Out of Control: SCADA Device Exploitation

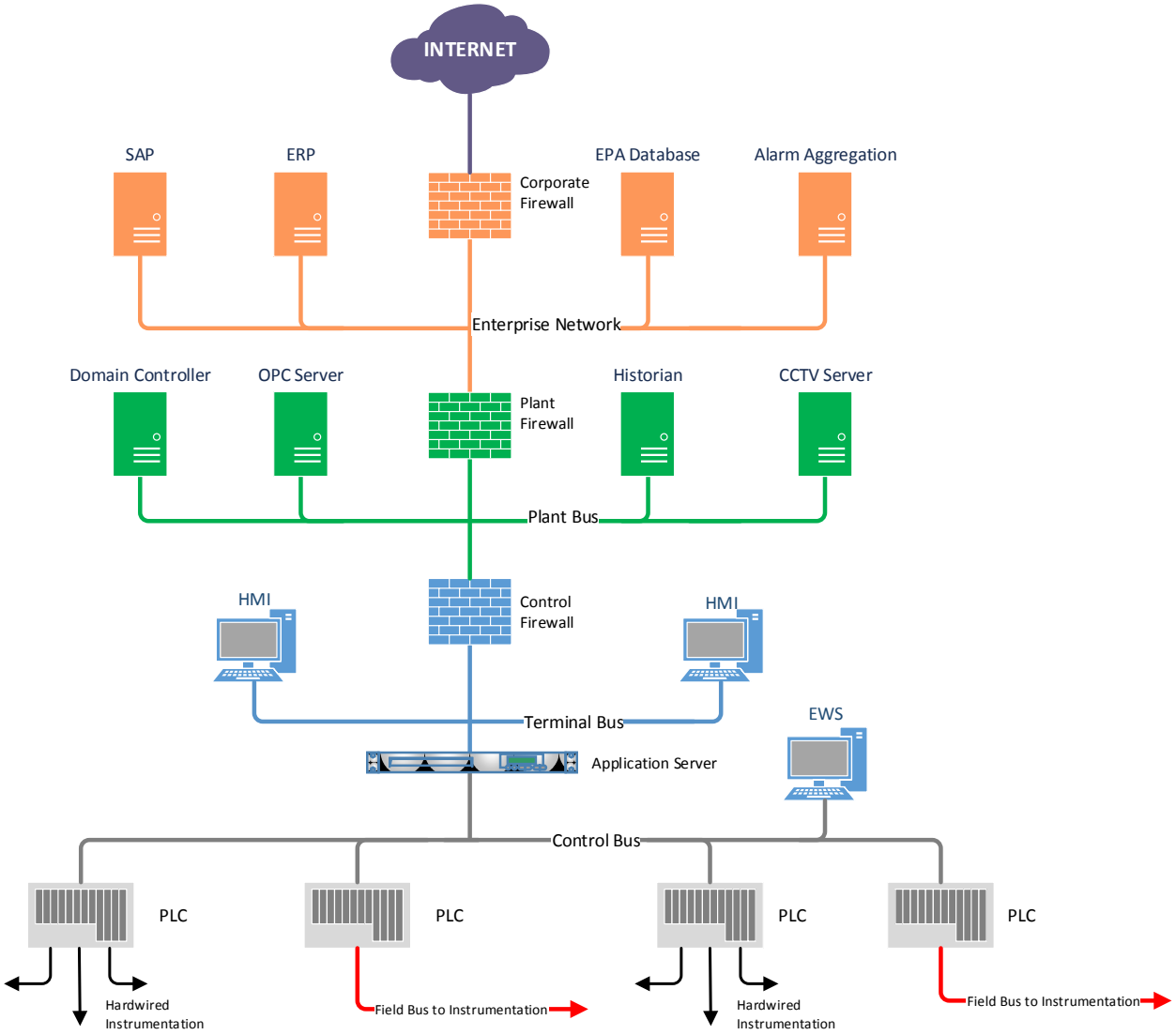
Contents

SCADA vs. DCS

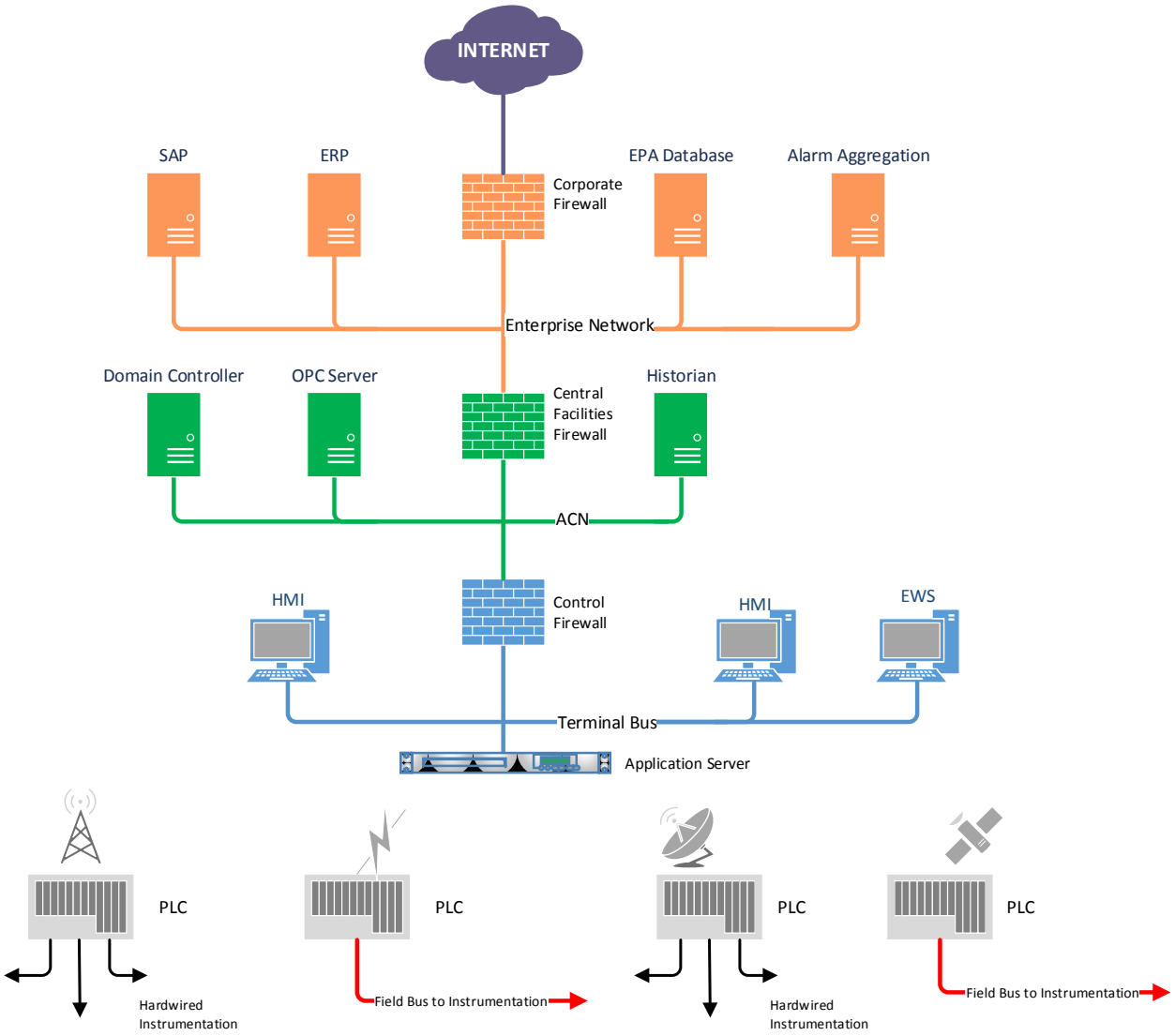
SCADA systems are traditionally used when real time control is not required. Devices are usually spread out over a large geographic region, such as a pipeline, and require remote access in order to operate. Distributed

Control Systems (DCS) are commonly used in real time control applications such as factories and assembly lines where all the components are in one location.

