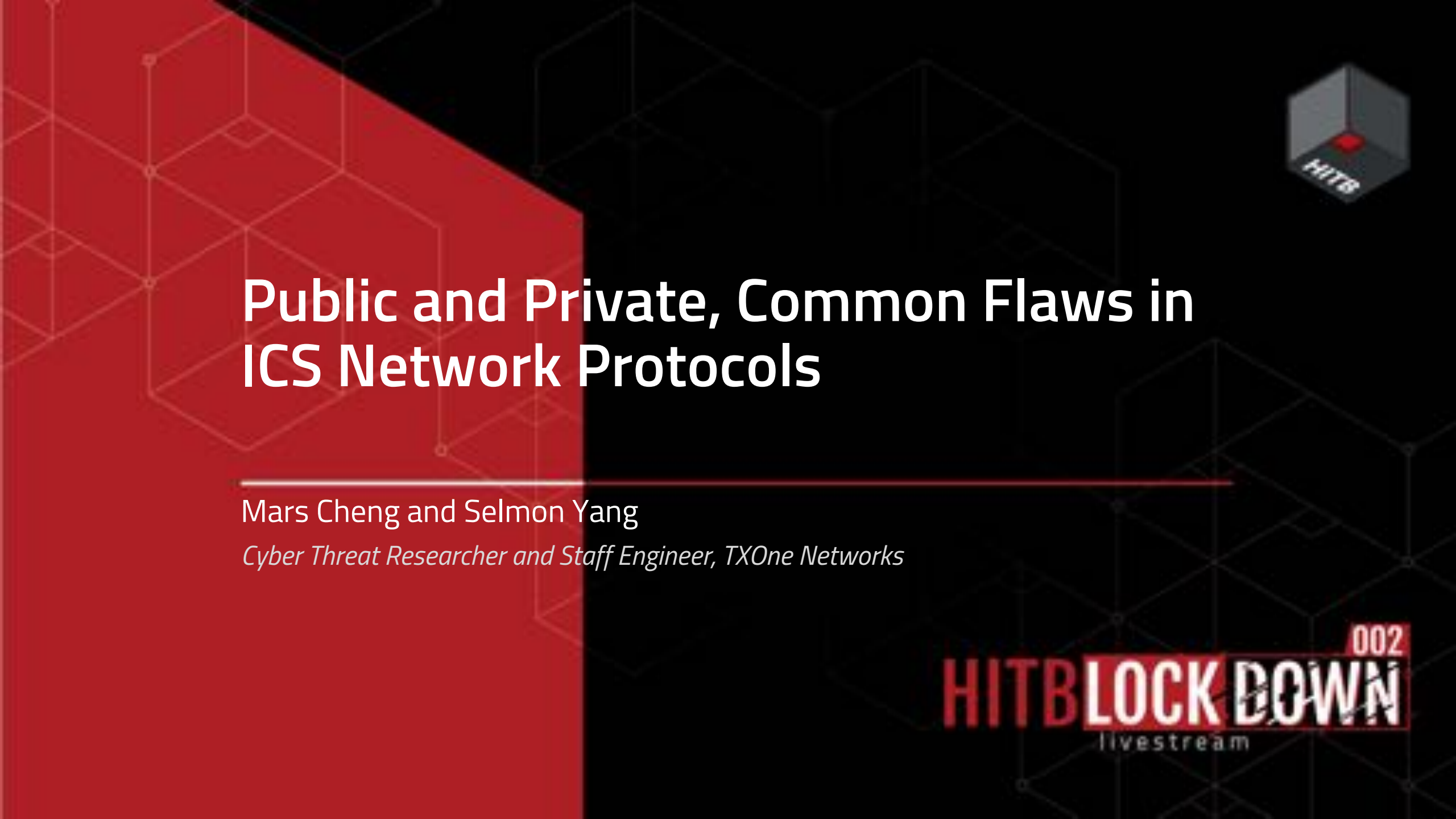
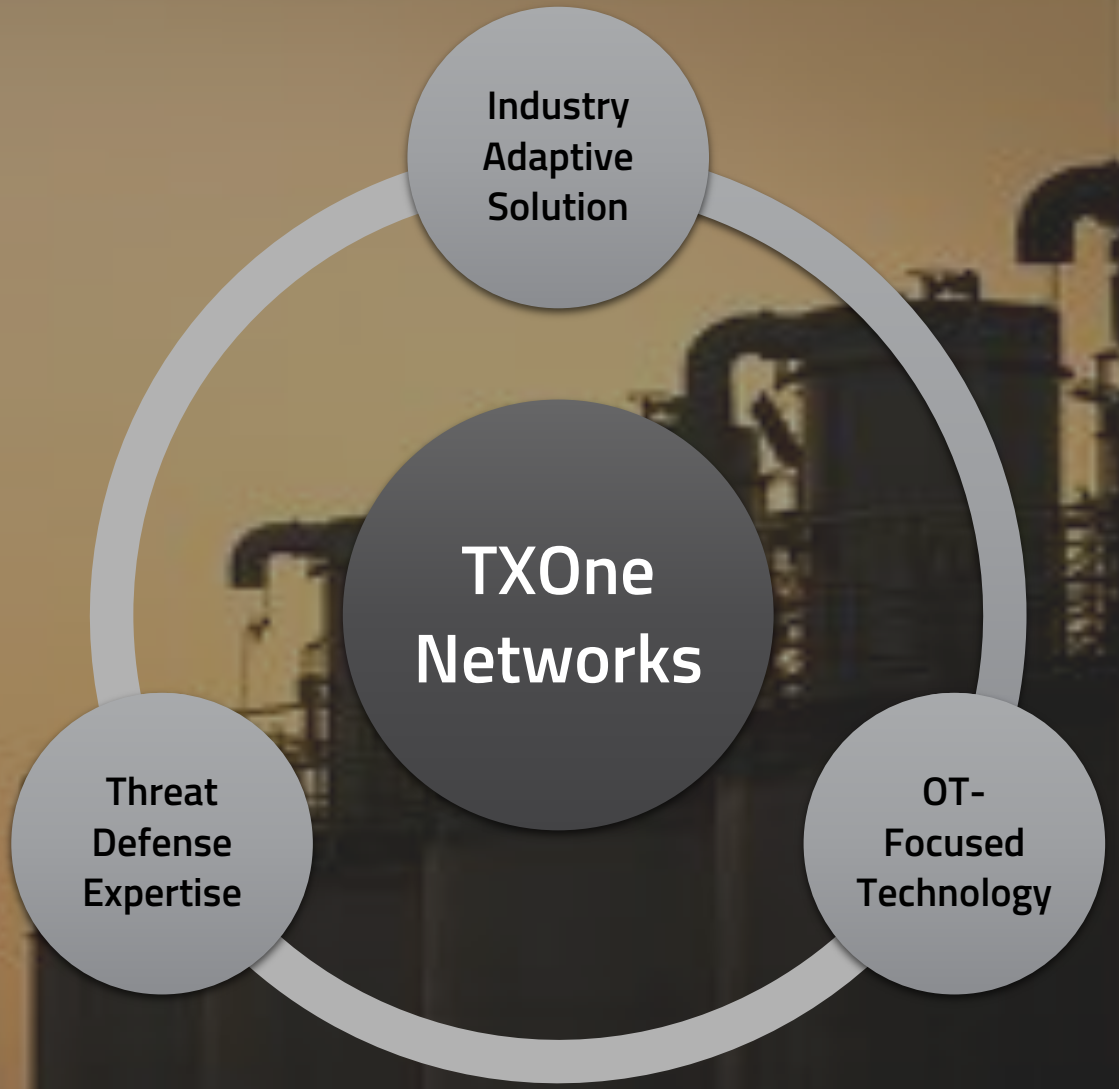

HITB
Public and Private, Common Flaws in ICS Network Protocols
Mars Cheng and Selmon Yang
Cyber Threat Researcher and Staff Engineer, TXOne Networks
002
HITB LOCKDOWN
livestream

WHO WE ARE

A Joint venture company of
Trend Micro Inc. and Moxa Inc.

30 years+ Cybersecuirty Threat Intelligent
30 years+ OT Network Expertise



To accelerate the industrial world to secure automation and data exchange

txOne networks

Who we are?

Mars Cheng

Cyber Threat Researcher

ICS/SCADA Security Research
Threat Hunting
Web, App, IoT, ICS/SCADA Penetration Testing
Applied Cryptography

Selmon Yang

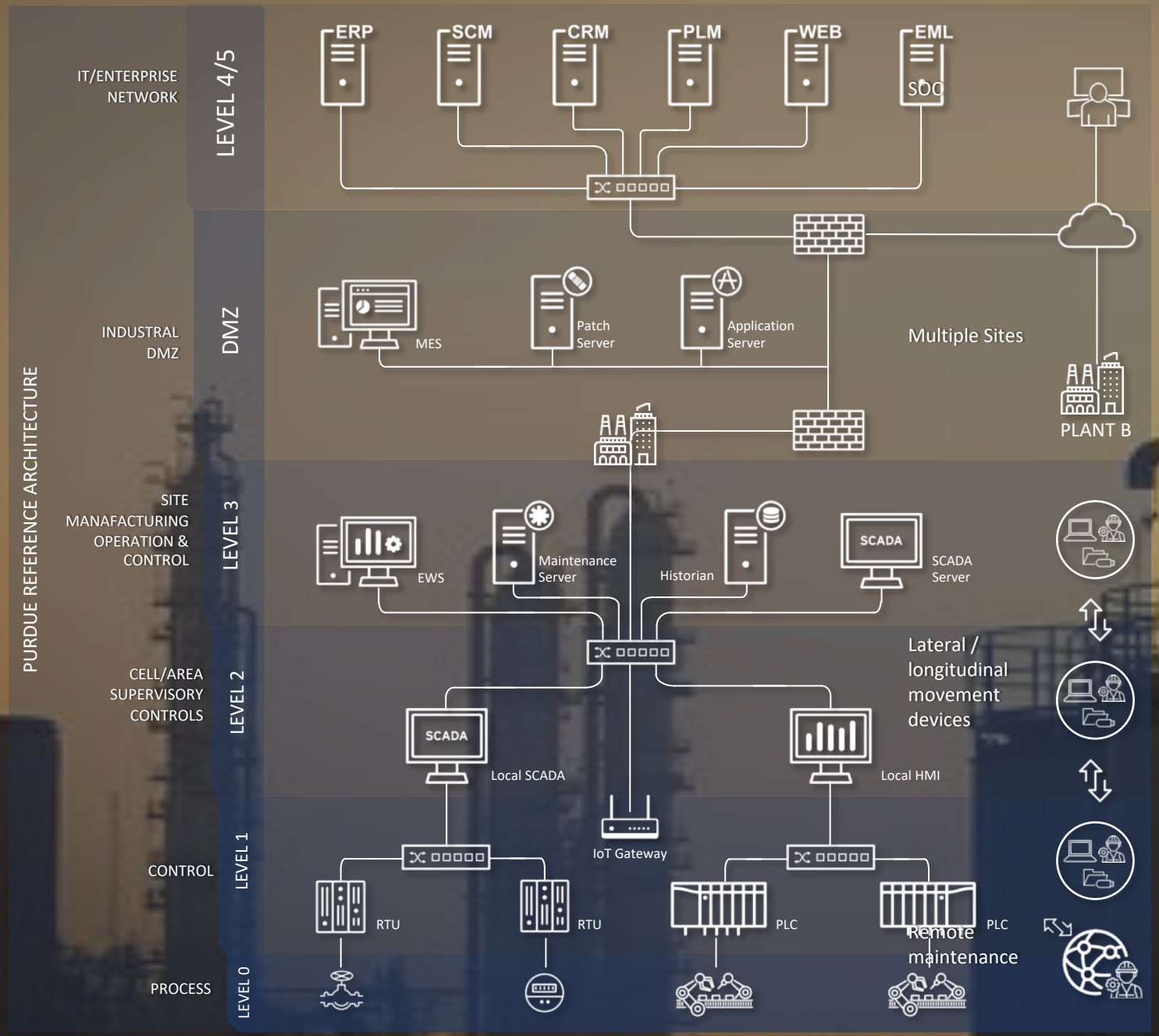
Staff Engineer

IT/SCADA Protocol Parsing
Linux Kernel Programming
Honeypot Deployment & Optimization

Outline

- ICS Architecture and Attack Vectors
- Public and Private: ICS Protocols
- Common Flaws in ICS Protocols
- How to Work Against ICS Network Protocol Attacks

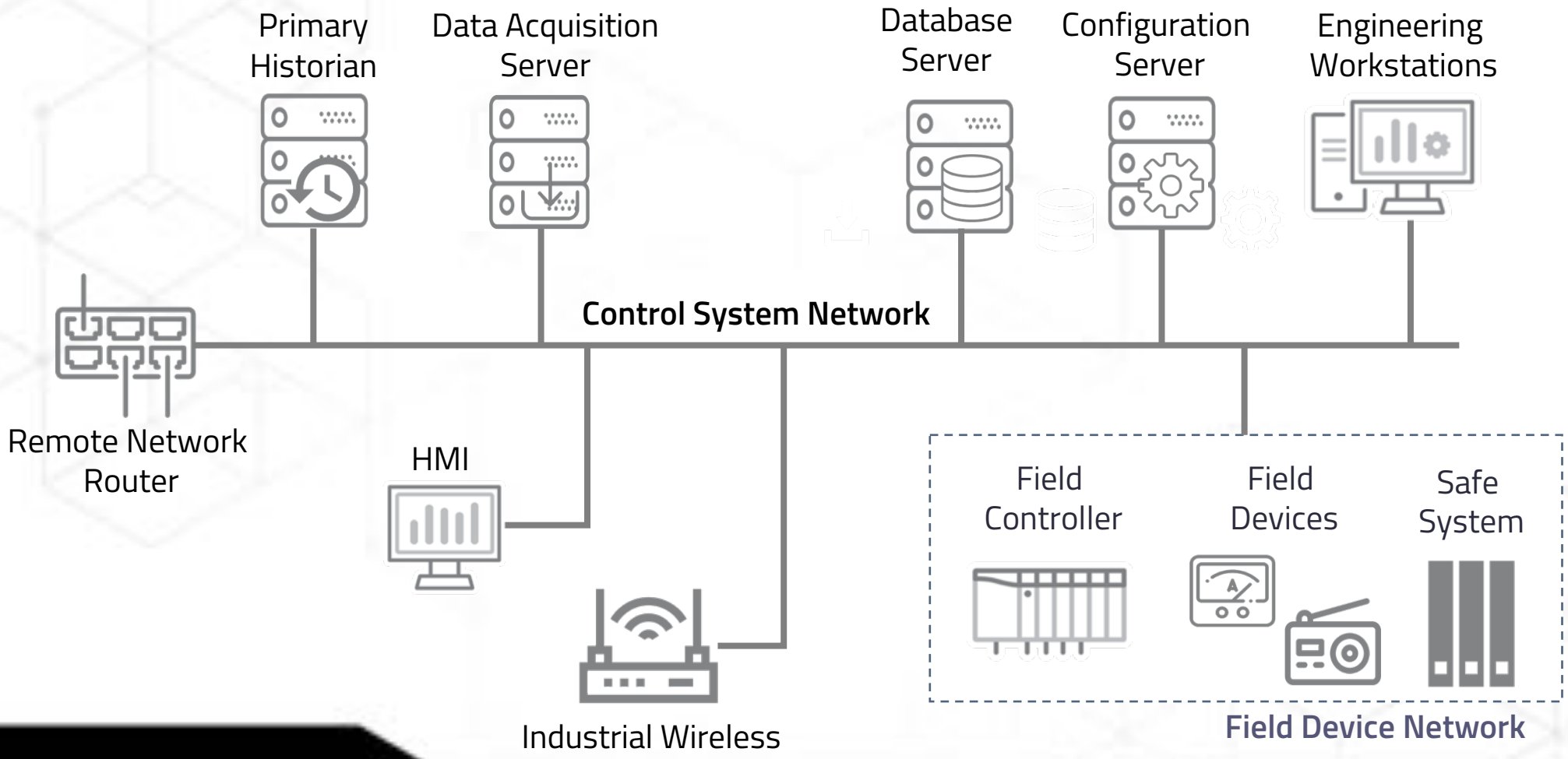
ICS Architecture and Attack Vectors



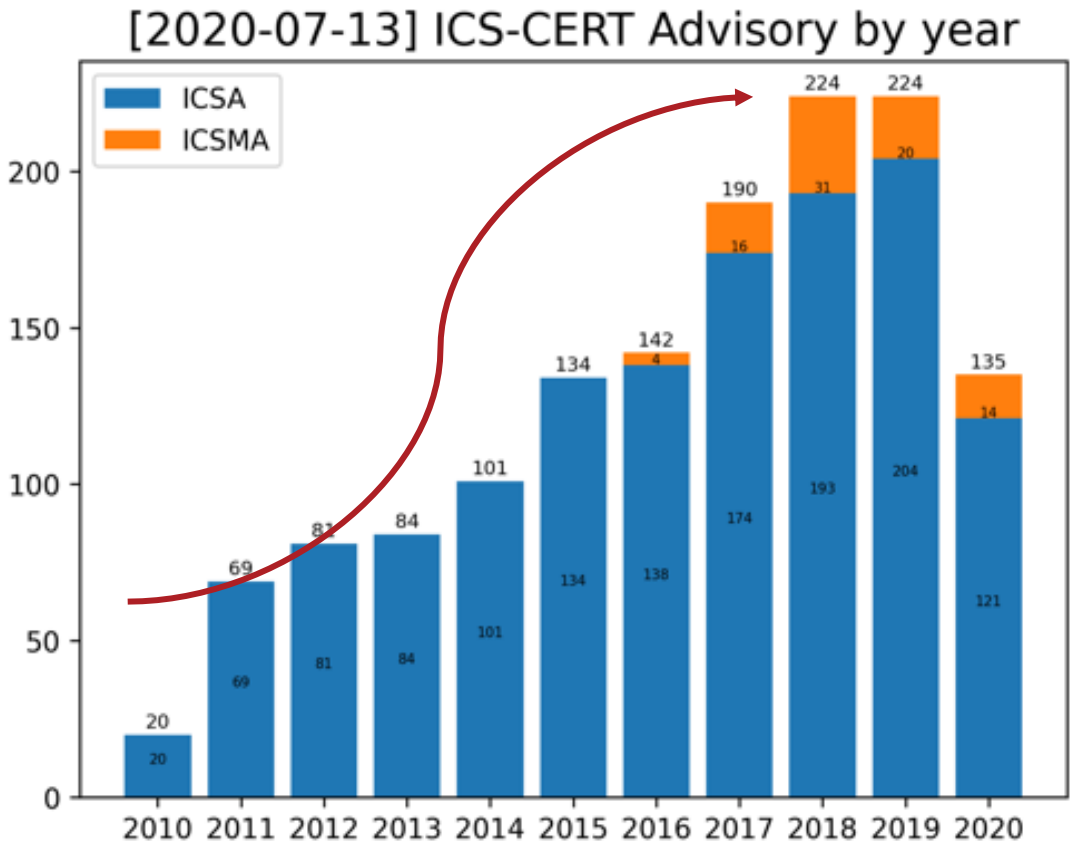
Information Technology (IT)

