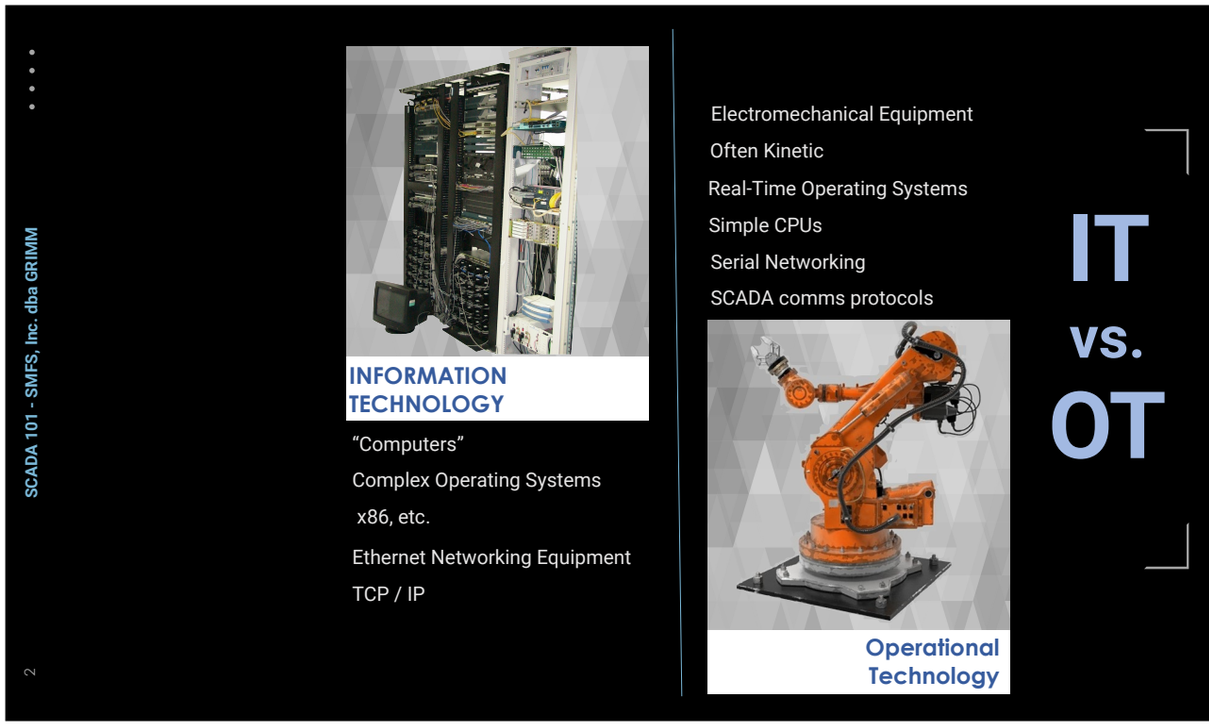
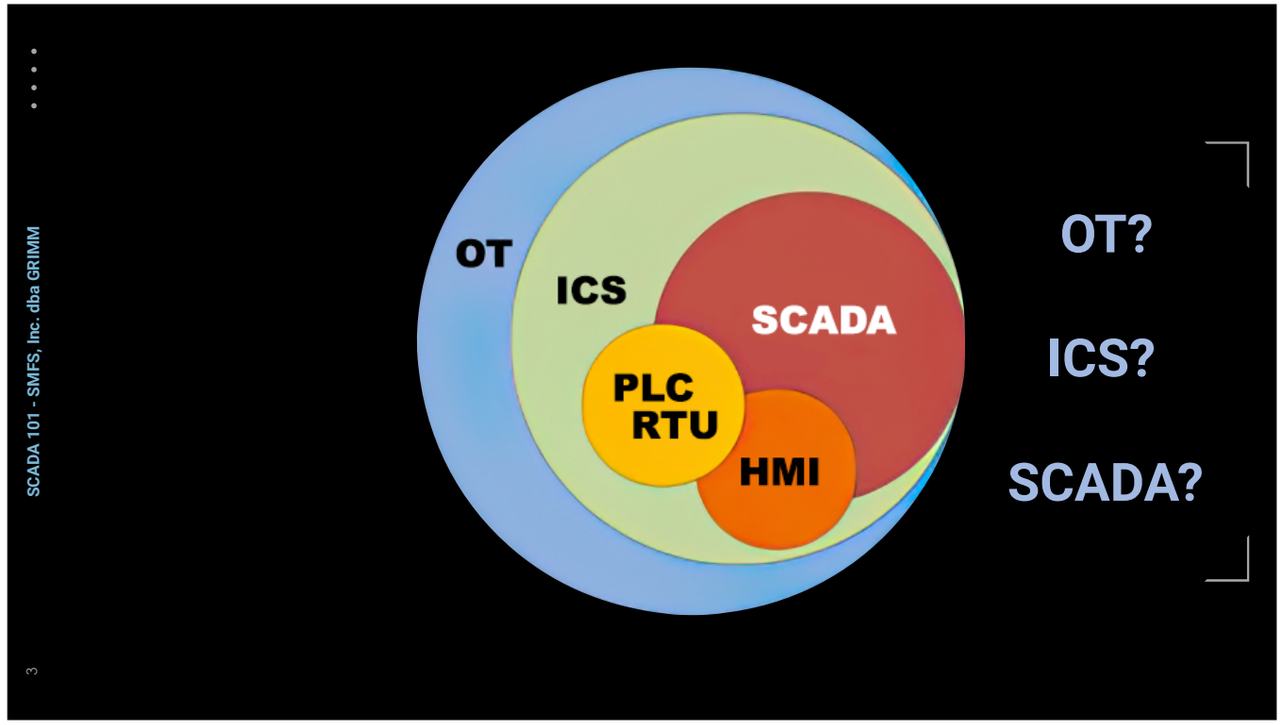
Hey everyone, I'm Johnny Christmas, one of the trainers here at GRIMM, and we've got a quick video for you today on SCADA! This is going to be for all the IT folks who are finding themselves moving towards converging their enterprise systems with Operational technology, and need to understand the basic terminology they may be coming across in general ICS conversation in order to better understand what happens on the other side of the wall! Let's jump right in . .



