Kaspersky ICS CERT

kaspersky

The internet of things security maturity model: a nudge for IoT cybersecurity

Ekaterina Rudina
Evgeny Goncharov

13.08.2019 Version 1.0

Contents

Developing cyberthreat protection strategies is a challenging task, especially for industrial systems and the internet of things. Numerous parties are involved in the processes of designing, developing, integrating, using and maintaining such systems.

The different parties involved will assess the risks associated with attacks differently. Some businesses may view security as a drawback (because of longer market entry times due to various security requirements), while others may view it as an advantage (a secure product will have a competitive edge from a marketing standpoint).

Ideally, the choice of security tools and measures should resolve the very complex task of optimizing corporate resources and serving the interests of the business within the framework of internal and external limitations.

The purpose of the IoT Security Maturity Model (IoT SMM) is to help choose protection measures against cyberthreats that correspond to the company's actual business needs.

The IoT security maturity model can be applied in the same way to relatively complex devices, to the components of IoT devices and infrastructures, and to the actual infrastructures themselves. The document IoT Security Maturity Model: Practitioner's Guide gives three examples of this security model as applied to: a bottling line from the system integrator's standpoint, an over-the-air update gateway for automotive electronic control unit firmware from the standpoint of a tier-1 supplier, and residential security cameras from the standpoint of the consumer.

Why the IoT security maturity model is necessary

Game of chicken

The main problem when developing a cyberthreat protection strategy for industrial systems and the internet of things stems from the fact that businesses typically see security as a matter of concern and a cost item. That's why provision of the necessary aspects of cybersecurity and protection from threats often becomes a collective game of chicken[1].

The manufacturers of industrial process automation products still try to shift responsibility for the provision of product security to their clients by stating their products must be used in an isolated environment (no connections to the internet or office networks, etc.). By doing so, they deliberately ignore the fact that most enterprises cannot meet such requirements because of their need to boost effectiveness.

On the other hand, representatives from a variety of industries often claim they cannot (or are reluctant to) apply certain security measures (e.g., install an operating system patch) or install tools (e.g., install an antivirus) on their industrial automation systems (e.g., an operator workstation) without first getting approval from the manufacturer. In this way, enterprises seek to shift, at least partially, responsibility when it comes to making decisions on the provision of security.

In the context of security provision, the search for a balance between business stimuli leads to a strategy of brinkmanship[2]. To maintain balance, each party, including the manufacturers of specific types of hardware, software, system integrators, service providers, agents and business owners, look for an optimum set of security measures while attempting to stay within budget.

'Sufficient security'

Different organizations will always have different assessments of their required security levels. Even if two organizations run similar risks, the consequences of potential incidents may be harsher for one of them than for the other.

In some cases, cyberattacks can pose a significant threat to organizations, even if they are not directly responsible for an incident. For example, the compromise of hardware from the world's leading manufacturer installed on the insecure network of a client creates risks for that manufacturer, although the client may well be to blame for failing to follow the manufacturer's recommendations and ensure proper configuration of the hardware. Alternatively, for an enterprise that is classified as a critical information infrastructure, emergencies stemming from

[1] The name "chicken" has its origins in a game in which two drivers drive towards each other on a collision course: one must swerve, or both may die in the crash, but if one driver swerves and the other does not, the one who swerved will be called a "chicken", meaning a coward. A more dynamic version of the game takes place in real time, when the decision is constantly changing, and the risk increases over time. An even more complex version involves several players.

[2] Brinkmanship is a strategy option in a dynamic game of chicken. Brinkmanship involves a threat that creates a risk but not an inevitable game outcome that would be undesired for all players. In this strategy, each player ignores a wish for his/her actions and gradually increases the risks until either player cedes or the game reaches an unfavorable outcome (in our case, the unfavorable outcome is damage caused by a security incident).

cyberattacks are impermissible, even if they are caused by exploitation of vulnerabilities left unpatched by a manufacturer.

The actual task of implementing a “sufficiently secure system” may be set differently for different participants of the chain of production, implementation and use of industrial IoT systems. The standpoints of the manufacturer and the consumer are probably the furthest from each other.

