CHAPTER

8 Risk and Vulnerability Assessments

INFORMATION IN THIS CHAPTER

- Cyber Security and Risk Management
- Methodologies for Assessing Risk within Industrial Control Systems
- System Characterization
- Threat Identification
- Vulnerability Identification
- Risk Classification and Ranking
- Risk Reduction and Mitigation

The concept of cyber security goes hand-in-hand with how an organization views and manages risk. Risk is often correlated to the vulnerabilities that may or may not exist with the organization's business enterprise, including risk to and from business systems, IT infrastructure, automation and control systems, and physical business assets that may be directly under the control of one of the aforementioned systems.

The overall process of implementing cyber security controls is meant to reduce business risk. However, if one does not understand their exposure to and tolerance of risk, then the overall effectiveness of these controls may be somewhat less than expected. The deployment of cyber security in terms of security policies, administrative procedures, business processes, and technological solutions is meant to target specifically identified areas of risk and reduce the impact to an organization should a cyber event occur targeting one of the business assets. If an organization fails to identify areas of risk, how can it properly select, implement, and measure security controls that are meant to reduce these risks?

This topic could fill an entire book. It is not practical to attempt to cover all aspects of risk and vulnerability management in a single chapter. Instead, this chapter will focus on the highlights associated with implementing a risk and vulnerability assessment process specifically designed for industrial systems. Detailed resources and references are provided throughout this chapter.

CYBER SECURITY AND RISK MANAGEMENT

WHY RISK MANAGEMENT IS THE FOUNDATION OF CYBER SECURITY

The concept of "functional safety" within most industrial facilities is a cornerstone in the overall operation of the facility, as well as an important key performance indicator (KPI) used in evaluating a company. The deployment of functional safety is well defined by leading international standards including IEC 61508/61511 and ANSI/ISA 84.00.01, which are based around the process of identifying risk in terms of Process Hazard Analysis (PHA), Hazards and Operability Analysis (HAZOP), and so on, and then using methods to specifically reduce these risks through the deployment of mechanical and instrumented systems. The concept of "operational security" closely aligns with functional safety in terms of risk identification, risk reduction through the deployment of security controls, and risk management through continuous and periodic monitoring of the industrial security systems. These ideas are documented in several standards on operational security (see Chapter 13,"Standards and Regulations").

The easiest way to understand the importance of risk and how it relates to not only the selection of cyber security controls and methods but also its overall effectiveness is to answer one simple question: Given a FIXED amount of MONEY, and a FIXED period of TIME to secure an industrial control system (ICS), what would you do?

There are many cyber security control "catalogues" that will list hundreds of various procedure and technological solutions that can be implemented (see Chapter 13, "Standards and Regulations"). The first step here must be to understand and establish an acceptable level of risk or what is called "risk tolerance." It is possible to manage this "unmitigated" risk in one of four ways:

1. Mitigation (you manage)
2. Transferal (others manage)
3. Avoidance (no one manages)
4. Acceptance (stakeholder's manage).

Risk mitigation is the process of reducing these catalogues of controls down to an effective list that is designed to help reduce specific risks to an organization. It should be obvious at this point, and by the fact that you are reading this book, that the risks facing organizations are constantly changing, and that with this dynamic landscape comes the possibility that risks may appear tomorrow that did not exist today. This is why cyber risk management is considered a continuous process of identification, assessment, and response, and not something that can be addressed once and left unvisited for long periods of time.

To look at how risk directly impacts industrial environments that depend on ICS to maintain a safe, efficient, and profitable environment, let us begin with a high-level identification of risk. What is the greatest threat facing your company's industrial systems?

1. People's Liberation Army Unit 61398
2. On-Site Control Systems Engineer

3. Anonymous “Hacktivists” Group
4. Vendor Site Support Specialist
5. Package Equipment Supplier.

These risks cover a broad range of threats that include both internal and external sources, which may use targeted or nontargeted methods, with both intentional and unintentional motives. Nearly 80% of the incidents impacting ICS are “unintentional” yet only 35% of these events were originated from an “outsider.”[1] Many organizations are resistant to objectively consider the actual threats to their industrial systems and risk they represent. Another report confirms that in the analysis of 47,000 incidents (not necessarily incidents against ICS), 69% of these events originated from internal threats acting carelessly rather than maliciously.[2] Embedded devices and network appliances were targeted in 34% of the incidents impacting ICS, while Windows-based ICS and enterprise hosts were targeted 66% of the time.[3]

When the top security controls deployed include anti-virus software, firewalls, antispyware software, VPNs, and patch management,[4] it is clear that these controls do not necessarily align with your most likely threats. It is also obvious at this point that the security controls that are necessary to protect against each of these threats may be quite different. It seems logical that with fixed budgets and schedules, risks should be prioritized and controls selected based on this ranking.

WHAT IS RISK?

There are numerous definitions of risk, depending on the entity used to define it, yet they all tend to contain several common elements. The definition that seems most aligned with the concepts of risk applied to operational security is from the International Organization for Standardization (ISO) who defines risk as “the potential that a given threat will exploit vulnerabilities of an asset … and thereby cause harm to the organization.” From this definition, it is illustrated that risk is a function of

- The *likelihood* of a given Threat Event
- Exercising a particular “*potential*” Vulnerability of an asset
- With resulting Consequences that impact operation of the asset.

There are two modifiers highlighted (“likelihood” and “potential”) that will be addressed shortly. A fundamental concept of risk management is that you can reduce or mitigate risk by addressing any one or all of these three elements. Many believe that the easiest method of reducing risk is through the identification and elimination of vulnerabilities that may potentially be exploited. The best example of this is through the deployment of a patch management program to regularly update asset software to remove identified security flaws and program anomalies that could impact performance. It is also possible that one could reduce risk by “containing” an event and limiting the extent of resulting damage. This method of risk reduction is often overlooked, and can in fact be less expensive and more effective when compared with other more obvious controls. An example of limiting damage following an initial breach is network segmentation and the creation of security zones and conduits (see Chapter 9, "Establishing Zones and Conduits") that is designed to limit the ability of a threat to propagate within the industrial network(s). Another example of limiting consequences following an initial attack is through more granular communication egress control—such as configuring "outbound" rules on host-based firewalls to minimize the extent to which a compromised host can function after a breach.

The **Threat Event** actually consists of components that all can significantly impact risk, including

- Threat Source or Actor to carry out the event
- Threat Vector to initiate the event
- Threat Target which the event attacks.

As before, addressing one or more of these elements can reduce risk. Vectors, such as communication paths or unprotected USB ports, can have security controls deployed that further restrict the entry points used to initiate an attack. The term "reducing the attack surface" refers to the method by which targets that could be compromised are protected or eliminated altogether. An example of this might be to disable unused communication services within an ICS controller that depend upon weak or vulnerable industrial protocols.

The terms Threat Source and Threat Actor are often used interchangeably and essentially refer to the human aspect of the attack. There are three characteristics of any Threat Source that must exist in order for a cyber-attack to occur. These include

- Capability to carry out the attack
- Intent to cause harm
- Opportunity to initiate the event.

There are a large number of tools, both open-sourced and commercial, that provide the ability to attack ICS assets with little or no Capability or specific system knowledge. What is often missing here is the Intent of the Source to actually cause damage or harm. Like the attack tools available, resources like Shodan and information-exchange communities like Expert Exchange provide sufficient Opportunity for would-be attackers to identify and attack potential ICS targets. It is very difficult for an organization to reduce risk by focusing on outside sources because much of this is not in their direct control. However, if the attack originates from an inside source, or if an outside attacker gains a foothold, from which additional attacks could be leveraged from the inside, the threat becomes more manageable.

So how does the On-Site Control System Engineer (i.e. insider) pose a threat to ICS? It is obvious that the insider in this case has extensive *Capability* and sufficient *Opportunity* to initiate the attack. The "malicious" insider possesses ample *Intent* to cause harm. What Intent does the "unintentional" insider possess when performing an accidental action that causes harm to the ICS? The actual Intent in this case is very low. However, due to other surrounding factors that are very high (in-depth system knowledge, elevated access privileges, direct access to ICS assets, use of unauthorized tools, intentional bypassing of security policies, etc.), the resulting net risk is very high. This is the primary reason that an insider, such as the On-Site Control Systems Engineer or ICS Vendor Site Support Specialist, are likely Targets in the early phases of a blended attack, since someone masquerading as an insider can be very difficult to detect and mitigate.

Vulnerabilities, both disclosed and latent or undisclosed, pose a real and obvious risk to industrial networks. A total of 832 vulnerabilities have been disclosed affecting ICS through July 2014, with more than 10% of the total discovered in the preceding six months.[5] More than 80% of all ICS vulnerabilities have been discovered since Stuxnet was reported in 2010.[6] It has become clear that security research and vulnerability identification of ICS components has taken on an important role. Traditional information security conferences like Black Hat and DEFCON now include ICS presentation content, dedicated tracks, and associated training workshops.

Information security focuses on assets that commonly comprise IT business systems, the data contained on these systems, and information as it is generated, transmitted, and stored. The **Consequences** that result from a successful cyber-attack can be large. The actual cost of the recent data breach at retailer Target in 2013 was still unknown at the time of publishing,[7] but some are estimating the cost to Target alone could exceed US$1 billion.[8] Target expects to spend US$100 million to upgrade their point-of-sale payment terminals following the breach.[9]

Consider now that operational security must manage risk to not only the direct ICS assets, but also those assets that are under the control of the ICS including the physical plant or mill, mechanical equipment, employees working in the facility, the surrounding community, and the environment. Consequences that result from a cyber-attack on an ICS are less likely to have a direct impact to the system itself, but rather cause the plant under control to operate improperly, which may impact product quality or production rates, possibly even tripping or shutting the plant down. Mechanical damage may occur, leading to costly repair or replacement and extended plant downtime. Hazardous materials could be released directly impacting the surrounding community often resulting in fines. Events could directly result in loss of human life.

Figure 8.1 illustrates the relationships between the concepts and terms previously mentioned and how each interdepends on others as part of the overall risk process.

STANDARDS AND BEST PRACTICES FOR RISK MANAGEMENT

There are a variety of nationally and globally recognized standards and best practices that focus on the concept of risk management. Most of these documents, however, form a foundation for "information security risk" rather than "operational security risk." In other words, these documents do not form the basis of a risk management framework that may be used to identify and disclose important risk factors necessary to support federal regulations (e.g. those risks typically reported in a company's Annual Report, Form 10-K, or similar), but rather only those risks facing IT systems. It should be clear from the previous section that operational security risk extends

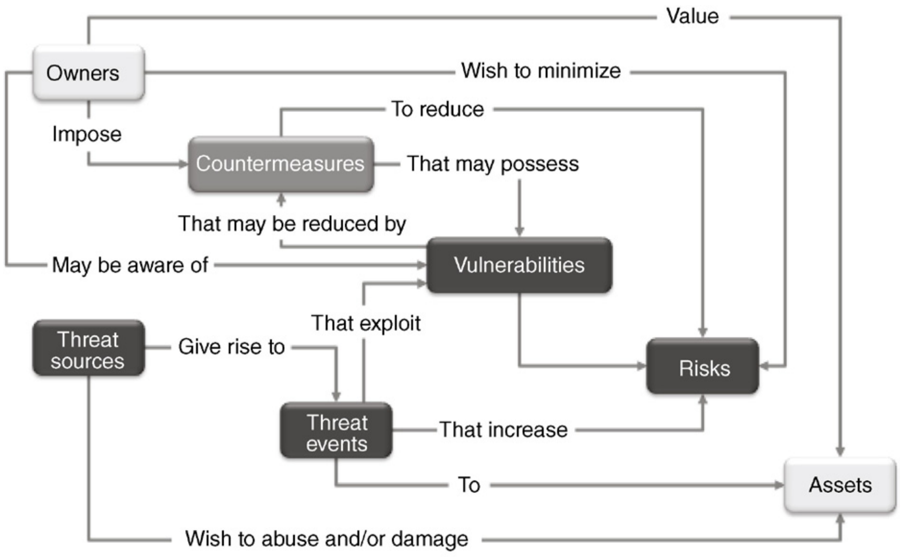


FIGURE 8.1 Understanding risk relationships.[26]

beyond the physical and logical ICS assets to the physical plant that is under control of the ICS components.

Some of the organizations that maintain recognized documents include the European Union Agency for Network and Information Security (ENISA), International Organization for Standardization (ISO), the US National Institute of Standards and Technology (NIST), and many others. See Chapter 13, "Standards and Regulations," for more information on industry best practices for conducting ICS assessments.

Table 8.1 lists a few of the current standards and best practices pertaining to risk management frameworks and assessment techniques.

Many of these documents contain similar requirements using slightly different vocabularies or minor sequence alterations. It becomes clear that many of these documents offer the same basic guidance addressing key requirements including

- Asset identification
- Threat identification
- Vulnerability identification
- Existing security controls identification
- Consequence identification
- Consequence analysis
- Risk ranking
- Security controls recommendations.

Few documents have been drafted and approved for direct applicability within manufacturing environments and upon the industrial systems commonly used. It is necessary for this reason to alter these methodologies in order to tailor the objectives and deliverables to more closely align with these industrial systems and the

Table 8.1 Risk Methodology Standards and Best Practices

Organization	Publication Number	Description
BSI	100-3	Risk Analysis based on IT-Grundschutz
CERT	OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
ENISA		Principles and Inventories for Risk Management / Risk Assessment Methods and Tools
ISO/IEC	27005	Information Security Risk Management
ISO/IEC	31000	Risk Management
ISO/IEC	31010	Risk Assessment Techniques
NIST	800-161	Supply Chain Risk Management Practices for Federal Information Systems and Organizations
NIST	800-30	Guide for Conducting Risk Assessments
NIST	800-37	Guide for Applying the Risk Management Framework to Federal Information Systems
NIST	800-39	Managing Information Security Risk: Organization, Mission, and Information System View

operational security risk reduction goals desired. Figure 8.2 represents one hybrid methodology that has been developed to illustrate the steps necessary to perform an effective ICS cyber risk assessment. Each of these components will be discussed in the remainder of this chapter.

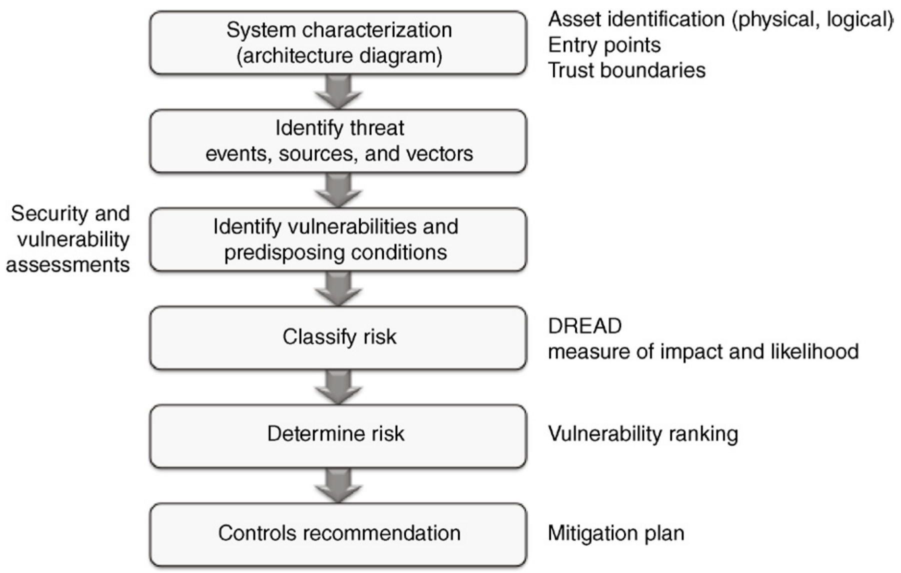

System characterization (architecture diagram)
Asset identification (physical, logical)
Entry points
Trust boundaries
Identify threat events, sources, and vectors
Security and vulnerability assessments
Identify vulnerabilities and predisposing conditions
Classify risk
DREAD
measure of impact and likelihood
Determine risk
Vulnerability ranking
Controls recommendation
Mitigation plan

FIGURE 8.2 Methodology for assessing risk to industrial control systems.