Now right away looking looking at this you may be thinking “OT? I thought we were going to be talking about SCADA! That’s a great observation, and a big elephant in the room that needs addressing. The acronyms OT, SCADA and ICS are often used interchangeably in conversation by OT professionals, and generally that’s fine but there are some basic distinctions you should be aware of.

First we’ll handle OT. Operational Technology refers to computing systems that are used to manage industrial operations as opposed to the more administrative operations traditionally handled by the Information Technology we’re all generally aware of. Operational systems include production line management, warehouse automation, mining operations control, oil & gas monitoring, and even the local systems inside aircraft, rail engines and even most personal vehicles! OT is often … (GO THROUGH OT BULLET LIST)

Now let’s focus a little more on clarifying those OT / ICS & SCADA lines (NEXT SLIDE)



Industrial control systems (ICS) is a major segment within the operational technology sector. It comprises systems that are used to monitor and control industrial processes. This could be mine site conveyor belts, oil refinery cracking towers, power consumption on electricity grids. ICSs are typically mission-critical applications with a high-availability requirement.

Most ICSs fall into either a continuous process control system, typically managed via programmable logic controllers (PLCs), remote terminal units (RTUs) and their associated control devices. These industrial control systems (ICS) are often managed via a Supervisory Control and Data Acquisition (SCADA) system that provides a graphical user interface for operators to easily manage the process under control; these consoles are often referred to as the aptly-named “Human-Machine Interface,” or HMI. We’ll talk about all of these devices very shortly, but for now, let’s talk about how THEY talk:

MODBUS (MODicon BUS)