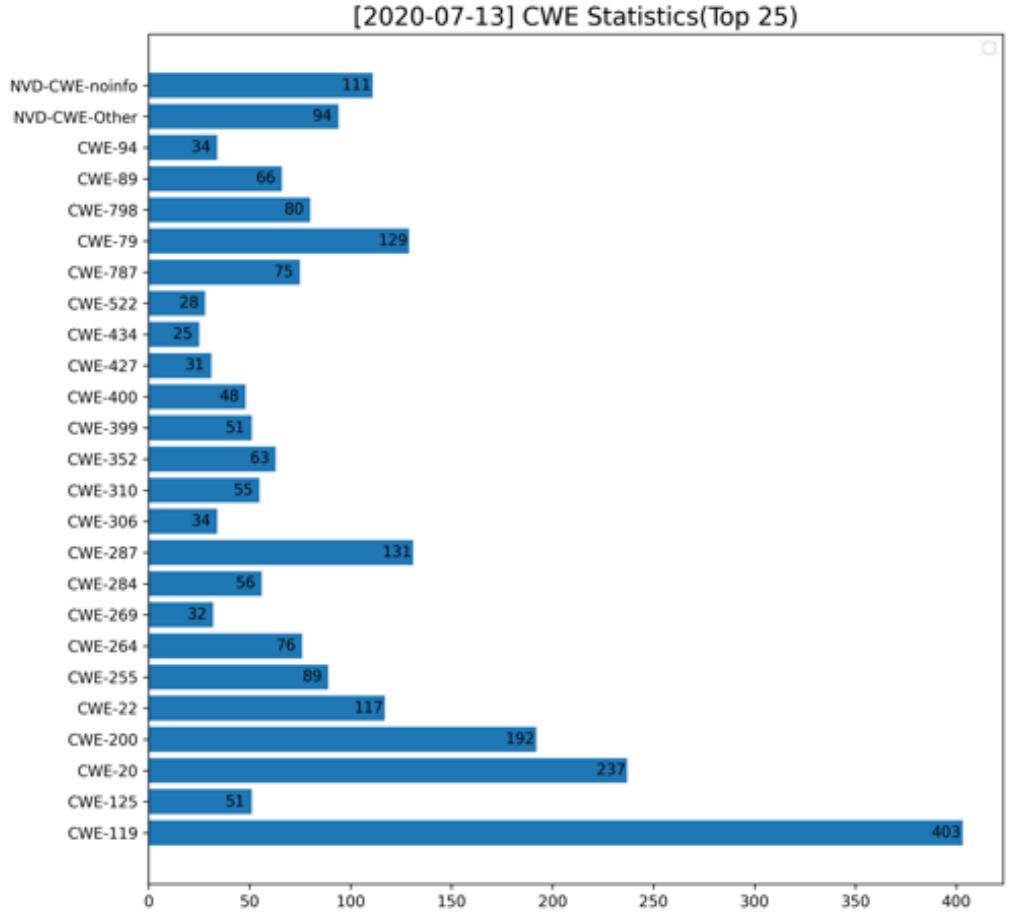
Operational Technology (OT)

Common ICS Architecture



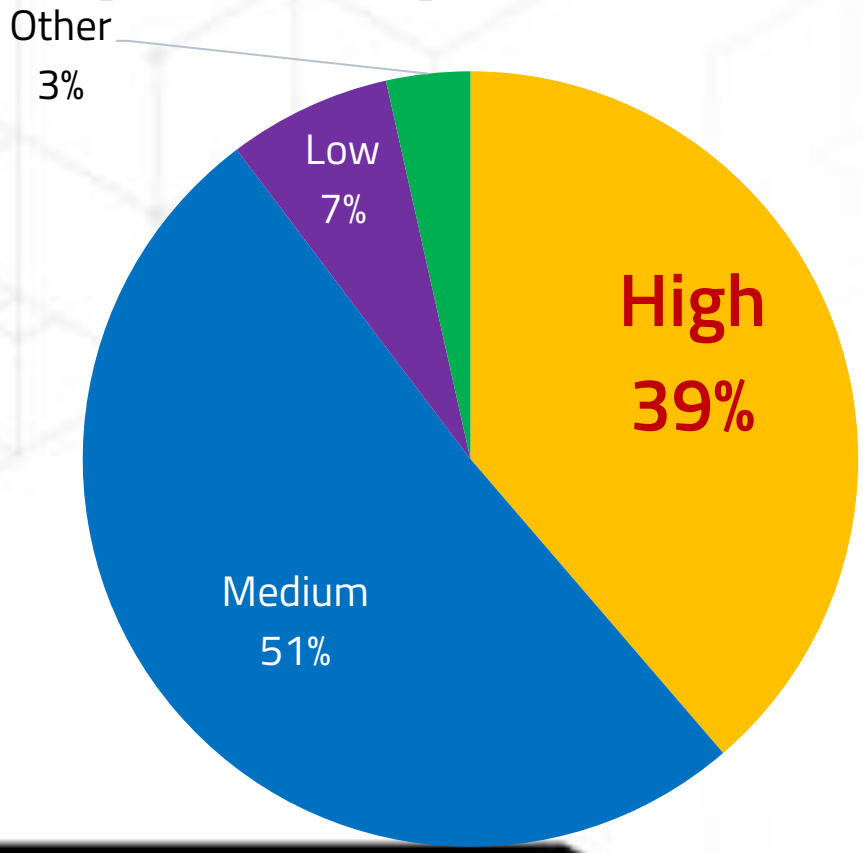
ICS-Related Vulnerabilities Information



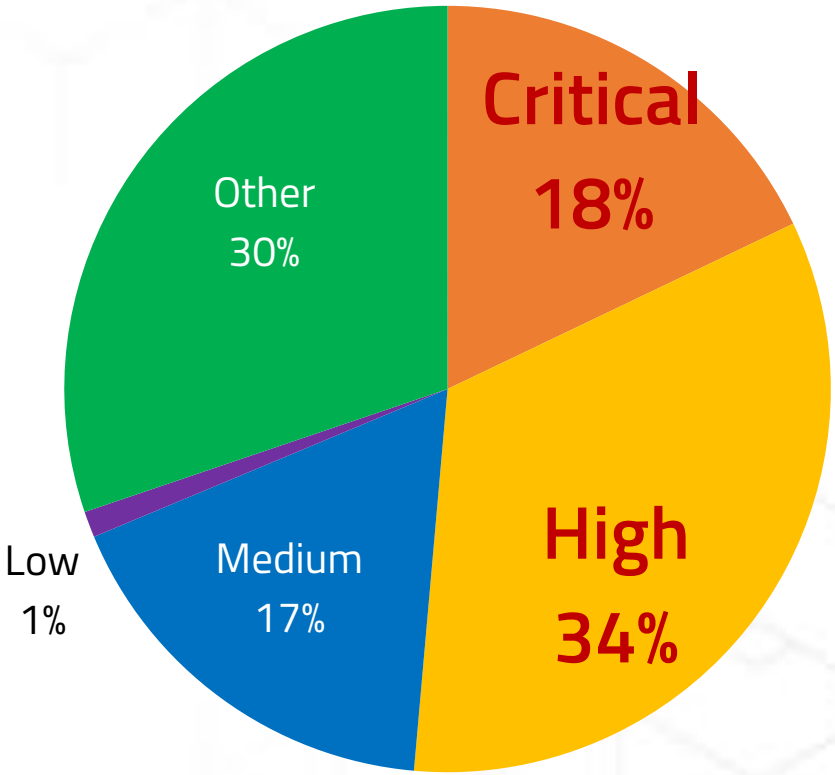


ICS-Related Vulnerabilities Information

[2020-07-13] CVSS2.0 Statistics



[2020-07-13] CVSS 3.0 Statistics



ICS ATT&CK Matrix-Stuxnet

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	**Damage to Property**
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial Comm Port	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	**Manipulation of Control**
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	**Manipulation of View**
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

11 Tactics
81 Techniques

ICS ATT&CK Matrix-Industroyer, CRASHOVERRIDE

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

11 Tactics
81 Techniques

ICS ATT&CK Matrix-Triton, Trisis, Hatman

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial Comm Port	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

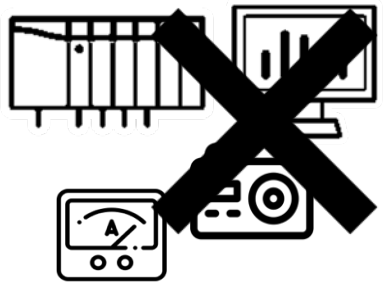
11 Tactics
81 Techniques

ICS/SCADA Security Threat Situation

Vulnerabilities are mostly critical and high risk levels

The number of vulnerability is rising year by year

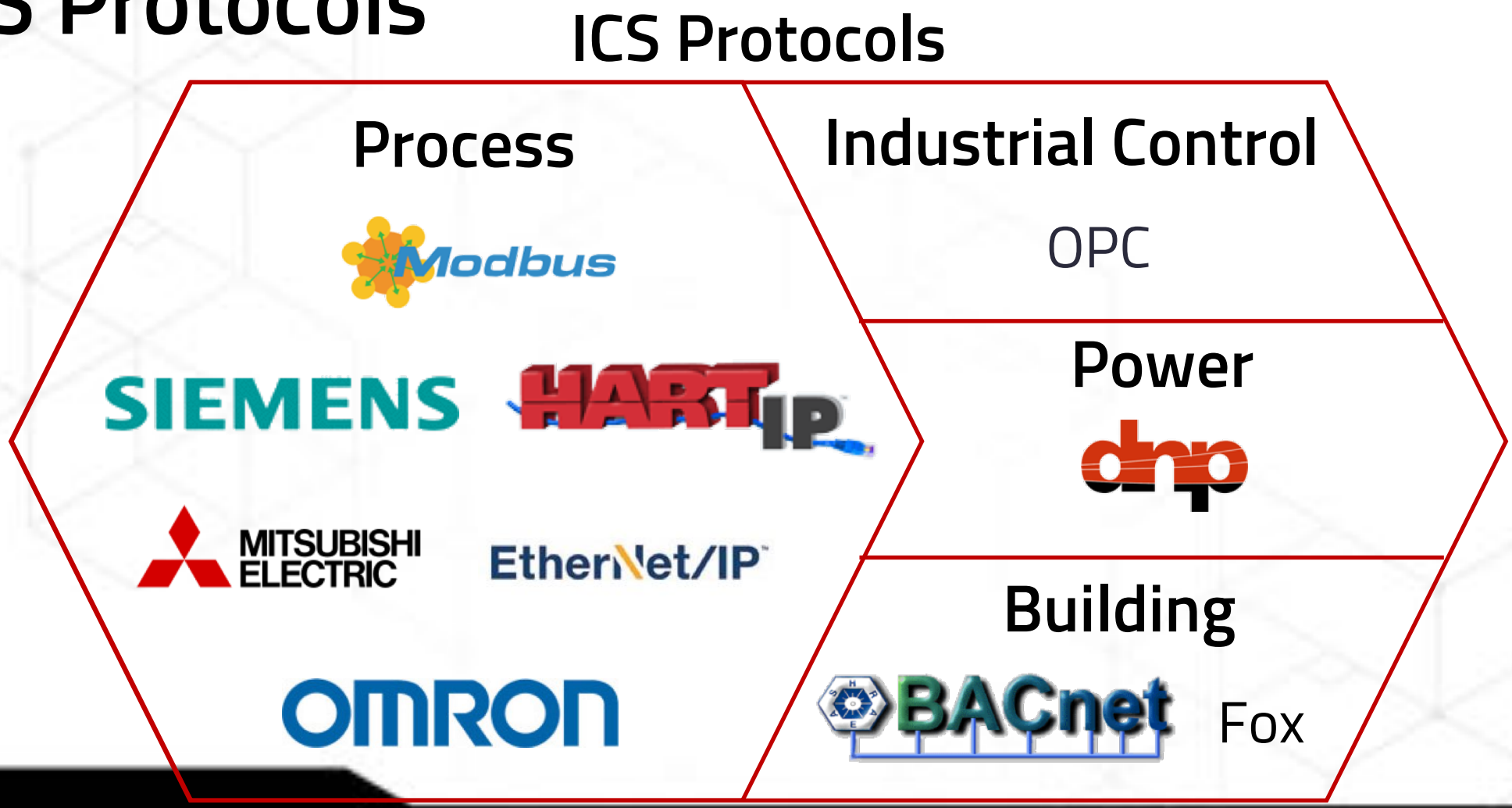
The security incidents have a huge impact



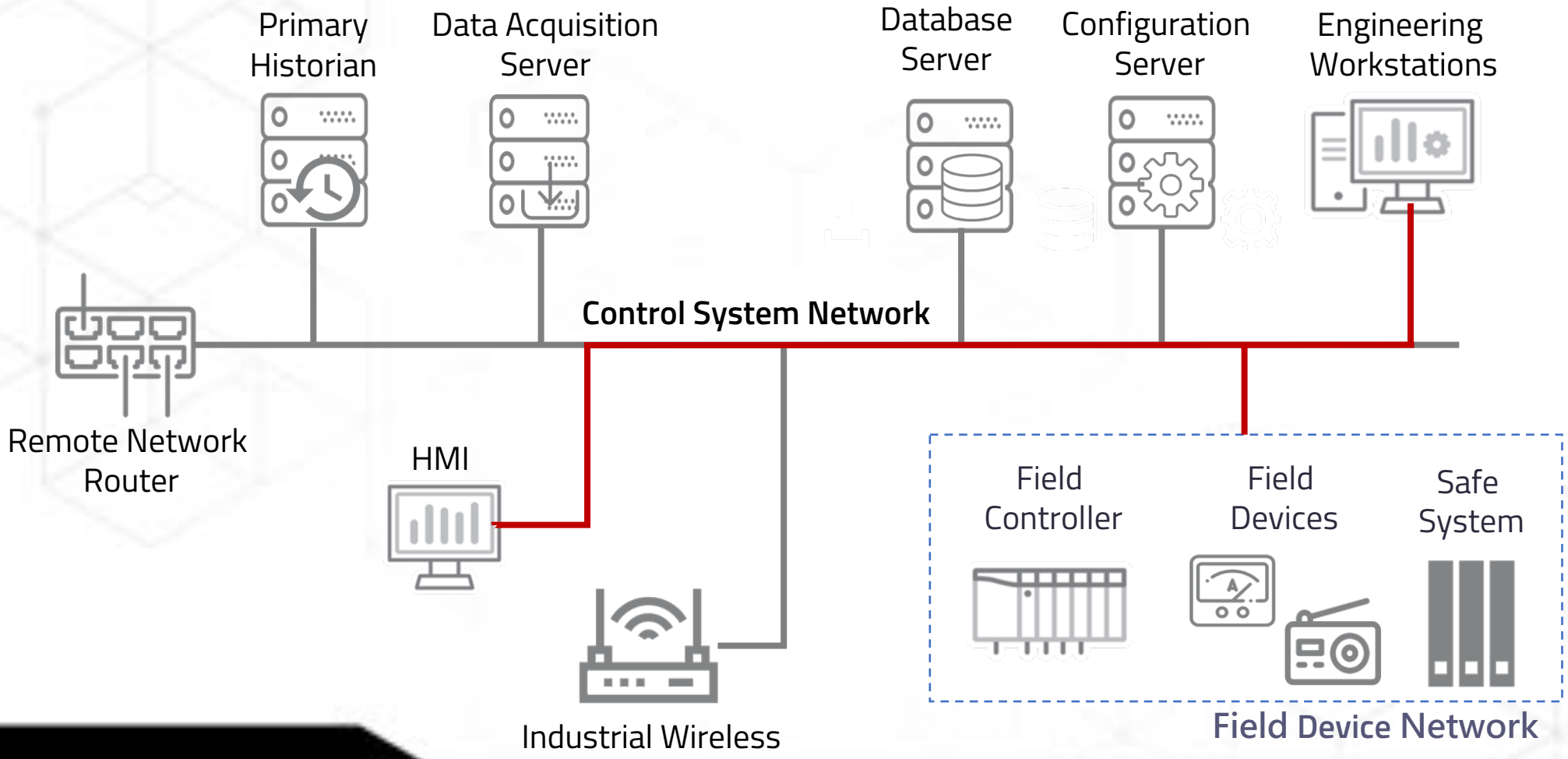
ICS/SCADA are not secure at all

Critical

ICS Protocols



The Communication of ICS Protocols



Critical Infrastructure Sectors (Taiwan)

High-Tech Park

Energy

Traffic

Communications

Government

Water

Finance

Medical

ICS Protocols and Critical Infrastructure Sectors (Singapore)


Aviation


Maritime


Water


Transport


Healthcare


Energy


Government

Security and Emergency Services


Banking and Finance


Infocomm


Media

ICS Protocols and Critical Infrastructure Sectors (Japan)

Aviation

Financial

Airport

Gas

Water

Information and communication

Medical

Electric power supply

Railway

Chemical

Credit card

Government and administrative

Petroleum

Logistics

ICS Protocols and Critical Infrastructure Sectors (US)

Chemical

Commercial Facilities

Communications

Critical Manufacturing

Emergency Services

Energy

Financial Services

Healthcare and Public Health

Transportation Systems

Food and Agriculture

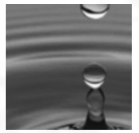
Defense Industrial Base

Government Facilities

Nuclear Reactors, Materials, and Waste

Dams

Information Technology



Water and Wastewater Systems

ICS Protocols and Critical Infrastructure Sectors (US)

Chemical

Commercial Facilities

Communications

Critical Manufacturing

Emergency Services

Energy

Financial Services

Healthcare and Public Health

Transportation Systems

Food and Agriculture

Defense Industrial Base

Government Facilities

Nuclear Reactors, Materials, and Waste

Dams

Information Technology

Water and Wastewater Systems

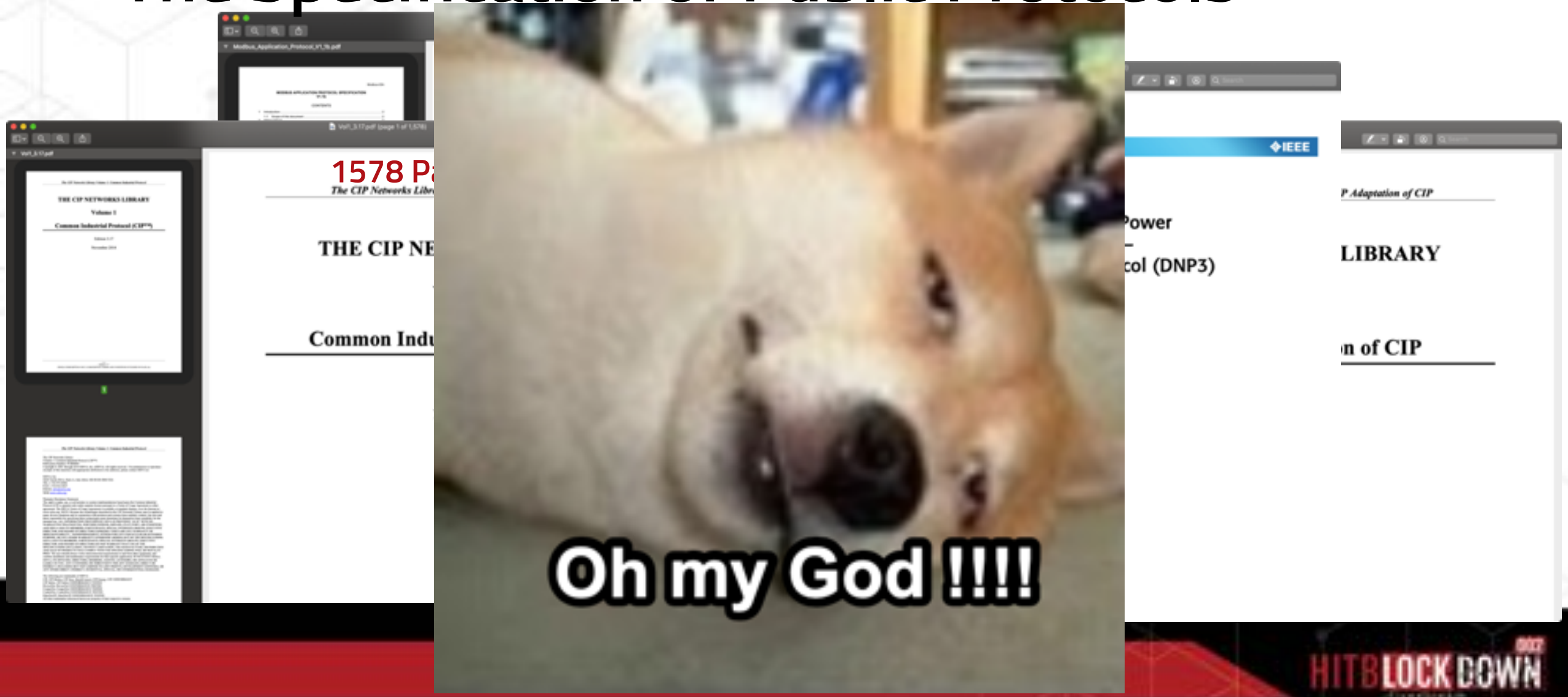
Public and Private: ICS Network Protocols

Why Public vs. Private Protocols?

