Factory on Fire

Managing OT Security

by: Ian Burkett

September 7, 2025

Forward

How do you prevent fires in your OT environment? OT security comes with some large challenges from the equipment being outdated, to a us-versus-them mentality between security, IT, and, most importantly, the engineers.

To the engineer, it feels that we are taking away their power and ability to function smoothly and efficiently. This is not completely incorrect. There was a simpler time when there was little to no defense-in-depth. Rather, technology was open and free. Enterprise IT and OT were unsegregated, but as IT grew in a defensive mindset, OT did not.

As devices were put on the network for ease of access, rules were relaxed and best policies were not yet created. Today, there are a plethora of best practices put out by vendors, NIST, CISA, regulators, and auditors.

Infighting is caused by security taking certain abilities from the engineers. But it is our job to educate them on why security is important. More importantly, we should be pushing to find solutions for them that fit our security needs while helping them maintain productivity.

Guides and frameworks exist everywhere, but what I share here comes from my personal experience. In my first security role, I learned invaluable lessons — lessons that shaped my growth, helped me handle difficult situations, and prepared me for the future.

I had an amazing team whom I admire and cherish. They helped me learn and grow as an individual and a security professional. Like all things in life, there is always room for reflection and improvement. I hope others find this helpful as a small, practical “how-to” for navigating security in the OT world.

Table Of Contents:

1. IT vs. OT: Same Triad, Different Priorities

The IT folks might not like to hear this, but the OT world plays by different rules. While the technology gap between IT and OT is shrinking, the **needs and priorities remain very different.** To build strong security programs, we must bridge that gap—helping IT understand what matters most to OT environments.

In IT, protecting **data** is always the top priority. In OT, protecting **operations and people** comes first.

OT Primary Goals:

- Stay online
- Maintain a safe working environment

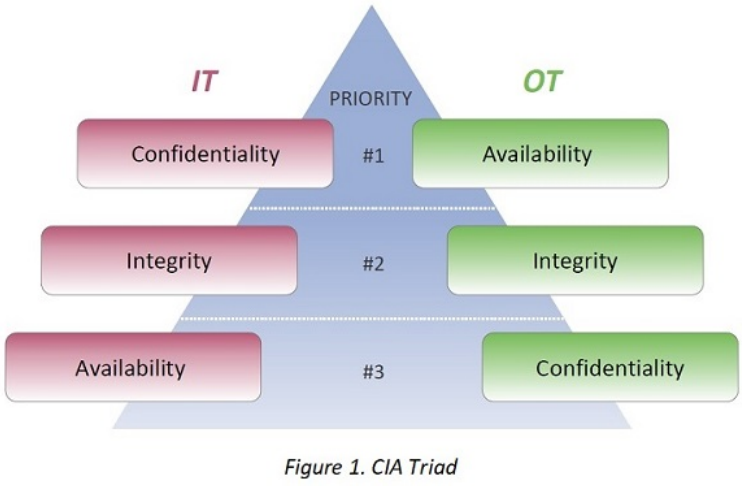

Figure 1: CIA Triad

When we look at the **CIA triad**, IT security teams usually emphasize **Confidentiality** and **Integrity**. Sensitive business and customer data must remain secure, so unencrypted traffic is a high risk. That mindset makes sense for IT systems.

But OT is different. Many industrial devices, especially older ones, simply do not support encrypted protocols. Even newer products can be surprisingly limited; I have worked with devices where every port was enabled by default, with no option to disable them.

For OT, the triad rotates in priority. The focus is on **Availability** and **Integrity**:

- **Availability**: Downtime in a manufacturing plant doesn't just cause inconvenience—it can lead to massive revenue loss.
- **Integrity**: Data must not be altered. Even if it isn't sensitive, changes in process data can create dangerous outcomes.

As long as systems stay online and processes remain accurate, management and engineers are generally satisfied.

The stakes, however, are high. Failure in OT doesn't just risk revenue—it can endanger human lives. Industrial systems control heavy machinery, hazardous chemicals, and processes that, if disrupted, could cause fires, injuries, or fatalities.

