GENEVA
DIALOGUE
ON RESPONSIBLE
BEHAVIOUR
IN CYBERSPACE

SECURITY OF DIGITAL PRODUCTS AND SERVICES: REDUCING VULNERABILITIES AND SECURE DESIGN

Good practices

Document for Comments
10 November 2020

CONTENTS

1. Introduction

This document is a **result of the partner inputs** during the fourteen discussions of the Geneva Dialogue on Responsible Behaviour in Cyberspace (GD) in 2020 (May-November 2020), and a number of written contributions submitted in that period. As part of the GD, partners agreed to focus on *defining secure design* and *vulnerability management*, as well as *associated implementation practices*.

Based on these discussions, this document sets out the **definitions related to secure design** that have been **agreed upon by the partners** and highlights some of the **best practices** that the partners are following. The document also emphasises the organisational and planning resources and processes needed to implement those best practices, and lists some of the key resources recommended by partners.

This document is primarily targeting those developing software, hardware, cloud, and system solutions – primarily companies[1], but other institutions and organisations as well. Best practices and certain challenges can also be useful to both regulators and customers[2] to better understand the environment in which digital products are being developed and secured.

During the work on definitions and best practices in 2020, the need for commonly shared baseline requirements repeatedly came up. Partners discussed that existing technical standards are great resources, yet are also too complex and their applications too fragmented regionally to make a real difference to cybersecurity. Therefore, the discussion will continue further towards developing an outlook on establishing common baseline requirements for secure design, a task that partners decided to take on in 2021.

The GD is an initiative led by DiploFoundation and the Swiss Federal Department of Foreign Affairs (FDFA). The partners that contributed to this year's discussions are BI.ZONE (Sber Group), Cisco, Ensign Infosecurity, FireEye, Huawei, Kaspersky, Microsoft, Papua New Guinea ICT Cluster, SICPA, Siemens, SwissRe, Tata Consultancy Services, UBS, Wisekey, and Vu Security. In addition, a number of observers and related organisations have contributed to the discussions, in particular Cybersecurity Tech Accord, the World Economic Forum; the Center for Security Studies (CSS) at ETH Zurich, the Graduate Institute of International and Development Studies (IHEID), the Swiss Digital Initiative Foundation (SDI), as well as the ICT4Peace Foundation

The well-diversified partner base of the GD has contributed to its alignment with other multi-stakeholder and industry initiatives and processes in the area of digital trust and security, such as the Charter of Trust, the Cybersecurity Tech Accord, and the Paris Call, in particular its principle No. 6 on lifecycle security.

For more information about the Geneva Dialogue, and to provide comments to the document, visit https://genevadialogue.ch/.

[1] Throughout the document we refer to such companies interchangeably as producers or suppliers, which are the synonyms for vendor, or manufacturer. While we are aware of possible nuances, those nuances rarely have direct relevance to the practices presented at this stage.
[2] Similarly, throughout the document we interchangeably use the terms customer and client, which are synonyms for consumers and users. Again, nuances don't have direct relevance to practices presented at this stage.

2. Secure design concepts and terminology

Reducing vulnerabilities can be achieved by factoring in security from the design phase, and throughout the product development life cycle (for software, hardware, systems, and services). Applying security into design requires an evolutionary, agile approach, that allows practices to adjust to the evolving and expanding threat landscape with millions of new vulnerabilities and malware pieces discovered. This presumes an upgrade of internal processes, resources, and capabilities.

These topics are dealt with by several related concepts, which continue to evolve, such as security by design, security by default, security development life cycle, and trustworthiness. While all are connected to security aware software development, operational management, and threat mitigation practices; there are no agreed common definitions of those within the industry. The terms are often used interchangeably, though they may have different meanings. The GD is offering its contribution towards establishing a common understanding of the concepts, along with some related corporate practices.

2.1 Security by design

Security by design is about *designing with security in mind*: addressing risks from an early stage and throughout the product development lifecycle. It may be understood as *designing with security controls from the beginning.*

It applies to the design, development, deployment, and maintenance of software, hardware, and services; and also to system integration, through secure process of integration, and security tools like firewalls or monitoring tools.

Security controls should be implemented at all stages:

- *Before the product hits the market*: through designing the product based on threat modelling and risk assessment, and developing and testing the code/design with security engineers.
- *After the product has hit the market*: by putting in place vulnerability management and disclosure processes, by considering security when the product is being deployed in various environments, and in support and maintenance.
- Regular, annual independent vulnerability assessments that assess whether processes are still current, and other checks. Ideally, security controls should focus on prevention rather than detection that identifies security issues post factum.

The process must be comprehensive and inclusive: considering engineering, security, business, and human resources aspects; and involving engineers, security professionals, and C-level management.

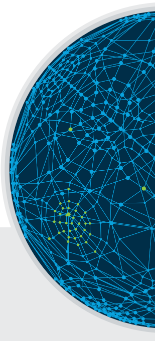
Examples

The Cybersecurity Tech Accord, in its first principle, invites partners to commit to 'design, develop, and deliver products and services that prioritize security, privacy, integrity and reliability, and in turn reduce the likelihood, frequency, exploitability, and severity of vulnerabilities'. In its fourth principle, it invites partnerships among stakeholders across proprietary and open source technologies to improve technical collaboration, co-ordinated vulnerability disclosure, and threat sharing to minimise the amount of malicious code being introduced into cyberspace.

2.2 Security by default

Security by default builds on security by design: delivering the product preconfigured in a secure way. It may be understood as making security settings an opt-out product function.

> '*Security by default*: Adopt the highest appropriate level of security and data protection and ensure that it is preconfigured into the design of products, functionalities, processes, technologies, operations, architectures, and business models.' (Charter of Trust, Principle 3)



Producers take the responsibility of managing security controls and reducing reliance on customers with a presumption that customers will not participate in enabling, managing, and controlling security settings. Certainly, customers may reclaim responsibility and apply security configurations differently, based on their own risk-based decisions.

Security by design is a prerequisite for security by default, since the configurability of a product's security must be added in the design phase. Ideally, these are two parallel processes. However, in cases like industry integrations or brownfield development (when new products must build on legacy systems), security by default may need to be implemented through adequate cost-effective security controls, or configuration of add-ons to (insecure) legacy.

Examples

The Charter of Trust's ten principles provide a holistic view to establishing trust in digital technologies, particularly in cybersecurity. It covers aspects such as ownership for cybersecurity, baseline organisational prerequisites, baseline product level prerequisites, third party risk, and training. It also provides a platform for exchange on these important matters. A particularly useful output of principle three is a set of baseline requirements for security by default, which include actions such as unique identification secure onboarding, secure login credentials procedures, backup features, security documentation, secure update processes, factory reset, and more (Achieving Security by Default for products, functionalities, and technologies).

2.3 Security development life cycle

Security development lifecycle (SDL or SDLC) is a model of implementing security by design and security by default by introducing security into each phase of the product development life cycle. SDL ensures that security risks are modelled and understood by producers as well as customers, in order to allow timely decisions that help manage and reduce risk.

SDL as a concept was particularly developed for software, but is now also being applied to cloud services, and Internet of things (IoT) devices.

Examples

The Microsoft Security Development Lifecycle is 'a set of practices that support security assurance and compliance requirements'. The SDL helps developers build more secure software by reducing the number and severity of vulnerabilities in software, while reducing development costs. The SDL is typically thought of as an assurance activity that helps engineers implement 'secure features'. In addition, Microsoft's Operational Security Assurance (OSA) model is the SDL for cloud and services. Because the cloud changes constantly, it has particular emphasis on security monitoring, encryption, and access. Two other resources are offered related to cloud: Seven Key Principles of Cloud Security and Privacy and Security design principles for cloud applications. Microsoft also offers Seven Properties of Highly Secure Devices.

