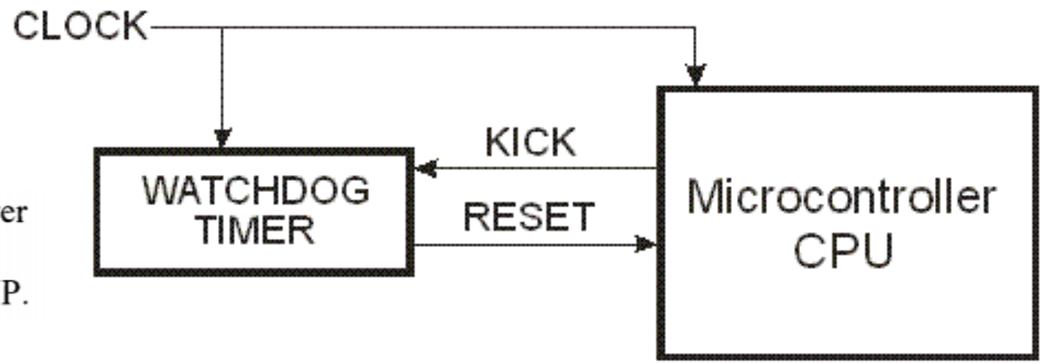
- **Failed PrivEsc / Backdoor allows for raw RWX**
  - You read / write / execute the wrong thing in the wrong place …



- **Getting into a fight with the watchdog**
  - Very common embedded way to shoot yourself in the foot

- **Missed diagnostics?**

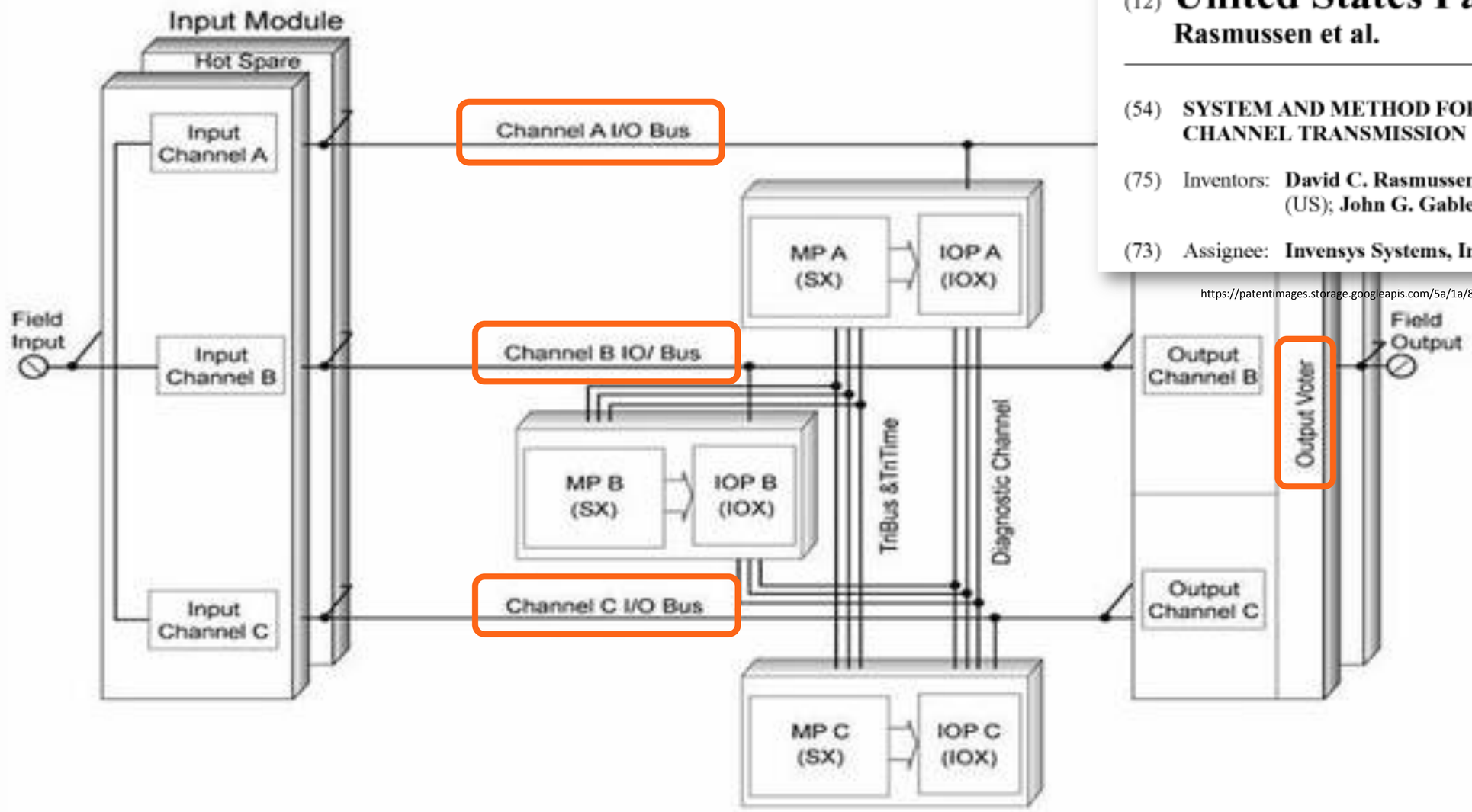
through the power up initialization and diagnostics. Power up sequence includes a series of Power up diagnostics – Microprocessor tests, RAM tests, Flash memory tests, Watchdog test, Clock Calendar test, etc. Power up sequence is also initiated by hardware and software reset of the EMP.