This is why OT security professionals must work hand-in-hand with engineers. Security changes should never be made in isolation—they must support operational safety and reliability above all else.

2. The Real Challenge of GRC (and Why No One Wants to Hear About It)

Most people don't want to hear the words **"GRC"** or even **"security."** To many, those words feel like the boogeyman—sparking frustration and resentment. Security teams are often seen as corporate drones that ruin the day, slow things down, and create red tape without improving anyone's life. And honestly, that reputation isn't entirely undeserved—at least when GRC is implemented poorly.

Unclear rules, low awareness, conflicting messages, and an endless stream of "no" can turn any discussion about new technology into a nightmare. In those cases, people will try to skip you altogether. If you don't have strong controls in place, they will bypass you.

And when IT staff get tired of the constant battles, they might start letting things slide. Why argue with end users every day about unpopular rules if there's no system of checks or accountability? That mindset is short-sighted, but it's real. **So ask yourself:**

1. Do you have frequent audits?
2. Do you measure progress with clear KPIs?
3. Is there a standard with real consequences if it's ignored?
4. Do you have controls to prevent unauthorized or sloppy changes?
5. Are you actively investigating when something looks wrong?

GRC is the **cornerstone** of any security program. It doesn't directly stop attacks, but it creates the structure that allows security to succeed. Policies are the foundation—but a policy on paper is useless unless it's enforced, communicated, and supported.

The hard truth? Policy is boring. No one wants to read it, no one fully understands it, and most people don't care—except the person who wrote it.

I once sat in a joint meeting with our security and IT teams during a "Know Your Policy" session. It turned out that some of our policies had existed for years but were never enforced. When we brought them up, IT leaders were frustrated—accusing security

of suddenly creating new rules out of nowhere. In reality, the rules had been in place for decades. This proved that even long-tenured staff had no awareness of unenforced policies.

So what does this mean for you as a security professional? Your policies must be **actionable** and **understandable**. You need tools and processes that support the teams impacted by these policies. If you don't, people will work around you, and shadow IT will thrive.

At minimum, create a **condensed, user-friendly version** of your policies. Keep it short, clear, and easy to find. Nobody wants to play "Where's Waldo?" when looking for critical information. If your IT staff struggles to locate or understand your rules, you can bet end users will ignore them completely.

When assets come into a facility they should be:

- Tagged physically to identify the machine
- Enrolled into a management service
- Tracked in a database with tag numbers, IPs, Host names, Owners, and special requirements.
- Update DNS

I especially recommend DNS. Nothing is more frustrating than static IPs with incorrect DNS names or devices that are behind a NAT router that have a local IP. Instead of an internal IP in your management software. I used to use our inventory software and AV to try to locate these devices but a 192.xxx.xxx.xxx instead of 10.xxx.xxx.xxx shows up you are out of luck for remoting to that machine or even running a reverse lookup.

In the best case scenarios someone knows where there devices are in the worst case you get to walk around to every device to find it. Knowing the location of your devices is a must. If you need to physically touch a computer but have no idea where it is you may not be able to locate it for days or weeks. You can pull the plug and see who screams but in some cases that pc might be hidden away and no one needs it or it doesn't need internet connection its just on the network.

OT technology is not just a windows computer on the shop floor they are also imbedded into machines. Some of those machines restrict the access to windows while the application is running. This means that you might not even know the machine is a traditional computer.

If policies are relaxed machines maybe slipped by IT and placed straight on the network. 802.1X can be expensive to implement. If your using natted routers/firewalls it may not require a internal Ip address. This causes mayhem when it comes to auditing and

locating a machine. It becomes a problem when installed services like antivirus report back a natted address and you cant find it or ping it. It creates issue with monitoring KPI because no one is able to locate it.

This goes the same for engineering devices like PLCS, HMI, Actuators, or anything that touches the network. It is everyone's responsibility to keep track of both IoT and PCs in their control.

Your company should find the best method to assign ownership to a person or a group.