Network Architecture



Standard DCS Architecture Figure 1



Standard SCADA Architecture Figure 2

Standard architecture of a SCADA network is comprised of several compartmentalized levels intended to isolate critical systems from each other as well as more harmful traffic usually found on the business network segment. SCADA networks can use many different means of communication between remote field devices and the control network.

Problems arise when the standard architecture is not implemented properly. The most common architecture issues are not implementing firewalls between the control network and the enterprise, or having firewall rules that allow all traffic to pass, effectively turning the firewalls into routers. Another scary, but still common architecture issue is no network segmentation at all where all devices are allowed to communicate and no firewalls are in place.

SCADA devices were originally designed to be placed on air gapped networks

Components

Historian

A Historian captures field data for viewing by the enterprise. It holds a record of everything collected from the field such as flow data, amount of product produced, or even safety records.

HMI – Human Machine Interface

Until recently, most HMIs were knobs and buttons in a control room. Modern HMIs are usually Windows workstation class machines running a display of the process being controlled. Other common HMIs are a panel display with a keypad that displays values and alarms for an operator. Typically HMIs will allow the set points of a process to be changed, but nothing else.

Application Server

An application server is a server class machine that aggregates data from PLCs and RTUs on the control network and speaks

EWS – Engineering Workstation

An engineering workstation provides engineers with the ability to manage and control PLCs and RTUs on the network, also usually in the form of a workstation class Windows machine. All the capabilities of an HMI are present in a EWS as well as the ability to design and update controller logic.

PLC – Programmable Logic Controller

A typical PLC has a power supply, network communications (serial or Ethernet), inputs, and outputs usually contained in separate modules that can be expanded. The PLC operates by scanning inputs, performing logic on those inputs and writing the result of the logic to an output. Inputs to the PLC are usually sensors on the control network. The output is connected to the physical elements of the control network such as motors, pumps, or valves.

The computational architecture of a PLC is quite basic; processers are not high speed and do not use advanced capabilities. Notably, memory management is absent. All processes running on a PLC have direct memory access.

RTU – Remote Terminal Unit

An RTU is very similar to a PLC except that it may have more advanced abilities such as interrupt based logic.

Instrumentation

Instrumentation on control networks usually consists of sensor networks like flow meters, temperature and pressure gauges. These devices often utilize wireless protocols to report their data back to the control network.

Attack Scenarios

Pivoting

Pivoting is a standard technique used in penetration testing to navigate from machine to machine. As the Enterprise Network has the most visibility to the Internet, it is the most logical starting point to begin penetrating the network down to the process control domain.

HMI

As the HMI is the eyes of the process, exploiting this machine can cause a process to run blind, i.e. the operations staff have no idea how the plant is actually operating. The HMI can also write setpoints to the controller, which can map directly to outputs on the controller or to scratch memory to be used in calculations by the logic solver. A compromised HMI can destroy a facility by putting certain safeties in bypass and writing bogus setpoints to the controller. HMIs can also acknowledge alarms, which are critical in identifying and mitigating critical problems with the process.

Application Server

In some architectures the application server acts as a buffer between the sensitive control bus and the HMIs. As PLCs are not meant to withstand constant traffic from a large quantity of data consumers, an Application Server polls each PLC for critical data and stores it in a database for HMI retrieval. This way, the control bus experiences minimal and more deterministic traffic to keep the process running smoothly. Application servers are also single points of failure, and a compromise can alter the view of the process to all terminals that query it for data. Instead of exploiting each HMI in a refinery, it is more fruitful to exploit the application server and modify operation's view of the process to every HMI.

