Out of Control: Demonstrating SCADA Exploitation

Brian Meixell

Eric Forner

Black Hat 2013

Agenda

- SCADA 101
- Attack Scenarios
- Common Vulnerabilities
- Remediation
- Exploit Demo

SCADA 101

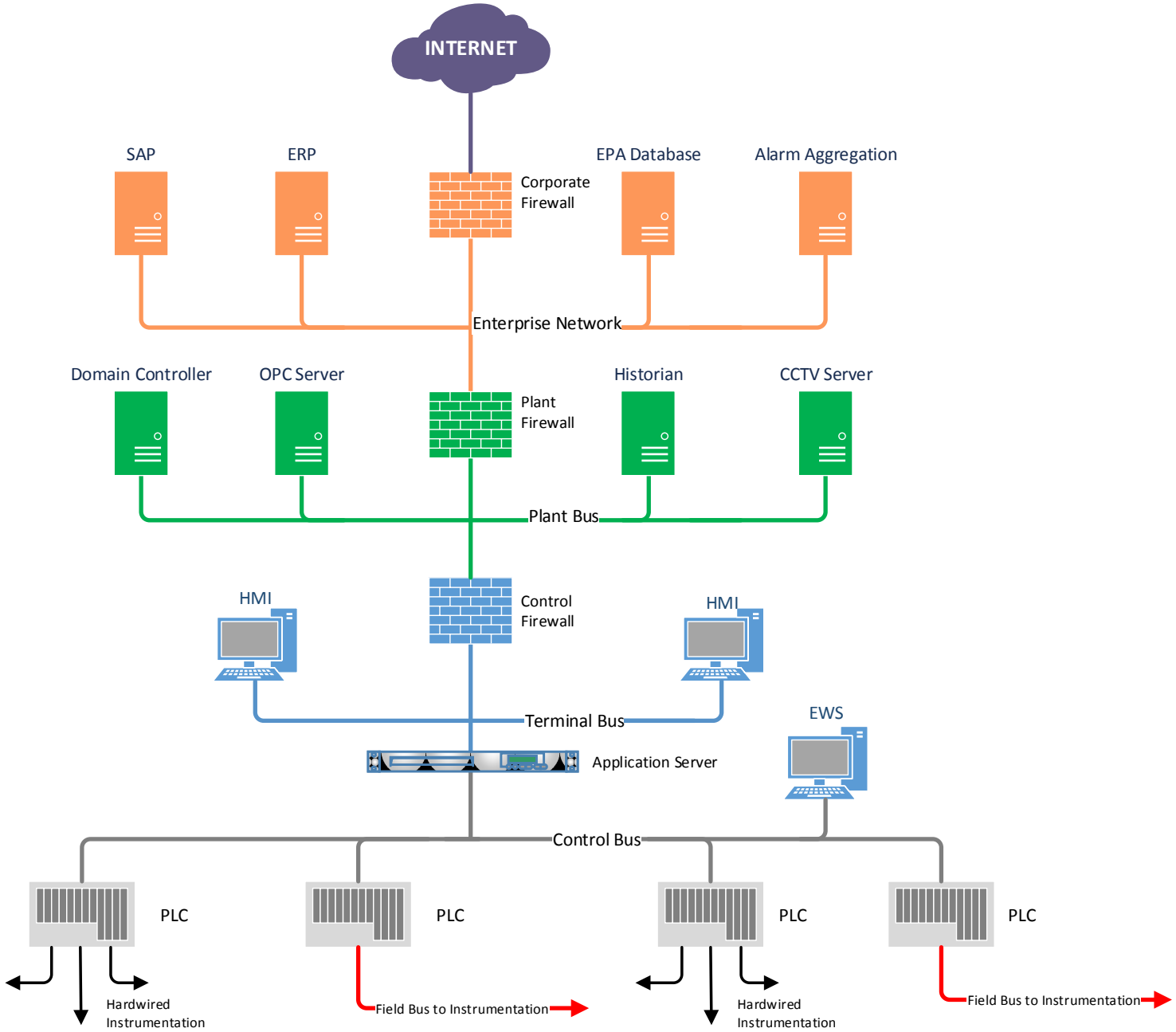
SCADA

Supervisory
Control
And
Data
Acquisition

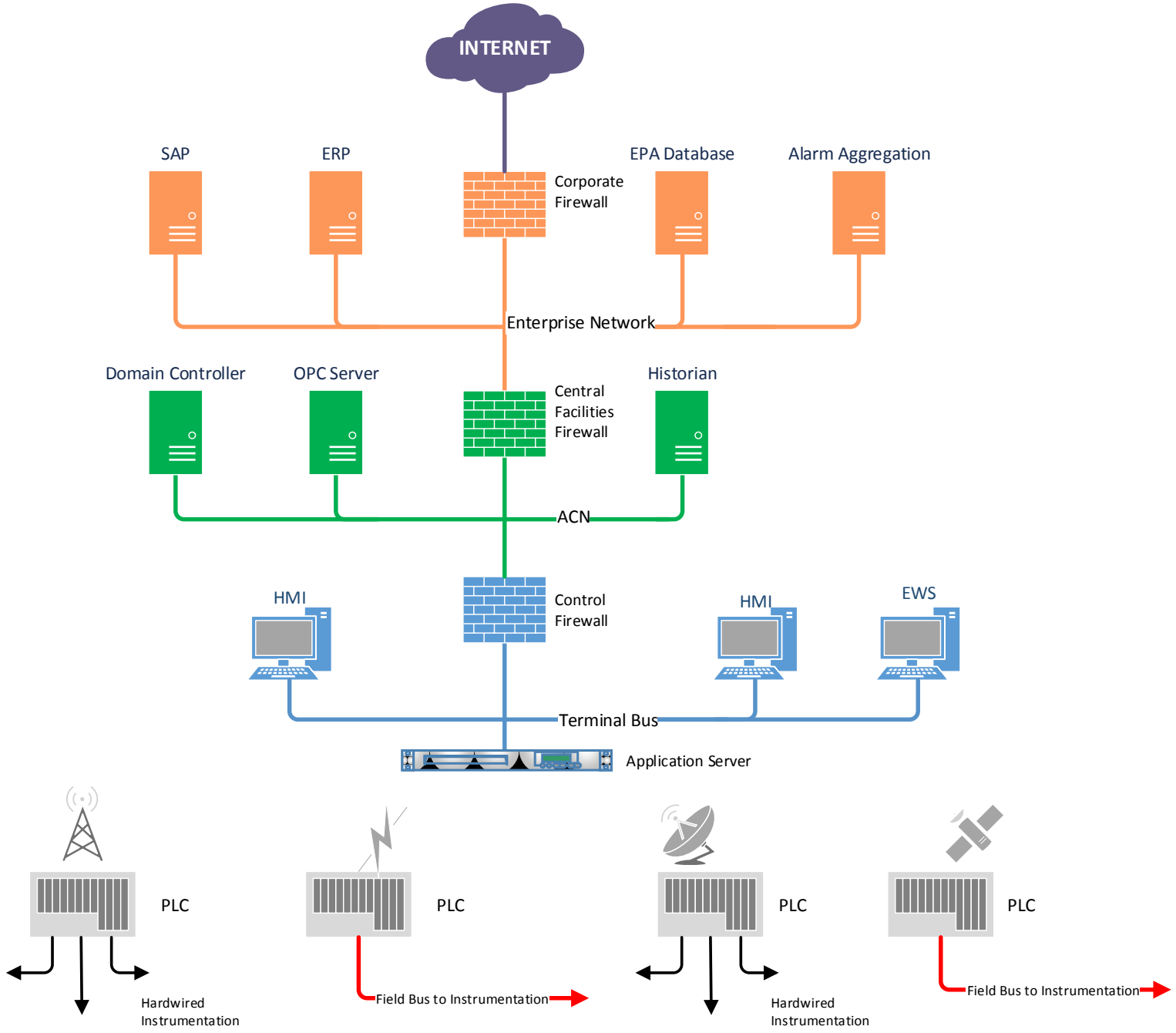
DCS

Distributed
Control
System

Standard DCS Network

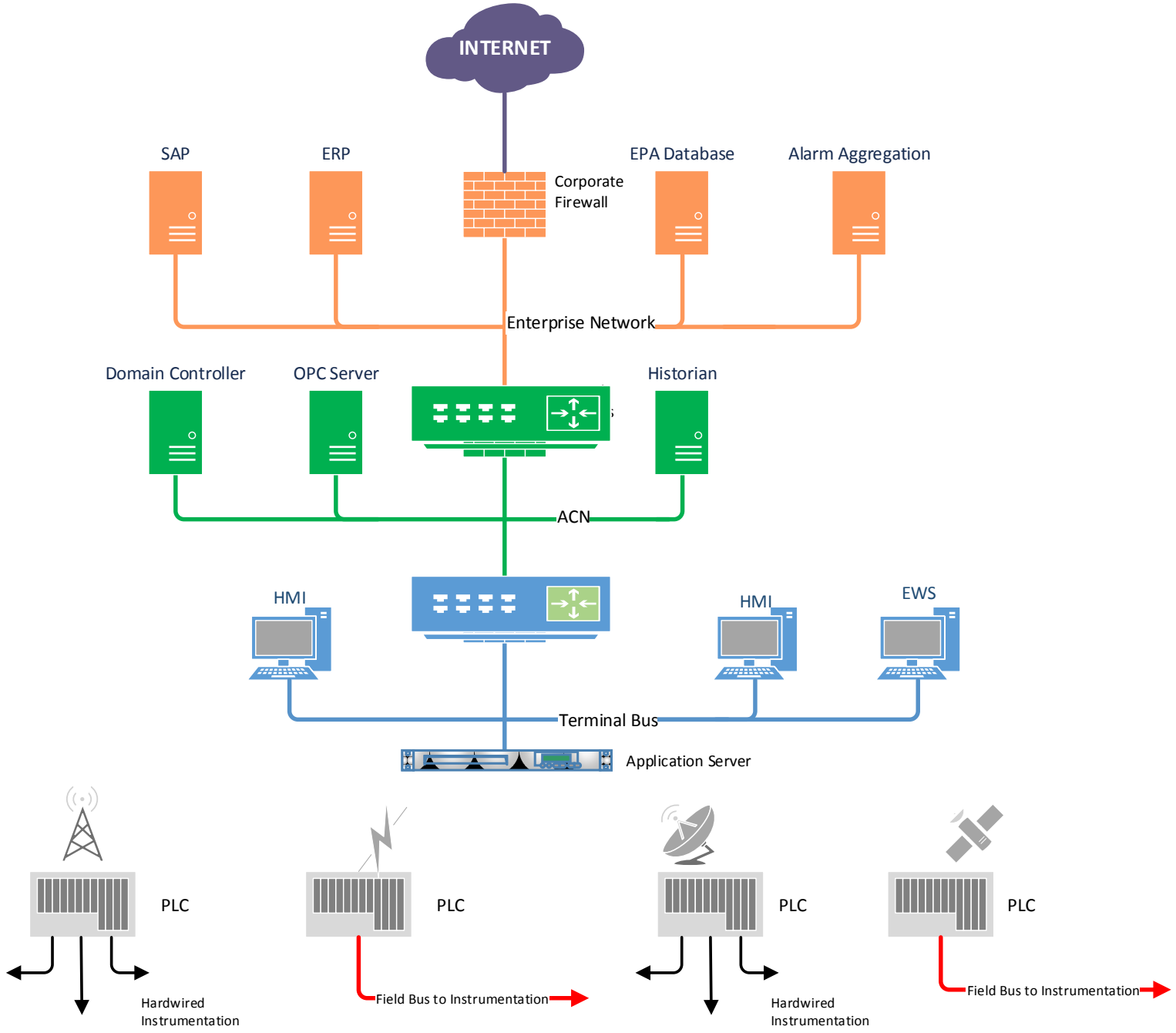


Standard SCADA Network