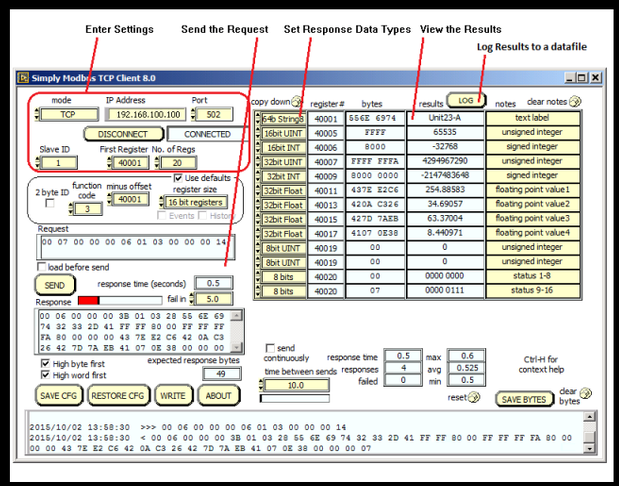
Master \ Slave

- Masters initiate communication and read/write values in slaves
- Slaves take action based on data in registers & coils, following pre-programmed instructions “Ladder Logic”

Registers \ Coils

Values stored in slave controllers

- Coils: On / Off
- Registers: Numerical



SCADA communications protocols number in into the hundreds, many of them closed and proprietary, and arguably the most common one is called “Modbus.” The Modbus protocol provides an industry standard method that devices can use for composing and parsing messages. This protocol was developed by Modicon, Incorporated, for industrial automation systems and Modicon programmable controllers, but has gained wide use across the SCADA realm due to it being an open standard.

Modbus devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device which looks up and/or processes information and sends its output to the master using Modbus. PLCs, heating elements and fluid level sensors are common slave devices, while a typical master device is a host computer running appropriate HMI software.

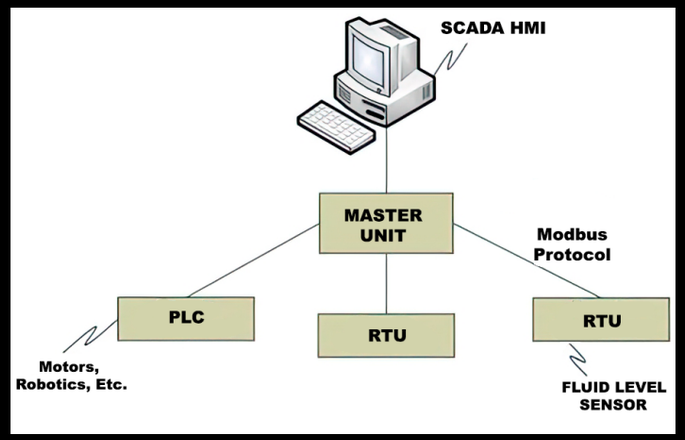
Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. Note that no instructions for activity are passed in this process; only values. Activity instructions are contained within the slave devices (often in the form of a basic form of logic called “Ladder Logic”), and the CPUs and operating systems in the devices determine when to carry out specific instructions based on the values we’re reading and writing as part of this modus communication process. This allows slave devices to operate autonomously, carrying out their various physical tasks, even when communication is severed.

For example, say we have a slave devices that is a robotic arm with logic programmed into to allow it to identify an incoming box on a conveyor belt, pick up the box, and place the box on a pallet, but only to carry out these instructions when a specific coil value is set to “1.” We use our HMI to tell the Master Modbus device to set this coil to 1, and the robot gets to work. If it never hears from us again, it will continue to carry out these tasks until it loses power, unless some part of its logic has a pre-set stop time. If it does, then the internal logic will set that coil to “0” and the robot will cease to function until our Master device sets the “1” coil again.

One very interesting thing to note, which you may have inferred from the image here is that Modbus can be encapsulated in TCP. This is true for many SCADA protocols, but not all of them. This is extremely handy as it allows the leveraging of ethernet-based networking equipment for communication between some (or sometimes even all) ICS devices.

