Responding

To the evolving threat

OSIsoft Super Regional 2018
August 21 2018

Ben Miller
bmiller@dragos.com
@electricfork

DRAGOS

Industrial Impacts

(The Public Ones)

Sewage Spill
2000

Centrifuge Failure
2010

Telvent Espionage
2012

Havex Espionage
2014

Blackouts
2015 & 2016

Safety Systems
2017

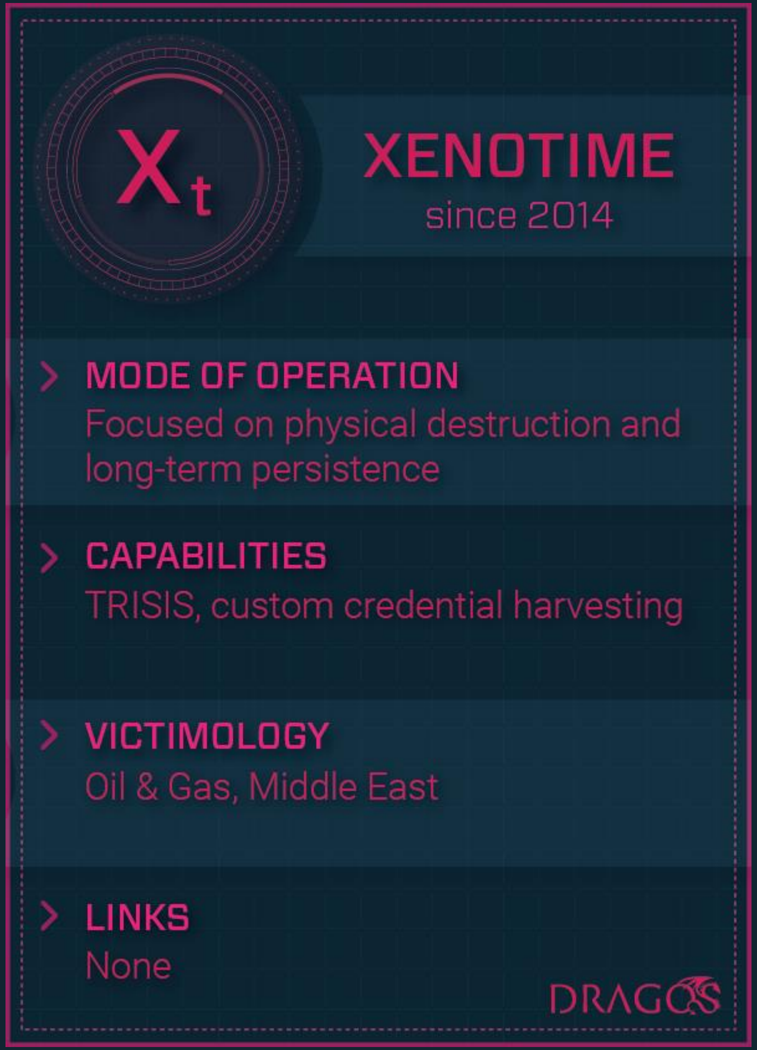
Defenders

(IT and OT)

Attackers

(IT and OT)