Vendors are only interested in ensuring the security of the products they provide, though they have to take into account the entire range of possible operating environments, as well as the expectations and requirements of all their clients. When it comes to security, a manufacturer’s main goal can be summed up as follows: spend minimum effort to satisfy client and prospective client expectations regarding product security.

The consumer, in turn, is not interested in the security of the product per se, but how it affects the security of the infrastructure into which that product is integrated. The consumer’s main security goal can be formulated as: obtain a product that will not compromise (or, ideally, will improve) the security of their infrastructure.

Having said that, tasks to ensure security that involve scrutiny of a specific product may differ substantially for both parties, as may the ways those tasks are fulfilled. To elaborate on this point, we have provided an Appendix with examples of problem statements for two hypothetical actors involved in the security provision process at an industrial facility, namely the facility owner (or operator) and the manufacturer of the software or hardware used at the facility.

Aim of the IoT security maturity model

The correct choice of security tools and measures is not always obvious. Moreover, the localized business goals and the security decisions motivated by those goals may prove to be different and even incompatible for the different actors in the security provision process (e.g., by the manufacturer and the consumer of OT products and services). Perhaps the most unpleasant aspect in this situation is when one party fails to understand the limitations, needs and reasons behind the security decisions taken by another party.

One of the aims of the model described in this article was to attempt to offer a common denominator and set up a system of coordinates in which the security decisions made by one party would be transparent and understandable for the other party.

In this article, we will attempt to avoid ambiguity by considering the standpoint of the software and hardware vendor in the context of choosing the tools and measures to ensure security of the manufactured product when it is used for its intended purposes.

The end goal of the IIC IoT Security Maturity Model (IoT SMM) is to ensure that cyberthreat protection measures correspond to the actual business needs. The goal is to create a specific description of the “sufficient security” state of the system, help those responsible for that system’s security to focus on the best methods of achieving that state, and identify the appropriate means of protection.

A mature security system

A *mature* security system is characterized by a sufficient set of protection measures that do not compromise its functionality. However, the definition of “sufficient protection” and the idea of “negative impact on functionality” are specific to each system.

This raises the question: how does the manufacturer, say, of complex industrial equipment decide on a “sufficient” set of protection measures given the specific characteristics, requests and restrictions on the implementation of those measures?

A request is created at the business-stakeholder level to “protect hardware from hacker attacks”. The core problem is that the business representatives are almost never specialists in information security. For example, a device may be vulnerable to attacks because of faulty software architecture. In the long term, the option of a costly hardware migration to an alternative, more attack-resilient platform may be considered. The current versions also require technical support and maintenance, including a vulnerability check and checks for security patch availability. Customer feedback about vulnerabilities and incidents also requires maintaining a dedicated service.

To address this problem of choosing the appropriate security tools and measures, the business needs to employ a systemic approach that will connect priorities to security goals, and security measures directly to the expected effect. As there are quite a number of methods to make the system more secure (or sufficiently compensate for its insecurity), these methods need to be ranked in order, so the most appropriate options can be selected.

Role of the choice architecture

Choosing a security strategy is difficult, and this difficulty is not really about finding the optimum price-efficiency balance for the security measures, but rather about the different narratives behind the definition of this optimum for the various stakeholders.

Given the differences in business needs and the unavailability of sufficient information about possible cyberattacks and their impact on the system’s operation, the vendor and the client (and, possibly, other interested organizations and entities, such as regulators) may consider different attack scenarios more probable and potentially more dangerous, and view different protection practices to be of a higher priority on the implementation list.

For example, let’s review the possible positions of a hypothetical SCADA system vendor and its client.

In theory, it’s in both the client’s and the vendor’s interests to obtain a secure-by-design system that requires minimum effort to maintain its security features. However, reality often puts paid to those intentions. In our example, the vendor’s software contains vulnerabilities, known or as yet unknown, and it has been deployed in an insecure environment. Over time, new variations of attacks emerge, some of which may not have been considered at the product development and implementation stages.

Evaluation of attack-related risks differs on the vendor’s side and the client’s side. Harmonization of vendor and client priorities, the choice of protection measures, the implementation scope of those measures and the implementation timelines all require a structured representation of the options, so that there is the possibility to perform at least an approximate evaluation of the balance between their effectiveness and the required resources, i.e., of the choice architecture.