Public

Private

The Specification of Public Protocols



Modbus/TCP Handshake Process

HMI

PLC

$M_1.Request$

$$Request \begin{cases} Modbus = \langle Function\ Code | Request\ Data \rangle \\ Modbus/TCP = \langle Transaction\ ID \mid Protocol\ ID \mid Length \mid Unit\ ID \rangle \\ TCP \\ IP \\ Ethernet \end{cases}$$

Modbus/TCP				Modbus	
Transaction Identifier	Protocol Identifier	Length Field	Unit ID	Function Code	Request Data

$M_2.Response$

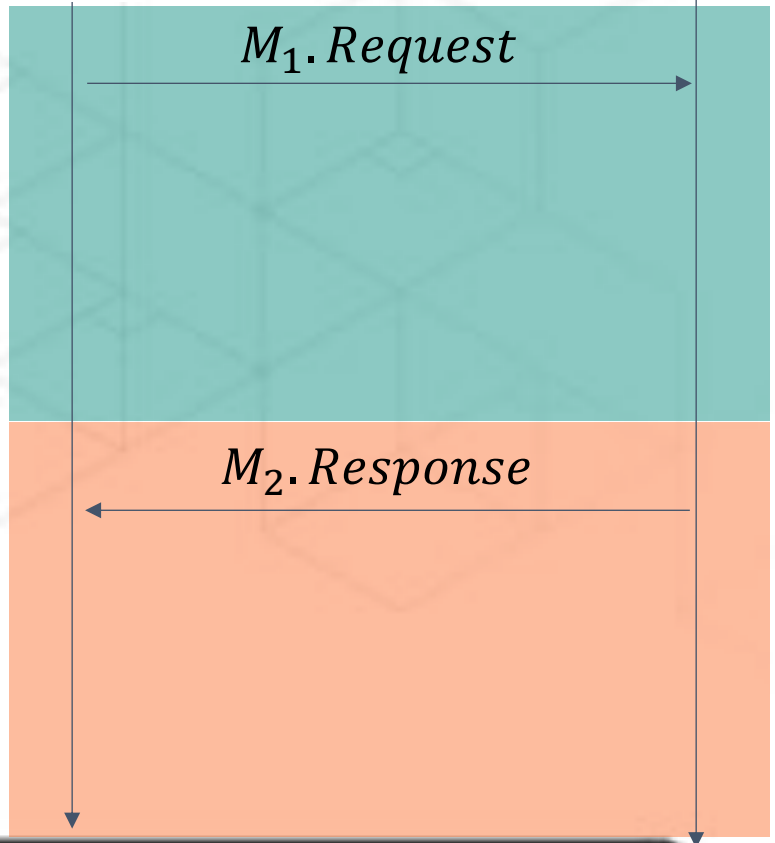
$$Response \begin{cases} Modbus = \langle Function\ Code | Response\ Data \rangle \\ Modbus/TCP = \langle Transaction\ ID \mid Protocol\ ID \mid Length \mid Unit\ ID \rangle \\ TCP \\ IP \\ Ethernet \end{cases}$$

Modbus/TCP				Modbus	
Transaction Identifier	Protocol Identifier	Length Field	Unit ID	Function Code	Response Data

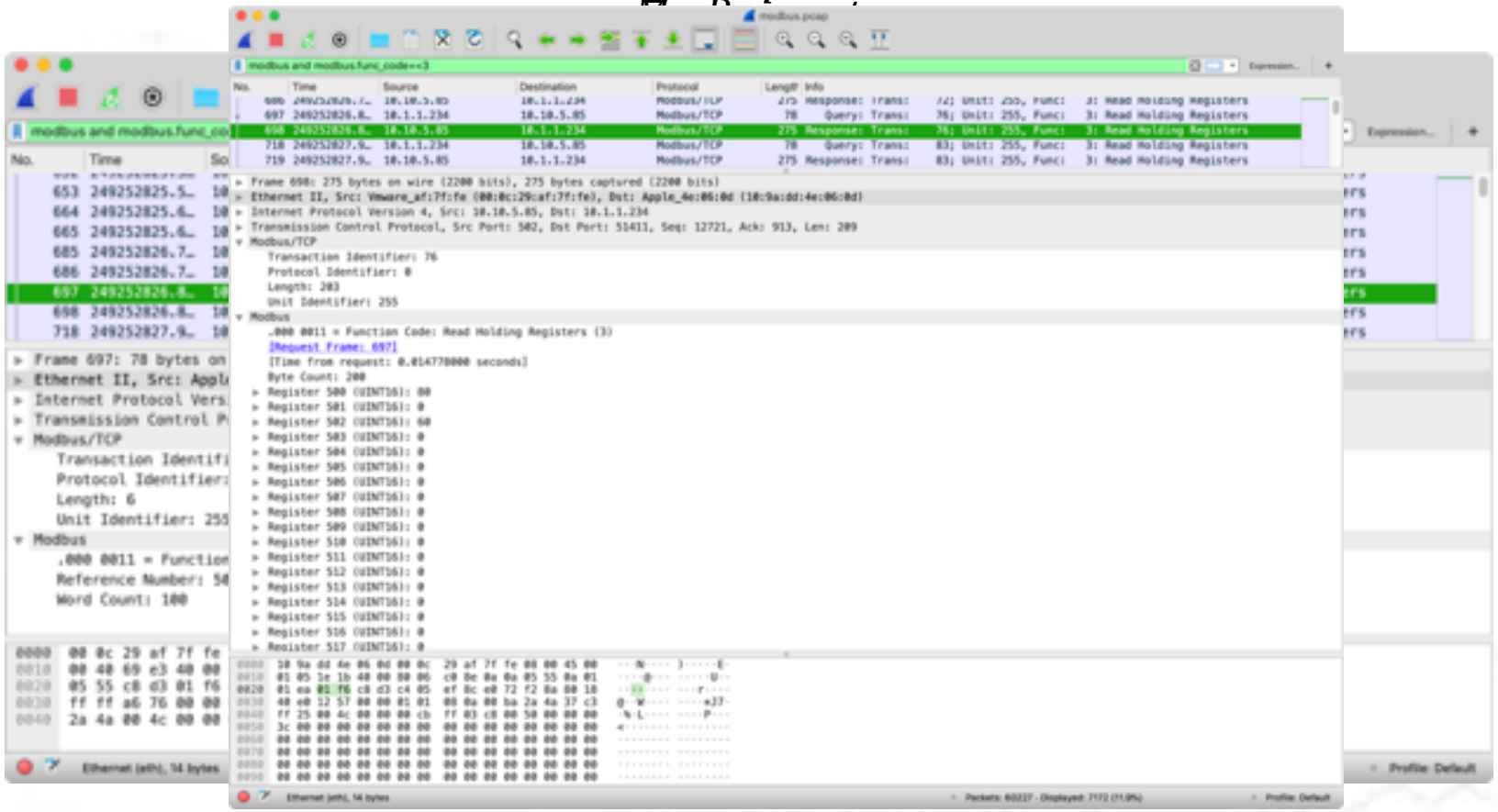
Modbus/TCP Handshake Process

HMI

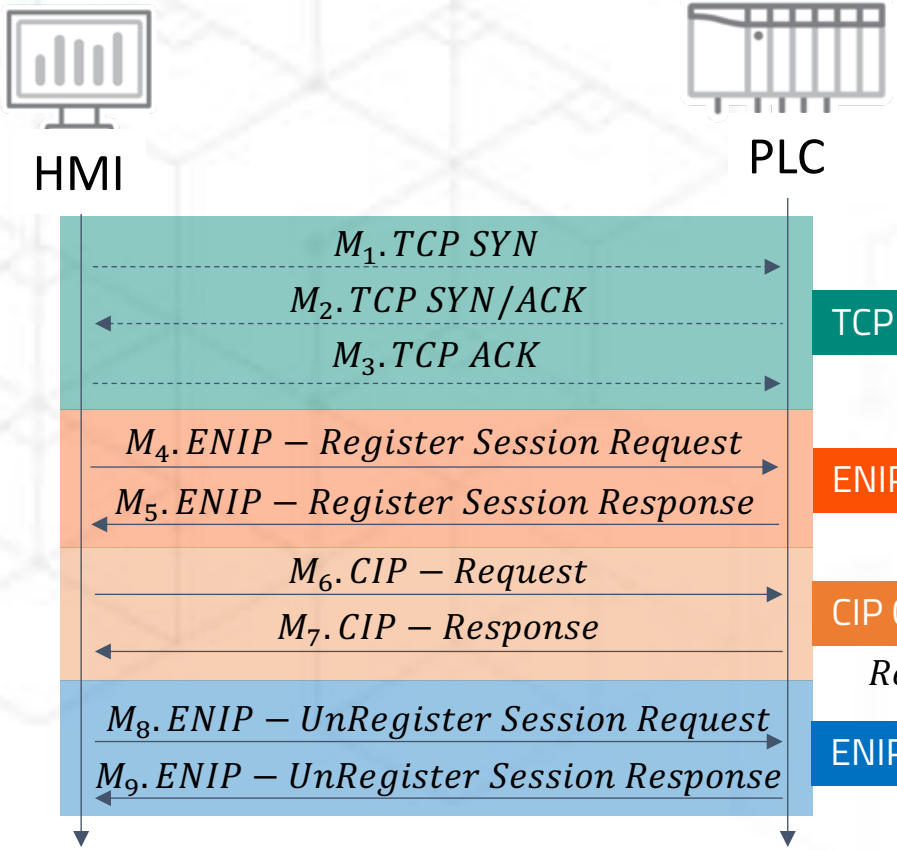
PLC



$M_2.Response$



EtherNet/IP CIP Handshake Process



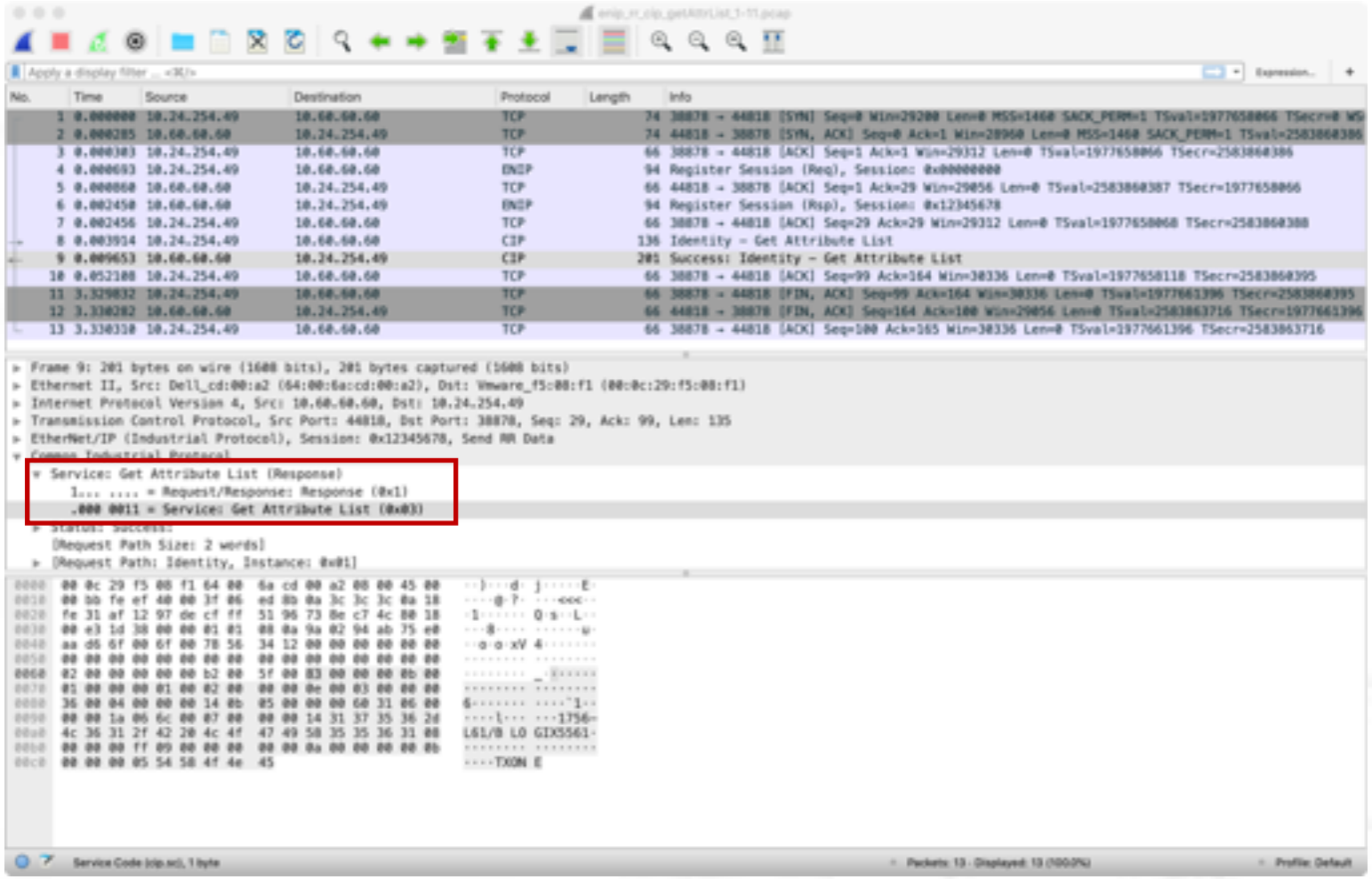
$$Request/Response \begin{cases} EtherNet/IP \begin{cases} Encapsulation\ Header \\ Command\ Specific\ Data \end{cases} \\ TCP/UDP \\ IP \\ Ethernet \end{cases}$$

$$Request/Response \begin{cases} CIP \begin{cases} CIP\ Motion | Motion\ Control | Semiconductor\ Profiles \\ Object\ Library\ (Communication, Applications, Time\ Synchronization) \\ Data\ Management\ Services\ Explicit\ and\ I/O\ Messages \\ Connection\ Management, Routing \end{cases} \\ EtherNet/IP \begin{cases} Encapsulation\ Header \\ Command\ Specific\ Data \end{cases} \\ TCP/UDP \\ IP \\ Ethernet \end{cases}$$

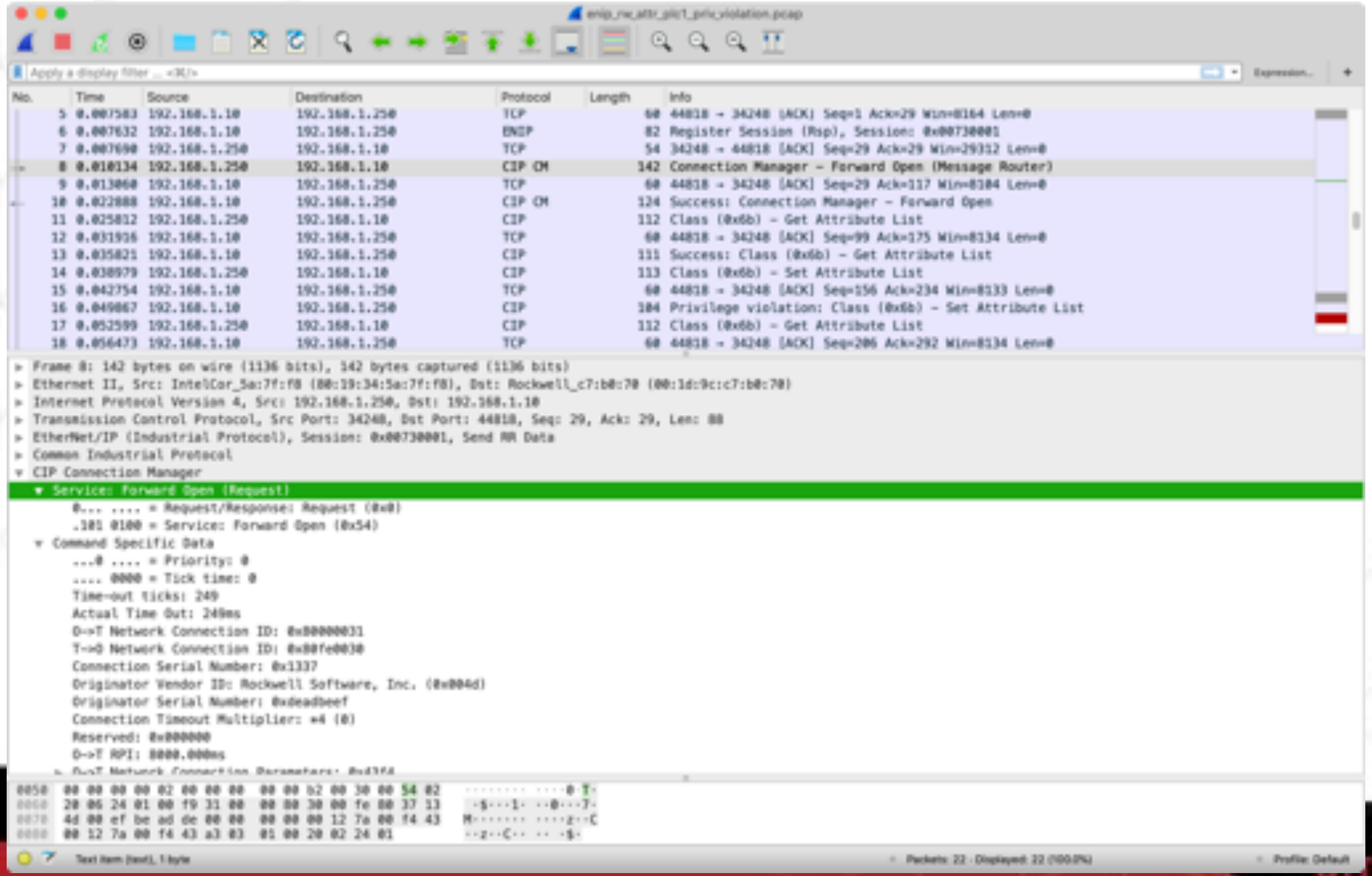
Command	Length	Session Handle	Status	Max Delay	Sender Context	Options	Command-specific Data
2 bytes	2 bytes	4 bytes	4 bytes	2 bytes	6 bytes	4 bytes	6 bytes