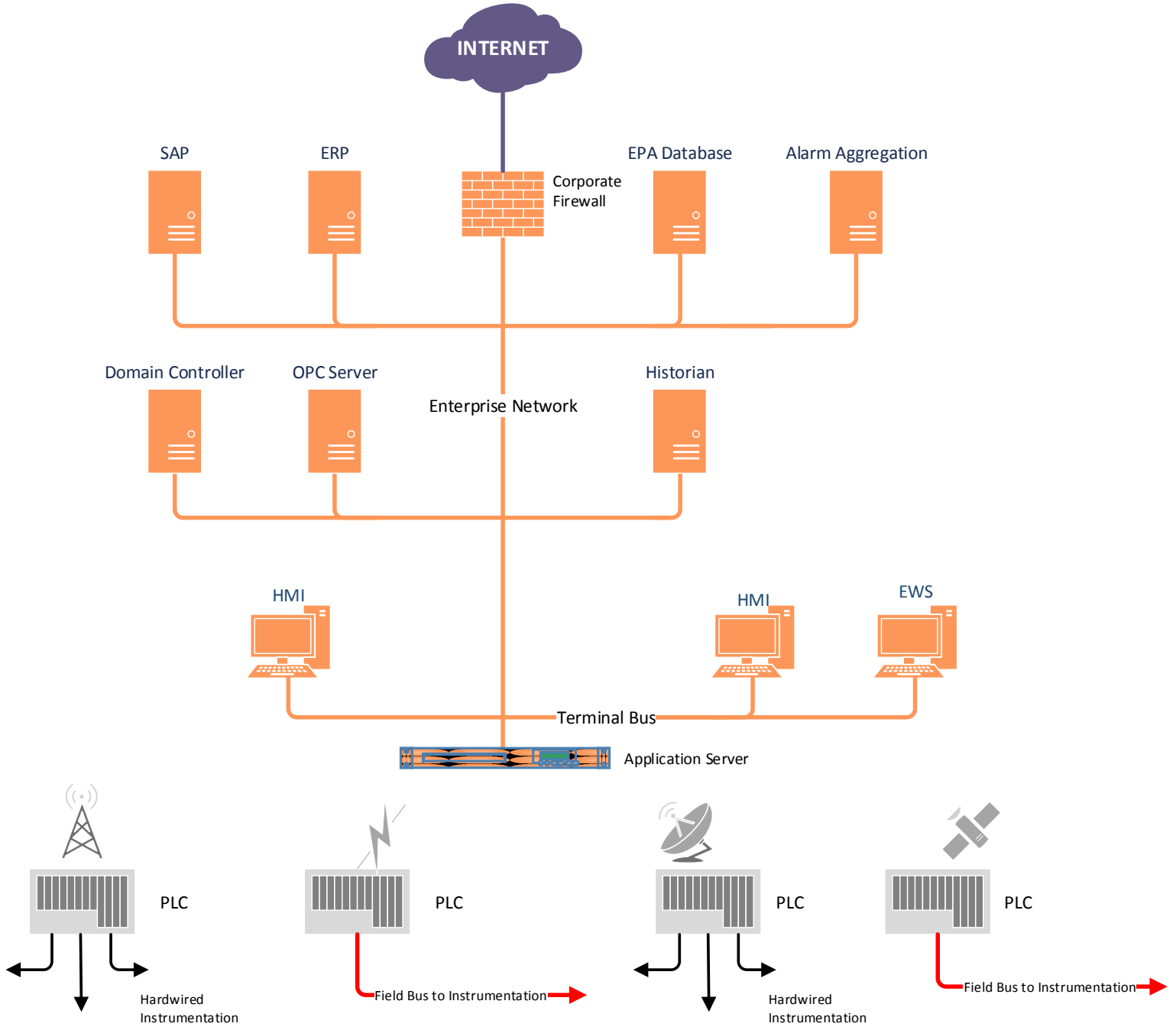


~~Standard~~ Typical SCADA Network

My Favorite Firewall Rule
ANY <---> ANY

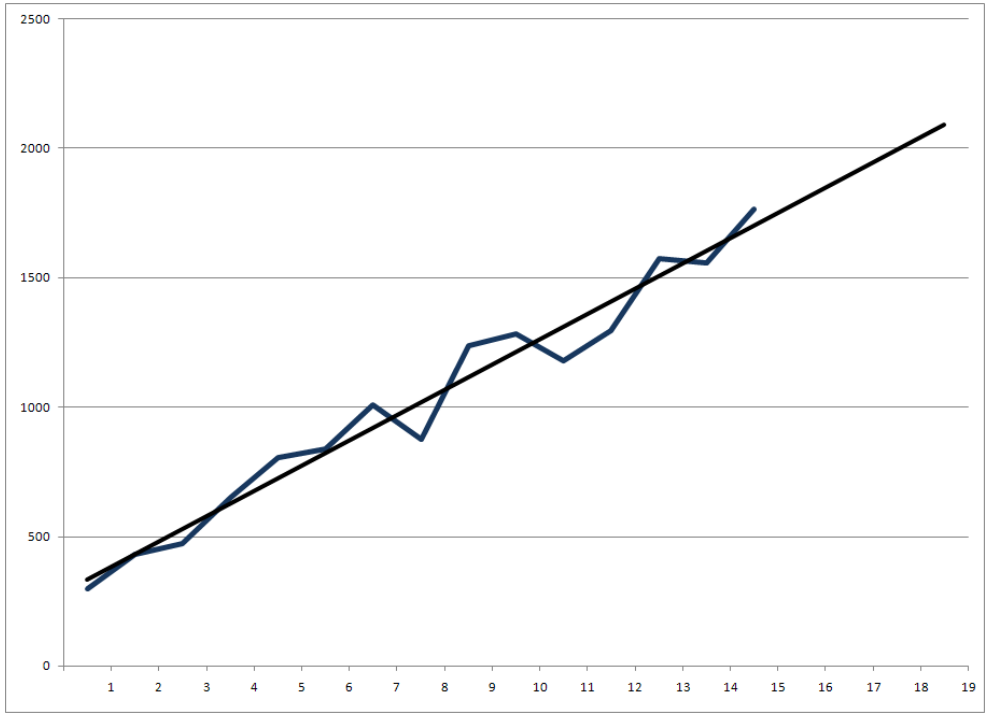


Or this...


INTERNET
SAP
ERP
Corporate Firewall
EPA Database
Alarm Aggregation
Domain Controller
OPC Server
Enterprise Network
Historian
HMI
HMI
EWS
Terminal Bus
Application Server
PLC
PLC
PLC
PLC
Hardwired Instrumentation
Field Bus to Instrumentation
Hardwired Instrumentation
Field Bus to Instrumentation