The Cisco Secure Development Lifecycle (CSDL) is a repeatable and measurable process designed to increase Cisco product resiliency and trustworthiness. The combination of tools, processes, and awareness training introduced during the development life cycle promotes defense-in-depth, provides a holistic approach to product resiliency, and establishes a culture of security awareness. CSDL involves staff training, standards and principles, threat awareness, secure design, and vetted solutions. The CSDL has been mandatory for all Cisco products since 2013, it applies industry leading practices and technology to build trustworthy solutions that have fewer field discovered product security incidents. Security by default is part of design, in particular the product security baseline requirements.

2.4 Trustworthiness

The concept of trustworthiness can be tied to engineering secure systems and components.

> Trustworthiness of products is made possible ‘by the rigorous application of .. design principles and concepts within a disciplined and structured set of processes that provides the necessary evidence and transparency to support risk-informed decision making and trades’. (NIST SP 800-160, pp. X).

However, trustworthiness could also be viewed more broadly as a fresh perspective on secure software development. It relies on SDL, considers non-technical issues, such as internal processes and reputation (for instance, trustworthiness of a producer itself in addition to the security and trustworthiness of a products), as the Software Trustworthiness Best Practices White Paper (particularly Chapter 2) of the Industrial Internet Consortium (IIC) outlines.

Examples

The Kaspersky Global Transparency Initiative (GTI) is a practical implementation of trustworthiness. The GTI is a combination of practices to enhance the security of products and engineering practices (e.g. vulnerability management programme, third party validation and assessment), together with steps to increase assurance in software and enhance transparency and trustworthiness. Some of the elements include Transparency Centers for external source code examinations and review of the company’s software development processes and engineering practices. Kaspersky is also developing a cyber-immunity concept for the industrial infrastructure environment, which provides an alternative view to secure software development for industrial and especially critical systems. The concept promotes security as an inherent feature for operation systems when systems are built with security in mind by default. In practice, it presumes redesigning operation systems on a microkernel architecture (as opposed to the classic kernel-applications-security hierarchy inherent in unprotected systems). As a result, all actions in these cyber-immune operating systems are prohibited by default, the system only performs explicitly permitted operations that are defined by customers in detail, and therefore this configuration makes critical systems immune to traditional threats.

Huawei’s Trust Center is the application of its end-to-end (E2E) cybersecurity assurance system - an open, transparent, and visible security assurance framework. It is based on compliance with the applicable laws, regulations, standards of relevant countries and regions, and industry best practices embedded into process and baselines. The E2E cybersecurity system incorporates aspects from corporate policies, organisational structures, business processes, technology, and standard practices into 12 corporate processes and business modules: strategy and governance, laws and regulations, processes, research and development, supplier management, manufacturing and logistics, service and delivery, verification and certification, traceability, defect and vulnerability resolution, health and rescue, and audit.

Microsoft’s Trust Center emphasises handling customer data securely and in compliance with privacy and legal requirements. In particular, maintaining data integrity in the cloud is based on protecting against cyber-threats with built-in automation and intelligence; privacy and the ability for customers to control their data; compliance with national, regional, and industry-specific requirements, and auditing to verify technical compliance and control requirements (as outlined in Microsoft’s White Paper ‘Managing compliance in the cloud).

Discussion

Security by design should be applied to hardware and systems as well. Typically, each component of the system is tested on default settings. However, when components are put together to form a system, settings are changed and new vulnerabilities may emerge. Therefore, both pre and post implementation testing is necessary for systems, which should include security configuration reviews, vulnerability testing, etc.

3. Main elements of secure design

3.1 Threat modelling

What is threat modelling about?

Threat modelling is a structured approach to threat scenarios; an engineering technique to identify possible threats, attacks, vulnerable areas, and countermeasures that could affect the product or the related environment (network, architecture, etc.). It is a practice that allows development teams to consider, document, and more importantly discuss the security implications of designs in the context of their planned operational environment, in a structured fashion (Microsoft SDL). Threat modelling should be used to validate a design's security (Cisco SDL), and in environments where there is meaningful security risk (Microsoft SDL). It can be applied at the component, application, or system level (Microsoft SDL).

When is the right time to do it?

There are two distinct time frames in which threat modelling is conducted for products. In the pre-market phase, producers do the technical assessments of threats during the initial design of the new solution. In the post-market phase, threat modelling is regularly conducted through the entire life cycle of the product, including maintenance (updates and patches), until the product's end of life or end of support. Continuous threat modelling and risk assessment aims to address the evolving threat landscape, thereby progressively refining a threat model through evaluating new threats with quantitative measures, and adding them to the risk profile if they are significant. It also addresses risks that particular applications of products can bring to customers in different environments.

The pre-market phase primarily concerns the producers: software developers, product implementers, and cyber-threat intelligence teams. The post-market phase also concerns the customers - procurement teams and business users – as well as other cybersecurity professionals. However, whereas producers have clearer insights into how their product may create risks from design and function vulnerabilities, customers have a different appreciation of the threat environment because of the applied nature of the products. Therefore, continuous threat monitoring should be practiced both by producer and customers.

Who should be involved?

As threat modelling results in concrete figures and impact, and makes the value and the cost of security clear, many different departments of the company should be involved. Senior management should be involved, to decide on the level of acceptable risks, and endorse the allocation of resources and actions for mitigation of risks in the product development. On the operational level, developers, cybersecurity and threat intelligence specialists should be involved on a continued basis to monitor the threat environment – both from the information security and the product security perspectives.

Main approaches

Threat modelling and risk assessment in the design phase should be based on understanding threat vectors and risks for a particular product. The nature of threat actors also needs to be taken into account; whether it is tactics, techniques, procedures of possible adversaries, or any other unique vulnerabilities that pertain to the sector they are in.

Cisco recommends five steps for systems and services threat modelling: (a) identify assets, (b) diagram the system, (c) analyse threats, (d) perform risk management and prioritisation, (e) identify fixes. In practice, Cisco engineers follow the flow of data through a system and identify trust boundaries and inflection points where data might be compromised. Once potential vulnerabilities and threats are identified, mitigation strategies can be implemented to minimise the risk.

For software development, Microsoft suggests five major threat modelling steps: (a) defining security requirements, (b) creating an application diagram, (c) identifying threats, (d) mitigating threats, (e) validating that threats have been mitigated.

Industry environments, where it is important to look into the system as a whole rather than focusing only on its components; rely on the ISA/IEC 62443 standard on security capabilities for control system components (and particularly part 3 on system security conformance metrics). In addition, the industry systems, particularly the critical infrastructure, need to observe not just security risks (which may allow some time for analysis and reaction) but also safety risks (which require immediate reaction in case of a major incident).

Examples

There are a number of well-known threat modelling methodologies, covered in a review by the Carnegie Mellon University and suggestions by Cisco.

The CIA method (confidentiality, integrity, availability) is often used as a basic approach to define what needs protecting in an organisation. From the industry experience, the STRIDE (spoofing, tampering, repudiation, information disclosure [privacy breach or data leak], denial of service, elevation of privilege) is often seen as the most mature threat modelling approach; but is focused on the design process rather than customer implementation. As a result, it does not prioritise human safety outcomes which is a must in medical devices, for instance. The PASTA (Process for Attack Simulation and Threat Analysis) model has a particular business objective. The Attack Trees method requires a deeper understanding of the system, while the Security Cards approach requires minimum prior knowledge but a lot of effort between teams within the company. The Common Vulnerability Scoring System (CVSS) helps measure the effectiveness of threat modelling through producing standardised scores for application vulnerabilities, IT systems and elements, and IoT devices. The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE), and the Risk Management Guide for Information Technology Systems by the US National Institute of Standards and Technology (NIST), are also the two useful tools for risk assessment.