Choice architecture is a systemization of the options that pushes people to choose courses of action and then to begin implementing those actions, which in this specific case is to create a more secure system.

Let’s look at a situation where a vulnerability is discovered in the vendor’s product.

It might be in the client's interest to immediately apply a security patch for the vulnerability. The vendor, however, may be in no rush and may have a number of reasons for delaying the release of a patch until the next version of the product is released.

For the vendor, releasing a patch means diverting resources from a product release that the market is waiting for. A security patch will not generate new clients, while a new product feature may. No one pays for the hours spent releasing security patches – the resources spent this way are seen as a loss. Releasing a patch also carries reputational risks for many vendors, because by doing so the vendor is officially declaring there was an error in its product. By releasing patches, the vendor increases the number of product versions that need to be maintained, which exacerbates the problem of consistency for those versions. And that's not the full list of expenses: the patches need to be delivered in ways that are suitable for the client, and then the client needs to be made aware that the update needs to be installed. By delaying a patch release, the vendor postpones all that work until the next product release, by which time the problem may be less relevant, or a whole series of problems can be addressed with just one patch – saving money on both development and testing.

The reverse situation can also apply. In certain cases, the client may want to receive patches accumulated in larger batches, as cumulative fixes or as part of new versions of the product that address the client's production and business goals more effectively and correct any security issues properly and more thoroughly. Organizing the patch management process at an enterprise is a difficult and expensive task. Due to the peculiarities of how automation systems and the workflow processes are implemented in certain industries, the only time for installing patches may be in those rare moments when production lines are suspended. Installation of a patch may potentially pose problems in terms of compatibility with the product's environment in the client's infrastructure. All updates on the client's side should ideally undergo a procedure of pre-testing that cannot always be done quickly – either for technical reasons or even due to formal restrictions related to the specific nature of production.

It would be a misconception to think that security patches are just simple fixes that don't affect the product's core functionality or architecture. Sometimes, correcting an error requires changes to the architecture – unfortunately, this is often unavoidable.

Of course, there are no guarantees that in the new version of the product the vulnerability will definitely be rectified in a more valid way than it would by releasing a patch. If the error can be corrected without introducing changes into the architecture (i.e., it can be corrected by setting more checks, changing the default values and/or introducing corrections into the user documentation), then, very probably, attempts will be made to correct it in the same way in the next version. Typically, changes are only introduced into the architecture if there are no other technical capabilities, or if introducing such changes is deemed cheaper than correcting the problems, one after another, that may arise in the future when using the existing infrastructure.

In such cases, the client's unwillingness or inability to correct product security vulnerabilities may become a problem for the vendor.

Because the initial positions of the vendor and the client may differ greatly, even toward such a seemingly simple protection measure as releasing and installing patches, a choice strategy must be offered to them that will result in them agreeing on a resolution.

How does the IoT security maturity model work

Hierarchy of security practices

The choice architecture and the core of the IoT security maturity model are a hierarchy of security practices. A security practice is, for example, the implementation of access control, protection of data while data is stored or transmitted, or management of security updates. A systemic approach to the choice of security options is supported by grouping practices according to their expected effects. To make the process of choosing as simple as possible, groups of practices are combined into domains at the top level.