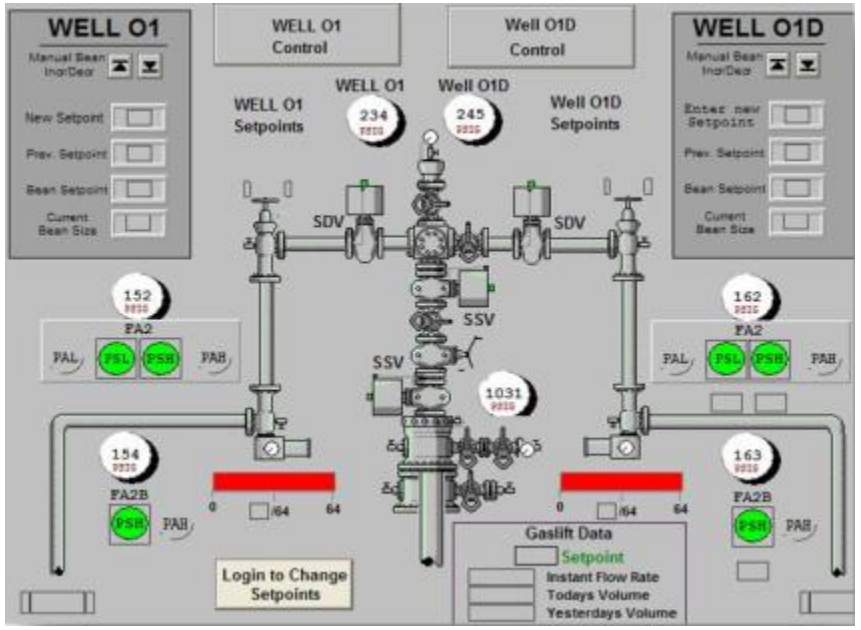
Components

Historian



Components

Human Machine Interface

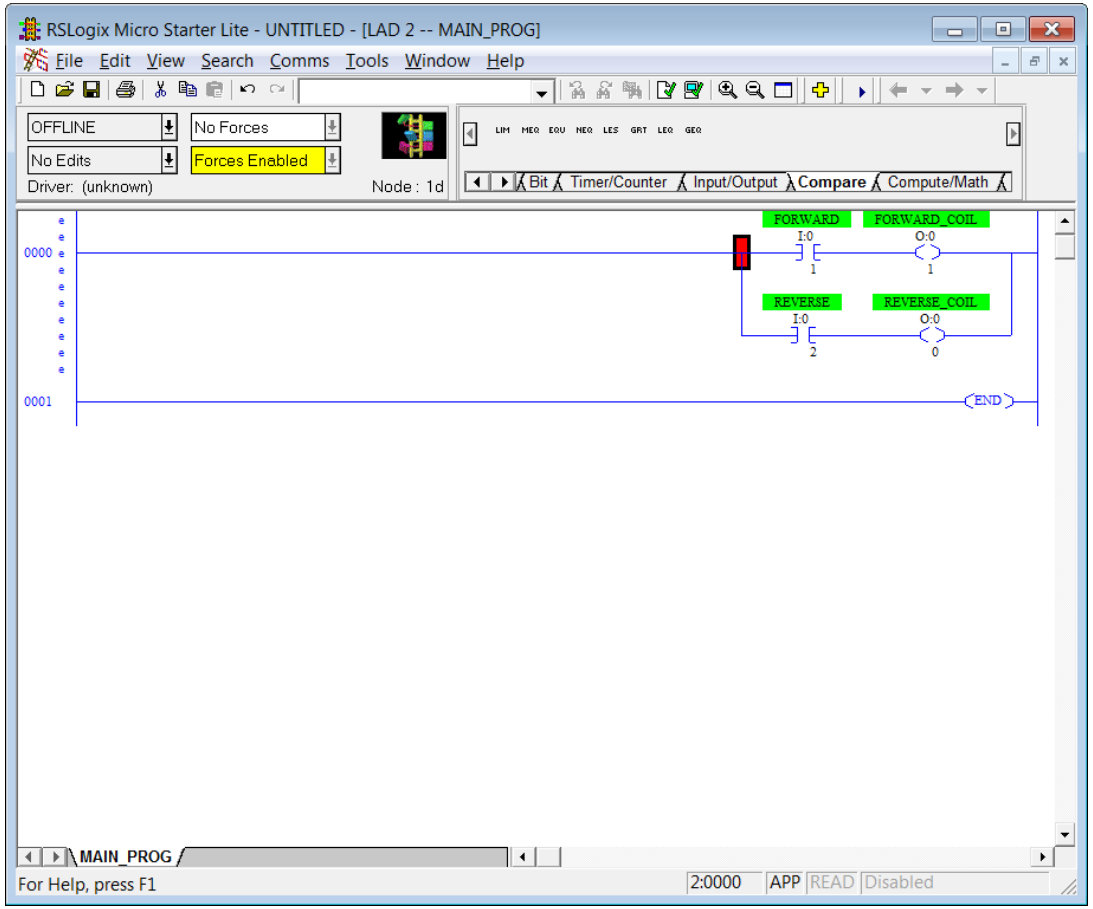


Components

Application Server

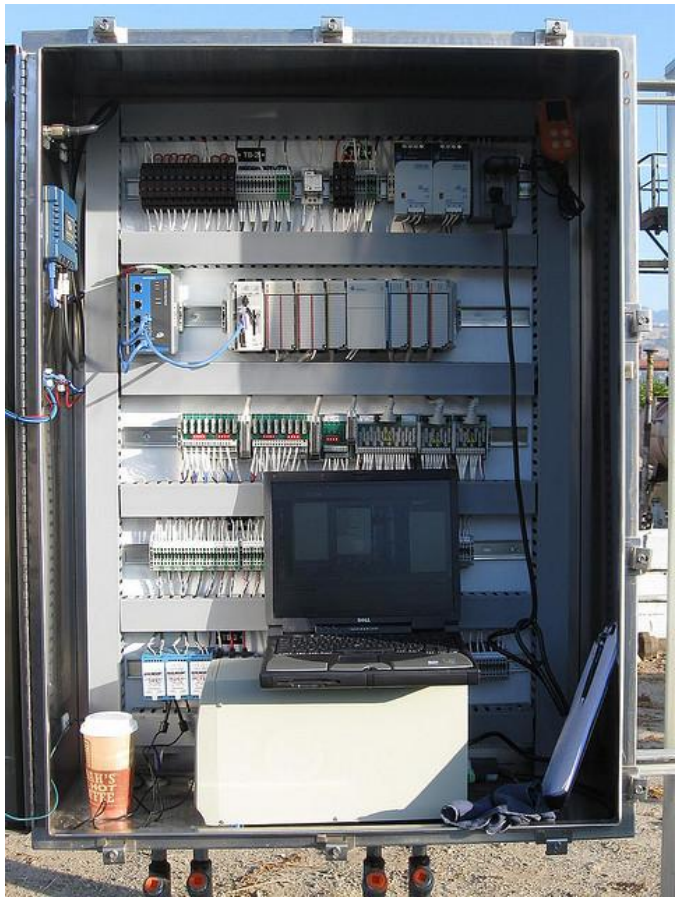
Components

Engineering Workstation



Components

Programmable Logic Controller

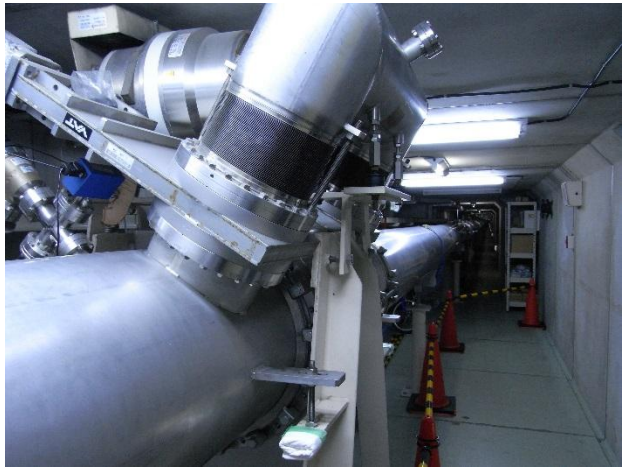


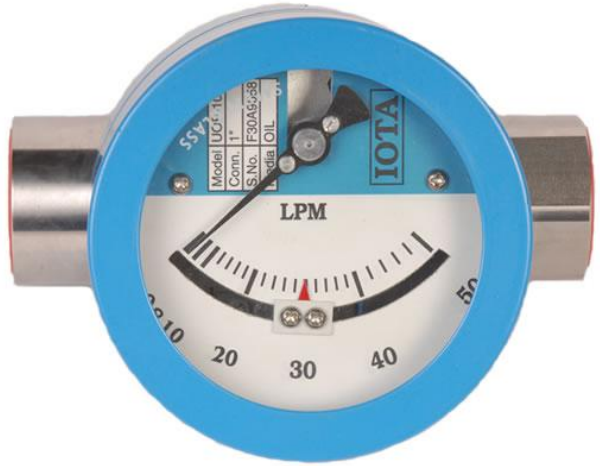
Components

Remote Terminal Unit

Components

Instrumentation



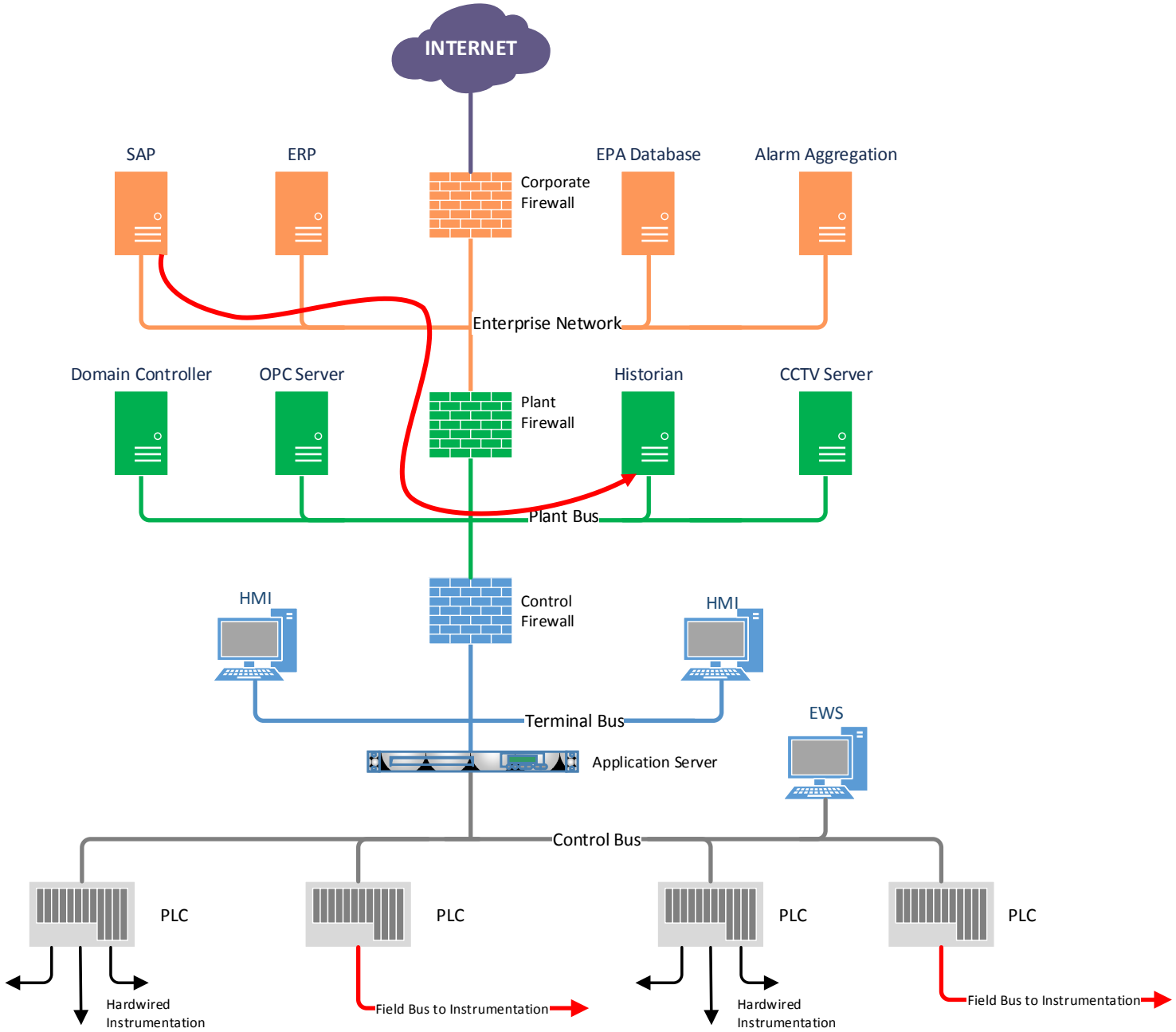


Attack Vectors

It’s all about the pivot

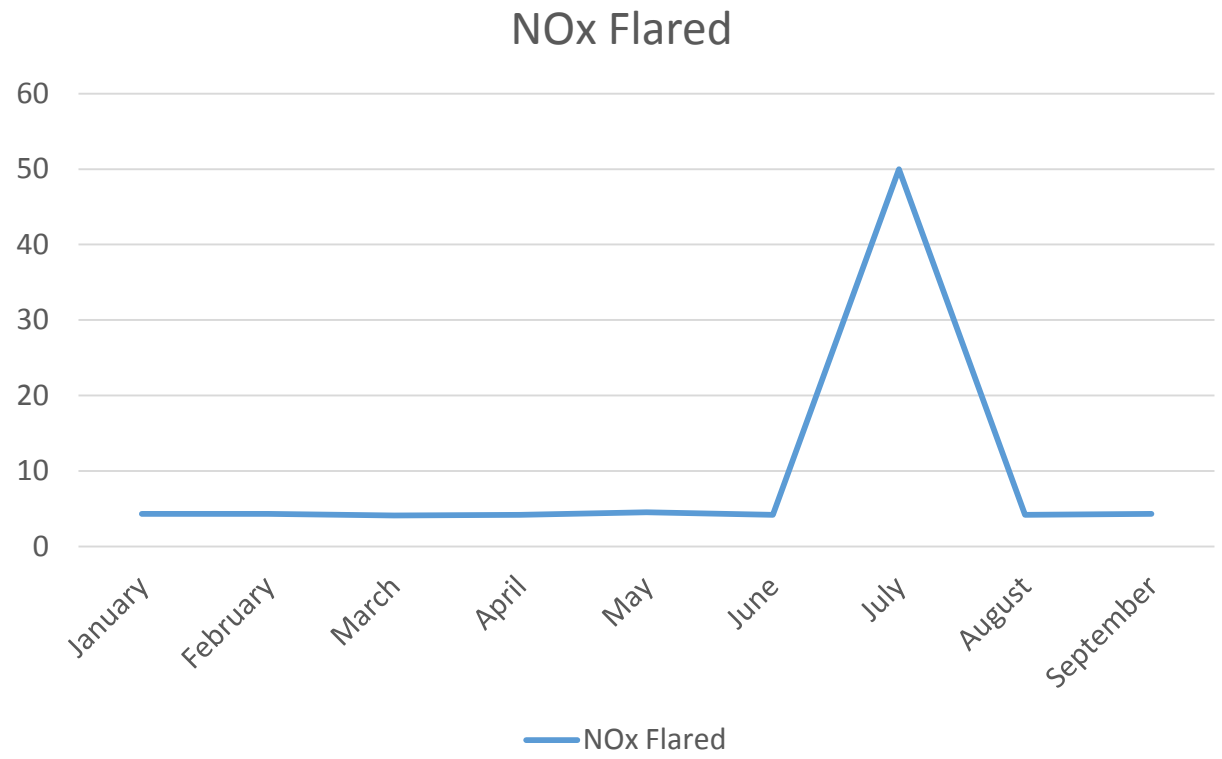
Typical Communications

- Historical Data
- Daily product totals
- New product orders
- Demand Calculations



So I've just owned a Historian…

- Windows Server Class Machine
- Cover up past poorly executed attacks