There are multiple sources that cover the latest threats. The Cisco TALOS threat intelligence, and external sources such as the common weakness enumeration (CWE), common attack pattern enumeration and classification (CAPEC), and the Open Web Application Security Project (OWASP) are some good examples.

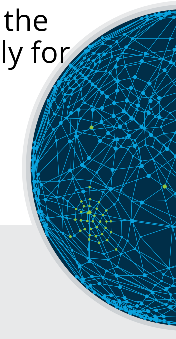
Basic threat modelling can be performed without specific tools and external costs, in a brainstorming session. However, for larger enterprises and more complex systems software tools and threat modelling as a service (TMaaS) may be necessary to manage the process and the related data. Microsoft provides its Threat Modeling Tool, aimed particularly at non-security experts and developers. Cisco has developed its own threat modelling tool which facilitates the process by exposing applicable threats based on the developers' data flow diagram and trust boundaries.

Customers, however, may look for different approaches, akin to military analysis of any given terrain. One useful threat modelling approach is terrain analysis (also presented by the MITRE paper).

The role of customers

Threat models very much depend on specific customers and the way products are implemented and utilised by them. Customers have different levels of necessary security requirements and acceptable risk levels. For instance, a certain vulnerability may be acceptable for a producer, as well as some of its customers, while other clients may find that risk unacceptable due to their business models. In addition, products within a network or an integrated environment are the largest source of vulnerabilities: they interact and are configured together to work as a holistic environment, which creates additional risks.

Direct co-operation with customers is necessary to ensure taking their perspectives into account from early stages, and to document all risks together, when possible. Co-operation enables continuous monitoring of the progress of implementation of controls to reduce risks to acceptable levels for the clients. Certainly, producers may not be able to co-operate directly with all customers (particularly for mainstream products) – which is where security by default plays a particularly important role.



Challenges

However, it is not possible to address all possible threats, otherwise it becomes impossible to allocate resources effectively. Quantifying risks and creating a heat map of aggregated threat actor tactics, techniques, and procedures (TTPs) – such as by applying MITRE ATT&CK and MITRE Shield frameworks, as well as the CVSS model - enables prioritisation of risk in product development or during the product life cycle. It enables the producer to see what the 'low hanging fruits' for adversaries are and address them, and then move to addressing vulnerabilities that are harder to exploit.[3]

Threat modelling requires observing the context in which a product or service will be used. This requires the involvement of customers, who often lack necessary insight of the process flow of the application, its 'ingredients', system architecture, and third party components. This calls for greater transparency from producers. The Software Bill of Materials (SBOM) may be a model to follow and contribute to, as a process that aims (through white papers for the industry) to address transparency in software, and clarity on how to address threats and supply chain risks. On the other hand, in most cases clients (particularly customers) do not have the necessary resources, capacity, or capability to continuously perform threat modelling, and producers may assist them through capacity building activities.

Producers of widely used products face a particular challenge in threat modelling based on the context in which the product will be used. One current approach is to put a clause in the contract about what kind of services can be run on certain products. This puts the expectation on customers to do their own risk management, risk analysis, and risk prioritisation exercises for particular applications of a product. It has become increasingly important to clarify mutual roles and responsibilities with clients and customers. This can be done by appreciating the RACI (responsible, accountable, consulted, and informed) responsibility assignment matrix, such as the Shared Responsibility Model advocated in Cloud Service Providers.

3.2 Supply chain and third party security

A common industry practice is to incorporate both proprietary and open source third party[4] components (TPC) into product offerings - whether it's software, cloud services, devices, or integrated systems. Poor implementation of security standards throughout the supply chain can create vulnerabilities to the broader ecosystem.

Therefore, third parties constitute one of the leading cybersecurity risks. Protecting product development processes from unintended or malicious third party interference (whether from other producers supplying vulnerable components or adversaries interfering with updates or patches) requires a whole-of-sector, multi-actor approach to supply chain risk management (SCRM).

The 'Common risk-based approach for the Digital Supply Chain' of the Charter of Trust's second principle 'Responsibility through the digital supply chain'; suggests that one of the basic standards for supply chain – besides identity and access management, and encryption – needs to be continuous protection: 'companies must offer updates, upgrades, and patches throughout a reasonable life cycle for their products, systems, and services via a secure update mechanism'.

Discussion

The concept of 'duty of care' for supply chain is increasingly being discussed in various fora. It suggests producers need to ensure that their own components are safe and up-to-date. In addition, suppliers should be transparent about what third party components they use in their products, and provide information about security assurance of third party components.

[3] Not all vulnerabilities need to be exploited by adversaries: some may cause faults in operations, or present risks in other ways.

[4] The term 'third party' is used to denote a party of the supply chain that is not in a direct relationship with the producer, and which contributes to the criticality of the function or essential service of products, regardless of whether it is a third party in the literal sense, or a fourth, fifth or further down the line.

A risk-based approach

The essence of the risk based approach for the digital supply chain, according to the Charter of Trust's Common risk-based approach for the Digital Supply Chain, is built around three main components: baseline requirements, supplier criticality, and verification.

- *Baseline requirements* 'are common for all digital suppliers and define the fundamentals that a supplier must address in order to ensure the cybersecurity foundations for their product/service'. In particular; data protection, security policies, incident response, site security, access, intervention, transfer and separation, integrity and availability, support, and training.
- *Supplier criticality* considers the fact that different suppliers may have different levels of criticality for the purchaser; the level of perceived criticality depends on risk factors, i.e., on the context viewed by the purchaser.
- *Verification* of baseline requirements is dependent on the criticality of the supplier.

A contract between a third party supplier and the customer should clearly define timelines and accountability around patching and other security relevant duties.

Examples

When it comes to baseline requirements, one corporate practice suggests adding transparency of third party components to baseline requirements. Another practice suggests defining baseline requirements for each third party, depending on what they supply (for instance, for manufacturing they may be required to add specific access controls to the building). A third practice suggests embedding baseline requirements to contracts. With regards to supplier criticality, one practice suggests shaping requirements based on particular clients, and linking this to their risk appetite and related threat modelling.

In practice, verification of compliance to baseline requirements by the supplier ranges from (a) for low criticality; self-declaration by the supplier, through accepting terms and conditions, (b) for medium criticality; self-assessment through a questionnaire, (c) for high criticality; evidence by the supplier as documented proof. Third party audits, as well as initial and periodic compliance checks, are also part of some corporate practices. Microsoft SDL suggests validation to be considered as well, depending on an organisation's risk appetite, the type of component used, and the potential impact of a security vulnerability. Bigger companies, such as Cisco, that include huge numbers of TPC may have a dedicated SCRM team which provides tests, controls, and audits.

