FORESCOUT

Automated Cybersecurity Across Your Digital Terrain

NAKATOMI SPACE

Lateral Movement as Level 1 Post-Exploitation in OT

Jos Wetzels
Security Researcher, Forescout

Who am I?

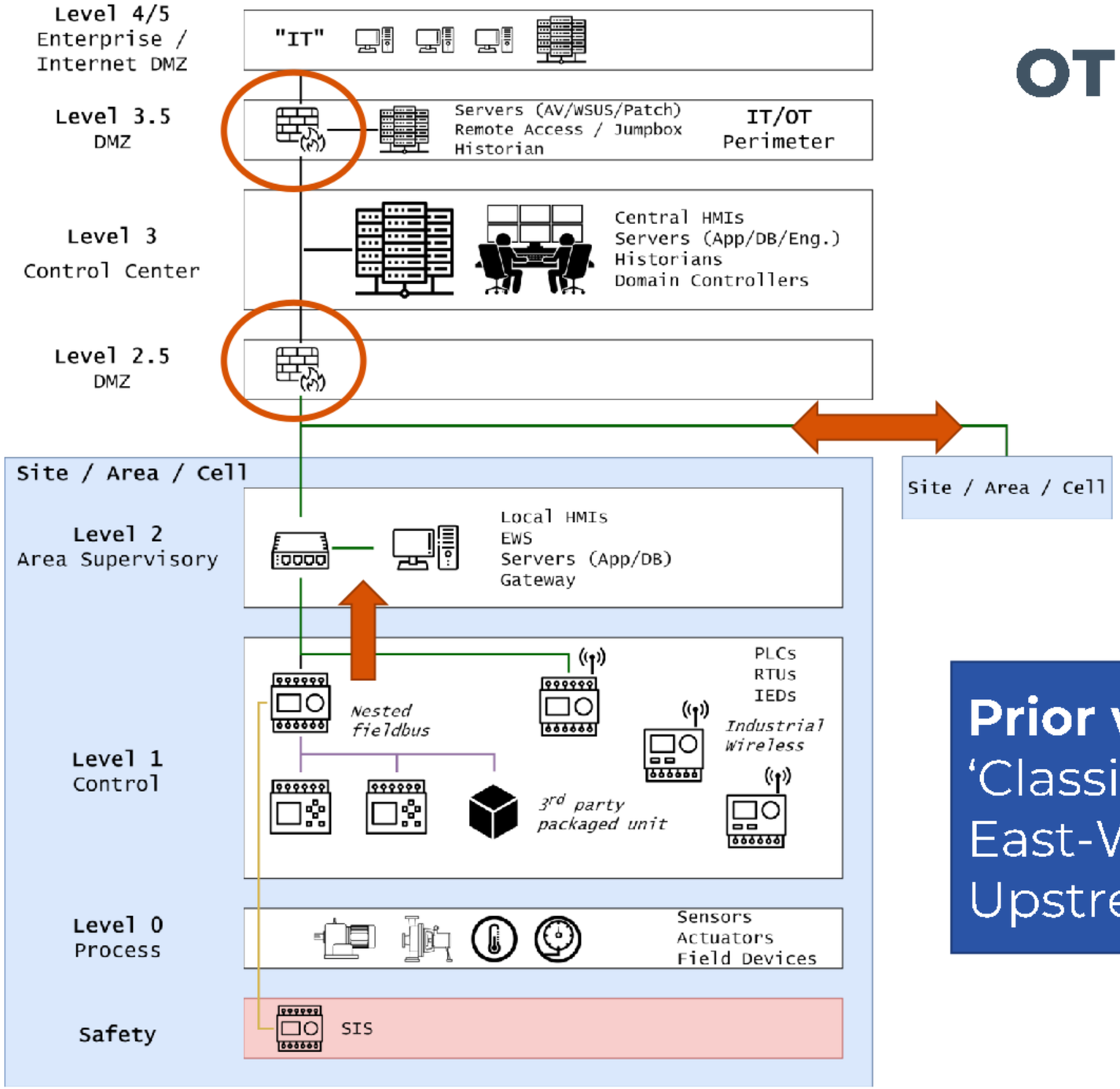
- Security Researcher @ Forescout
  - Focus on OT / IoT, embedded systems in general
- Joined Forescout in 2018 via SecurityMatters
  - OT-focused cybersecurity vendor
- Previously, researcher @ University of Twente (NL)
- Frequent speaker at security conferences, such as Black Hat, DEF CON, CCC, HITB, etc.

jos.wetzels@forescout.com

twitter: @s4mvartaka

https://www.linkedin.com/in/jos-w-61539598/

OT Lateral movement



Prior work

'Classical' perimeters at L3.5/L2.5

East-West @ L2+

Upstream to L2

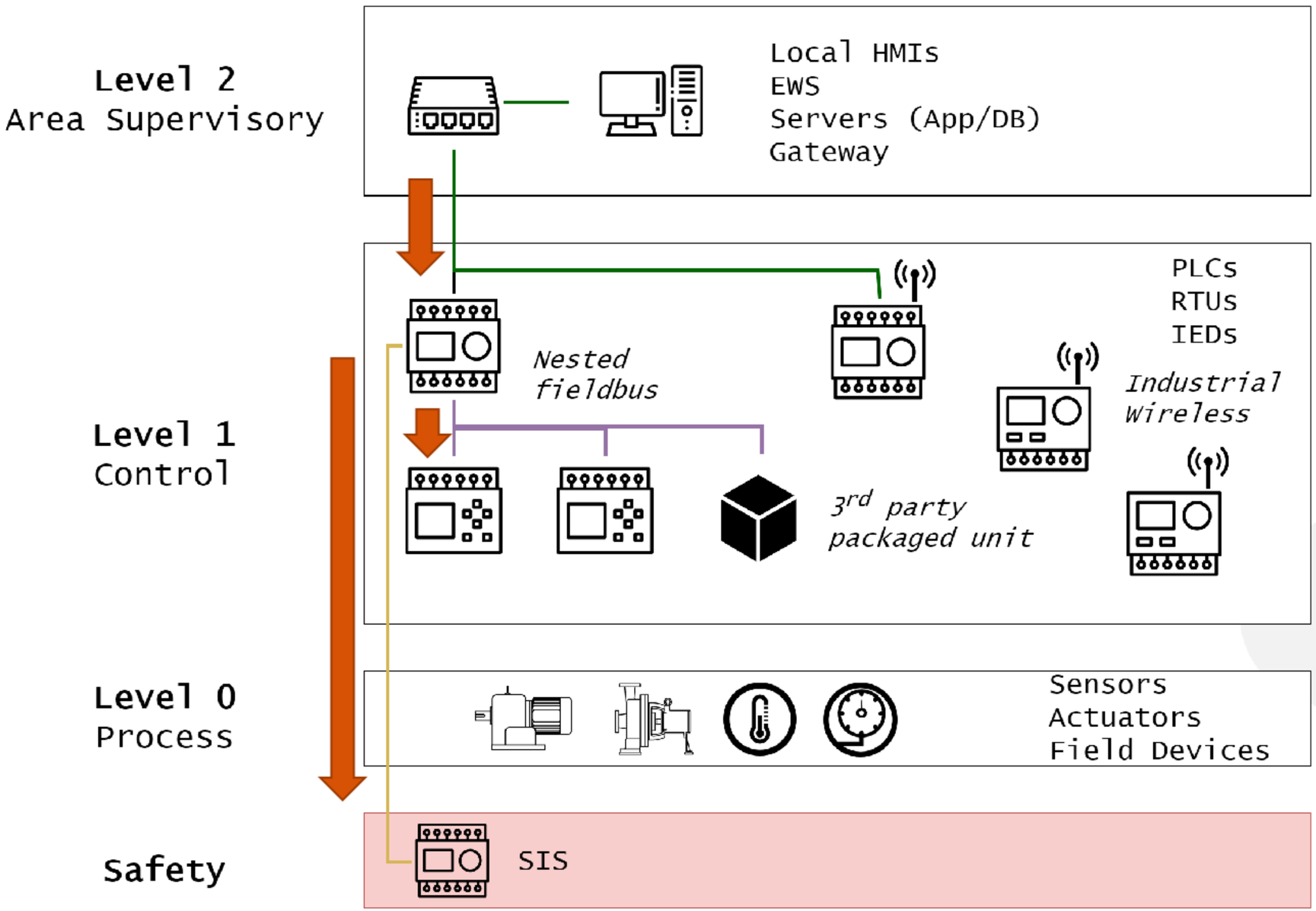
Nakatomi (Cyber)Space*

- **OT has lot of "*network crawl space*"**
  - Highly complex systems-of-systems
- **Lot of stuff beyond typical Ethernet networks**
  - Fieldbus networks (PROFIBUS/NET, CANopen, etc.)
  - RF networks (WirelessHART, 900MHz, TETRA WAN)
  - PTP links to 3rd party systems
- **Often complete lack of visibility**
  - Perimeters at this level often unacknowledged
  - Little awareness of possibility for maneuver
  - No ability to detect activity

Architectural elements with latent potential to enable traversing it in unintended and often overlooked ways

* https://bldgblog.com/2010/01/nakatomi-space/

Deep Lateral Movement



Focus
East-West @ L1
"Deep downstream"

Examples
Nested Fieldbus
Industrial Wireless
3rd Party PUs
BPCS / SIS links

Different Networks
Non-routable (PTP)
Non-IP (serial, RF)

Going *through* may require L1 RCE

- Has been demonstrated against many vendors now
- Several L1 post-exploitation TTPs have been publicly explored
  - Persistence[1,2,3,4,5]
  - Privilege escalation[2]
  - Evasion[2,6]
  - C2[7]
  - Exfiltration[8,9]
  - "*OT payloads*" (impair process control + inhibit response)[1,3,10,11,12]
- But no lateral movement at L1

[1] MITRE S0603, [2] MITRE S1009, [3] MITRE S1006
[4] INCONTROLLER: New State-Sponsored Cyber Attack Tools Target Multiple ICS - Mandiant
[5] Cyber-Security in Building Automation Systems - Forescout
[6] The Race to Native Code Execution in PLCs – T. Keren et al.
[7] Evil bubbles – M. Krotofil et al.
[8] Exfiltrating reconnaissance data from air-gapped ICS/SCADA networks - D. Atch et al.
[9] Greetings from the '90s M. Krotofil et al.
[10] Ghost in the PLC – A. Abbasi et al.
[11] A diet of poisoned fruit – J. Wetzels et al.
[12] Hey, My Malware Knows Physics! – L. Garcia et al.

Why bother? Reason #1: Perimeter crossing

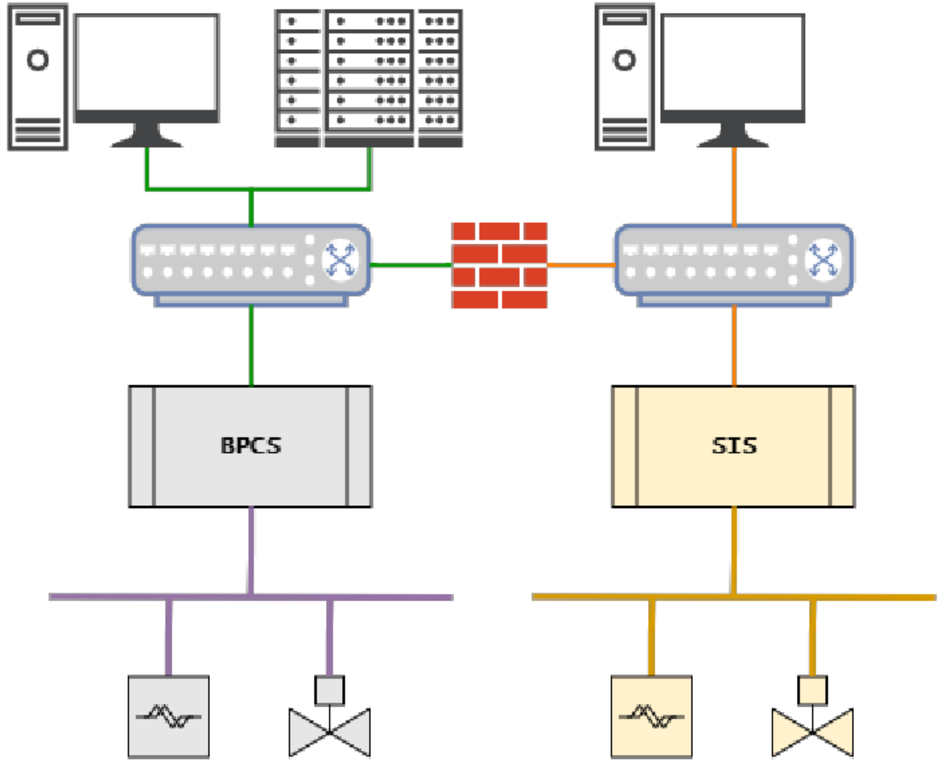
I need to move across hardened or unacknowledged perimeters

