TLP:CLEAR

SHIFTING THE BALANCE OF CYBERSECURITY RISK:

PRINCIPLES AND APPROACHES FOR SECURE BY DESIGN SOFTWARE

TLP:CLEAR

Communications Security Establishment

Canadian Centre for Cyber Security

Centre de la sécurité des télécommunications

Centre canadien pour la cybersécurité

Contents

OVERVIEW: VULNERABLE BY DESIGN

Technology is integrated into nearly every facet of daily life, as internet-facing systems increasingly connect us to critical systems that directly impact our economic prosperity, livelihoods, and even health, ranging from personal identity management to medical care. One example of the disadvantage of such conveniences are the global cyber breaches resulting in hospitals canceling surgeries and diverting patient care. Insecure technology and vulnerabilities in critical systems may invite malicious cyber intrusions, leading to potential safety[1] risks.

As a result, it is crucial for software manufacturers to make secure by design and secure by default the focal points of product design and development processes. Some vendors have made great strides driving the industry forward in software assurance, while others continue to lag behind. The authoring organizations strongly encourage every technology manufacturer to build their products based on reducing the burden of cybersecurity on customers, including preventing them from having to constantly perform monitoring, routine updates, and damage control on their systems to mitigate cyber intrusions. We also urge the software manufacturers to build their products in a way that facilitates automation of configuration, monitoring, and routine updates. Manufacturers are encouraged to take ownership of improving the security outcomes of their customers. Historically, software manufacturers have relied on fixing vulnerabilities found after the customers have deployed the products, requiring the customers to apply those patches at their own expense. Only by incorporating secure by design practices will we break the vicious cycle of constantly creating and applying fixes. **Note:** The term "secure by design" encompasses both secure by design and secure by default.

To accomplish this high standard of software security, the authoring organizations encourage manufacturers to prioritize the integration of product security as a critical prerequisite to features and speed to market. Over time, engineering teams will be able to establish a new steady-state rhythm where security is truly designed-in and takes less effort to maintain.

Reflecting this perspective, the European Union reinforces the importance of product security in the Cyber Resilience Act, emphasizing that manufacturers should implement security throughout a product's life-cycle in order to prevent manufacturers from introducing vulnerable products into the market.

[1] The authoring organizations recognize that the term "safety" has multiple meanings depending on the context. For the purposes of this guide, "safety" will refer to raising technology security standards to protect customers from malicious cyber activity.

To create a future where technology and associated products are safer for customers, the authoring organizations urge manufacturers to revamp their design and development programs to only permit the shipping of products secure by design and default. Well before development, products that are secure by design are conceptualized with the security of customers as a core business goal, not just a technical feature. Secure by design products start with that goal before development starts. Existing products can evolve to a secure by design state over multiple iterations. Secure by default products are those that are secure to use "out of the box" with little to no configuration changes necessary, and security features available without additional cost. Together, these two philosophies move much of the burden of staying secure to manufacturers and reduce the chances that customers will fall victim to security incidents resulting from misconfigurations, insufficiently fast customer patching, or many other common issues.

The Cybersecurity and Infrastructure Security Agency (CISA), National Security Agency (NSA), Federal Bureau of Investigation (FBI) and the following international partners[2] provide the recommendations in this guide as a roadmap for software manufacturers to ensure security of their products:

- Australian Cyber Security Centre (ACSC)
- Canadian Centre for Cyber Security (CCCS)
- United Kingdom's National Cyber Security Centre (NCSC-UK)
- Germany's Federal Office for Information Security (BSI)
- Netherlands' National Cyber Security Centre (NCSC-NL)
- Norway's National Cyber Security Center (NCSC-NO)
- Computer Emergency Response Team New Zealand (CERT NZ) and New Zealand's National Cyber Security Centre (NCSC-NZ)
- Korea Internet & Security Agency (KISA)
- Israel's National Cyber Directorate (INCD)
- Japan's National Center of Incident Readiness and Strategy for Cybersecurity (NISC) and Japan Computer Emergency Response Team Coordination Center (JPCERT/CC)
- OAS/CICTE Network of Government Cyber Incident Response Teams (CSIRT) Americas
- Cyber Security Agency of Singapore (CSA)
- Czech Republic's National Cyber and Information Security Agency (NÚKIB)

The authoring organizations recognize the contributions by many private sector partners in advancing security by design and security by default. This product is intended to progress an international conversation about key priorities, investments, and decisions necessary to achieve a future where technology is safe, secure, and resilient by design and default. To that end, the authoring organizations seek feedback on this product from interested parties and intend to convene a series of listening sessions to further refine, specify, and advance our guidance to achieve our shared goals.

For more information on the importance of product safety, see CISA's article, The Cost of Unsafe Technology and What We Can Do About It.

[2] Hereafter referred to as the "authoring organizations."

WHAT'S NEW

The initial publication of this report generated a significant amount of conversation within the software industry. Daily news of organizations and individuals being compromised highlights the need for more conversation regarding how to address chronic and systemic problems in software products.

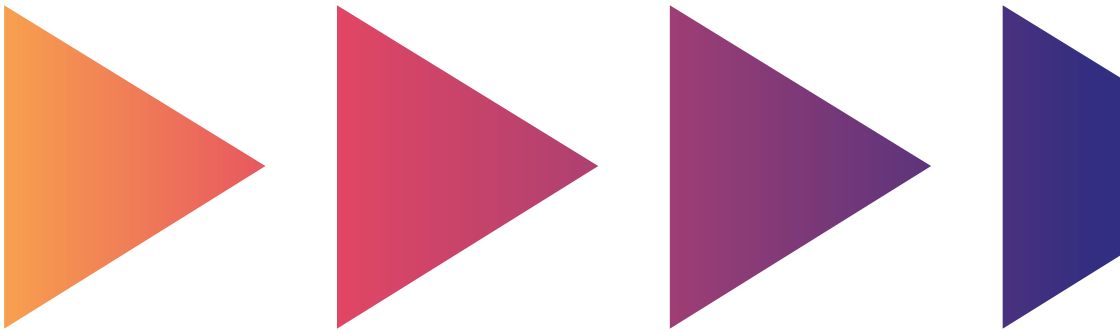
After the release in April 2023, the authoring organizations (henceforth referred to as "we" and "our") received thoughtful feedback from hundreds of individuals, companies, and trade associations. The most common request in the feedback was to provide more detail on the three principles as they apply to both software manufacturers and their customers. In this document, we expand on the original report and touch on other themes such as manufacturer and customer size, customer maturity, and the scope of the principles.

Software is everywhere and no single report will be able to adequately cover the entire range of software systems, development of software products, customer deployment and maintenance, and integration with other systems. For guidance below that does not clearly map to a particular environment, we look forward to hearing from the community how the practices described in this paper led to particular security improvements.

This report applies to manufacturers of artificial intelligence (AI) software systems and models as well. While they might differ from traditional forms of software, fundamental security practices still apply to AI systems and models. Some secure by design practices may need modification to account for AI-specific considerations, but the three overarching secure by design principles apply to all AI systems.

We recognize that transforming a software development lifecycle (SDLC) to align with these secure by design principles is not a simple task and may take time. Further, smaller software manufacturers may struggle to implement many of these suggestions. We believe that the software industry needs to make widely available the tools and procedures that make products safer. As more people and organizations focus their attention on software security improvements, we believe there is room for innovations that will narrow the gap between larger and smaller software manufacturers to the benefit of all customers.