TPC management

Typically, management of TPC by producers spans across the entire product lifecycle, and thus should be embedded in corporate processes and organisations. SAFECode whitepaper 'Managing Security Risks Inherent in the Use of Third-party Components' outlines four steps of the TPC life cycle (***figure 1***):

1. Maintain list of TPC
2. Access security risk (and auditing)
3. Mitigate or accept risk
4. Monitor for TPC changes

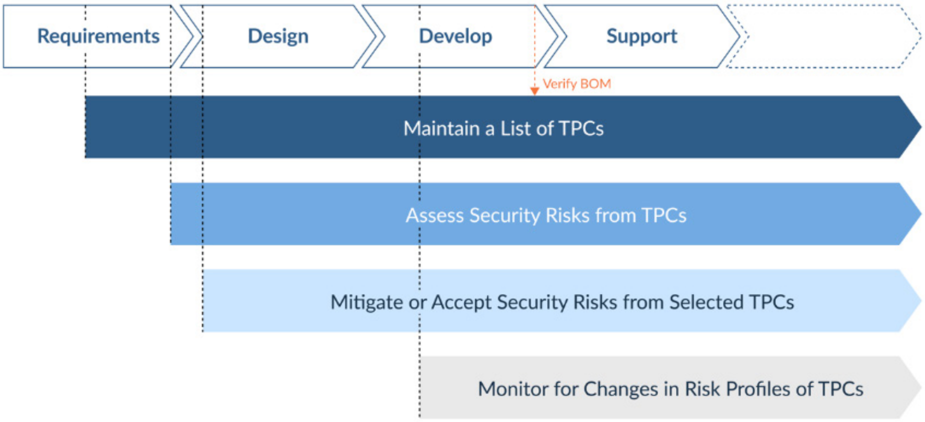

Requirements
Design
Develop
Support
Verify BOM
Maintain a List of TPCs
Assess Security Risks from TPCs
Mitigate or Accept Security Risks from Selected TPCs
Monitor for Changes in Risk Profiles of TPCs

***Figure 1:** TPC life cycle follows the SDL. Source: SAFECode Source, pp. 13.*

Examples

One of the key elements in managing security risks is identifying TPC and establishing and maintaining the product bill of materials (BoM)[5]. Microsoft SDL suggests having an accurate inventory of third party components, and a plan to respond when new vulnerabilities are discovered. To gain visibility into its potential third party software security threats, Cisco uses integrated tools, including a Central Repository of Intellectual Property. The repository internally tracks products using third party software through a centrally maintained repository, with third party code distributed outside the company. This allows for rapid identification of all affected Cisco products in case a vulnerability is found in TPC.

Another practice suggests maintaining an inventory of both first and third party components, including open source ones. BI.ZONE also warns that there is a particular risk with open source solutions and libraries, and suggests using tools for checking software dependencies and vulnerabilities, such as the OWASP Dependency-Check or the Checkmarx Open Source Analysis. To maintain inventory of own products and prevent eventual modification of the products on the way to customers, Cisco issues an unique ID for each device it creates (like model and serial numbers).

A continuous monitoring of TPC involves ensuring they are up-to-date, and that suppliers react to discovered vulnerabilities. For this, Cisco uses scanning and decomposition - internal tooling to inspect source code and images, to improve third party repository accuracy and completeness. Cisco also provides quick responses to third party vulnerabilities, through notifications to the supplier. It is particularly important to monitor for TPC that have reached end-of-life (EoL), and thus are left without support. To avoid their own products to be supplied after EoL, including through grey markets, Cisco performs a supply chain control for disposing the product over the EoL.

Transparency

Continuous communication between the supplier and customer allows the customer to express concerns about different parts of the software (e.g. encryption methods, source code elements) in early stages of design or deployment. Suppliers should ensure transparency of their products, including the TPC used. For instance, when a company buys a firmware for its facility, it should know vulnerable points, which should also allow it to know the total cost of ownership and maintenance.

Examples

One practical instrument is the BoM, which allows customers to understand the elements of the product. Transparency Centers, operating under the framework of Kaspersky's GTI, provide a three-layer approach to executive briefings about functionality of software, company's engineering and data management practices, including the TPC. Whatever the tool, it is important to have it in a user-friendly form, so it is understandable by many, instead of only by very advanced users and regulators.

The Huawei Cyber Security Transparency Centre (HCSTC) aims to address European objectives and needs, as well as sharing important technical information on its solutions, to prevent cybersecurity threats and vulnerabilities. It is a showcase of Huawei's E2E cybersecurity practices, from strategies and supply chain to research and development (R&D) and products. But more importantly, it provides a product security testing and verification platform. It provides a testing and validation environment and platform for products and solutions to customers, including white box (source code) and black box security.

Challenges

No transparent and universally developed criteria exists for managing cyber supply chain risks and assessing producers. In addition, different types of security requirements are being created and discussed in different jurisdictions, which could result in further fragmentation.

[5] BOM is a comprehensive inventory of all parts and components needed to build a product.

There is a general lack of information exchange about supply chain risks, both in the private and public sector, particularly for critical technology, sensitive products or sensitive areas of applications. It is important to establish processes that ensure that sensitive information can be shared, even though this might not be easy due to legal complexities and limitations. It is also important to enhance independent testing or certification of third party products for cyber risks. Finally, geopolitics increasingly impacts the global supply chain, including information sharing processes, which could potentially create disruptions.

3.3 Secure development and deployment

One of the key elements of security by design is the secure development and deployment of products. Security should be embedded into all components of the 'pipeline': from development, building and testing, releasing and deployment, to validation and compliance.

Development and deployment

In case of software development, automated 'continuous integration' and 'continuous delivery' (CI/CD) pipelines should include security rules and checks, such as responsible coding, scanning source codes for vulnerabilities, dynamic analysis of code, checking dependencies for vulnerabilities, and unit tests with security checks.

Examples

Cisco SDL, for instance, pays attention to secure coding standards - uniform set of rules, guidelines and best practices for programmers, training, and experience - to ensure threat-resistant code. Cisco also leverages a growing number of vetted common security modules – libraries that enhance the engineers' ability to confidently deploy security features (e.g., CiscoSafeC, CiscoSSL, and other libraries that focus on secure communications, coding, and information storage).

At the same time, it is necessary to improve developers' secure coding skills and attention through training and exercises – including games. The Secure Code Warrior, for instance, is a training tool for developers which gamifies secure coding's good and bad practices, and encourages knowledge retention and transition among developers.

Risks hidden in the build environment, including all corresponding toolchains, should also be reduced, to avoid unauthorised and unreviewed changes of build parameters that are responsible for stack protection (such as StackGuard or Control Flow Guard) decreasing the level of software security.

Security by default should be taken into consideration during development and deployment. In particular, specific security controls should be developed, and set to high default security levels before deployment.

Security testing and validation

Security testing is another essential element in secure design. Continuous and automated testing reduces risk vectors and helps resolve vulnerabilities before the product goes out to the market. However, automated tools are still limited in effectively finding vulnerabilities.

In case of software, testing for vulnerabilities and validation involves static and dynamic testing, vulnerability assessment, fuzzing (inputting massive amounts of random data attempting to crash the subject), and penetration testing. Certainly, the integrated systems need to be tested as well, since the integrations of many products (even well-tested ones) and their configuration within a particular network or environment is a major cause of vulnerabilities in a system.

Examples

Microsoft performs Static Analysis Security Testing (SAST) for analysing the source code prior to compilation, which ensures secure coding policies are being followed. SAST is typically integrated into the commit pipeline to identify vulnerabilities each time a software is built or packaged. In some cases, it is integrated into the developer environment, to spot certain flaws such as the existence of unsafe or other banned functions and replace those with safer alternatives (as the developer actively codes). In Cisco, development teams run SAST with security checks enabled - internal analysis, field trials, and limited business unit deployments. SAST detects source code vulnerabilities, targets potential buffer overflows, tainted inputs, integer overflows, and enables fixing high-priority issues.

Dynamic Analysis Security Testing (DAST) further performs run-time verification of the fully compiled or packaged software checks functionality, that is only apparent when all components are integrated and running. Microsoft typically achieves this using a tool or suite of prebuilt attacks, like web app scanning tools or fuzzing, that specifically monitor application behaviour for memory corruption, user privilege issues, and other critical security problems.