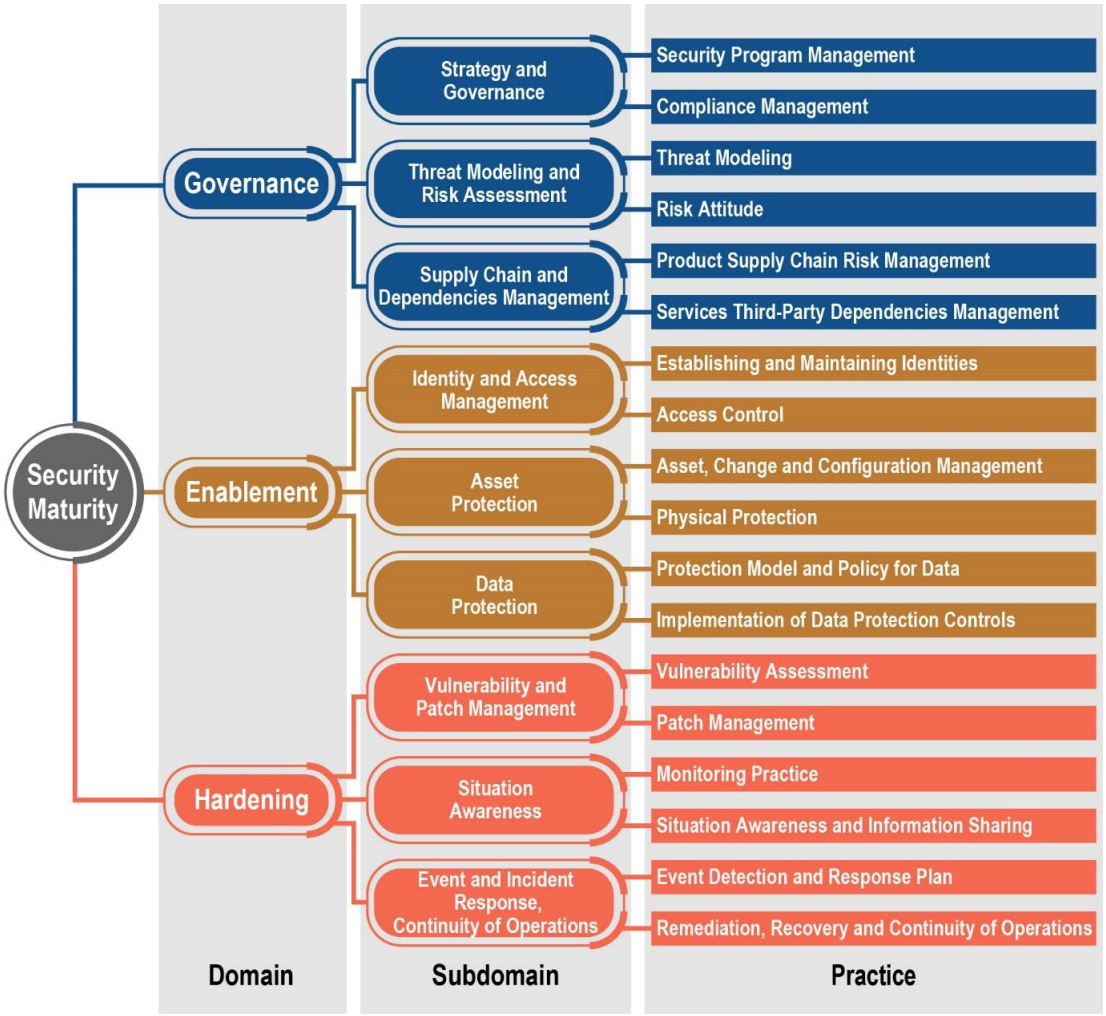
The three top-level security domains include:

(1) Security management and organizational measures (Governance),
(2) Provision of security by design (Enablement),
(3) Security hardening (Hardening).

The vendor decides whether a certain domain should be assigned priority over another based on business needs and the peculiarities of each specific system. (Business needs take priority over system peculiarities.)

The term “choice architecture” for the hierarchy of security practices does not mean that only a single option should be chosen. Even when there are security hardening procedures available, systems must not be left insecure by design, and conversely, any systems must be capable of prompt implementing security measures. In any case, organizational measures, be it a security program or a security policy for the supply chain, foster confidence that the technical measures are effective. The measures from these domains also need to be planned.

At the second level, each domain falls into three subdomains that classify security practices according to the problem they address. And ultimately, each subdomain references two practices, each of which addresses a certain task. For example, Governance includes the Supply Chain and Dependencies Management subdomain, which in turn consists of Product Supply Chain Risk Management and Third Party Dependencies Management (see diagram).



