Fred Cohen & Associates - Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Control Requirements for Control Systems... Matching Surety to Risk

Control systems represent a different sort of information technology than most designers and auditors are used to. Unlike the more common general purpose computer systems in widespread use, these control systems are critical for the moment to moment functioning of mechanisms that, in many cases, can cause serious negative physical consequences. Generally, these systems can be broken down into sensors, actuators, and programmable logic controllers (PLCs) themselves controlled by supervisory control and data acquisition (SCADA) systems. They control the moment to moment operations of motors, valves, generators, flow limiters, transformers, chemical and power plants, switching systems, floor systems at manufacturing facilities, and any number of other real-time mechanisms that are part of the interface between information technologies and the physical world. When they fail or fail to operate properly, regardless of the cause, the consequences can range from a reduction in product quality to the deaths of tens of thousands of people, and beyond. And this is not just theory, it is the reality of incidents like the chemical plant release that killed about 40,000 people in a matter of an hour or so in Bhopal India and the Bellingham Washington SCADA failure of the Olympic Pipeline Company that, combined with other problems in the pipeline infrastructure at the time, resulted in the deaths of about 15 people and put the pipeline company out of business. Whether it is a critical infrastructure, a building HVAC or elevator system, or a warehouse, control systems are there and have to be secured for the overall system to be secure.

Why control systems are different

Control systems are quite a bit different from general purpose computer systems in several ways. These systems differences in turn make a big difference in how they must be properly controlled and audited and, in many cases, make it impossible to do a proper audit on the live system. Some of the key differences to consider include, without limit:

- They are usually real-time systems. Denial of services or communications for periods of thousandths of a second or less can sometimes cause catastrophic failure of physical systems, which in turn can sometimes cause other systems to fail in a cascading manner. This means that real-time performance of all necessary functions within the operating environment must be designed and verified to assure that such failures will not happen. It also means that they must not be disrupted or interfered with except in well controlled ways during testing or audits. It also means that they should be as independent as possible of external systems and influences.
- They tend to operate at a very low level of interaction, exchanging data like register settings and histories of data values that reflect the state or rate of change of physical devices such as actuators or sensors. That means that any of the valid values for settings might be reasonable depending on the overall situation of the plant they operate within and that it is hard to tell whether a data value is valid without a model of

the plant in operation to compare the value to.

- They tend to operate in place for tens of years before being replaced and they tend to exist as they were originally implemented. They don't get updated very often, don't run antivirus scanners, and don't, in many cases, even have general purpose operating systems. This means that the technology of 30 years ago has to be integrated into new technologies and that designers have to consider the implications over that time frame in order to be prudent. Initial cost is far less important than life cycle costs and consequences of failure tend to far outweigh any of the system costs.
- For the most part, they don't run the same protocols as other systems, relying on things like DNP, perhaps within ICCP, or ModBus and OPC. These often get executed over serial ports and are often limited to 300 to 1200 baud modem speeds, and have memory on the order of a few thousand bytes.
- Most of these systems are designed to operate in a closed environment with no connection outside of the control environment. However, they are increasingly being connected to the Internet, wireless access mechanisms, corporate networks, and other remote and distant mechanisms running over intervening infrastructure. Such connections are extremely dangerous and commonly used protective mechanisms like firewalls and proxy servers are rarely effective in protecting control systems to the level of surety appropriate to the consequences of failure.
- Current intrusion and anomaly detection systems largely fail to understand the protocols that control systems use and, even if they did, don't have plant models that allow them to differentiate between legitimate and illegitimate commands in context. Even if they could do this, the response times for control systems is often too short to allow any such intervention, and stopping the flow of control signals is sometimes more dangerous than allowing potentially wrong signals to flow.
- Control systems typically have no audit trails of commands executed or sent to them, have no identification, authentication, or authorization mechanisms, and execute whatever command is sent to them immediately unless it has a bad format. They have only limited error detection capabilities and, in most cases, erroneous values are reflected in physical events in the mechanisms under control rather than error returns.
- When penetration testing is undertaken, it very often demonstrates that these systems are highly susceptible to attack. However, doing such tests is quite dangerous because as soon as a wrong command is sent to such a system or the system slows down during such a test, the risk is run of doing catastrophic damage to the plant. For that reason, actual systems in operation are virtually never tested and should not be tested in this manner.
- Many control systems have remote maintenance dial-in access for vendors. This sort of remote maintenance means that strong change control and testing before changes cannot be done on these systems. It also means that a vendor issuing a wrong command could cause a critical system to fail if inadequate redundancy is in place.

In control systems, integrity, availability, and use control are the most important objectives for operational needs, while accountability is vital to forensic analysis. But confidentiality is rarely of import from an operational standpoint at the level of individual control mechanisms. The design and review process should be clear in its prioritization. This is not to say that confidentiality is not important. In fact, there are examples such as reflexive control attacks and gaming the financial system in which control system data has been exploited. But given the option of having the system operate safely or leaking information about its state, safe operation should be given precedence.