State of the art



XENOTIME

since 2014

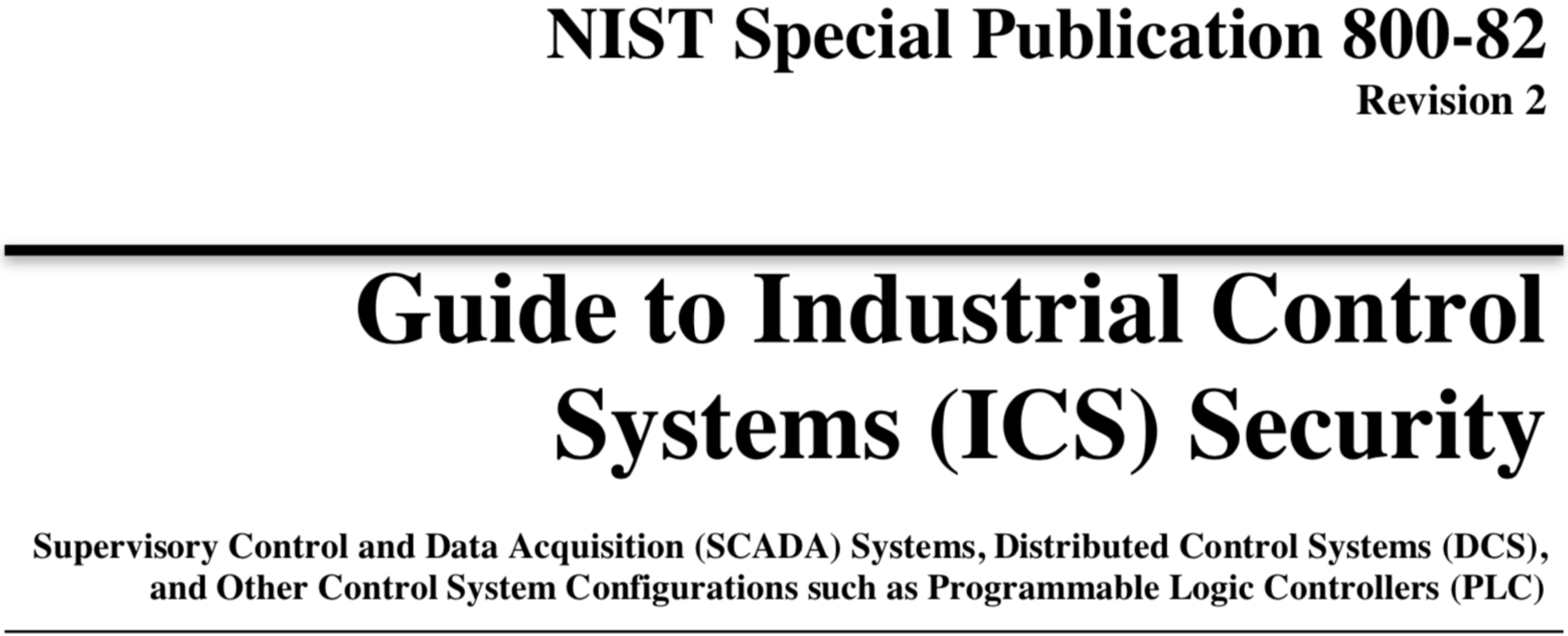
- **MODE OF OPERATION**
  Focused on physical destruction and long-term persistence
- **CAPABILITIES**
  TRISIS, custom credential harvesting
- **VICTIMOLOGY**
  Oil & Gas, Middle East
- **LINKS**
  None

DRAGOS

TRISIS

Authored by XENOTIME

- TARGET
  Triconex Safety Systems (3008 / PowerPC)
- CAPABILITIES
  Memory Resident Rootkit
- CLASSIFICATION
  Memory Resident Rootkit
- DELIVERY
  Windows host with network access via legitimate Tristation Protocol



NIST Special Publication 800-82

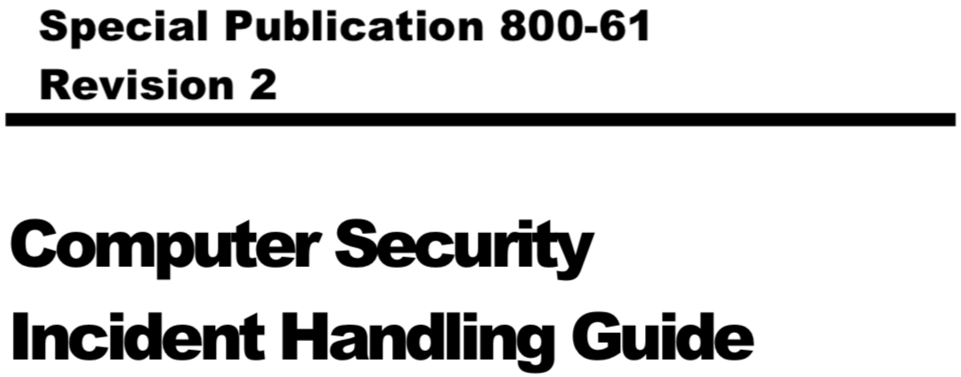
Revision 2

Guide to Industrial Control Systems (ICS) Security

Supervisory Control and Data Acquisition (SCADA) Systems, Distributed Control Systems (DCS), and Other Control System Configurations such as Programmable Logic Controllers (PLC)