Source: https://betterembsw.blogspot.com

Option B: TMR?

Midnight Blue



(12) **United States Patent**
Rasmussen et al.

(54) **SYSTEM AND METHOD FOR VALIDATING CHANNEL TRANSMISSION**

(75) Inventors: **David C. Rasmussen**, Placentia, CA (US); **John G. Gabler**, Irvine, CA (US)

(73) Assignee: **Invensys Systems, Inc.**

https://patentimages.storage.googleapis.com/5a/1a/88/f75a93ace8c548/US8037356.pdf



CONCLUSIONS

TRITON Cost & Complexity Assessment



- **Obtaining Necessary Materials**
  - Documentation: public, very detailed (NRC)
  - Engineering Software: mostly free (sketchy websites) or cheap
  - Triconex Setup: > $7500 (used), > $20000 (new) [xN for bricking]
    - 1x Chassis + 1x PWS + 3x MP + 1x TCM + 1x IO
  - Firmware: In update util / flash, unencrypted

- **Protocol RE / Vulnerability Discovery**
  - Prior work on TSAA / TriStation relation (B. Lim et al., January 2017)
  - Engineering software with debug symbols
  - Unauthenticated protocol with sensitive functionality

TRITON Cost & Complexity Assessment



- **Exploit Development**
  - No safety / control program signing
  - Program compiled for & executed directly on main processor
  - Keyswitch has to be in program mode
- **Implant Development**
  - Required (simple) Privesc Exploit
  - Required deep firmware RE
  - No Triconex attestation / introspection (internal monitoring)!
  - TMR / Diagnostic Self-Tests
- **OT Payload Development**
  - Hardest part: deep firmware RE + understand position of *particular* SIS instance in process

TRITON Attacker Investment Risk

Midnight Blue

- **Vulnerability Discovery & Exploit Development: low**
  - Easy RE, no real VD or XD
  - Scales well beyond target to all Triconex
- **Implant Development: medium**
  - RE investment but scales at least within Triconex V10 with address mods
- **OT Payload Development: high**
  - Likely doesn’t scale beyond facility
  - Requires comprehensive (and difficult) OT recon
  - Complicated to test (do it live? $$$ lab setup?)

Open Questions

Midnight Blue

- **If part of broader ICS arsenal, where's the rest?**
- **In what light should TRITON dev cost be seen?**
  - Expensive for a one-off, cheap for a scalable one-time upfront?
- **What does the attack failure tell us?**
  - Implant development = Software development = 99% Frustration
  - Maybe stability sacrificed in R&D cost/benefit judgement?
- **If or when for copycats?**
  - Either of TRITON or as blueprint against other SIS



Thank You Note

Midnight Blue

- **Ali Abbasi**, Uni Bochum, Germany
- **Thorsten Holz**, Uni Bochum, Germany
- **Felix ‘FX’ Lindner**, Recurity Labs
- Various security community folks who kindly contributed to our knowledge and experience

TOOLKIT

Midnight Blue

- **Disassemblers / Decompilers**: IDA, Radare2, Binary Ninja, Capstone Engine, Hopper, RetDec
- **Debuggers**: GDB, WinDBG, OllyDbg, Immunity Debugger
- **Emulators**: Unicorn Engine, Qemu
- **Raw File Analysis Tools**: Unix file, multidiff, any hex editor, file carvers
- **Firmware Analysis Tools**: binwalk, FACT, IDAPython Embedded Toolkit, FIRMADYNE, BinCAT, BinDiff, Diaphora
- NOTE: You'll end up writing a ton of IDAPython scripts ('*one-off*' or not)
- NOTE: You'll also spend ages finding tools / info on weird & ancient chips


Questions?
@s4mvartaka
j.wetzels@midnightbluelabs.com
www.midnightbluelabs.com
@marmusha
marmusha@gmail.com