Function Code

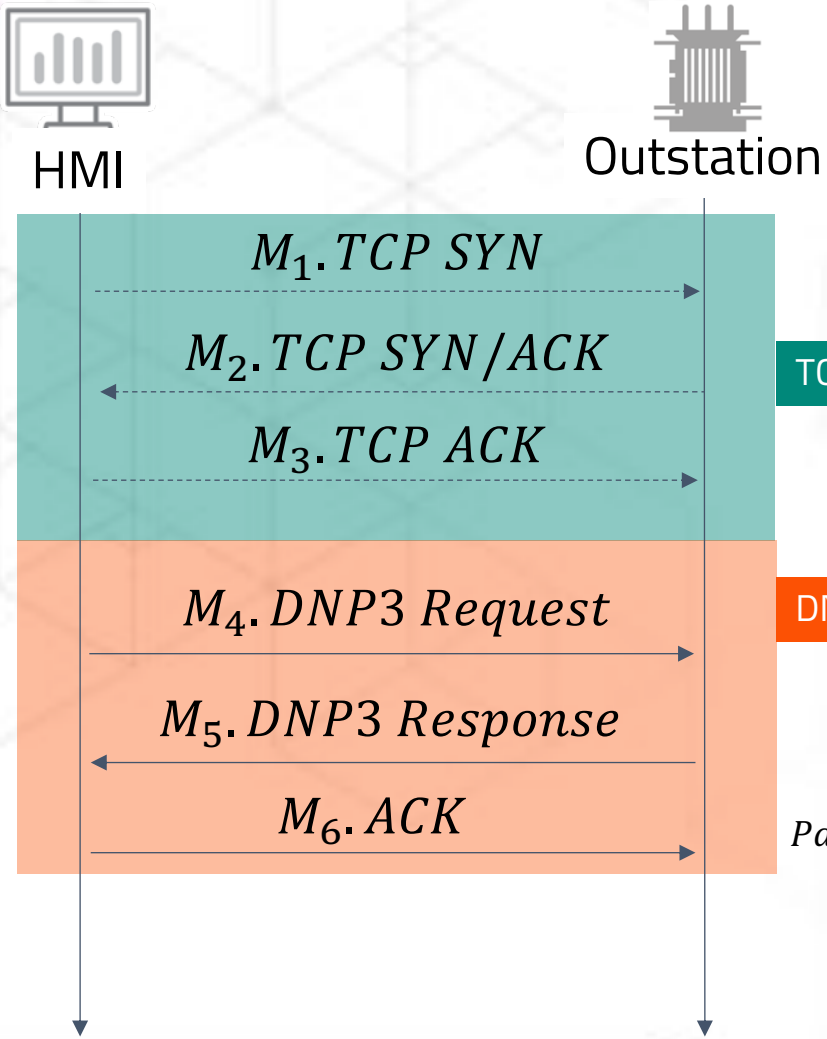
- Get Attributes All 0x01
- Set Attributes All 0x02
- Get Attribute List 0x03
- Set Attribute List 0x04
- Start 0x06
- Stop 0x07



EtherNet/IP Traffic



DNP3 Handshake Process



$$Packet \begin{cases} DNP3 \begin{cases} DNP3\ Application\ Layer = \{Control|Function|Internal\ Indicators|Data\ Objects\} \\ Application\ Chunks = \{Length|\boldsymbol{CRC}\} \\ Pseudo\ Transport\ Layer = \{Transport\ Control\} \\ Data\ Link\ Layer = \{Start\ Byte|Length|Control|Destination|Source|\boldsymbol{CRC}\} \end{cases} \\ TCP \\ IP \\ Ethernet \end{cases}$$

DNP3 Handshake Process

HMI

Outstation

$M_1.TCP\ SYN$

$M_2.TCP\ SYN/ACK$

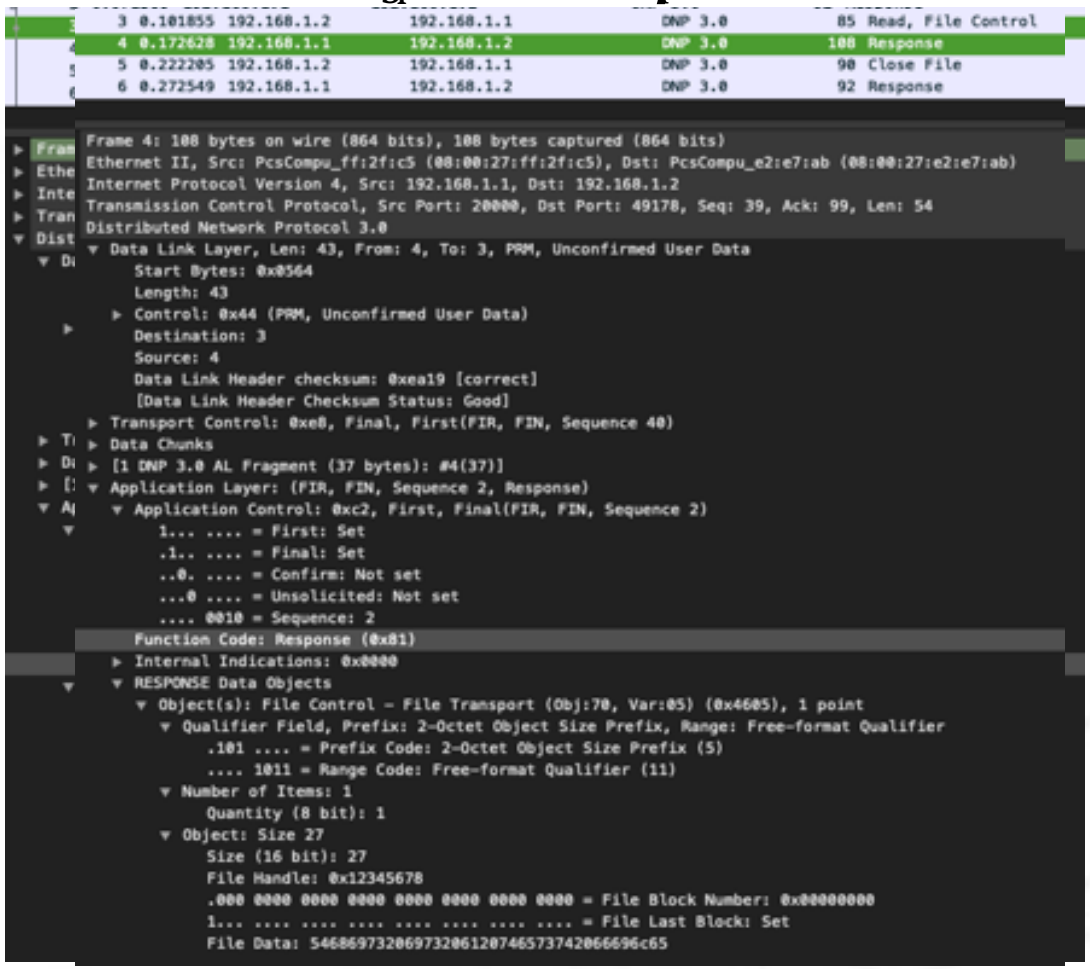
$M_3.TCP\ ACK$

$M_4.DNP3\ Request$

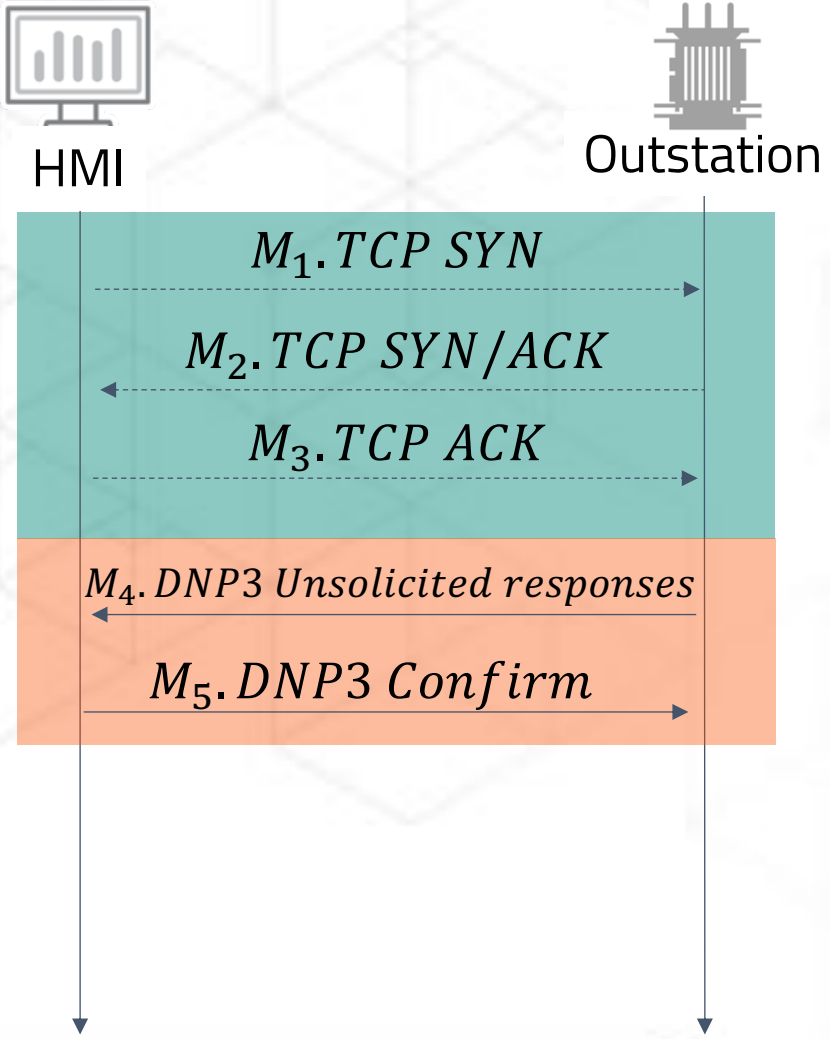
$M_5.DNP3\ Response$

$M_6.ACK$

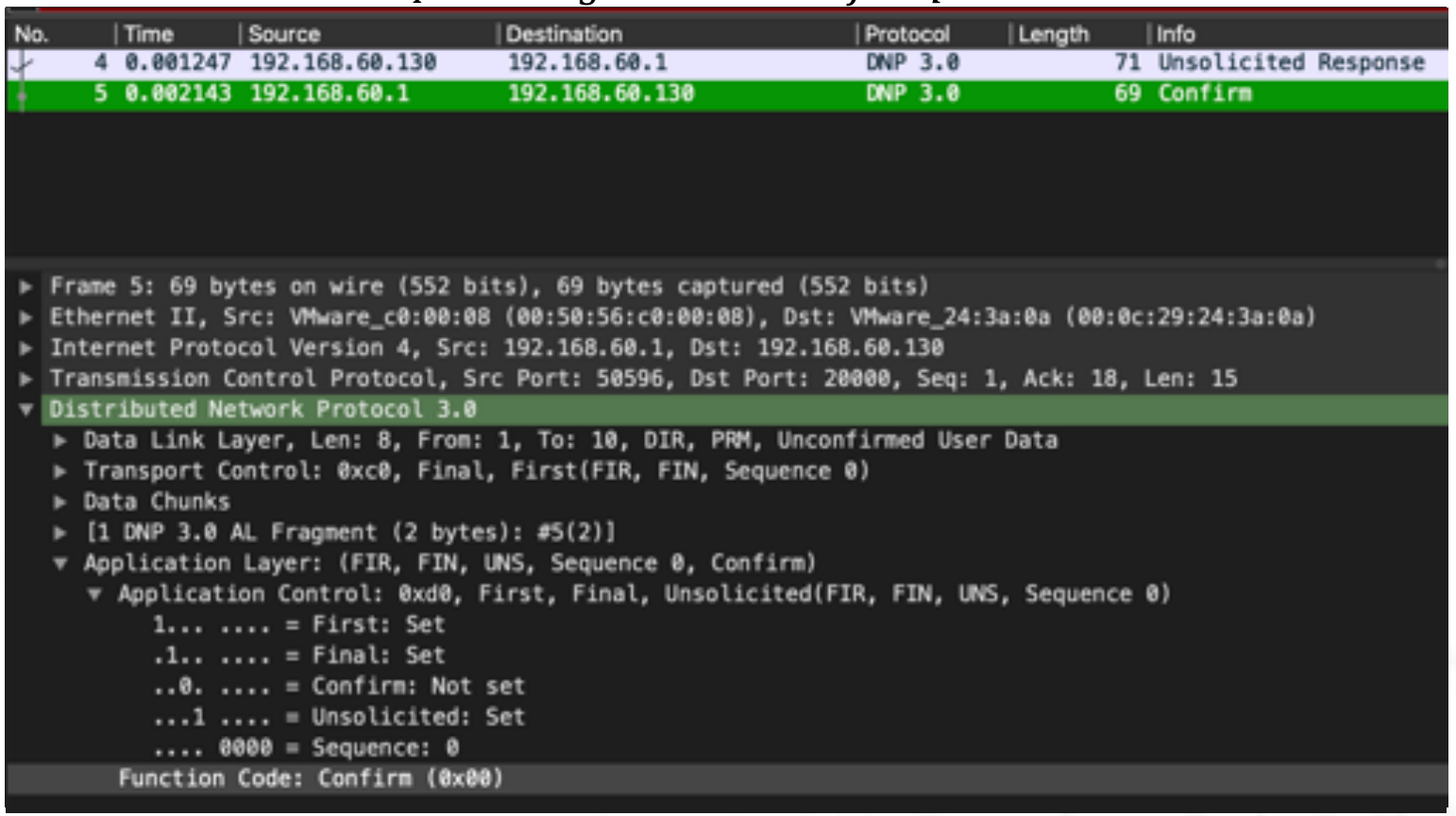
$M_5.DNP3\ Response$



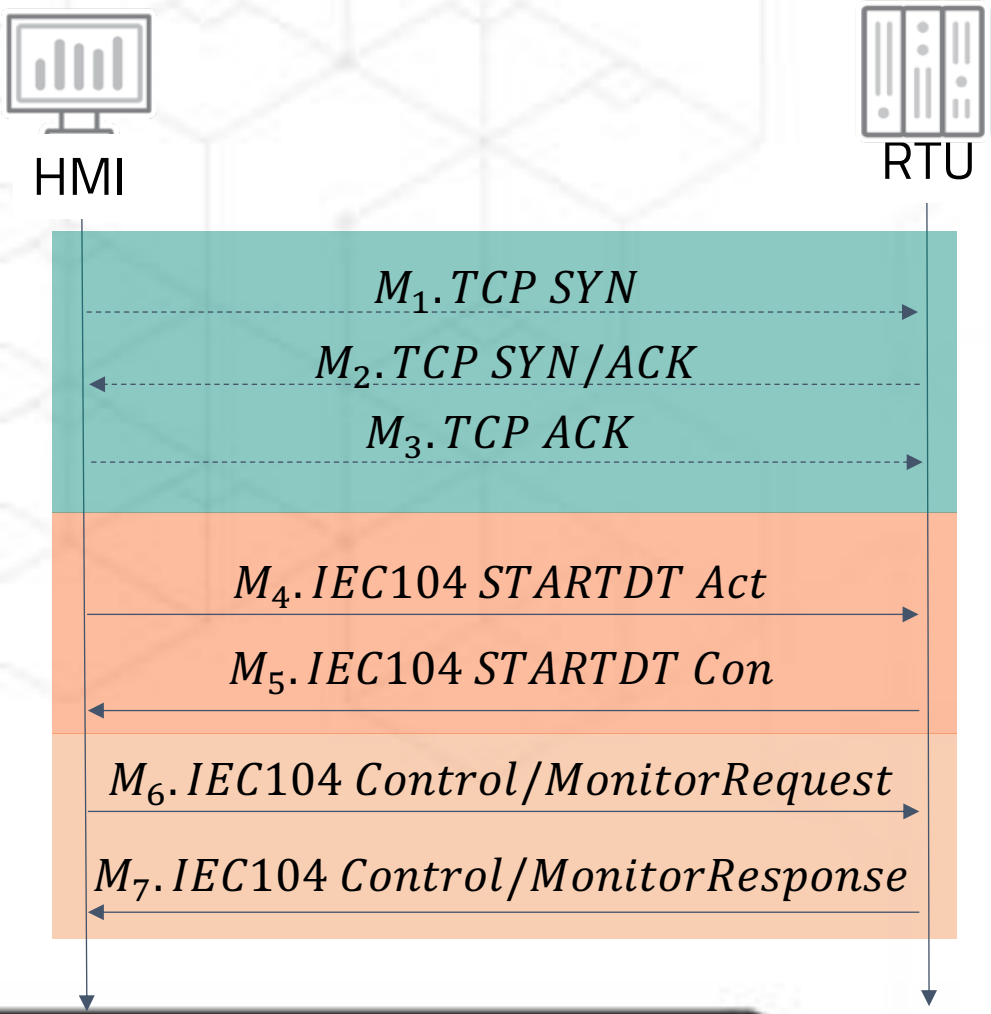
DNP3 Handshake Process (Unsolicited responses)