- Destroy a company's Health/Safety record
- Modify view of plant state to corporation
- Pivot to juicier targets

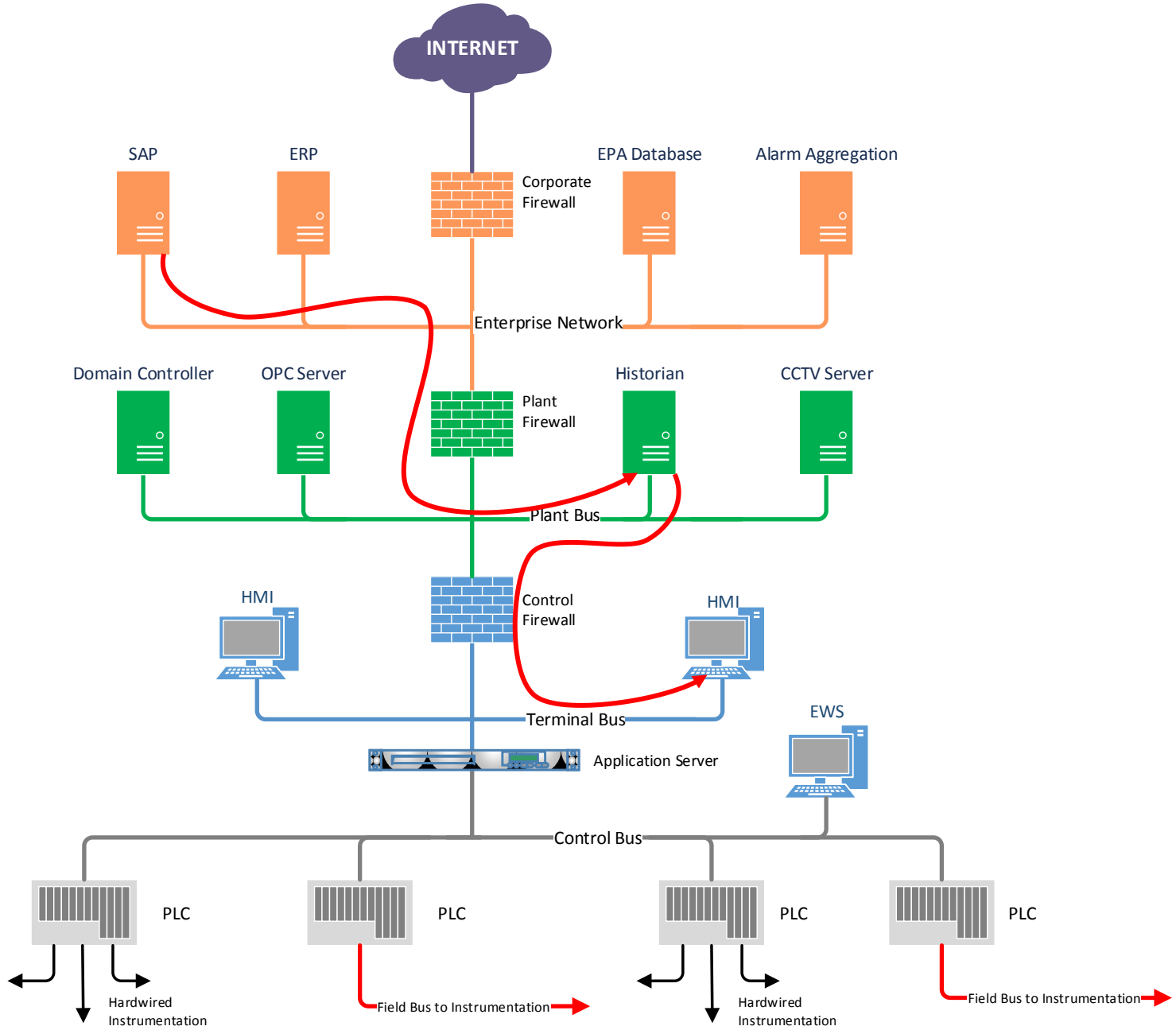


Attack Vectors

It’s all about the pivot

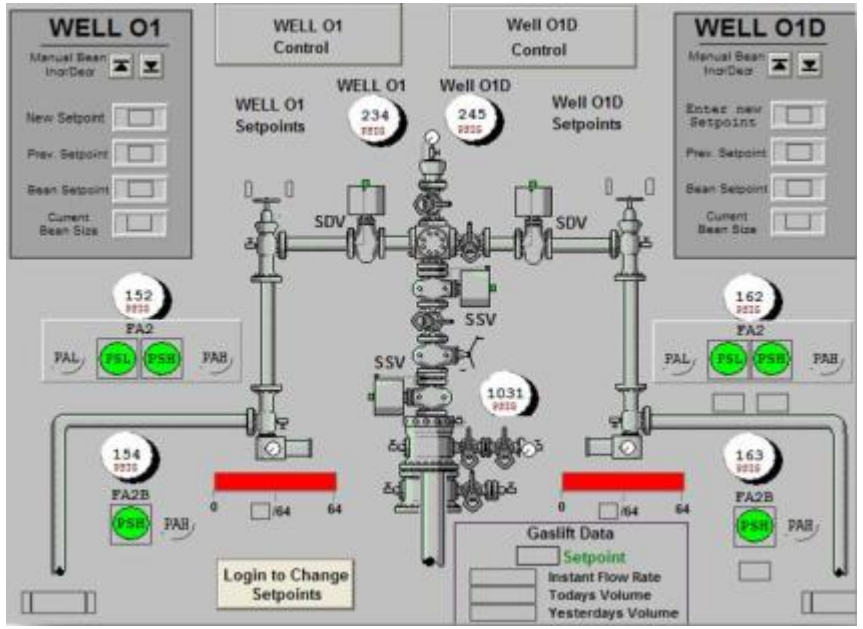
Typical Communications

- Historical Data
- Network Statistics
- OPC
- Domain Services



So I’ve just owned an HMI

- Windows Workstation Class Machine
- Write setpoints
- Spoof operator’s view of process
- CAUTION
  - Each HMI spoof process must be synchronized
  - Required IPC of some kind
- Pivot to juicier targets

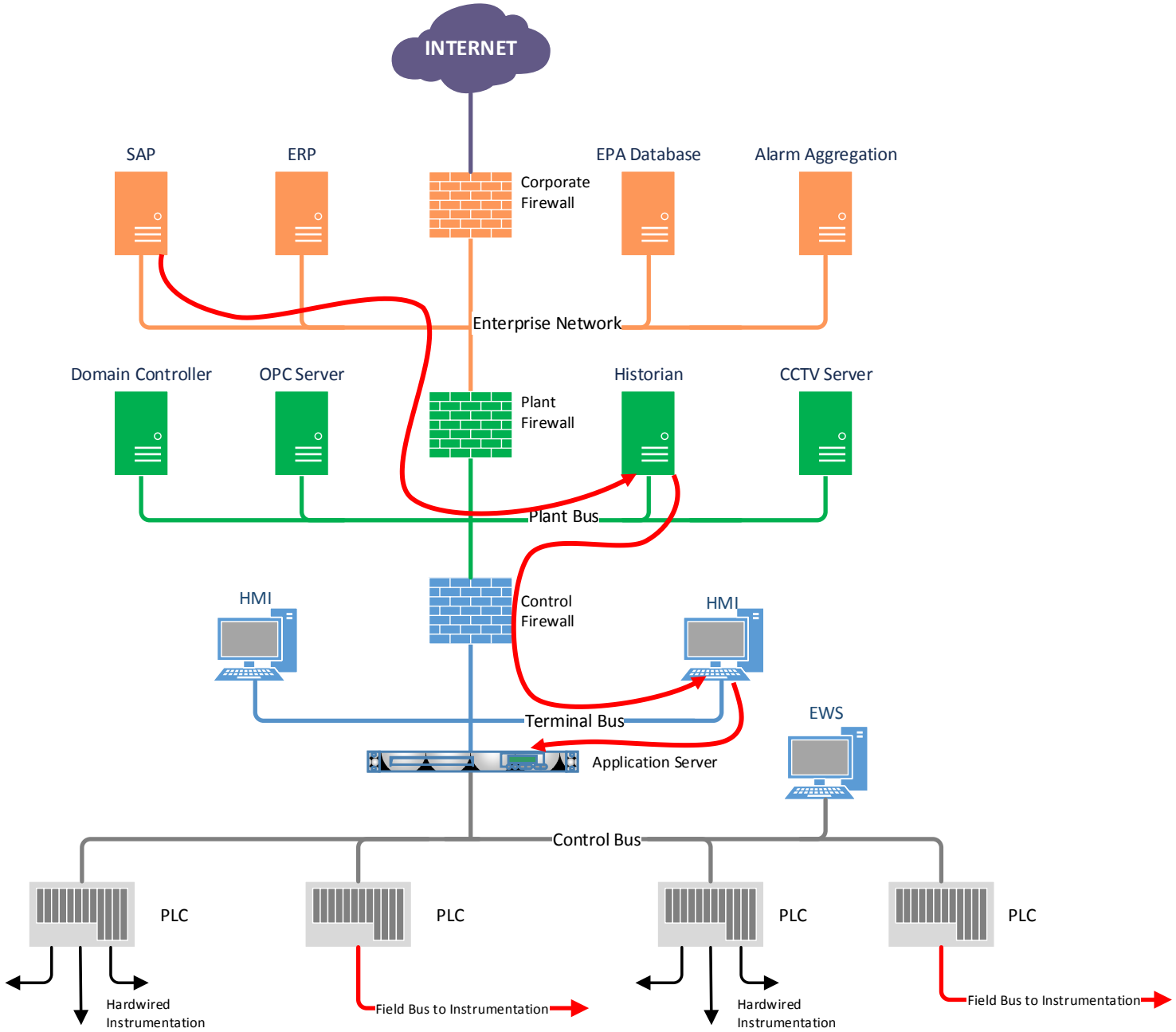


Attack Vectors

It’s all about the pivot

Typical Communications

- Setpoint writes
- I/O Value Reads
- Alarm Notifications
- Control Bus Diagnostics