The hierarchy of domains, subdomains and security practices
Source: IoT Security Maturity Model: Description and Intended Use White Paper

How security maturity can be measured

In order to set priorities and compare the implementation of security measures, there needs to be a measurement scale. There are two parameters that can be used to assess the effectiveness of a chosen cyberthreat protection method that have been carried over from traditional IT security to the IoT environment. The first parameter includes how well the actual approach is implemented, how well its application is systemized, and how comprehensive its implementation is (so-called comprehensiveness). A good example of this parameter is the assessment of web application security which can be applied with varying degrees of diligence. Threats can be described in general: theft of credentials, password brute force attack, or DDoS attack. This would be the first, *minimum* level of comprehensiveness. The operation scenarios of applications can be analyzed to refine the threat model – this would be the second, *ad hoc* level. The OWASP Top 10 attack systematization can be used during the refinement, and the STRIDE model can also be added to this; this would take us to the third, *consistent* level of comprehensiveness. Ultimately, an entire formal process of periodical re-evaluation of threats can be organized that would include all the above methods – this would deliver the maximum, fourth, *formalized* level.

It's very important to note that comprehensiveness is not yet maturity. A banking web application requires the most comprehensive approach, while a web application for comparing the current time in different time zones may be limited to looking at operational scenarios to identify potential problems. In other words, unlike comprehensiveness, maturity is a relative value.

The second parameter that's of importance for the internet of things is how specific the approach needs to be, given the requirements of the industry or even a specific system. Assessment of devices created, for example, in the automotive industry (and many other industries) must focus primarily on preventing physical harm to people's lives and health, and damage to the environment. Significant threats to a medical device are those that can cause changes to its specific operational parameters, even if those changes are insignificant (e.g., doses of a patient's medicine). Shifting the focus to specific problems (*scope*) for the implementation of a specific security practice also directly defines its maturity when it involves the internet of things. Here, we review three options: general non-specific (*General*) implementation, industry-specific (*Industry*) implementation and system-specific (*System*) implementation. The latter variant is important, as many solutions are now emerging at the intersection of industries, as well as previously unseen special-purpose solutions, such as those for "smart home".

The maturity level is thereby determined by the comprehensiveness of the implementation of security practices and the specific nature of its implementation for an IoT application. Each application (organization, system and separate solution) requires different levels of comprehensiveness and scope. The target level of maturity will consequently be different for different applications. The current level of maturity is measured against it target level, which is defined by the IoT security maturity profile.

What is an IoT security maturity profile?

The pairing of comprehensiveness + scope for all security practices is called an IoT security maturity profile. If the pairing defines the targets for a specific system, then it is called the IoT security maturity target. There are 36 parameters described in detail, or two parameters per each practice. To obtain them, you need to perform a simple procedure, successively defining the security goals at the top-level domains, the tasks for subdomains and the assignment of security practices at the bottom level of the hierarchy. The picture of the target maturity profile gradually becomes clearer as you analyze the security provision targets and tasks. The target maturity profile is a description of 100% security maturity for the system that should be seen as the end goal during its development.

When starting the process of describing the target IoT security maturity profile at the business level, the priority areas for security development associated with the top-level domains are identified. Then the set goals and the related levels are used as defaults, i.e., as basic levels of comprehensiveness and scope, for technical specialists to use in their proposals for implementing security practices. The use of business priorities as defaults for the level of comprehensiveness of security measures helps to simplify the problem statement process for ensuring security. The appropriate procedure is described in the document IoT Security Maturity Model: Practitioners Guide.

Example of using the model

We will now provide an example to better demonstrate how the IoT security maturing model is used for long-term planning of security tasks and setting priorities in the implementation of security measures.

We will look at a heating, air conditioning and ventilation (HVAC) system at an enterprise. The system includes process controllers, SCADA systems, human-machine interface consoles, as well as communication networks and systems. The consoles may be available from the outside for remote control purposes, and thus may be an entry point for a threat actor, as was the case in the notorious attack on the Target retail network.

A security maturity assessment can be conducted for any of the components or for the entire system. We will briefly consider the latter case.

From the point of view of a system integrator directly responsible for the operation of the system, hardening security is a priority for a functioning system. To achieve that, a vulnerability analysis is carried out, and software and firmware update management is implemented. An important point is the implementation of security monitoring (audit) and adoption of an incident response policy that is part of the enterprise's general policy. To ensure coordination of these practices, sufficient attention needs to be paid to *risk attitude*. All these practices require well-judged methods and tools to implement them (these methods and tools must be of the third level of comprehensiveness). The same level of assurance is required for maintaining the system's operational continuity, which must be maintained even during attacks, since a disruption of HVAC operation could cause significant business risks. Detection of a disruption and restoring operation is done in a system-specific way. In other words, attention also needs to be paid to a specific approach to implementing the practice (the *System* scope level).