$M_4.DNP3\ Unsolicited\ responses$ $M_5.DNP3\ Confirm$



IEC 104 Handshake Process



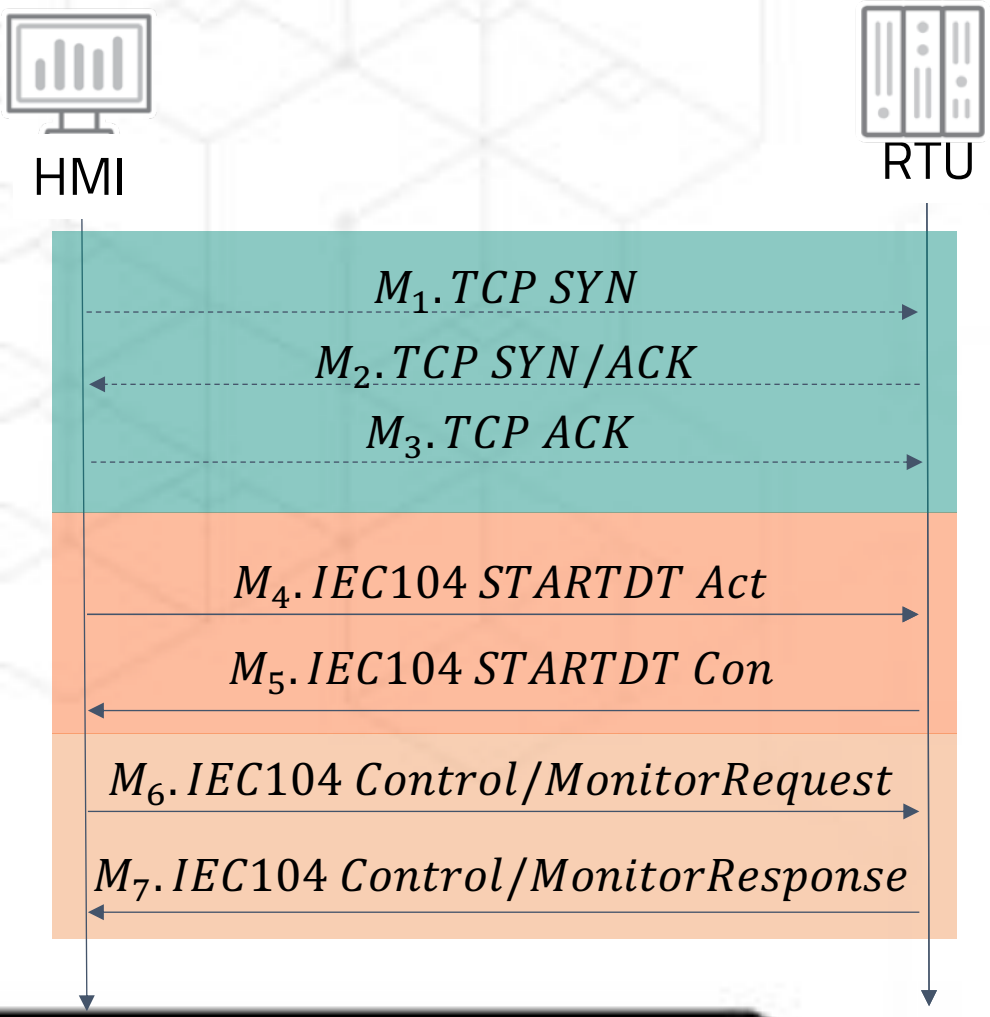
$Packet$ { $Application\ Protocol\ Control\ Information\ (APCI)$ { $Start\ Byte (0x68)$, $Length\ of\ APDU$, $Type = \{I, S, U\}$ }, TCP, IP, $Ethernet$ }

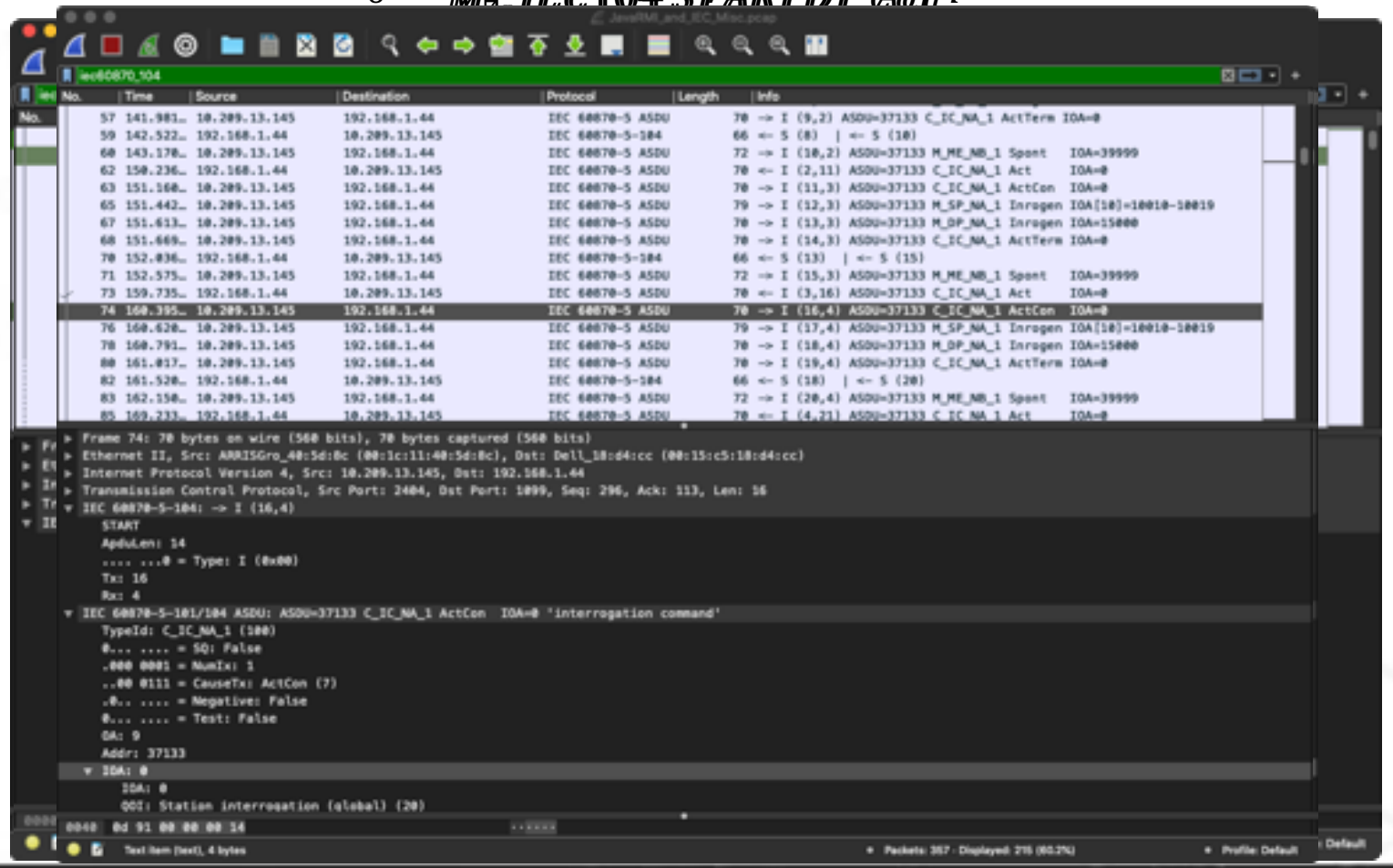
8 bits		
Type Identification		0
S Q	Number of objects	
T	P/N	Cause of transmission (COT)
Originator address (ORG)		
ASDU address fields (2 bytes)		
Information object address (IOA) fields (3 bytes)		
Information Elements		
Time Tag		
Information Object 2		
Information Object N		

$Packet$ { $Application\ Service\ Data\ Units (ASDU)$, $Application\ Protocol\ Control\ Information (APCI)$, TCP, IP, $Ethernet$ }

$$Application\ Protocol\ Data\ Unit = APCI + ASDU$$

IEC 104 Handshake Process

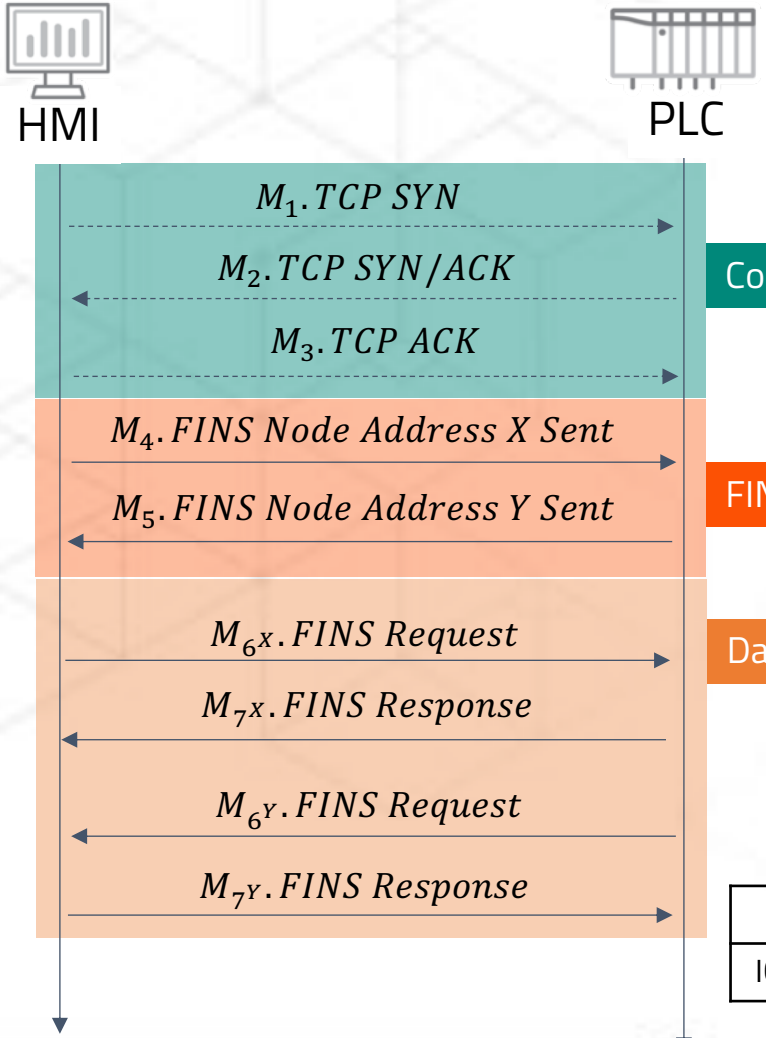




Function List

- Process information in monitor direction (21)
  - 0x0a (10) M_ME_TA_1:Measured value, normalized value with time tag
- Process telegrams with long time tag IO (11)
- Process information in control direction (7)
  - 0x2d (45) C_SC_NA_1 : Single command
- Command telegrams with long time tag (7)
- System information in monitor direction (1)
- System information in control direction (8)
  - 0x65 (101) C_CI_NA_1 : Counter interrogation command
- Parameter in control direction (4)
- File transfer (8)
  - 0x78 (120) F_FR_NA_1 : File ready

ORMON FINS Handshake Process



$$Request \begin{cases} FINS\ Frame = \langle Request\ Command | Data \rangle \\ FINS\ Header = \langle ICF|RSV|GCT|DNA|DA1|DA2|SNA|SA1|SA2|SID \rangle \\ FINS/TCP\ Header = \langle Magic|\ Length|\ Command|\ Error\ Code \rangle \\ TCP/UDP \\ IP \\ Ethernet \end{cases}$$

FINS Header										FINS Frame	
ICF	RSV	GCT	DNA	DA1	DA2	SNA	SA1	SA2	SID	Request Command	Data

$$Response \begin{cases} FINS\ Frame = \langle Request\ Command | Response\ Code | Data \rangle \\ FINS\ Header = \langle ICF|RSV|GCT|DNA|DA1|DA2|SNA|SA1|SA2|SID \rangle \\ FINS/TCP\ Header = \langle Magic|\ Length|\ Command|\ Error\ Code \rangle \\ TCP/UDP \\ IP \\ Ethernet \end{cases}$$

FINS Header										FINS Frame		
ICF	RSV	GCT	DNA	DA1	DA2	SNA	SA1	SA2	SID	Request Command	Response Code	Data

ORMON FINS Handshake Process

$M_1.TCP\ SYN$

$M_2.TCP\ SYN/ACK$

$M_3.TCP\ ACK$

$M_4.FINS\ Node\ Address\ X\ Sent$

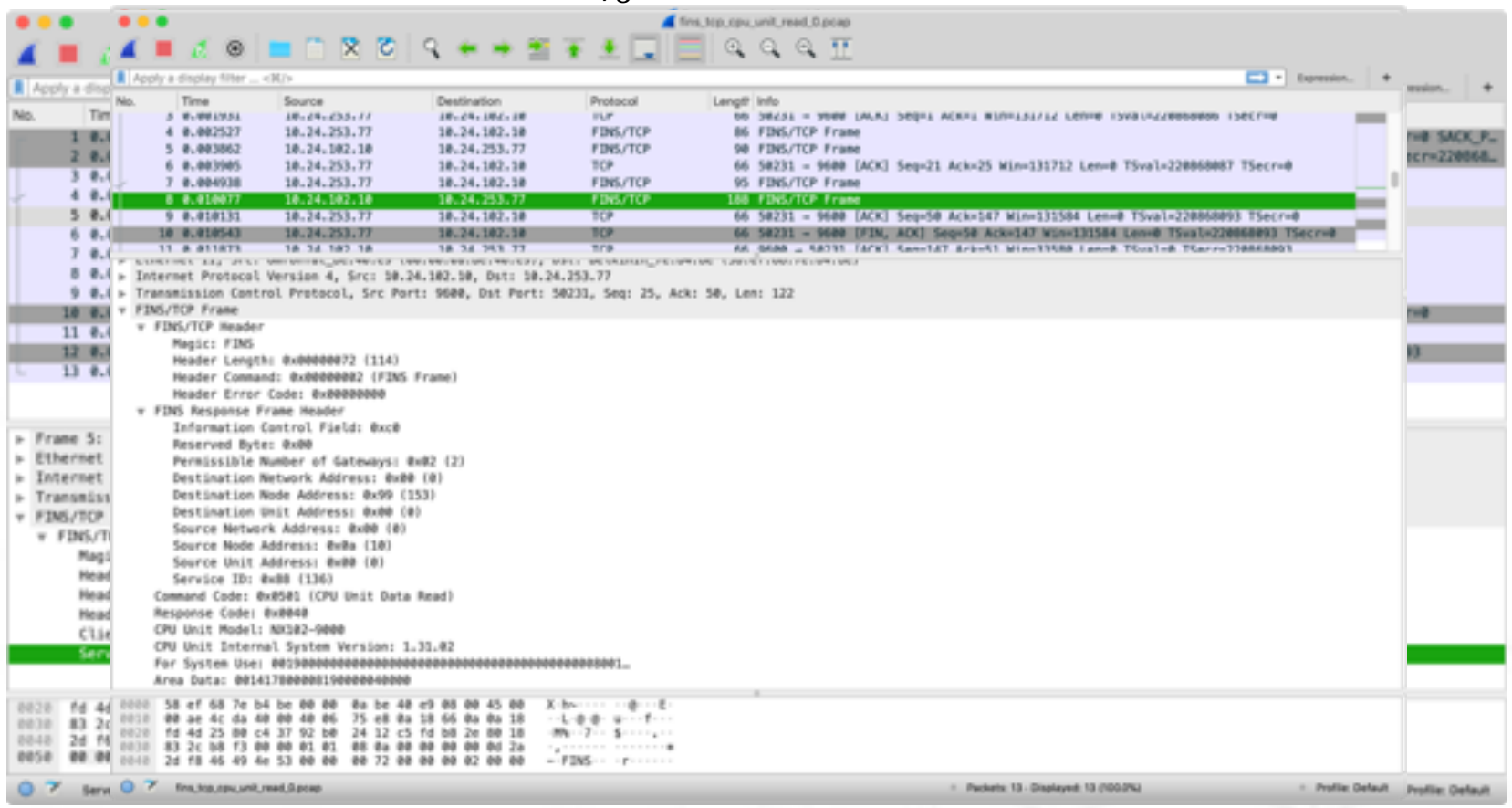
$M_5.FINS\ Node\ Address\ Y\ Sent$

$M_6x.FINS\ Request$

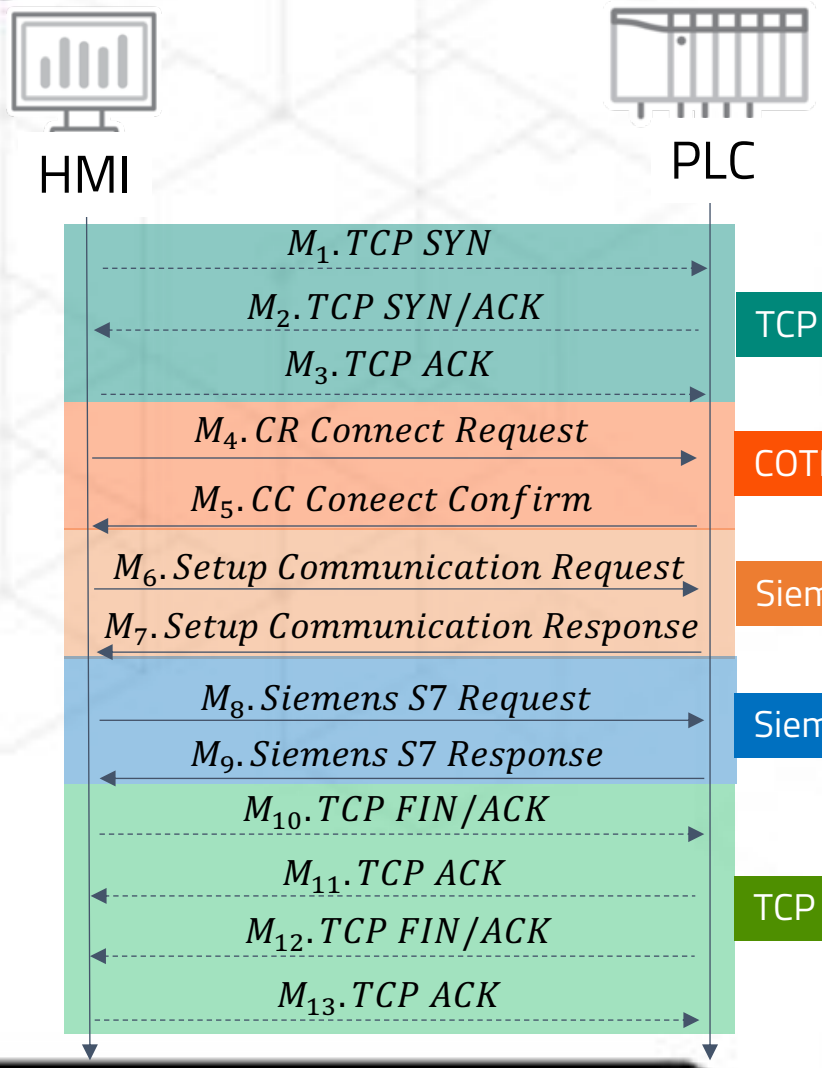
$M_7x.FINS\ Response$

$M_6Y.FINS\ Request$

$M_7Y.FINS\ Response$



Siemens S7 Handshake Process



TCP 3-Way Connection Set-up

COTP Connection Establishment

$Packet$ {
$ISO\ 8073/X.224\ COTP\ Connection-Oriented\ Transport\ Protocol$
$ISO\ transport\ services\ on\ top\ of\ the\ TCP\ (TPKT)$
TCP
IP
$Ethernet$
}