This update to the original secure by design report is part of our commitment to build partnerships with the many interconnected stakeholder communities that underpin our technological ecosystem. It is the result of feedback from many parts of that ecosystem, and we will continue to listen and learn from perspectives. Although there are many challenges ahead, we are incredibly optimistic as we learn more about people and organizations that have already adopted a secure by design philosophy, often with success.



HOW TO USE THIS DOCUMENT

We urge software manufacturers to adhere to the principles within this document. Software manufacturers can demonstrate their commitment by publicly documenting their actions taken, in line with the steps listed below. We encourage software manufacturers to find tactics that meet the spirit of these principles and to create artifacts that will build a compelling case to even skeptical current and potential customers that they are embodying the secure by design philosophy.

In addition to actions software manufacturers should take, customers can also leverage this document. Companies buying software should ask hard questions of their vendors, drawing inspiration from the examples of adhering to the principles listed in this document. In doing so, customers can help to shift the market towards products that are more secure by design. An example of questions customers can ask of vendors is given in CISA's Guidance for K-12 Technology Acquisitions.

We encourage enterprise customers to incorporate these practices into procurement processes, vendor due diligence assessments, enterprise risk acceptance decisions, and other steps taken when evaluating vendors. Customers should also push their vendors to publicly document the secure by design actions each vendor takes. Collectively, this can create a strong demand signal for security, which can encourage and enable software manufacturers to take steps towards greater security. In other words, just as we seek to create a pervasive secure by design philosophy within software manufacturers, we need to create a "secure by demand" culture with their customers.

Secure by Design

"Secure by design" means that technology products are built in a way that reasonably protects against malicious cyber actors successfully gaining access to devices, data, and connected infrastructure. Software manufacturers should perform a risk assessment to identify and enumerate prevalent cyber threats to critical systems, and then include protections in product blueprints that account for the evolving cyber threat landscape.

Secure information technology (IT) development practices and multiple layers of defense— known as defense-in-depth—are also recommended to prevent malicious actors from compromising systems or obtaining unauthorized access to sensitive data. The authoring organizations further recommend manufacturers use a tailored threat model during the product development stage to address all potential threats to a system and account for each system's deployment process.

The authoring organizations urge manufacturers to take a holistic security approach for their products and platforms. Secure by design development requires the strategic investment of dedicated resources by software manufacturers at each layer of the product design and development process that cannot be "bolted on" later. It requires strong leadership by the manufacturer's top business executives to make security a business priority, not just a technical feature. This collaboration between business leaders and technical teams extends from the preliminary stages of design and development, through customer deployment and maintenance. Manufacturers are encouraged to make hard tradeoffs and investments, including those that will be "invisible" to the customers (e.g., migrating to programming languages that eliminate widespread vulnerabilities). They should prioritize the features, mechanisms, and implementation of tools that protect customers rather than product features that seem appealing but enlarge the attack surface.

There is no single solution to end the persistent threat of malicious cyber actors exploiting technology vulnerabilities, and products that are "secure by design" will continue to suffer vulnerabilities; however, a large set of vulnerabilities are due to a relatively small subset of root causes. Manufacturers should develop written roadmaps to align their existing product portfolios with more secure by design practices, ensuring to only deviate in exceptional situations.

The authoring organizations acknowledge that taking ownership of the security outcomes for customers and ensuring this level of customer security may increase development costs. However, investing in secure by design practices while developing innovative technology products and maintaining existing ones can substantially improve the security posture of customers and reduce the likelihood of compromise. Secure by design principles not only strengthen the security posture for customers and brand reputation for developers but the practice also lowers maintenance and patching costs for manufacturers in the long term.

The Recommendations for Software Manufacturers section listed below provides a list of product development practices and policies for manufacturers to consider.

Secure by Default

"Secure by default" means products are resilient against prevalent exploitation techniques out of the box without added charge. These products protect against the most prevalent threats and vulnerabilities without end-users having to take additional steps to secure them. Secure by default products are designed to make customers acutely aware that when they deviate from safe defaults, they are increasing the likelihood of compromise unless they implement additional compensatory controls. Secure by default is a form of secure by design.

- A secure configuration should be the default baseline. Secure by default products automatically enable the most important security controls needed to protect enterprises from malicious cyber actors, as well as supply the ability to use and further configure security controls at no additional cost.
- The complexity of security configuration should not be a customer problem. Organizational IT staff are frequently overloaded with security and operational responsibilities, thus resulting in limited time to understand and implement the security implications and mitigations required for a robust cybersecurity posture. Manufacturers can aid their customers by optimizing secure product configuration—securing the "default path"—ensuring their products are manufactured, distributed, and used securely in accordance with "secure by default" standards.

Manufacturers of products that are "secure by default" do not charge extra for implementing added security configurations. Instead, they include them in the base product like seatbelts are included in all new cars.

Security should not be a luxury option, but should be considered a right customers receive without negotiating or paying more.

RECOMMENDATIONS FOR SOFTWARE MANUFACTURERS

This joint guide provides recommendations to manufacturers for developing a written roadmap to implement and ensure IT security. The authoring organizations recommend software manufacturers implement the strategies outlined in the sections below to take ownership of the security outcomes of their customers through secure by design and default principles.