So I've just owned an Application Server

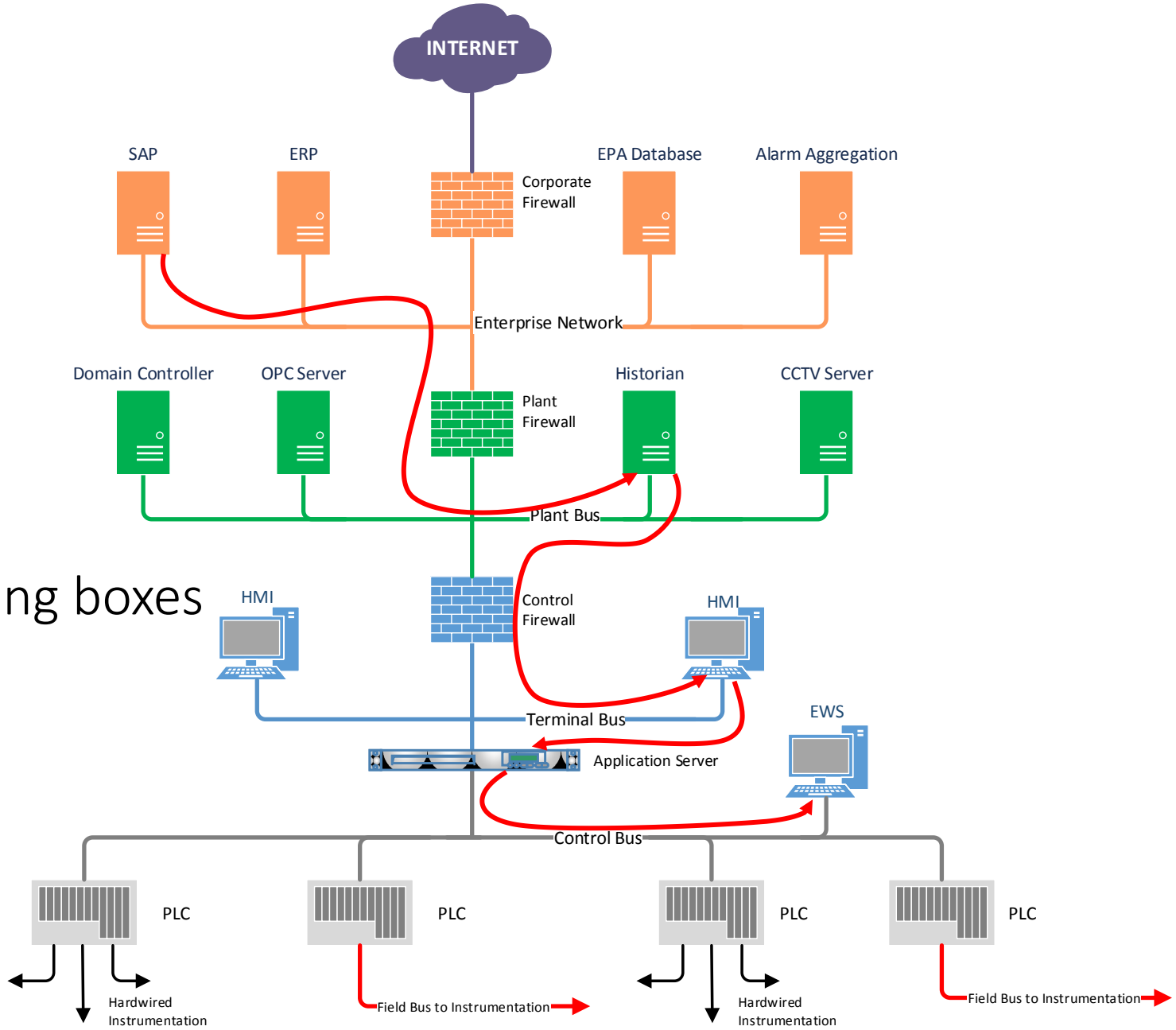
- Windows Server Class Machine
- Spoof view of process to all downstream components
  - Real-time and Historical Data
- No synchronization across components necessary
  - Simply modify the backend database values
- Pivot to Juicier targets

Attack Vectors

It’s all about the pivot

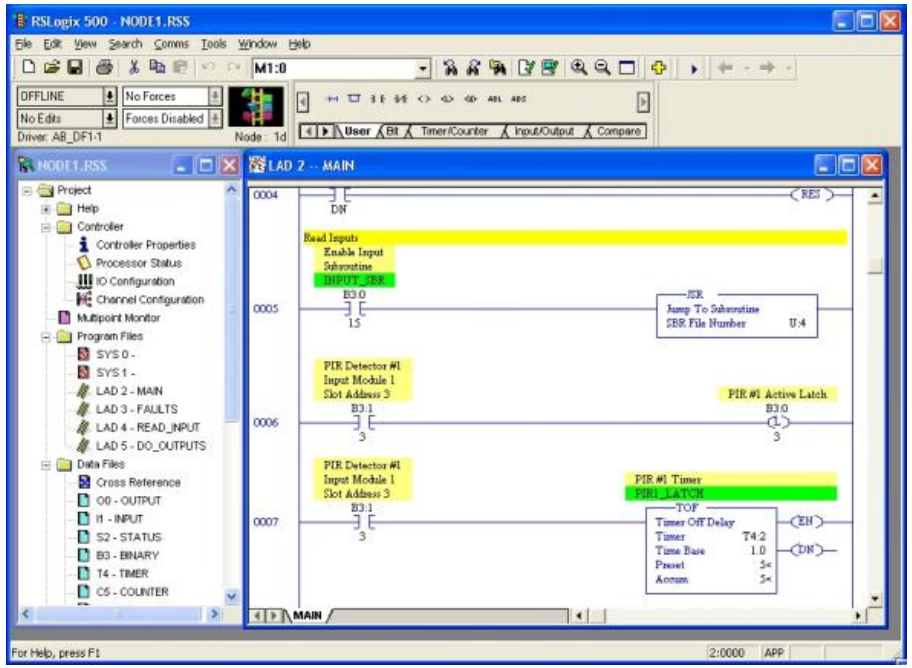
Typical Communications

- Doesn’t matter!
- The benefits of dual homing boxes



So I’ve just owned an Engineering Workstation

- Windows Server/Workstation Class Machine
- Modify actual logic of controllers
- Download online updates to controllers
  - Does not take down process but can subtly change it
- Remove engineered safety logic
- Steal PLC source code
- Pivot to embedded hardware

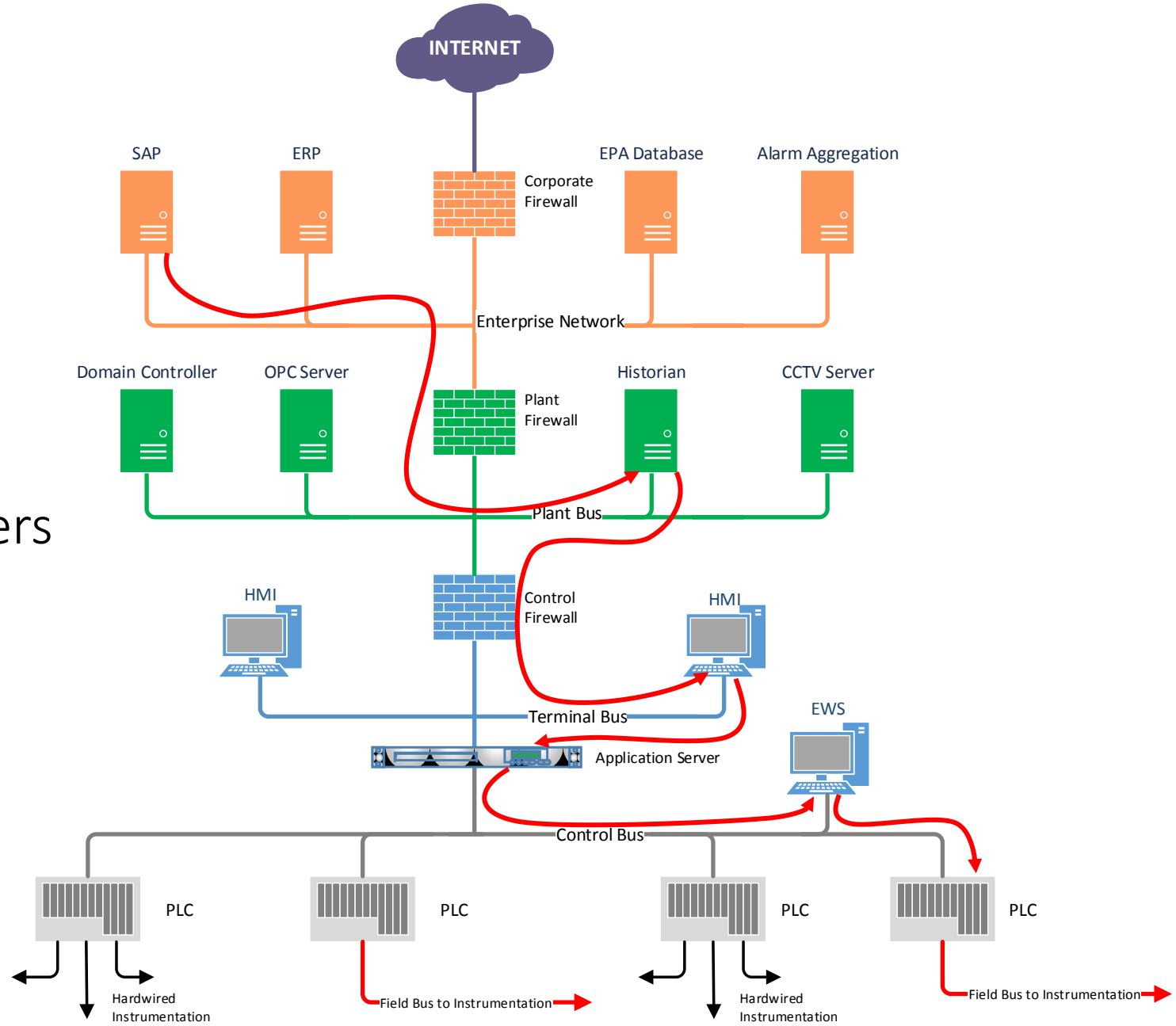


Attack Vectors

It’s all about the pivot

Ownage Imminent

- Logic updates to controllers
- Reconfigure network
- Reconfigure controllers
- Full visibility into process