If a device is multi-homed (incl. serial/RF/etc. links) between different zones, it is a ***perimeter*** device

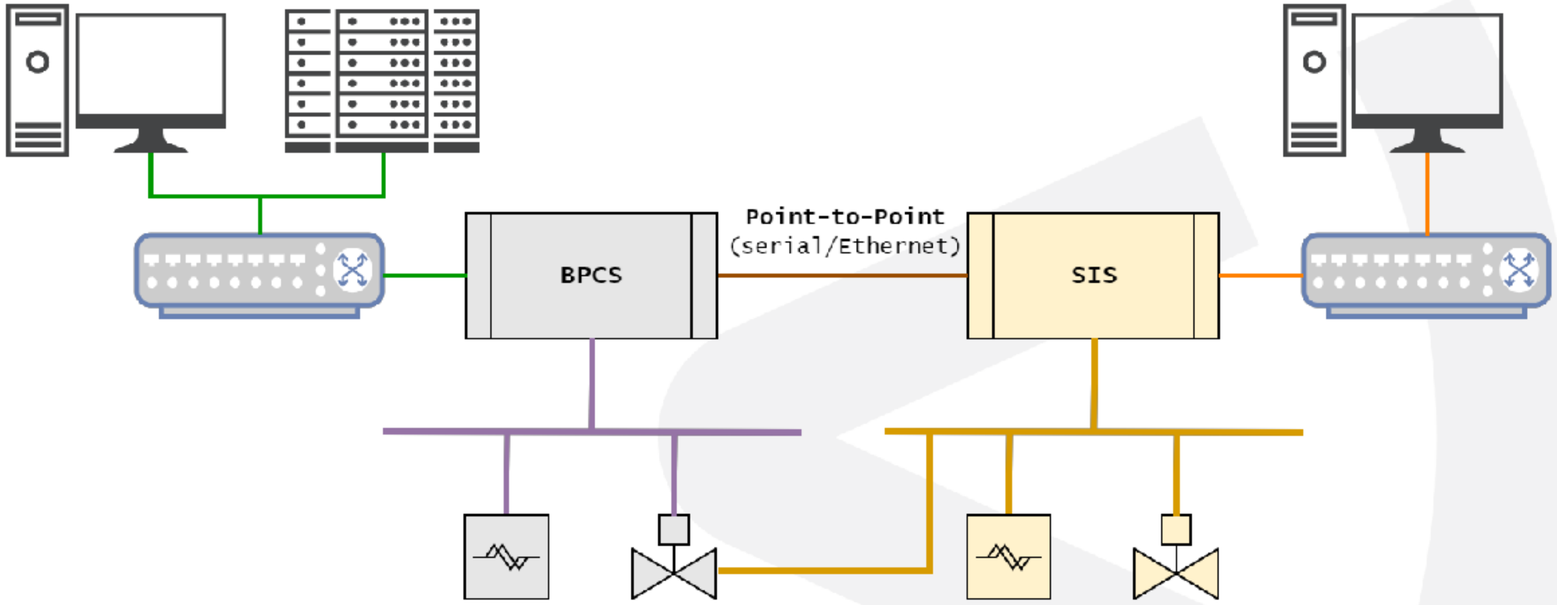
BPCS / SIS architectures

Can be generalized to any *distinct* but *interacting* control systems

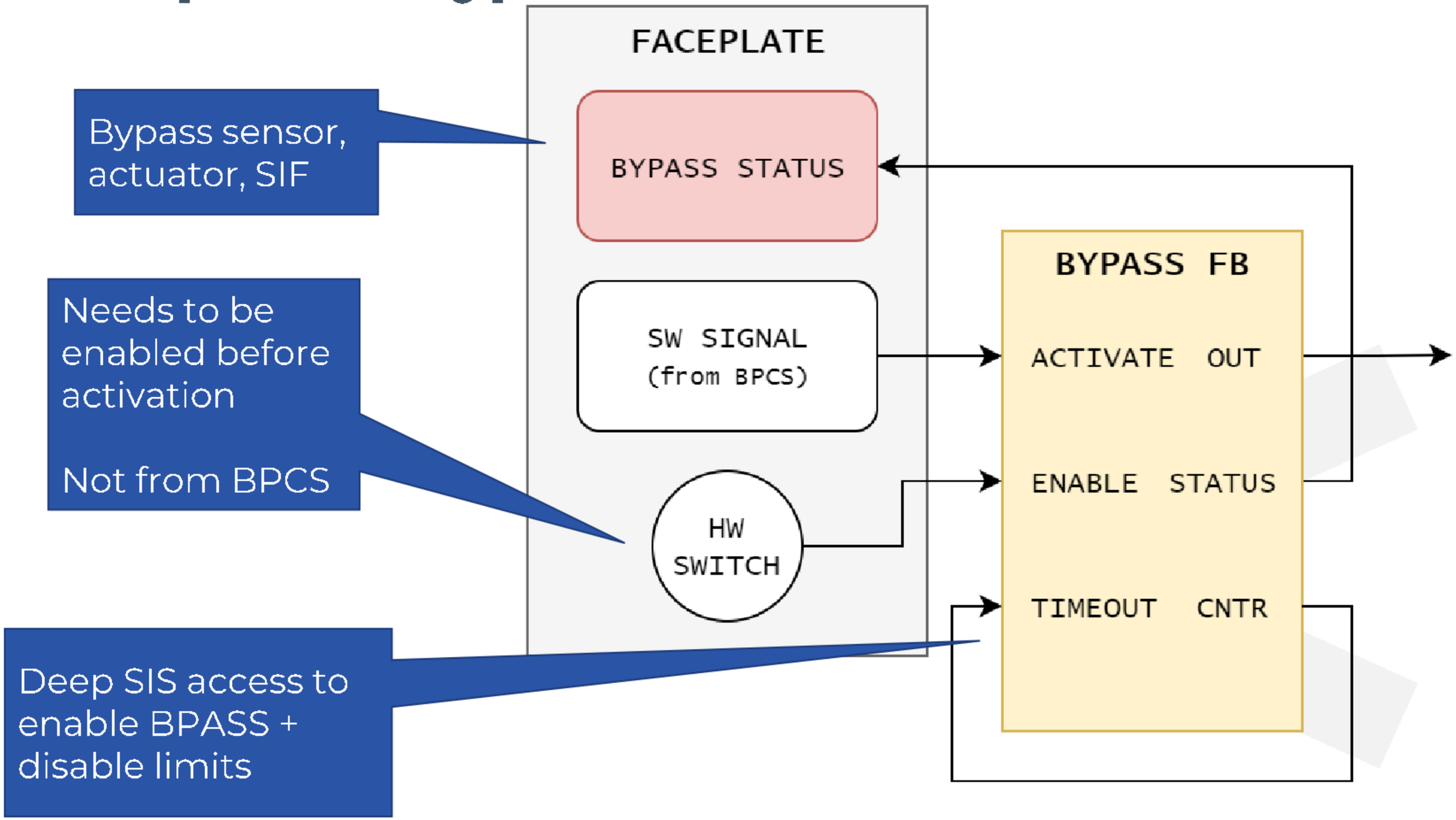
Integrated



Interfaced / “Shared”

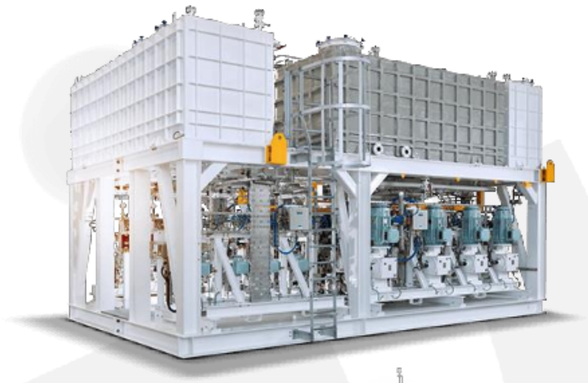


Example: SIS Bypasses

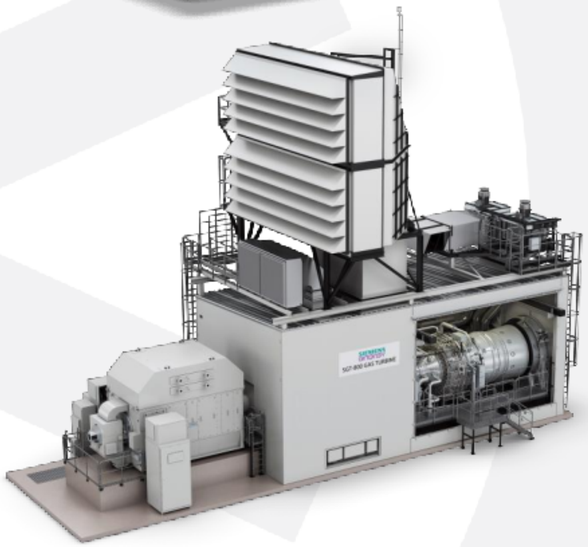


Packaged Units (PU)

- Blackbox control systems with specific function
  - HVAC, chemical injection, water treatment, gas turbine
  - Can range from subsystem to entire plant



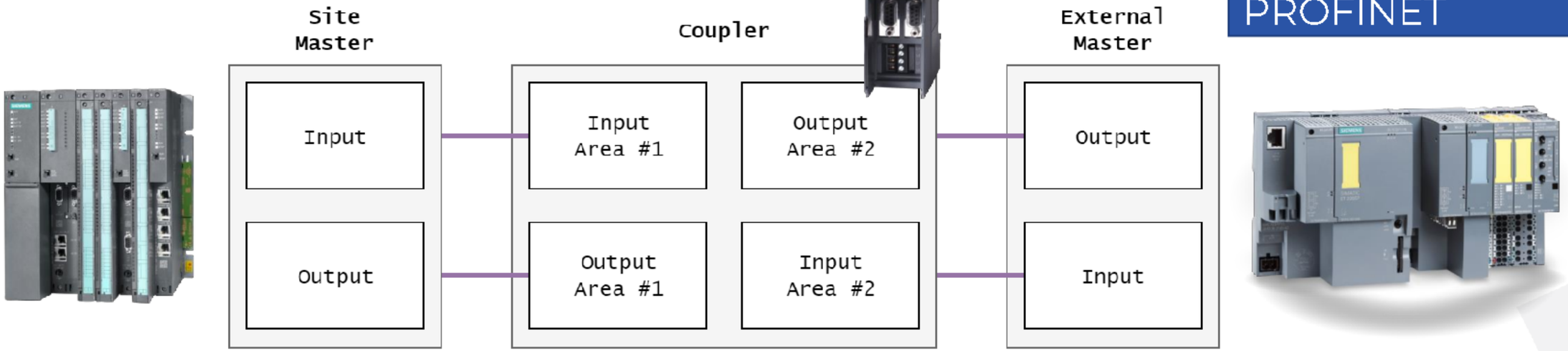
- Control/Monitoring interface to PCN/SCADA
  - Limited PVs / setpoints exposed
  - No direct control over PU internals