The integrator also needs threat analysis and modeling, as well as a certain level of guarantee that the attack is not being launched from the contractors' side and is not the fault of vendors who are unwilling or cannot correct the vulnerabilities in the software or hardware they have supplied. These practices are implemented according to general system operation and access scenarios (the second (*Ad hoc*) level of comprehensiveness), and so is the management and control of management system access. Next, physical protection is required for individual components. Other practices are either not required at all or are implemented at the minimum level. While implementing security measures, you need to make sure that the system meets the standards and requirements adopted in the industry, meaning the compliance indicator will be industry specific.

The IoT security maturity target is summarized in the form of a diagram. The comprehensiveness and scope values for each practice are assessed separately.

The IoT security maturity target for the HVAC system

As well as creating an IoT security maturity target, we can assess the current security maturity status for a specific installation of the system, and then analyze the gaps between the current and target levels. A visualization of these gaps can be made on the basis of heat maps or a radar chart, though the most convenient method is to use bar graphs cross-hatched in different ways, where the hatching style shows the gaps in the scope value for each practice.

The following diagram is taken from the IoT SMM White Paper and shows the visualization of gap analysis at the subdomain level.

	Target	Current	Target Scope	Current Scope	Comprehensiveness Gap	Scope Gap
Security Strategy & Governance	3	1	1	1		
Threat Modeling & Risk Assessment	2	1	1	1		
Supply Chain & External Dependencies Management	2	0	1	1		
Identity and Access Management	3	1	1	1		
Asset, Change & Configuration Management	3	2	1	1		
Data Protection	3	1	1	1		
Vulnerability & Patch Management	3	1	3	2		
Situational Awareness	3	2	2	2		
Event & Incident Response, Continuity of Operations	3	1	3	2		

Visualization of gap analysis between the current and security maturity target
Source: IoT Security Maturity Model: Description and Intended Use White Paper

Why the IoT security maturity model is effective for decision making

The overall effectiveness of the protection system is determined by management decisions, which always entail a choice of: what means to use, how radically to change the composition of the system components, and what measures to spend resources on. This choice is extremely difficult.

Because the security priorities of the stakeholders differ, two factors are of importance: simplification of the planning process and organization of approvals. Structuring practices allows you to simplify the planning process and decision making. A well-organized communication mechanism between technical staff and risk owners on the business side helps establish a consistent approach to this planning. Representatives from the technical side will be able to objectively describe the tasks and purpose of the safety practices that can be implemented, and their potential effectiveness in relation to various types of risk. Using this experience, the business representatives prioritize security measures, set goals and create short- and long-term plans to achieve them.

This task of long-term planning is an economic one, similar to investing, or choosing an insurance program, or any other exercise with conflicting incentives. A modern approach to solving such problems involves the use of so-called nudge – building a choice architecture that supports effective decision making in a particular area. IoT SMM with the established process of forming an IoT security maturity target is a framework for the choice architecture (or simply nudge) in the field of information security of the internet of things, which allows you to take the first step (as well as the second, third and so on) in the task of building a secure system, whether it's for large-scale manufacturing or a fitness bracelet.

The use of the maturity model thereby allows us to optimize the formulation of the security problem, i.e., to determine the level of "sufficient security", to assess and plan the amount of work that needs to be done to achieve it with the required detail, starting from the level of security domains down to individual practices.

It may seem counterproductive for a company that produces solutions and services to protect against cyberthreats to use the IoT security maturity model for a rational approach to choosing protection measures, or a rational rejection of them. However, that's not the case. It has been proven that, by measuring the intentions of people, you can influence their actions. If you ask people about their intentions, they are more likely to act in accordance with their answer. The IoT security maturity model as a choice architecture pushes people who make decisions about business development towards using security solutions and paves the way for safer development of the internet of things.

P.S.