Let’s take a look at how a network of ICS devices might look: (next slide)

SCADAS NETWORKS



What we're looking at in this basic network is:

The SCADA HMI unit that shows the process under management in a graphic display with status messages and alarms shown at the appropriate place on the screen. Operators can typically use the SCADA system to enter controls to observe the status of a system, receive any alarms indicating out-of-band operation, or to enter system adjustments to modify the operations in real-time. For instance, there might be a control to turn a valve off, or turn a thermostat down. In fact, the thermostat you likely have in your house is a bona-fide HMI, as many HMIs are standalone pieces of hardware only capable of handling a single system, often with just a few buttons and a small screen. They can also, as shown here, be standard PCs or virtual machines with custom HMI applications installed.

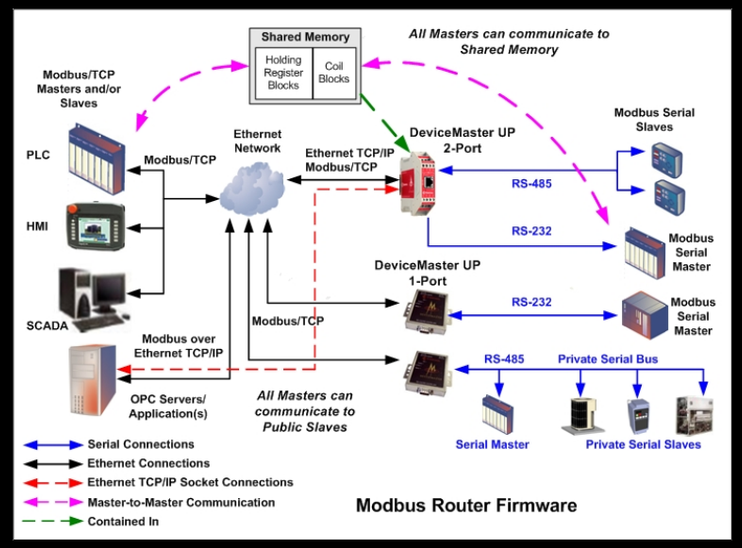
Next, the Master Control Unit that attaches PLCs and RTUs to the SCADA system. Programmable Logic Controllers, as the name implies, are very basic processing units that can be preprogrammed to carry out some very basic logic based on certain conditions. Due to the extremely basic real-time operating systems running on these basic controllers, the Master Control unit must pass data to and from the SCADA system in real-time with low latency. If a message arrives out of turn or too late, or is a message the target device does not know how to handle, then unexpected, and sometimes even destructive behavior can result. This is a primary reason as to why active IT tools which transmit across a network, such as nmap, should never be used in a SCADA environment. The risk is VERY real.

RTUs are similar to PLCs, but are more robust, in both environmental tolerances, as well as capability. They can often even run compiled code from common languages such as C# or Visual Basic, and be fitted with modular circuitboards to meet the specific communication needs of the devices they control, and the ICS environment. They can even support local logging and TCP/IP which makes them excellent for environments where stable communications may be difficult.

Communication links can in fact be Ethernet for a production system, but basic serial lines are also very common. There are even ICS-specific radio

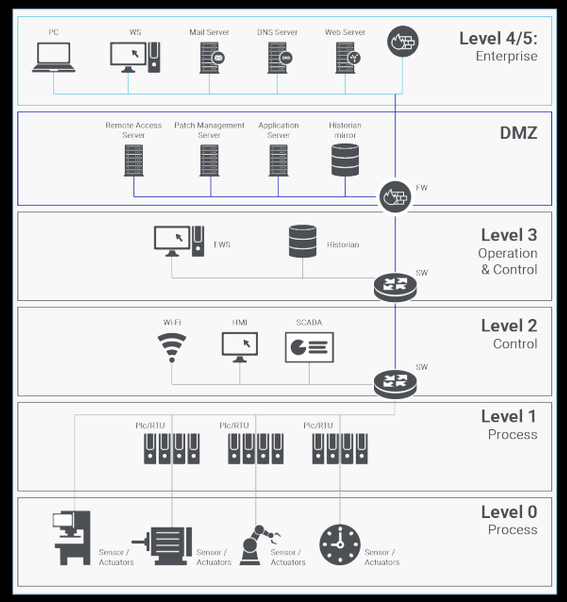
protocols for a wirelessly distributed operation or just a telemetry link for equipment in a remote area without communications facilities. (NEXT SLIDE)