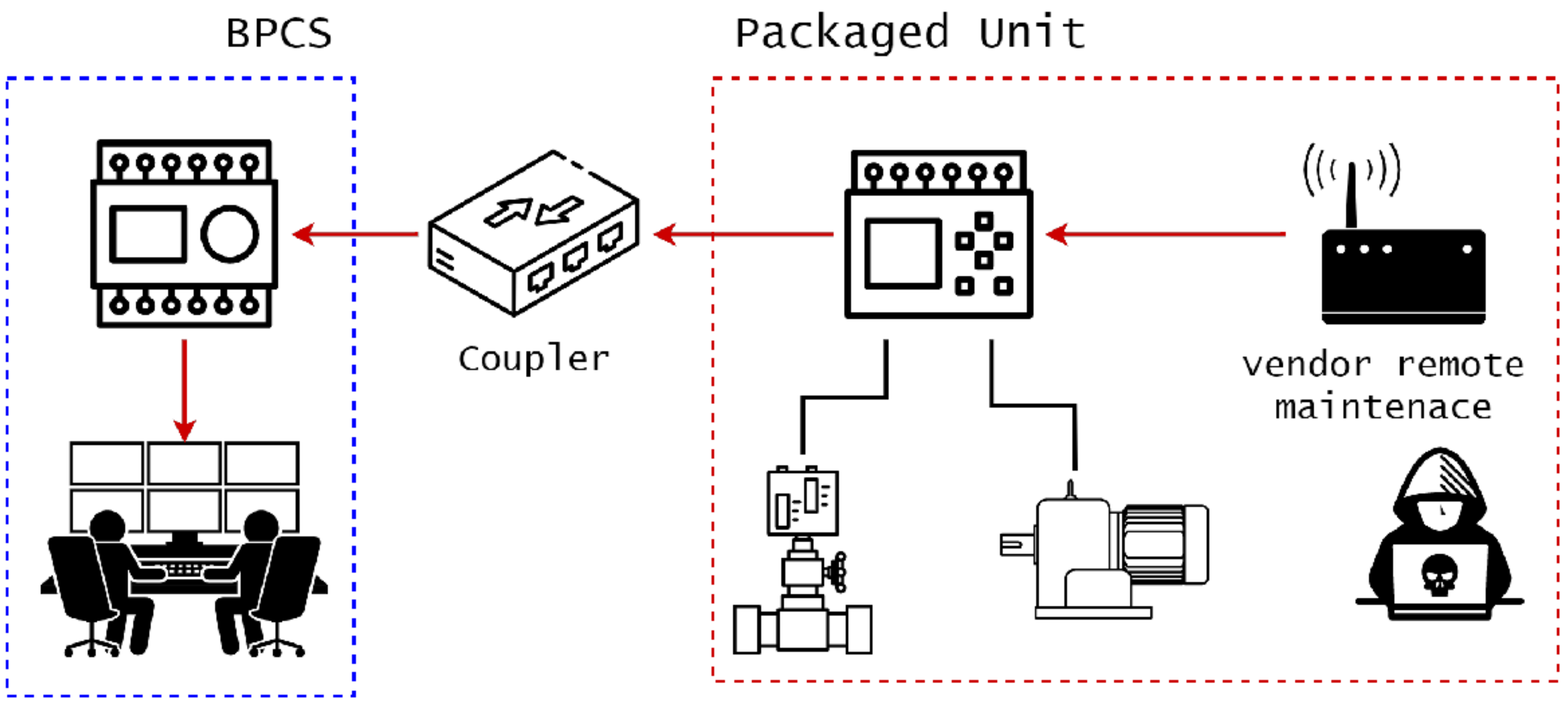


- Maintenance often done by 3rd party
  - E.g. cellular modem
  - Indirectly exposes PCN to external connectivity

Example: Fieldbus Couplers

Connect e.g. PROFIBUS DP ↔ PROFINET





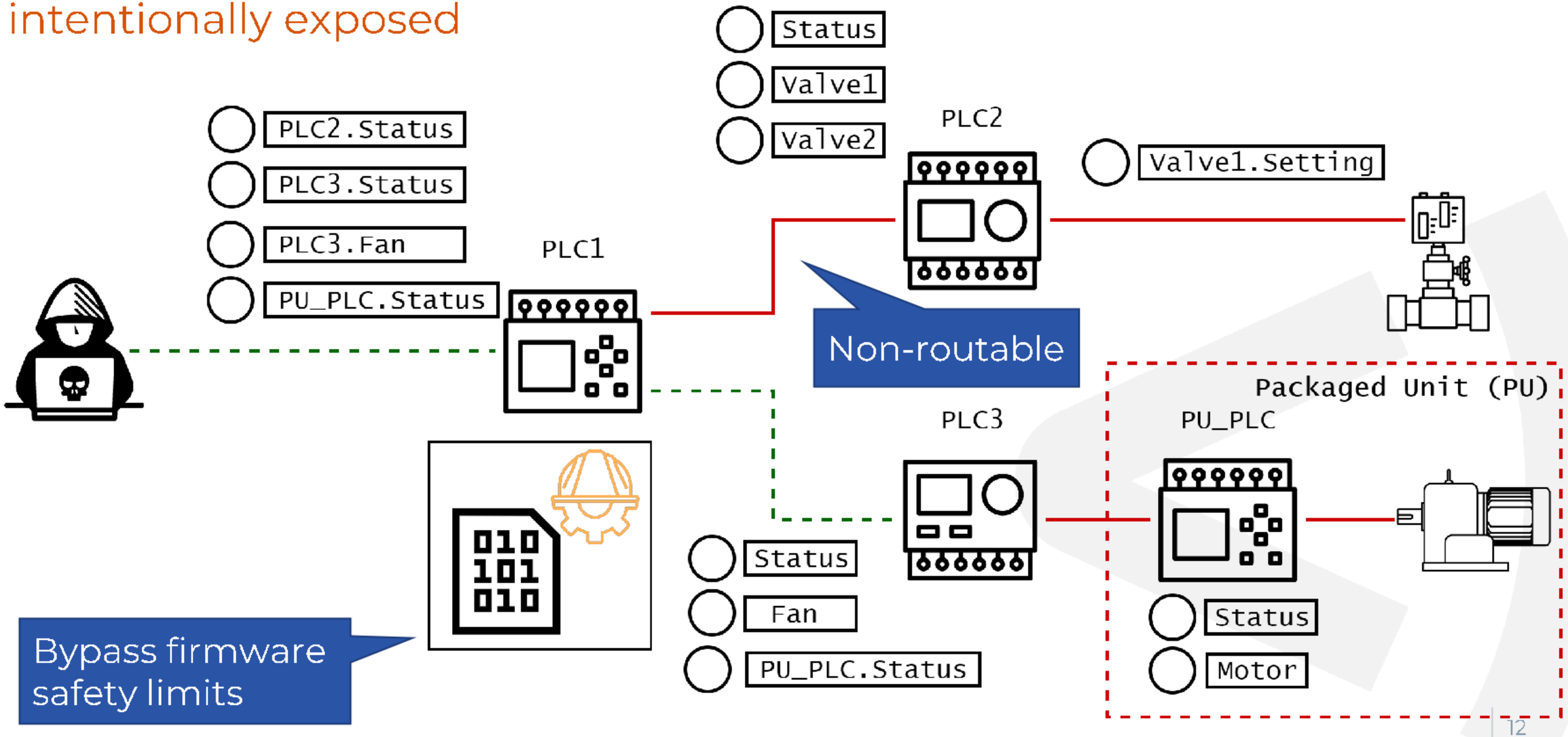
Often considered sufficient perimeter due to limited capabilities

Used to be 'dumb'
Increasingly 'smart'

Perimeter assumptions not evaluated for new attack surface

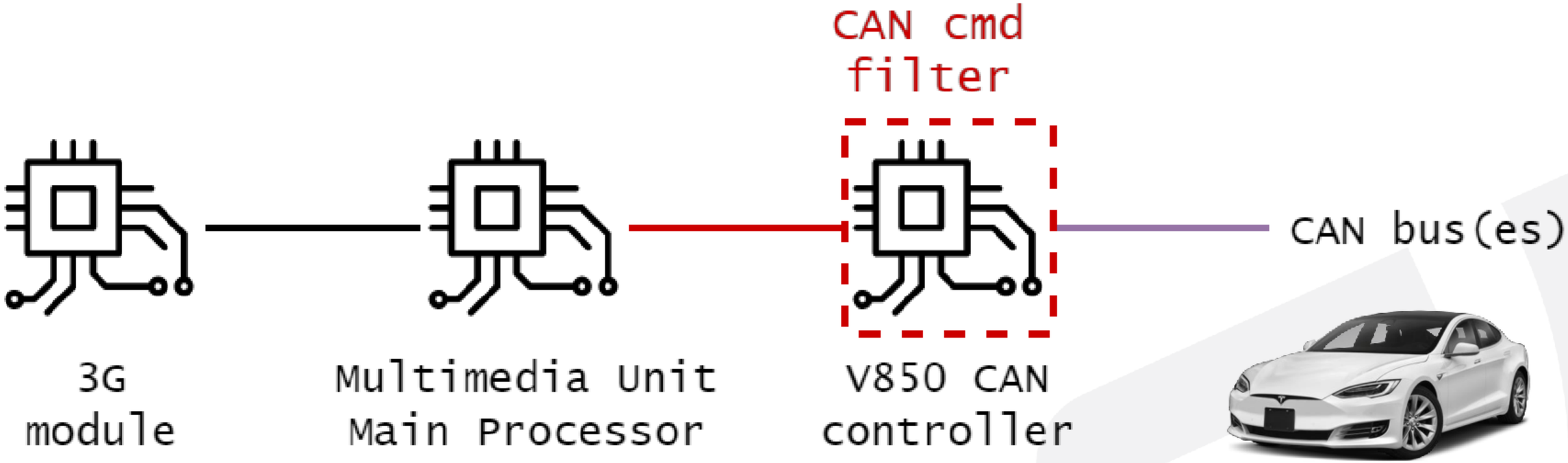
Why bother? Reason #2: Granular control

I want to talk to nested devices in a way not possible through what's intentionally exposed



Very common in automotive exploitation

RCE on CAN controller / GW to bypass filter → unrestricted CAN access



[*] C. Miller et al. (2015), [**] Tencent Keenlab (2016), [***] Computest (2018)

What do vendors & standards say?

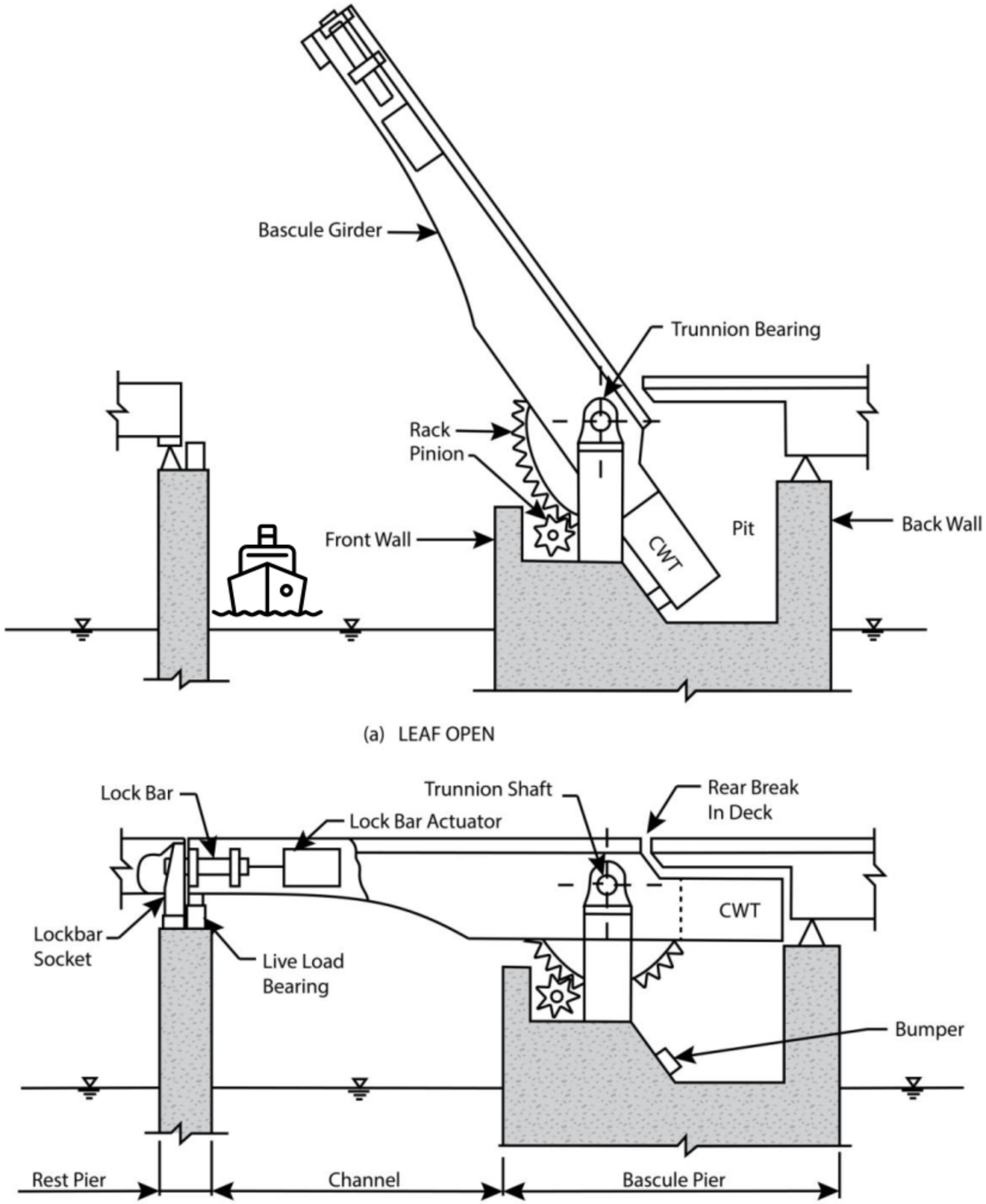
- General acceptance of integrated, interfaced and common architectures
- Usual segmentation advice
- Non-routable or serial PTP links are seen as sufficiently segmented
- Little attention to backplane security in multi-zone devices