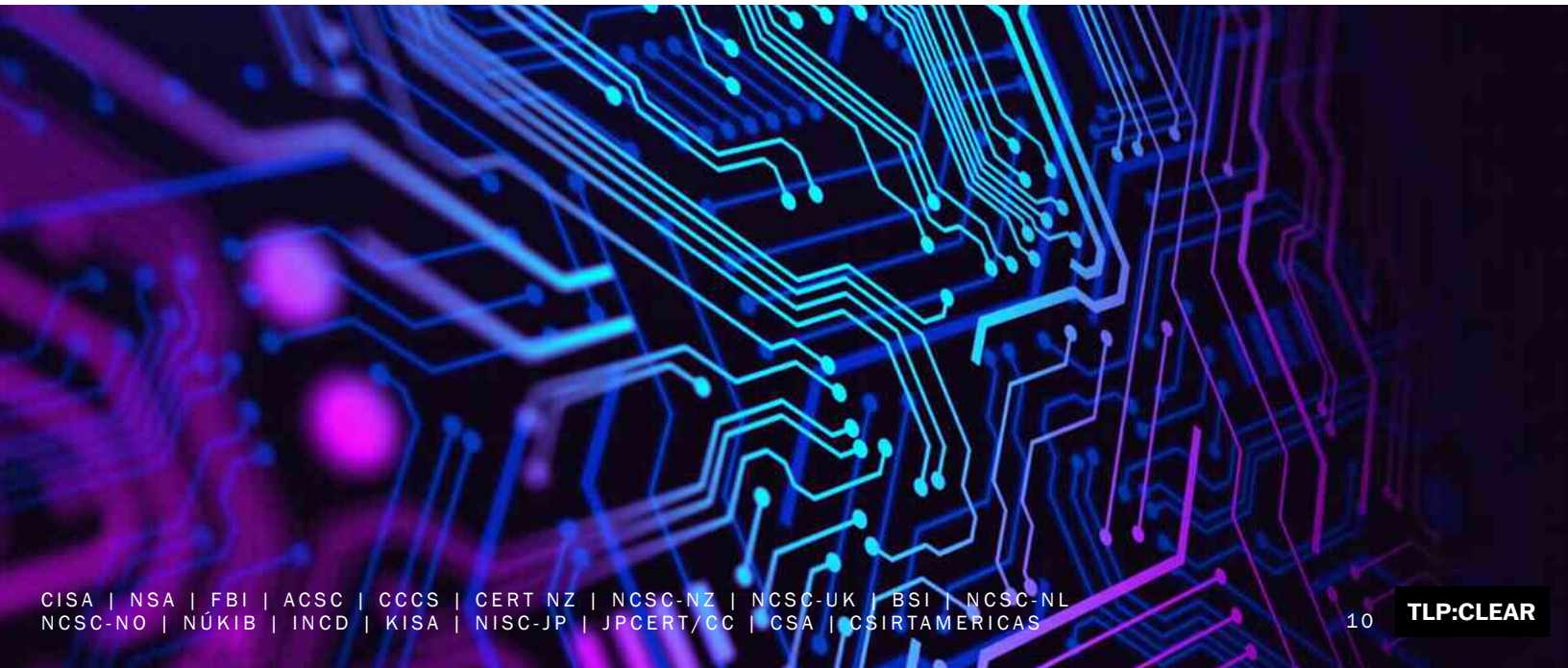
SOFTWARE PRODUCT SECURITY PRINCIPLES

Software manufacturers are encouraged to adopt a strategic focus that prioritizes software security. The authoring organizations developed the following three core principles to guide software manufacturers in building software security into their design processes prior to development, configuration, and shipment of their products.

1. **Take ownership of customer security outcomes** and evolve products accordingly. The burden of security should not fall solely on the customer.

2. **Embrace radical transparency and accountability.**
Software manufacturers should pride themselves in delivering safe and secure products, as well as differentiating themselves from the rest of the manufacturer community based on their ability to do so. This may include sharing information they learn from their customer deployments, such as the uptake of strong authentication mechanisms by default. It also includes a strong commitment to ensure vulnerability advisories and associated common vulnerability and exposure (CVE) records are complete and accurate. However, beware of the temptation to count CVEs as a negative metric, since such numbers are also a sign of a healthy code analysis and testing community.

3. **Build organizational structure and leadership to achieve these goals.**
While technical subject matter expertise is critical to product security, senior executives are the primary decision makers for implementing change in an organization. Executives need to prioritize security as a critical element of product development across the organization, and in partnership with customers.



To enable these three principles, manufacturers should consider several operational tactics to evolve their development processes.

Convene routine meetings with company executive leadership to drive the importance of secure by design and secure by default within the organization. Policies and procedures should be established to reward production teams that develop products adhering to these principles, which could include awards for implementing outstanding software security practices or incentives for job ladders and promotion criteria.

Operate around the importance of software security to business success. For example, consider assigning a "software security leader" or a "software security team" that upholds business and IT practices that directly link software security standards and manufacturer accountability. Manufacturers should ensure they have robust, independent product security assessment and evaluation programs for their products.

Use a tailored threat model during resource allocation and development to prioritize the most critical and high-impact features. Threat models consider a product's specific use-case and enable development teams to fortify products. Finally, senior leadership should hold teams accountable for delivering secure products as a key element of product excellence and quality.

As part of the October 2023 update to this guidance, these three principles are expanded upon through the following explanations, demonstrations, and evidence.

PRINCIPLE 1: Take Ownership of Customer Security Outcomes

EXPLANATION

Modern best practices dictate that software manufacturers invest in product security efforts that include **application hardening, application features,** and application **default settings.**

Software manufacturers need to implement **application hardening** by using processes and technologies that raise the cost for a malicious actor wishing to compromise applications. Application hardening protocols and procedures help products resist attacks by intelligent malicious actors. Terms like hardening, product security, and resilience are all closely related to product quality. The idea is that security must be "baked in," and not "bolted on." [1] By baking in security, software manufacturers can not only increase their customers' security but also increase their products' quality. Sample tactics include ensuring user input is validated and sanitized, and isn't entered directly into code (i.e., by using parameterized queries instead), using a memory safe programming language, rigorous software development life cycle (SDLC) management, and using hardware-backed cryptographic key management.

Applications need to support **application features** that relate to cybersecurity. Sometimes called "capabilities," these features extend the functionality of a product or service in ways that help maintain or increase the security posture of a customer.

Sample security-related features include supporting transport layer security (TLS) for all network connections, single sign on (SSO) support, multi-factor authentication (MFA) support, security event audit logging, role-based access control (RBAC), and attribute-based access control (ABAC).

Some of these product features are configurable allowing customers to more easily integrate the product into their existing environments and workflows. Those configurations mean applications must have **default settings** set until customers configure them. Those default settings need to be set securely "out of the box" so that customers expend fewer resources to make their stack of technology products more secure.

Each of these elements – application hardening, application security features, and application default settings – plays a role in the security of the application, and the resulting security posture of the customer. Software manufacturers should think about each of these elements and how they relate to each other. Manufacturers should think about more than just their investments to incorporate these elements into their products. Manufacturers should take it a step further and consider how those elements change the real-world security posture of their customers, for better or for worse.

Manufacturers should take ownership of their customers' security outcomes rather than measuring themselves solely on their efforts and investments. The responsibility should be placed upstream, with the manufacturers, where it has the greatest likelihood of reducing the chances of compromise.

Unfortunately, that's not the case today. Too many manufacturers place the burden of security on the customer rather than investing in comprehensive **application hardening.** For example, when the manufacturer patches one vulnerability, we often see similar vulnerabilities exposed because they addressed the symptom rather than the root cause of that defect. The product might implement different mitigations in various parts of the code base for the same class of vulnerability. As a case in point, after the manufacturer fixed one input sanitization vulnerability, researchers or attackers found code paths that did not benefit from the improved input sanitization. The manufacturer applied fixes one at a time rather than unifying the codebase to eliminate that class of vulnerability across the entire application.

Application features can create both benefits and risk for customers. Features that allow integration points with many external systems and versions can greatly increase the value of a product. And yet supporting features without a retirement plan, like a networking protocol, can leave customers vulnerable if they lack an understanding of the implications of ongoing use of that feature. For example, some products continue to use networking protocols that have their origins in the 1990s or 2000s and are now known to be unsafe. There are numerous factors that can slow how fast customers upgrade and deploy modern security measures. They may use products that integrate with the rest of the organization's network, but lack modern security measures, preventing the IT team from modernizing. Still, software manufacturers can factor these patterns into their planning process to encourage customers to stay current.

Application default settings are an added area of potential risk for customers. Manufacturers often choose certain default settings, making it easier for customers to use the application features they want. The downside is that this practice increases the attack surface for customers who may not need certain features and protocols that are enabled by default. Additionally, many security controls are toggled off by default or require customers to take time to configure their settings to increase security. Explicit threat modeling is a tactic that may help inform the decision of which features should be on by default or which settings are needed to be secure by default. Another tactic is to investigate ways to make features more discoverable for the administrator.

Some manufacturers ship products with defaults that can create risk for some or all their customers. Rather than set safer defaults, they often opt to produce a **hardening guide** that customers must implement at their own expense. Hardening guides suffer from several common problems. Some hardening guides are hard to find and are not well supported. Others are complex to implement, occasionally requiring software development to write an extension module. Still, others assume the reader has extensive cybersecurity experience to understand the ways in which various settings change the attack surface. Practitioners who have an incomplete understanding of the ways in which attackers work may fail to properly implement hardening guide instructions, especially if the instructions do not make the trade offs clear. Further, not all hardening guides are written by engineers who are intimately familiar with attacker tactics and economics, causing them to create hardening guides that are ineffective even if faithfully implemented. Millions of customers are taking on the responsibility to harden multiple instances of software or systems, often in resource-constrained environments. Relying on hardening guides simply doesn't scale.