METHODOLOGIES FOR ASSESSING RISK WITHIN INDUSTRIAL CONTROL SYSTEMS

The methodology illustrated in Figure 8.2 defines the process that will be used to identify threats and vulnerabilities that could compromise the operation of not only the ICS, but also the equipment directly and indirectly under its control. This methodology blends elements of a traditional Risk Assessment with Security Testing. The Risk Assessment elements will define the overall "strategy" used to select security controls based on presumed risk, while the Security Test will define the "operations" of the system to verify the completeness that security and associated controls exist within the system under consideration. It can sometimes be confusing the difference between assessing risk and assessing security. This should become clearer shortly.

SECURITY TESTS

The benefit one receives from any security test is commonly thought to be proportional to the number of vulnerabilities that the test identifies. These vulnerabilities may either be due to the (lack of) security capabilities of the system under consideration, or the thoroughness of the assessment. The objective should be to establish a methodology that is based on criteria that help drive consistency from assessment to assessment, and that allows common vulnerabilities that may exist across multiple systems to be uncovered.

Vulnerabilities are discovered, disclosed, and patched daily, along with new exploits and mitigation techniques targeting these weaknesses. Any assessment, audit or test that is conducted therefore only represents a snapshot in time. This is the motivation behind a "repetitive" process that is triggered by external events that may include

- Changes to the system like a component upgrade or system migration.
- Changes to the threat landscape such as the release of a new exploit kit like Gleg's SCADA+ Pack for Immunity CANVAS or a new campaign like Dragonfly/Havex.
- Elapsed periods of time.

The purpose of these Security Tests will be to focus on the identification of not only system vulnerabilities, but also the security controls that may (or may not) be deployed and whether or not they are still effective against the changing threat landscape. This area of focus is shown in Figure 8.3.

The goals of the Security Test will be to assess the current level of security the system under consideration provides in a particular installation. This means that it is important to look at not only the system-specific details (e.g. ICS vendor, network vendor, software, and hardware revisions) but also site-specific factors (e.g. geographical location; compliance with corporate policies, procedures, guidelines and standards; service level agreements (SLA); and project-specific documentation). This will then facilitate the identification of vulnerabilities within the system under

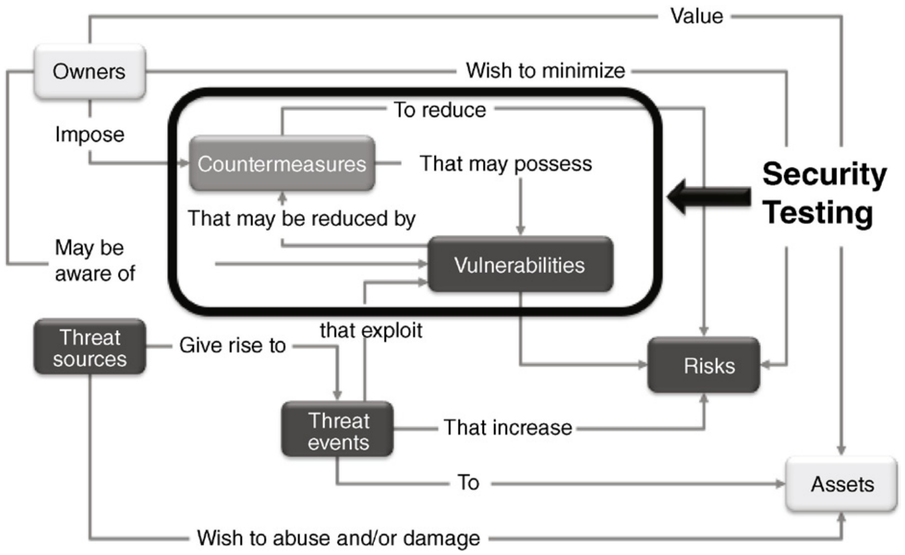


FIGURE 8.3 Objectives of security testing.

consideration. These vulnerabilities may not necessarily be technical flaws, but could be procedural or engineering errors. Once these vulnerabilities are identified, they will then be ranked in terms of severity and actions will be developed to remediate or mitigate these weaknesses.

Vulnerabilities can be found either by evaluating the system in the form of an assessment, or by attempting to attack the system in a manner consistent with what a hacker or external threat may do to compromise the system commonly referred to as a "penetration test" or "ethical hacking" exercise. Penetration tests provide an accurate representation of how the system appears to a potential attacker, and what actions might be required for the attack to be successful. They are also valuable in demonstrating whether or not a component or system can in fact be compromised through the discovery of exploitable vulnerabilities. The results or return on investment from the penetration test are likely to be heavily dependent on the skills and capability of the tester. These types of tests do not typically identify a high percentage of the actual vulnerabilities that exist within the system, and could negatively impact the system.

Shift the attention now from the external threats to the internal ones. It has been mentioned earlier that the insider threat typically has the potential for much greater impact to the ICS, and also possesses significant knowledge of the particular system. The purpose of these Security Tests is not to exploit a system, but to determine the relative level of security a system possesses and identify ways to improve the overall level of security that remains. This is the primary reason that the details to follow will be based on assessing the vulnerabilities a system possesses from the point of view of the insider—who like an outsider also represents a credible Threat Actor.

CAUTION

Penetration Testing or "Ethical Hacking" is rarely performed on operational ICS systems and networks due to the risks to ICS operation. It was mentioned that most of these types of tests aim to identify exploitable vulnerabilities. The primary goals of safety and reliability mean that no test shall have any risk of impact to the operation of a component or the system under test. To perform adequate penetration tests in a safe manner, a dedicated nonproduction test environment should be utilized.

Security Audits

Security Audits are commonly performed to test a particular system against a specific set of policies, procedures, standards, or regulations. These criteria are commonly developed based on knowledge of "known" threats and vulnerabilities. They are also further complicated by the fact that once a new, emerging or sophisticated threat is discovered, it can take time for the documents to be adjusted from any deficiencies that the threat may have exploited. Audits do not typically uncover unexpected or latent vulnerabilities for this reason.

Audits can be conducted using either active collection techniques that require direct access to the system(s) under consideration, or passive techniques that commonly employ questionnaires and checklists. For this reason, audits usually do not require as many resources to conduct as a more thorough security assessment or test.

Security and Vulnerability Assessments

Security and Vulnerability Assessments provide ICS users and businesses with a well-balanced cost versus value security evaluation mechanism. There are both "theoretical" and "physical" methodologies that can be used—both discussed shortly. The premise of this type of assessment is to look at the entire solution for the system under consideration. This means that for each ICS system and subsystem, all servers, workstations, and controllers are included. Third-party equipment, such as field instruments, analytical systems, PLCs, RTUs, IEDs, and custom application servers, are included. Semitrusted or demilitarized zones are considered in the assessment, as well as all communication to trusted and untrusted zones.

The active and passive network infrastructure is included covering switches, routers, firewalls, wiring closets, patch panels, and fiber-optic routing. Remote access is included (if applicable) covering not only access from users external to the facility (e.g. remote engineering access, remote vendor support) but also communications that originate outside the local control zone(s) but still may remain within the plant perimeter (e.g. engineering access via administration buildings, patch management systems, and security monitoring appliances).

User identification, authentication, authorization, and accounting functionality is also included to help uncover potential weaknesses in identity and authorization management (IAM) systems like Microsoft Active Directory and RADIUS.

It is not practical to perform complete Vulnerability Assessments against 100% of the hosts within an ICS architecture. Vulnerability Assessments therefore tend to focus on a subset of critical nodes. The results typically yield accurate results, because many policies that are deployed within industrial networks apply to all hosts. If you assess one host and find that it is not patched in a timely manner, it is likely that all hosts within that architecture will possess similar vulnerabilities. Another consideration is that there is a large amount of duplication and redundancy within industrial networks, so that assessing a small subset of hosts can actually reflect a large percentage of the composite architecture.

CAUTION

Vulnerability Assessments are performed at the component level, and therefore are designed to identify if known vulnerabilities exist in the target of evaluation. It may be a safe alternative to bypass any tests against online ICS devices (particularly embedded devices like controllers) when a simple review of the vulnerability tool can reveal if it is capable of discovering any vulnerabilities.

ESTABLISHING A TESTING AND ASSESSMENT METHODOLOGY

The challenge in establishing a repeatable methodology for testing and assessing an ICS lies in the lack of any consistent industry guidance. The two primary frameworks discussed that are commonly deployed in IT environments—penetration tests and vulnerability scans—each have positive aspects that should be applied to a credible ICS process. However, there are significant gaps that remain that must be addressed before attempting an online assessment of an operational ICS. The following recommendations are provided to assist in improving these processes to suit the particular needs of an organization.

Tailoring a Methodology for Industrial Networks

It is now time to tailor what we have learned into a specific methodology that can be used to suit the particular system under evaluation; whether it be a distributed control system (DCS) used in a petroleum refinery or petrochemical plant, or a SCADA system used in a wastewater treatment facility. The overall focus of a security test targeting an industrial network needs to cover a broad range of technologies and components. It shall evaluate the security of all ICS perimeters, including not only local area networks, but wireless networks, remote networks connected via remote access methods, modems, and potential “sneaker nets” that typically do not appear on network architecture diagrams.

The information obtained will be used to evaluate the overall network architecture and understand the basic organization of security zones and conduits, how firewalls have been deployed on the conduits between various zones—including the existence of one or more “functional” semitrusted or demilitarized zones. Communication channels (conduits) between ICS field networks, field controllers, and supervisory equipment will be analyzed. The objective will be to look for weaknesses that could allow unauthorized access to the industrial networks.

It is important to include “social” aspects in the evaluation. A great deal can be learned from how the various personnel that interact with the industrial systems use the components to perform their assigned responsibilities. This should include key functional roles, including operational personnel who are ultimately responsible for interacting with the ICS to control the facility, engineering personnel who administer and configure the ICS, and maintenance personnel (including possible vendor support staff) who service and support the ICS.

The idea of understanding thoroughly the system under evaluation cannot be stated enough. The exact definition of a penetration test varies, though it is widely accepted that the goal of any pen test is to "breach security and penetrate the system"—in other words, to successfully exploit a vulnerability or weakness. Failed attempts within an operational industrial network can cause instability, performance issues, or a system crash. These may lead to not merely a denial-of-service condition, but a potentially serious loss-of-view or loss-of-control situation within the ICS that may result in serious impact to the manufacturing process. The general rule is that pen tests should never be performed on an active, online ICS component, but rather limited to offline, lab, or development systems.

CAUTION

It is always important to remember the priorities of an industrial system when performing any online activities on an ICS or industrial network:

- Human health and safety
- Availability of all components on the system
- Integrity (and timeliness) of data communication.

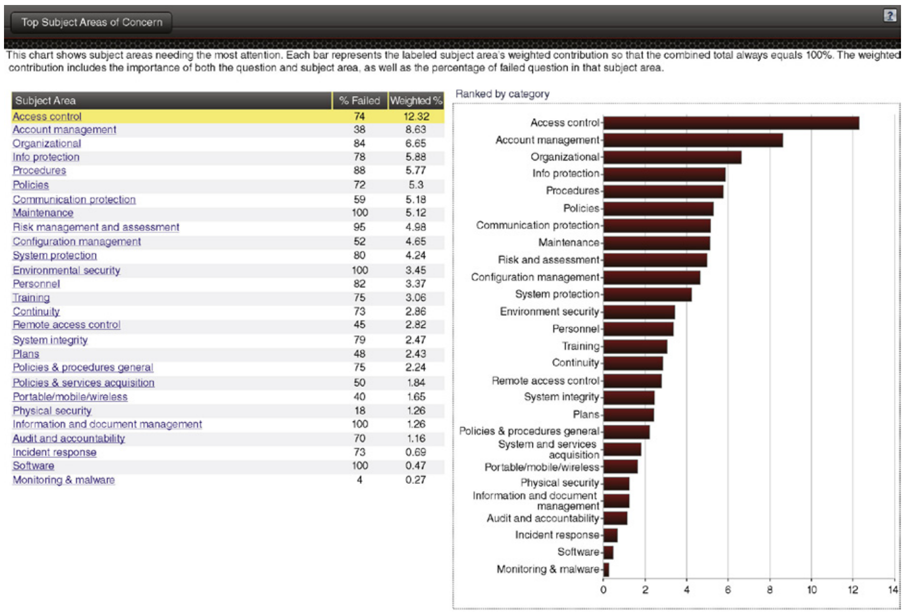
Security assessments and tests should never impact any of these priorities!

Theoretical versus Physical Tests

There may be industrial systems that need a timely assessment; however, the risk to operational integrity is too great to allow even the slightest risk that the tests will impact manufacturing operations. These situations may require a "theoretical" assessment to be performed, which can provide some level of security assurance regarding the system under evaluation without physically contacting any ICS component. This type of assessment is based on a standardized method of completing questionnaires based on a given security baseline in a sort of "interview" format. Accurate results can only be expected when the assessment is conducted as a group exercise and consists of a knowledgeable, cross-functional team representing engineering, operations, maintenance, procurement, HSE (health, safety, environment), and so on.

Theoretical assessments can also be used as an initial mechanism to raise awareness within organizations that are beginning an internal cyber security program. The results of these assessments can be very valuable in understanding major gaps and implementing subsequent, more in-depth analysis.

The US Department of Homeland Security (DHS) Industrial Control System Cyber Emergency Response Team (ICS-CERT) has developed the Cyber Security Evaluation Tool (CSET) as a tool for conducting offline assessments. The CSET provides a step-by-step process of assessing an ICS based on security practices that


Top Subject Areas of Concern

This chart shows subject areas needing the most attention. Each bar represents the labeled subject area's weighted contribution so that the combined total always equals 100%. The weighted contribution includes the importance of both the question and subject area, as well as the percentage of failed question in that subject area.

Subject Area	% Failed	Weighted %
Access control	74	12.32
Account management	38	8.63
Organizational	84	6.65
Info protection	78	5.88
Procedures	88	5.77
Policies	72	5.3
Communication protection	59	5.18
Maintenance	100	5.12
Risk management and assessment	95	4.98
Configuration management	52	4.65
System protection	80	4.24
Environmental security	100	3.45
Personnel	82	3.37
Training	75	3.06
Continuity	73	2.86
Remote access control	45	2.82
System integrity	79	2.47
Plans	48	2.43
Policies & procedures general	75	2.24
Policies & services acquisition	50	1.84
Portable/mobile/wireless	40	1.65
Physical security	18	1.26
Information and document management	100	1.26
Audit and accountability	70	1.16
Incident response	73	0.69
Software	100	0.47
Monitoring & malware	4	0.27

FIGURE 8.4 Sample CSET output.

are compared against a set of recognized industry standards. The answers provided generate output in the form of a prioritized list of recommendations with actionable items to improve the security of the system under evaluation based on the standards baseline. Figure 8.4 illustrates a sample output generated by CSET.