Communication Methods



If we take a look at this diagram we can see there are quite a number of possible communication media that can be in use at any given time. You're likely very familiar with Ethernet, but be aware that the common serial protocols of RS-232 and RS-485 are still extremely popular for shorter runs in an OT environment. We don't need to pick through everything that's in this slide, as a quick skimming of it is all that's necessary to give you the basic understanding of the layout potential for a common SCADA environment. You can see here how modbus communicates in it's master-slave manner through HMIs and PLCs, occasionally encapsulates inside TCP if needed, and even communicates across various serial lines.

Now, let's see how this all works when combined with an IT environment while still managing to keep IT and OT safely separated.



THE PURDUE MODEL

In the 1990s, Theodore J. Williams, along with members of the Purdue University Consortium for computer integrated manufacturing, developed the Purdue Enterprise Reference Architecture (PERA) as a model for enterprise architectures. The Purdue model does an excellent job of defining the different levels of critical infrastructure that are used in production lines and the way to secure them. Here is a quick overview of the different levels:

Level 4/5 – is the Information Systems Enterprise: This is typically the IT network as we know it today, where the primary business functions occur. This is the level that provides business direction and orchestrates manufacturing operations. Enterprise resource planning (ERP) systems drive plant production schedules, material use, shipping, and inventory levels. Popular ERP systems include offerings from Oracle, SAP, Microsoft, and Epicor. While Level 5 is generally your computers and network equipment, because this is a communications diagram, Level 4 is reserved for the specific applications that run on those systems.

Next we have a Demilitarized zone (DMZ), sometimes referred to as “Level 3.5.”This is A recent addition over the last decade, and it includes security systems, such as firewalls and proxies, used to separate or air gap the IT and OT worlds. This is where the IT and OT worlds “converge,” increasing the attack surface for the OT systems. Many plants either do not have this layer or have very limited capabilities. The need for more visibility into the OT layers in order to increase both efficiency through data science as well as provide the necessary visibility for security monitoring has created an increased need for bidirectional data flows between OT and IT systems. In order to minimize direct connectivity between layers 4&5 and 3, this DMZ was created in order to function as a place to drop off and pick up data. You can see here this is great for both log aggregation and patch management.

As for Level 3 This is where the production workflow is managed on the operations floor. Customized systems based on operating systems, such as Windows, are used to perform batch management, record data, and manage operations and plant performance. The systems at this level are called manufacturing execution systems (MES) or manufacturing operations management systems (MOMS). MES/MOMS are specific to the products being processed/manufactured. This layer also consists of databases or historians to record the operations data. The communication between the enterprise

level and manufacturing level typically occurs through a dedicated backhaul network to the main data center or headquarters.

Next, Level 2 is the Control systems we discussed earlier. Here, (SCADA) systems are used to supervise, monitor, and control physical processes. As we discussed, SCADA can manage systems over long distances away from the physical location of the plants, while programmable logic controllers (PLCs) are usually deployed within the plant (down on Level 1) The HMI here in Level 2 allows for basic controls and monitoring, while the SCADA systems aggregate data and send it upstream for recording by the historian in level 3. Devices and strategies at this layer typically communicate over modbus and the other SCADA protocols, and it isn't uncommon to find 2 or more protocols in use in the same physical environment.

Down at Level 1 we have Intelligent devices: the Sensing and manipulation physical processes occurs at this level with process sensors, analyzers, actuators, and related instrumentation. To drive efficiencies, sensors are increasingly communicating directly with their vendor monitoring software in the cloud via cellular networks from here, relying heavily on Internet of Things architectures.

Finally, way down at Level 0 we have the Physical processes: the actual physical actions taking place in the environment as set forth by the devices in Level 2.

And it’s say easy as that! I hope this was helpful for you all; don’t be afraid to check out our website below to see if GRIMM can help you get a handle on any your security needs, from automotive vulnerability research to threat hunting training, we’ve got you covered. Thanks, and have a great day.