2.1 Asset Management in OT

Asset management ties into both endpoint security and compliance, but I feel for this document its's best placed under **GRC**. Why? Because without governance and accountability, asset management quickly becomes a nightmare.

Why knowing the location matters

In the best case, someone knows exactly where their devices are. In the worst case, you're walking the floor trying to find one. Physical location is a must. If you need to touch a system but don't know where it is, you could lose days or weeks searching. Sure, you can pull the plug and wait to see who screams—but sometimes the PC is hidden away, unused, reporting is wrong or it doesn't even require intranet to function.

Problems arise when:

- No one has kept track of OT devices or PCs.
- There's no accountability for ownership.
- Devices themselves are misconfigured or undocumented.

These problems create compounding issues such as:

- Lack of visibility both remotely and physically.
- Lack of accountability for non compliant standards.
- Slow response times to incidents if you are unable to find the device.
- General hassle to KPIs and metrics.

What Should Happen When Assets Arrive?

- Be **physically tagged** for easy identification.
- Be **enrolled into a management service**.
- Be **tracked in a database** with tag numbers, IPs, hostnames, owners, and any special requirements.

- Have **DNS updated**.

A common pain point I found is **DNS**. I've seen antivirus and inventory tools report back local '192.x.x.x' addresses instead of the correct '10.x.x.x' range, thanks to NAT routers or static IPs with bad records. When that happens, remoting into the system or even running a simple reverse lookup becomes impossible. Tracking down the device turns into a guessing game.

And remember, OT assets aren't just PCs on the shop floor. Many are **embedded into machines**, and some even block access to the underlying OS while the application runs. These devices can appear invisible to anyone who doesn't know better.

The risks are real. If assets bypass IT and land directly on the network, security tools may only see the NAT address, not the actual device. That breaks more than visibility—it undermines audits, monitoring, and incident response.

Asset management must go beyond endpoints. **PLCs, HMIs, actuators, sensors, and IoT devices** all count. The critical step is accountability: every asset should be tied to a person or group. Without ownership, asset management becomes just another spreadsheet—and in OT, that's a dangerous blind spot.

2.2. Change Management

A strong change management process improves both visibility and troubleshooting. Without it, every incident turns into a blame game: "IT changed this," "Security changed that," or "The engineers have been making tweaks all morning." With no record, it's impossible to know who did what—or why.

To avoid this, IT and Security should follow a clear approval process:

- **Document the change** — Write out exactly what you want to do.
- **Create a rollback plan** — Know how to undo the change if it fails.
- **Test if possible** — Use a development or virtual environment to validate changes before going live.
- **Get approval** — Sit down with senior leadership or the right stakeholders and walk through the plan.
- **Agree on timelines and tasks** — Secure approval with digital or paper signatures.

Ideally, use a ticketing system such as Service Now or a formal change management tool. But even if all you have is pen and paper, a signed approval is better than nothing.

This process not only protects you when issues arise, it also creates an audit-able record—something both internal and external auditors will expect to see.

2.3. Vulnerability Management and Risk Exceptions

Vulnerabilities are inevitable, and sometimes risks must be accepted. The key is to ensure that approvals are clear, and that the **risk owner** understands they are accountable for any future issues. If they're not comfortable with that responsibility, then an action plan must be agreed upon to remediate the risk.

Risk management should be data-driven. Use established risk matrices to calculate scores, assign priorities, and set timelines for remediation. Some issues—such as replacing outdated equipment—may require significant time and resources. In those cases, compensating controls can reduce exposure, but the risk must still be tracked and reviewed.

Best practice in my opinion: set a maximum review period of **six months**. Even a "friendly check-in" ensures that risks are not forgotten. Higher-risk vulnerabilities may require shorter timelines or scheduled meetings for regular updates.

Finally, all **non-standard or legacy devices** should be flagged and monitored. This visibility gives you a clearer picture of your overall threat landscape and ensures that risks are managed, not ignored.