There is a conduit between the BPCS zone and the SIS zone, presumably to provide read only data from the SIS to the BPCS. In this case segregation has been achieved by using a dedicated point-to-point serial connection. Note that the discrete I/O also shown

* IEC 61508 intro, ** HSE OG-0086 ed2, *** NERC CIP-005-6, **** NIST SP 800-82r2

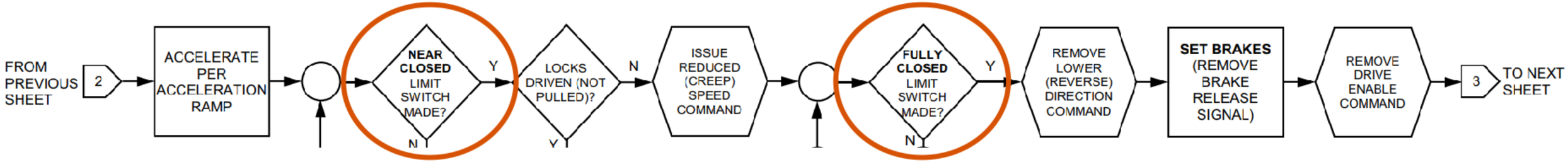
Proof-of-Concept Scenario

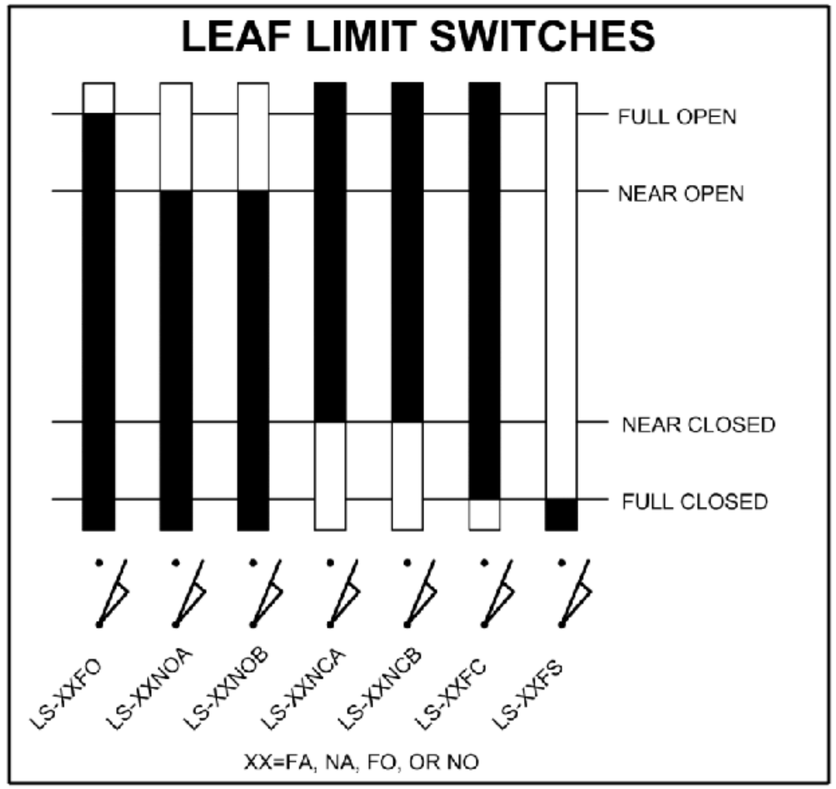
Scenario: Movable Bridge



^Bridge Maintenance Reference Manual – FDOT, structurae.net

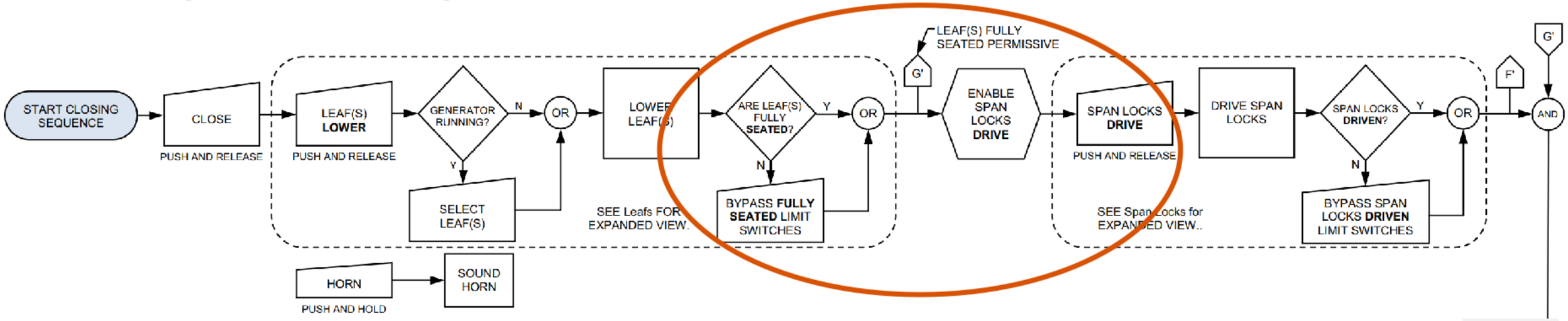
Bridge closing sequence – Limit Switches





*Bridge Maintenance Reference Manual - FDOT

Bridge closing sequence – Lock Bar



Attack Scenarios

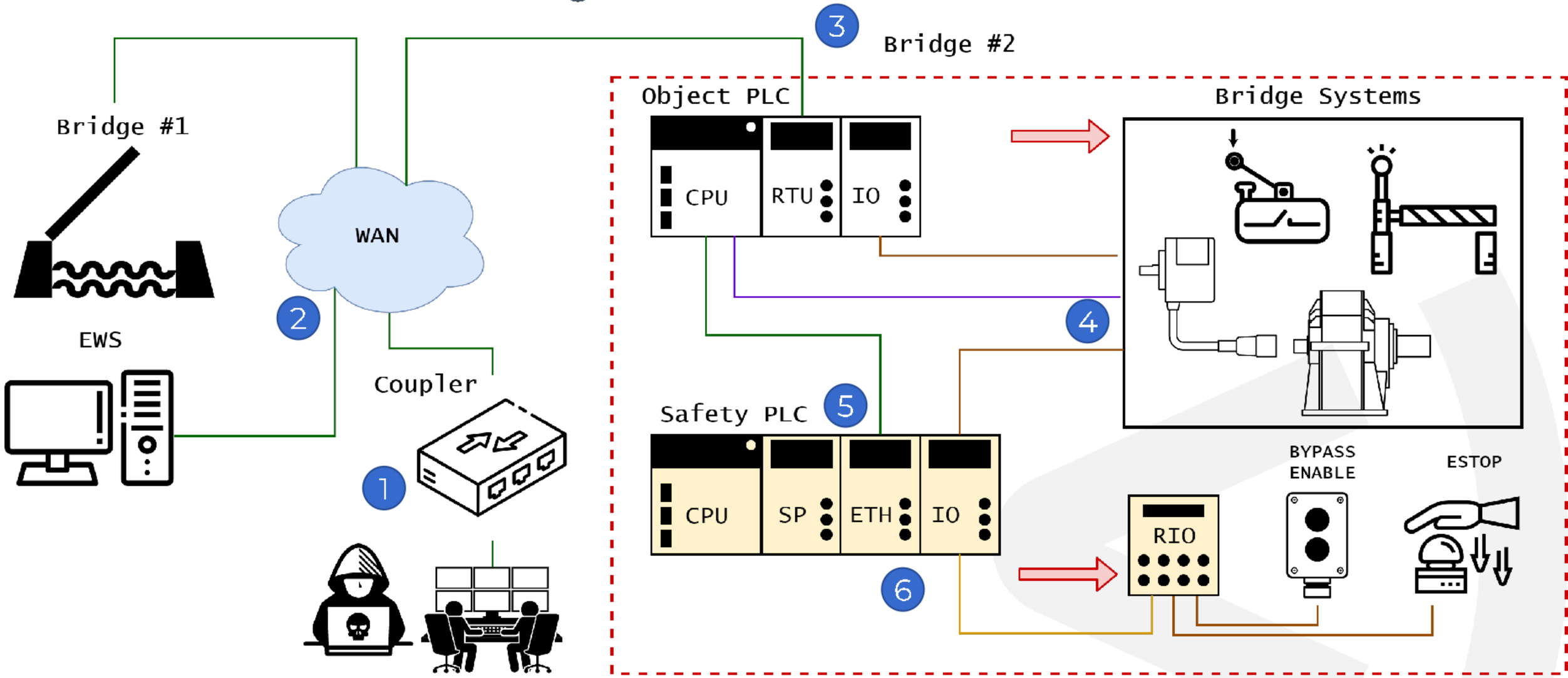
- **Scenario 1 : Close at full speed, hit bearings**
  - *Without* decel. to creep speed
  - Lock bar driven before closing

  - *Bypass* leaf/lock *limit switches*

- **Scenario 2 : Close at full speed, trigger E-STOP**
  - Wait until max velocity
  - E-STOP not graceful, CWT inertia

  - Bypass creep speed

Attack Path – Likely can't do this from SCADA



(1) RCE on Coupler (2) Auth Bypass (3) RCE on Object PLC

(4) Move into fieldbus (5) Cross SIS PTP link (6) Enable SIS bypass across backplane

Coupler → Object PLC RTU module

Cannot talk directly to M340 via Wago 750-852 coupler

Various protocols

Limited Modbus Mapping

No TDA (routing)

Get RCE on coupler via N-day → Proxy traffic to M340

Wago Coupler FW

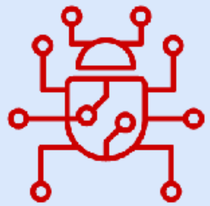
Supervisor mode
No mitigations
No tsk separation
RWX memory areas

NUCLEUS NET
TCP/IP Stack

NUCLEUS
Memory Alloc

Modbus
Handler

Hook Modbus handler,
turn into proxy



Stager Payload

Proxy Implant

CVE-2021-31886* on Wago 750-852

```
oid __cdecl Control_Task(UNSI

 FSP_CB *control_blocka; // [
 CHAR nu_drive[3]; // [sp+14h
 MNT_LIST_S *mount_list; // [
 NU_TASK *pointerToThisTask;
 FTP_SERVER server; // [sp+20
 CHAR commandBuf[8]; // [sp+1
 CHAR *buffer; // [sp+160h] [
 INT32 bytesReceived; // [sp+
```

- Stack bof in Nucleus FTPd "USER" cmd
  - Check via strlen() but copy until '\r' → use fake 0x00
  - Overwrite FTP_Events linked list after user buff
  - Disconnect → trigger unlink → write-4

  - RWX .bss area suitable for shellcode
  - Write shellcode ptr to span_process_packet func ptr

  - New FTP session → overwrite buffer ptr with shellcode ptr
  - Write shellcode via subsequent FTP data
  - LLC frame to trigger shellcode via span_process_packet