The value of the CSET tool to many organizations is that it provides a high-level of consistency when performing evaluations, since the same questions are asked given the same set of standards requirements. A future release of the CSET tool will also support the ability for the user to input their own question set to assess systems against in-house or custom security practices that may not align exactly with the standards and best practices included with the tool (see Table 8.2 for a list of included standards).

Online versus Offline Physical Tests

Physical tests that utilize actual hardware and software that comprise the components included with the system under evaluation can either be performed on an actual running industrial network that is in operation, or in an environment that is not connected to a physical process and performing real-time control operations. There are advantages and disadvantages of each technique, each of which must be evaluated by an organization against the established test goals prior to commencing the activities.

The most significant benefit of an online test is that it represents a completely functional and operational ICS architecture that includes all the systems, networks, and data integration. Offline environments typically reflect a small subset of the overall architecture, and can omit key components that are a valuable piece of an assessment including complete network topology and connections with third-party systems and applications.

Table 8.2 Standards and Best Practices used in DHS CSET Tool

GENERAL CONTROL SYSTEM STANDARDS
NIST SP800-82 – Guide to Industrial Control Systems Security
NIST SP800-53 – Recommended Security Controls for Federal Information Systems – Appendix I
SECTOR-SPECIFIC STANDARDS
CFATS – Risk-based Performance Standards Guidance 8 (Cyber)
INGAA Control Systems Cyber Security Guidelines for the Natural Gas Pipeline Industry
NEI 0809 Cyber Security Plan for Nuclear Power Reactors
NERC – CIP Reliability Standard CIP-002-009
NISTIR 7628 Guidelines for Smart Grid Cyber Security
NRC – Regulatory guide 5.71 – Cyber Security Programs for Nuclear Facilities
DHS - TSA – Pipeline Security Guidelines
INFORMATION TECHNOLOGY SPECIFIC STANDARDS
NIST SP800-53 – Recommended Security Controls for Federal Information Systems – Appendix I
REQUIREMENTS MODE ONLY STANDARDS
DHS - Catalog of Control System Security – Recommendations for Standards Developers
Council on Cyber Security - Consensus Audit Guidelines (20 Critical Controls)
Dept. of Defense - Instruction 8500.2 – Information Assurance Implementation
ISO/IEC 15408 – Common Criteria for Information Technology Security Evaluation

The reason that offline tests are discussed is that there will be circumstances where it is not possible to perform online tests against critical, high-risk ICS components. In these situations, offline tests can be performed yielding reasonable results from a "component-level" point-of-view. The accuracy of the test results can be greatly improved if an online backup image of the critical component is obtained and then loaded on an offline platform. This will not only allow additional, more rigorous tests, such as possible component testing to target the offline host, but will also evaluate the reliability of the backup-restore utilities (also a vital security control). Table 8.3 provides some additional advantages and disadvantages of online and offline test methods.

Another important characteristic of a security test is understanding the difference between observing the systems with minimal knowledge of the actual system configuration (topology, applications, authentication credentials, etc.) or looking into the system and collecting as much information as possible that may reveal less obvious or latent weaknesses. The primary goal of an ICS security test should be to secure the system as best as possible, rather than only securing those vulnerabilities that

Table 8.3 Online versus Offline Testing Considerations

Online Tests	Offline Tests
Represents realistic network configurations	Can contain realistic configuration of ICS components
Contains volatile ICS components	Can include virtualization technologies
Include complete architecture, including third-party components	Difficult to include all third-party components
Could be used to test susceptibility of network vulnerabilities to attack	Lacks realistic network architecture
Can test less critical third-party components for vulnerabilities	Best at testing ICS components and their vulnerabilities
	Can be used to test ability to exploit vulnerabilities (Ethical Hacking)

Table 8.4 White Box versus Black Box Testing Considerations

White Box	Black Box
Intent of assessment is to identify security vulnerabilities that could lead to an exploit; not ability to exploit	Realistically represents system in way Attacker sees system
Requires Asset Owner to disclose significant information for successful test	Protects Asset Owner intellectual property
Provides most comprehensive look at vulnerabilities and risk	Does not provide complete exposure to risk
Often includes false positives	

may be visible to a potential attacker. A system is considered more resilient to future attacks when a test is conducted in the latter manner. This is the reason the preferred practice for ICS security assessments is to follow a “white box” approach. Table 8.4 provides some of the key differences between these types of tests. The benefits of white box testing over black box will be discussed in more detail in the section on “Vulnerability Identification.”

SYSTEM CHARACTERIZATION

Once the premise of the security test that will be conducted has been defined as “physical” and “online,” the first activity performed is to characterize or identify all physical and logical assets that comprise the system under evaluation. Asset inventory and documentation is difficult, and as a result can often contain gaps. This is why documentation that is obtained prior to the commencement of the test should only act as a starting point, and should always be validated for accuracy. A security test is designed to secure a target system by identifying security weaknesses within the architecture. It is very difficult to assess assets that are not identified or known beforehand!

System characterization and asset identification is best performed using a zone concept. This approach provides the ability to take an architecture and create a zone perimeter, which will be called a "trust boundary" at this time. Once this trust boundary is established, it is then important to delineate all of the external entry points that require penetration of the perimeter. The concepts of zones/conduits and trusted/untrusted relationships is discussed in Chapter 9, "Establishing Zones and Conduits." Figure 8.5 represents the reference architecture of a single zone that will be used to discuss the concepts of trust and entry points.

The reference architecture contains three physical assets within the zone (SCADA HMI, Engineering Workstation and Controller) and one asset on the conduit (Firewall). Entry points from trusted users can also be used as attack vectors from untrusted users or potential attackers. This is why this important first step is to understand all of the mechanisms that are possible to introduce "content" into the assets, as well as those assets that are currently deployed and utilized, to understand the initial attack surface of the architecture. A practical example of an unused or hidden entry point to an asset may be the built-in wireless capabilities (802.11, Bluetooth, etc.) of the SCADA HMI and the Engineering Workstation that the platform possesses but may not be currently in use. Table 8.5 summarizes these entry points, as well as the data or "content" that is typically introduced via these mechanisms.

A different way of looking at security is to consider the relationship between the asset and the controls that are deployed to protect the asset. In the vast majority of cases, security controls are specified and implemented to protect specific "logical" assets rather than "physical" ones. As an example, consider the installation of anti-virus software (AVS) on a host computer. The primary security objective of the

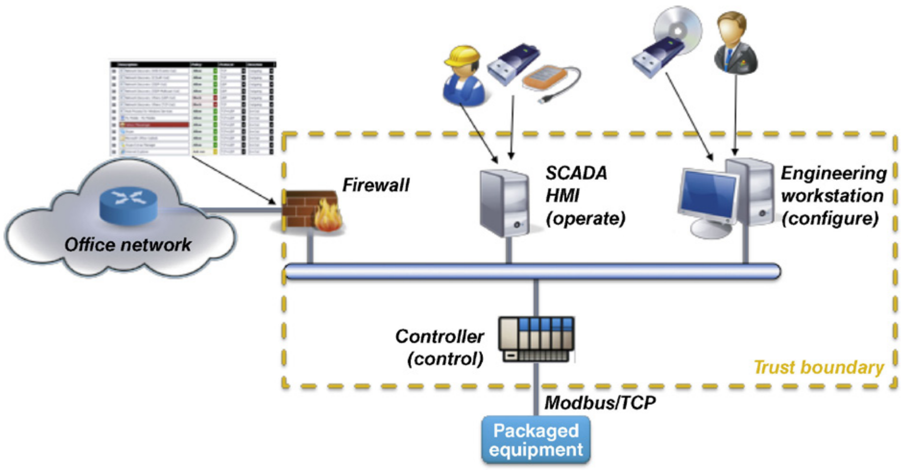


FIGURE 8.5 Trust boundary and entry points.

Table 8.5 System Characterization – Identifying Entry Points

Entry Point Name	Entry Point Description	Data Flows Associated with Entry Point	Assets Associated with Entry Point
Firewall	Internal Firewall between Office and Control Networks	AD Authentication (LDAP)	Engineering Workstation
		AD Authentication (LDAP)	Operator Workstation
		File Sharing (SMB)	Engineering Workstation
		File Sharing (SMB)	Operator Workstation
		Historical Data (OPC)	Operator Workstation
Modbus Port on Controller	Modbus Port on Embedded Controller to Packaged Equipment	Modbus/TCP	Controller
Keyboard	Keyboard on EWS	Keyboard Input	Engineering Workstation
Keyboard	Keyboard on OWS	Keyboard Input	Operator Workstation
CD/DVD Drive	CD/DVD Drive on EWS	Software, Data Files	Engineering Workstation
CD/DVD Drive	CD/DVD Drive on OWS	Software, Data Files	Operator Workstation
USB Port	USB Port on EWS	Software, Data Files, Backup	Engineering Workstation
USB Port	USB Port on OWS	Software, Data Files, Backup	Operator Workstation
Wireless	WLAN/Bluetooth on EWS	Software, Data Files	Engineering Workstation
Wireless	WLAN/Bluetooth on OWS	Software, Data Files	Operator Workstation

AVS is to prevent the unauthorized execution of malicious code on the platform. The reason is that malicious code is often designed to target the information contained on a computer, such as credentials and local files (the "logical" assets of the host), rather than the computer itself (the "physical" asset in this case).

There are always exceptions to any general rule. The Shamoon attack of 2012 was able to render local hard disks inoperable by corrupting the master boot record (MBR) that left the computer inoperable.[10] The fact remains that most security controls are protecting logical assets within an architecture, and for this reason, it is important to have an understanding of the logical assets that are contained within a particular physical asset. Table 8.6 provides some examples of common logical assets within industrial networks.

Table 8.6 System Characterization – Identifying Logical Assets

Physical Asset	Logical Asset	Threat Event (Threat to Logical Asset)
Firewall	Firmware	Modify Firmware to change behavior of Firewall
	Management Port	Modify Firmware, Modify Configuration, Elevation of Privilege
	Identification & Authentication Services	Elevation of Privilege
	Log Files	Modify Logs to remove Audit Trail
	Communication Interfaces	Denial-of-Service
	Configuration	Modify Configuration to change the behavior or the Firewall
Network	Switch Ports	DoS, Laptop connection Injects Malware, Elevation of Privilege
	Switch Configuration	Modify Switch Configuration to change behavior of Switch
Controller	Static Control Logic Configuration	Modify Configuration to change the behavior of Controller
	Control Logic Algorithm Library	Modify Control Algorithms to change the behavior of the Control Algorithms
	Dynamic Control Data	Modify Dynamic Data to change the results of Control Algorithms
	I/O Database	Modify I/O Data to change the results of Control Algorithms
	Controller Firmware	Modify the Controller Firmware to change the behavior of the Controller
	Modbus Interface	DoS, Send Elicit Instructions
	Ethernet Interface	DoS, Inject Code (malware), Send Elicit Instructions
Engineering Workstation	Windows OS	DoS, Elevation of Privilege
	Stored Files	Copy Sensitive Information, Modify or Delete Files
	Engineering & Configuration Apps	Modify stored Configurations, Send Commands to Controller, Modify online Configuration
	DLL’s	Man-in-the-Middle attack
	Ethernet Interface	DoS, Inject Code (malware), Gain Remote Access
	Keyboard	DoS, Elevation of Privilege, Modify Anything
	CD/DVD Drive	Inject Code (malware), Copy Sensitive Information
	USB Interface	Inject Code (malware), Copy Sensitive Information
	Modem	DoS, Inject Code (malware), Gain Remote Access

Table 8.6 System Characterization – Identifying Logical Assets *(cont.)*

Physical Asset	Logical Asset	Threat Event (Threat to Logical Asset)
Operator Workstation	Windows OS	DoS, Elevation of Privilege
	Stored Files	Copy Sensitive Information, Modify or Delete Files
	HMI Application	Send Commands to Controller
	DLL's	Man-in-the-Middle attack
	Ethernet Interface	DoS, Inject Code (malware), Gain Remote Access
	Keyboard	DoS, Elevation of Privilege, Modify Anything
	CD/DVD Drive	Inject Code (malware), Copy Sensitive Information
	USB Interface	Inject Code (malware), Copy Sensitive Information
	Modem	DoS, Inject Code (malware), Gain Remote Access

DATA COLLECTION

Documentation is validated and the system assets are characterized or identified via a variety of data collection methods. As an assessor becomes more familiar with the system(s) under evaluation, it will become easier to rapidly identify the critical physical and logical assets that will form the basis of a Hardware and Software Inventory. Online sources are a vital part of this activity, as this will identify all devices connected to the industrial network (and DMZs if included in the test). This will not only validate and update existing documentation, but can uncover hidden and undocumented devices and appliances that could represent significant risk to the industrial architecture. Online data collection will provide the ability to accurately identify all open communication ports and running applications/services on a particular device. This information will later be used to evaluate potential attack vectors within the system.

There are a variety of scanning tools that exist, both open-sourced and commercial, to assist with this activity. Scanning tools can however have catastrophic effects on some ICS components, and should never be used without extensive, offline testing, or without approval from the business owners. The most dangerous tools tend to be "active," which are highly automated and typically inject data onto the network. These active scanners are typically unfriendly to ICS components and are recommended for use only in offline environments or during manufacturing outages until thoroughly tested. Passive test tools can be used, which are less risky and pose minimal threat of impact to the ICS. These tools will be discussed later in "Scanning of Industrial Networks."

TIP

There have been numerous documented incidents where active, automated tools have been used on industrial networks and have resulted in ICS shutdowns. The business impact of such a shutdown may not only damage the credibility of the individual(s) performing the test, but also that of the security program itself, and seriously undermine the program's business value.

There is an extensive amount of information that can be obtained from a pool of offline resources. This includes technical documentation for the various components comprising the ICS, such as vendor manuals, project specific drawings, specifications, build books, and maintenance records. System configuration data can provide extensive information regarding hardware configuration, software applications, versions, firmware, and so on. These configuration data are readily available for most ICS platforms, network appliances, third-party appliances, and corporate interfaces, and should be requested in advance of the physical test start. Prior assessments, whether internal or external, can provide a valuable source of information that may not be appropriate for standard system documentation, but is vital to improving the outcome of any security testing.

SCANNING OF INDUSTRIAL NETWORKS

Device Scanners

There are different types of "scanners" that can be used depending on the purpose of the scan. The most basic types are designed to identify devices, and may offer additional capabilities that include the identification of specific applications and communication services available on these hosts. The Network Mapper or `nmap` is one of the most popular device scanners used, and is available for most common operating systems. It has evolved in the open-sourced community and includes capabilities of host discovery, host service detection, operating system detection, evasion and spoofing capabilities, and the ability to execute customized code via the Nmap Scripting Engine (NSE).

Basic device identification tools like `ping` are built-in to most commercial operating systems; however, there are limited capabilities of executing this command across a large number of possible hosts. The `ping` command utilizes the Internet Control Message Protocol (ICMP) to generate requests to target devices. The results of this application can be inaccurate, as many hosts now block ICMP messages via host-based applications. Security appliances rarely forward ICMP messages, making this application ineffective when used against a typical ICS zone-based architecture.