3. Defense in Depth

3.1. Endpoint Security

Security improvements in OT don't happen overnight. Some battles are won quickly, while others drag on for months—or even years. The key is to have a plan, get it approved, and work through issues systematically. A Coworker once told me 80% problems will be solved quickly, but the last 20% will be the majority of the problems.

Always assign a **risk owner** for unresolved issues. Once a risk is documented and approved, accountability shifts to the owner—they can't claim ignorance later.

Common Endpoint Security Challenges in OT

- **Legacy systems** — Machines running XP, Windows 7, or outdated server versions that can't be upgraded without risking production downtime.
- **Knowledge gaps** — When experienced staff leave, new personnel may not fully understand how software or process works, creating gaps in whitelisting (network ports, AV, or applications).
- **USB usage** — Uncontrolled removable media continues to be a widespread issue.
- **Unapproved remote software** — Engineers may resist slow approval processes when it comes to vendor support, leading to shadow IT tools that bypass security. This often creates tension between maintaining production support and enforcing standards.

3.2. Base-lining

Every device—whether IoT, IT, or OT—should have a **baseline configuration**. A baseline establishes a known-good state, making it easier to detect deviations, misconfigurations, or malicious changes.

IT/Enterprise Devices:

For Windows and Linux systems:

- Start with a **golden image** configured with required security controls.
- Record any adjustments to the image.
- Avoid bloated images with preinstalled applications; instead, use application control or endpoint management tools to add software later.
- Track changes such as unauthorized installs, security setting modifications, and whitelist updates.

OT/Industrial Devices

For devices like PLCs, baseline after they are fully tested and working. The baseline should capture:

- Firmware/software versions
- Serial numbers
- IP addresses and host-names
- Ladder logic (or equivalent programming)

It's critical to disable remote configuration changes where possible as some of these are a physical switch on the plc and not just a setting. Remember to record all security-relevant settings this task is not just a security task but a engineering task as well.

To strengthen baselines, use professional tools like **Dragos**, free but powerful tools like **Malcolm and Hedgehog, Security Onion,** or other agent-less scanners. OT attacks often involve subtle process changes—detecting these anomalies requires knowing what "normal" looks like. A solid baseline gives you that reference point.

3.3. Antivirus

Every device should have antivirus protection. Windows comes with **Defender**, which integrates easily if licensed properly. Traditional host-based antivirus tools are effective but limited—they mostly rely on **signatures** that must be updated frequently (ideally every four hours). If a file is slightly modified, it may slip past detection. This makes them vulnerable to modern, polymorphic threats.

Modern solutions go further:

- **NGAV (Next-Gen Antivirus)** — Moves beyond signatures by using **heuristics, behavioral analysis, and machine learning**. Stronger against **fileless attacks** and unknown threats.
- **EDR (Endpoint Detection and Response)** — Continuously monitors endpoints, recording **process activity, file execution, network connections, and user behavior**. Provides tools to investigate incidents, contain compromised hosts, and build a forensic timeline.
- **XDR (Extended Detection and Response)** — Expands EDR's reach by **correlating events across multiple domains**: endpoints, firewalls, SIEMs, email systems, cloud services, and more. Gives a **holistic view** of attacks across the environment.
- **MDR (Managed Detection and Response)** — A **service model** where a third-party SOC monitors your environment, triage alerts, and escalates only validated, high-priority threats. Useful for organizations without the people or infrastructure to run 24/7 detection in-house.

These approaches provide much better visibility and response capability than standard antivirus, but they also require investment in people, training, and infrastructure.

3.4. Whitelisting

Application whitelisting can be a tricky and sometimes risky control if not implemented carefully. For example, whitelisting entire folders introduces the possibility that an attacker could drop malicious files into those locations, which would then bypass scanning. Because of this, all whitelisted files, folders, and paths should be strictly documented and continuously monitored.

There are situations where whitelisting specific applications is necessary. In practice, many issues arise when antivirus or endpoint security tools block legitimate applications—especially those performing administrative functions or using uncommon execution paths.