Cisco further validates its products' ability to withstand attacks with a minimum of three regiments of Cisco SDL Vulnerability Testing: protocol robustness testing, common attacks and scans by common open source and commercial hacker tools, and web application scanning. For this, Cisco has developed its Security Test Package as a single, easy-to-install collection of tools; custom tests occasionally supplement standard test suites, such as dedicated penetration testing and security risk assessments. Vulnerabilities found during testing are triaged by product teams and reviewed by Cisco's Product Security Incident Response Team (PSIRT). Microsoft also performs penetration testing, to uncover potential vulnerabilities resulting from coding errors, system configuration faults, or other operational deployment weaknesses; and find that such tests typically find the broadest variety of vulnerabilities.

From a systems integration perspective, it is commonly advocated for pre-implementation and post-implementation testing to root out vulnerabilities that occur as a result of implementation or configuration missteps. Product testing then typically documents the results of default settings.

Third party involvement in conducting tests is common practice. This may include bug-bounty programmes for vulnerabilities, penetration testing, or other tests by external teams such as ATT&CK Evaluations by MITRE.

In a critical infrastructure environment, testing and validation supported by independent third party certification may be particularly important. However, there are a number of challenges with this approach: (a) lack of clarity on concerns to be addressed, (b) challenge to choose a credible third party trusted by the market, (c) limited effectiveness of one-off external audits (dynamic products updates would require frequent audits, which demand resources). A third party audit of business and software development processes (such as the Service Organization Controls – SOC2 - Reporting Framework), rather than products, may be better suited: it provides assurance in the products through evaluating security controls implemented in the product development and release of the updates. Importantly, while companies should be transparent about results of such audits, they should also ensure those are presented and explained to general customers in plain language, rather than lengthy and technical reports.

3.4 Vulnerability processes and support

Due to the highly complex process of developing digital products, and the interdependence of the products of various producers, it is almost impossible to prevent vulnerabilities. While applying security elements throughout design and development can reduce the number and criticality of vulnerabilities; companies should implement security measures throughout a product's life cycle. In particular, companies should set up processes to react on discovered and reported vulnerabilities and mitigate related risks by developing and distributing fixes and supporting customers with deployment.

Examples

The 2015 report of the UN Group of Governmental Experts (UN GGE), in its article 13(j), invites states to 'encourage responsible reporting of ICT vulnerabilities and share associated information on available remedies to such vulnerabilities'. Principle 6 on 'Lifecycle Security' of the Paris Call on Trust and Security in Cyberspace, calls for considering the security of digital products and services throughout their life cycle and supply chain. Similarly, principle 2 of the Charter of Trust on 'Responsibility through the digital supply chain' stipulates continuous protection: 'Companies must offer updates, upgrades, and patches throughout a reasonable lifecycle for their products, systems, and services via a secure update mechanism'.

There are several related terms used by various actors: *vulnerability response, management*, *reporting*, *handling*, and *disclosure* – and particularly *co-ordinated and responsible disclosure*. This sometimes causes confusion and misunderstanding, particularly as the terminology differs across stakeholders (e.g. the security community uses one set of terms with particular understanding of what they mean, while policymakers borrow some terms but interpret them in a different way). The Geneva Dialogue is offering its contribution towards establishing a common understanding of the terminology, along with some related corporate practices.

Vulnerability response *(VR)*: An overarching term (though possibly ambiguous), in line with the Carnegie Mellon computer emergency response team (CERT) guidance definition:

> 'Vulnerability Response (VR) is the overall set of processes and practices that deal with the existence of vulnerabilities in systems. VR encompasses everything from reducing the introduction of vulnerabilities as part of a Secure Development Lifecycle (SDL) through the remediation of deployed vulnerabilities via patch deployment.'

VR includes testing and validation before the release of the product, as well as vulnerability management and vulnerability disclosure after the product is released. In the context of security of digital products, VR shouldn't be confused with ICT incident response.

Vulnerability management (VM) refers to producer practices and security controls to ensure that products are running with the latest security updates and mitigation. VM is the day-to-day management of the process of remediating reported or disclosed security flaws in one's own product, including monitoring and mitigating the effects of vulnerabilities in third party components used. This understanding is in line with the Carnegie Mellon CERT definition:

> '*Vulnerability Management* (VM) is the common term for tasks such as vulnerability scanning, patch testing, and deployment. VM practices nearly always deal with the output of CVD practices, not the inputs. VM practices focus on the positive action of identifying specific systems affected by known (post-disclosure) vulnerabilities and reducing the risks they pose through the application of mitigations or remediation such as patches or configuration changes.'

VM is somewhat related to the 'patch management' process; how a company manages different security flaws in its own products. In a narrow sense, it includes:

- Scanning products and systems affected by reported or disclosed vulnerabilities.[6]
- Working with other teams on ensuring the fix or patch is prepared (developing fixes is a cross-team effort).
- Fix and patch testing.
- Distributing the update, and supporting the affected customers where applicable (however, applying fixes is the responsibility of customers).

In a broader sense, VM also includes:

- Receiving notifications and reports about discovered vulnerabilities (from other CERTs, public disclosures, or directly from researchers) in a company's own products or third party components.
- Analysing the vulnerability, verifying its validity, its possibility to be exploited, thus the related risks and need for patching.
- Communication and co-ordination of risk handling related to the reported vulnerabilities.
- Issuing notifications to CERTs, customers, and publications of security advisories (in some cases also without any need for fixes or when fixes aren't ready).

[6] Due to the fact that, when combined, multiple products may cause vulnerabilities in a system, there should be security validation considerations which look into how vulnerabilities emerge in a system, and how they can be exploited. Thus, VM should have a strong link with other VR elements.

Typically, the product security teams – PCERT or PCSIRT - lead and are in close co-operation with development teams, legal teams, information security teams, and PR teams. VM is an ongoing process, and it must be a day-to-day activity because there are new threats and attacks every day.

Examples

Microsoft SDL *suggests establishing a standard Incident Response Process, with a dedicated product security incident response Team (PSIRT). The process should name the contact point in case of a security emergency (typically a PSIRT), and establish the protocol for security servicing, including plans for reacting on vulnerabilities of the code inherited from other groups within the organisation and for third party code. The incident response plan should also be tested before it is needed.*

Siemens *also has its ProductCERT which act as the central contact point for security researchers and others to report potential vulnerabilities in its products. As part of its cyber incident handling and vulnerability handling (IHVH) portfolio, ProductCERT also manages the investigation (including with regards to the vulnerabilities of the third parties used); internal co-ordination in responding to identified security issues; and public reporting of security issues, including issuing advisories to inform customers about necessary steps to securely operate Siemens products and solutions. Besides IHVH, the ProducCERT also works on regular testing of new products and patches through security vulnerability monitoring (SVM), and the Siemens Extensible Security Testing Appliance (SiESTA) platform – a unique testing box that can be used to test the security of Siemens products for known and unknown vulnerabilities.*

Kaspersky's *Product Security Team (PST) is the entry point for all issues relating to product and infrastructure security risks, through bug bounty (with up to US$100 000 awarded), co-operation with vulnerability co-ordination and bug bounty platforms like Hacker1, and an internal vulnerability report tool which provides information about the scope, rules, and policies in how reports on vulnerabilities would be processed and handled by Kaspersky. PST is responsible for both assisting the design, development and testing, and vulnerability management: preparing the initial requirements, code auditing, vulnerability response, risk analysis, vulnerability assessment, providing mitigations, fuzzing, penetration testing, and more. PST works closely with the product development teams through dedicated security champions. To optimise on human resources, PST is focused on enhancing product security and ensuring CVD (through co-operation with researchers).*

Vulnerability reporting (VRep) is when third parties report the vulnerabilities they discovered to producers.