Some big questions to ask

While each specific control system has to be individually considered in context, there are some basic questions that should be asked with regard to any control system and a set of issues to be considered relative to those questions.

Question 1: What is the consequence of failure and who accepts the risk?

The first question that should always be asked with regard to control systems is the consequences associated with control system failures, followed by the surety level applied to implement and protect those control systems. If the consequences are higher, then the surety of the implementation should be higher. The consequence levels associated with the worst case failure, ignoring protective measures in place, indicates the level at which risks have to be reviewed and accepted. If lives are at stake, likely the CEO has to accept residual risks. If significant impacts on the valuation of the enterprise are possible, the CEO and CFO have to sign off.

In most manufacturing, chemical processing, energy, environment, and other similar operations, the consequences of a control system failure are high enough to require top management involvement and sign-off. That means that these executives must read the audit summaries, and that the chief scientist of the enterprise should probably understand the risks and describe them to the CEO and CFO before sign-off. If this is not who is making these decisions, an audit team should report this result to the board as a high priority item to be mitigated.

Question 2: What are the duties to protect?

Along with the responsibility for control systems comes civil and possibly criminal liability for failure to do the job well enough and for the decision to accept a risk rather than mitigate it. In most cases, such systems end up being safety systems, having potential environmental impacts, and possibly endangering surrounding populations.

Duties to protect include, without limit, legal and regulatory mandates, industry specific standards, contractual obligations, company policies, and possibly other duties. All of these duties must be identified and met for control systems and, for most high valued control systems, there are additional mandates and special requirements. For example, in the automotive industry, safety mechanisms in the cars that are not properly operating because of a control system failure in the manufacturing process might produce massive recalls, and there may be a duty to have records of inspections associated with the requirements for recalls that are unmet within some control systems. Of course designers should know the industry the operate in, as should auditors, and without such knowledge, items such as these may be missed.

Question 3: What controls are needed and are they in place?

Control systems in use today were largely created at a time when the Internet was not widely connected. As a result, they were designed to operate in an environment where connectivity was very limited. To the extent that they have remote control mechanisms, those mechanisms are usually direct command interfaces to control settings. At the time they were designed, the systems was protected by limiting physical access to equipment and limiting remote access to dedicated telephone lines or wires that run with the infrastructure elements under control. When this is changed to a non-dedicated circuit, when the telephone switching system no longer uses physical controls over dedicated lines, when the telephone link is connected via a modem to a computer network connected to the Internet, or when a direct Internet Protocol connection to the device is added, the design assumptions of isolation that made the system relatively safe are no longer valid.

Few designers of 25 years ago knew about today's threats, and none knew that the Internet would connect their control system to foreign military information warfare experts and saboteurs. Memory and processing were precious, expensive and used carefully to get the desired functionality out of them. They designed for the realities of the day. Today's designers are often unaware of the risks of updated technologies and the extent to which these technologies are prone to failures. Modern control systems may have embedded systems that run operating systems with many millions of lines of code that do things ranging from periodic checks for external updates to running flight simulators from within spreadsheet programs. Almost none of this unnecessary functionality is known to the designers that use these systems, and the resulting unpredictability of these systems means that increased vigilance must be used to make certain that they do what they are supposed to and nothing else.

When connecting these systems to the Internet, such connections are typically made without the necessary knowledge to do them safely. Given the lack of clarity in this area, it is probably important to not make such connections without having the best experts consider the safety of those changes. This sort of technology change is one of the key things that makes control systems susceptible to attack, and most of the technology fixes put in place with the idea of compensating for those changes do not make those systems safe. Here are some examples of things we have consistently seen in reviews of such systems:

- The claim of an “air gap” or “direct line” or “dedicated line” between a communications network used to control distant systems and the rest of the telephone network is almost never true, no matter how many people may claim it. The only way to verify this is to walk from place to place and follow the actual wires, and every time we have done it we have found these claims to be untrue.
- The claim that “nobody could ever figure that out” seems to be a universal form of denial. Unfortunately, people do figure these things out and exploit them all the time, and of course our teams have figured them out in order to present them to the people who operate the control systems, demonstrating that they can be figured out.
- Remote control mechanisms are almost always vulnerable, less so between the SCADA and the things it controls when the connections are fairly direct, but almost always for mobile control devices, any mechanisms using wireless, any system with unprotected wiring, any system with a way to check on or manage from afar, and

anything connected either directly or indirectly to the Internet.