Incident Response

NIST
National Institute of Standards and Technology
U.S. Department of Commerce



Special Publication 800-61
Revision 2

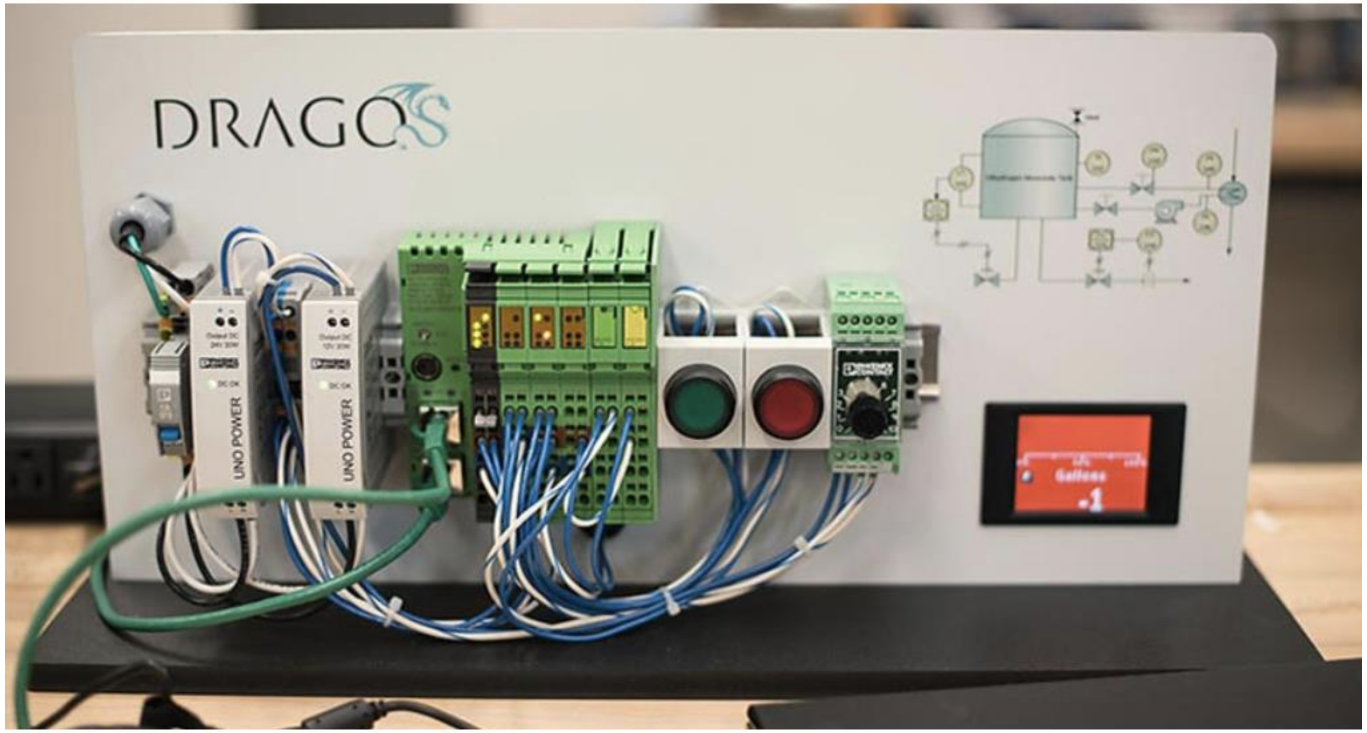
Computer Security Incident Handling Guide

- Respond systematically to events and incidents
- Make sure the appropriate actions are taken
- Minimize impact caused by incidents
- Apply lessons to future incidents and how they are handled

Detection and Analysis

- Attack Vectors
- Signs of an Incident
- Sources of Precursors and Indications
- Incident Analysis
- Incident Documentation
- Incident Prioritization
- Incident Notification

ILC 191 ETH 2TX



User manual

UM EN ILC 1XX

Installing and operating the ILC 130 ETH,
ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX,
and ILC 190 ETH 2TX Inline controllers

PHOENIX CONTACT
INSPIRING INNOVATIONS

What Forensically Matters

- Where is the serial number / model number?
- How do you identify the MAC Address? IP Address?
- Do we know what the embedded OS is?
- What interfaces exist?
- Which interfaces can you download programs or update firmware?
- Is there removable storage?
- What is stored on the removable storage?
- What file system is used on the removable storage?
- What modes are possible and implications?