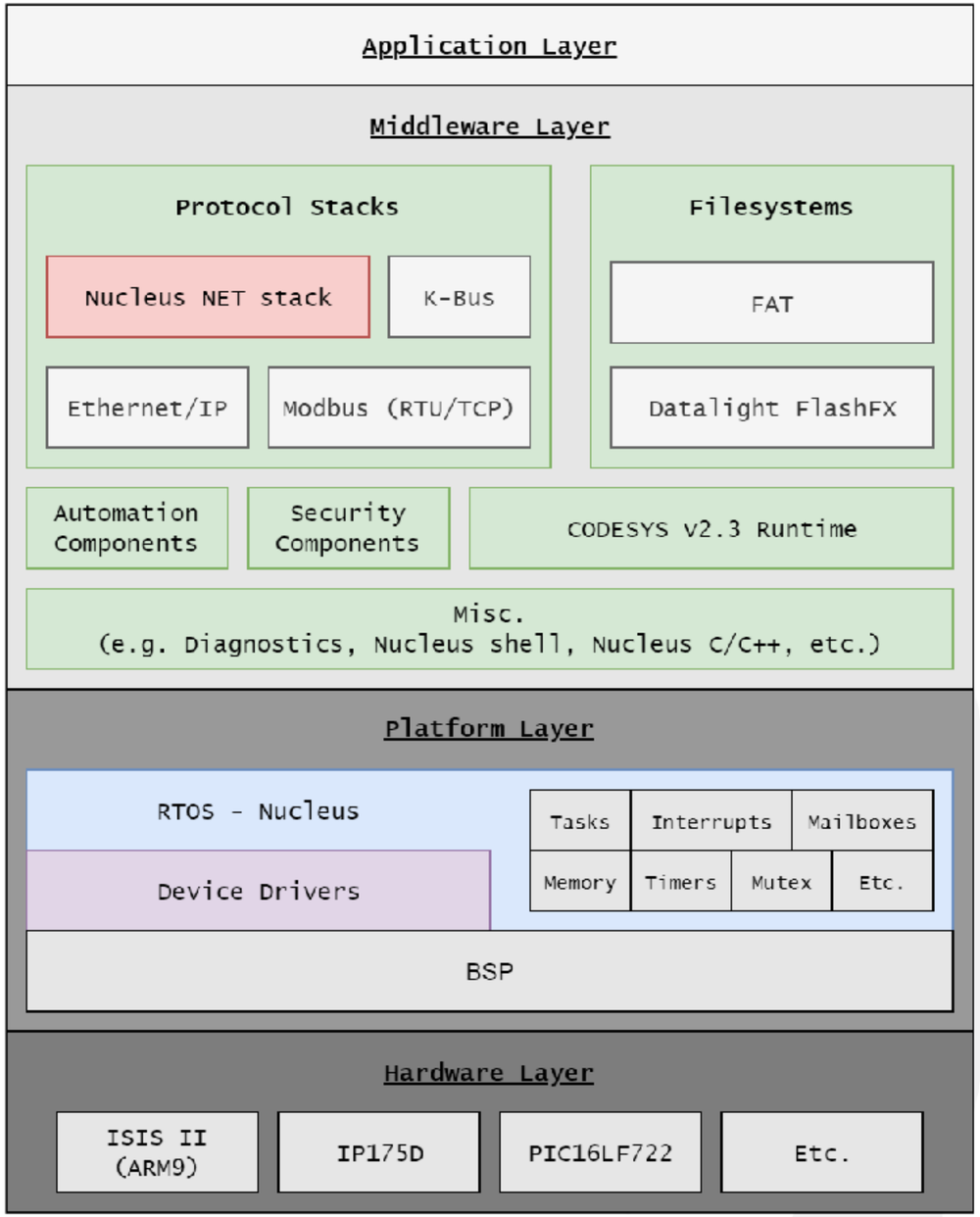
- Supervisor mode, no task separation → No need for privesc

* NUCLEUS:13, Dissecting the Nucleus TCP/IP stack – Forescout & Medigate Labs

* As of v01.09.25 (16)

Wago 750-852 Firmware*

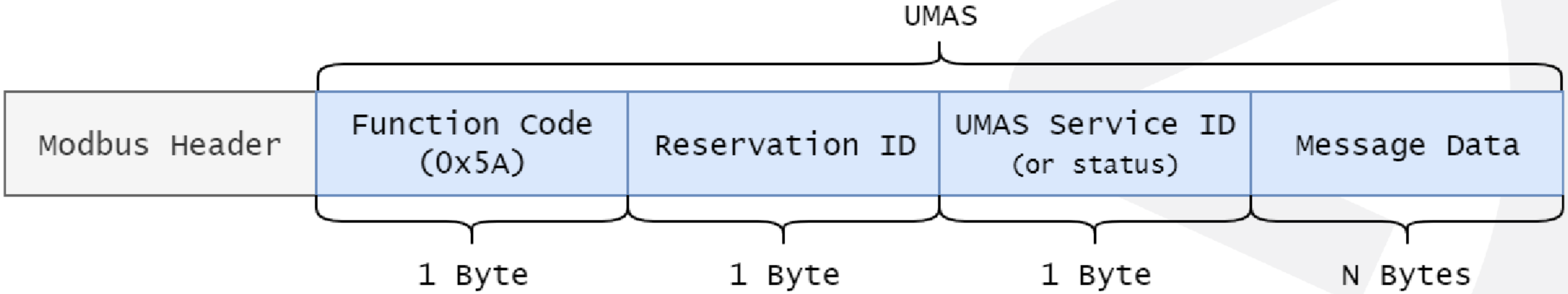
- Wago 750 Firmware ZIP
  - .bif: descriptive text file
  - .hex: Intel hex fw
- 60456550.hex → loaded at base address
  - Nucleus RTOS on ARM
  - No symbols
  - Use BinDiff / Diaphora / debug strs
- Create Nucleus Task for stable implant
  - Runs in background
- Hook Modbus TCP handler
  - Proxy incoming FC 0x5A to M340
  - Allow tunneling through coupler



Object PLC: Schneider Electric UMAS

- Proprietary SE Modicon engineering protocol under Modbus FC 0x5A
  - Much prior work, well-reversed (up to a point)[1,2,3,4]
  - Start/Stop PLC, download/upload logic, read/write memory blocks, etc.

- SE ControlExpert Security Features
  - Project File Encryption (AES-CBC-256)
  - Program/Safety password (weak crypto, client-side)[4]
  - UMAS historically unauth, introduced Application Password[2,3,4]



[1] Project Basecamp – Digital Bond
[2] The secrets of Schneider Electric's UMAS protocol – P. Nesterov et al.
[3] Going Deeper into Schneider Modicon PAC Security – G. Jian
[4] Examining Crypto and Bypassing Authentication in Schneider Electric PLCs (M340 / M580) – N. Miles

CVE-2021-22779: Auth Bypass

- Read secret from mem → Don't need to know pwd…

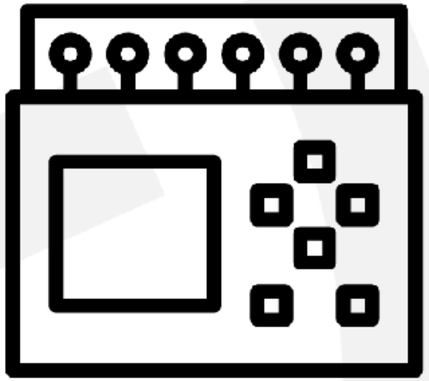
EnhancedCyberReserve v1

Read Memory Block:
secret = [B64(salt) + B64(SHA2(salt+pwd))]

Exchange Client & Server Nonce

Take Reservation: auth=SHA2(snonce + secret + cnonce)

Authenticated Request:
[SHA2(SHA2(hwid+cnonce) + msg + SHA2(hwid+snonce))]
[nested UMAS]

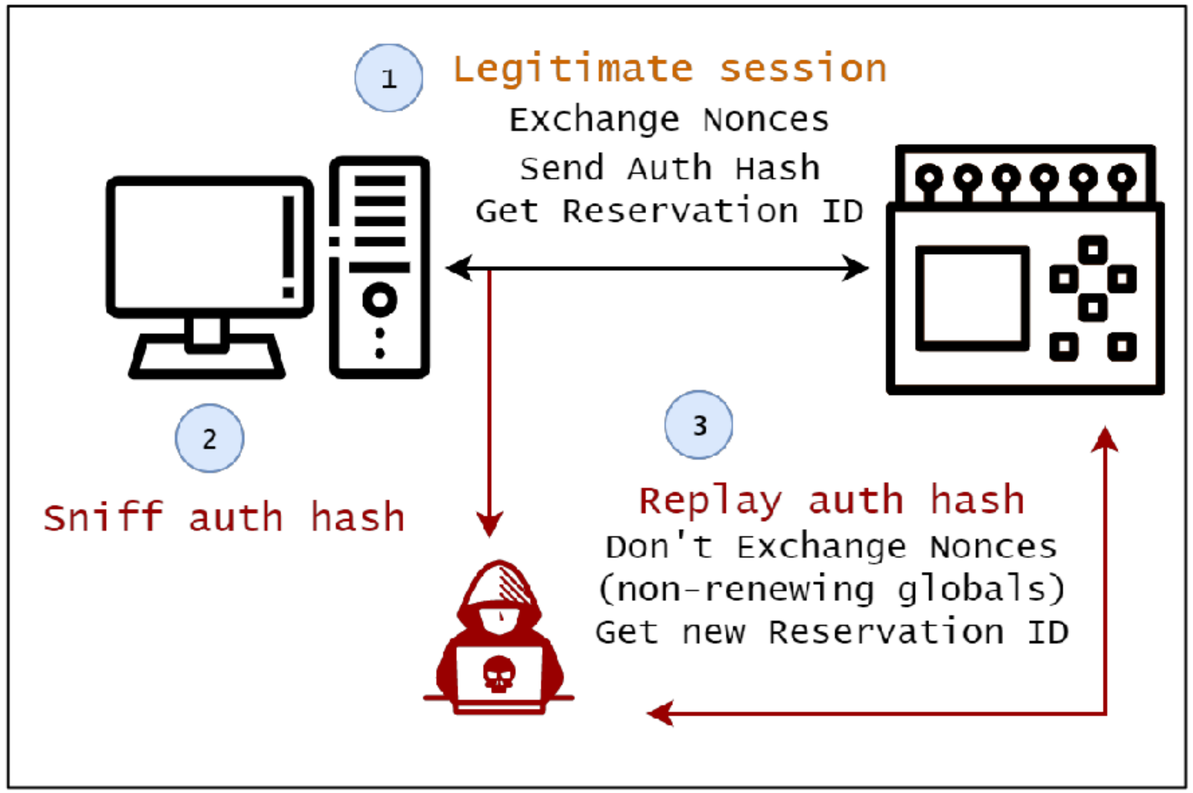


[1] Project Basecamp Digital Bond, [2] The secrets of Schneider Electric's UMAS protocol P. Nesterov et al., [3] Going Deeper into Schneider Modicon PAC Security G. Jian
[4] Examining Crypto and Bypassing Authentication in Schneider Electric PLCs (M340 / M580) N. Miles, [5] ModiPwn G. Kauffman et al.

