Tofino Security

White Paper

Version 1.1
Published March 27, 2013

Solving the SCADA/ICS Security Patch Problem

Contents

Authors

Eric Byres, P. Eng., ISA Fellow,
CTO and VP of Engineering, Tofino Security
eric.byres@belden.com
www.tofinosecurity.com

Tofino Security, a Belden Brand

Executive Summary

Since the discovery of the Stuxnet worm in 2010, there has been exponential growth in government security alerts regarding Industrial Control Systems (ICS) and Supervisory Control and Data Acquisition (SCADA) products. It is now clear that these systems were never designed with security in mind and that many contain numerous security-related vulnerabilities.

How to address these flaws is an important question, especially for the many legacy control systems in use today. In the IT world, one solution to security vulnerabilities has been an onslaught of product patches. Can the IT world's strategy of continuous patching work for the ICS world?

This paper explores the challenges of designing and deploying patches for security flaws on control system products like DCS, PLCs and RTUs. We look at vendor data on patch deployment rates in ICS products, the patch rates likely required from end users in the future, and what can be realistically achieved.

We close with an exploration of compensating control-based solutions for security vulnerabilities in the world of automation and control. A combined approach of scheduled patching supported by rapid deployment of compensating controls is outlined. By combining these approaches, companies can reliably secure their control systems.

Welcome to the Patch Treadmill

Since their introduction in the 1960s, electronic Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA) products have been designed for safety, reliability, efficiency and ease-of-use. Security has not been a design consideration. As a result, the control products, the protocols they use and their underlying subsystems are not secure.

This lack of security was a minor problem until the discovery of Stuxnet. Designed specifically to attack Siemens PLC and HMI products, this worm brought ICS and SCADA to the attention of the "security researcher" and hacker communities. Looking for newer and softer targets to exploit, people looking for security holes shifted their focus from Information Technology (IT) products like Windows® to products like HMI software, PLC CPUs and RTUs. As they made this move, they brought highly sophisticated vulnerability discovery tools and techniques to bear on products that had never faced basic security tests. The unfortunate result is that security researchers, hackers and governments are now discovering, publishing and exploiting vulnerabilities in control system products on a continuous basis.

One way this is revealed is the exponential growth in government security alerts. The US government's ICS-Computer Emergency Response Team (ICS-CERT) tracks and publishes Security Advisories for known security vulnerabilities found in industrial products. In the entire decade prior to the discovery of Stuxnet (July 2010), ICS-CERT published 5 security advisories involving 3 vendors. In 2011 there were 215 publicly disclosed vulnerabilities, 104 security advisories and 39 vendors involved[1]. By late 2012, the total publically disclosed vulnerabilities topped 569.[2]

[1] ICS-CERT Advisories and Reports Archive, https://www.us-cert.gov/control_systems/ics-cert/archive.html

[2] McBride, Sean; Documenting the "Lost Decade:" An Analysis of Publicly-Disclosed ICS-Specific Vulnerabilities since 2001", SCADA Security Scientific Symposium (S4). Miami Beach, January, 2012

This disturbing trend was confirmed by Kevin Hemsley of ICS-CERT;

> *In 2011 ICS-CERT experienced a 753% increase in reported disclosures of vulnerabilities in industrial control system (ICS) products. Security researchers (white, gray, and black hats) across the globe are increasing their research in the ICS product arena and the potential impact to critical infrastructure. Coordinated vulnerability disclosures of control system products are increasing rapidly, but so are the instances of unanticipated or full disclosures. The overall pace for ICS vulnerability disclosure is rising at a dramatic pace.*[3]