We started working on the IoT Security Maturity Model (IoT SMM) as part of the Industrial Internet Consortium (IIC) in March 2017. The Security Applicability subgroup, which deals with issues of applying security practices to real-world IoT applications, had already begun looking at the maturity model, though it was namely in the spring of 2017 that the actual approach now described in the documents appeared. About a year later, the basic IoT Security Maturity Model: Description and Intended Use document was released (the current version is available here). In February 2019, a comprehensive user guide was released entitled IoT Security Maturity Model: Practitioners Guide (also available for download).

Appendix.
Examples of problem statements for security provision

In this Appendix, we provide examples of problem statements for the provision of security for two fictional participants involved in the security process at an imaginary industrial facility – the owner (or operator) of the industrial facility and the manufacturer of software or equipment used at the facility. We want to emphasize that these examples are not universal, although we tried to ensure the hypothetical owner and manufacturer were typical of their category. We hope these detailed examples will help business participants to better understand the arguments for making security decisions for each of them.

For the owner or operator of an industrial enterprise

- Identify the company's need to improve the facility's security
  - Define how relevant different types of threats are to the enterprise:
    - Is there a threat of a targeted attack by another state's secret services?
    - Could the enterprise's systems be of interest to cybercriminal groups?
    - Could ordinary criminals benefit from using cybertools to target the enterprise's assets?
    - Could an attack on the enterprise's systems be ordered by unscrupulous competitors?
    - Could the enterprise's systems or staff become an intermediate target in an attack against partners/clients?
  - Define the scale of possible reputational and financial damage and legal risks to the company and its staff in the event of a successful cyberattack against various information systems and automation systems.
  - Define the range of security requirements from regulatory bodies and assess the risks of non-compliance.
- Formulate the security goals that must be achieved to ensure continuous operation of the enterprise and receipt of planned earnings.
  - Ensure the regulator's mandatory requirements are met.
  - Protect the information and automation systems that are most important to the enterprise's business and production process from attacks by cybercriminal groups (launched to steal money or receive ransom to unblock systems).
  - Provide maximum protection for enterprise systems necessary to ensure staff safety and environment protection from all possible actions by potential cybercriminals.
  - Protect against the legal risks associated with potentially insufficient protection from new and sophisticated threats by delegating the responsibility for protection to a third party (e.g., the developer of the tools or the supplier of protection services).
  - Transfer the financial risks associated with insufficient protection from selected types of threats and the implementation of all other types of cyberthreats for which no protection was envisaged to a third party (e.g., to an insurance company).
- Set deadlines for each formulated security goal.
- Define the resources that the company is ready to allocate to achieve the set security goals, plan resource allocation to achieve the set goals by the stated deadlines.

- Select the processes, measures and means of implementing the stated security objectives in a timely manner, taking into account the planned budget allocation.
  - Conduct an inventory of the organization's IT and OT assets.
  - Classify IT and OT systems in terms of their importance to the business, safety of the production process, taking into account the requirements of regulators.
  - Conduct a technical audit of the organization's information security status.
  - Organize the process of analyzing information about vulnerabilities in IT and OT systems, the process of prioritizing security patches and their timely installation.
  - Install adequate security tools on all the enterprise's most important IT and OT systems.
  - Organize an information security detection and response service, using either the enterprise's own resources or the relevant services of an external provider.
  - Organize the process of training employees in cybersecurity basics.
  - Organize a process to ensure suppliers and contractors fulfill the requirements of the company's information security policy.
  - Organize the process for testing and certification of new IT and OT systems before they are purchased for the enterprise's needs.
- Since the implementation of each of these measures will require additional expenses in one form or another, determine priorities, the roadmap for implementing the selected measures and plan the allocation of resources for each measure.

For the software and hardware manufacturer

- Determine the company's need to improve product safety:
  - Determine the scale of possible reputational and financial losses in the event of a product being compromised during an attack on a single client and in the event of a large-scale attack on many clients. Define what acceptable and unacceptable risk is.
  - Determine the degree of influence regulator security requirements have in major territories/markets.
  - Determine the competitive advantages that can be achieved by enhancing the safety of the product being developed and assessing the potential benefits of their implementation.
- Formulate security goals that, if achieved, will have a positive impact on product sales.
  - Ensure a sufficient degree of trust in the company and product in terms of security:
    - Protect the product from possible compromise during its development and delivery (guarantee the safety of the product development and delivery process, ensure the inability of third parties – component suppliers, distributors, integrators – to affect security);
    - Ensure the security of user/customer data – both on the product side and on the developer infrastructure side;
    - Demonstrate a quick response to information about security issues found in the product (release and delivery of security fixes) or in the product development/delivery infrastructure;
    - Demonstrate that the development company follows the best practices for secure product development;
    - Demonstrate compliance with regulatory requirements;
    - Revise marketing policy to highlight the company's focus on security issues.
  - Assess the current state of product security on the client side:
    - Track the installation of security updates for the product;
    - Track the appearance of information about attacks on clients of the product;
    - Track information about new vulnerabilities in the product environment on the client side (for example, in supported OS).