Nearly all devices depend on the Address Resolution Protocol (ARP) to translate Layer 2 hardware addresses (MAC) to Layer 3 IP addresses. This type of traffic is common and continuous in all networks, as it is the primary mechanism used for devices to establish and maintain communication within the same LAN subnet. There are tools based on ARP, including `arping` and `arp-scan`, that can be effectively used to identify hosts on a network, and in some cases, can even identify hosts across security perimeters protected by firewalls.

The `nmap` tool does all of its data collection via network-based, external packet injection and analysis. This means that it sends a large amount of traffic toward a host and analyzes the responses. This tool may be a realistic representation of how an attacker views the hosts on the network, but is in fact a very poor tool when used as a method of identifying system assets. The concept of a white box test suggests that tools should be used to characterize "actual" features of a system and not just what is "identifiable." The Network Statistics or `netstat` tool is another command-line feature that is available on most operating systems. The parameters may change slightly between operating systems, but the usefulness of this command comes from its ability to display a number of host-based network features including "active" and "listening" network connections, application and associated service/communication port mapping, and routing tables. This tool can become a valuable asset when trying to identify the applications and services that are running on a particular host (as required by many regulations and standards including NERC CIP). It has the ability to identify active sessions with remote hosts, and the services used by these hosts—vital information in establishing a network data flow mapping. This is a command-line tool and therefore does not inject packets on the network that could compromise time-sensitive network communication between ICS components making this a "friendly" and "passive" tool.

Vulnerability Scanners

Vulnerability scanners form the next major type of commonly used network security scanners. There are a variety of both open-sourced (e.g. OpenVAS) and commercial (e.g. Tenable Nessus, Qualys Guard, Rapid7 Nexpose, Core Impact, SAINT scanner) products available. These applications are designed to identify vulnerabilities that may exist within a target by comparing these hosts against a database of known vulnerabilities. The ability to detect vulnerabilities can vary widely from product to product, as the vulnerability databases are managed by the application and not a common repository.

It was mentioned earlier that the number of vulnerabilities disclosed targeting ICS and industrial network components is growing. It is essential that the tool chosen for vulnerability assessment within the industrial networks is capable of identifying vulnerabilities for the targeted hosts. It would make little sense to deploy a tool that was not able to recognize ICS components when conducting a vulnerability scan on an industrial network.

Vulnerability scanners often include features that allow them to perform device scanning that occurs in advance of service and application identification that comprises the actual vulnerability analysis. These tools are often capable of accepting input from other dedicated device scanners in order to improve the efficiency of the vulnerability scans. More detailed information on vulnerability scanners will be provided later in "Vulnerability Identification."

Traffic Scanners

Traffic scanners form another class of scanning tool that is commonly used in security testing activities. These tools are designed to collect raw network packets and provide them for subsequent analysis that may include host identification, data flows, and firewall rule set creation. The basic form of traffic scanner is the `tcpdump` (formerly `ettercap`) for Linux and `windump` for Windows. These command-line tools are designed primarily for the purpose of capturing and saving network traffic.

Wireshark is an application that is commonly used for analysis of network traffic in the form of pcap files. Though Wireshark can be used for raw packet collection, it is not recommended to use this application for this purpose due to both security and memory performance issues. Wireshark provides the ability to filter traffic based on various criteria, create conversation lists for a number of network protocols, and extract payloads that may exist within data streams.

Wireshark utilizes protocol "dissectors" so that the protocols used in the various Open Systems Interconnection (OSI) layers can be dissected and presented before passing them to the next layer, allowing specific protocol details at each layer to be visualized in the Wireshark GUI. A sample of some of the built-in Wireshark dissectors for industrial protocols is shown in Table 8.7.

Table 8.7 Wireshark Industrial Protocol Dissectors

Protocol Description
Building Automation Control Networks
Bristol Standard Asynchronous Protocol
Common Industrial Protocol
Component Network over IP
Controller Area Network
ELCOM Communication Protocol
EtherCAT
Ethernet for Control Automation Technology
Ethernet POWERLINK
EtherNet/IP
FOUNDATION Fieldbus
GOOSE
HART over IP
IEC 60870-5-104
IEEE C37.118 Synchrophasor Protocol
Kingfisher RTU
Modbus
OMRON FINS
OPC Unified Architecture
PROFINET
SERCOS
TwinCAT
ZigBee

Microsoft has developed the Microsoft Message Analyzer, which is the successor to Microsoft Network Monitor. This application provides many of the capture and visualization features of Wireshark. As the name implies, this tool is more than a network traffic analyzer, but rather a multifunction tool that allows event logs and text logs to be imported and analyzed, along with trace files that can be collected locally or imported from other tools like Wireshark and `tcpdump`. The Microsoft Network Monitor does not support the dissection of industrial protocols like Wireshark, but has features that make it a valuable tool in any security testers application toolkit.

Live Host Identification

Several examples are provided below to illustrate how some of the various tools can be used to perform live host identification on an industrial network. All of these examples are run on a Linux host using the root account. These commands should always be practiced and tested in an offline environment prior to executing on an operation system. Many of these tools contain numerous options where simple typographic errors can have drastic impact on the execution of the tool.

"Quiet" / "Friendly" Scanning Techniques

The first example demonstrates how `arping` is used to send a single ARP request (`-c 1`) to one target (`192.168.1.1`) via a specific network interface (`-i eth0`):

```
# arping -i eth0 -c 1 192.168.1.1
```

The next example shows how the `arp-scan` command can be used to scan the entire subnet (`-l`) that corresponds to the configuration of a particular network interface (`-I eth0`) [notice that this command uses a capital "I" were the previous used a lowercase "i"], sending requests every 1000 ms (`-i 1000`) and providing verbose output (`-v`):

```
# arp-scan -I eth0 -v -l -i 1000
```

The **arp-scan** command can also specifically designate a network to scan (`192.168.1.0/24`) using CIDR notation and does not necessarily have to be configured on the local network interface (`-I eth0`). This makes this tool very useful to scan general network ranges without actually receiving an address on the target network.

```
# arp-scan -I eth0 -v -i 1000 192.168.1.0/24
```

The next example uses the `tcpdump` command to initiate a packet capture that does not attempt to resolve addresses to hostname (`-n`) using a specific network interface (`-i eth0`) that writes the output to a file (`-w out.pcap`) and only includes traffic with a specific IP destination address (`dst 192.168.1.1`) and communication port (`and port 502`):

```
# tcpdump -n -i eth0 -w out.pcap dst 192.168.1.1 and port 502
```

Can you identify what is wrong with the previous example? What traffic does it actually capture? Since the command only captures traffic with a specific destination address, it will never see the return responses that would consist of packets that now have the same IP address as the source (src). A modified example that captures both sides of the communication includes a new filter (`dst x or src x`) and looks like this:

```
# tcpdump -n -i eth0 -w out.pcap dst 192.168.1.1 or src 192.168.1.1
```

Potentially "Noisy"/"Dangerous" Scanning Techniques

There may be times when the use of more active tools is required for a security test. This may include offline tests, tests that occur during production outages, or after testing and understanding the predicted response of the target device. The first example uses the `nmap` command to perform a ping sweep (`-sn`) on a single subnet (`192.168.1.0/24`):

```
# nmap -sn 192.168.1.0/24
```

Additional options can be added to the `nmap` command to probe a target using a SYN scan (`-sS`) omitting name resolution (`-n`) and setting the timing of the scan (`-T3`) that provides service version identification (`-sV`) and operating system identification (`-O`) using a range of TCP ports (`1-10240`) against a subnet range of targets (`192.168.1.0/24`) and saving the output to a file in XML format (`-oX out.xml`):

```
# nmap -sS -n -T3 -sV -O -p 1-10240 -oX out.xml 192.168.1.0/24
```

A very powerful command-line tool to create and send specific packets on to the network is the `hping3` command. This is a Linux tool that can be very useful in testing firewalls and the performance of the rule sets against various criteria. This tool is classified as noisy since it does inject traffic onto the network, so it should be checked for compatibility with the target hosts before deploying in an operational network.

The first example sends a single packet that only contains the TCP header flag SYN set (`-S`) to a single target (`192.168.1.1`) using the port for Modbus/TCP (`-p 502`):

```
# hping3 -S -p 502 192.168.1.1
```

This next example performs a function similar to the `nmap –sS` option described earlier, by scanning a range of ports (`--scan 1-10000`) on a single target (`192.168.1.1`). The second example redirects the output into the "grep" application and only displays lines that contain the string "S..A" signifying that the response contained a packet with the TCP header SYN + ACK flags set:

```
# hping3 --scan 1-10000 192.168.1.1
# hping3 -scan 1-10000 192.168.1.1 | grep S..A
```

Port Mirroring and Span Ports

Most networks today are built using switches that provide a single collision domain between the host and the switch that it is connected. The switch is then responsible for maintaining a local hardware address (MAC) table and forwarding traffic as needed to the access ports that contain the desired MAC destination address. This means that the only types of traffic that can be monitored from a computer's network interface is the traffic specifically destined for the computer and local network broadcast and multicast traffic. It is necessary to enable a feature called "port mirroring" or "span ports" on the adjacent switch that will forward all network traffic within the switch to not only the desired target's access port, but also the mirrored port. This modification requires privileged access to the switch, and will forward a significant amount of traffic to the mirrored port. Attention should be paid to disabling this feature when it is no longer needed.

The next example illustrates the steps to create a span port on a Cisco Catalyst 2960 switch that mirrors traffic from Fast Ethernet ports 1–23 to port 24:

```
C2960# configure terminal
C2960(config)# monitor session 1 source interface range fe 0/1 - 23
C2960(config)# monitor session 1 destination interface fe 0/24
```

This technique allows a security tester to connect to each switch and collect a representation of the network traffic that exists locally within or transfers via uplinks through the switch. This is often a beneficial step in a security test that can provide a snapshot of actual network traffic that is collected passively and can be used for additional analysis and reporting.

Most industrial networks will consist of a number of network switches that may be configured in a redundant manner. This will require that samples be collected from all switches and then consolidated to create a single snapshot of the complete industrial network. The `mergecap` utility installed with Wireshark provides the capability to take multiple libpcap-formatted files and merge them into a single file for subsequent analysis:

```
# mergecap -w outfile.pcap infile1.pcap infile2.pcap … infilen.pcap
```

There is always going to be some level of risk when performing scans of industrial networks that actively inject new traffic and target network-based hosts. Table 8.8 has been provided as a final reminder that actions typically performed on IT networks (where the primary targets are Windows-based hosts) are different from those provided on OT or industrial networks. Many of the results from common IT actions can be obtained using alternative techniques.

CAUTION

Always remember than any tool used in an online ICS environment should be thoroughly tested for potential impact prior to use in a production environment. The procedures for any online test should also include an action plan that should address the steps to be taken in the event of an unexpected consequence occurring during the test.

Table 8.8 Minimizing the Risk of Network Scans to ICS

Target	Typical IT Action	Suggested ICS Action
Hosts, Nodes, Networks	Ping Sweep	• Visually example router configuration files • Print local route and arp table • Perform physical verification • Conduct passive network listening • Use of IDS on network • Specify a subset of targets to programmatically scan
Services	Port Scan	• Do local port verification (netstat) • Scan a duplicate, development, or test system on a non-production network
Vulnerabilities within a Service	Vulnerability Scan	• Perform local banner grabbing with version lookup in CVE • Scan a duplicate, development, or test system on a non-production network

Command Line Tools

Up to this point, the majority of the tools discussed were run from an "assessment console" or other computer that traditionally is loaded with a hardened version of Linux and is connected to the industrial network. Many of the tools mentioned were friendly or minimally invasive to most ICS components; however, they all still injected new traffic onto the network. This may not be allowed in some environments because there is even the slightest chance that these actions could negatively impact the availability and performance of the ICS. There are alternatives that will allow the same, if not more data to be collected, yet via local interaction with the keyboard and monitor rather than remotely over the network. These tools are installed on most systems, allowing a robust assessment to be conducted with existing equipment, and can significantly improve the ability to thoroughly analyze Windows hosts. These tools also support the ability to write the output to editable files that can then be merged and combined with other data for easy analysis and reporting.

There are a variety of options available, most depending on the version of operating system installed on the target. For the purposes of this section, these tools will be focused on a Windows-based ICS host platform and the tools discussed will be those available as early as Windows XP Professional and Windows Server 2003.

TIP

Every tester needs to have a solid library of reference texts that can be called upon to assist in performing ICS security tests. The Windows Command-Line Administrator's Pocket Consultant[11] provides one of the most comprehensive reference guides to Windows command-line utilities that are often forgotten in the world of the Windows GUI!

`ipconfig` is a common Windows command-line tool that not only displays all current network configuration values, but can also be used to refresh Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Information provided by `ipconfig` includes

- Hardware (MAC) address
- IP address (IPv4 and IPv6)
- Subnet mask
- Default gateway
- DHCP server
- DNS server
- NetBIOS over TCP/IP enabled/disabled.

The next example uses the `/all` option to provide a complete report of network settings. The output is also redirected (>) to a text file (`host.ipconfig.text`) for collection and use.

```
C: > ipconfig / all > host.ipconfig.text
```

The Network Statistics (`netstat`) command is the authoritative method to determine what applications are running on a computer and how they map to associated communication ports and service names. It displays sessions that are both local and remote to the host, as well as active connections. Information provided by `netstat` includes

- Active TCP connections
- Ports on which the computer listening
- Ethernet statistics
- IP routing table
- IPv4 and IPv6 statistics.

There are several parameters that can be supplied with the command. The following example requests all active connections (`-a`) and the associated TCP/UDP ports in numerical form (`-n`) on which the computer is listening, along with the executable associated with the connection (`-b`). The output has again been redirected (>) to a text file (`host.netstat.text`). This command requires elevated privileges when User Account Control (UAC) is enabled on Windows. The second example adds an additional parameter limiting the information to the TCP protocol (`-p TCP`). The third and fourth examples show that the output can be piped (|) into a second utility (`findstr`) that can parse the output similar to the Linux "grep" command and only provide those connections that are active ("`ESTABLISHED`") or waiting ("`LISTENING`"). As with most commands, additional details can be found by adding `/?` after the command with no parameters.

```
C: > netstat -anb > host.netstat.text
C: > netstat -anbp TCP > host.netstat.text
C: > netstat -anb | findstr "ESTABLISHED"
C: > netstat -anb | findstr "LISTENING"
```

The Network Statistics commands may not return the name of an executable associated with a running service when running on some platforms, but rather the Process Identification (PID) for the service. This requires the `tasklist` command to be executed providing a list of all running applications and services with their associated PID.

```
C: > tasklist > host.tasklist.text
```

It is valuable during a security test to collect detailed configuration information about each host included in the activity. The System Information (`systeminfo`) command provides valuable information that supports the Hardware and Software Inventory activities (shown later), as well as

- Operating system configuration
- Security information
- Product identification numbers
- Hardware properties (RAM, disk space, network interface cards).