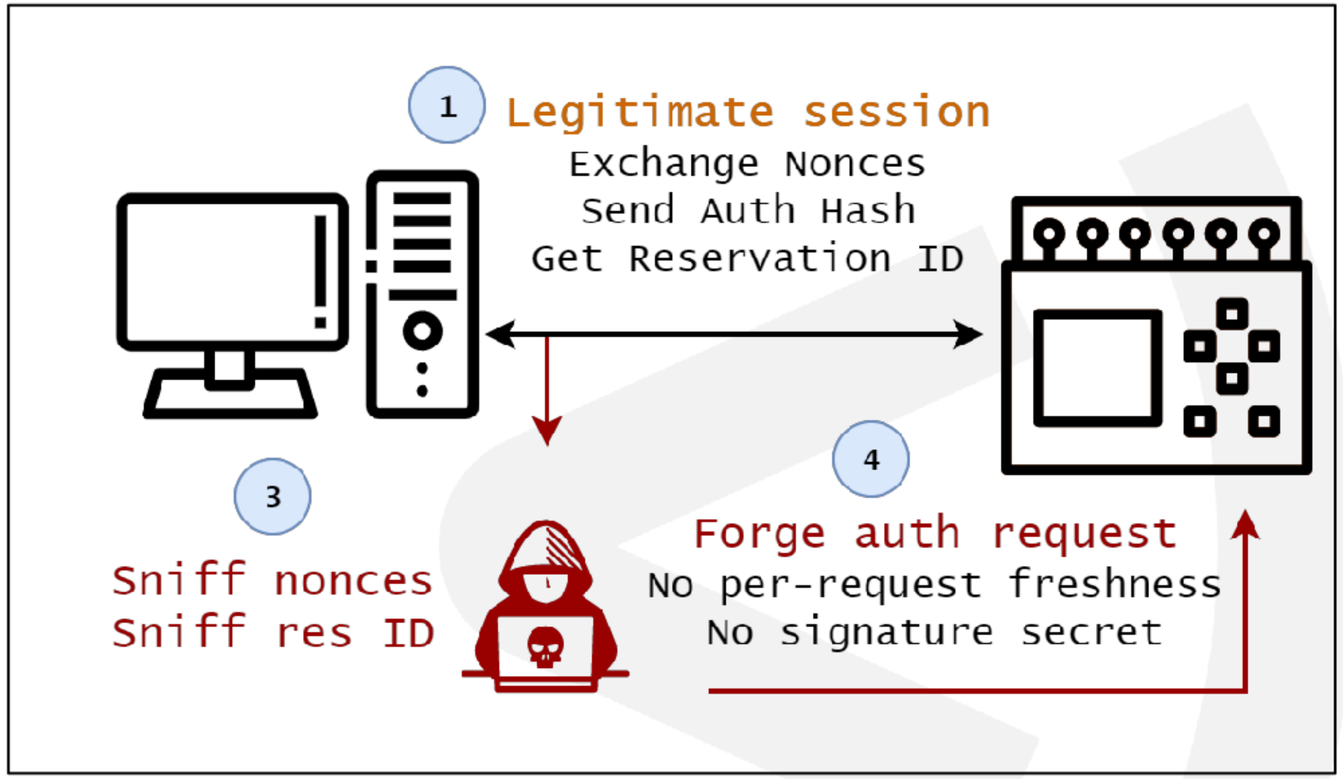
CVE-2022-45789 – Authentication Bypass*

- Patch → PW no longer in mem block, **however**

Reservation Replay



Authenticated Request Forgery



* Affects latest M340 and M580 CPU module FW, see SEVD-2023-010-06

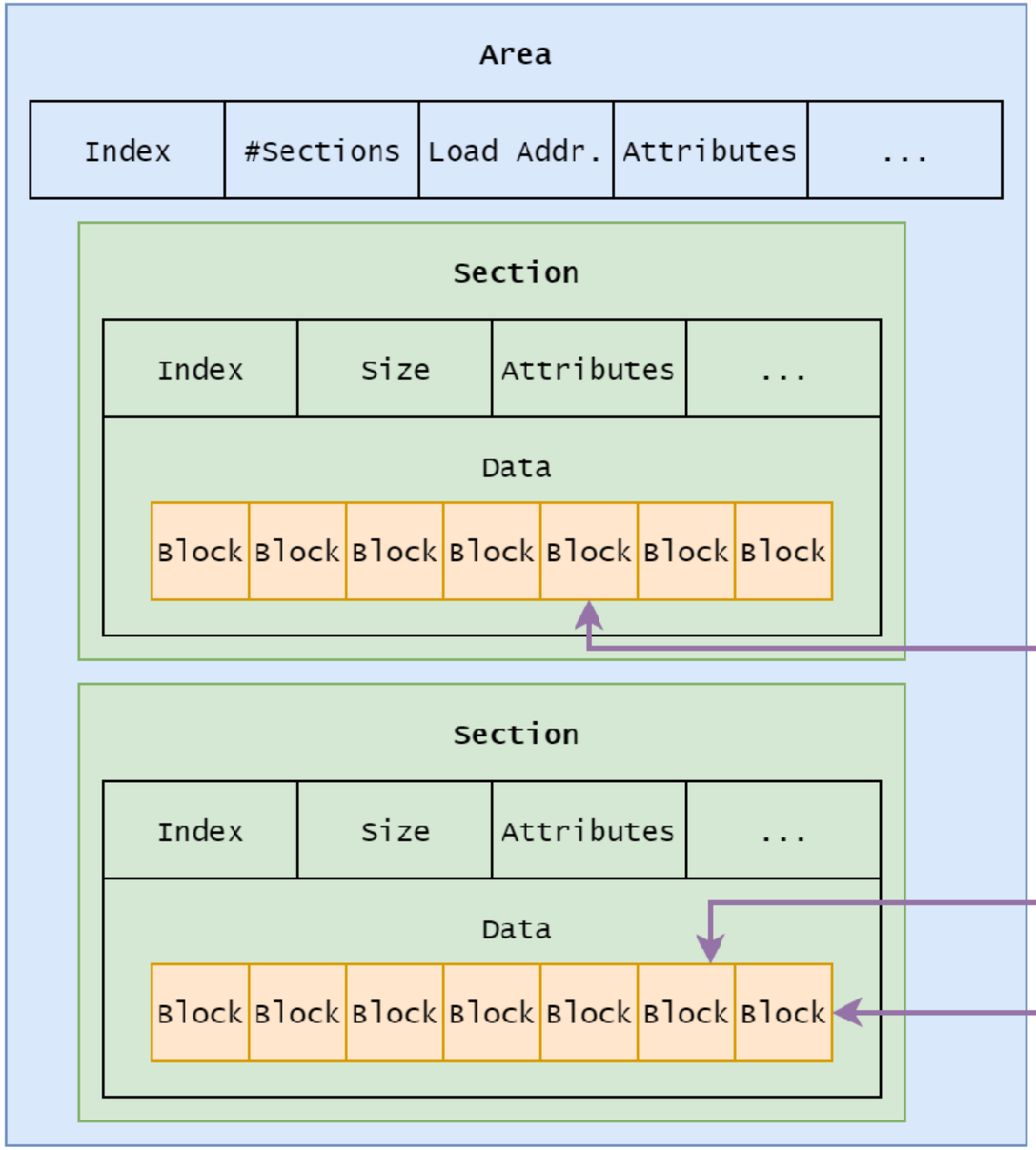
Route to CPU Module RCE

- Different approaches in prior work
  - UMAS: Download logic (0x31) [1,2], vulnerable messages[3,4]
  - TCP/IP stack RCE (M580 but not M340)[5]

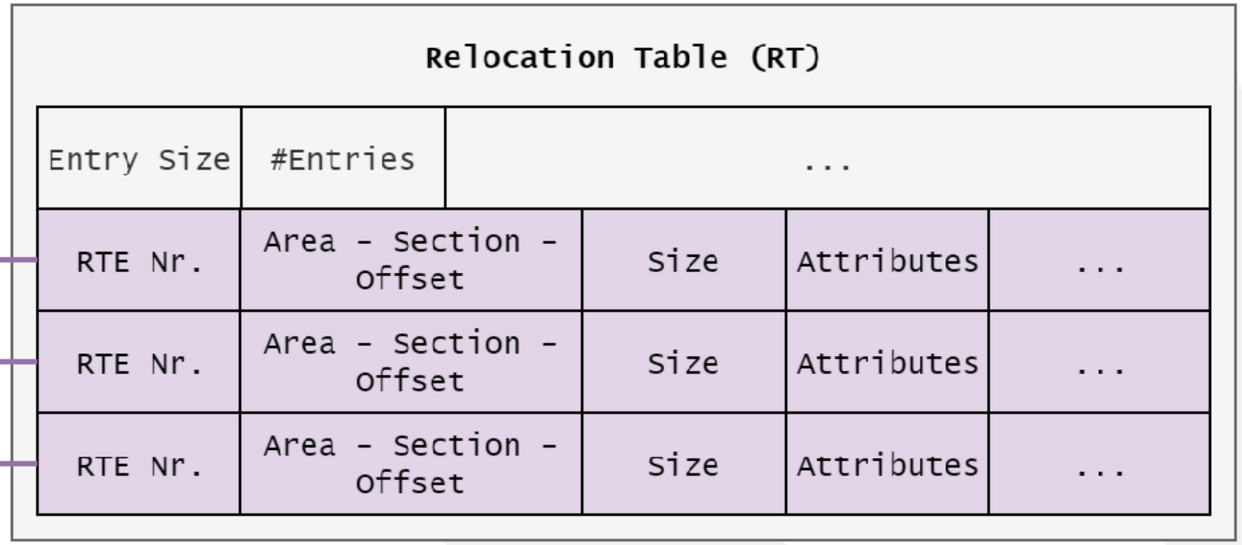
- Want method allows *hotpatching* on *updated* PLC
  - No logic restarts
  - DFIR hostile (project checksums, invisible in source)
  - Using *obscure* protocol features to evade most IDS

[1] TALOS-2018-0742 J. Rittle
[2] Applying a Stuxnet Type Attack to a Modicon PLC – F. Dola
[3] Going Deeper into Schneider Modicon PAC Security – G. Jian
[4] ModiPwn – G. Kauffman et al.
[5] Exploring and Exploiting PLCs with Urgent/11 Vulnerabilities – B. Hadad et al.

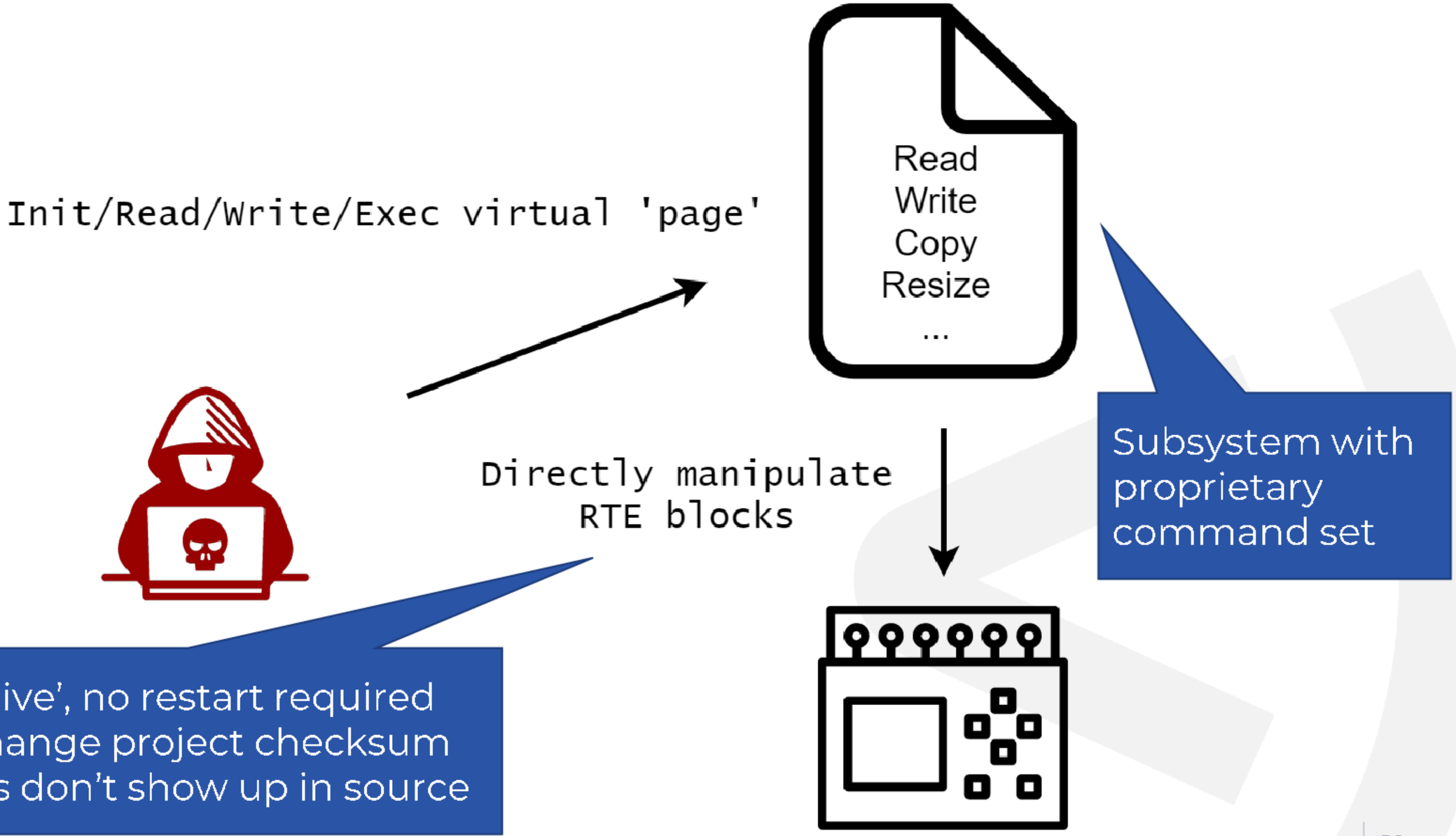
Background: Modicon Application Binary File (APX)



- Block Types
  - Data / Exec / Upload Info / FB Data / Constant / etc.