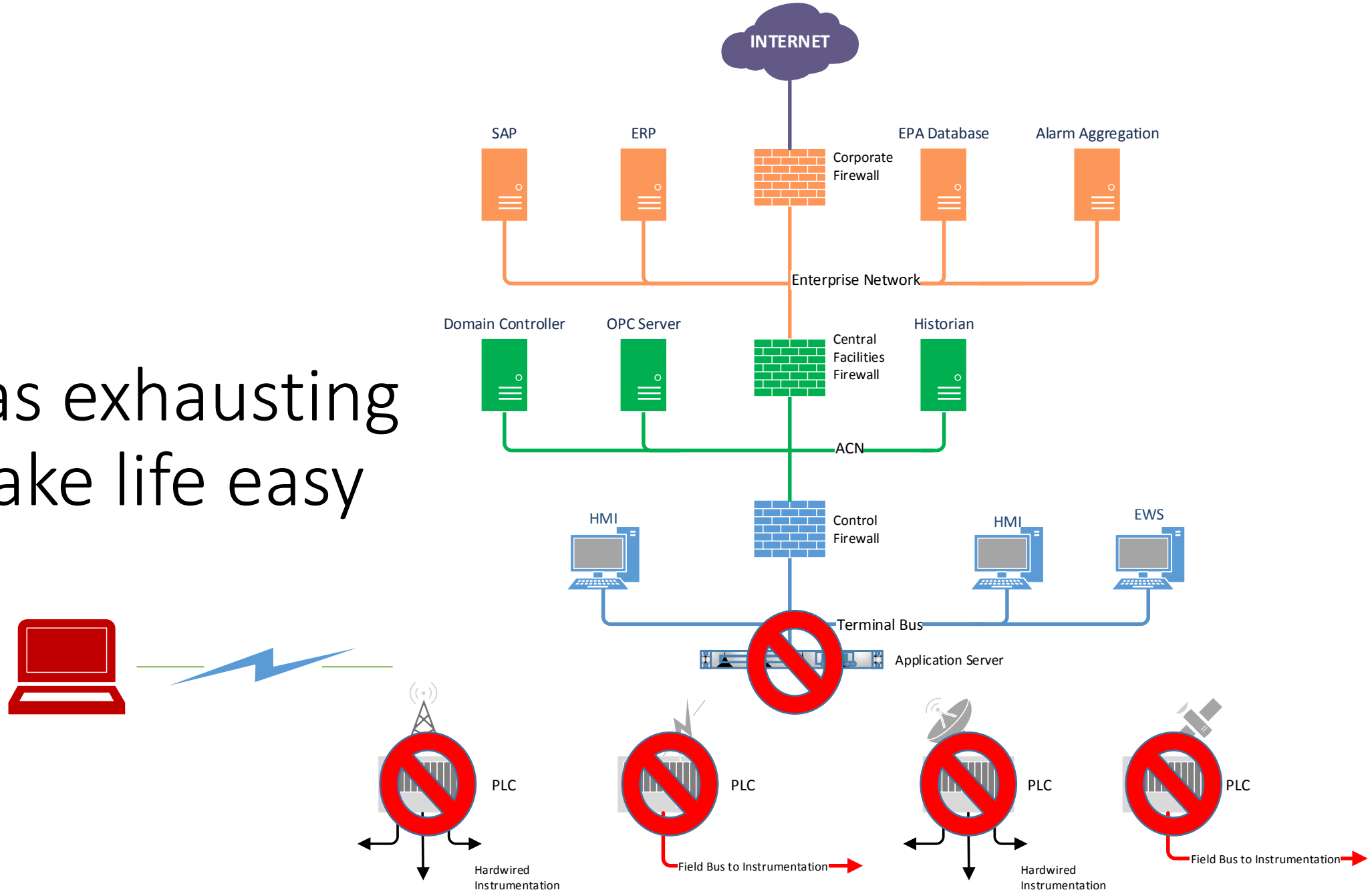
So I’ve just destroyed the process…

- Embedded Hardware
  - VxWorks
  - Linux (Usually BusyBox)
  - Old crusty RTOS
- Modify logic while online
- Write arbitrary memory
  - Input Table
  - Output Table

You'd think it would take a Nation State

- It doesn't!
- Most Windows based machines are woefully out of date
  - Vendors must approve all patches
  - Any change in a system requires a complicated MOC processes
  - If it ain't broke, don't fix it mentality
- Many Controllers are laughably insecure
  - Designed for availability and ease of troubleshooting

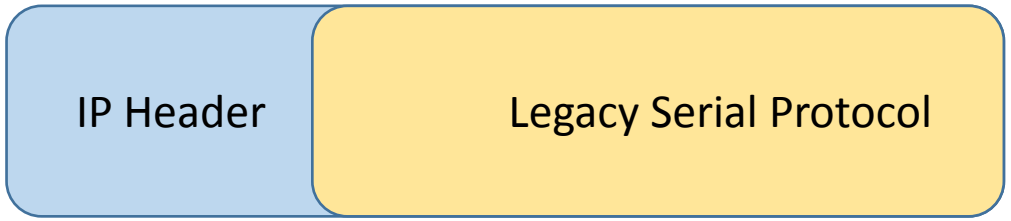
That was exhausting
Let’s make life easy



But they seem patched and I have no 0-day

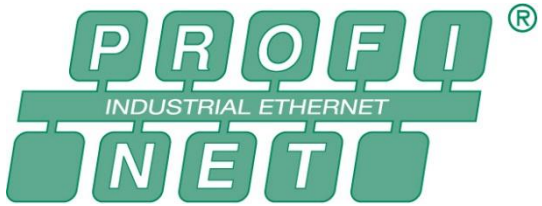
- Industrial protocols
  - Not encrypted by design
  - Hardware too weak to support encryption
  - Many are little more than encapsulated serial frames

- Controllers now have many services
  - FTP
  - HTTP
  - Debug
    - CVE-2005-3715, CVE-2005-3804, CVE-2006-0374

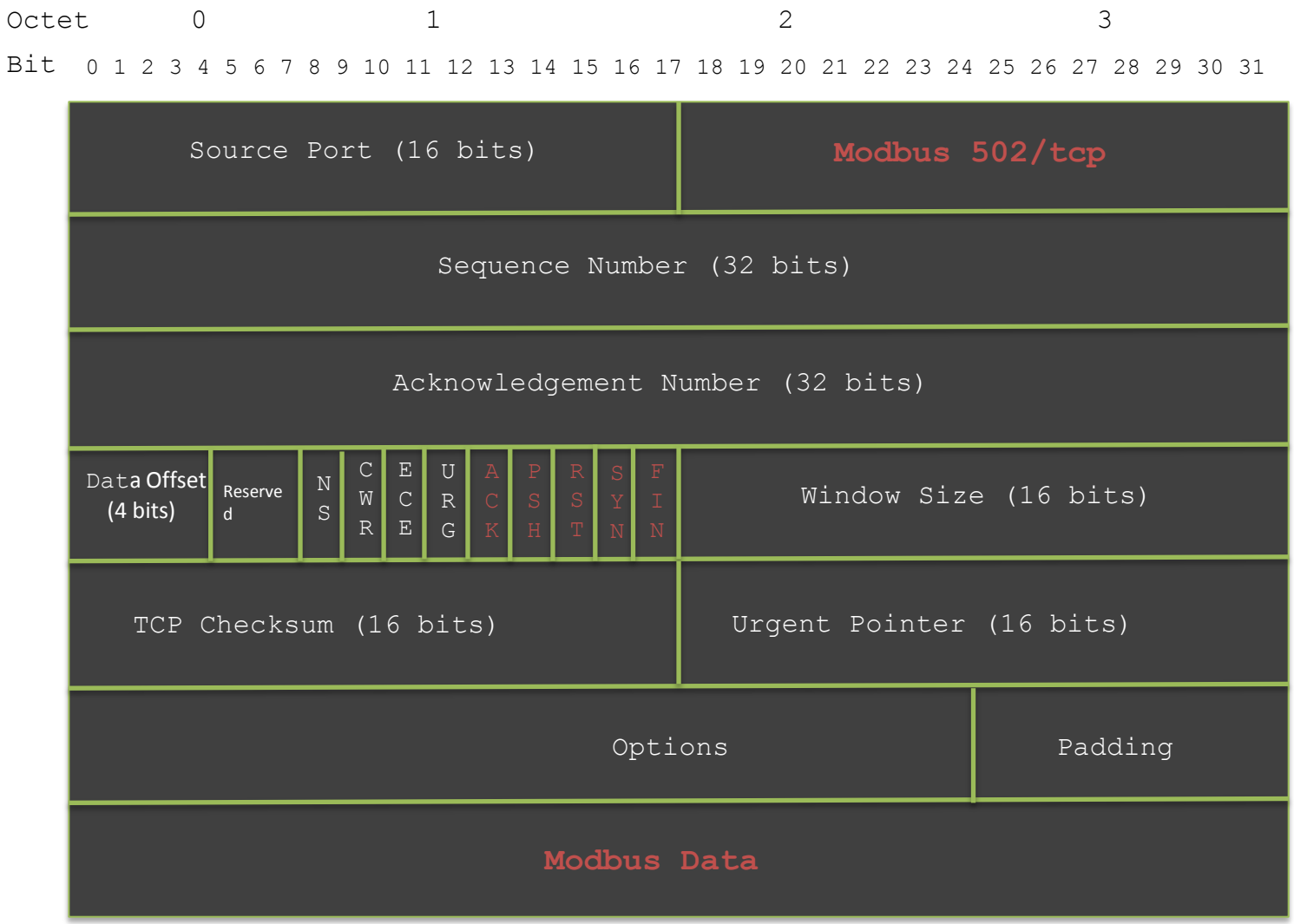


Industrial Protocols

- Modbus/TCP
  - TCP port 502
  - Minor changes from Modbus/RTU developed in 1970s
- Ethernet/IP
  - TCP port 44818
  - Similar to SNMP
  - Fully compatible with serial brethren
    - ControlNet, DeviceNet
- IP encapsulated serial
- Encryption never a concern



At the end of the day, it's all just bits



At the end of the day, it’s all just bits

A look at Modbus over TCP

Transaction ID		Protocol ID	Data Length	Unit ID	Function Code	Data
MSB	LSB	0x00 0x00	# of bytes	0xFF (Typ)	See Table	DATA

Function Code	Function
0x01	Read Coil
0x02	Read Discrete Input
0x03	Read Holding Register
0x04	Read Input Register
0x05	Write Single Coil
0x06	Write Single Register