- Encryption, VPN mechanisms, firewalls, intrusion detection sensors, and other similar security mechanisms designed to protect normal networks from standard attacks are rarely effective in protecting control systems connected to or through these devices from attacks that they face. And many of these techniques are too slow, cause delays, or are otherwise problematic for control systems. Failures may not appear during testing or for years, but when they do appear, they can be catastrophic.
- Insider threats are almost always ignored and typical control systems are powerless against them. However; many of the attack mechanisms depend on a multi-step process that starts with changing a limiter setting and is followed by exceeding normal limits of operation. If detection of these limit setting changes were done in a timely fashion, many of the resulting failures could be avoided.
- Change management in control systems is often not able to differentiate between safety interlocks and operational control settings. Higher standards of care should be applied to changes of interlocks than changes in data values because the interlocks are the things that force the data values to within reasonable ranges. As an example, interlocks are often bypassed by maintenance processes and sometimes not reverified after the maintenance is completed. Standard operating procedure should mandate safety checks including verification of all interlocks and limiters against known good values and external review should keep old copies and verify changes against them.
- If accountability is to be attained, it must be done by an additional audit device that receives signals through a diode or similar mechanism that prevents the audit mechanism from affecting the system. This device must itself be well protected in order to keep forensically sound information required for investigation. However; since there is usually poor or no identification, authentication, or authorization mechanism within the control system itself, attribution is problematic unless explicitly designed into the overall control system. Alarms should be in place to detect loss of accountability information, and such loss should be immediately investigated. A proper audit system should be able to collect all of the control signals in a complex control environment for periods of many years without running out of space or becoming overwhelmed.
- If information from the control system is needed, it should run through a digital diode. For example, power grid status is public for real-time analysis. This should be one way!
- If remote control is really needed, that control should be severely limited and implemented only through a custom interface using a finite state machine mechanism with syntax checks in context, strict accountability, strong auditing, and specially designed controls for the specific controls on the specific systems. It should fail into a safe mode and be carefully reviewed and should not allow any safety interlocks or other similar changes to be made from afar.
- To the extent that distant communication is used, it should be encrypted at the line level where feasible; however, because of timing constraints, this may be of only limited value. To the extent that remote control is used at the level of human controls, all traffic should be encrypted and the remote control devices should be protected to

the same level of surety as local control devices. That means, for example, that if you are using a laptop to remotely control such a mechanism, it should not be used for other purposes, such as email, Web browsing, or any other non-essential function of the control system.

- Nothing should ever be run on a control system other than the control system itself. It needs to have dedicated hardware, infrastructure, connectivity, bandwidth, controls, and so forth. The corporate LAN should not be shared with the control system, no matter how much there are supposed to be guarantees of quality of service. If Voice over IP (VoIP) replaces plain old telephone service (POTS) throughout the enterprise, make sure it is not replaced in the control systems. Fight the temptation to share an Ethernet between more than two devices, to go through a switch or other similar device, or to use wireless, unless there is no other way. Just remember that the entire chain of control for all of these infrastructure elements may cause the control system to fail and induce the worst case consequences.
- Finally, experience shows that people believe a lot of things that are not true. This is more so in the security arena than in most other fields and more critical in control systems than in most other enterprise systems. When in doubt, don't believe them. Trust but verify.

Perhaps more dangerous than older systems that we know have no built-in controls, are modern systems that run complex operating systems and are regularly updated. Modern operating platforms that run control systems often slow down when updates are underway or at different times of day or during different processes. These slowdowns sometimes cause control systems to slow unnecessarily. If an antivirus update causes a critical piece of software to be detected in a false positive, the control system could crash. And if a virus can enter the control system, the control system is not secure enough to handle medium or high consequence control functions. Many modern systems have built-in security mechanisms that are supposed to protect them, but the protection is usually not designed to assure availability, integrity, and use control, but rather to protect confidentiality. As such, they aim at the wrong target, and even if they should hit what they aim at, it won't meet the need.

Quick close

The list of things to look out for is indeed extensive, and this article has only touched the surface. But hopefully it will get you started in thinking through controls for control systems. As always, IT general controls apply to control systems, and should be applied with additional rigor for higher consequences, but specialized knowledge is helpful as well.

As in all systems, it is important to take a systematic comprehensive approach to get toward reasonable solutions. The approach we take is based on the models of security architecture developed over many years, and as available at http://all.net/ under “Security Architecture”. But any other well developed approach will do, so long as care is taken to be comprehensive and detailed understanding of the nature of the systems and the implications of that nature are present and taken into account.

References – now - soon to come – and fish for yourself

Search the Internet for each of the real-time control protocols using a term like “DNP protocol”

http://www.inl.gov/scada/standards/index.shtml (Idaho National Laboratory) is an excellent site dealing with standards for SCADA systems.

NIST (http://www.isd.mel.nist.gov/projects/processcontrol/) has a process control security standard requirements forum that helps define the standards that have to be met in different circumstances.

More details on these sorts of systems will appear in my upcoming book “Introduction to Critical Infrastructure Protection”, due out in mid-2008!

Upcoming Events

May ?, 2008 – and forward...