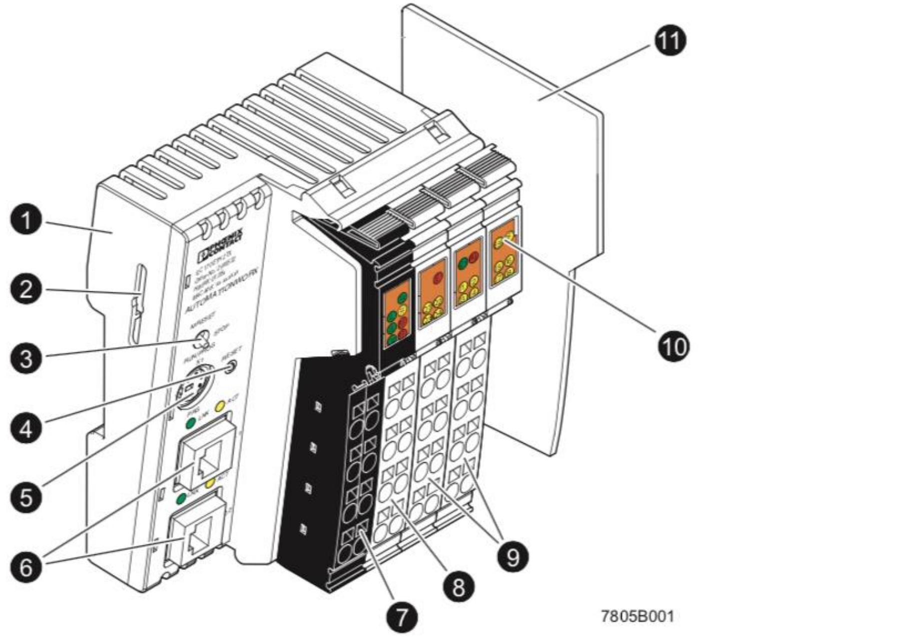


Figure 2-8 Structure of the Inline controller (ILC 170 ETH 2TX, ILC 190 ETH 2TX; shown in the figure: ILC 170 ETH 2TX)