Unexplored UMAS CSA Requests (0x50)



- Happens 'live', no restart required
- Doesn't change project checksum
- Exec mods don't show up in source

CVE-2022-45788 – Modicon CPU RCE*

Can't write directly to code blocks

But can *copy* to code blocks (permission check set to 'ignore')

Code Block

Data Block

3 Get RCE when block executes as part of logic

2 Copy from data block to code block (find cave or expand block, then hijack control flow)

1 Write payload to data block (find cave or expand block)

```
if ( !ignore )
{
  if ( rte_ptr )
  {
    if ( (rte_ptr->attr & 0x10000) != 0 )
    {
      return 0x9191;
    }
    else
    {
      blocktype = rte_ptr->attr & 0xF;
```

* Affects latest M340, M580, M1E, MC80, Quantum, Premium CPU module FW, see SEVD-2023-010-05

SE BMXP3420302 Firmware*

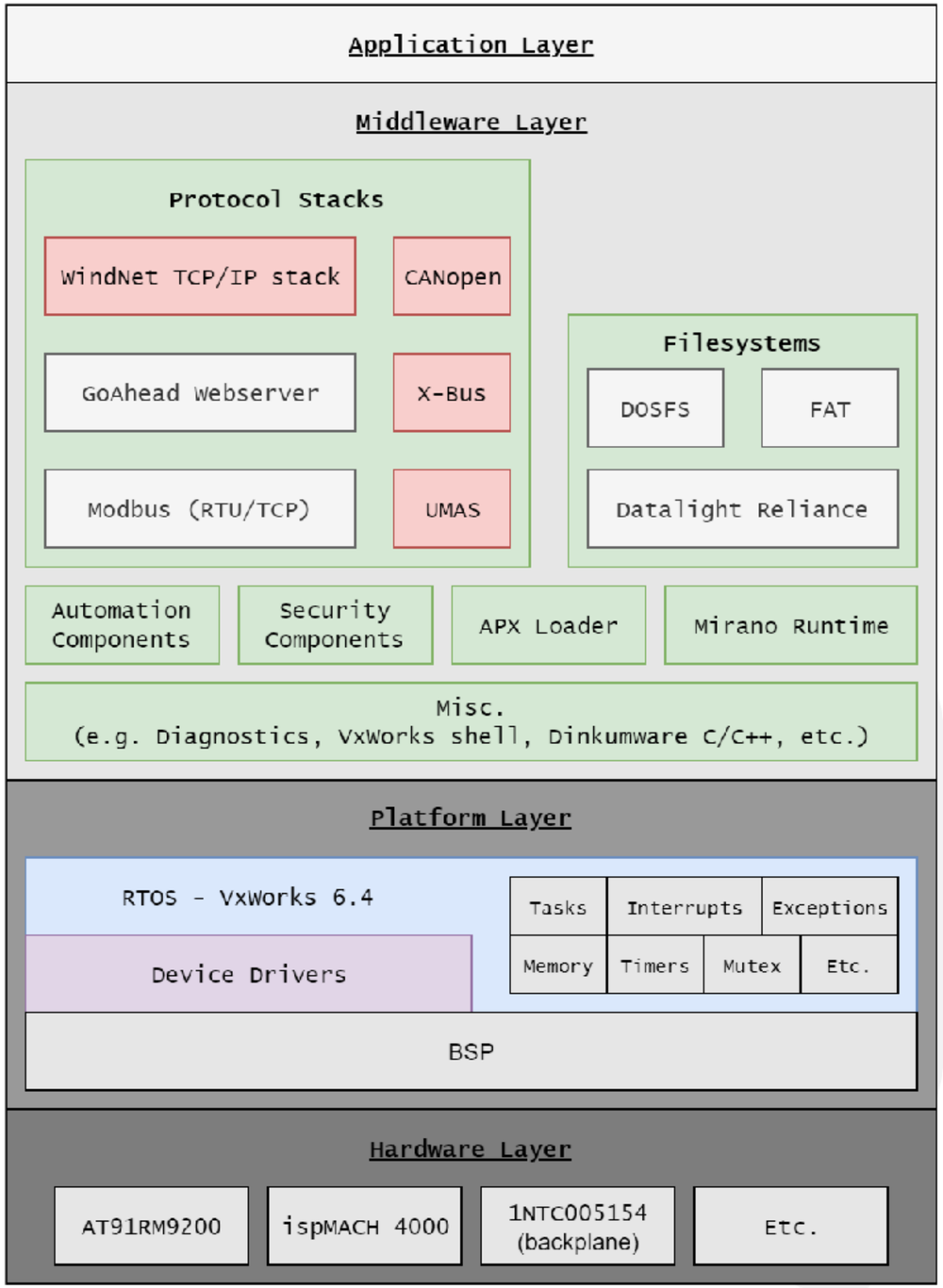
* As of v3.50

- SE Firmware LDX = ZIP

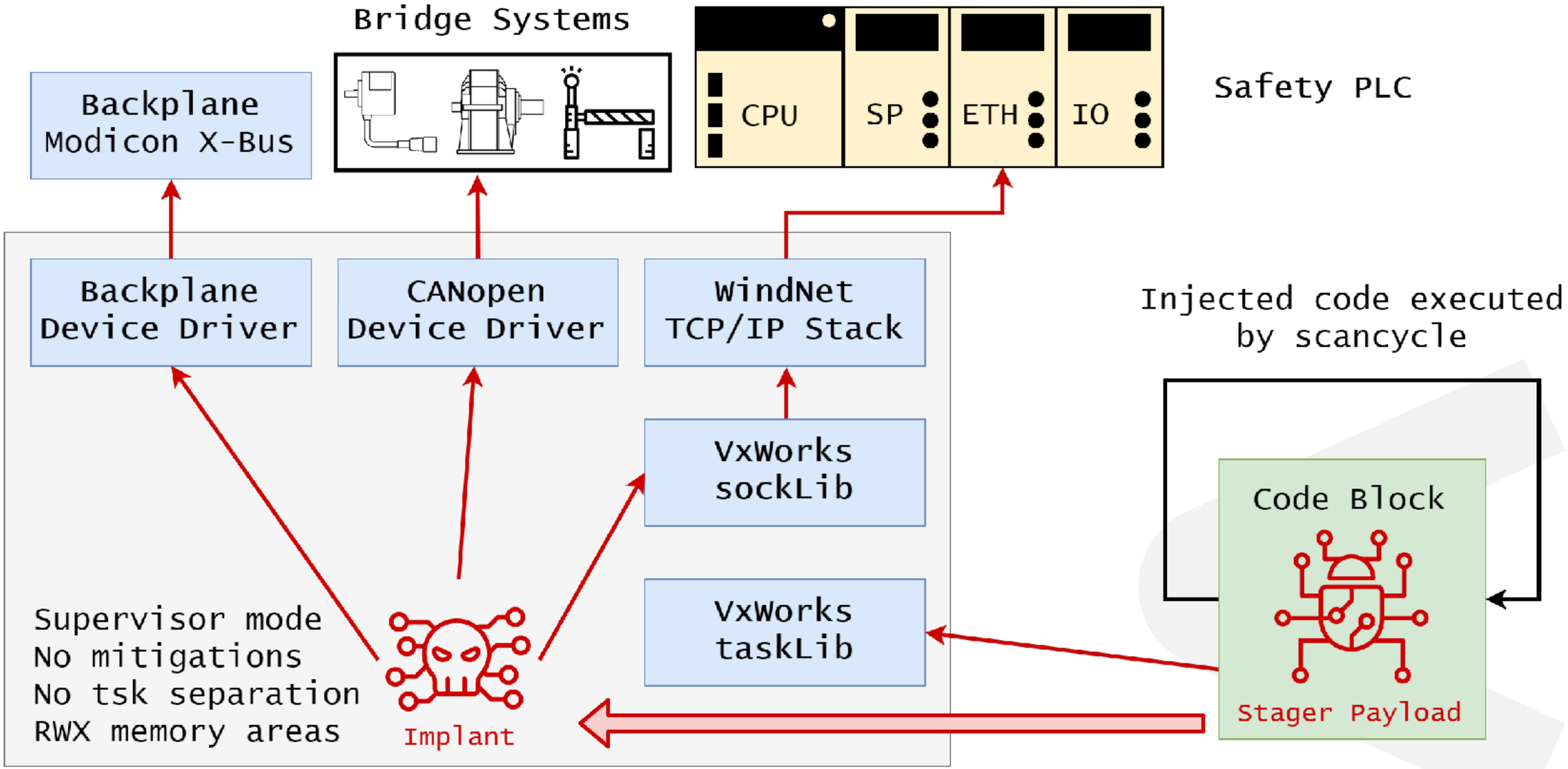
- vxWorks_bmx*.bin → UNITYM binary
  - Segment base @ 0x20000000
  - FW code start @ 0x20010110
  - Runtime base @ 0x28000000
  - VxWorks 6.4 on ARMv4 (so no XN)
  - Manually reconstruct symbol table

- Runtime exec blocks via sas_UserCodeExec
  - Scancycle timer is in the way
  - Hook triggerable func to escape

```
v4 = kl_userTimeEn(result);
v5 = sas_UserCodeExec(v4);
kl_userTimeDis((int)v5);
```



Stager Payload & Implant



Modicon CPU Module FW

Relocate implant code + Spawn dedicated task
Cleanup manipulated blocks (anti-DFIR)

* BMXP3420302 as of v3.50

CANopen payload

- Talk to **M340 CANopen API**, use CiA funcs

```
can_SWrite_SDO(ND, 0x1F51, 1, START_BOOT,
can_SWrite_SDO(ND, 0x1F51, 1, ERASE_FLASH,
...
can_SWrite_SDO(ND, 0x1F50, 1, block[i],
```

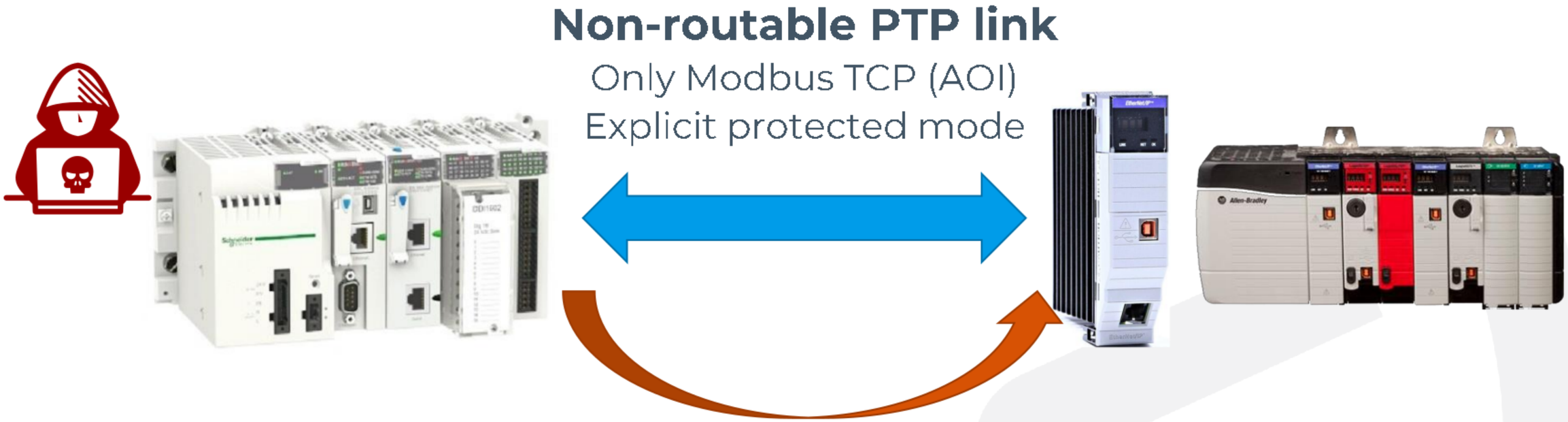
Index	SDO Name
0x1023	OS CMD[2]
0x1024	OS CMD Mode[2]
0x1025	OS Debugger[2]
0x1026	OS Prompt[2]
0x1F50	Download Program[3]
0x1F51	Program Control[3]

- **RCE via SDO: override firmware (safety) limits**
  - In-band code dndl – trigger bootloader via NMT/SDO
  - Memory read/write – hotpatching RCE
  - If auth at all: (static) 32-bit value written to some SDO

[1] CAN in Automation (CiA) 302 2, [2] CAN in Automation (CiA) 301, [3] CAN in Automation (CiA) 302 3

Object PLC → Safety PLC Ethernet module

Cannot talk directly to GuardLogix CPU module or route CIP



Exploit N-day vuln in TCP/IP stack for RCE on Ethernet Module → hop to rest of SIS

Allen-Bradley GuardLogix Safety PLC
1756-EN2T/D Ethernet Module

CVE-2019-12256* on Allen-Bradley 1756-EN2T/D

- Send malformed IP options (URGENT/11) via VxWorks raw sockets
  - Multiple Source Record Route (SRR) opts generate ICMP error response
  - Stack buffer overflow (opts copied to response without validation)

```
srr_opt->ptr = 4;
while ( offset_to_current_route_entry > 0 )
{
  memcpy((char *)srr_opt + (unsigned __int8)srr_opt->len, current_route_e
  current_route_entry -= 4;
  offset_to_current_route_entry -= 4;
  srr_opt->len += 4;
}
memcpy((char *)srr_opt + (unsigned __int8)srr_opt->len, icmp_param + 12,
v18 = srr_opt->len + 4;
```

- Only XN enabled
  - Pick SRRs to align stack overwrite
  - Write-4 ROP + stack fixup → cont. exec
  - Large unused RWX 'LOAD' segment
  - Chop shellcode into chunks of 4 → write to RWX seg via ROP chain

- Only slight diffs with Armis exploit* against 1756-EN2TR/C
  - ROP chain construction, RWX/gadget/func addrs

- Supervisor mode, no task separation → No need for privesc
  - Spawn VxWorks task for stable implant

* Exploring & Exploiting PLCs with URGENT/11 – B. Hadad et al.