Just ask nicely

- How to turn on a pump

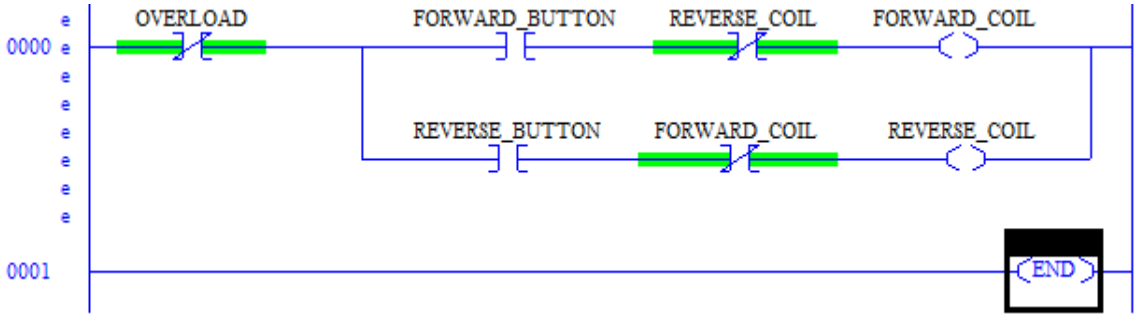
Trans ID		Protocol ID	Data Length	Unit ID	Function Code	Data
0xFF	0xFF	0x00 0x00	0x00 0x06	0xFF (Typ)	0x05	0x00 0x00 0xFF 0x00

Data Byte	Meaning
1: 0x00	MSB of Reference Number
2: 0x00	LSB of Reference Number
3: 0xFF	ON (0x00 for OFF)
4: 0x00	Always zero

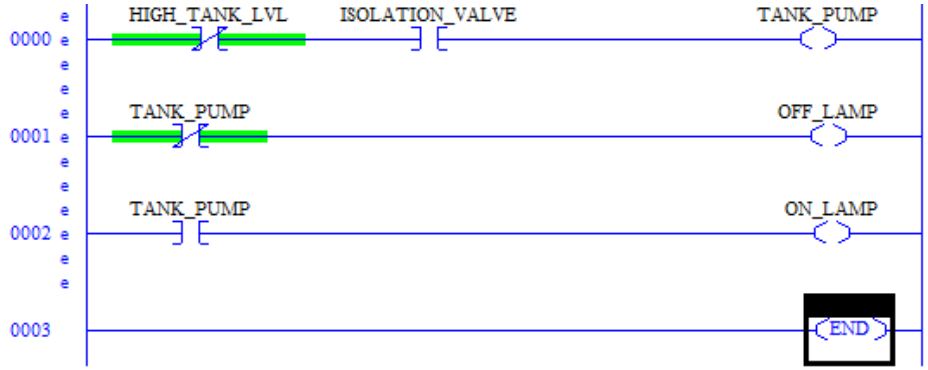
CAUTION: The reference number mapping to the pump must be known
Nuke Button Approach: Just write every output ON

Complications

- Control Engineers implement safety logic
- Interlocks

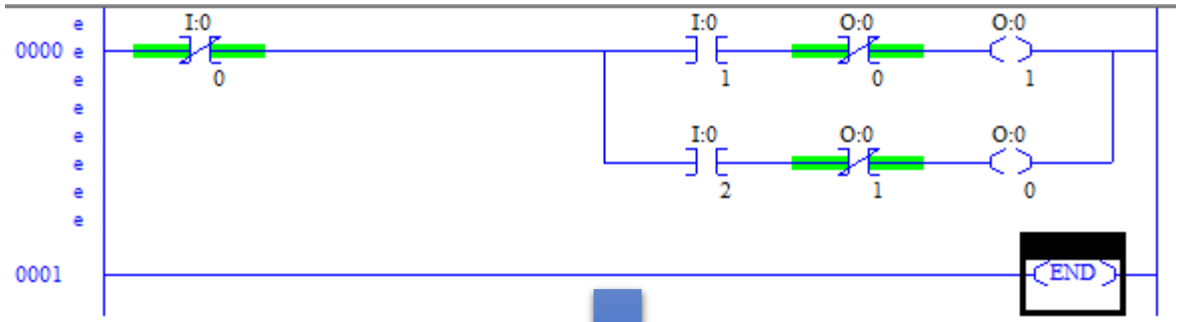


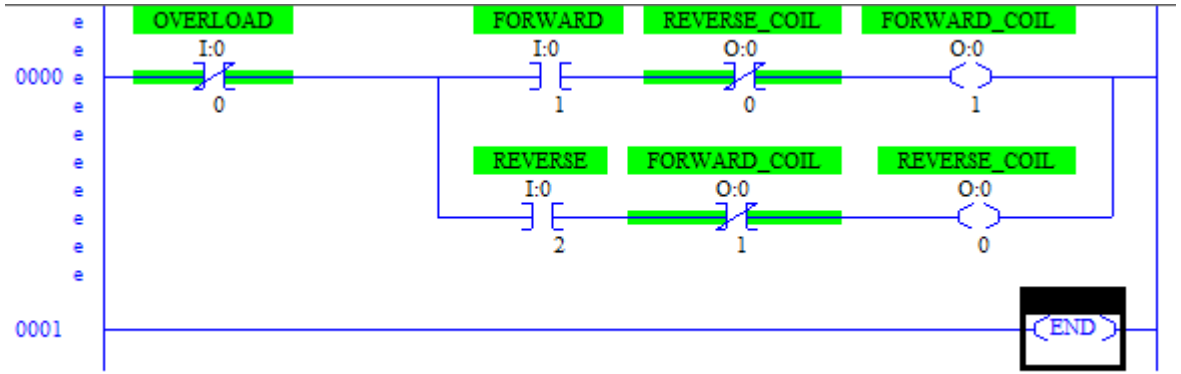
- Permissives

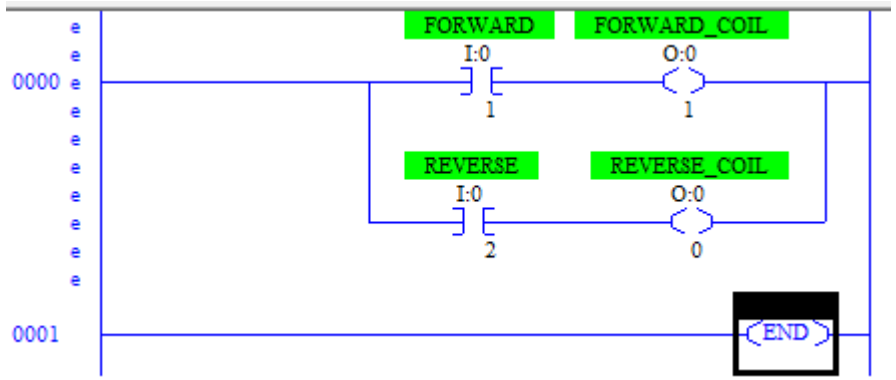


Solutions

- Remove safety logic!
  - Modify logic on the fly
  - Required feature of most Controllers
- Updates can be made from EWS
- Logic source is also on EWS

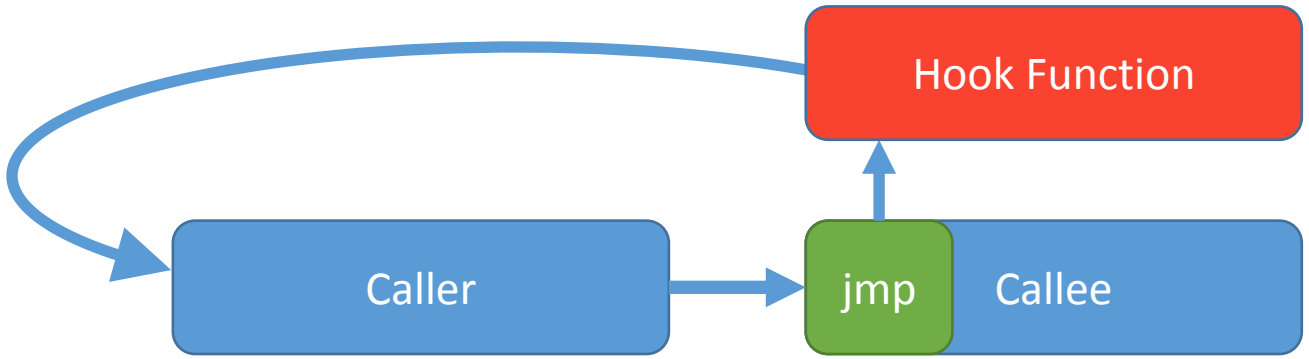






Spoofing

- Windows
  - Send/Recv hooking
  - Spoof process must be synced across all HMIs in peer to peer
    - IPC of some sort required
  - Preferable to hook on the Application Server
    - All data from the controllers funnels through it
- Non-spoofed external laptop can still see actual process

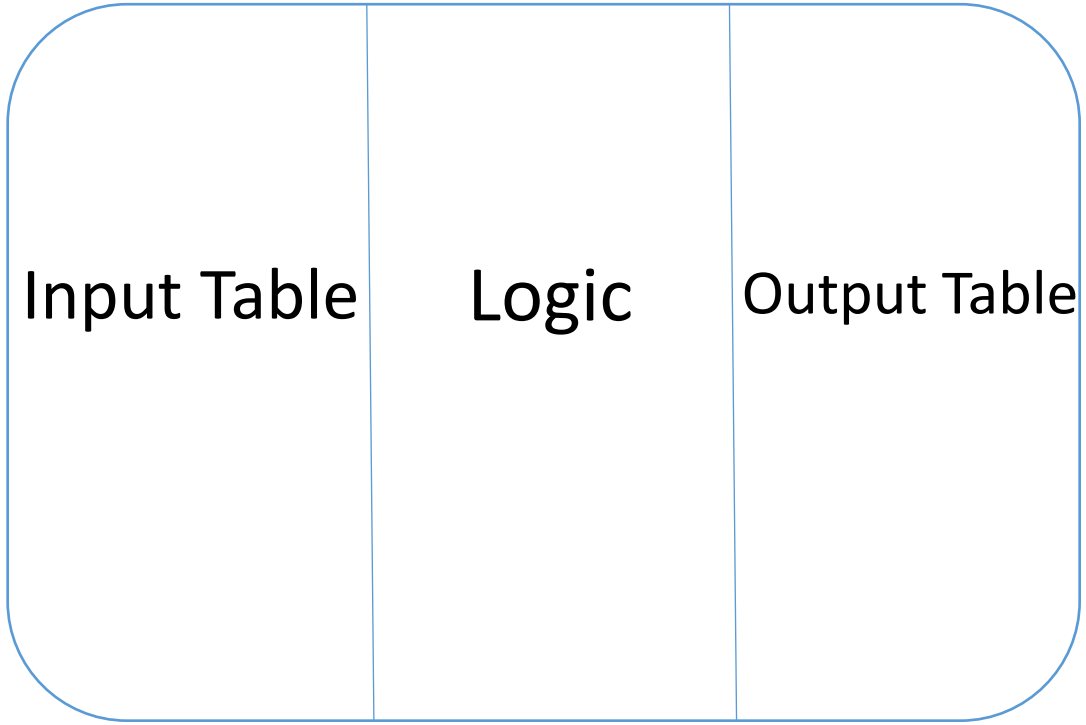


Spoofing

- Embedded Hardware hooking
- Ethernet Module send/recv hooking
- Man-in-the-middle at embedded level
- Ethernet Module internals are usually:
  - Old stripped down/broken Linux TCP stack
  - Custom made TCP stack