VRep is the first step in the process. Reporting is typically done directly to the producer (if there is a point of contact); otherwise, it can go through intermediaries (such as CERTs or private entities). VRep is not equal to disclosure: reporting may not always be followed by disclosure – for instance when a vulnerability is assessed and not validated as real or relevant. Also, the researcher may decide to report the vulnerability, but then make a full disclosure without co-ordinating with the producer.

Vulnerability handling (VH) focuses on analysis of a vulnerability that is discovered or reported to the company, as well as remediation – producing and testing a fix and a release (though not a distribution or deployment) of an update.

Since patches can have cascading effects across systems that deploy them, it is essential they are analysed and tested prior to issuing updates, to ensure they do not introduce new vulnerabilities.

In broader terms, VH may also refer to handling vulnerabilities in internal IT systems and operations.

Vulnerability disclosure (VD) is an overarching term for the process of sharing vulnerability information between relevant stakeholders. It includes reporting and disclosing the existence of vulnerabilities and their mitigation to various stakeholders, including the public. This understanding is in line with ISO/IEC 29147 (ISO/IEC 29147:2014 Information technology—Security techniques—Vulnerability disclosure) definition:

> 'Vulnerability disclosure is a process through which vendors and vulnerability finders may work cooperatively in finding solutions that reduce the risks associated with a vulnerability.
> It encompasses actions such as reporting, coordinating, and publishing information about a vulnerability and its resolution.'

VD is a process where a company should co-ordinate with other stakeholders regarding the disclosure - including researchers, customers, mediators such as CERTs, the government (in case there are specific policy or regulatory requirements), as well as the public (including journalists) in case of a full disclosure (as opposed to limited disclosure). VD is not necessarily co-ordinated – not least because co-ordination may fail.

Co-ordinated vulnerability disclosure (CVD) refers to co-ordinated information sharing and mitigation efforts about reported vulnerabilities, with producers, researchers, and other interested stakeholders. CVD should ensure that vulnerabilities are addressed in a co-ordinated manner. This understanding is in line with the Carnegie Mellon CERT definition:

> ‘Coordinated Vulnerability Disclosure is the process of gathering information from vulnerability finders, coordinating the sharing of that information between relevant stakeholders, and disclosing the existence of software vulnerabilities and their mitigations to various stakeholders, including the public.’

CVD is predominantly about communication and information sharing aspects of vulnerability management. CVD may involve a number of actors: the finder and reporter of a vulnerability, producer, customer or deployer of the product, co-ordinator or mediator. The Carnegie Mellon CERT CVD Guide also defines various levels of disclosure: no disclosure, private disclosure, limited disclosure, and full disclosure.

Operational requirements for CVD include shaping and publicising policy and procedures. A vulnerability policy enables researchers to know the point of contact for initial reporting, and what to expect from the company in terms of processes. Whether a company has a vulnerability policy or not, researchers may be discovering vulnerabilities in their products. The policy should also clarify the co-ordination aspects: co-ordination of reporting, inputs about disclosed vulnerabilities, efforts within the company, and efforts between the company and security researchers.

Examples

***Cisco’s** security vulnerability policy prescribes that VD should: (a) provide equal and simultaneous access to global security vulnerability information, (b) ensure simultaneous and actionable notifications of bug fixes and security patches, (c) follow international standards and guidelines for potential vulnerability disclosure (like ISO 29147). In practice, Cisco will publicly disclose Cisco Security Advisories if its PSIRT has observed active exploitation of a vulnerability – or there is a potential for increased public awareness of such a vulnerability – that could lead to increased risk for Cisco customers (in which case the announcement may or may not include patches). Or if PSIRT has completed the incident response process with enough software patches or workarounds available to address the vulnerability, or subsequent public disclosure of code fixes is planned to address high-severity vulnerabilities. All advisories are disclosed to customers and the public simultaneously, logged in a publication listing, and customers are provided with support to search, automate, and digest them (using different tools such as Cisco Software Checker, Bug Search, Cisco PSIRT openVuln API, CVRF Repository or OVAL Repository).*

***Siemens’s** Vulnerability Handling and Disclosure Process starts with reporting and analysis of reported vulnerabilities. It then continues with internal VH, through convening a task force with the product’s manager, so that a solution can be developed as soon as possible. Certain partners, such as the national and governmental CERTs that have a partnership with Siemens ProductCERT, may be notified about a security issue in advance. Communication with the reporting party is maintained. Once the fix is ready (if needed), ProductCERT verifies its efficiency before publication. Finally, a disclosure is made by releasing the fix through existing customers’ notification processes or public advisory. Siemens also maintains a vulnerability portal and tracker for customers, and holds a monthly ‘patch day’ which is of particular relevance for the operation technology (OT) customers.*

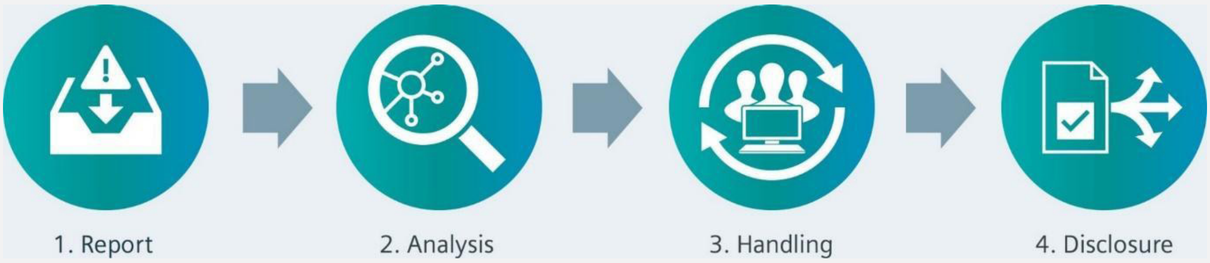


***Figure 2:** Siemens’ vulnerability handling and disclosure process encompasses report, analysis, handling, and disclosure.*
Source: Siemens

Responsible vulnerability disclosure (RVD) may suggest that the researcher should turn to the producer first to report a discovered vulnerability, giving them some time (commonly 15-45 days) to issue a fix before making the vulnerability public, in order to avoid wider-scale exploitation. At minimum, it suggests researchers (or companies or state agencies) do not exploit or commercialise vulnerabilities. RVD, however, may also suggest that the producer should act responsibly and work towards issuing a fix at the earliest convenience.

More broadly, RVD implies that there is an ethical part of the process, and proactive investment by either party (regardless of other parties) in ensuring the end goal of 'minimum risks to users' is achieved. RVD is also common in broader communities – for instance, 'responsible reporting' of vulnerabilities is used in the 2015 UN GGE report. Thus, RVD can be used interchangeably with CVD depending on the target group being addressed, or when there is a need to emphasise ethical aspects.

Some actors avoid the term 'responsible' as it is emotionally loaded, and can be interpreted differently by various parties involved. For producers, 'responsible' means not disclosing without a patch available; for researchers, it may mean that the company issues a patch; some may see it as the responsibility of customers to apply the patches.

A clear methodology within the industry is necessary to address responsibility and ethics. There are several documents to start with: the Carnegie Mellon CERT lists 'ethical considerations' as one of the principles of the CVD; FIRST has developed the Ethics for Incident Response and Security Teams; and Kaspersky has developed the Ethical principles for Responsible Vulnerability Disclosure.