The next example shows the `systeminfo` command with the output redirected (>) to a text file (`host.systeminfo.text`) for retention.

```
C: > systeminfo > host.systeminfo.text
```

The Window Management Instrumentation Command-line (`wmic`) utility provides a powerful set of systems management features that can be executed independently, interactively, or as part of a batch file. Access to the Windows Management Instrumentation (WMI) system allows comprehensive system information to be extracted and stored in a variety of formats that support retention and analysis (CSV, HTML, XML, text, etc.). The next example uses `wmic` to query the system and provides a listing of all installed software (`product get`) output in HTML format (`/format:htable`) and saved as a file (`/output:”host.products.html”`).

```
C: > wmic /output:”host.products.html” product get /format:htable
```

Some other examples of how `wmic` can be used include local group management (`group`), network connections (`netuse`), quick fix engineering (`qfe`), service application management (`service`), local shared resource management (`share`), and local user account management (`useraccount`).

```
C: > wmic /output:”host.group.html” group list full /format:htable
C: > wmic /output:”host.netuse.html” netuse list full /format:htable
C: > wmic /output:”host.qfe.html” qfe list full /format:htable
C: > wmic /output:”host.service.html” service list full /format:htable
C: > wmic /output:”host.share.html” share list full /format:htable
C: > wmic /output:”host.useraccount.html” useraccount list full /
format:htable
```

A summary of the `wmic` command-line tool is shown here.

```
C:>wmic /?
[global switches] <command>

The following global switches are available:
/NAMESPACE           Path for the namespace the alias operate against.
/ROLE                Path for the role containing the alias definitions.
/NODE                Servers the alias will operate against.
/IMPLEVEL            Client impersonation level.
/AUTHLEVEL           Client authentication level.
/LOCALE              Language id the client should use.
/PRIVILEGES          Enable or disable all privileges.
/TRACE               Outputs debugging information to stderr.
/RECORD              Logs all input commands and output.
/INTERACTIVE         Sets or resets the interactive mode.
/FAILFAST            Sets or resets the FailFast mode.
/USER                User to be used during the session.
/PASSWORD            Password to be used for session login.
/OUTPUT              Specifies the mode for output redirection.
/APPEND              Specifies the mode for output redirection.
/AGGREGATE           Sets or resets aggregate mode.
/AUTHORITY           Specifies the <authority type> for the connection.

/?[:<BRIEF|FULL>]    Usage information.
For more information on a specific global switch, type: switch-name /?
The following alias/es are available in the current role:

ALIAS               - Access to the aliases available on the local system
BASEBOARD           - Base board (also known as a motherboard) management
BIOS                 - Basic input/output services (BIOS) management
BOOTCONFIG           - Boot configuration management
CDROM                - CD-ROM management
COMPUTERSYSTEM       - Computer system management
CPU                  - CPU management
CSPRODUCT            - Computer system product information from SMBIOS
DATAFILE             - DataFile Management
DCOMAPP              - DCOM Application management
DESKTOP              - User's Desktop management
DESKTOPMONITOR       - Desktop Monitor management
DEVICEMEMORYADDRESS - Device memory addresses management
DISKDRIVE            - Physical disk drive management
DISKQUOTA            - Disk space usage for NTFS volumes
DMACHANNEL           - Direct memory access (DMA) channel management
ENVIRONMENT          - System environment settings management
FSDIR                - Filesystem directory entry management
GROUP                - Group account management
IDECONTROLLER        - IDE Controller management
IRQ                  - Interrupt request line (IRQ) management
JOB                  - Provides access to the jobs scheduled using
schedule service
LOADORDER            - Mgmt of system services that define execution
dependencies
LOGICALDISK          - Local storage device management
LOGON                - LOGON Sessions
```

```
MEMCACHE                - Cache memory management
MEMLOGICAL              - System memory management (config layout & avail
of mem)
MEMPHYSICAL             - Computer system's physical memory management
NETCLIENT               - Network Client management
NETLOGIN                - Network login information (of a particular user)
management
NETPROTOCOL             - Protocols (and their network characteristics)
management
NETUSE                  - Active network connection management
NIC                     - Network Interface Controller (NIC) management
NICCONFIG               - Network adapter management
NTDOMAIN                - NT Domain management
NTEVENT                 - Entries in the NT Event Log
NTEVENTLOG              - NT eventlog file management
ONBOARDDEVICE           - Mgmt of common adapter devices built into the
motherboard
OS                      - Installed Operating System/s management
PAGEFILE                - Virtual memory file swapping management
PAGEFILESET             - Page file settings management
PARTITION               - Management of partitioned areas of a physical disk
PORT                    - I/O port management
PORTCONNECTOR           - Physical connection ports management
PRINTER                 - Printer device management
PRINTERCONFIG           - Printer device configuration management
PRINTJOB                - Print job management
PROCESS                 - Process management
PRODUCT                 - Installation package task management
QFE                     - Quick Fix Engineering
QUOTASETTING            - Setting information for disk quotas on a volume
RECOVEROS              - Info that will be gathered from mem when the os fails
REGISTRY                - Computer system registry management
SCSICONTROLLER          - SCSI Controller management
SERVER                  - Server information management
SERVICE                 - Service application management
SHARE                   - Shared resource management
SOFTWAREELEMENT         - Mgmt of elements of a software product installed
on a system
SOFTWAREFEATURE         - Management of software product subsets of
SoftwareElement
SOUNDDEV                - Sound Device management
STARTUP                 - Mgmt of commands that run automatically when users
log on
SYSACCOUNT              - System account management
SYSDRIVER               - Management of the system driver for a base service
SYSTEMENCLOSURE         - Physical system enclosure management
SYSTEMSLOT              - Mgmt of physical connection points (ports, slots,
periph)
TAPEDRIVE               - Tape drive management
TEMPERATURE             - Data management of a temperature sensor
TIMEZONE                - Time zone data management
UPS                     - Uninterruptible power supply (UPS) management
USERACCOUNT             - User account management
VOLTAGE                - Voltage sensor (electronic voltmeter) data management
```

```
VOLUMEQUOTASETTING  - Associates disk quota setting with specific disk
volume
WMISET              - WMI service operational parameters management

For more information on a specific alias, type: alias /?

CLASS     - Escapes to full WMI schema.
PATH      - Escapes to full WMI object paths.
CONTEXT   - Displays the state of all the global switches.
QUIT/EXIT - Exits the program.

For more information on CLASS/PATH/CONTEXT, type: (CLASS | PATH |
CONTEXT) /?
```

Hardware and Software Inventory

The command-line tools that were just discussed form the basis of the toolset that can be used to create a Hardware and Software Inventory. These inventories are a vital first step in any security program that helps to ensure accurate documentation of the industrial network and its connected equipment, as well as a quick reference that can be used when security vulnerabilities are published or software updates are available. The development of these inventories may be one of the most valuable deliverables from a physical security test. The steps to developing these inventories are outlined as follows:

1. Use `arp-scan` to identify all network-connected hosts. This command must be run on each Layer 3 broadcast domain or subnet. This can also be accomplished in a passive manner by obtaining a consolidated network capture file obtained using `tcpdump` and importing this into Wireshark. Wireshark contains several *Statistics* features, including the ability to display *Endpoints*. This list represents all devices that are actively communicating on the network. This method does not identify nodes that were not communicating on the network when the capture files were collected.
2. Confirm that the identified hosts are authorized for the industrial network. If not, physically inspect the node and determine appropriate actions. Update the system architecture drawings with any newly discovered information.
3. Collect host platform information for each network-connected device. This should include base hardware and operating system information, network configuration details, BIOS revisions, firmware details, and so on. This can be obtained using the `systeminfo` command, or via a third-party Simple Network Management Protocol (SNMP) application. For non-Windows based devices, this typically requires specific, manual activities depending on the device. Some may offer web services that display information via a standard web browser (many PLC vendors offer these web pages as standard features), while others may require the engineering or maintenance tools for the device to be used to collect this information.
4. Collect application information for each network-connected device. This should include application vendor, name, revision, installed patches, and anything else

that characterizes what and how the application has been installed on the target. This can be obtained using the `wmic` command with the `product get` option.

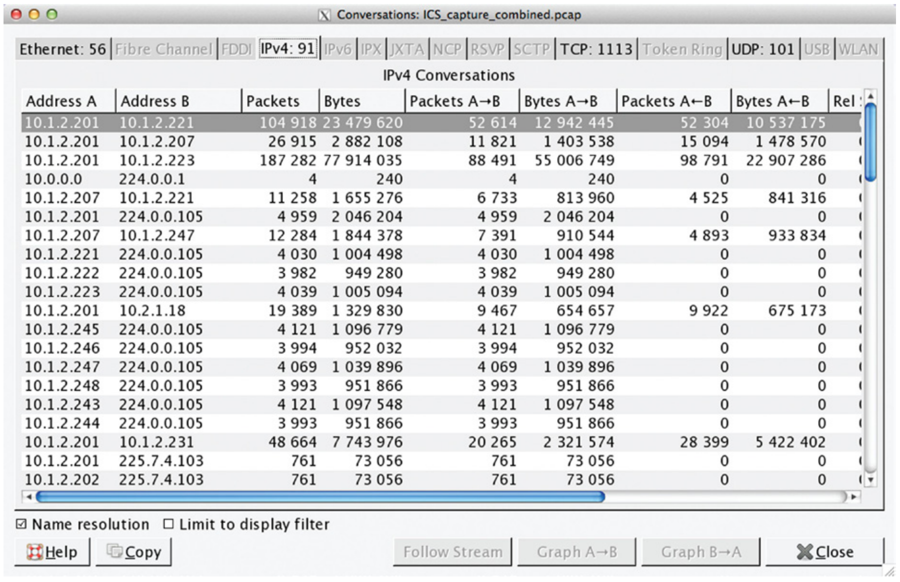
5. Consolidate this information into a spreadsheet or portable database, depending on size. The data provided are sensitive in nature, and as such these documents should be appropriately classified and controlled per local policy.

Data Flow Analysis

It is not uncommon that asset owners, and sometimes ICS vendors, do not completely understand the underlying communications and associated data flow that exist between hosts that comprise an ICS. Many are unclear of the value of such an exercise, and therefore do not put a priority on its creation. It is important as systems are migrated from previous "flat" architectures to those that are segmented into various security zones that the communication channels that exist between these zones are documented. If they are not understood, it can become very difficult to manage the security conduits that are used to connect these zones. This is likely the reason that misconfiguration of firewalls occurs—failure to understand the data flow through the firewall, and failure of the suppliers to provide sufficient documentation on data flow requirements.

The steps required to create a data flow diagram are rather simple, and will allow any asset owner, system integrator, or supplier to create this for any system. There are two pieces of data that are required. The first is a snapshot of the network traffic for the system operating under normal conditions. This can be collected as described previously using `tcpdump`. Multiple network capture files may be required, which can be merged into a single file for analysis using the `mergecap` utility.

Wireshark is used to then open the consolidated capture file, and to perform simple analysis of the network via the *Statistics* features using *Conversations*. The output


Conversations: ICS_capture_combined.pcap
Ethernet: 56 | Fibre Channel | FDDI | IPv4: 91 | IPv6 | IPX | JXTA | NCP | RSVP | SCTP | TCP: 1113 | Token Ring | UDP: 101 | USB | WLAN
☑ Name resolution ☐ Limit to display filter
Help
Copy
Follow Stream
Graph A→B
Graph B→A
Close

IPv4 Conversations

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel :
10.1.2.201	10.1.2.221	104 918	23 479 620	52 614	12 942 445	52 304	10 537 175	
10.1.2.201	10.1.2.207	26 915	2 882 108	11 821	1 403 538	15 094	1 478 570	
10.1.2.201	10.1.2.223	187 282	77 914 035	88 491	55 006 749	98 791	22 907 286	
10.0.0.0	224.0.0.1	4	240	4	240	0	0	
10.1.2.207	10.1.2.221	11 258	1 655 276	6 733	813 960	4 525	841 316	
10.1.2.201	224.0.0.105	4 959	2 046 204	4 959	2 046 204	0	0	
10.1.2.207	10.1.2.247	12 284	1 844 378	7 391	910 544	4 893	933 834	
10.1.2.221	224.0.0.105	4 030	1 004 498	4 030	1 004 498	0	0	
10.1.2.222	224.0.0.105	3 982	949 280	3 982	949 280	0	0	
10.1.2.223	224.0.0.105	4 039	1 005 094	4 039	1 005 094	0	0	
10.1.2.201	10.2.1.18	19 389	1 329 830	9 467	654 657	9 922	675 173	
10.1.2.245	224.0.0.105	4 121	1 096 779	4 121	1 096 779	0	0	
10.1.2.246	224.0.0.105	3 994	952 032	3 994	952 032	0	0	
10.1.2.247	224.0.0.105	4 069	1 039 896	4 069	1 039 896	0	0	
10.1.2.248	224.0.0.105	3 993	951 866	3 993	951 866	0	0	
10.1.2.243	224.0.0.105	4 121	1 097 548	4 121	1 097 548	0	0	
10.1.2.244	224.0.0.105	3 993	951 866	3 993	951 866	0	0	
10.1.2.201	10.1.2.231	48 664	7 743 976	20 265	2 321 574	28 399	5 422 402	
10.1.2.201	225.7.4.103	761	73 056	761	73 056	0	0	
10.1.2.202	225.7.4.103	761	73 056	761	73 056	0	0	

FIGURE 8.6 Performing data flow analysis with Wireshark.

shown in Figure 8.6 reflects the host-to-host sessions that were active when the network captures were collected. The TCP tab would then reveal the TCP ports used for these sessions. The output shows that there were 91 active host-to-host sessions that utilized 1113 different TCP port pairs and 101 UDP port pairs. Additional filtering could be used to eliminate the multicast traffic on 224.0.0.0/8 and 225.0.0.0/8 to reduce the pairings even further.

The `netstat` command is also used to develop a mapping of the local host services and what network devices are using these services. The added value of this method is that it will provide some indication of the applications and service names associated with the communication channels that are identified between hosts. The Wireshark method only reveals the TCP and UDP port numbers. A common method is a hybrid of both techniques that provides for the quick creation of an overall diagram using Wireshark, with additional details regarding the communications established using `netstat`.

THREAT IDENTIFICATION

The methodology described in Figure 8.2 continues with the identification of threats covering threat events, threat sources/actors, and threat vectors. This step is likely the most difficult step in the entire process, and for that reason, is commonly omitted. This is because it can be very difficult to describe all aspects of the unmitigated risk that is present for a particular industrial environment. It was described earlier that cyber security controls are applied to logical assets rather than physical assets. The identification of physical and logical assets occurred during the System Characterization phase. These assets must now be mapped to specific threats that can later be assessed as to whether appropriate controls are in place to secure these assets from the identified threats.

Threat mapping can be performed in one of several fashions, including organization by physical asset, by threat source (outsider, insider), or by intent (intentional, unintentional). The easiest method for most learning the process is to first create an organization by physical asset, which is then expanded to logical assets after completing System Characterization. It is now time to consider the threats that face each of these assets. What may be discovered is that what was perceived to be a risk before the process actually represents very little risk. Conversely, the process might also reveal that assets that are often overlooked represent the greatest unmitigated risk to the ICS and therefore should be the highest priority for mitigation through the deployment of appropriate security controls.

THREAT ACTORS/SOURCES

Many develop industrial cyber security programs under the (unqualified) assumption that the greatest Threats Sources exist outside the company and are hostile and malicious in nature. This leads organizations to deploying security controls that are specifically designed to help prevent these threats from compromising the ICS. These threats are real and do face some risk to industrial networks; however, they typically do not represent the greatest risk to the architecture. Table 8.9 provides a list of some of the common threat actors facing IT and OT systems.