An application's settings should be continuously evaluated whether the settings were the default or set by the customer, against the manufacturer's current understanding of the threat landscape. Applications should be made with clear indicators about the potential risks that may result from those settings and should make those indicators known. Just like a modern car has an indicator about seatbelts and expresses that indicator by sounding an alert if you try to drive without buckling up, software should express indicators about the state of security of a system. If an application is configured to not require MFA for administrator accounts, it should make the administrators regularly aware that they and their entire organization are in danger if they do not configure MFA. Additionally, if an application is configured to support older protocols that are now known to implement weak cryptography, it should regularly make it clear to the administrators that the organization is in danger and provide resources to resolve the situation. We urge manufacturers to implement routine nudges that are built into the product rather than relying on administrators to have the time, expertise, and awareness to interpret hardening guides. Opportunities clearly exist for innovation to balance security and usability considerations.

Each of the above elements creates an untenable situation in which customers need to research, fund, purchase, staff, deploy, and monitor additional **security products** to reduce the chance of compromise. Small and medium sized organizations (SMOs) are generally unable to facilitate these options. They face scarcity in expertise, funding, and time which taxes bandwidth and function, forcing security to a lower priority, and, in the aggregate, exacerbates collective risk. Conversely, security investments by the relative few manufacturers will scale. A common phrase that summarizes the problem is that the software industry needs more secure products, not more security products. Software manufacturers should lead that transformation.

> ***The software industry needs more secure products, not more security products. Software manufacturers should lead that transformation.***

Today, we sometimes read comments from manufacturers explaining that a customer was compromised due to not enabling a particular security feature or following specific hardening guidance. Instead, after a compromise, manufacturers should explain whether a particular security feature or specific hardening guidance would have prevented the compromise and consider making it the default at no charge. In those cases where the product itself was not sufficiently hardened in the design and implementation phases, the manufacturer should explain how they are working to eliminate that class of vulnerability from their product lines.

Software manufacturers have a responsibility to ensure that their products are designed and developed with security as a top priority. To that end, they should **objectively measure the results** of their efforts in the field. We call on manufacturers to not just focus on their internal efforts, but to objectively measure and regularly report the results and effectiveness of a product's security efforts and configurations, and to build a feedback loop that creates changes in the SDLC that lead to measurable improvements in customer safety and more secure products. Reporting should include anonymized data that the academic and security research community could use to track high-level trends and measure progress ecosystem wide.

DEMONSTRATING THIS PRINCIPLE

Software manufacturers and online services should find ways to demonstrate successes in implementing this principle. They should seek to provide evidence in the form of artifacts for outsiders to examine. No single artifact by itself will prove that a manufacturer is implementing a robust secure by design program, but by providing various artifacts they will build a case of the manufacturer's commitment to developing secure products. This approach is in the spirit of "show, rather than tell."

To demonstrate this principle, software manufacturers should consider steps such as those in the following list. The authoring organizations recognize that few software manufacturers will be able to immediately implement these practices and produce corresponding artifacts at the start of their secure by design journey. Further, software manufacturers will need to prioritize this list depending on how the customers deploy the product in the field to achieve the largest security benefits.

SECURE BY DEFAULT PRACTICES

1. **Eliminate default passwords.** Default passwords continue to be implicated as the cause of many attacks every year. Making a commitment to eliminate this chronic problem will deny easy access to attackers. Similarly, manufacturers should consider what password practices should be implemented, such as minimum password length and disallowing known breached passwords.
2. **Conduct field tests.** As technology continues to evolve and become more complex, it is increasingly important for software manufacturers to conduct security-centric user testing to understand their products' security posture in the field. Similar to how user research informs software development requirements, software manufacturers should also conduct security-focused user research to understand where the security user experience (UX) falls short. By observing how customers deploy and use their products in real-world environments, software manufacturers can gain valuable insights into the usability and effectiveness of their security features and controls. These insights can help identify areas for improvement and refine their products to better meet the security needs of customers. For example, field tests might suggest changes in UX flow, defaults, alerting, and monitoring. Field tests may also show where past improvements in the product's design reduce the velocity of security patches, reduce configuration errors, and minimize attack surface.

Manufacturers should consider the following:

- Do customers correctly implement the hardening guide?
- Do the product's existing security features perform as expected in the field?
- Do those features actually resist real-world attacks?
- Which features would better reduce the likelihood of compromise?

Note: To gain deeper insights into these elements, software manufacturers may wish to partner with customers to conduct red team exercises to see how the product resists attacks. These field tests might take place at the customer's physical site, virtually, or via telemetry from the application in a privacy-preserving manner.

3. **Reduce hardening guide size.** Manufacturers can improve customers' security postures by streamlining or even eliminating product hardening guides and focusing on the most critical security measures that customers should prioritize when deploying their products. Rather than overwhelming customers with a laundry list of security measures, manufacturers should identify the top security risks that their products are susceptible to and provide clear and concise guidance on how to mitigate these risks. In addition, manufacturers should provide customers with tools and automation that simplify the process of implementing security controls, such as scripts that can easily be deployed in their environment. These tools should additionally be able to verify and clearly show the changes made from the original baseline. By streamlining hardening guides and providing customers with easy-to-use tools and automation, manufacturers can reduce the burden on their customers and help ensure that their products are deployed in a secure manner. One tactic would be to consider implementing the Pareto principle to reduce the number of steps for the common use cases (the 80%), and then providing contextual guidance and tooling for less common scenarios (the 20%). In this way, software manufacturers will be making

the simple things simple, and the hard things possible. Field testing will be a powerful tool in measuring how long it takes customers to discover, understand, and implement hardening guides. Manufacturers should consider how the product could nudge administrators to take action within the product itself rather than relying on them to implement tasks from a hardening guide.

4. **Actively discourage use of unsafe legacy features.**
   Prioritize security through clear upgrade paths over backwards compatibility. Publish blog posts showing the adoption of safer features and protocols, and deprecate unsafe features by announcement, possibly from within the product itself. A significant number of customers have demonstrated that they will not keep their systems current with modern network, identity, and other critical security features. In some cases, customers fear existing functionality will break with an upgrade. By making upgrades as seamless as possible, customers will likely upgrade and get security fixes more often and quickly. Software manufacturers should aggressively nudge customers along upgrade paths that reduce customer risk.

5. **Implement attention grabbing alerts.**
   Similar to seat belt chimes in cars that continuously make noise when seat belts are not fastened, manufacturers should implement timely and repeated alerts when users or admins are in truly unsafe states, warning administrators that they are using deprecated protocols in their environments and suggest upgrade paths. Implement timely and repeated alerting when users or admins, or the application configuration, are in an unsafe state. Make the unsafe mode clear to the administrators on a regular basis. An additional feature could require a super administrator to acknowledge the lack of MFA on their account upon each login, or even disable certain key features until they enable MFA. There is room to innovate to achieve these goals while not creating alert fatigue.

6. **Create secure configuration templates.**
   These templates can pre-set certain configurations to safe settings based on an organization's risk appetite. While it might be overly simplistic to have low/medium/high security templates, that example illustrates how many settings could be updated to manage risk for the organization. Templates can be supported by hardening guides on the risks the manufacturer has identified.

SECURE PRODUCT DEVELOPMENT PRACTICES

1. **Document conformance to a secure SDLC framework.** Secure SDLC frameworks provide objectives and examples across people, processes, and technologies. Consider publishing a detailed description of which secure SDLC framework controls have been implemented and describe any alternate controls which have been used. Within the US, consider using the NIST Secure Software Development Framework (SSDF). While not a checklist, the SSDF "describes a set of fundamental, sound practices for secure software development."