Direct Memory Access

- Some vendors leave debug processes listening for troubleshooting
- Direct access to physical memory
- Controllers have very specific (and interesting) memory regions



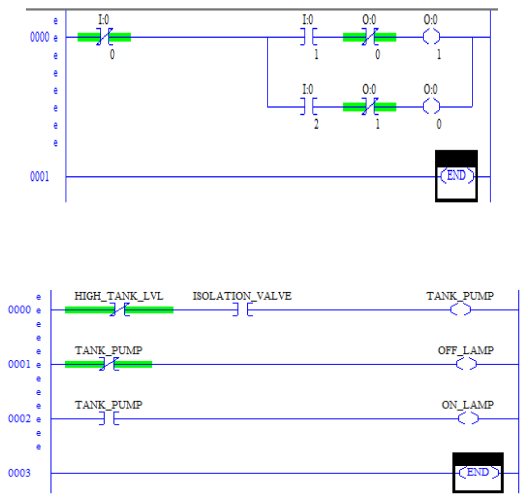
I’m a PLC and I’m OK

Scan Inputs → Perform Logic → Write Outputs

Input Table

00111010111110110101
11011010101010101010
10101010111110101010
00000101011111101011
00001111010101011011
01101010101010010001
01011110101001010100
00010101000010010000
01010100101111111100
00101010010100110100
10101001010101011010
11001010101010000000

Logic



Output Table

00111010111110110101
11011010101010101010
10101010111110101010
00000101011111101011
00001111010101011011
01101010101010010001
01011110101001010100
00010101000010010000
01010100101111111100
00101010010100110100
10101001010101011010
11001010101010000000

The world is your oyster

Controller's view of the world

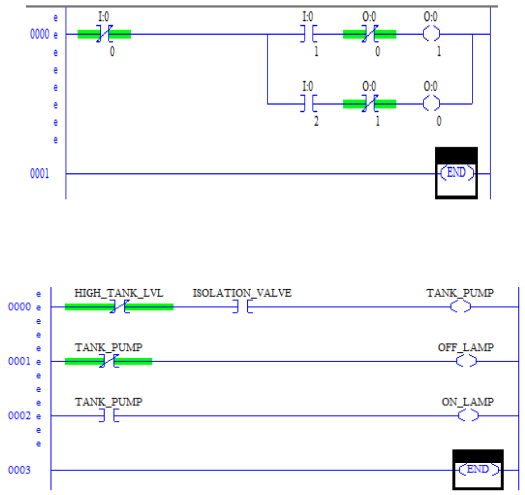
Writes Outputs based on logic

Controller acting on the world

Input Table

00111010111110110101
11011010101010101010
10101010111110101010
00000101011111101011
00001111010101011011
01101010101010010001
01011110101001010100
00010101000010010000
01010100101111111100
00101010010100110100
10101001010101011010
11001010101010000000

Logic



Output Table

00111010111110110101
11011010101010101010
10101010111110101010
00000101011111101011
00001111010101011011
01101010101010010001
01011110101001010100
00010101000010010000
01010100101111111100
00101010010100110100
10101001010101011010
11001010101010000000

Remediation

Network Level

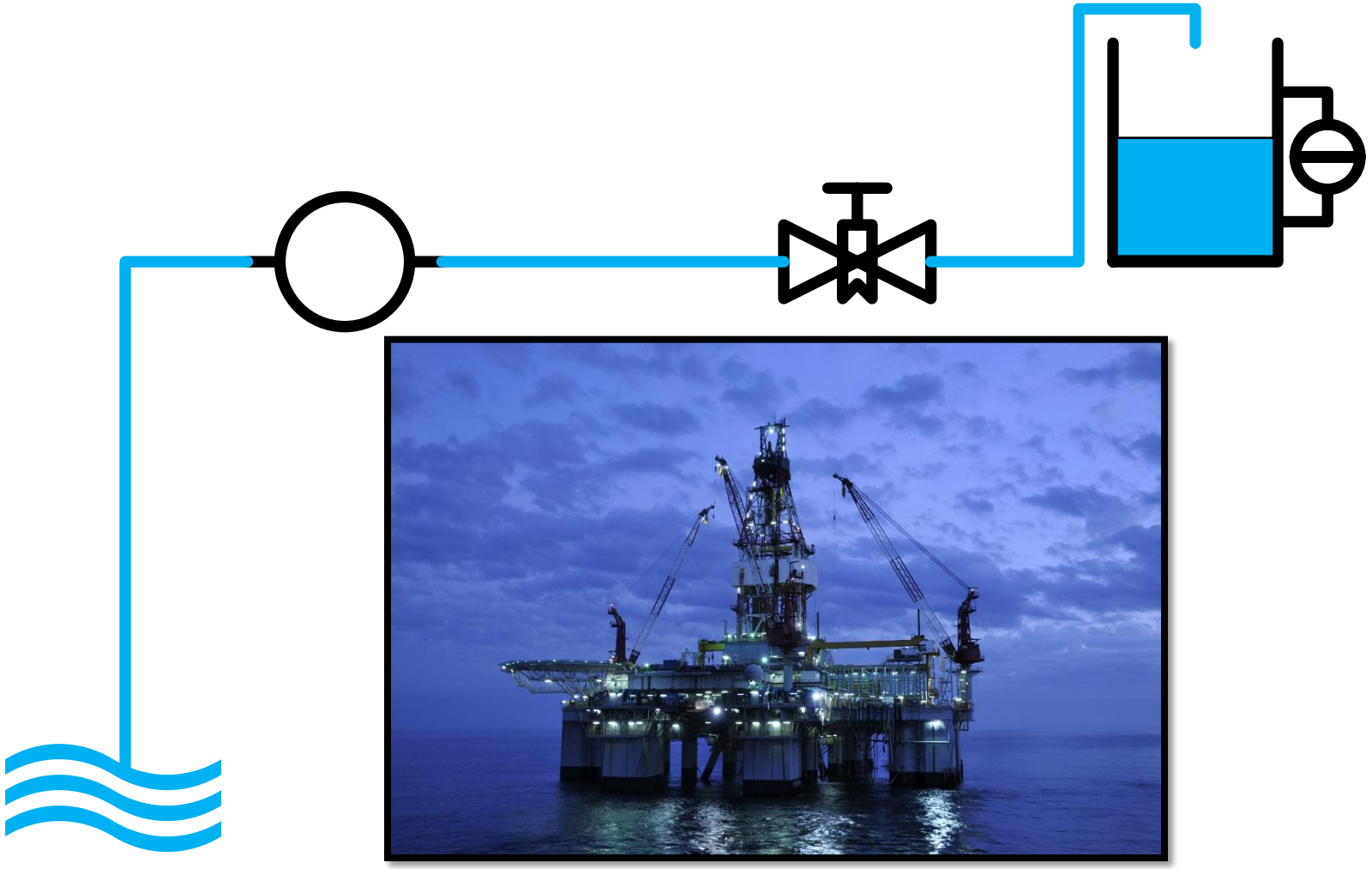
Remediation

Host Level

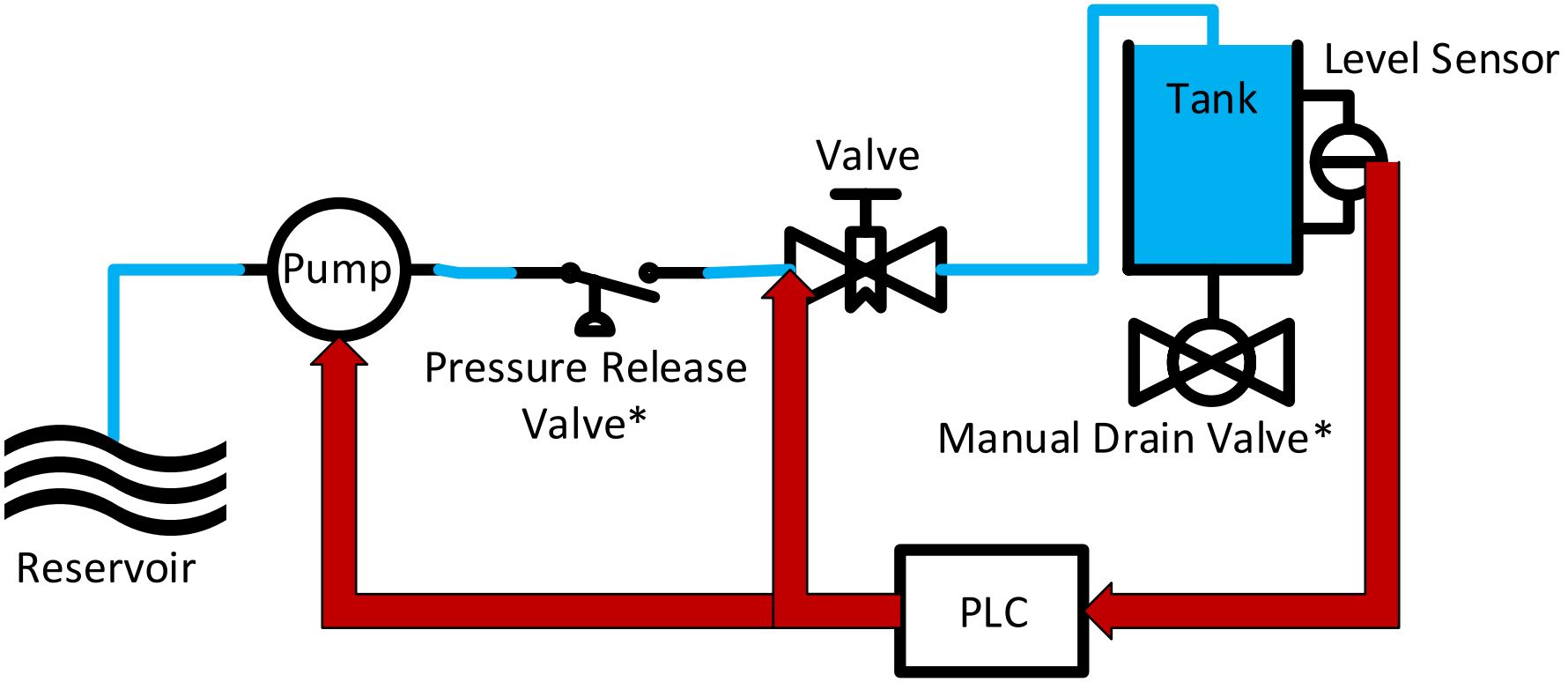
Remediation

Device Level

Exploit Demo



Exploit Demo



*Only present for demo purposes

Thanks

Cimation

Dillon Beresford - Research

Ryan Moos - Fabrication


Cimation®
Innovation Beyond Automation