Siemens S7 Communication Establishment

Siemens S7 Communication

$Packet$ {
$S7\ Commonication\ \{Header|Parameter|Data\}$
$ISO\ 8073/X.224\ COTP\ Connection-Oriented\ Transport\ Protocol$
$ISO\ transport\ services\ on\ top\ of\ the\ TCP\ (TPKT)$
TCP
IP
$Ethernet$
}

TCP 4-Way Connection Close

Siemens S7 Handshake Process

HMI

PLC

$M_1.TCP\ SYN$

$M_2.TCP\ SYN/ACK$

$M_3.TCP\ ACK$

$M_4.CR\ Connect\ Request$

$M_5.CC\ Coneect\ Confirm$

$M_6.Setup\ Communication\ Request$

$M_7.Setup\ Communication\ Response$

$M_8.Siemens\ S7\ Request$

$M_9.Siemens\ S7\ Response$

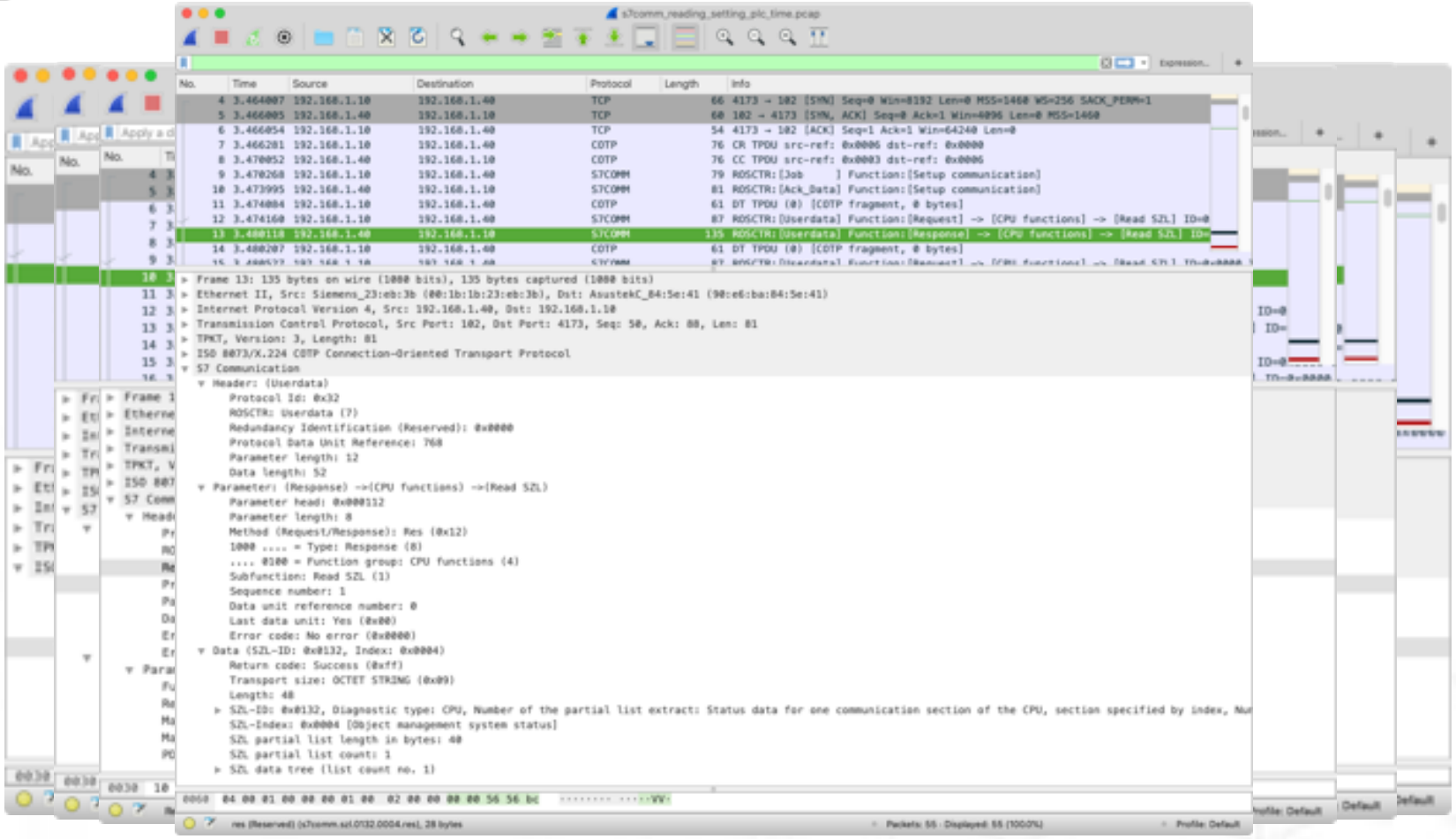
$M_{10}.TCP\ FIN/ACK$

$M_{11}.TCP\ ACK$

$M_{12}.TCP\ FIN/ACK$

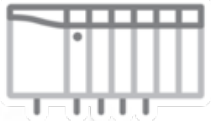
$M_{13}.TCP\ ACK$

$M_8.Siemens\ S7\ Request$

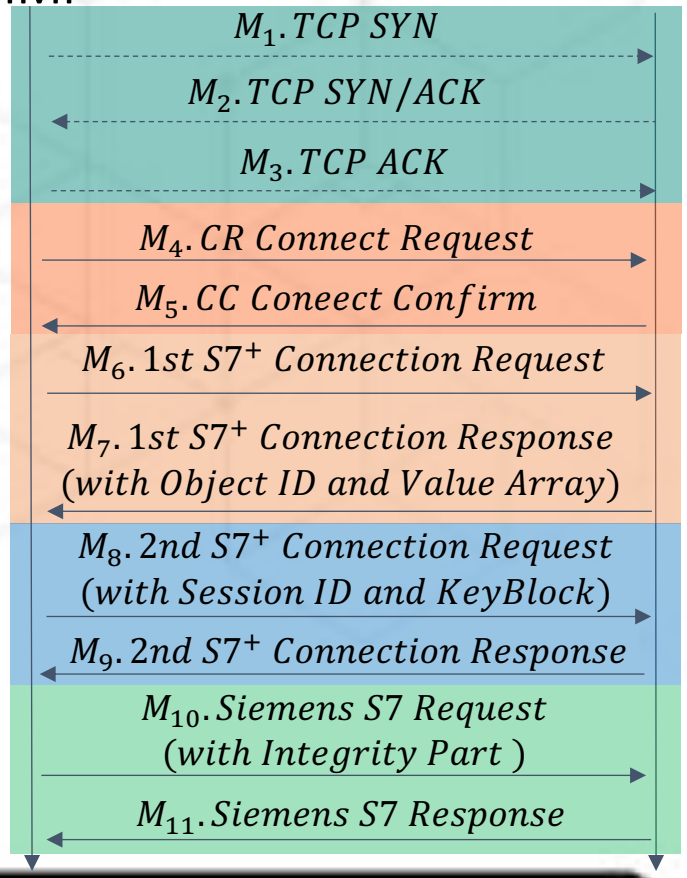


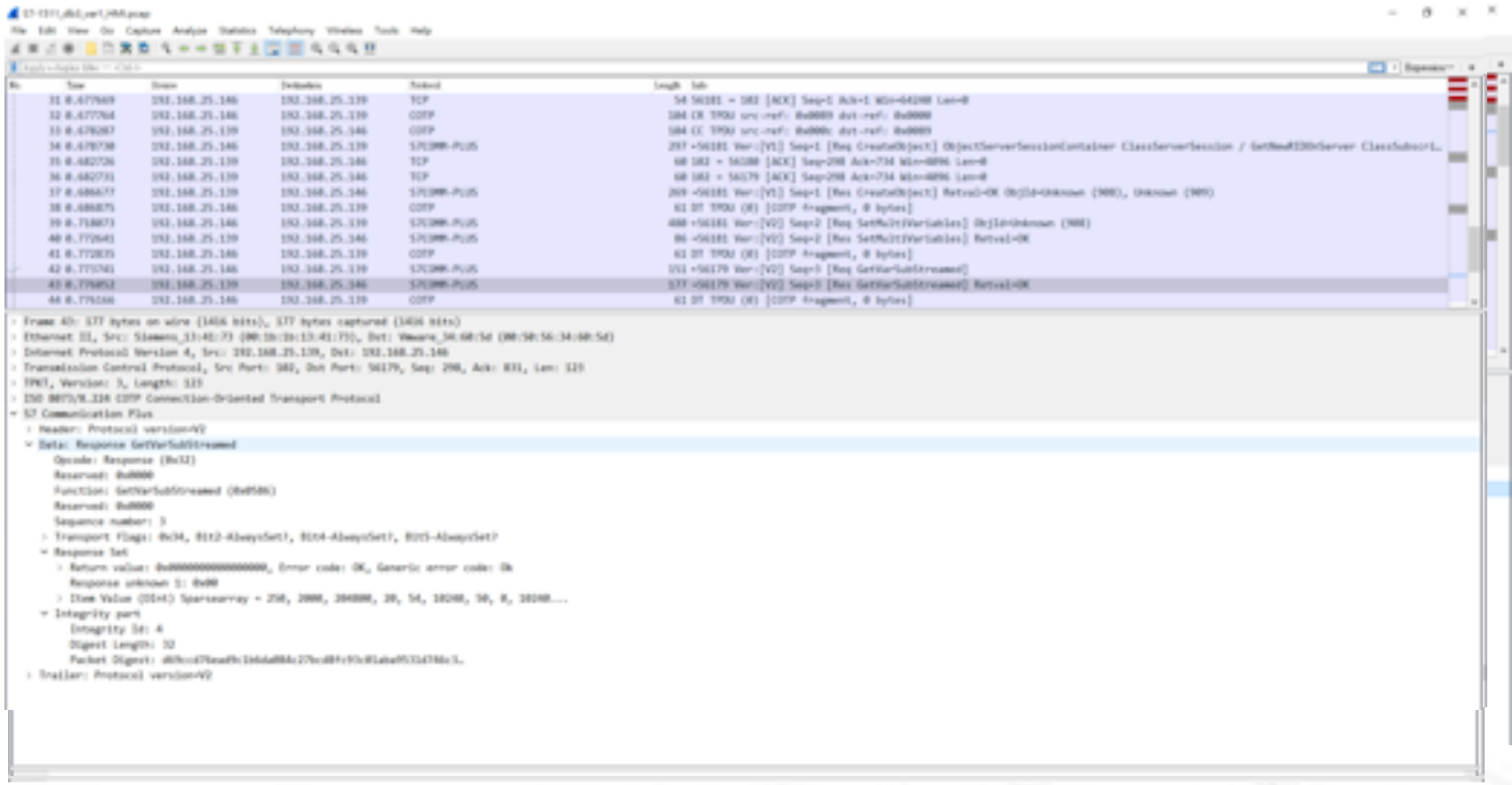
Siemens S7 Plus Handshake Process

HMI



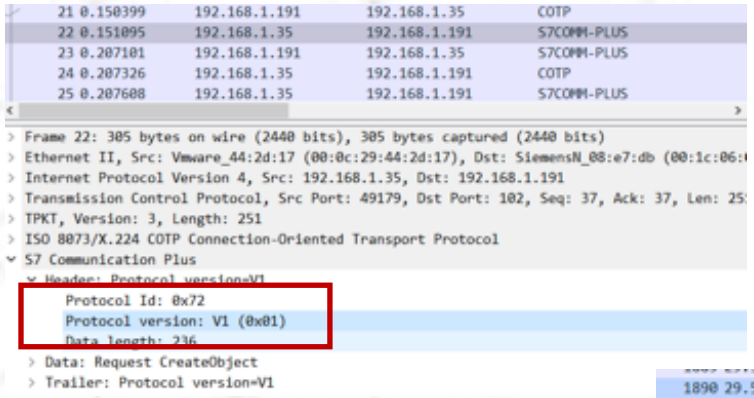
PLC



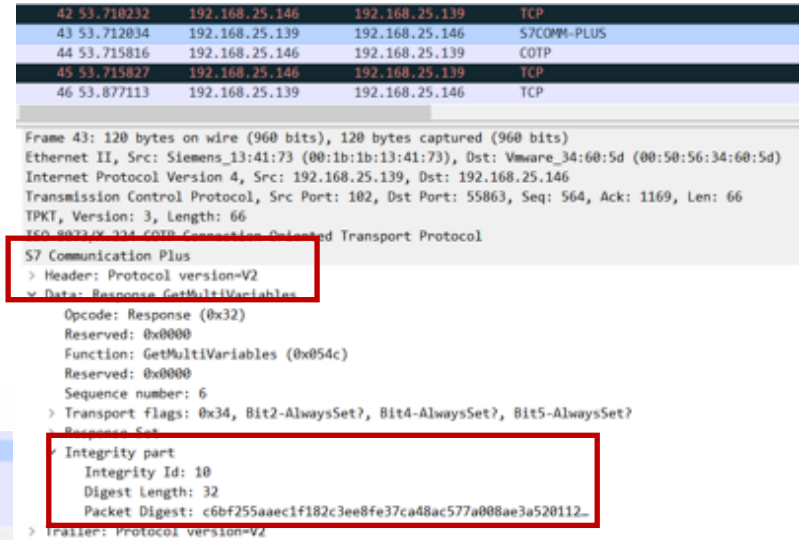


Siemens S7 Plus Version

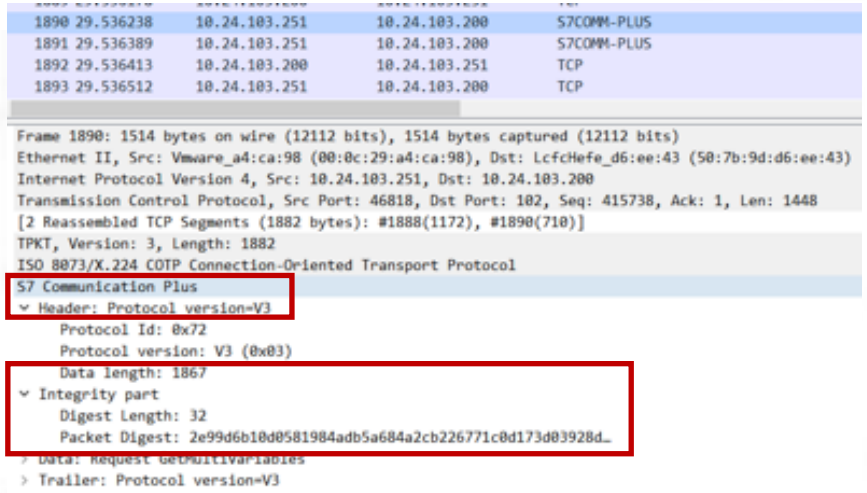
V1



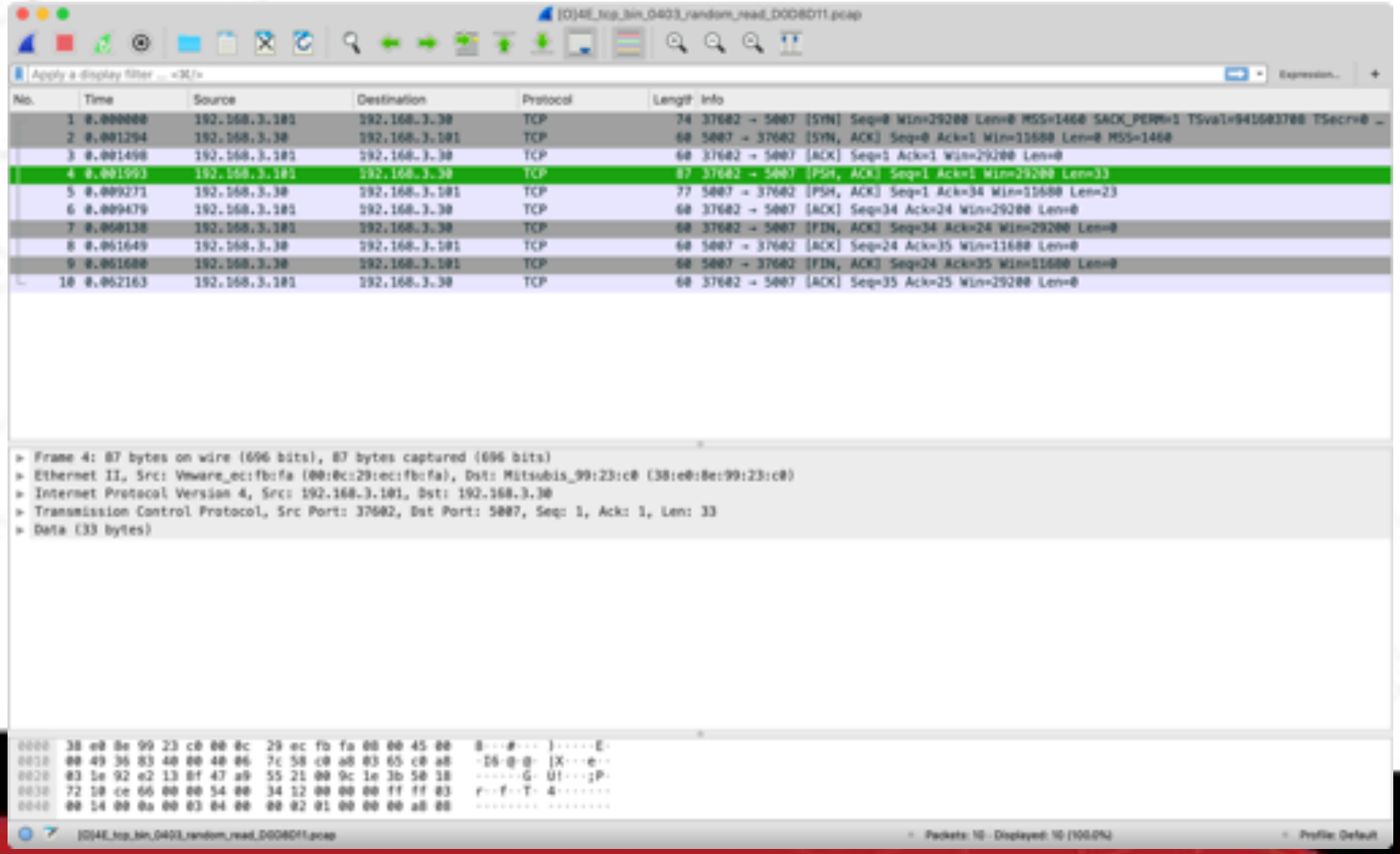
V2



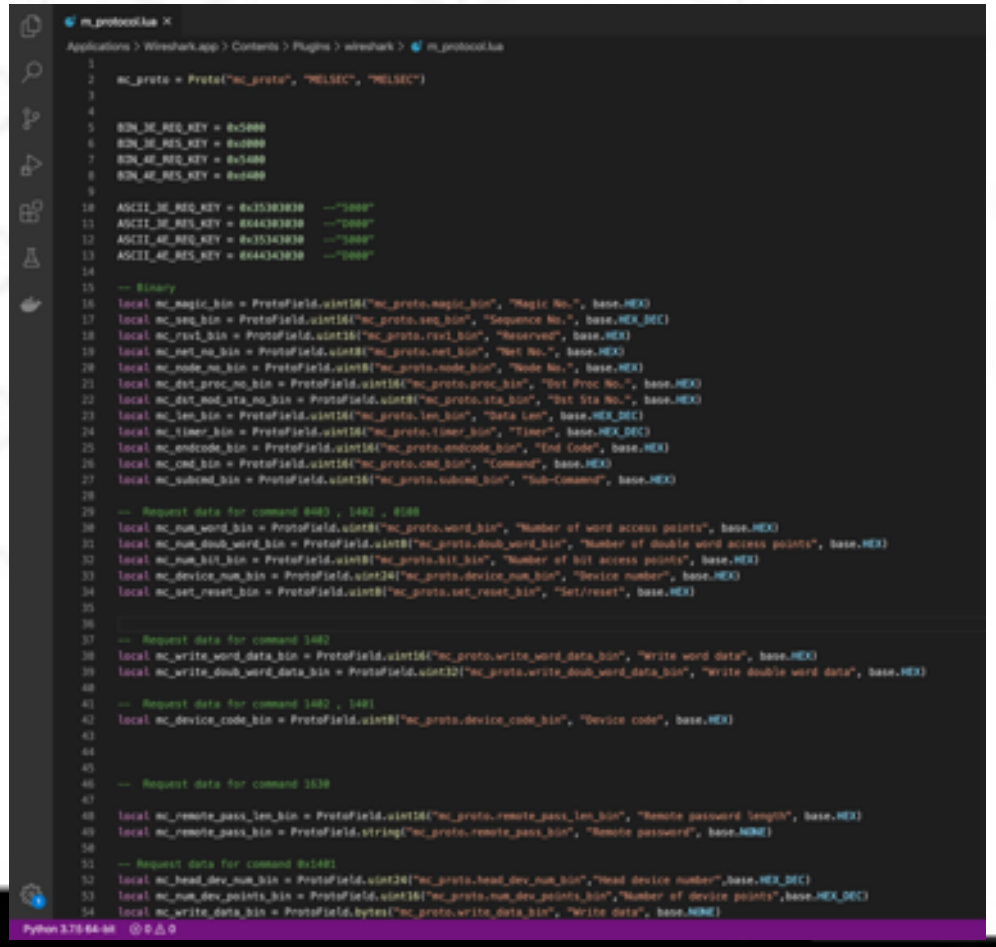
V3

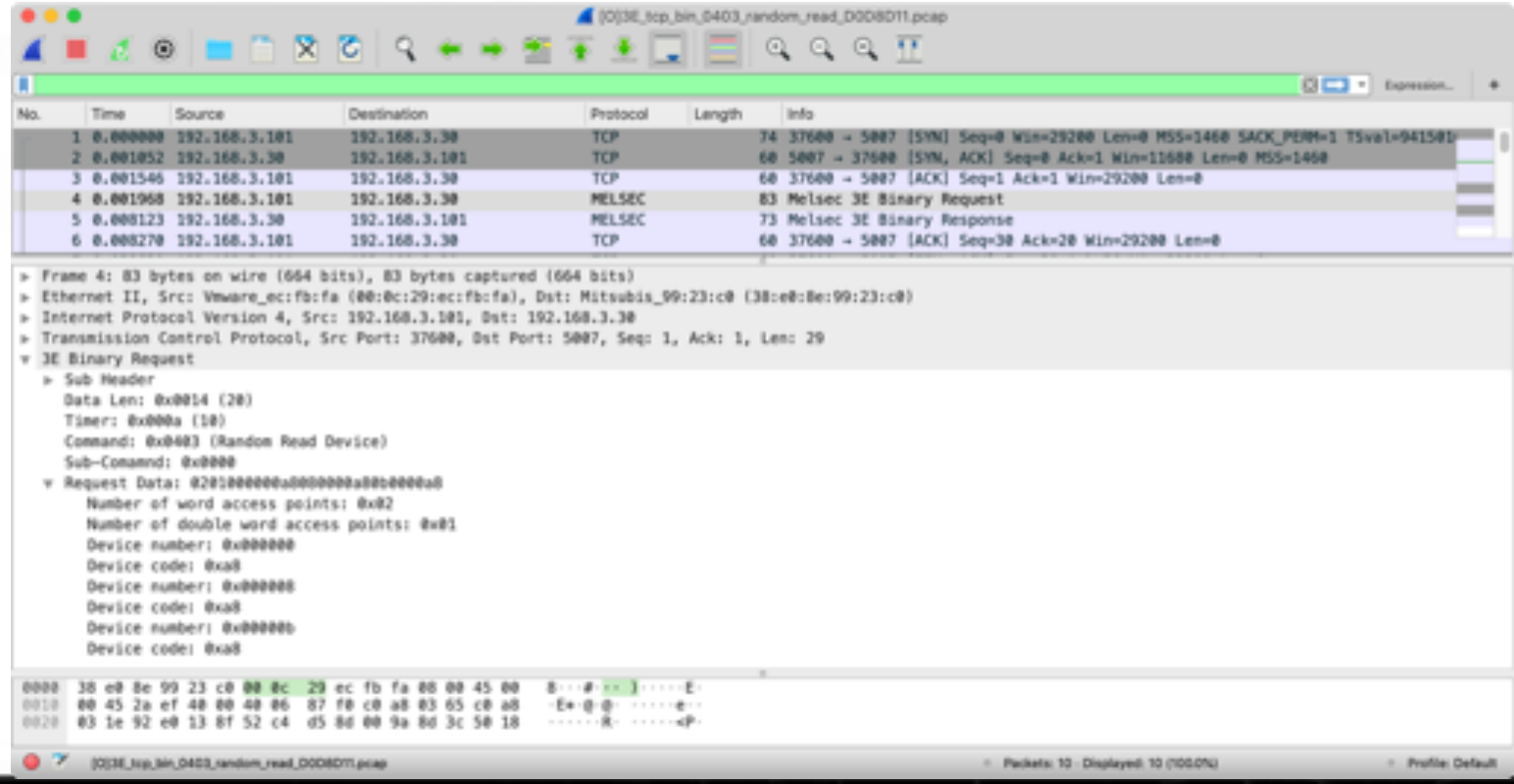


Wireshark can’t analyze MELSEC Traffic

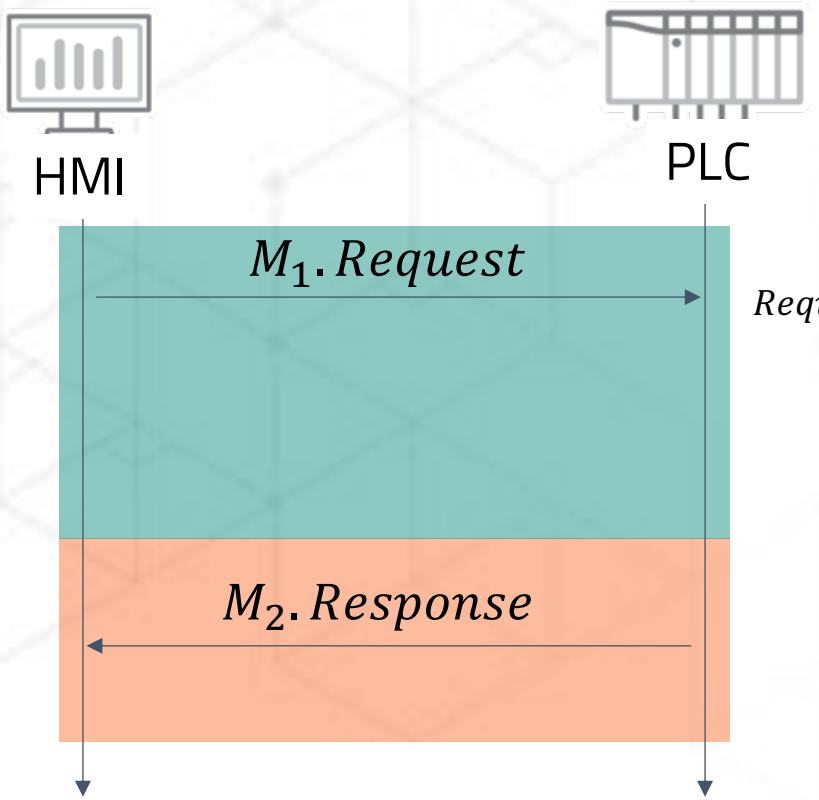


Again, we built a Lua Plugin for MELSEC





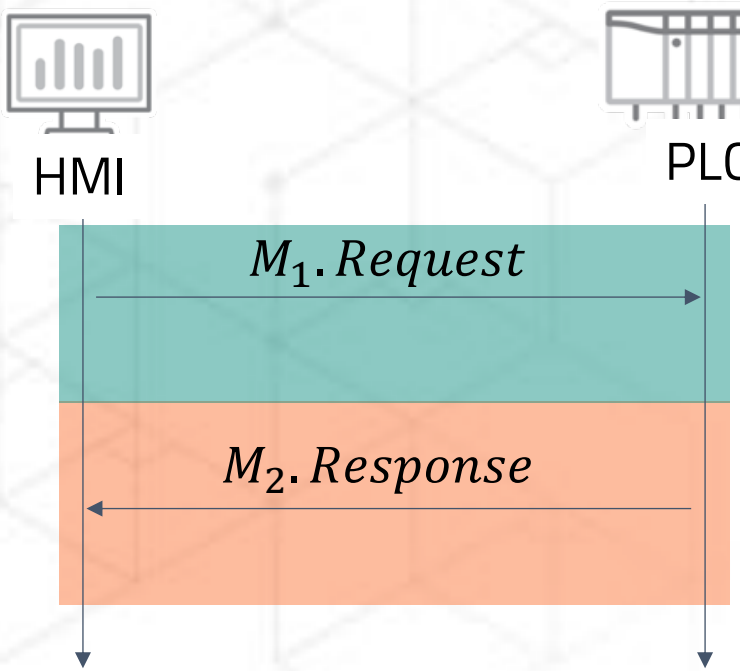
Mitsubishi Melsec/TCP Handshake Process