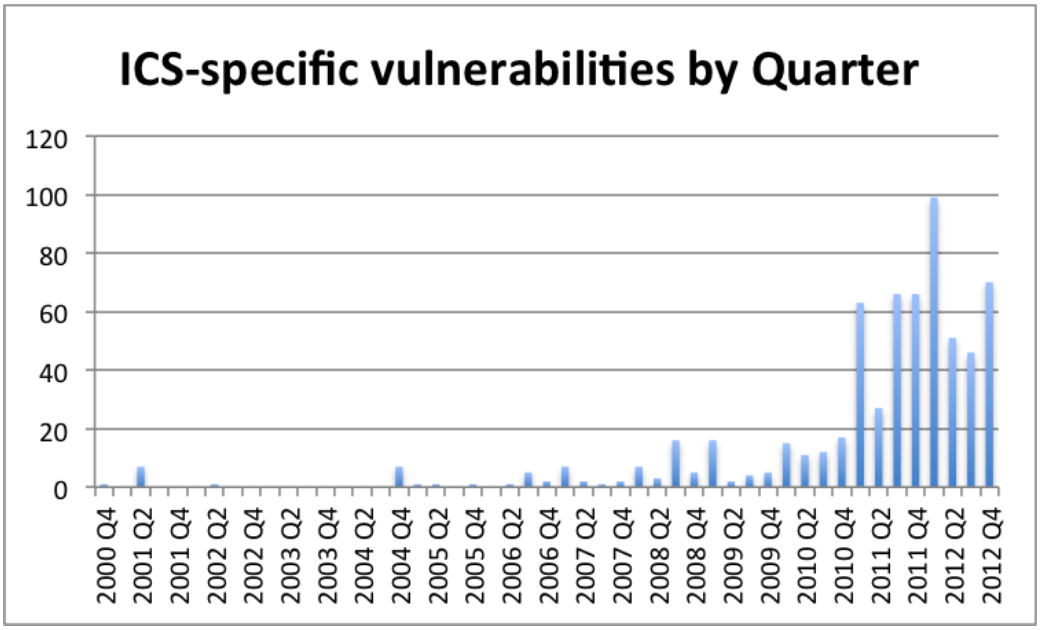


Figure 1: ICS Specific Vulnerabilities in the Public 2001-2021 (by quarter). *Slide provided by Sean McBride of Critical Intelligence Inc.

Typically, these vulnerabilities are disclosed to the world prior to the ICS vendors having patches available for the affected products. Furthermore, 40% of disclosed vulnerabilities included working attack code[4]. Individuals wanting to attack a control system can download exploit tools and run them against a target with little understanding of control systems or the consequences of their actions. And download and attack they do - ISC-CERT reported over 20,000 reports of unauthorized internet access to control systems in the last half of 2012[5]

It is also worth noting that security vulnerabilities aren't just issues when hackers or worms infiltrate a control system. In August 19, 2006 operators at Browns Ferry Nuclear plant had to "scram" the reactor due to a potentially dangerous "high power, low flow" condition. Redundant drives controlling the recirculating water system failed due to "excessive traffic" on the control network. The cause of this incident was attributed to a combination of unnecessary traffic on the control network and a software flaw first noted in 1998 in an ISS Security Advisory[6].

[3] Hemsley, Kevin; SCADA Security: The fight to protect critical infrastructure, 23rd Annual FIRST Conference, Vienna, June 2011

[4] Hemsley, Kevin; SCADA Security: The fight to protect critical infrastructure, 23rd Annual FIRST Conference, Vienna, June 2011

[5] http://www.automation.com/content/cyber-attacks-on-industrial-systems-increasing-rapidly

[6] ISS Security Advisory, ICMP Redirects Against Embedded Controllers, December 10, 1998

Since the researchers, hackers and security issues have essentially migrated from the IT environment, it might be useful to look for a solution there too. In that world (especially for personal computers), security vulnerabilities are addressed with the application of software patches. Unfortunately, this results in a constant cycle that requires multiple patches over the life of a product.

As an example, consider Adobe® Reader 9.0. Between February 2008 and December 2012, Adobe released 33 critical security updates for this product[7]. That translates to a patch approximately every seven weeks for a single software application. Add the fact that most computers contain dozens, if not hundreds, of applications and the result is that the typical IT computer needs patching (with a full reboot) at least once per week. It is highly unlikely that operators of control systems would tolerate production shutdowns of that frequency.

How Many Patches Does a Control System Need?

Of course, it is possible that a control system might require fewer patches than an IT system. Perhaps the software footprint is smaller or the code quality better. If that were the case, then the patching cycle could possibly be extended to once per year, and perhaps even be synchronized with annual maintenance shutdowns. With this scenario, patching could be a workable solution to address software vulnerabilities.

To determine if this was an option, the author participated in the analysis of a process control network (PCN) in a U.S. refinery in the fall of 2008. The survey found that there were 85 computers (workstations and servers) on the refinery PCN, as well as a similar number of industrial controllers. Reliable data was available for only 78 of the computers, but we were able to determine there were 272 distinct processes and/or applications running in those control system computers.

Searching the National Vulnerability Database (NVD)[8] found that 48 of these processes had one or more serious security vulnerabilities. Spread across the refinery PCN, there were 5,455 publically known vulnerabilities, an average of 70 per machine. This number was reduced by almost 50% by instituting an aggressive operating system patch program for Windows. Unfortunately, there were still 2,284 published vulnerabilities remaining in the plant because the applications involved did not have a means of automated patching.