2. **Document Cybersecurity Performance Goals (CPG) or equivalent conformance.** When an organization attests that they conform to the NIST SSDF standard, they are asserting that their SDLC is informed by well-understood best practices. However, it is not sufficient for them to only have a robust SDLC. They also need to protect their own enterprise and development environments from malicious actors who would seek to manipulate the security properties of the product while it is still in development. This is not a theoretical class of attack, but one that has been carried out with adverse effects to customers, and by extension national security. Organizations should consider publishing details on the organization's conformance to the CISA CPGs, the NIST Cybersecurity Framework (CSF), or other cybersecurity program frameworks.

3. **Vulnerability management.** Some manufacturers have a vulnerability management program that focuses on patching vulnerabilities discovered internally or externally, and little more. More mature programs incorporate extensive data-driven analysis of vulnerabilities and their root causes, taking steps to systemically eliminate entire classes of vulnerability[3]. They implement formal programs around setting quality planning, quality control, quality improvement, and quality measurement. They view defect management as a business matter, not merely a security matter. These programs are not dissimilar in some ways to quality and safety programs in other industries.

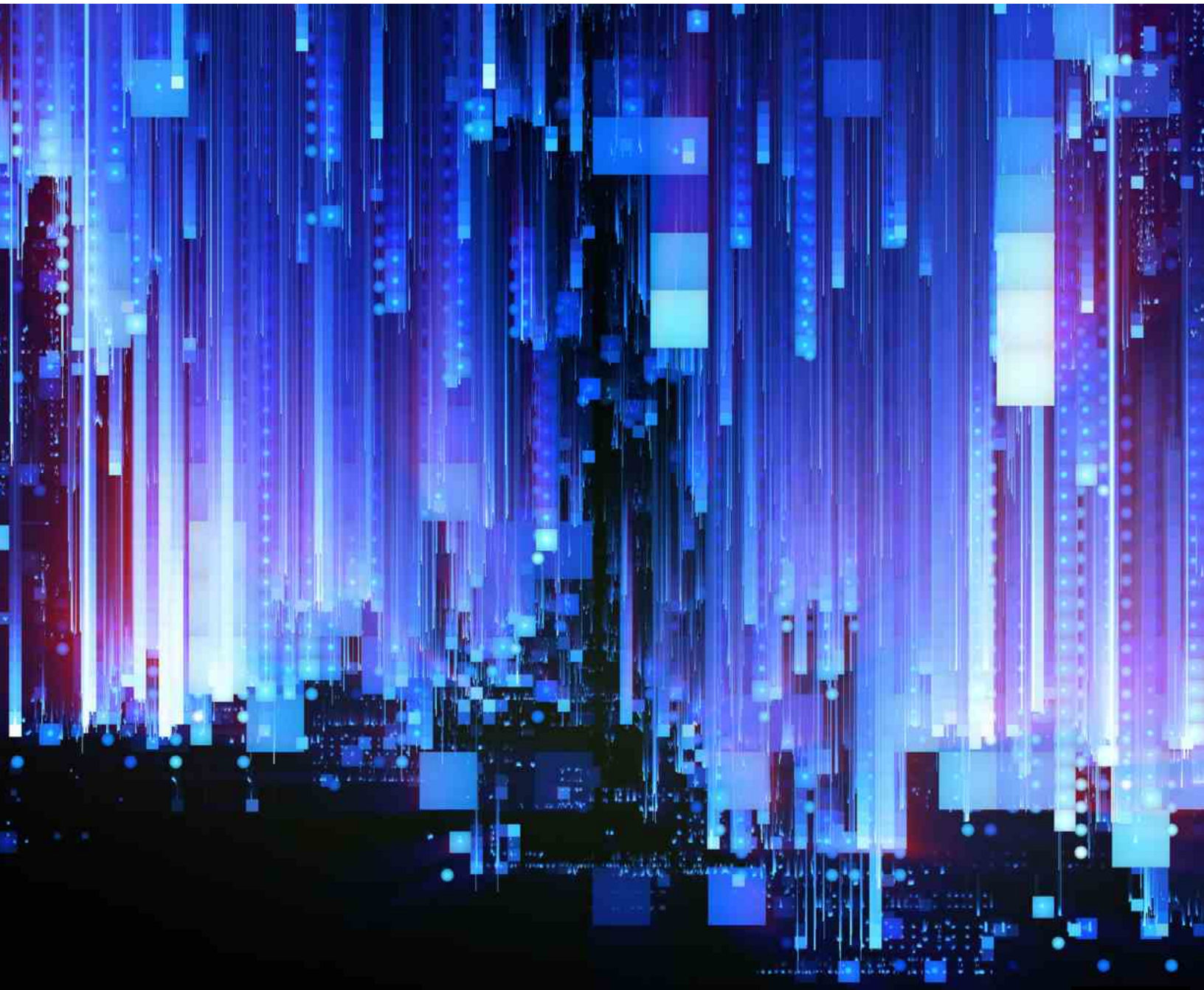
4. **Responsibly use open source software.** When open source software is used, be responsible by vetting open source packages, fostering code contributions back to dependencies, and helping sustain the development and maintenance of critical components. For reference, Japan's Ministry of Economy, Trade, and Industry (METI) has published "Collection of Use Case Examples Regarding Management Methods for Utilizing OSS and Ensuring Its Security."

5. **Provide secure defaults for developers.** Make the default route during software development the secure one by providing safe building blocks for developers. For example, given the prevalence of SQL injection vulnerabilities causing real-world harm, ensure that developers use a well-maintained library to prevent that class of vulnerability. Also known as "paved roads" or "well-lit paths," this practice ensures both speed and security, and reduces human error.

6. **Foster a software developer workforce that understands security.** Ensure that your software developers understand security by training them on secure coding best practices. Further, help transform the broader workforce by updating hiring practices to evaluate security knowledge and working with universities, community colleges, bootcamps, and other educators to weave security into computer science and software development curriculums.

[3] NIST SSDF, PO 1.2, Example 2: "Define policies that specify the security requirements for the organization's software, and verify compliance at key points in the SDLC (e.g., classes of software flaws verified by gates, responses to vulnerabilities discovered in released software)."

7. **Test security incident event management (SIEM) and security orchestration, automation, and response (SOAR) integration.** In addition to conducting field tests, work jointly with popular SIEM and SOAR providers in conjunction with select customers to understand how incident response teams use logs to investigate suspected or actual security incidents. Few software developers have experience responding to an incident and may create log entries that don't help responders as much as they would expect. By working both with SIEM and SOAR technologies and real incident response professionals, the development team can create logs that tell the correct and complete story, saving time and reducing uncertainty during an incident.
8. **Align with Zero Trust Architecture (ZTA).** Align product deployment guides with, for example, the NIST ZTA models and the CISA Zero Trust Maturity Model. Encourage customers to incorporate these principles in their environments.



PRO-SECURITY BUSINESS PRACTICES

1. **Provide logging at no additional charge.** Cloud services should commit to generating and storing security-related logs at no additional charge. On-premises products should likewise generate security-related logs at no additional charge. Further, the product should log security events by default since many customers may not understand their value until after an incident. These tactics may require a thorough review of what security events should be logged to provide cybersecurity state awareness, how a customer may configure logging, for what time period logs are retained, how log integrity and storage are protected, and how logs can be analyzed. In some cases, the review may suggest the need for a refactoring of the application's log management architecture to help make them actionable and at a cost that works for the manufacturer. Working with incident response (IR) experts can increase the chances that the logs will be useful to investigators in the field. See the section on SIEMs.