$$Request \begin{cases} MELSEC = \langle Sub-Header \mid Access\ Route \mid \textbf{Request Data Length} \mid \textbf{Timer} \mid \textbf{Request Data} \rangle \\ TCP \\ IP \\ Ethernet \end{cases}$$

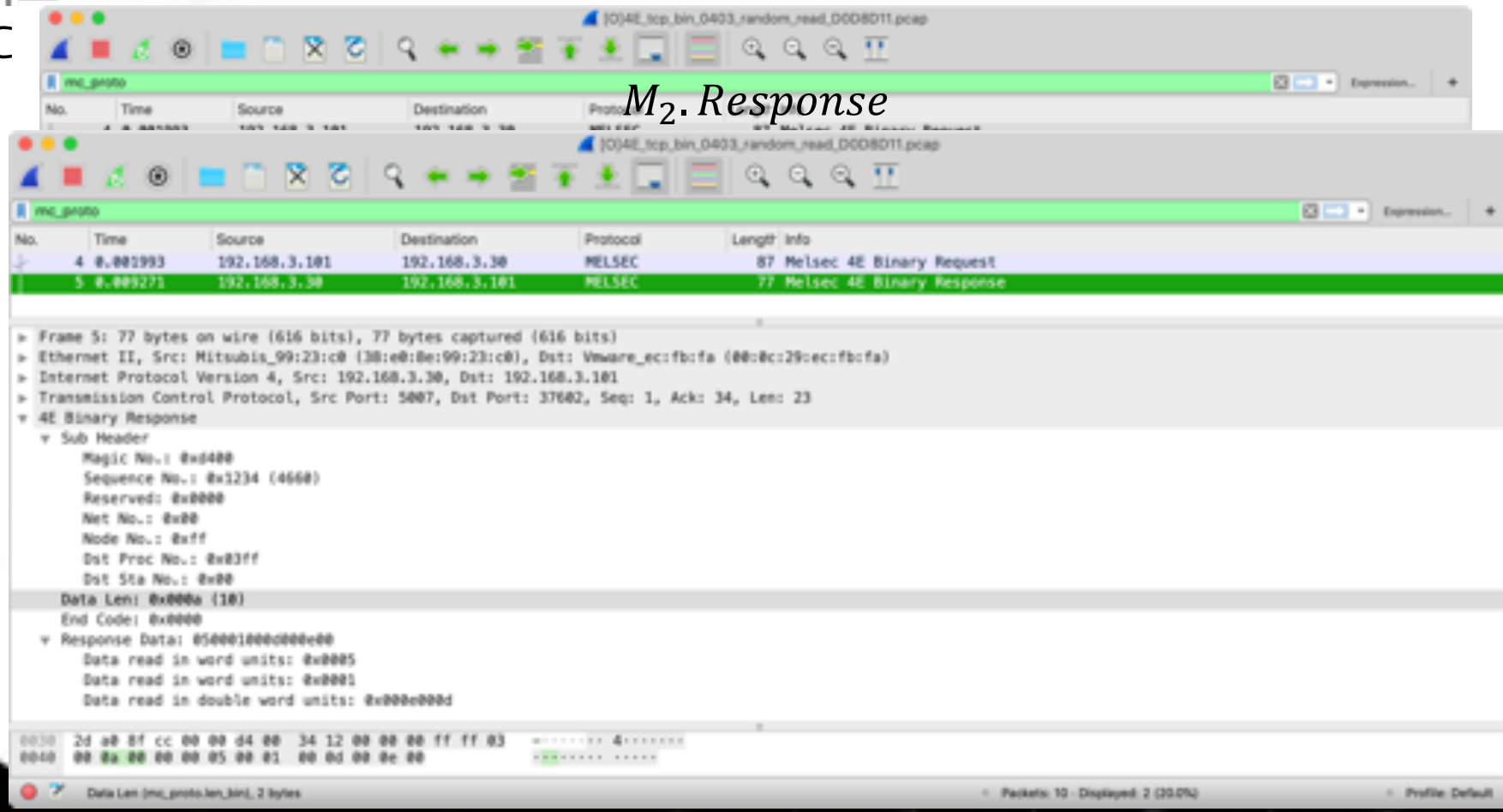
$$Response \begin{cases} MELSEC = \langle Sub-Header \mid Access\ Route \mid \textbf{Response Data Length} \mid \textbf{End Code} \mid \textbf{Response Data} \rangle \\ TCP \\ IP \\ Ethernet \end{cases}$$

Mitsubishi Melsec/TCP Handshake Process



$M_1.Request$

$M_2.Response$



Common Flaws in ICS Network Protocols

Insecure by Design

Type	Protocols	Handshake		Authentication	Message Encryption
Public	Modbus/TCP	TCP Connection		✗	✗
	DNP3/TCP	TCP Connection		✗	✗
	EtherNetIP/CIP	ENIP Connection based		✗	✗
	IEC104	TCP Connection + STARTDT		✗	✗
Private	Melsec/TCP	TCP Connection		✗	✗
	OMRON FINS/TCP	TCP Connection + FINS/TCP session based		✗	✗
	S7COMM	TCP Connection + COTP + S7COMM Session		✗	△(when EWS compile PLC program)
	S7COMM Plus	TCP Connection + COTP + S7COMM+ Session	V1	✗	✗
			V2	✗	✓(HMAC-SHA256)
			V3	✓ (EWS <-> PLC)	✓(HMAC-SHA256)

Attacks on ICS Protocols

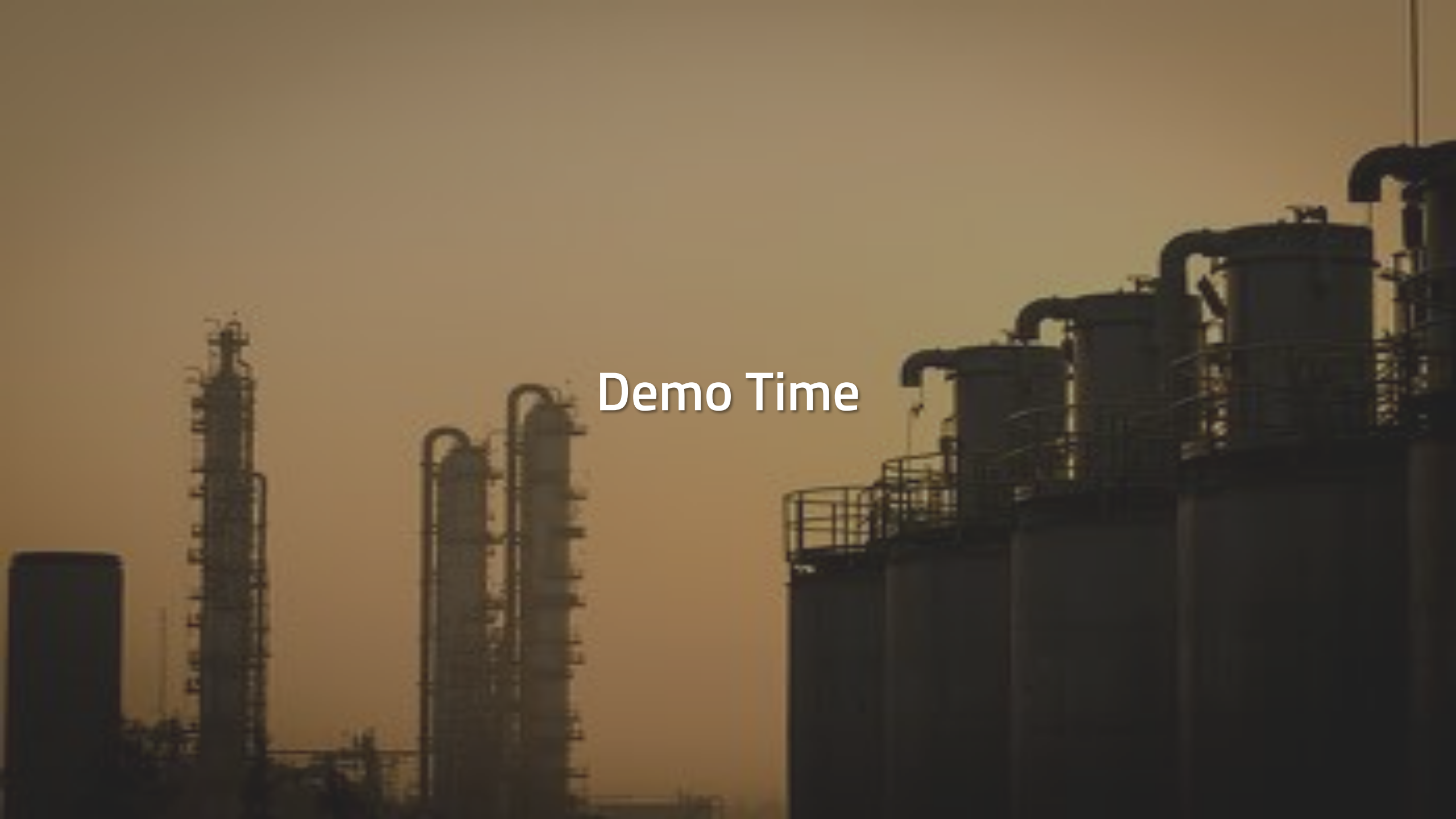
? Unknown

Type	Protocols		T814 Denial-of-Service	T836 Modify Parameter	T856 Spoof Reporting Message	T843 Program Download	T855 Unauthorized Command Message
Public	Modbus/TCP		✓	✓	✓	?	✓
	DNP3/TCP		✓	✓	✓	?	✓
	EtherNetIP/CIP		?	✓	✓	✓	✓
	IEC104		✓	✓	✓	?	✓
Private	Melsec/TCP		?	✓	✓	✓	✓
	OMRON FINS/TCP		?	✓	✓	✓	✓
	S7COMM		✓	✓	✓	✓	✓
	S7COMM Plus	V1	?	✓	?	✓	✓
		V2	?	✓	?	✓	✓
		V3	?	✓	?	✓	✓