    - Help customers ensure safety during operation of the product:
        - Reduce the ability of cybercriminals to use the product to attack a client's systems and infrastructure (the product should not be a convenient tool in the hands of attackers or a weak link in the customer's infrastructure);
        - Increase protection of the product on the customer side against attempts to compromise it (the product should not be an easy target for an attacker);
        - Increase awareness of customers and partners (e.g., integrators involved in product implementation and maintenance) about potential security threats;
        - Help customers install product security updates in a timely manner;
        - Motivate customers to eliminate security gaps in their infrastructure and environment;
        - Offer professional security services to customers.
- Set deadlines for each of the stated security objectives.
- Determine what resources the company is ready to allocate to achieve its security goals, plan the allocation of resources in order to achieve goals by a given date.
- Select the processes, measures and means of implementing the stated security objectives in a timely manner, taking into account the planned budget allocation.
    - Introduce the practice of targeted threat analysis and setting security objectives as part of the process of collecting and analyzing requirements for all new products and new versions of existing products.
    - Implement a procedure for analyzing the safety of the architecture when developing a product at its design stage and analyzing the impact on product security of all proposed architectural changes in the product development and support process.
    - Assign the position of a 'security architect' who is responsible for making all strategic decisions and architectural changes that affect product security.
    - If necessary, look at the issue of architectural changes that increase the product's resistance to the actual attack vectors.
    - If necessary, add additional technical security equipment to the product.
    - Implement the practice of static and dynamic code analysis.
    - Implement the practice of learning to develop secure code for the entire development team or its key participants.
    - Organize an internal product safety testing process as part of the development process.
    - Assign the position of a 'security champion' who is responsible for the implementation of technical measures and means of testing and ensuring security and making tactical decisions to ensure product security.
    - Organize a product security testing process by an external research team.
    - Organize the process of collecting and analyzing information about vulnerabilities detected in the product.
    - Organize the process of collecting and analyzing information about vulnerabilities in third-party components.
    - Organize the process of assessing the security properties of third-party components when choosing a product implementation.
    - Implement a process of setting security requirements (and monitoring their implementation) for suppliers of software and hardware components.
    - If necessary, consider switching to other, more secure components and third-party technologies.
    - For equipment manufacturers, consider moving to a more secure/attack-resistant platform (hardware, OS, runtime, etc.).
    - Organize the process of collecting and analyzing information about attacks on customers that use security weaknesses and product vulnerabilities.
    - Organize the process of developing and releasing security patches.

  - Organize the process of notifying customers about detected security issues and their fixes.
  - Organize the process of delivering security patches to customers.
  - Make changes to the company's PR and marketing strategies to properly present information about the properties and problems of product security.
  - Organize training on the security of suppliers and customers.
  - Solve communication problems with technology partners and customers (for OEM vendors) that have different views on the need and importance of disclosing information about security issues.
- Taking into consideration the fact that each of the measures will require additional expenses in one form or another (the allocation of additional human resources, the acquisition of special automated analysis tools, or the purchase of services from security service providers), determine priorities, a roadmap for implementing the selected measures and plan the allocation of resources for each measure.

Kaspersky Industrial Control Systems Cyber Emergency Response Team (Kaspersky ICS CERT) is a global project of Kaspersky aimed at coordinating the efforts of automation system vendors, industrial facility owners and operators, and IT security researchers to protect industrial enterprises from cyberattacks. Kaspersky ICS CERT devotes its efforts primarily to identifying potential and existing threats that target industrial automation systems and the industrial internet of things.

Kaspersky ICS CERT ics-cert@kaspersky.com