But this was only part of the issue. What about the ICS applications that were not listed in the NVD in 2008? Recent ICS-CERT data makes it clear that SCADA and ICS applications have vulnerabilities. However in 2008 these were not listed in the NVD because no one had yet analyzed and reported on these products. How many of these latent vulnerabilities were hiding in this facility?

To help answer this, we used a simple model for predicting software errors known as "defect density" calculation. Academic research has shown that most commercial software contains between 3 and 10 defects for every thousand lines of code (KLOC). Analysis of defects suggests that between 1% and 5% of these result in vulnerabilities[9] [10] [11]. That works out to be about 0.03

[7] Adobe security bulletins and advisories, https://www.adobe.com/support/security/#readerwin

[8] https://web.nvd.nist.gov/view/vuln/search

[9] Alhazmi, OH et al., "Measuring, analyzing and predicting security vulnerabilities in software Systems", Computers & Security (2006), doi:10.1016/j.cose.2006.10.002

[10] Longstaff, T. CERT experience with security problems in software, Carnegie Mellon University, June 2003

[11] Anderson, Ross. Security in open versus closed systems – the dance of Boltzmann, Coase and Moore. In: Conference on open source software: economics, law and policy, Toulouse, France; June 2002, p. 1–15

vulnerabilities per KLOC for high quality software and 0.5 vulnerabilities per KLOC for poorly written software.

What does that mean in real life? As an example, Windows XP contains about 40 million lines of code (40,000 KLOC). About 1106 moderate or severe vulnerabilities have been listed in the NVD for Windows XP as of October 2012. That works out to be a Vulnerability/KLOC ratio of about 0.0276. Therefore Windows XP is on the low end of vulnerabilities and is pretty good software from a security point of view.

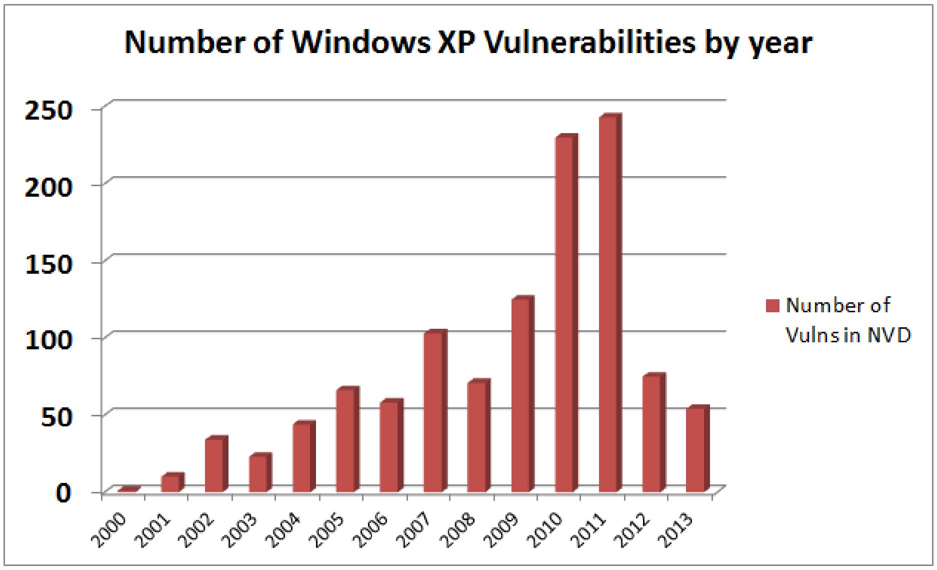


Figure 2: The number of Windows XP vulnerabilities discovered each year as listed in the NVD. Since XP has approximately the same number of lines of code as the applications found on the typical control system computer, it is likely that the vulnerability disclosures for ICS systems will follow a similar pattern.

By scanning each computer, we estimated that the control computers in the refinery contained an average of 60,181 KLOC of application software that had never been listed in the NVD. The majority of this software was ICS and SCADA software, so it was not surprising it had not been studied by the hacking community at that time. Of course, the story might be different today.

Next, we assumed that all these ICS/SCADA applications consisted of high quality software. From this, we used the low vulnerability to KLOC ratio of 0.03 to estimate latent vulnerabilities for the refinery computers. The result was an average of 1,805 yet-to-be-discovered vulnerabilities hiding on every single control system computer. That is a lot of patching!