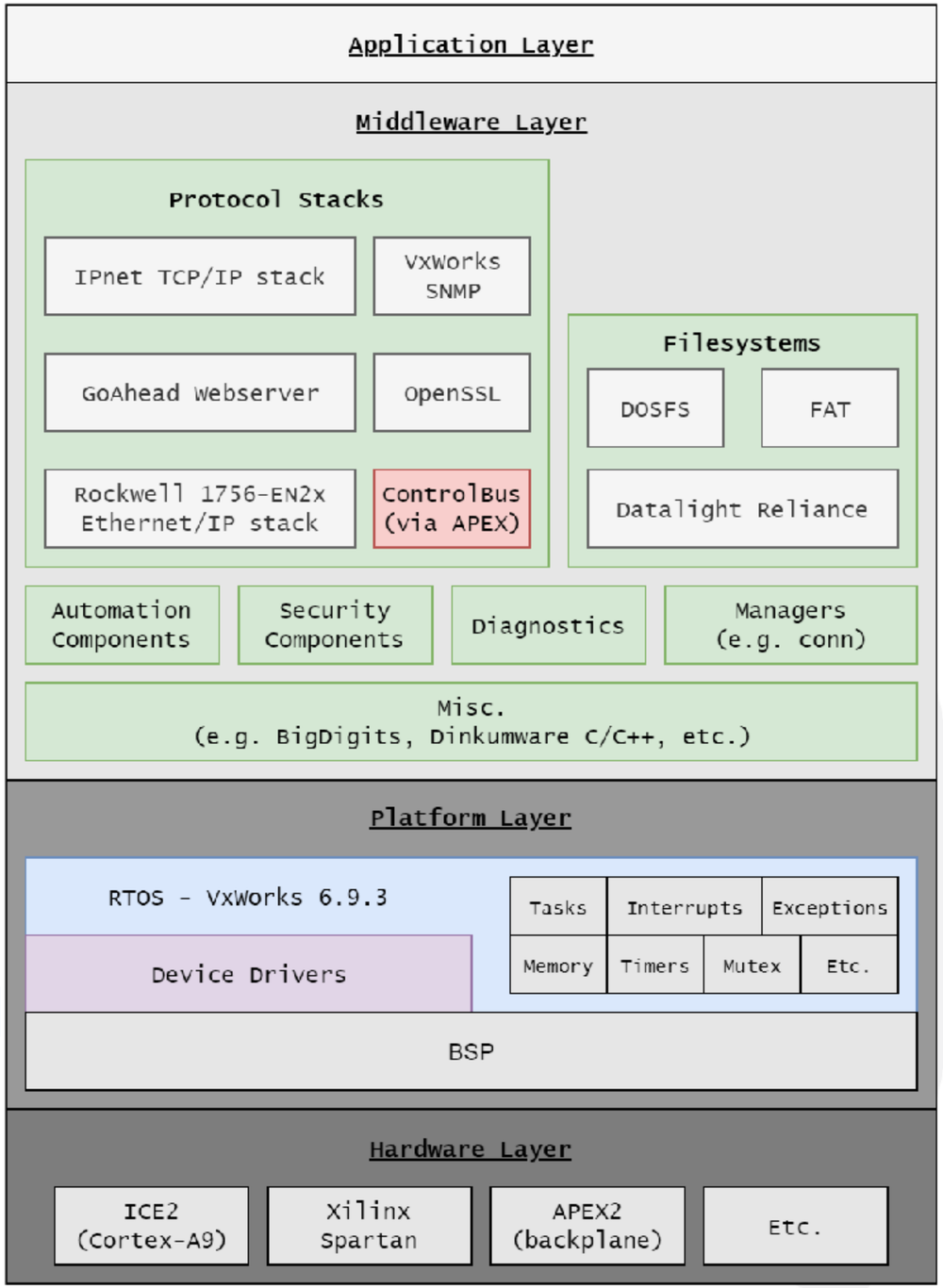
AB 1756-EN2T/D Firmware*

- Allen-Bradley Firmware ZIP
  - .nvs: descriptive text file
  - .plt: binary fw
  - .der: certificates

- PN-497069.plt → ELF binary
  - Segments pre-loaded
  - VxWorks 6.9.3 on ARM
  - Manually reconstruct symbol table
  - Implant talks to display & backplane drivers

```
symbol <0, aAccessDescript_0, ACCESS_DESCR
                              ; DATA XREF: usrSt
                              ; usrStandaloneIni
symbol <0, aAccessDescript_1, ACCESS_DESCR
symbol <0, aAccessDescript_2, ACCESS_DESCR
symbol <0, aAcmAllocateele, ACM_AllocateEl
symbol <0, aAcmAllocatetar, ACM_AllocateTa
```

```
_ZN12bsp_ApexImpl12DownloadCodeEv
_ZN12bsp_ApexImpl13StartFirmwareEv
_ZN12bsp_ApexImpl13InitBackplaneEb
_ZN12bsp_ApexImpl9IsFaultedEv
_ZN12bsp_ApexImpl13IsCbaAssertedEv
_ZN12bsp_ApexImpl13IsCbbAssertedEv
```



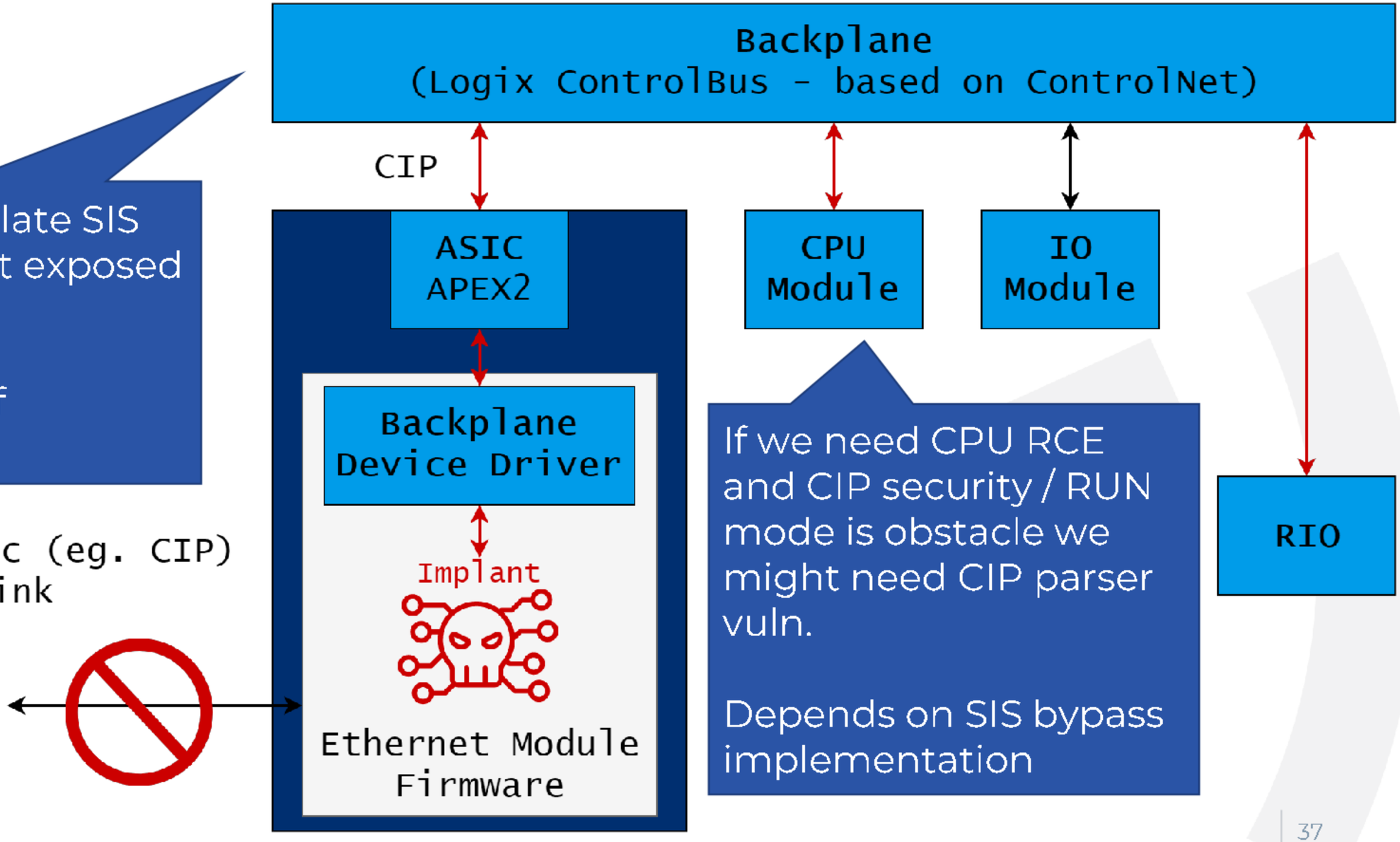
* As of v11.01

Move across Safety PLC backplane

Use CIP to manipulate SIS bypass settings not exposed outside Safety PLC

Also the usual stuff (eg modify logic)

No routable traffic (eg. CIP) via PTP link



If we need CPU RCE and CIP security / RUN mode is obstacle we might need CIP parser vuln.

Depends on SIS bypass implementation

Disclosure

* SEVD-2023-010-05, SEVD-2023-010-06

- Coordinated disclosure with Schneider Electric
  - Issues reported in April and July 2022
  - Advisories* released in January 2023, updated in March 2023

- CVE-2022-45788 (RCE)
  - Remediations available for M580 (excluding safety), M1E
  - Mitigations for others

- CVE-2022-45789 (auth bypass)
  - Currently mitigations only

- We suggested retrofit fix: Secure Remote Password(SPR) + HMAC
  - Auth user to PLC with SRP (zero-knowledge, MitM-resistant, discrete-log based)
  - Derive HMAC key from shared SRP key K
  - Sign messages with HMAC

(some) Mitigation, Detection, and DFIR advice

Attack Step	Controls
Wago 750 implant	• Alert on UMAS to non-Modicon devices • Monitor Modbus TCP statistics
UMAS Auth Bypass (CVE-2022-45789)	• Restrict UMAS flow to EWS (IP ACLs, FW) • Look for auth request (SVC 0x38) without none exchange (SVC 0x6E)
UMAS RCE (CVE-2022-45788)	• Alert on UMAS CSA (SVC 0x50) • Monitor watchdog errors • Upload PLC project, extract & carve APX, look for malicious ARM shellcode
1756-EN2T* RCE (CVE-2019-12256)	• Monitor IP & assert statistics
1756-EN2T* implant	• Monitor task statistics

Task Statistics

Name	Entry Point	ID	Priority
tJobTask	1e7208	efc4e8	0
tExcTask	1e69fc	7f85b8	0
tErfTask	10b9c	f00f70	10
tLogTask	1e76bc	f04110	0
tNet0	1bdc8	f11e00	50

IP Statistics

Forwarding	1
Default TTL	64
In receives	812
In header errors	4
In address errors	0
Forwarded datagrams	0

- For full overview, see report*

* https://www.forescout.com/resources/l1-lateral-movement-report

Conclusions

- There's likely a lot of network 'crawl space' that's not on your radar
- If a L1 device sits between segments, it needs a perimeter security profile
- Stop treating certain links (serial, PTP, couplers, non-routable) as if they're immune
- Impact of compromise not limited to explicit link capabilities or 1st order connectivity
- With *deep access*, things become possible which change potential impact

Thank you.

FORESCOUT

Full report

https://www.forescout.com/resources/l1-lateral-movement-report