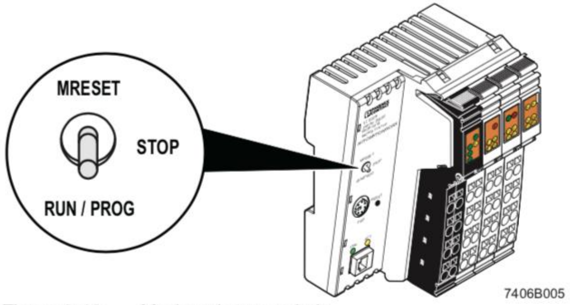
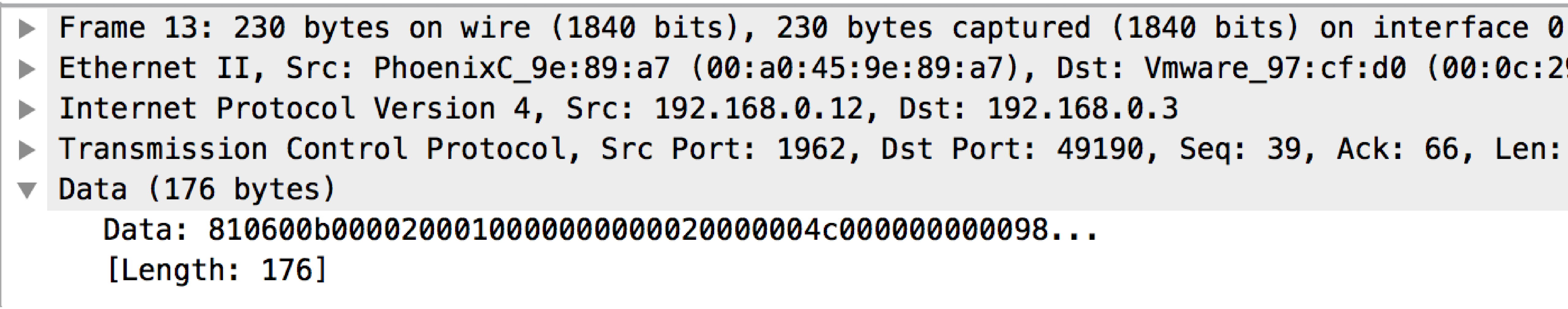
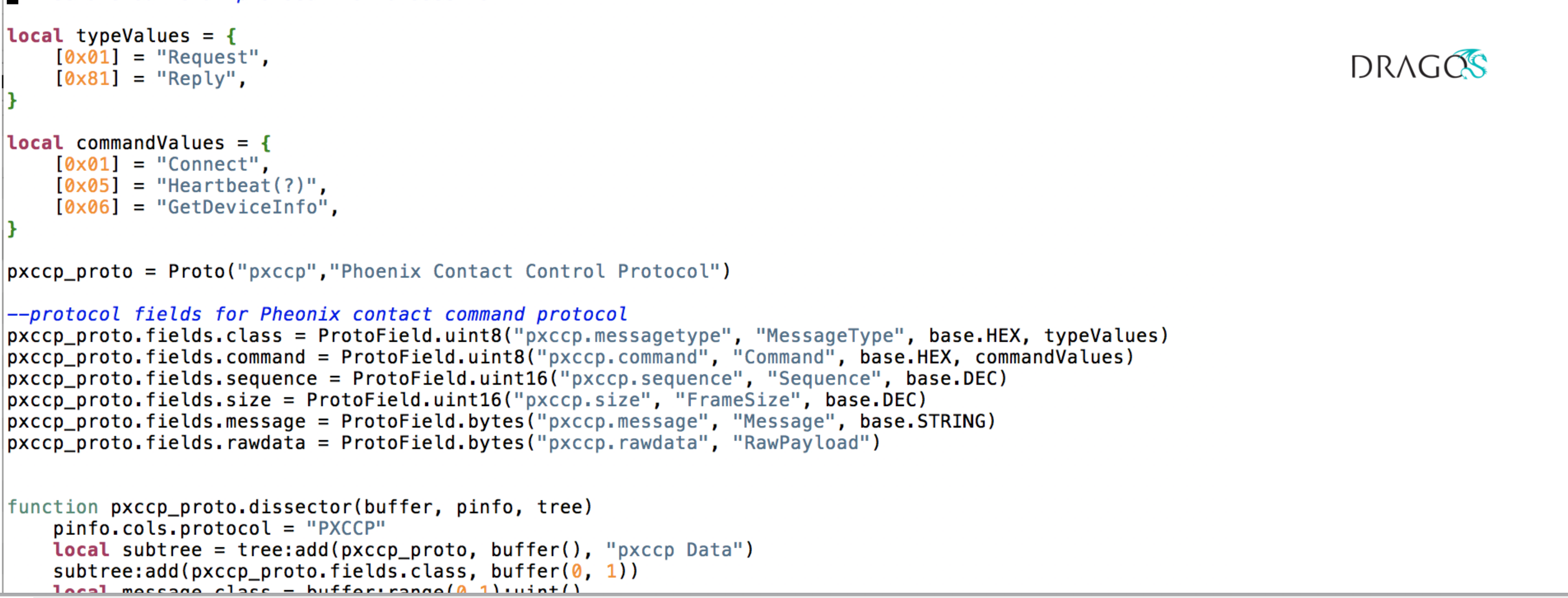