The best approach is to formally register and approve software through a change-control process and configure security tools to recognize this approved software. This ensures that legitimate applications are not blocked, while reducing the attack surface for unauthorized or malicious programs.

Whitelisting is not only for applications but also removable media in general you should refrain and find all possible alternatives to whitelisting removable media.

3.5. Removable Media

Removable media—especially USB storage—should be blocked wherever possible. Input devices like keyboards and mice can remain, but storage-capable devices introduce major risks. Even when vendors provide USB sticks for updates, these should not be allowed.

Instead, vendors should deliver updates through a **controlled service** managed by your organization. Files should be scanned, then distributed via tools like **MoveIT** or a secure network share. This can create friction with engineers, so training is key—help them understand how to use shared services and work with vendors to adopt them.

Special cases, such as license keys stored on USB dongles or OT software tied to MAC addresses, can complicate things. For these:

Possible Solutions:

- **Dongle servers** — Centralize license dongles in a small server, accessible over the network.
- **License servers (on-prem or cloud)** — Replace dongles with centralized cloud license management, though it may come at a cost.

If none of these are possible, conduct a **risk assessment** and get formal approval before whitelisting any exceptions.

3.6. Application and Version Control

All software should go through a **verification and approval process** before installation. Once approved, it should only be installed through authorized deployment tools. This prevents situations where users require admin rights and install unvetted, potentially malicious applications.

Challenges include:

- **Approval bottlenecks** — Vetting new applications can slow down production, especially if developers or engineers need frequent updates. This friction decreases over time as the bulk of applications are approved.
- **Rapid development** — In-house applications may evolve faster than the approval process.

There is a balance that needs to be communicated with engineering departments that deployment of their in-house software shouldn't be ever day with a new update. There should be a dedicated development environment where changes can be made and tested. Before pushing to prod to see if there changes work properly.

This is not just an application issue this is a management problem that has to be explained and dealt with. Unfortunately in most cases money wins out and simulated environments

are expensive to run and in house solutions cost computational resources that your team might not have in there server rack.

Version control isn't just for IT applications—it should also extend to **PLC logic and OT software**. Tracking versions ensures unauthorized or unsafe changes can be identified quickly.

3.7. System Hardening

Finally, periodic system hardening tasks should be enforced:

- Remove unnecessary local accounts.
- Disable unused services.
- Uninstall unused or unapproved software.
- Eliminate written passwords—both stored on machines and taped near devices.

These are the lowest hanging fruit. There more in-depth tasks but in general with these and the previous tasks your should be able to cover system hardening at a basic level.

4. Network & Monitoring

4.1. Separation of IT and OT

The real challenge in OT security often begins with **separating IT and OT environments**. If your systems were designed with separation from the start, life is easier. But separating while production is already running is another story—it changes how engineers support operations, and resistance is usually strong.

Two common models for segmentation are:

- **ISA/IEC 62443 Model** (zones and conduits)
- **Purdue Model** (layered architecture)

Both are valid approaches; the choice depends on how your organization wants to structure and secure operations.