Table 8.9 Common Threat Actors/Sources[27]

Adversarial
Outside individual
Inside individual
Trusted insider
Privileged insider
Ad hoc group
Established group
Competitor
Supplier
Partner
Customer
Nation state
Accidental
User
Privileged user
Administrator
Structural
Information technology equipment
Environmental controls
Software
Environmental
Natural disaster (e.g. fire, flood, tsunami)
Man-made disaster (e.g. bombing, overrun)
Unusual natural event (e.g. solar EMP)
Infrastructure failure (e.g. telecommunications, electrical power)

Documented incident reports from several sources confirm that the majority of incidents, and the greatest risk to a protected architecture, are from insiders or trusted partners. Unfortunately, the majority of security controls deployed do very little to protect the ICS from these threats. Consider as an example the On-site Control System Engineer who configures and administers the ICS. The engineer's job is very demanding causing him/her to find ways to improve efficiency and productivity by installing a suite of untested and unqualified applications on his/her engineering workstation. The engineer also knows that the corporate anti-virus software and host-based firewall often interfere with his/her applications, and since he/she is the administrator, he/she disables these features from the workstation. The original malicious payload may have originated from an external source, but it is now an insider (the engineering) who is going to initiate the event. What controls are left to protect the entire system from a cyber event caused by the insertion of his/her infected USB flash drive into his/her engineering workstation that has elevated privileges and global industrial network access? This is how Stuxnet infected its target! This is the reason why an objective risk process is necessary.

This is not a simple exercise, and for that reason, it may be beneficial to begin the threat identification activities by focusing on four different threat sources: intentional (malicious) outsider, intentional (malicious) insider, unintentional outsider, unintentional (accidental) insider.

THREAT VECTORS

The Threat Vector identifies the method by which the threat source will impact the target. This directly corresponds to the Entry Points in the context of the methodology established in this section. The reason for introducing the concept of an Entry Point as a means of identifying Threat Vectors is that it provides a mechanism of looking beyond traditional IT access mechanisms (e.g. USB flash drives and networks) and introduces more of the human factor including the use of policies and procedures. Entry Points are also intentionally identified before diving into the threat identification phase to allow individuals to consider less obvious mechanisms (e.g. an unused wireless LAN adapter).

The establishment of the Trust Boundary provides a vital role of scoping and limiting the potential Entry Points or vectors entering a zone. Consider as an example an industrial network that is connected to the business network via a firewall. The entry point into the ICS in this case is the network connection through the firewall. The business network on the other hand, will have its own set of Entry Points and Threat Vectors that could potentially allow unauthorized access from untrusted zones (i.e. the Internet) to the trusted business zones. This is not in scope when evaluating the Entry Point into the ICS zones. What this has effectively done is consider unauthorized external traffic on the business network the same as authorized local traffic, since the security controls used on the conduit into the ICS (the firewall in this case) must handle all traffic accordingly. This approach provides necessary resilience when unauthorized external actors have masqueraded as potentially trusted insiders.

Table 8.10 provides basic guidance on the selection of possible ICS Entry Points and Threat Vectors.

THREAT EVENTS

The Threat Event represents the details of the attack that would be carried out by a particular Threat Source. When the source is an adversarial one, the Threat Event is typically described in terms of the tactics, techniques, and procedures (TTP) used in the attack. Multiple actors could possibly use a single event, and likewise, a single actor could use multiple events. This is why the first attempt at developing an ICS risk assessment worksheet can quickly become a very complex task; however, there

Table 8.10 Common Threat Vectors

DIRECT
Local area network – Wired
Local area network – Wireless
Personal area network (NFC, Bluetooth)
USB port
SATA/eSATA port
Keyboard / mouse
Monitor / projector
Serial port
Webcam
Electrical supply
Disconnect switch
INDIRECT
Application software (via media)
Configuration terminal (via serial port)
Modem (via serial port, internal card)
Human (via keyboard, webcam)

is a high likelihood of reusability on subsequent assessment exercises. The list that is initially developed could contain numerous events that are later determined to be unrelated or not relevant to the particular system under evaluation. It is best, however to not eliminate any information during the early steps of the exercise.

The "Guide to Conducting Risk Assessments" published by the US National Institute of Standards and Technology provides a comprehensive appendix of Threat Events that can be used in conducting an ICS assessment. Some of the relevant events from this list have been provided in Table 8.11.

IDENTIFICATION OF THREATS DURING SECURITY ASSESSMENTS

It is likely that during a Security Assessment threats will be discovered and will need to be added to the spreadsheet for tracking and measuring risk throughout the exercise. These threats are typically found when analyzing the data that are collected early in the process that could reveal any of the following:

- Infected media discovered from anti-virus logs
- Infected desktop or laptop workstations discovered from Windows Event logs
- Corrupted static data discovered from local disk evaluation
- Data copied to untrusted location discovered from network resource usage
- Accounts not deactivated discovered from local/domain account review
- Stolen credentials discovered when used to access unauthorized hosts
- Overload communications network discovered when reviewing network statistics

Table 8.11 Common Threat Events[28]

Adversarial Threat Events
Perform network reconnaissance/scanning
Perform organizational reconnaissance and surveillance
Craft spear phishing attacks
Create counterfeit/spoof website
Craft counterfeit certifications
Inject malicious components into the supply chain
Deliver malware to organizational systems
Insert subverted individuals into organizations
Exploit physical access to organization facilities
Exploit poorly configured or unauthorized systems exposed to the Internet
Exploit split-tunneling
Exploit multitenancy in a cloud environment
Exploit known vulnerabilities
Exploit recently discovered vulnerabilities
Exploit vulnerabilities using zero-day attacks
Violate isolation in multitenant environment
Compromise software of critical systems
Conduct attacks using unauthorized ports, protocols and services
Conduct attacks levering traffic/data movement allowed across perimeter
Conduct Denial-of-Service (DoS) attack
Conduct physical attack on organization facilities
Conduct physical attack on infrastructure supporting organizational facilities
Conduct session hijacking
Conduct network traffic modification (man-in-the-middle) attack
Conduct social engineering campaign to obtain information
Conduct supply chain attacks
Obtain sensitive information via exfiltration
Cause degradation of services
Cause integrity loss by polluting or corrupting critical data
Obtain unauthorized access
Coordinate a multistate (hopping) attack
Coordinate cyber-attacks using external (outside), internal (insider) and supply chain vectors
Nonadversarial Threat Events
Spill sensitive information
Mishandling of critical information by authorized users

(Continued)

Table 8.11 Common Threat Events *(cont.)*

Nonadversarial Threat Events
Incorrect privilege settings
Communications contention
Fire (Arson)
Resource contention
Introduction of vulnerabilities into software products
Disk error

The tasks associated with Threat Identification will not only improve one's overall awareness of the system, its operation, and the environment that it operates in, but also will provide useful information that can later be combined with identified weaknesses to prioritize the action plan and mitigating controls that will be selected to secure the industrial systems.

VULNERABILITY IDENTIFICATION

The activity of vulnerability identification is the next step in the process, and is the basis for performing a detailed evaluation of the complete ICS as defined by the security test rules of engagement. This activity will combine automated tools, such as vulnerability scanning applications, with manual analysis of data collected throughout the exercise. A vulnerability is not just the presence of unpatched software designed to correct published vulnerabilities, but also the use of unnecessary services and applications that cannot be determined by simply scanning for the presence (or absence) of software. Vulnerabilities may exist in the form of improper authentication, poor credential management, improper access control, and inconsistent document. A rigorous vulnerability assessment looks at all of these and more.

The assessment phase depends a great deal on automated vulnerability scanning software. It also involves the review of relevant application, host, and network configuration files. The implementation of any existing security controls is reviewed and documented for effectiveness, and the overall physical aspects of the ICS are inspected. The idea behind such a thorough process is to attempt to review and discover many of the more common ICS vulnerabilities. Some of the more common ICS vulnerabilities are shown in Table 8.12.

The potential vulnerabilities as shown in Table 8.12 are meant to serve as a form of reminder when performing the actual assessment. The objective is to identify backdoors or "holes" that may exist in the industrial network perimeter. Devices with little or no security features and those that are susceptible to attack need to be identified so that they can be placed in special security zones and secured separately. Networks are reviewed to uncover possible opportunities for communications hijacking

Table 8.12 Common ICS Vulnerabilities

Category	Potential Vulnerabilities
Networks	Poor Physical Security Configuration Errors Poor Configuration Management Inadequate Port Security Use of Vulnerable ICS Protocols Unnecessary Firewall Rules Lack of Intrusion Detection Capabilities
Configuration	Poor Account Management Poor Password Policies Lack of Patch Management Ineffective Anti-Virus / Application Whitelisting
Platforms	Lack of System Hardening Insecure Embedded Applications Untested Third-Party Applications Lack of Patch Management Zero-Days
ICS applications	Poor Code Quality Lack of Authentication Use of Vulnerable ICS Protocols Uncontrolled File Sharing Zero-Days Untested Application Integration Unnecessary Active Directory Replication
Embedded devices	Configuration Errors Poor Configuration Management Lack of Device Hardening Use of Vulnerable ICS Protocols Zero-Days Insufficient Access Control
Policy	Inadequate Security Awareness Social Engineering Susceptibility Inadequate Physical Security Insufficient Access Control

and man-in-the-middle (MitM) attacks. Every network-connected ICS component is assessed to discover improper or nonexistent patching of both software and firmware that could potentially compromise the network. Suppliers can also be included in the assessment to ensure that insecure coding techniques and software development lifecycles do not introduce unnecessary risk.

VULNERABILITY SCANNING

Vulnerability Scanning is the process of methodically reviewing the configuration of a set of hosts by attempting to discover previously identified vulnerabilities that may be present. Automated tools are available, with some of these described earlier under "Vulnerability Scanners." It is also possible to perform this exercise manually if the use of an automated tool against a critical host is not allowed due to the potential for any negative impact to the performance and availability of the host.

Manual Vulnerability Scanning consists of collecting information using some of the command-line tools described earlier, and individually comparing the revision information of the operating system, applications and services against databases of known vulnerabilities. Two of the popular databases of vulnerabilities are the National Vulnerability Database[12] (NVD) hosted by NIST, and the Open-Source Vulnerability Database[13] (OSVDB). There are more than 100,000 vulnerabilities tracked between these two databases, with most vulnerabilities also tracked against a "common enumeration" system known as Common Vulnerabilities and Exposures (CVE).

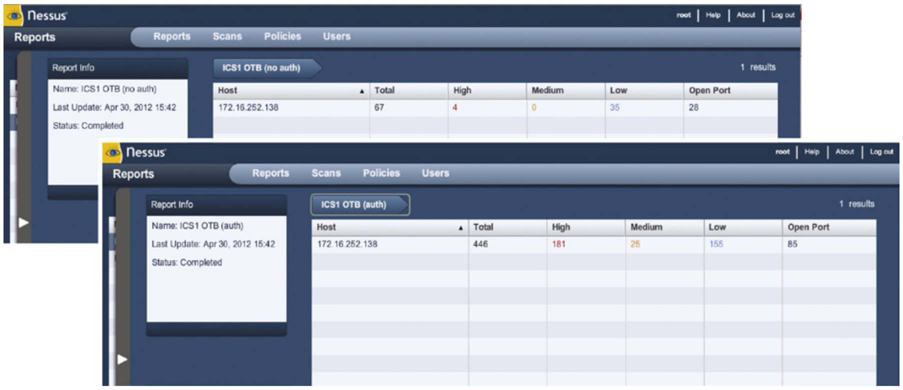
An example of a simple manual vulnerability assessment is detailed here:

1. The `wmic` command is used with the `product get` option to list all of the installed applications running on a Windows 2003 Server host.
2. The SCADA application software is shown as "IGSS32 9.0" with the vendor name "7-Technologies" and a version of 9.0.0.0.
3. Using OSVDB, "igss" is entered in the Quick Search field and several results are returned. Selecting the most recent item, a link is provided to an advisory published by ICS-CERT that confirms that the installed version of software has a published vulnerability.
4. The advisory contains information on how to download and install a software patch from the software provided.

It is apparent that this process can be very time-consuming, and that a great deal of cross-referencing must be performed. The use of automated tools simplifies this process by systematically assessing the target and quickly comparing the information extracted against a local database of documented vulnerabilities. Vulnerability scanning applications depend on external data to maintain a current local database, so the application should be updated before conducting any assessments. It is also recommended to always include the update sequence number or data used when generating a vulnerability report with the security test.

As mentioned earlier, there are several commercial vulnerability scanners available. The important feature to consider when using a particular product—commercial or open-sourced—is the ability to assess the applications that are installed on the target system. Even if there are no application-specific vulnerabilities in the database (as would be the case with many embedded ICS devices), the scanner may still be able to provide useful information regarding active services and potential weaknesses associated with those services.

What is important when using a vulnerability scanning application is to obtain as accurate of results as possible. The way that this is most often performed is via



Report Info
Name: ICS1 OTB (no auth)
Last Update: Apr 30, 2012 15:42
Status: Completed

ICS1 OTB (no auth) 1 results

Host	Total	High	Medium	Low	Open Port
172.16.252.138	67	4	0	35	28

Report Info
Name: ICS1 OTB (auth)
Last Update: Apr 30, 2012 15:42
Status: Completed

ICS1 OTB (auth) 1 results

Host	Total	High	Medium	Low	Open Port
172.16.252.138	446	181	25	155	85

FIGURE 8.7 Authenticated versus unauthenticated vulnerability scan results.

an "authenticated scan." This performs an effective "white box" assessment of the target by authenticating remotely on the device and then performing a variety of internal audits, including Registry reviews and network statistics. These results provide an accurate reflection of the true security posture of the target, and not just what is visible to a potential attacker. An authenticated scan is also more "friendly" on the target and does not typically inject as much hostile traffic into the network interfaces against various listening services. Figure 8.7 shows an example of the Nessus vulnerability scanner from Tenable Network Security where a "black box" unauthenticated scan yielded only four high-severity vulnerabilities, while a scan against the same target using authentication yielded 181 high-severity vulnerabilities.

The most common method of vulnerability scanning utilizes active mechanisms that place some packets on the network. The "aggressiveness" of the scan can be controlled in many applications, but as with any active technique, close attention must be paid to the potential impact of the scanner on the target.

Passive vulnerability scanners are available that collect the information needed for analysis via network packet capture rather than packet injection. Unlike active scanners that represent a "snapshot" view of the vulnerabilities on the target, passive methods provide a continuous view of the network. They are able to enumerate the network and detect when new devices are added. This type of scanner is well suited for industrial networks because of the static nature of the network topology and the regular traffic patterns and volumes that exist.

Host-based vulnerability scanners are also available; however, they would not likely be accepted within the ICS zones on industrial networks due to the fact that they must be installed on the target. These scanners do facilitate compliance auditing of configurations and content inspection, so they do fit a need. A good example of a host-based scanner would be the Microsoft Baseline Security Analyzer (MBSA).

It should be obvious at this point that vulnerability scanners are only capable of assessing a target against vulnerabilities that are known. In other words, it offers no guidance of any "zero-day" or those vulnerabilities that exist that have been discovered but the presence has not been communicated. This is why a strong defense-in-depth security program must depend on the ability to prevent, detect, respond, and correct against not only the threats that are known today, but also those threats that may appear tomorrow.

CAUTION

A vulnerability scanner should never be used on an online ICS and industrial network without prior testing and approval from those directly responsible for the operation of the ICS.

TIP

Just because a system has no vulnerabilities does not mean that it has been configured in a secure manner.