2. **Eliminate hidden taxes.** Publish a commitment to never charge for security or privacy features or integrations. For example, within the larger scope of identity and access management (IAM), there are services called single sign-on (SSO) services. Some manufacturers charge more to connect their system to a SSO service (sometimes referred to as an identity provider). This "SSO tax" means that good identity and access management is out of reach for many SMOs, preventing them from achieving a strong security posture. Some services charge more to enable MFA for users. **Security should not be priced as a luxury good but considered a customer right.** Some manufacturers have argued that few customers request these features, and they cost more to maintain. These arguments ignore the fact that few customers will call to complain or bargain, not all customers actually understand what the benefits of these features are, and that all features cost something to maintain. Yet we don't see many manufacturers charging extra for availability or data integrity. The costs to support those key attributes are built into the price all customers pay, much like the costs to include seatbelts, collapsible steering columns, and airbags that save lives in accidents.

3. **Embrace open standards.** Implement open standards, especially around common network and identity protocols. Avoid proprietary protocols when open standards are available.

4. **Provide upgrade tooling.** Many customers are reluctant to adopt the latest version of the product, including deploying newer and more secure features like secure network connections. Software manufacturers can increase customer adoption of new upgrades by providing tooling to help reduce uncertainty and risk. Offer free licenses for customers to test upgrades and patches in a test environment as a way to motivate customers.

PRINCIPLE 2: Embrace Radical Transparency and Accountability

EXPLANATION

Software manufacturers should pride themselves in delivering safe and secure products, as well as differentiating themselves from the rest of the manufacturer community based on their ability to do so.

Let's address a common concern about transparency. When practitioners discuss radical transparency, there is a tendency for the conversation to get bogged down in a concern that they are providing a "roadmap for attackers." However, the overwhelming evidence is that attackers are doing just fine without such roadmaps, and such concerns should take a back seat to transparency that benefits direct customers, indirect customers, supply chains, and the entire software industry.

Transparency helps the industry establish conventions—in other words, what "good" looks like. It helps those conventions change over time in response to customer needs, changes in threat actor tactics or economics, or technology evolution. Transparency helps manufacturers with fewer resources learn from those with more mature and capable resources. Conversations about information sharing should expand beyond real-time threat indicators, to include the elements below.

Transparency forces decisions around security to be made early in the development process, and to be a continuous activity of business leaders as well as engineers and

security professionals. Transparency builds accountability into the product.

A note on the choice of the adjective "radical" in front of "transparency." Today, it is uncommon for software manufacturers to publish detailed information about how they develop and maintain software and how they mature their programs using data over time. In the software industry, few manufacturers offer guided tours of how they design their software. There are few opportunities for software manufacturers to see how peer organizations structure their SDLC programs, and how those programs hold up in the customer environments against real attackers. The collective industry would benefit from more information sharing on topics such as strategies to measure the cost of security defects and to eliminate classes of vulnerability. As a result of these common practices, every software manufacturer must learn how to deal with product security on their own. Perhaps by not placing a luxury tax on security features, safety and security therefore become a cost center rather than a profit center, and companies would benefit by lightening the load through collaboration and transparency.

We want to focus on the tactics that will materially accelerate the evolution of the software industry. We can no longer afford to make opportunistic, incremental improvements. If we are to collectively overcome the threats posed by intelligent and adaptive adversaries, we must embrace levels of transparency that will feel uncomfortable today, but that will drive the industry forward. There are manufacturers today who embody some of these secure by design principles. As William Gibson said, "the future is already here, it's just not very evenly distributed." **Radical transparency will help distribute that information and benefit the defenders more than our adversaries.**

Transparency can do more than help peer organizations mature their SDLCs. Prospective customers and investors can learn more about the investments and tradeoffs manufacturers have made, and the security posture those investments have created for customers. Manufacturers who embrace radical transparency will give customers information to help them make purchasing decisions not just on price and features, but on security as well.

As hard as organizations work to secure their supply chain and their SDLC, companies have had their builds processes compromised in the recent past. Embracing radical transparency should lead to public disclosure of the attack as well as the improvements the company made to prevent and detect future attacks. That form of information sharing will help other organizations learn without having to suffer the same fate.

DEMONSTRATING THIS PRINCIPLE

To demonstrate this principle, software manufacturers should take steps including the following:

SECURE BY DEFAULT PRACTICES

1. **Publish aggregate security relevant statistics and trends.** Example topics include MFA adoption by customers and administrators and use of unsafe legacy protocols.
2. **Publish patching statistics.** Detail what percent of customers are on the latest version of the product, and what you are doing to make updates easier and more reliable.
3. **Publish data on unused privileges.** Publish aggregate information on excessive permissions across your customer base as well as the nudges and other changes to the product you are making to reduce the customers' attack surfaces. These unused privileges are likely to be good candidates for administrator alerts, like seatbelt chimes.

SECURE PRODUCT DEVELOPMENT PRACTICES

1. **Establish internal security controls.** Many companies have seen the benefits of moving their data to cloud providers. Now those cloud providers become the target of attackers. Software as a Service (SaaS) providers should publish statistics of their internal controls. For example, SaaS providers should publish statistics on their internal deployment of phishing-resistant MFA, like Fast Identity Online (FIDO) authentication. Ideally, they should be able to say that no staff member can access customer or other sensitive data without authenticating via phishing-resistant MFA.

2. **Publish high-level threat models.** Secure by design products start with written threat models that describe what the creators are trying to protect and from whom. Effective threat models are informed by the way intrusions happen in the wild, and should cover both the enterprise and development environments, as well as the way the software manufacturers intend for it to be used in customer environments.

3. **Publish detailed secure SDLC self-attestations.** Manufacturers following NIST SSDF, or other similar frameworks are actively working towards a mature software development lifecycle. Publishing a self-attestation of which controls the manufacturer has enacted, and for which products, would demonstrate a commitment to adhering to these best practices and provide an increased level of confidence to their customers. Other certification schemes include the Israel Cyber Supply Chain Methodology, for instance.

4. **Embrace vulnerability transparency.** Publish a commitment that will ensure that identified product vulnerabilities will be published as CVE entries that are correct and complete. That's especially true for the common weakness enumeration field that identifies the root cause of the vulnerabilities. The more correct and complete the public CVE database is, the more the industry can track how products are becoming more secure, and which classes of vulnerabilities are most prevalent. However, beware of the temptation to count CVEs as a negative metric, since such numbers are also a sign of a healthy code analysis and testing community. As manufacturers implement a secure by design philosophy, it's possible that at first their raw CVE count will go up due to more comprehensive discovery and remediation of vulnerabilities in existing code. Manufacturers should publish analysis of past vulnerabilities, including any patterns and measures that were taken to address the entire class of vulnerabilities. For example, if a large percentage of a company's CVEs were related to cross-site scripting (XSS), documenting the root cause analysis, response (such as shifting to web template frameworks that prevent XSS), and results would signal to customers that they will not be victimized by a class of vulnerability for which mitigations have been understood for decades.

5. **Publish Software Bills of Materials (SBOMs).** Manufacturers should have command of their supply chains. Organizations should build and maintain SBOMs [2] for each product, request data from their suppliers, and make SBOMs available for downstream customers and users. This will help demonstrate their diligence in understanding the components they use in the creation of their products, their ability to respond to newly identified risks, and can help customers understand how to respond if one of the modules in the supply chain has a newly found vulnerability.