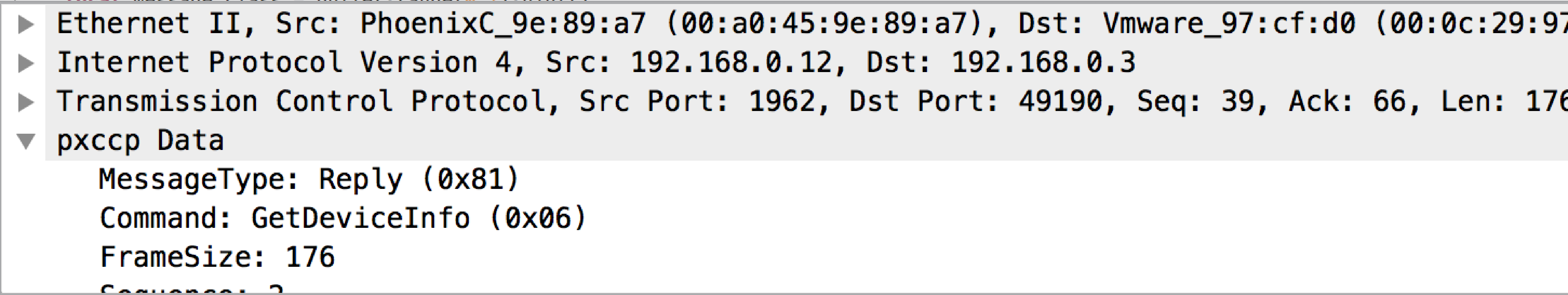
Figure 2-10 Mode selector switch



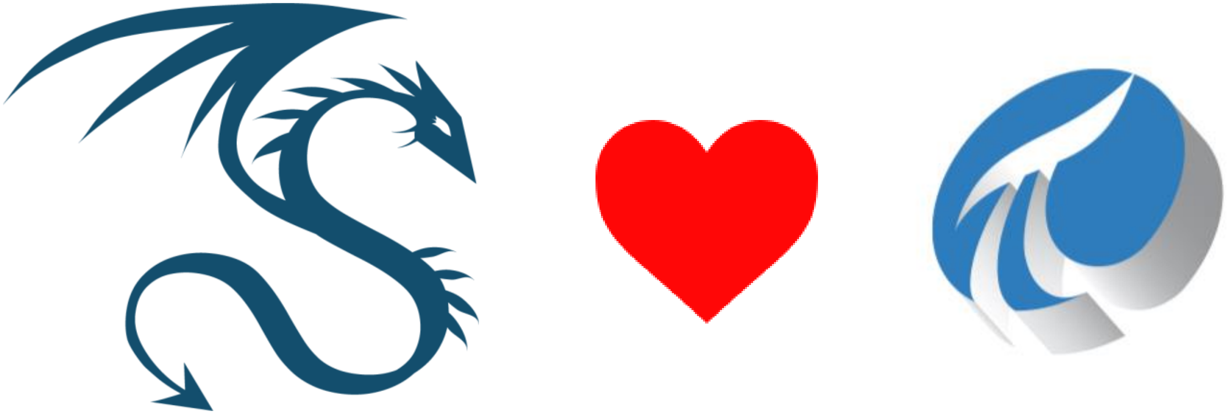
```
▶ Frame 13: 230 bytes on wire (1840 bits), 230 bytes captured (1840 bits) on interface 0
▶ Ethernet II, Src: PhoenixC_9e:89:a7 (00:a0:45:9e:89:a7), Dst: Vmware_97:cf:d0 (00:0c:2
▶ Internet Protocol Version 4, Src: 192.168.0.12, Dst: 192.168.0.3
▶ Transmission Control Protocol, Src Port: 1962, Dst Port: 49190, Seq: 39, Ack: 66, Len:
▼ Data (176 bytes)
    Data: 810600b0000200010000000000002000004c000000000098...
    [Length: 176]
```



```
local typeValues = {
    [0x01] = "Request",
    [0x81] = "Reply",
}

local commandValues = {
    [0x01] = "Connect",
    [0x05] = "Heartbeat(?)",
    [0x06] = "GetDeviceInfo",
}

pxccp_proto = Proto("pxccp","Phoenix Contact Control Protocol")

--protocol fields for Pheonix contact command protocol
pxccp_proto.fields.class = ProtoField.uint8("pxccp.messagetype", "MessageType", base.HEX, typeValues)
pxccp_proto.fields.command = ProtoField.uint8("pxccp.command", "Command", base.HEX, commandValues)
pxccp_proto.fields.sequence = ProtoField.uint16("pxccp.sequence", "Sequence", base.DEC)
pxccp_proto.fields.size = ProtoField.uint16("pxccp.size", "FrameSize", base.DEC)
pxccp_proto.fields.message = ProtoField.bytes("pxccp.message", "Message", base.STRING)
pxccp_proto.fields.rawdata = ProtoField.bytes("pxccp.rawdata", "RawPayload")


function pxccp_proto.dissector(buffer, pinfo, tree)
    pinfo.cols.protocol = "PXCCP"
    local subtree = tree:add(pxccp_proto, buffer(), "pxccp Data")
    subtree:add(pxccp_proto.fields.class, buffer(0, 1))
    local message_class = buffer:range(0,1):uint()
```



```
▶ Ethernet II, Src: PhoenixC_9e:89:a7 (00:a0:45:9e:89:a7), Dst: Vmware_97:cf:d0 (00:0c:29:97
▶ Internet Protocol Version 4, Src: 192.168.0.12, Dst: 192.168.0.3
▶ Transmission Control Protocol, Src Port: 1962, Dst Port: 49190, Seq: 39, Ack: 66, Len: 176
▼ pxccp Data
      MessageType: Reply (0x81)
      Command: GetDeviceInfo (0x06)
      FrameSize: 176
      Sequence: 2
```



Thank You.

www.dragos.com
info@dragos.com
@dragos_inc

1745 Dorsey Road
Hanover, MD 21076