Looking back at the history of Windows XP vulnerabilities, it is reasonable to assume that all these SCADA/ICS vulnerabilities won't be disclosed at one time. Instead, we can expect a relatively small number of vulnerability disclosures in the first two to five years, as the security researcher community begins to investigate the products in the industrial space. This is likely the period we are in now. Then somewhere between five to ten years after ICS/SCADA products are exposed to widespread security scrutiny, a virtual avalanche of vulnerabilities may occur, resulting in the need to install new control system patches on a weekly basis.

It is worth noting that the above analysis has not considered the firmware in the PLC and DCS controllers. These will also have vulnerabilities and need patches over their lifetime. Controllers typically contain between 1,000 KLOC and 5,000 KLOC of firmware, so based on the same analysis used above, we predicate that they are likely to contain between 30 and 150 vulnerabilities each. Again, if the vulnerability disclosure curves are similar to those we have seen in the IT sector, we can expect a low number of patches in the immediate future, followed by an epidemic of vulnerabilities in a few years.

The above analysis clearly indicates that the frequency of patching needed to address future ICS/SCADA vulnerabilities in both controllers and computers is likely to exceed the tolerance of most ICS/SCADA operators for system shutdowns. Unlike the IT world, most industrial processes operate 24x7 and demand high uptime. Weekly system reboots for patching will be unacceptable for most industrial operations.

The Impact of Patches

Even if patches could be installed without shutting down the process (for example, through the staged patching of redundant controllers), there are still issues with the entire patching strategy. In a landmark study of the patches for post-release bugs in operating system software, Yin et al showed that between 14.8% and 24.4% of all fixes are incorrect and have impacts to end users[12]. Of these incorrect fixes, 43% of these resulted in crashes, hangs, data corruption or additional security problems. In other words, there is a 1 in 12 chance that any patch will impact the safety or reliability of a control system.

Nor are patches always effective at solving the security issues they were designed to address. According to Hemsley, the ICS-CERT has seen a 60% failure rate in patches fixing the reported vulnerability in control system products. Clearly, faulty patches may both fail to properly resolve the vulnerabilities and “break” functionality that is present in the existing control system.

Even good security patches can cause issues for control systems operators. As we discussed earlier, most require the shutdown and restart of the manufacturing process. In addition, they can remove functionality previously relied on by the control system. For example, one of the vulnerabilities the Stuxnet worm exploited was a hardcoded password in Siemens’ WinCC SQL database. While many security analysts criticized Siemens for not quickly releasing a patch to remove the password, it turned out that this “cure” would have been worse than the disease. Customers who manually changed the password quickly discovered that many critical control functions depended on being able to access accounts using the password.

To make matters worse, patches often require staff with special skills to be present when they are installed. For example, the vulnerability that the Slammer worm used in January of 2003 actually had a patch (MS02-039) that was released in 2002. Unfortunately, this didn’t help an oil company with numerous production platforms in the Gulf of Mexico. The company started rolling out the patch in the summer of 2002, but issues with server restarts required Windows experts to be present during patching. Since very few of these experts were safety certified for platform access, most platforms were still not patched when Slammer hit six months later.

When There are no Patches

Of course, using patches to fix vulnerabilities assumes that the vendor has created a patch. According to McBride[13], this is the exception, not the rule – as of January 2012, less than half of the 364 public vulnerabilities recorded at ICS-CERT have patches available.

[12] Zuoning Yin, Ding Yuan, Yuanyuan Zhou, Shankar Pasupathy and Lakshmi N. Bairavasundaram. How Do Fixes Become Bugs? -- A Comprehensive Characteristic Study on Incorrect Fixes in Commercial and Open Source Operating Systems, Proceedings of the 19th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE'11), September 2011

[13] McBride, Sean; Documenting the “Lost Decade:” An Analysis of Publicly-Disclosed ICS-Specific Vulnerabilities since 2001”, SCADA Security Scientific Symposium (S4). Miami Beach, January, 2012

While some security experts accuse the vendors of indifference or laziness[14], there are many factors that prevent the quick release of a patch. In 2010, the author was informed by a major ICS vendor that internal testing had revealed security issues in a mission critical product. Unfortunately, these vulnerabilities were part of an embedded OS supplied by a 3rd party. This OS supplier had refused to address the vulnerabilities and thus the ICS vendor (and its customers) was faced with a situation where no patches were possible.