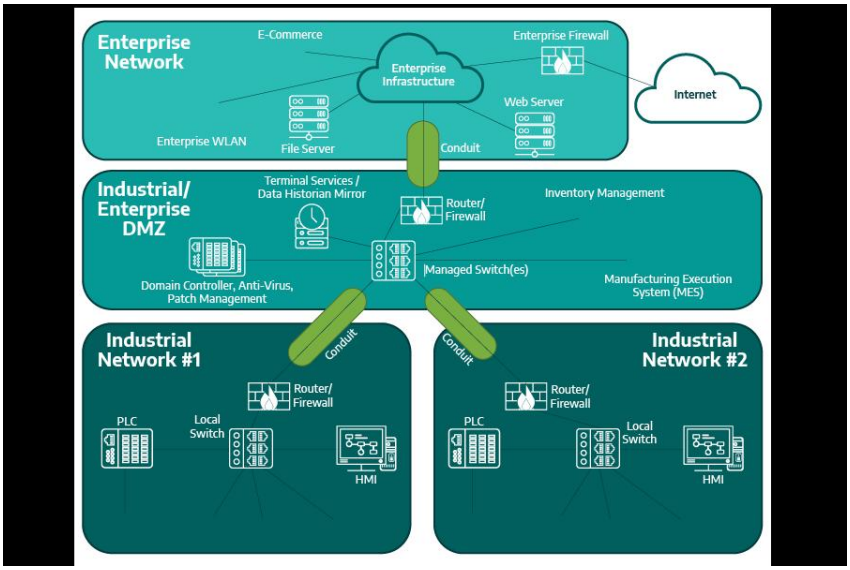

Figure 2: ISA/IEC 62443 Zones and Conduits

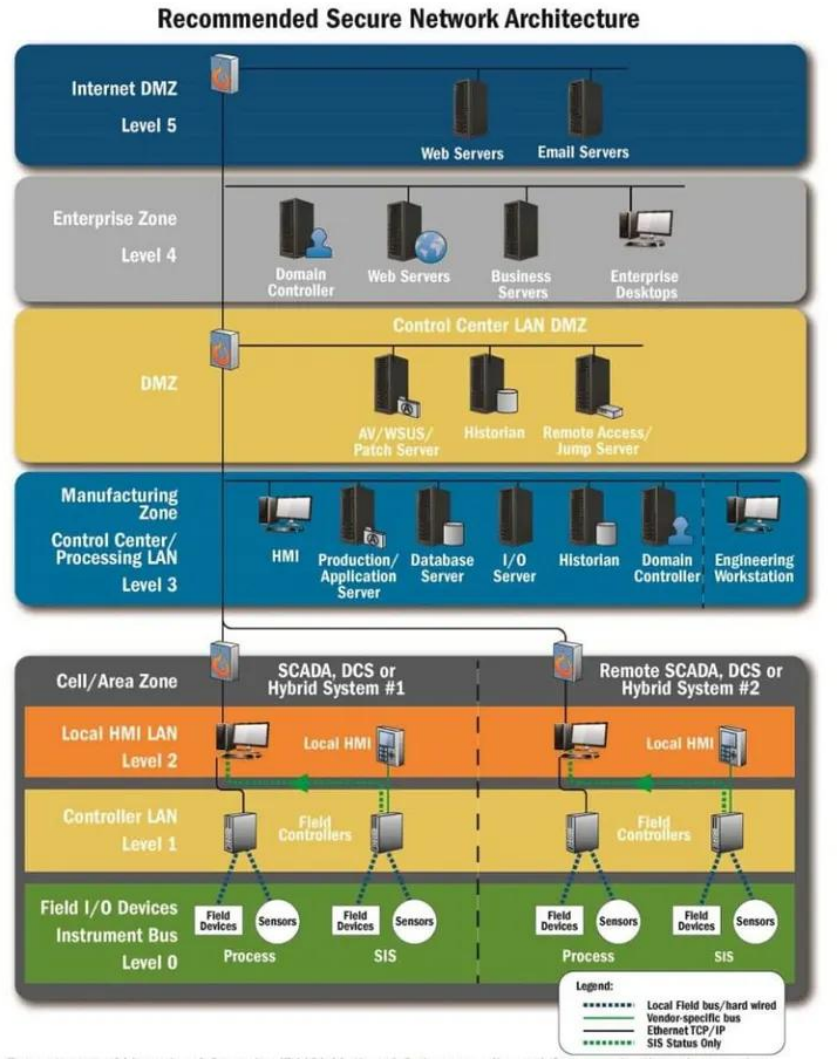

Figure 3: Purdue Enterprise Reference Architecture

4.2. Network Segmentation

I won't cover every technical detail of segmentation, domain trusts, or routing—but I will highlight common challenges I have faced in practice:

- **Remote access for engineers** — Decide early how engineers will connect for remote support. Some tools work on Windows but not Linux, so plan accordingly.
- **Separate machines or VMs** — You may need to provide dedicated laptops or virtual machines for OT access. This adds cost, increases server load, and creates new IT management responsibilities.
- **Domain trust issues** — If IT and OT run separate domains, altering trust will break any "shadow servers" or hidden dependencies that engineers may have been running on the office network. Same with implementing firewalls for the separation.