Examples

***Cisco** encourages responsible response, which is based on ethical response to security incidents, quick detection and remediation of product vulnerabilities, admitting mistakes, working to make things right, and providing timely and actionable notifications of bug fixes and security patches.*

***Siemens** maintains a Hall of Thanks for researchers that ethically report security issues.*

***Kaspersky's** Ethical Principles for Responsible Vulnerability Disclosure addresses the problem of the lack of transparent approaches by companies and state actors toward how they work in vulnerability disclosure. Particularly, whether they adhere to the principles of multi-co-ordinated vulnerability disclosure, and whether they're willing to co-operate to develop remediation. These principles were developed following the non-binding norm agreed in the 2015 UN GGE report on responsible ICT vulnerability reporting; and the FIRST guidelines on Code of Ethics and vulnerability co-ordination. The aim of these principles is to encourage other actors to provide transparency in their vulnerability disclosure approaches.*

Assessment of efficiency

Putting in place vulnerability response and disclosure procedures will increase the security of products, yet it may not be easy to assess the real effectiveness of these measures. One criterion could be the maturity in the process – speed of reaction, clarity and transparency of procedures, and level of co-ordination with other stakeholders may create more secure products. One could also look at some quantitative parameters, such as the speed of fixing vulnerabilities, change in the number of (critical) vulnerabilities discovered (internally or externally) and published. And alike, the Common Vulnerability Scoring System (CVSS) of FIRST may be used as a measurement, since it is a widely used common denominator. However, the optimisation of processes towards these parameters may not necessarily lead to more secure products in the end (e.g., greater speed may bring more failure).

Trust and co-operation

The transparency of vulnerability processes is not only ethical, but also critical for enabling co-ordination and co-operation with other stakeholders. This includes transparency about the prioritisation of handling discovered vulnerabilities, which can create greater understanding between researchers and producers, and help the company to allocate resources smartly.

Prioritisation can be based on the level of criticality, through relying on common and widely used scoring and category systems for vulnerabilities; such as the CVSS, or MITRE’s Common Weakness Enumeration (CWE) and Common Vulnerabilities and Exposures (CVE). In addition, the risk assessment of whether the vulnerability was exploited or is likely to be actively exploited, and the impact of its exploitation, should set the priorities of the response. Providing such context and guidance to stakeholders can help prioritisation and accelerate vulnerability patching. Finally, transparency needs to be complemented by greater awareness and education about the elements among various parties.

Co-operation among various stakeholders and parties is essential. It is especially important for companies to maintain good relationships with researchers and partners, through transparent policies and fair procedures, as well as by offering rewards – often in the form of public acknowledgement. This will facilitate threat intelligence and insight sharing. Sharing information about vulnerabilities and supply chain risk monitoring, in particular related to open source, can also be based on models like Information Sharing and Analysis Centres (ISAC). In some cases, a mediating party may be involved: a third party like a CERT or a commercial platform, serving to collect reports and (sometimes) liaise with companies. Such settings may make processes more predictable and trusted to all parties involved. This is particularly important when time is of essence, for example when vulnerabilities have been found in products that are part of the supply chain, or in critical sectors.

Patching and support

Ideally, a disclosure should result in the development and release of patches. However, another challenge may be the capacity and readiness of the producer to distribute the patch. Some customers may miss the alerts, and thus disclosure through public channels, including media, may be needed. In some regulated environments there might be specific rules and requirements about distributing the updates and fixes, such as in the EU under the Network and Infrastructure Security (NIS) Directive (see Art. 14 which refers to Security requirements and incident notification).

Deploying the fix or patch is a particular challenge to address. Many customers don’t install patches, for various reasons: some due to the lack of awareness and skills; others can’t afford downtime or are otherwise limited by regulations; for some, fact patching may introduce new risks or system dysfunctionality of components that rely on the product being patched. In industry and operational technology (OT) networks, patching involves complex processes of testing the patch and its overall consequences for the entire system, and the downtime for patches needs to be scheduled well in advance. In addition, all the OT environments are different from one another, and patching requires a lot more customisation than in ordinary IT systems – sometimes even including physical patches. The consequences of mishandling patch deployment, however, may impact health and lives: healthcare and medical device producers rush to enable Internet of medical things (IoMT) systems, yet they often lack capacity to embed security into the production environment.

Examples

A common industry practice to help customers plan patch deployment timely is for producers to set a certain day in a week when they issue patches regularly (such as ‘patch Tuesday’ by Microsoft and Siemens). To avoid ‘vulnerable Wednesday’ – i.e., exploits of still unpatched vulnerabilities in critical systems – producers often alert partners about critical vulnerabilities ahead of time, so they can incorporate protections ahead of the patch. Besides, companies can advise customers to install critical security patches within a certain period of time – typically within 1 month or 90 days of the release, and to follow a strict process for installing patches. Siemens suggests providing additional support through separation of security and functional patches (to allow faster and more reliable patching), functional support in products for patching, or automation for advisory and patch distribution (such as through creating an application programming interface [API]).

In order to reclaim the burden of patching from the users, some OT and IoT producers are considering introducing remote patching capabilities. The users would still be allowed to disable such features, to reduce the risk of the exploitation of vulnerabilities in the very remote patching solution. Another approach being explored is the tiering of software components: issuing separate fixes for patching on multiple levels – firmware, operating system, and affected applications. Finally, even when the vulnerable system must keep running – such as the expensive decades old and insecure Supervisory Control and Data Acquisition (SCADA) systems – the use of adequate cost-effective security controls may reduce security risks to acceptable levels.

4 Adjusting the mindset and internal processes

Secure design requires a shift of the mindset throughout the organisation, which requires changes in internal processes. The Charter of Trust's principle 1 on 'Ownership for cyber and IT security', suggests companies to establish the right mindset throughout an organisation because 'it is everyone's task'.

To accomplish such a mindset shift, a company needs to put in place internal organisational processes, along the lines of the main elements discussed in this document earlier. For instance, companies have to: (a) prepare to continuously perform threat modelling and risk assessment, (b) introduce supply chain verification, (c) enhance secure development and testing and verification of procedures and capabilities, (d) establish a dedicated product CERT, (e) develop, test, and publicise vulnerability management and disclosure procedures. This, in turn, requires a company to 'ensure that the organization's people, processes, and technology are prepared to perform secure software development at the organization level and, in some cases, for each individual project,' as the US National Institute of Standards and Technology (NIST) Secure Software Development Framework (SSDF) suggests.

While security should be (a part of) everyone's jobs – from developers to managers – not everyone needs to be an expert. Instead, the organisational setup should require co-operation between different teams to ensure security. Different departments need to be involved throughout the product design life cycle. Breaking down the boundaries between security teams and developer teams is of particular importance, as the F-Secure whitepaper suggests. The involvement of the C-level management from the very beginning is equally important.

Examples

*The **Charter of Trust's** principle 1 further anchors the responsibility of cybersecurity at the highest business levels by designating specific Chief Information Security Officers (CISO). **BI.ZONE** suggests cybersecurity to be an executive priority, as the support of the senior management is the main requirement for the success of a security programme. This involves building strategies, shaping crisis management plans and interaction across departments, and understanding the changes in the cyberthreat landscape and adjusting accordingly. In addition, boards should discuss the impact (in)security can have on a company's future earnings, products, services, and business model.*

*At **Kaspersky**, the CEO and CTO give impetus, guidance, and prioritisation in ensuring product security through internal and external processes. BI.ZONE prepares special business cases, and the risk profile of organisations, to demonstrate the importance of secure development life cycle implementation to senior management. In this regard, the CISO should act not only as a technical specialist, but also as a business manager responsible for business efficiency; taking part in elaborating business strategies within the board and engaging in risk management processes. Ultimately, the senior business levels should be responsible for risk-taking: in Siemens, for instance, top management takes ownership over security by default decisions (even if it seems to be 'only' about configuration).*

One of the major pillars of change is building a security culture inside teams, and creating self-sustaining systems with high levels of cybersecurity awareness. A minimum level of security education and training for employees shall be regularly deployed (e.g., through training, certifications, and awareness). There is a particular need for training engineers, so they can implement security into the design phase, and co-operate with security teams. Training programmes should involve multiple teams, be practical and interactive - including simulations based on realistic scenarios that can map requirements and gaps in the existing arrangements.