In a 2011 case involving another ICS vendor, vulnerable backdoors were found in a PLC by an independent security researcher, who publically exposed them. The vendor was able to design a patch to remove backdoors, but then learned that these backdoors were widely used by troubleshooting teams for customer support. To complicate matters, the company had a very thorough quality assurance (QA) process for product changes that required four months to complete. Thus, even if customers were willing to sacrifice support for security, they were faced with a four month window of exposure while they waited for the proper testing of patches to be completed.

Do ICS/SCADA Users Even Want to Patch?

The last example highlights a core problem with a patch-based strategy for control system security. Many customers are reluctant to patch their controllers, as it may degrade service and increase downtime. The vendor noted in the previous example privately reported to the author that they have a 10% patch download rate for released patches.

The author's experience with an ICS security product confirms the reality of low patch acceptance in the field. In September 2010, Tofino Industrial Security System version 1.6 was released. This upgrade addressed a number of security and performance issues and was offered to users at no charge if downloaded in 30 days. All registered users were contacted via multiple emails and the offer was repeated for an additional 30 days due to low initial acceptance. After two months, only 30% of the Tofino users had bothered to download the free upgrade. How many actually installed it is unknown.

It is clear that the IT strategy of continuous patching will not work for ICS and SCADA systems. Vendors face multiple issues when trying to create "quick" patches for a published vulnerability. This includes the fact that safety and QA requirements often delay patch releases. In other cases, a reasonable and safe patch just isn't possible.

The ICS/SCADA customer faces similar issues. The most obvious being downtime or safety risks when patching a critical controller or server. Patch support for legacy products is also an issue – many expect a control product to operate for 20 years, putting it well outside the typical IT support window. Finally, as noted in the Slammer worm example, patches can require significant staff resources to install safely.

Addressing Vulnerabilities Through Compensating Controls

While continuous patching will not work on the plant floor, there are other alternatives. In the telecommunication industry, the concept of compensating controls is widely accepted as a means for safely delaying patch deployments to fit with annual or semi-annual maintenance schedules. For example, vendors of backbone telecommunications equipment often suggest configuration changes to their clients that will block exploits of a known vulnerability without requiring a patch. Microsoft also offers this service to its customers – included in most Security Bulletins is a section called "Workarounds", which they define as follows:

[14] http://www.digitalbond.com/tools/basecamp/

> *"Workaround refers to a setting or configuration change that does not correct the underlying vulnerability but would help block known attack vectors before you apply the update."*

To date, only a few ICS or SCADA vendors have offered this strategy, but the possibilities for compensating controls are numerous. Possible compensating controls include:

- Product Reconfiguration (e.g. "Disable the HTTP port")
- Suggested Firewall Rules (e.g. "Block all HTTP traffic")
- Suggested IDS Rules/Signatures (e.g. Install signatures for logins using a default password)

The idea with these controls is to prevent any message that might exploit a known vulnerability from ever getting to the device in the first place. In other words, if you can't directly fix the flaw, make sure that the conditions where it is an issue cannot occur.

There are some clear benefits to the user with this approach. First, it is safer. Patches that turn out to be flawed usually cannot be removed from a system. However, removing an incorrect compensating control is often trivial. Second, patches can affect the entire operating system in a controller or computer. This often results in unintended consequences that are hard to predict. On the other hand, by using a compensating control such as blocking a vulnerable service it is easier to understand the impact on the industrial process. With compensating controls, the asset owner is in control of his/her fate.

There are also considerable benefits for the control system vendor who chooses to use a compensating controls strategy rather than a patch-only strategy. To begin, compensating controls can be released independent of product development and typically require less QA effort. This translates into a faster response to the customer's security needs. Furthermore, since the compensating controls are independent of the product firmware, they often have less impact on product functionality, lowering customer resistance to using them. Finally, this strategy allows support of legacy products that are too old to justify the effort of a full firmware release.

Of course there are limitations to the compensating controls strategy. For example, vulnerabilities that involve encrypted sessions (such as HTTPS) cannot be addressed with special firewall rules, because firewalls can't typically decrypt and inspect the traffic. But for a large number of the PLC and DCS vulnerabilities we have seen, the technique works well.

Requirements for Success

For compensating controls to work on the plant floor, there are a number of key requirements. First, they must be engineered to have a low impact on process reliability and safety. Any security "solution" that impacts process reliability or safety will be rejected by the customer.