EWS

Engineering Workstations are the central repositories of logic held by the PLCs. They are used by engineering staff to diagnose and modify communications and logic on the control network. They have the necessary tools to download new code to the controllers and can be used as a last resort HMI during a failure of the primary machines. Exploitation of an EWS is especially dangerous, because they have the tools to completely modify the contents of a controller, as well as force Inputs/Outputs. Engineering workstations can also hold the 'source' for the logic, giving valuable information to an attacker that is attempting to map the process.

PLC

As the PLC is the main piece of hardware that directly control the process, exploitation of this component can be catastrophic. The PLC is usually the most protected part of the control network, not necessarily because of security, but because they can be so fragile and must be operational for the process to continue. Even something as simple as a DOS attack on a PLC can bring the entire process to a halt, or worse.

Common Vulnerabilities

Industrial Protocols

Industrial protocols were designed to be lightweight and provide reasonably deterministic communications between controllers and terminals. Security was implemented as an afterthought or, perhaps, not at all. Simple serial protocols such as Modbus and DNP3 can and have been wrapped in IP datagrams and sent over the Internet.

Significantly altering a process does not have to include component exploitation, it can be as easy as assembling a well formed IP packet of an industrial protocol and sending it to a controller. IP whitelisting and port security do not exist in standard PLCs, so it will take any well-formed IP packet and performed the requested actions on the actual process

Common Open Industrial Protocols examples are:

Modbus/TCP	*TCP Port 502*
Ethernet/IP	*TCP Port 44818*
Profinet	*TCP/UDP Port 34962 (Typ)*

Debug Service

At the device level, a debug service left enabled by the vendor typically allows full access to physical memory. An example was seen with VxWorks. These services are meant to aid developers and vendor troubleshooters in the case of a customer problem out in the field. Just as vendor support staff can access these services, so too can malicious entities. These debug services allow direct memory access, which for Controllers includes bit mappings to real world equipment. Arbitrarily modifying memory can have disastrous consequences. The three main memory areas involved with process control are the following:

Input Table: This table is populated by scanning the Input cards of the controller. Each input is mapped to a memory location in the input table. This area of memory gives the controller a view of how the actual process is operating.

Logic Memory: This area of memory hold the actual logic code. This code can either be compiled or interpreted, depending on the individual controller. Modifying this portion of memory can have extremely unpredictable consequences, just like modifying a binary.

Output Table: This table is populated by the logic as it solves each ladder as it goes through a scan cycle. Once the cycle is complete, the output table is 'pushed' to the Output cards where signals are sent to the physical equipment. Modifying this portion of memory directly impacts the process, as each bit in the table maps to instrumentation.

Remediation

Network Level

Network attacks can be remediated by removing public facing devices and utilizing a private network for remote devices. Proper network architecture is a key component of preventing unauthorized access to critical systems. Network flow control and security devices such as firewalls and Intrusion Detection Systems (IDS) allow administrators to set restrictive traffic rules on a network and obtain a historical view of normal traffic.

There are special concerns for wireless devices used to access remote systems. Frequency Hopping Spread Spectrum (FHSS) technology should be utilized when possible in wireless systems. Of course, network traffic should be encrypted whenever possible.

Host Level

Protecting hosts against attacks in SCADA systems is very similar to corporate environments, except additional considerations can be taken. Often, control systems are unable to be patched with the latest available security and bug fixes due to support issues and legacy systems. Due to the nature of control systems, the hosts are usually only performing a limited set of activities and are not used as general purpose machines. Therefore, control system hosts are a good candidate for whitelisting technology.

Antivirus products are important, if only to protect against older threats and rogue USB devices that may come into contact with hosts. However, as with other products on a control network, updates are a challenge because internet access is restricted.

Controller Level

Most of the attack remediation at the controller level is only applicable to device manufacturers. However, command level filtering such as device level firewalls are available that can filter which commands are available to be issued to devices such as a PLC.

Device manufacturers can reduce attack surface by removing unnecessary and insecure protocols from firmware images. Hard coded credentials such as service level accounts and secret keys should also be removed or stored securely. Firmware itself should be encrypted. The biggest thing vendors can do is test their own products for security issues and audit code.