The goal is not just technical isolation but also ensuring engineers can still do their jobs without resorting to unapproved workarounds. It is a valid complaint when you take away how people have been working for years or decades and offer no compensating resource.

This is also a two way street. Do not let engineers bully you because they have ignored rules because it was easier to do things the wrong way. I would say the best course of action again is to have a clear idea of what is needed, how the services will be rolled out and clearly explain over and over again what needs to be done.

Reality is some things will be over looked, people will ignore to the last minute and you will have to work on those problems on a case by case basis.

A good example of this is unapproved software in the shop floor. Engineers have unapproved remote applications with there vendors to remote into the machine. As we separated and removed unwanted software. The engineers had to switch to our central solution. This process took days if not weeks to get set up. The engineers went from hours to get remote support to waiting weeks because the service to awhile to be approved.

4.3. Firewalls

Separating IT and OT requires a **firewall** to control traffic between networks. This is where things get tricky, because in my experience most people do not truly understand how their applications communicate. Firewalls force that knowledge gap into the open.

Even with a **Next-Gen Firewall (NGFW)**, problems arise. Some standard applications may be misclassified or go unrecognized, causing legitimate traffic to suddenly stop.

Common issues I've seen:

- **High ports** — Services that use dynamically changing ports make firewall rules difficult to maintain.
- **Asymmetric connections** — Some applications establish an outbound connection on one port (e.g., App A → TCP 80 → App B), but then the other side opens a new return connection on a random high port (e.g., TCP 30,000). Since this wasn't expected, the firewall blocks it. Next time, it may use port 40,000 instead.
- **Lack of application knowledge** — Often no one knows which ports or IPs an application actually uses. The mindset is, It's someone else's job to make the software work—I just run it." Firewalls expose this lack of visibility, forcing teams to investigate.
- **Rule granularity** — The more precise your rules, the more time-consuming they are to implement and the higher likely hood the application has hidden features on specific ports that will be prevented and create issues in production.

For Project Leaders

- **Create a clear roll-out plan** — Define actionable steps, timelines, and responsibilities.
- **Engage early with engineering and IT** — Communicate clearly what tasks must be completed before rollout begins.
- **Have approvers in every operating time zone** — Waiting days or weeks for responses creates unnecessary delays.

For On-Site Teams

- **Get asset management right** — This is the moment where knowing exactly what devices you have becomes critical.
- **Document ports and services** — Gather as much detail as possible. If you can, monitor traffic ahead of time to confirm what's truly in use.
- **Be cautious with application-based rules** — These may work in enterprise IT but are inconsistent and unreliable in OT environments.
- **Set expectations with local management** — Explain what will happen, how it will happen, and the risks involved. Make sure leadership understands downtime is inevitable during roll-out, rollback, and re-implementation.
- **Define the process for adding new rules** — Clearly communicate how new devices will be integrated into the network. Proper preparation avoids a cat-and-mouse game with IT and keeps production moving smoothly.

Firewall roll-outs in OT are rarely seamless, but with preparation, communication, and

accountability, you can minimize disruption and build trust between IT, Security, and engineering.

5. Wrap-up

In my view, most of these challenges stem from a **lack of communication and shared understanding** between IT, OT, and Security. Each side assumes the other "should just know" what's needed, which doesn't solve anything.

I've often found myself in the middle as a local security professional, bridging that gap—translating how engineers think to other security teams, and how security requirements impact operations and vice versa.. That middle ground is critical. Security cannot be a "hands-off" discipline where policy is written and blindly followed. It's an evolving process that must balance business needs with technical controls.

There are countless guides, frameworks, and best practices available. But I've found that writing and talking through real-world challenges is the best way to spark ideas, highlight blind spots, and move toward solutions that work in practice—not just on paper.