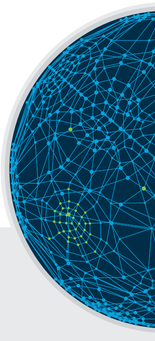
Examples

Microsoft SDL emphasises the need to provide training for developers, service engineers, and program and product managers. EnSign, for instance, conducts simulation exercises for executive management and board of directors, as well as other teams like corporate communications, human resources, business leaders, finance and procurement, IT, OT, physical security, cybersecurity, and risk and compliance.

*Simulation exercises engage all stakeholders in the organisation, especially if threat scenarios relate to code repositories and development, test, and production environments. In training the developers, EnSign uses advanced products and games like Secure Code Warrior, which strengthen their knowledge about secure coding practices. CTFZONE, a large regional 'capture the flag' (CTF) competition organised by **BI.ZONE**, serves as an interactive training for developers based on practical examples and cases or attack simulations on applications.*

One should also consider providing training support to customers and third parties, in order to enable smoother co-operation, increase mutual understanding, and allow them to benefit from the transparency on the product security and vulnerability related policies. Kaspersky's Cyber Capacity Building Program provides dedicated training on product security, to help government organisations, academia, and other companies to develop skills and knowledge for product security evaluation. It applies various techniques including; threat modelling, source code reviews, code fuzzing, and vuln management and disclosure.

Cisco hosts the SecCon conference, as an opportunity for employees worldwide to obtain cutting-edge information and build upon their existing security skills. SecCon integrates security awareness, security expertise, and networking opportunities, to create powerful lasting connections and advancements within Cisco. SecCon is organised in three different regions: Asia-Pacific, Japan and China (APJC), Europe, the Middle East and Africa (EMEA) and America. Importantly, SecCon is organised in collaboration with Cisco partners, to share knowledge with the community.



Recommended resources

Concepts

Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework, by NIST: Secure Software Development Framework (SSDF) provides a list of managerial/organisational and technical aspects for ensuring secure software development.

Foundational Cybersecurity Activities for IoT Device Manufacturers (NISTIR 8259), by NIST: provides a list of factors that IoT device manufacturers should take into account in secure development and deployment.

IoT Device Cybersecurity Capability Core Baseline (NISTIR 8259A), by NIST: provides the list of core capabilities that every IoT device should have such as logical identifier, capability to update, data protection capacity, access by authorised users, etc.

Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems (Special Publication 800-160 Vol. 1), by NIST: With an engineering perspective, NIST builds upon ISO, IEC, and IEEE standards for systems and software engineering and infuses systems security engineering methods, practices, and techniques into those systems.

Cybersecurity act supplementary references, including Security by design framework and checklist for CII operators, by the CSA of Singapore: provides guidelines for Security by Design in particular for IoT and critical systems, including tendering and acquisition processes, and guidelines for conducting risk assessment.

Good Practices for Security of IoT - Secure Software Development Lifecycle by ENISA: good practices on software design, development, testing, deployment and integration, maintenance and disposal, and the overall security in SDL (looking at people, processes, and technologies).

Software Trustworthiness Best Practices, by the Industrial Internet Consortium:
publication which discusses the concept of trustworthiness in software development, and contains examples of software failures according to different stages of the life cycle.

Secure development and deployment guidance, by the UK National Cyber Security Centre (NCSC): outlines eight principles to help developers improve and evaluate their development practices, and offers a set of basic guidelines mainly targeted at SMEs.

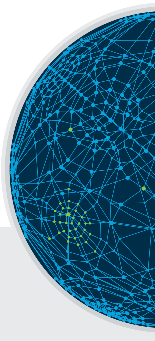
Cloud security guidance, by the NCSC: provides guidelines on cloud security.

Secure by default platforms, by the NCSC: white paper shares thoughts on desirable characteristics for building secure multimedia services.

Principles and Practices for Medical Device Cybersecurity, by International Medical Device Regulators Forum: includes a number of recommendations for stakeholders regarding best practices in the pre-market (focus is on medical device manufacturers) and post-market (includes numerous stakeholders) management of medical device cybersecurity.

Seamless Security by a Coalition to Reduce Cyber Risk (CR2): a white paper that describes how international, national, and sectoral frameworks can leverage a common baseline, and make it easier for companies to, among other, implement best-in-class cybersecurity practices consistently across their supply chain.

Security Guidance for Critical Areas of Focus in Cloud Computing, by Cloud Security Alliance: a practical, actionable roadmap to establish a stable, secure baseline for cloud operations.



Cybersecurity for resilient economies and White papers on cybersecurity risk management, by Coalition to Reduce Cyber Risk (CR2): providing main processes and principles for developing effective cybersecurity risk management policies.

Supply chain

Secure 5G networks: the EU toolbox, by the European Commission: Sets out a co-ordinated European approach based on a common set of measures, aimed at mitigating the main cybersecurity risks of 5G networks, and provides guidance in the selection and prioritisation of measures that should be part of national and EU risk mitigation plans.

Guidelines on Outsourcing, by the Monetary Authority of Singapore: Singapore's 'gold standard' Guidelines for financial institutions on risk management of outsourcing arrangements.

Guidelines for outsourced service providers, by the Association of Banks in Singapore (ABS): guidelines to manage shared and material or key service providers.

Supply Chain Risk Management Practices for Federal Information Systems and Organizations (Special Publication 800-161), by NIST: Produces a detailed supply chain risk management framework that can be applied, with a set of audit requirements on material outsourcing.

Cyber Supply Chain Risk Management, by NIST: A project related to cyber supply chain risk management.

Vulnerability processes and support

Guide to Coordinated Vulnerability Disclosure, by the Carnegie Mellon CERT

Guidelines and Practices for Multi-Party Vulnerability Coordination and Disclosure, Ethics for Incident Response and Security Teams, and PSIRT Framework, by FIRST

Coordinated Vulnerability Disclosure (CVD) Process, by the US Cybersecurity and Infrastructure Security Agency (CISA)

Vulnerability repositories

- https://nvd.nist.gov/
- http://www.cnnvd.org.cn/
- https://www.cnvd.org.cn/
- https://jvndb.jvn.jp/en/
- https://cve.mitre.org/
- https://www.securityfocus.com/
- https://securitytracker.com/

Related standards

ISO/IEC TS 27101: Information technology — Security techniques — Cybersecurity — Framework development guidelines [under development]

ISO/IEC TR 27103:2018: Information technology — Security techniques — Cybersecurity and ISO and IEC Standards

ISO/IEC 27017:2015: Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services

ETSI EN 303 645: Cyber Security for Consumer Internet of Things: Baseline Requirements

ISO/IEC 29147:2018: Information Technology – Security Techniques – Vulnerability Disclosure: on techniques and policies for vendors to receive vulnerability reports and publish remediation information.

ISO/IEC 30111:2019: Information Technology – Security Techniques – Vulnerability Handling Processes: on vulnerability handling processes for software, hardware and online services.

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Foreign Affairs FDFA

DiPLO
www.diplomacy.edu