ICS ATT&CK Matrix map to ICS Protocols Attack

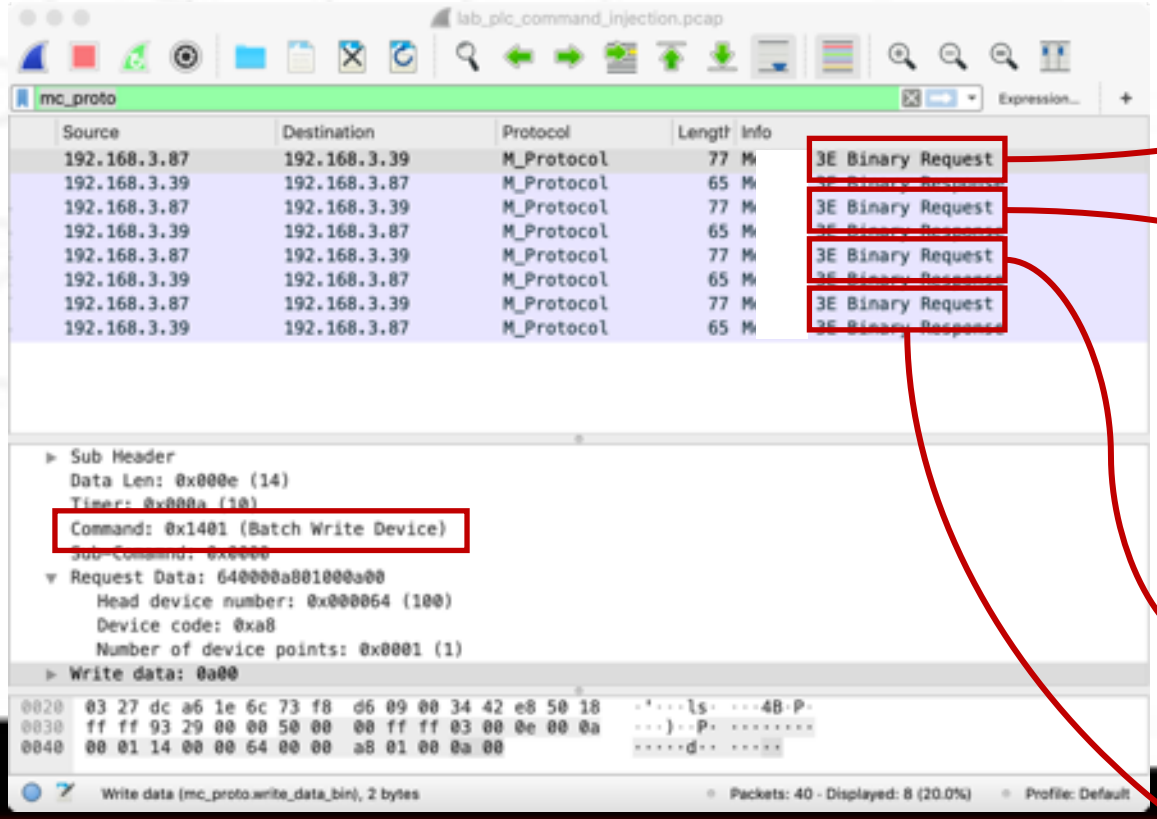
Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial Comm Port	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

11 Tactics
81 Techniques


Demo Time

T836-Modify Parameter with Mitsubishi Melsec

T836-Modify Parameter



10s

```
Internet Protocol Version 4, Src: 192.168.3.87, Dst: 192.168.3.39
Transmission Control Protocol, Src Port: 56486, Dst Port: 7788, Seq: 1, Ack: 1, Len: 23
3E Binary Request
  Sub Header
    Data Len: 0x000e (14)
    Timer: 0x000a (10)
    Command: 0x1401 (Batch Write Device)
    Sub-Comamnd: 0x0000
  Request Data: 640000a801000a00
    Head device number: 0x000064 (100)
    Device code: 0xa8
    Number of device points: 0x0001 (1)
  Write data: 0a00
```

```
Internet Protocol Version 4, Src: 192.168.3.87, Dst: 192.168.3.39
Transmission Control Protocol, Src Port: 56497, Dst Port: 7788, Seq: 1, Ack: 1, Len: 23
3E Binary Request
  Sub Header
    Data Len: 0x000e (14)
    Timer: 0x000a (10)
    Command: 0x1401 (Batch Write Device)
    Sub-Comamnd: 0x0000
  Request Data: 640000a801000200
    Head device number: 0x000064 (100)
    Device code: 0xa8
    Number of device points: 0x0001 (1)
  Write data: 0200
```

2s

```
Internet Protocol Version 4, Src: 192.168.3.87, Dst: 192.168.3.39
Transmission Control Protocol, Src Port: 56510, Dst Port: 7788, Seq: 1, Ack: 1, Len: 23
3E Binary Request
  Sub Header
    Data Len: 0x000e (14)
    Timer: 0x000a (10)
    Command: 0x1401 (Batch Write Device)
    Sub-Comamnd: 0x0000
  Request Data: 640000a801000800
    Head device number: 0x000064 (100)
    Device code: 0xa8
    Number of device points: 0x0001 (1)
  Write data: 0800
```

8s

```
Internet Protocol Version 4, Src: 192.168.3.87, Dst: 192.168.3.39
Transmission Control Protocol, Src Port: 56521, Dst Port: 7788, Seq: 1, Ack: 1, Len: 23
3E Binary Request
  Sub Header
    Data Len: 0x000e (14)
    Timer: 0x000a (10)
    Command: 0x1401 (Batch Write Device)
    Sub-Comamnd: 0x0000
  Request Data: 640000a801001e00
    Head device number: 0x000064 (100)
    Device code: 0xa8
    Number of device points: 0x0001 (1)
  Write data: 1e00
```

30s

T843-Program Download with Mitsubishi Melsoft

T856-Spoof Reporting Message with Modbus/TCP

T855-Unauthorized Command Message with Omron FINS

Common Flaws in ICS Protocols

T880-Loss of Safety

T829-Loss of View

T827-Loss of Control

T826-Loss of Availability

T815-Denial of View

Impact

T813-Denial of Control

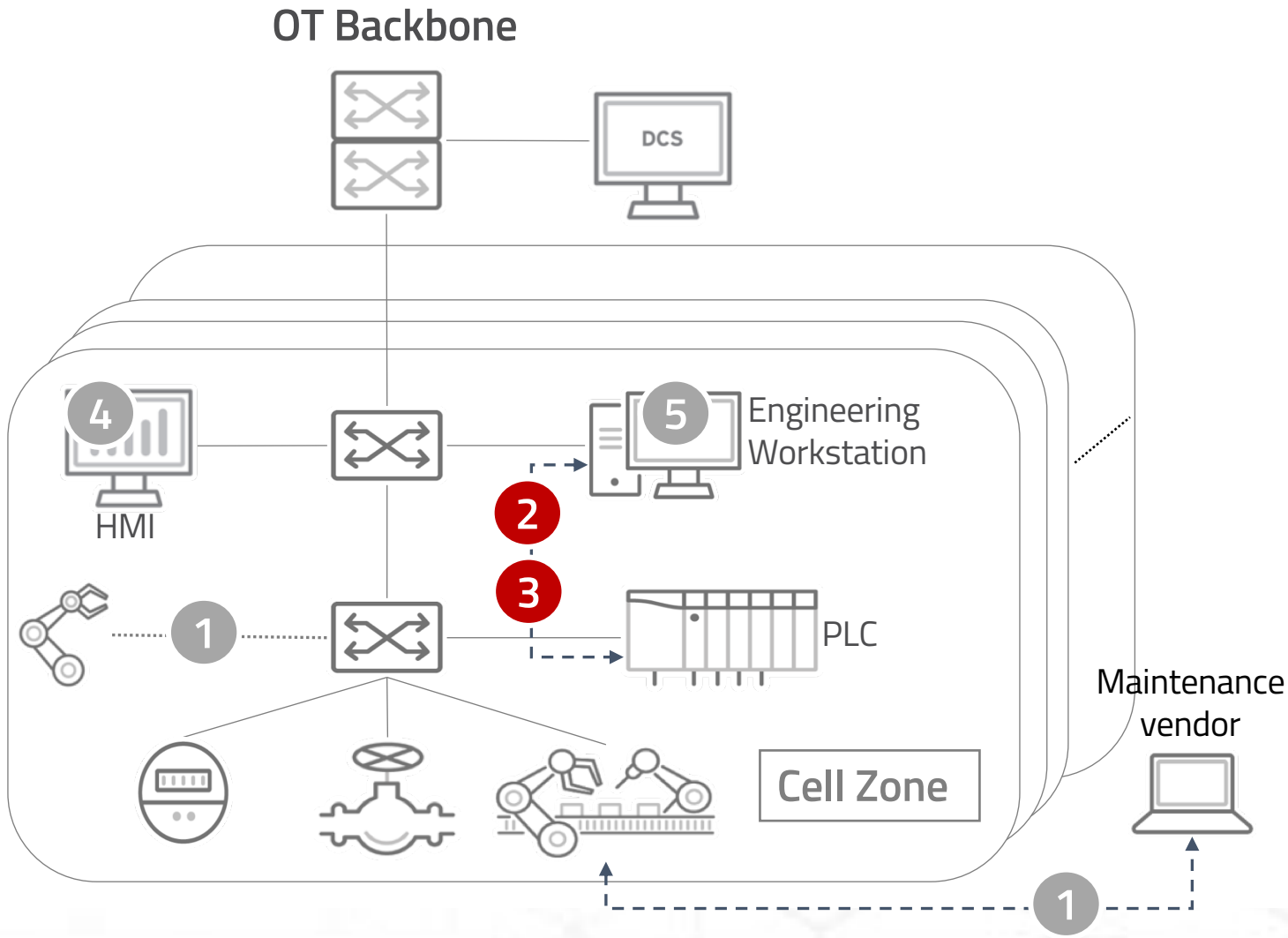
T832-Manipulation of View

T831-Manipulation of Control

How to Defend Against ICS Network Protocol Attacks

Vulnerable OT Environment

1. **Shadow OT**
   - Unknown devices and unknown connections
2. **Insecure Authentication**
   - Flaws come from design or implementation
3. **Insecure Protocols**
   - Simply unencrypted
4. **Unpatched Devices**
   - Patching is not feasible or available
5. **Insecure 3rd-Party Software**
   - Vulnerable, and might be compromised from supply chain in the beginning



Suggested Strategies from ICS CERT

Manage authentication
4%

Implement application whitelisting
38%

Implement secure remote access
1%

Monitor and respond
2%

Configuration/patch mgmt.
29%

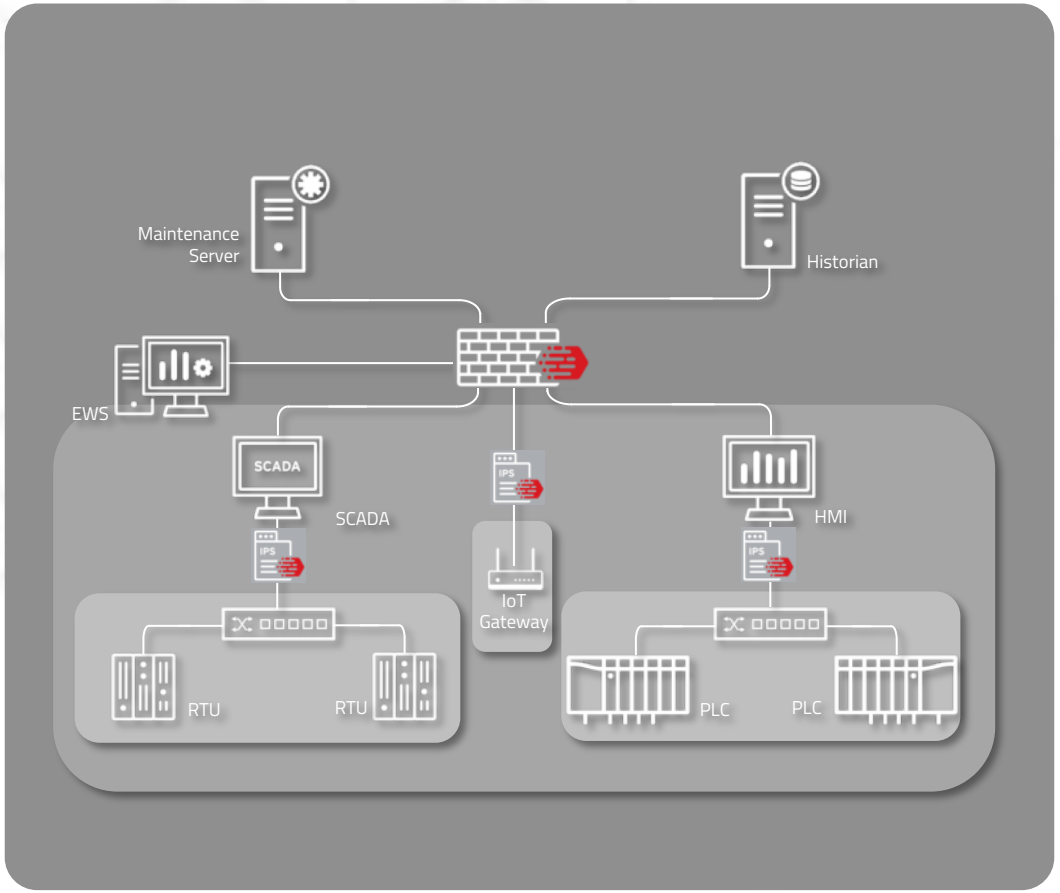
Build a defendable environment
9%

Reduce your attack surface
17%

Incidents responded by ICS-CERT : https://ics-cert.us-cert.gov/sites/default/files/documents/Seven%20Steps%20to%20Effectively%20Defend%20Industrial%20Control%20Systems_S508C.pdf

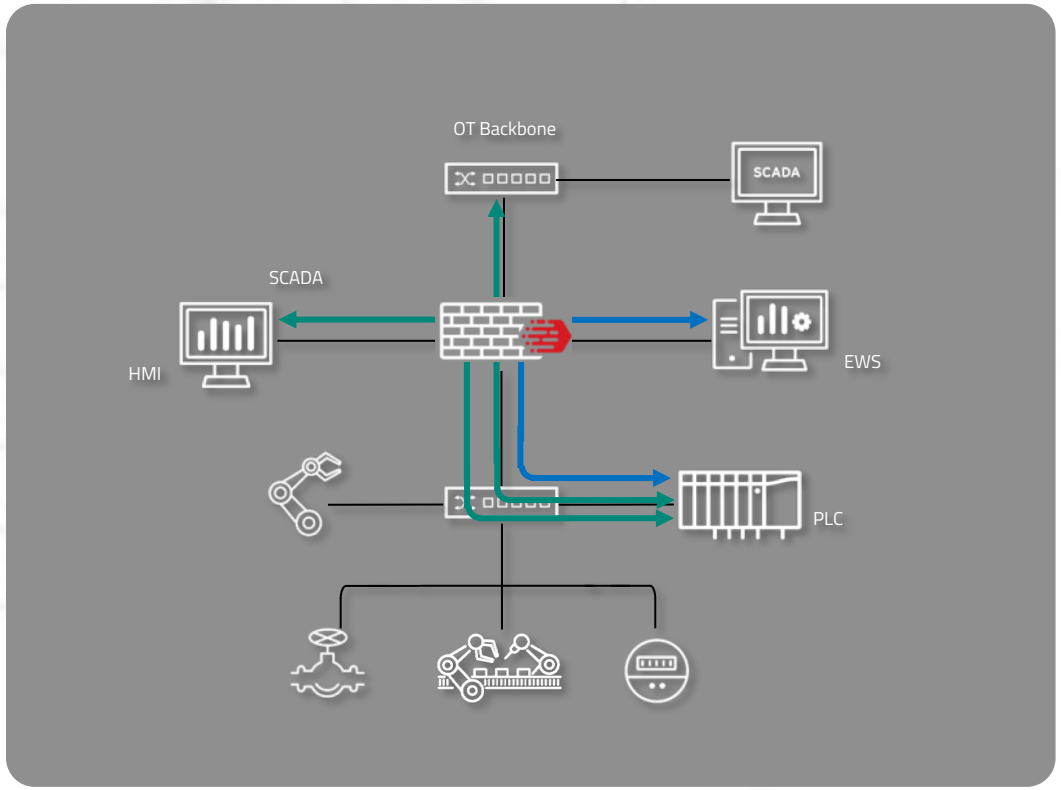
Implementing FIVE Tactics to prevent 98% incidents

Effective Segmentation, Virtual Patch, Containment



- Divide a big flat L2 network into secured segments
- Virtual Patch (IPS)
  - Containment of malware and worms
  - Shield device vulnerabilities
  - Deeply inspect IT protocols: SMB, RDP, …
- Industrial-Grade Hardware

Granular Control Over Popular OT Protocols



- Asset and protocol visibility
- Fine-grained access control in different levels
  - Devices
  - Protocols (Modbus, Melsec/SLMP, CC-Link IE, Ethernet/IP, Profinet, S7COMM, HSMS/SECS-II, …)
  - Control Commands (read, configure, shutdown, …)
- Greatly lower the possibility of Denial-of-Service attacks by OT trojans

Read Only Full Access

Network Whitelisting Control against Siemens S7 attack


AtCP

Thank You!

HITB LOCKDOWN 002
livestream

Mars Cheng and Selmon Yang, mars_cheng@txone-networks.com