For reference, Japan's Ministry of Economy, Trade, and Industry (METI) has published "Guide of Introduction of Software Bill of Materials (SBOM) for Software Management." Transparency should extend to firmware in embedded devices and the data and models used in AI/machine learning (ML). Beyond assisting in purchasing decisions and operational capabilities, SBOMs play an important role in the infrastructure to detect and respond to malicious supply chain attacks.

6. **Publish a vulnerability disclosure policy.** Publish a vulnerability disclosure policy that (1) authorizes testing against all products offered by the manufacturer and conditions for those tests, (2) provides legal safe harbor for actions performed consistent with the policy, and (3) allows public disclosure of vulnerabilities after a set timeline. Manufacturers should perform root-cause analysis of discovered vulnerabilities and, to the greatest extent feasible, take actions to eliminate entire vulnerability classes. See CISA's Vulnerability Disclosure Policy Template for reference language.

PRO-SECURITY BUSINESS PRACTICES

1. **Publicly name a secure by design senior executive sponsor.** In many organizations, security (like quality) is delegated to technical teams who have limited ability to make structural changes to dramatically improve the security of the products. Publicly naming a top business executive to oversee the secure by design program will transform the security of products into a top-level business concern.

2. **Publish a secure by design roadmap.** Manufacturers should document changes made to their SDLC to improve customer security, including details about field-test reports, actions taken to eliminate entire classes of vulnerability, and other items listed in the other principles. As in the case of quality improvement efforts, security improvement programs have distinct phases of planning, control, and improvement. In the spirit of showing rather than telling, publishing the roadmap and the details behind these phases will build confidence that the products are secure by design. After achieving meaningful progress, manufacturers can detail them in transparency reports. Doing so not only demonstrates a commitment to secure by design principles but can inspire others to adopt similar programs by showing an existence proof.

3. **Publish a memory-safety roadmap.** Manufacturers can take steps to eliminate one of the largest classes of vulnerability by migrating existing products and building new products using memory safe languages. While this may not be possible in all cases, manufacturers can consider developing application wrappers in memory safe languages instead of re-writing entire applications. This can also include how manufacturers are updating hiring, training, code review, and other internal processes, as well as ways they are helping the open source community do the same.

4. **Publish results.** While updating their SDLC to embody a secure by design philosophy, organizations will find quick wins, more resource intensive wins, and some unexpected setbacks. By presenting their internal successes and roadblocks, the entire industry can learn from the results.

PRINCIPLE 3: Lead from the Top

EXPLANATION

While the overall philosophy is called "secure by design," the incentives for customer safety begin well before the product design phase. They begin with business goals and implicit and explicit objectives and desired outcomes. Only when senior leaders make security a business priority, creating internal incentives, and fostering an across-the-board culture to make security a design requirement will they achieve the best results.

While technical subject matter expertise is critical to product security, it is not a matter that can be solely left to technical staff. It is a business priority that must start at the top.

Some people have wondered if a software manufacturer is embracing the first two principles and producing meaningful artifacts, is the third principle necessary? How a company establishes its vision, mission, values, and culture will affect the product, and those elements have a heavy component at the top. We see this in other industries that have made dramatic improvements in safety and quality. Noted quality expert J.M. Juran wrote:

> ***Attainment of quality leadership requires that the upper managers personally take charge of managing for quality. In companies that did attain quality leadership, the upper managers personally guided the initiative. I am not aware of any exceptions. [3]***

We believe that security is a sub-category of product quality. When security and quality become business imperatives rather than technical functions left solely to technical staff, organizations will be able to respond to the security needs of their customers more quickly and efficiently. Moreover, investing the necessary resources to ensure that software security is a core business priority from the beginning will reduce the long-term costs of addressing software defects–and in turn, lower the national security risks.

In the same way that leadership teams have implemented corporate social responsibility (CSR) programs, there is growing awareness that corporate boards, including those of software manufacturers, should take a more active role in guiding cybersecurity programs. The term corporate cyber responsibility (CCR) is sometimes used to describe this emerging idea.

DEMONSTRATING THIS PRINCIPLE

To demonstrate this principle, software manufacturers should take steps including the following:

1. **Include details of a secure by design program in corporate financial reports.** If the manufacturer is a publicly traded company, add a section in each annual report devoted to secure by design efforts. It is common for automobile annual financial reports to include sections on driver and passenger safety, including information about centralized and distributed quality and safety committees. Detailing the secure by design program in a financial report will demonstrate that the organization is linking customer security and corporate financial outcomes and not simply adopting a term in marketing materials because it is in vogue.

2. **Provide regular reports to your board of directors.** Chief information security officer (CISO) reports to corporate boards usually include information about current and planned security programs, threats, suspected and confirmed security incidents, and other updates centered on the security posture and health of the company. In addition to receiving information about the security posture of the enterprise, boards should request information about product security and the impact it has on customer security. Boards should not look solely to the CISO, but primarily to other members of company management to drive customer risk down.

3. **Empower the secure by design executive.** There is a significant difference between an organization where the technical teams have "executive buy-in," and those where business leaders personally manage the customer security improvement process using standard business processes. The term "executive buy-in" implies that someone had to sell the idea of a customer safety program rather than it being a top-level business goal. This executive must be empowered to influence product investments to achieve customer security outcomes.

4. **Create meaningful internal incentives.** While being mindful to not create perverse incentives, align reward systems to improve customer security to match other valued behaviors and outcomes. From the secure by design executive to product management, software development, support, sales, legal, and other organizations, weave customer security incentives into hiring, promotions, salaries, bonuses, stock options, and other common processes in the running of the business. For example, when establishing criteria for promoting software developers, include considerations for improving the security of the product along with other criteria like uptime, performance, and feature improvements.

5. **Create a secure by design council.** In some industries, it's common for organizations to create a central quality council, and to embed quality representatives in key divisions or business units. By including both centralized and distributed members, these groups work to improve quality against top level goals while receiving telemetry from deep in the organization. Similarly, a secure by design council would improve security against secure by design goals throughout the organization.

6. **Create and evolve customer councils.** Many software manufacturers have customer councils comprised of customers from different regions, industries, and sizes. These councils can provide a great deal of information about customer successes and challenges in deploying the company's products. Structure the council agenda with dedicated topics addressing customer safety, even if it's not currently top of mind for the participants. Consider where the customer council reports and how to tap participants for insights into the product's security as deployed. For example, does the council have a bias towards marketing and sales purposes, or product management? The secure by design executive should help steer these customer interactions and should link them with other elements in this paper, such as field studies.

SECURE BY DESIGN TACTICS

The Secure Software Development Framework (SSDF), also known as the National Institute of Standards and Technology's (NIST's) SP 800-218, is a core set of high-level secure software development practices that can be integrated into each stage of the software development lifecycle (SDLC). Following these practices can help software producers become more effective at finding and removing vulnerabilities in released software, mitigate the potential impact of the exploitation of vulnerabilities, and address the root causes of vulnerabilities to prevent future recurrences.