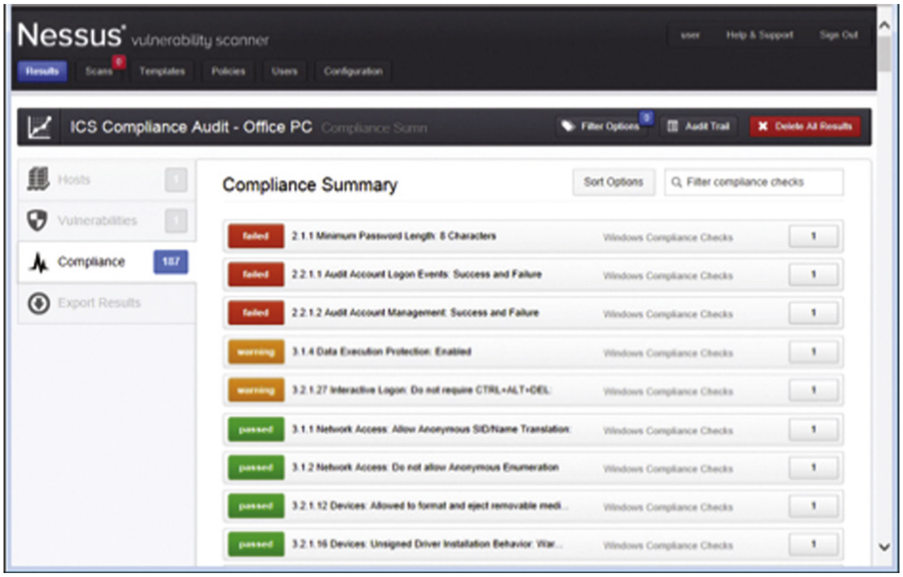
CONFIGURATION AUDITING

Vulnerability scanners are designed to assess a particular target against a set of known software vulnerabilities. Once the device has updated its firmware, installed the security updates for the operating system, and/or confirmed that the application software does not have any known weaknesses, the target is now considered safe … right? Wrong! The absence of software vulnerabilities does not mean that the software has actually been installed, configured, and even hardened in a manner that helps to reduce the possibility of a breach.

This is known as configuration "compliance auditing," and compares the current configuration of a host against a set of acceptable settings. These settings may be determined by an organization's security policy, a regulatory standard, or a set of industry-recognized benchmarks. Organizations that provide configuration benchmarks include NIST,[14] Center for Internet Security,[15] National Security Agency,[16] and Tenable Network Security.[17] The repository of compliance and audit files provided by Tenable is an aggregate of many available from other parts (such as CIS, NSA, and CERT) as well as custom developed files that are designed to provide a measure of compliances against published recommendations from BSI (Germany), CERT (Carnegie Melon University), and others.

The Nessus vulnerability scanner provides the ability to import predesigned or customized files that can be applied against target systems. These audits can be performed on the configuration of operating systems, applications, anti-virus software, databases, network infrastructure, and content stored on file systems. Figure 8.8 shows the output from a typical compliance audit. Tools are available that support the creation of audit files from existing policy inf files.

The US Department of Energy funded a project and partnered with Digital Bond to develop a set of security configuration guidelines for ICS.[18] The project developed Nessus audit configuration files for more than 20 different ICS components (see Table 8.13). These audit files provide a method by which asset owners, system integrators, and suppliers can verify that the systems have been configured in an optimal, preagreed manner against a consistent set of metrics. These audit files are available free-of-charge on the Digital Bond website, and were written using a syntax that provides for customization.[19]



ICS Compliance Audit - Office PC Compliance Summ

Compliance Summary

failed	2.1.1 Minimum Password Length: 8 Characters	Windows Compliance Checks	1
failed	2.2.1.1 Audit Account Logon Events: Success and Failure	Windows Compliance Checks	1
failed	2.2.1.2 Audit Account Management: Success and Failure	Windows Compliance Checks	1
warning	3.1.4 Data Execution Protection: Enabled	Windows Compliance Checks	1
warning	3.2.1.27 Interactive Logon: Do not require CTRL+ALT+DEL	Windows Compliance Checks	1
passed	3.1.1 Network Access: Allow Anonymous SID/Name Translation:	Windows Compliance Checks	1
passed	3.1.2 Network Access: Do not allow Anonymous Enumeration	Windows Compliance Checks	1
passed	3.2.1.12 Devices: Allowed to format and eject removable medi...	Windows Compliance Checks	1
passed	3.2.1.16 Devices: Unsigned Driver Installation Behavior: War...	Windows Compliance Checks	1

FIGURE 8.8 Compliant auditing report example.

VULNERABILITY PRIORITIZATION

Not all vulnerabilities that are discovered during a security test are necessarily exploitable. The development of exploits can prove to be valuable in determining if the vulnerabilities represent a real threat; however, the cost should be weighed against the benefits when considering this activity. What proves more effective is an objective method of rating the severity of vulnerabilities as they are discovered within a particular architecture. A vulnerability that exists on an Internet-facing corporate web server does not represent the same amount of risk as that vulnerability existing on a web server on a protected

Table 8.13 Bandolier Project ICS Details

Vendor	Platform
ABB	800xA PPA
Alstom Grid	e-terraplatform
CSI Control Systems International	UCOS
Emerson	Ovation
Matrikon	Security Gateway Tunneller
OSIsoft	PI Enterprise Server
Siemens	Spectrum Power TG
SISCO	AX-S4 ICCP Server
SNC-Lavalin ECS	GENe SCADA
Telvent	OASyS DNA

security zone that is nested deep within the organization. The outcome of this rating exercise can then be used to prioritize the corrective action plan following any site security test, allowing more severe (aka those representing a higher net risk to the organization) vulnerabilities to be mitigated before less severe ones are considered.

Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) is a free, open, globally accepted industry standard that is used for determining the severity of system vulnerabilities. The CVSS is not owned by any organization, but rather is under the custodial care of the Forum of Incident Response and Security Teams (FIRST). Each vulnerability is provided with one to three different metrics that produce a score on a scale of 0–10 that reflect the severity as the vulnerability is applied in different situations (see Figure 8.9). Each score consists of a "vector" that represents the value used for each component in calculating the total number. This scoring system allows vulnerabilities to be prioritized based on the actual risk they pose to a particular organization.

The Base metric and score are the only mandatory component of the CVSS, and is used to present the characteristics of a vulnerability that are constant with time and across different user environments. This score is commonly provided by the party responsible for disclosing the vulnerability, and is included with many advisories and security alerts. The Base score in the context of risk management can be thought of as a measure of "gross unmitigated" risk.

The Temporal metric and score provide refinement of the severity of the vulnerability by including the characteristics of a vulnerability that change over time, but not across different user environments. An example of how this number can change over time is that a vulnerability is initially disclosed and there are no public exploits available (Exploitability may be "unproven" or "proof of concept"). When a tool like the Metasploit Framework from Rapid7 makes an automated exploit module available, the Temporal score would increase to reflect this change (Exploitability may now be "functional" or "high"). The same can apply to the availability of a patch or update to correct the vulnerability. The patch may not be immediately available, but is published at some time in the future. The Temporal score does not consider any unique characteristics of a particular user or installation.

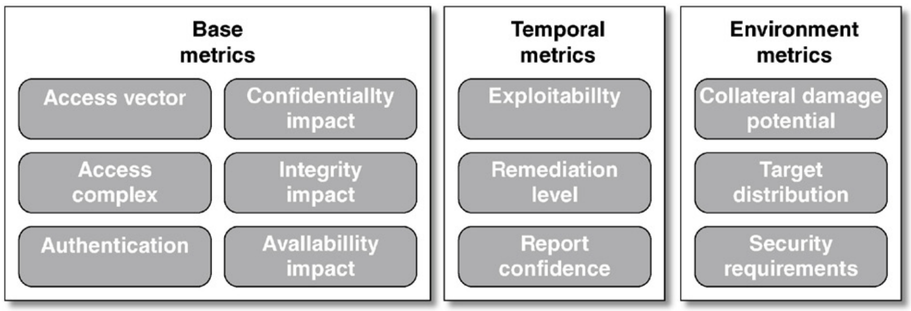


FIGURE 8.9 Common Vulnerability Scoring System (Version 2).

The Environmental metric and score reflects the characteristics of the vulnerability that are relevant and unique to a particular user's environment. This when calculated offers the best indication of "net unmitigated" risk to those systems and environments that possess the vulnerability.

There are quantitative formulas[20] that can be used to calculate the individual scores based on each vector. The NVD website of NIST provides an online calculator[21] that can be used.

RISK CLASSIFICATION AND RANKING

The process of Risk Classification and Ranking provides a means for evaluating the threats and vulnerabilities identified so far, and creating an objective method to compare these against one another. This activity supports the creation of the budget and schedule required to implement the security program of the industrial systems. Classification and ranking is important in making an "effective" security program that addresses the goals of both business operations and operational security.

CONSEQUENCES AND IMPACT

The data collection aspect of the security test is complete, and it is now time to prioritize the results through classification and ranking. The process to this point as shown in Figure 8.2 has resulted in a set of physical and logical assets that have been matched against one or more threats as defined by the actor (person or persons who would initiate the attack), the vector (entry point used to introduce the malicious content of the attack), and the event (methods used to perform the attack). The assets have also been assessed to determine if there are any vulnerabilities or flaws that could possibly be exploited by an attacker. Remembering the earlier definition of risk, the last piece of information needed is a determination of the consequences or impact to operations that would occur should the cyber event occur. The term "operations" has been used here instead of "industrial systems" because remember the primary purpose of an ICS is to control a manufacturing facility and not merely to process information.

Once the risk assessment team shifts its focus from "impact to the system" to "impact to the plant or mill," the severity of the unmitigated risk can become significant. Table 8.14 provides some examples of the consequences that could occur should any ICS component fail to perform their intended function. These consequences can have local (plant), regional (surrounding community), or global (national, multinational) impact.

Many would challenge that a single cyber event could have global consequences. It was reported in a US Department of Homeland Security's National Risk Profile that old and deteriorating infrastructure in the United States could pose significant risks to the nation and its economy.[22] Now consider natural gas pipelines as part of this deteriorating infrastructure and how there have been more than 2800 "significant" gas pipeline accidents in the United States since 1990.[23] The ICS monitors and controls the parameters associated with the mechanical integrity of these pipelines. What is the attractiveness of

Table 8.14 Common ICS Consequences

Common ICS Consequences
Impact to quality
Customer reputation
Loss of production
Loss of intellectual property
Economic (micro) impact
Mechanical stress or failure
Environmental release
Catastrophic equipment failure
Localized loss of life
Generalized panic
Economic (macro) impact
Widespread loss of life

this target? Adversaries would have little chance of victory if the battle was fought on a traditional military battleground, but in cyberspace, the odds shift dramatically and the ICS is a critical target in any cyber war launched against infrastructure.

HOW TO ESTIMATE CONSEQUENCES AND LIKELIHOOD

The challenge that many face in risk classification is how to apply a measure of the "likelihood" that a cyber-attack will occur. Traditional IT information risk processes consider likelihood as a measure of time—will this event happen in one month, one year or longer. If straight quantitative methods of calculating risk were used, a very serious threat with multiple vulnerabilities could quickly be subdued by applying a low likelihood number and assuming that the event does not occur until some point in the future. Can you see the flaw here? If the same event that can occur today can also occur next year, does it not mean that the cost associated with the consequences would be greater? Absolutely—factors such as inflation, cost of capital, population growth, and many others will cause the cost of the event to grow, yet this is not fed back into the initial calculation model. Using the previous pipeline as an example, if you do nothing to maintain the pipeline then it is going to fail at some point in the future. The consequences are likely to be greater than today as well because the pipeline was originally built in a rural area, but in 20 years it is now part of a residential area.

These situations illustrate the need to use some other form of estimating the likelihood of a cyber event and the consequences should the event occur. The DREAD model, named from the first letter of each of the five rating categories, was developed by Microsoft as part of their Software Development Lifecycle (SDL) to provide a method to classify security bugs. This model (shown in Table 8.15) provides an indirect means of calculating consequences and likelihood by looking at these factors in a different

Table 8.15 DREAD Model[29]

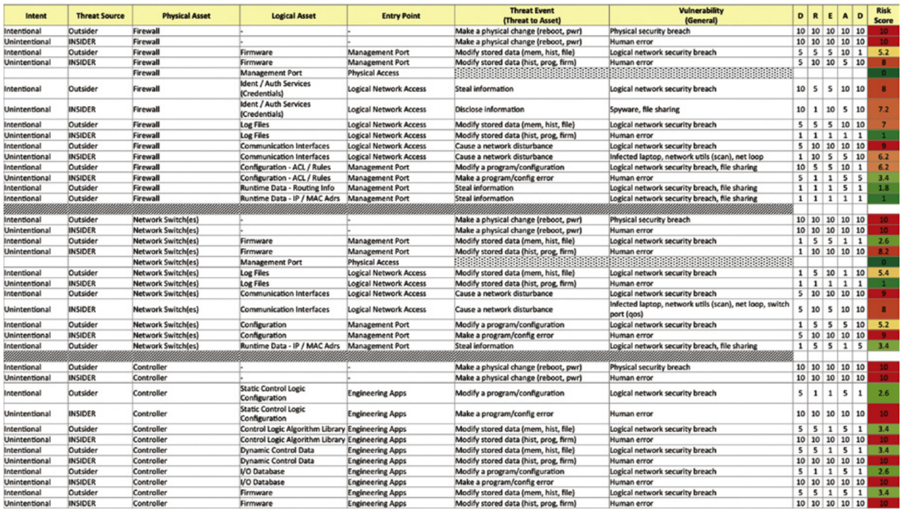
	Rating	**High**	**Medium**	**Low**	**Indirectly Measures**
D	**Damage Potential**	Attacker can subvert the security; get full trust authorization; run as administrator; upload content	Leaking sensitive information	Leaking trivial information	**Consequences**
R	**Reproducibility**	Attack can be reproduced every time; does not require a timing window; no authentication required	Attack can be reproduced, but only with a timing window and a particular situation; authorization required	Attack is very difficult to reproduce, even with knowledge of the security vulnerability; requires administrative rights	**Likelihood**
E	**Exploitability**	Novice programmer could make the attack in a short time; simple toolset	Skilled programmer could make the attack, then repeat the steps; exploit and/or tools publicly available	Attack requires and extremely skilled person and in-depth knowledge very time to exploit; custom exploit/tools	**Likelihood**
A	**Affected Users**	All users; default configuration; key assets	Some users; non-default configuration	Very small percentage of users; obscure feature; affects anonymous users	**Consequences**
D	**Discoverability**	Published information explains the attack; vulnerability is found in the most commonly used feature; very noticeable	Vulnerability is in a seldom-used part of the product; only a few users should come across it; would take some thinking to see malicious use	Bug is obscure; unlikely that users will work out damage potential; requires source code; administrative access	**Likelihood**

way. For example, rather than asking if the threat of a vulnerability being exploited is likely to occur in the next six months, why not consider how easy it is to obtain the knowledge (exploit code) necessary to exploit the vulnerability. If the information is readily available via the Internet or open-source tools, the likelihood that this vulnerability will be exploited is much greater than if no proof-of-concept code has ever been developed. Similarly, the vulnerability is far more likely to be exploited if the necessary skill level of the attacker is low (e.g. a script kiddie could perform the attack).

The DREAD model provides a “qualitative” method of assigning a value to each of the five classifications that can be useful for group assessment exercises where it can be difficult to get consensus on an exact figure (dollar amount, number of months, etc.). A number value can be assigned to each ranking allowing the DREAD model to be implemented as a spreadsheet that is used along with the asset, threat, and vulnerability data that have been previously obtained. The Six Sigma Quality Function Deployment (QFD) is an appropriate methodology to introduce at this point, as this can be applied directly to the DREAD model transforming the qualitative parameters (high, medium, low) into quantitative values that can be analyzed statistically.