Second, any compensating control must be simple to deploy. In the words of a manager of a chemical company to the ISA-99 Committee on Control System Security: *"We have to make this [security] something a plant superintendent, engineer, or senior operator can do in their spare time, or it will flop."* For example, if the compensating control involves hardware, then the field technician should be required to do no more than:

1. Attach the hardware to a DIN rail or panel.
2. Attach instrument power.
3. Plug in network cables.
4. Walk away…

Good examples of this *"Zero Configuration Deployment"* strategy are the *"Fixed Configuration Firewalls"* offered by a number of Safety Integrated System (SIS) vendors. These firewalls contain factory configured protocol and signature rule sets designed specifically to match product and vulnerability requirements. The benefits they offer include simple installation and the fact that they can typically be installed in a live system without requiring a shutdown. Furthermore, they are designed to be easily upgradeable to address new threats as they appear.

Figure 3: Some typical fixed configuration firewalls provided by automation vendors for securing mission critical operations such as safety systems and pipeline compressor stations.

Compensating Controls Secures Exposed PLCs

An excellent example of using compensating controls to quickly defend against publicly announced vulnerabilities was demonstrated by Schneider Electric in late 2011. In December of that year, security researcher Ruben Santamarta publicly disclosed details of multiple vulnerabilities in Schneider's Modicon PLC product line[15]. At the time of Ruben's disclosure, Schneider had produced a fix for two of the reported vulnerabilities, but was still working on patches for the others. To help customers secure their PLCs while the other patches were being developed and tested, Schneider produced a guide entitled "*Mitigation of the PLC Vulnerabilities Using a Tofino SA"*[16].

This detailed how-to guide explained how to use a Hirschmann Tofino industrial firewall to filter out harmful traffic before it reached the PLC. As shown in Figure 4, these firewalls are placed in front of one or more PLCs. The firewalls are then configured with rules designed to address each of the vulnerabilities.

[15] http://reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1

[16] http://www.schneider-electric.com/download/us/en/details/20609399-RES207869/?reference=Res207869

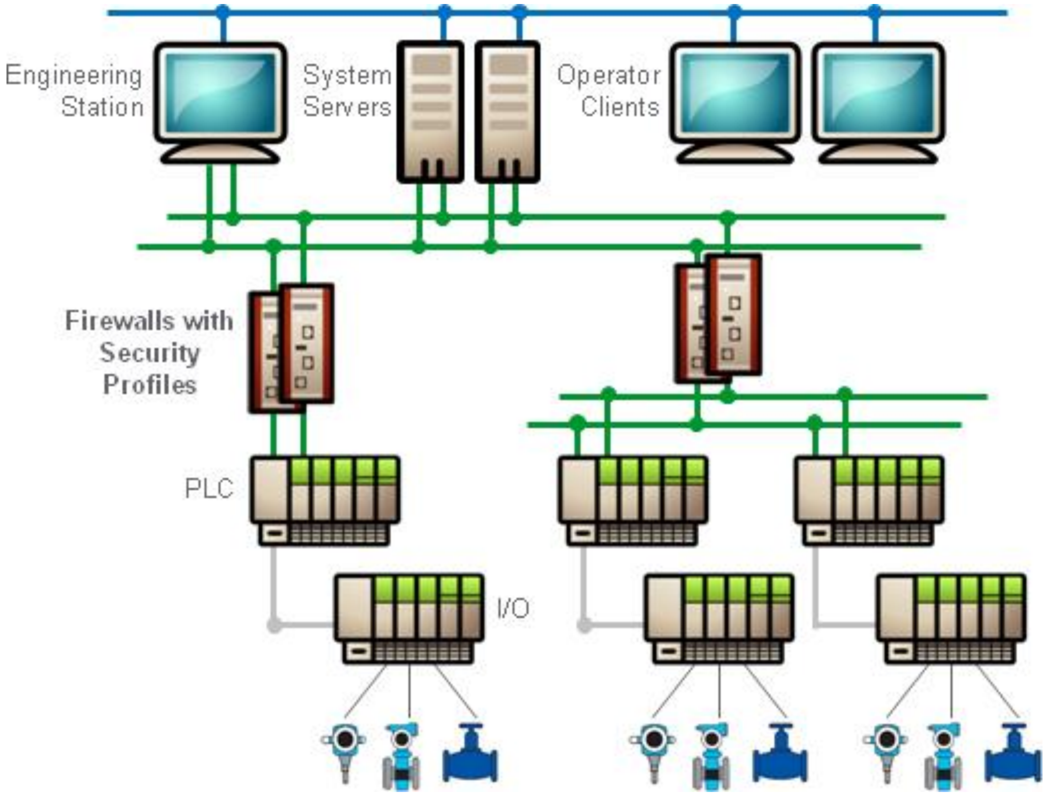


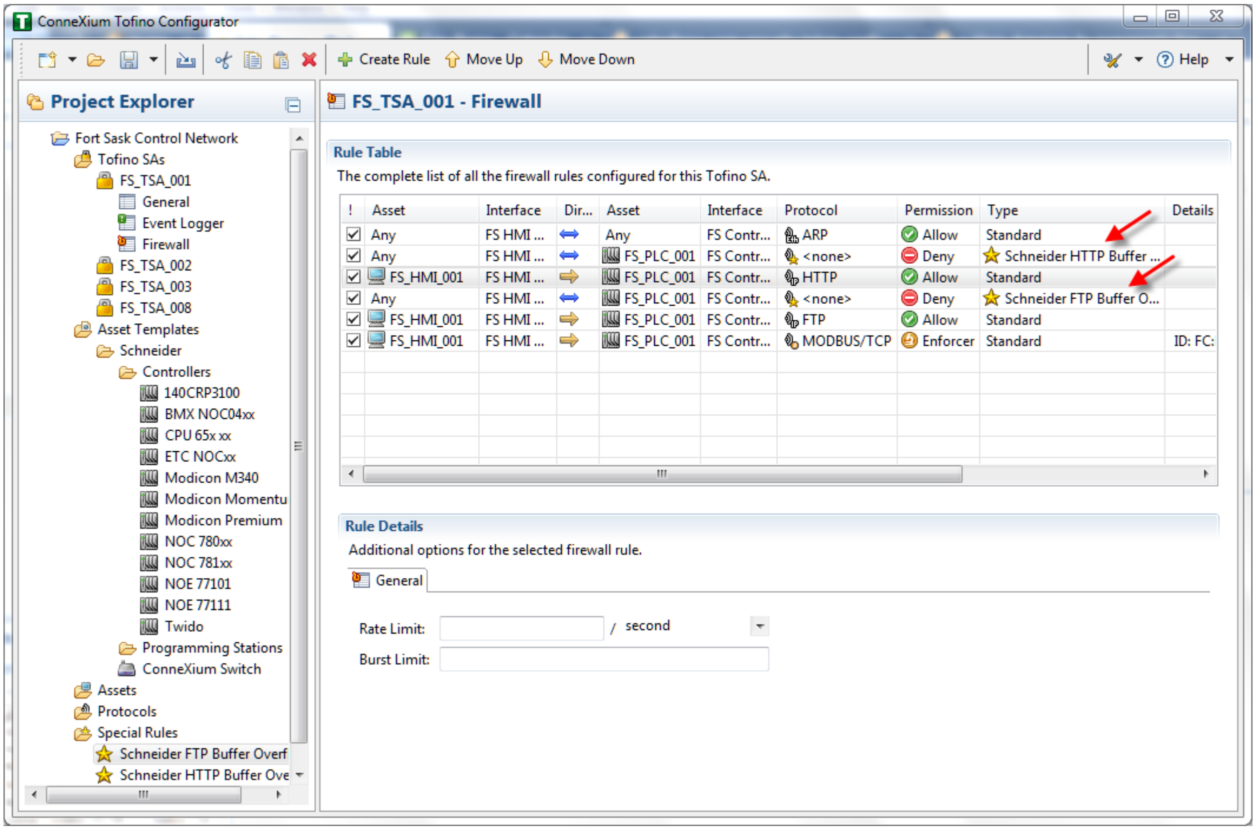
Figure 4: Tofino Firewalls protecting PLCs were recommended by Schneider Electric as a compensating control against publically released vulnerabilities.

In the Schneider Electric example, some of the mitigations were pretty simple to create. For example, blocking a debug service vulnerability was simple. All the user had to do was install a firewall. As long as they didn't specifically add rules allowing the debug traffic, the vulnerability was mitigated.

Other mitigations can be more intricate.

For example, the FTP Buffer Overflow vulnerability (ICS-ALERT-12-020-03) could have been addressed by just blocking all FTP traffic. Unfortunately, that approach may not be acceptable for many facilities – FTP traffic can be essential in some processes.

To address this, Tofino Security worked with Schneider to create "Special Rules". These rules contained algorithms that looked specifically for the behaviors that suggest that the FTP protocol is being exploited. If this behavior is discovered, then FTP messages are immediately blocked by the firewall.