RISK RANKING

The application of QFD to the DREAD model will allow the data to be consolidated and used alongside the asset, threat, and vulnerability data. Figure 8.10 illustrates part of an example spreadsheet for the complete process used against the reference architecture shown in Figure 8.5. The mapping was accomplished using values of 10 = high, 5 = medium, and 1 = low. This was done in order to provide adequate numerical separation between a high or “significant” item and a low or medium event. With this numbering scheme, two medium ratings would equal one high. Other



Intent	Threat Source	Physical Asset	Logical Asset	Entry Point	Threat Event (Threat to Asset)	Vulnerability (General)	D	R	E	A	D	Risk Score
Intentional	Outsider	Firewall	-	-	Make a physical change (reboot, pwr)	Physical security breach	10	10	10	10	10	10
Unintentional	INSIDER	Firewall	-	-	Make a physical change (reboot, pwr)	Human error	10	10	10	10	10	10
Intentional	Outsider	Firewall	Firmware	Management Port	Modify stored data (mem, hist, file)	Logical network security breach	5	5	5	10	1	5.2
Unintentional	INSIDER	Firewall	Firmware	Management Port	Modify stored data (hist, prog, firm)	Human error	5	10	10	5	10	8
		Firewall	Management Port	Physical Access	▨	▨						0
Intentional	Outsider	Firewall	Ident / Auth Services (Credentials)	Logical Network Access	Steal information	Logical network security breach	10	5	5	10	10	8
Unintentional	INSIDER	Firewall	Ident / Auth Services (Credentials)	Logical Network Access	Disclose information	Spyware, file sharing	10	1	10	5	10	7.2
Intentional	Outsider	Firewall	Log Files	Logical Network Access	Modify stored data (mem, hist, file)	Logical network security breach	5	5	5	10	10	7
Unintentional	INSIDER	Firewall	Log Files	Logical Network Access	Modify stored data (hist, prog, firm)	Human error	1	1	1	1	1	1
Intentional	Outsider	Firewall	Communication Interfaces	Logical Network Access	Cause a network disturbance	Logical network security breach	5	10	10	10	10	9
Unintentional	INSIDER	Firewall	Communication Interfaces	Logical Network Access	Cause a network disturbance	Infected laptop, network utils (scan), net loop	1	10	5	5	10	6.2
Intentional	Outsider	Firewall	Configuration - ACL / Rules	Management Port	Modify a program/configuration	Logical network security breach, file sharing	10	5	5	10	1	6.2
Unintentional	INSIDER	Firewall	Configuration - ACL / Rules	Management Port	Make a program/config error	Human error	5	1	1	5	5	3.4
Intentional	Outsider	Firewall	Runtime Data - Routing Info	Management Port	Steal information	Logical network security breach, file sharing	1	1	1	5	1	1.8
Intentional	Outsider	Firewall	Runtime Data - IP / MAC Adrs	Management Port	Steal information	Logical network security breach, file sharing	1	1	1	1	1	1
▨	▨	▨	▨	▨	▨	▨						
Intentional	Outsider	Network Switch(es)	-	-	Make a physical change (reboot, pwr)	Physical security breach	10	10	10	10	10	10
Unintentional	INSIDER	Network Switch(es)	-	-	Make a physical change (reboot, pwr)	Human error	10	10	10	10	10	10
Intentional	Outsider	Network Switch(es)	Firmware	Management Port	Modify stored data (mem, hist, file)	Logical network security breach	1	5	5	1	1	2.6
Unintentional	INSIDER	Network Switch(es)	Firmware	Management Port	Modify stored data (hist, prog, firm)	Human error	1	10	10	10	10	8.2
		Network Switch(es)	Management Port	Physical Access	▨	▨						0
Intentional	Outsider	Network Switch(es)	Log Files	Logical Network Access	Modify stored data (mem, hist, file)	Logical network security breach	1	5	10	1	10	5.4
Unintentional	INSIDER	Network Switch(es)	Log Files	Logical Network Access	Modify stored data (hist, prog, firm)	Human error	1	1	1	1	1	1
Intentional	Outsider	Network Switch(es)	Communication Interfaces	Logical Network Access	Cause a network disturbance	Logical network security breach	5	10	10	10	10	9
Unintentional	INSIDER	Network Switch(es)	Communication Interfaces	Logical Network Access	Cause a network disturbance	Infected laptop, network utils (scan), net loop, switch port (qos)	5	10	5	10	10	8
Intentional	Outsider	Network Switch(es)	Configuration	Management Port	Modify a program/configuration	Logical network security breach	1	5	5	5	10	5.2
Unintentional	INSIDER	Network Switch(es)	Configuration	Management Port	Make a program/config error	Human error	5	10	10	10	10	9
Intentional	Outsider	Network Switch(es)	Runtime Data - IP / MAC Adrs	Management Port	Steal information	Logical network security breach, file sharing	1	5	5	1	5	3.4
▨	▨	▨	▨	▨	▨	▨						
Intentional	Outsider	Controller	-	-	Make a physical change (reboot, pwr)	Physical security breach	10	10	10	10	10	10
Unintentional	INSIDER	Controller	-	-	Make a physical change (reboot, pwr)	Human error	10	10	10	10	10	10
Intentional	Outsider	Controller	Static Control Logic Configuration	Engineering Apps	Modify a program/configuration	Logical network security breach	5	1	1	5	1	2.6
Unintentional	INSIDER	Controller	Static Control Logic Configuration	Engineering Apps	Make a program/config error	Human error	10	10	10	10	10	10
Intentional	Outsider	Controller	Control Logic Algorithm Library	Engineering Apps	Modify stored data (mem, hist, file)	Logical network security breach	5	5	1	5	1	3.4
Unintentional	INSIDER	Controller	Control Logic Algorithm Library	Engineering Apps	Modify stored data (hist, prog, firm)	Human error	10	10	10	10	10	10
Intentional	Outsider	Controller	Dynamic Control Data	Engineering Apps	Modify stored data (mem, hist, file)	Logical network security breach	5	5	1	5	1	3.4
Unintentional	INSIDER	Controller	Dynamic Control Data	Engineering Apps	Modify stored data (hist, prog, firm)	Human error	10	10	10	10	10	10
Intentional	Outsider	Controller	I/O Database	Engineering Apps	Modify a program/configuration	Logical network security breach	5	1	1	5	1	2.6
Unintentional	INSIDER	Controller	I/O Database	Engineering Apps	Make a program/config error	Human error	10	10	10	10	10	10
Intentional	Outsider	Controller	Firmware	Engineering Apps	Modify stored data (mem, hist, file)	Logical network security breach	5	5	1	5	1	3.4
Unintentional	INSIDER	Controller	Firmware	Engineering Apps	Modify stored data (hist, prog, firm)	Human error	10	10	10	10	10	10

FIGURE 8.10 Risk and vulnerability assessment worksheet.

possibilities include using a 1,3,7 system so that three medium ratings would exceed one high, and so on. The numbers 1,2,3 have not been used because this would place inappropriate weighting on low or "insignificant" items compared to high ones.

The use of a spreadsheet tool, such as Microsoft Excel, will allow the values calculated from the DREAD model to be compared across all of the listed items, forming the basis of a ranked list of items and priority of events to address those security weaknesses discovered during the security test. The spreadsheet example in Figure 8.10 utilizes the *Conditional Formatting* features of Excel and applies a shading scale over the range of 0–10. This provides easy visual recognition.

RISK REDUCTION AND MITIGATION

The methodology discussed in this chapter provides a consistent, repeatable means to assess the security implemented around an ICS and the industrial networks. The process has yielded a prioritized list of items in terms of net "unmitigated" risk to the ICS and the plant under its control. Some risks may have been mitigated to an acceptable level following the security and vulnerability assessment. The final activity for those remaining risk items is to apply a range of cyber security controls or countermeasures to the assets within the ICS in order to reduce or mitigate these risks. The selection and implementation of security controls is discussed elsewhere in this book, and is available through numerous standards and controls catalogues.

Improving the cyber resilience of an ICS should be one of the many benefits obtained through the implementation of an industrial security program. This resiliency is accomplished when security controls are selected that span the Security Life Cycle shown in Figure 8.11. The Life Cycle is used to illustrate the continuous

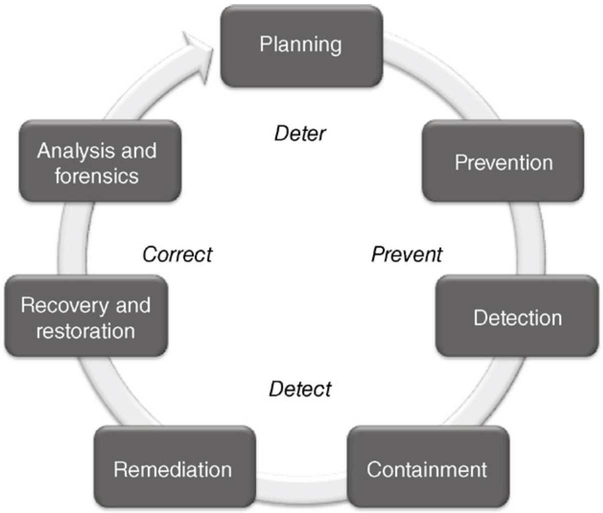


FIGURE 8.11 Security life cycle model and actions.

process of addressing cyber security that not only begins with threat deterrence and prevention, but equally balances threat detection and correction necessary to identify cyber events in a timely manner and to respond accordingly in order to minimize the consequences of the event and return the manufacturing facility to normal operation in a safe and timely manner.

Organizations often devote large portions of their security budget on mechanisms to prevent an attack from occurring.[24] External parties often notify these same organizations that lack a balanced investment in controls to detect an event of a breach long after the attack.[25] Security should be considered as a long-term "strategic" investment rather than a short-term or one-time "tactical" expense. Those that invest and build manufacturing facilities understand the long-term life cycle of the capital investment, so it makes sense that the operational security used to protect these same facilities is treated in a similar manner and receives continuous attention (and budget) like other operational expenses (maintenance, improvements, training, etc.).

The Security Life Cycle can be used as an effective tool when mapping security controls to each phase, and will help identify potential short- and long-term weaknesses in the security strategy that could affect the overall resilience of the security program.

SUMMARY

The implementation of an industrial cyber security program is an investment of both time and money. The primary objective is always to secure not just the industrial networks and those systems that utilize it, but to also aide in securing the plant or mill that depends on these systems to remain operational in a safe, efficient, and environmentally responsible manner. Risk management is a daily part of those that lead and manage these facilities, and the management of cyber risk is a vital component. Threats to these industrial sites can originate inside and outside the organization where vulnerabilities can be exposed and exploited both maliciously and accidently. The consequences from these events can span simple "inconveniences" all the way to catastrophic mechanical failures that may result in plant shutdowns, fires, hazardous releases, and loss of life.

The evolution of industrial automation and control systems and industrial networks over the past 40 years has left many organizations with systems that possess numerous security weaknesses that cannot be replaced overnight. The process of upgrading and migrating the plethora of integrated industrial systems comprising an ICS can take years. A balanced approach to industrial security must be followed that provides the balanced and objective evaluation of risk in terms of threats, vulnerabilities, and consequences in order to align an organization's short- and long-term goals while workings toward a more safe and secure industrial automation environment.

ENDNOTES

1. Repository of Industrial Security Incidents, "2013 Report on Cyber Security Incidents and Trends Affecting Industrial Control Systems," June 2013.
2. "2013 Data Breach Investigations Report," Verizon, April 2013.
3. Repository of Industrial Security Incidents, "2013 Report on Cyber Security Incidents and Trends Affecting Industrial Control Systems," June 2013.
4. "15th Annual Computer Crime and Security Survey," Computer Security Institute, 2010/2011.
5. Open-Source Vulnerability Database, <http://osvdb.org>, sited July 1, 2014.
6. "ICS Vulnerability Trend Data," <http://www.SCADAhacker.com/resources.html>, sited July 1, 2014.
7. "Data breach costs still unknown: Target CEO," CNBC, <http://www.cnbc.com/id/101694256>, sited July 28, 2014.
8. "Analyst sees Target data breach costs topping $1 billion," TwinCities Pioneer Press, <http://www.twincities.com/ci_25029900/analyst-sees-target-data-breach-costs-topping-1>, sited July 28, 2014.
9. "The Target Breach, By the Numbers," Krebs on Security, <http://krebsonsecurity.com/2014/05/the-target-breach-by-the-numbers/>, sited July 28, 2014.
10. "The Shamoon Attacks," Symantec Security Response, <http://www.symantec.com/connect/blogs/shamoon-attacks>, sited July 29, 2014.
11. Wm. Stanek, "Windows Command-Line Administrator's Pocket Consultant," Microsoft Press, 2nd Edition, 2008.
12. "National Vulnerability Database Version 2.2," National Institute of Standards and Technology / U.S. Dept. of Homeland Security National Cyber Security Division, <http://nvd.nist.gov>, sited July 30, 2014.
13. "Open-Source Vulnerability Database," <http://osvdb.org>, sited July 30, 2014.
14. "National Checklist Program Repository," National Institute of Standards and Technology, <http://web.nvd.nist.gov/view/ncp/repository>, sited July 30, 2014.
15. "CIS Security Benchmarks," Center for Internet Security, <https://benchmarks.cisecurity.org>, sited July 30, 2014.
16. "Security Configuration Guides," National Security Agency / Central Security Service, <http://www.nsa.gov/ia/mitigation_guidance/security_configuration_guides/index.shtml>, sited July 30, 2014.
17. "Nessus Compliance and Audit Download Center," Tenable Network Security, <https://support.tenable.com/support-center/index.php>, sited July 30, 2014.
18. "Bandolier," DigitalBond, <http://www.digitalbond.com/tools/bandolier>, sited July 30, 2014.
19. "Nessus Compliance Checks Reference," Revision 53, Tenable Network Security, July 2014.
20. "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," Forum of Incident Response and Security Teams, <http://www.first.org/cvss/cvss-guide.html>, sited July 30, 2014.
21. "Common Vulnerability Scoring System Version 2 Calculator," National Institute of Standards and Technology – National Vulnerability Database, <http://nvd.nist.gov/cvss.cfm?calculator&version=2>, sited July 30, 2014.

22. "DHS Says Aging Infrastructure Poses Significant Risk to U.S.," Public Intelligence, <http://publicintelligence.net/dhs-national-risk-profile-aging-infrastructure/>, sited July 31, 2014.
23. "Aging Natural Gas Pipelines Are Ticking Time Bombs, Say Watchdogs," FoxNews, <http://www.foxnews.com/us/2011/02/28/aging-natural-gas-pipelines-ticking-time-bomb-say-experts/>, sited July 31. 2014.
24. "15th Annual Computer Crime and Security Survey," Computer Security Institute, December 2010.
25. "Risk Intelligence Governance in the Age of Cyber Threats," Deloitte Consulting, January 2012.
26 "Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model," Version 3.1 - Revision 4, September, 2012.
27 "Guide to Conducting Risk Assessments," Special Publication 800-30, National Institute for Standards and Technology, September 2012.
28 "Guide to Conducting Risk Assessments," Special Publication 800-30, National Institute for Standards and Technology, September 2012.
29 "Threat Modeling," Microsoft Developer Network, <http://msdn.microsoft.com/en-us/library/ff648644.aspx>, sited July 31, 2014.