Project Explorer

- Fort Sask Control Network
  - Tofino SAs
    - FS_TSA_001
      - General
      - Event Logger
      - Firewall
    - FS_TSA_002
    - FS_TSA_003
    - FS_TSA_008
  - Asset Templates
    - Schneider
      - Controllers
        - 140CRP3100
        - BMX NOC04xx
        - CPU 65x xx
        - ETC NOCxx
        - Modicon M340
        - Modicon Momentu
        - Modicon Premium
        - NOC 780xx
        - NOC 781xx
        - NOE 77101
        - NOE 77111
        - Twido
      - Programming Stations
      - ConneXium Switch
  - Assets
  - Protocols
  - Special Rules
    - Schneider FTP Buffer Overf
    - Schneider HTTP Buffer Ove

FS_TSA_001 - Firewall

Rule Table

The complete list of all the firewall rules configured for this Tofino SA.

!	Asset	Interface	Dir...	Asset	Interface	Protocol	Permission	Type	Details
☑	Any	FS HMI ...	↔	Any	FS Contr...	ARP	Allow	Standard	
☑	Any	FS HMI ...	↔	FS_PLC_001	FS Contr...	<none>	Deny	★ Schneider HTTP Buffer ...	
☑	FS_HMI_001	FS HMI ...	⇨	FS_PLC_001	FS Contr...	HTTP	Allow	Standard	
☑	Any	FS HMI ...	↔	FS_PLC_001	FS Contr...	<none>	Deny	★ Schneider FTP Buffer O...	
☑	FS_HMI_001	FS HMI ...	⇨	FS_PLC_001	FS Contr...	FTP	Allow	Standard	
☑	FS_HMI_001	FS HMI ...	⇨	FS_PLC_001	FS Contr...	MODBUS/TCP	Enforcer	Standard	ID: FC:

Rule Details

Additional options for the selected firewall rule.

General

Rate Limit: / second

Burst Limit:

Figure 5: Two "Special Rules" designed specifically to detect and block attempts to exploit HTTP and FTP buffer overflow vulnerabilities. These rules are included with device definitions so that they are automatically assigned to the appropriate PLCs.

All these mitigations were made easy to deploy through the use of "Security Profiles". A security profile is a collection of firewall rules, special rules, and protocol definitions designed to address the vulnerabilities for a specific control product. The profile can include complex checks (such as text searches for the attempted use of a default password) that a traditional firewall cannot achieve. Combining the security profiles with the Tofino industrial firewall allowed Schneider to provide a defense for their customers that was immediately effective and that did not require any changes to automation equipment or network configurations.

Schneider clients, ICS and SCADA system owners, benefited by receiving a single, easy-to-deploy package of tailored rules that could be deployed without impacting operations. Site engineers could also confirm that the new rules would not harm their operations by using Tofino Test Mode. This feature allows rules to be tested without actually blocking traffic. The result is that industrial facilities can defend themselves against new threats without having to rely solely on patches for their PLCs, DCS and network hardware.

Final Thoughts on Strategies for Secure Control Systems

There is no denying that ICS and SCADA systems are difficult and risky to patch rapidly. Patching works only when it is based on both proper change control and aligns with maintenance schedules. Furthermore, a patch strategy depends on the rapid release of well-designed and tested software updates for all control products as vulnerabilities are discovered.

With the massive amount of legacy SCADA and ICS equipment currently deployed, this is likely to be the exception rather than the rule.

At the same time, if the ICS-CERT, NVD and the software defect density models are to be believed, there will be a lot of control system vulnerabilities requiring patches in the next decade. These two colliding realities make it clear that the IT model of frequent patching to address vulnerabilities will not succeed in the industrial setting.

This is not to suggest that vendors stop creating patches or companies stop applying them. Addressing vulnerabilities directly in the product is critical for reliable long-term security. But as we have explained throughout this paper, a strategy that depends on rapid patch creation and then deployment is doomed to failure. Instead, by using both compensating controls and patching, the patch cycle can be made more manageable and safer.

It has been well proven in both military and cyber studies that the effective coordination of different security defenses offers the most effective and reliable means to counter attackers. The same applies to addressing security vulnerabilities in ICS and SCADA systems. If we want to successfully prevent hackers, malware and DoS attacks from exploiting flaws in our manufacturing or process systems, we need to use a combined approach of scheduled patching supported by rapid deployment of compensating controls. Only when this occurs can we reliably secure our control systems.