The authoring organizations encourage the use of secure by design tactics, including principles that reference SSDF practices. Software manufacturers should develop a written roadmap to adopt more secure by design software development practices across their portfolio. The following is a non-exhaustive illustrative list of roadmap best practices:

- **Memory safe programming languages (SSDF PW.6.1).** Prioritize the use of memory safe languages wherever possible. The authoring organizations acknowledge that memory specific mitigations may be helpful shorter-term tactics for legacy codebases. Examples include C/C++ language improvements, hardware mitigations, address space layout randomization (ASLR), control-flow integrity (CFI), and fuzzing. Nevertheless, there is a growing consensus that adoption of memory safe programming languages can eliminate this class of defect, and software manufacturers should explore ways to adopt them. Some examples of modern memory safe languages include C#, Rust, Ruby, Java, Go, and Swift. Read NSA's memory safety information sheet for more.
- **Secure Hardware Foundation.** Incorporate architectural features that enable fine- grained memory protection, such as those described by Capability Hardware Enhanced RISC Instructions (CHERI) that can extend conventional hardware Instruction-Set Architectures (ISAs), as well as other features like Trusted Platform Modules and Hardware Security Modules. For more information visit, University of Cambridge's CHERI webpage.
- **Secure Software Components (SSDF PW 4.1).** Acquire and maintain well-secured software components (e.g., software libraries, modules, middleware, frameworks) from verified commercial, open source, and other third-party developers to ensure robust security in consumer software products.
- **Web template frameworks (SSDF PW.5.1).** Use web template frameworks that implement automatic escaping of user input to avoid web attacks such as cross-site scripting.
- **Parameterized queries (SSDF PW 5.1).** Use parameterized queries rather than including user input in queries, to avoid SQL injection attacks.
- **Static and dynamic application security testing (SAST/DAST)** (SSDF PW.7.2, PW.8.2). Use these tools to analyze product source code and application behavior to detect error-prone practices. These tools cover issues ranging from improper management of memory to error prone database query construction (e.g., unescaped user input leading to SQL injection). SAST and DAST tools can be incorporated into development processes and run automatically as part of software development. SAST and DAST should complement other types of testing, such as unit testing and integration testing, to ensure products comply with expected security requirements. When issues are identified, manufacturers should perform root-cause analysis to systemically address vulnerabilities.

- **Code review** (SSDF PW.7.1, PW.7.2). Strive to ensure that code submitted into products goes through quality control techniques such as peer review by other developers or "error seeding."
- **Software Bill of Materials (SBOM)** (SSDF PS.3.2, PW.4.1). Incorporate the creation of SBOM[4] to provide visibility into the set of software that goes into products.
- **Vulnerability disclosure programs** (SSDF RV.1.3). Establish vulnerability disclosure programs that allow security researchers to report vulnerabilities and receive legal safe harbor in doing so. As part of this, suppliers should establish processes to determine root causes of discovered vulnerabilities. Such processes should include determining whether adopting any of the secure by design practices in this document (or other similar practices) would have prevented the introduction of the vulnerability.
- **CVE completeness.** Ensure that published CVEs include root cause or common weakness enumeration (CWE) to enable industry-wide analysis of software security design flaws. While ensuring that every CVE is correct and complete can take extra time, it allows disparate entities to spot industry trends that benefit all manufacturers and customers. For more information on managing vulnerabilities, see CISA's Stakeholder-Specific Vulnerability Categorization (SSVC) guidance.
- **Defense-in-Depth.** Design infrastructure so that the compromise of a single security control does not result in compromise of the entire system. For example, ensuring that user privileges are narrowly provisioned, and access control lists are employed can reduce the impact of a compromised account. Also, software sandboxing techniques can quarantine a vulnerability to limit compromise of an entire application.
- **Satisfy Cybersecurity Performance Goals (CPGs).** Design products that meet basic security practices. CISA's Cybersecurity Performance Goals outline fundamental, baseline cybersecurity measures organizations should implement. Additionally, for more ways to strengthen your organization's posture, see the UK's Cyber Assessment Framework which shares similarities to CISA's CPGs. If a manufacturer fails to meet the CPGs— such as not requiring phishing-resistant MFA for all employees— then they cannot be seen as delivering secure by design products.

The authoring organizations recognize that these changes are significant shifts in an organization's posture. As such, their introduction should be prioritized based on tailored threat modeling, criticality, complexity, and business impact. These practices can be introduced for new software and incrementally expanded to cover additional use cases and products. In some cases, the criticality and risk posture of a certain product may merit an accelerated schedule to adopt these practices. In others, practices can be introduced into a legacy codebase and remediated over time.

[4] Some of the authoring organizations are exploring alternate approaches to gaining security assurances around the software supply chain.

SECURE BY DEFAULT TACTICS

In addition to adopting secure by design development practices, the authoring organizations recommend software manufacturers prioritize secure by default configurations in their products. These should strive to update products to conform to these practices as they are refreshed. For example:

- **Eliminate default passwords.** Products should not come with default passwords that are universally shared. To eliminate default passwords, the authoring organizations recommend products require administrators to set a strong password during installation and configuration or for the product to ship with a unique, strong password for each device.

- **Mandate multifactor authentication (MFA) for privileged users.** We observe that many enterprise deployments are managed by administrators who have not protected their accounts with MFA. Given that administrators are high value targets, products should make MFA opt-out rather than opt-in. Further, the system should regularly prompt the administrator to enroll in MFA until they have successfully enabled it on their account. Netherlands' NCSC has guidance that parallels CISA's, visit their Mature Authentication Factsheet for more information.

- **Single sign-on (SSO).** IT applications should implement single sign on support via modern open standards. Examples include Security Assertion Markup Language (SAML) or OpenID Connect (OIDC.) This capability should be made available by default at no additional cost.

- **Secure Logging.** Provide high-quality audit logs to customers at no extra charge or additional configuration. Audit logs are crucial for detecting and escalating potential security incidents. They are also crucial during an investigation of a suspected or confirmed security incident. Consider best practices such as providing easy integration with security information and event management (SIEM) systems with application programming interface (API) access that uses coordinated universal time (UTC), standard time zone formatting, and robust documentation techniques.

- **Software Authorization Profile.** Software suppliers should provide recommendations on authorized profile roles and their designated use case. Manufacturers should include a visible warning that notifies customers of an increased risk if they deviate from the recommended profile authorization. For example, medical doctors can view all patient records, but a medical scheduler has limited access to certain information that is required for scheduling appointments.

- **Forward-looking security over backwards compatibility.** Too often, backwards-compatible legacy features are included, and often enabled, in products despite causing risks to product security. Prioritize security over backwards compatibility, empowering security teams to remove insecure features even if it means causing breaking changes.

- **Track and reduce "hardening guide" size.** Reduce the size of "hardening guides" that are included with products and strive to ensure that the size shrinks over time as new versions of the software are released. Integrate components of the "hardening guide" as the default configuration of the product. The authoring organizations

recognize that shortened hardening guides result from ongoing partnership with existing customers and include efforts by many product teams, including user experience (UX).

- **Consider the user experience consequences of security settings.** Each new setting increases the cognitive burden on end users and should be assessed in conjunction with the business benefit it derives. Ideally, a setting should not exist; instead, the most secure setting should be integrated into the product by default. When configuration is necessary, the default option should be broadly secure against common threats.

The authoring organizations acknowledge these changes may have operational effects on how the software is employed. Thus, customer input is critical in balancing operational and security considerations. We believe that developing written roadmaps and executive support that prioritize these ideas into an organization's most critical products is the first step to shifting towards secure software development practices. While customer input is important, we have observed important cases where customers have been unwilling or unable to adopt improved standards, often network protocols. It is important for the manufacturers to create meaningful incentives for customers to stay current and not allow them to remain vulnerable indefinitely.

HARDENING VS LOOSENING GUIDES

Hardening guides may result from the lack of product security controls being embedded into a product's architecture from the start of development. Consequently, hardening guides can also be a roadmap for adversaries to pinpoint and exploit insecure features. It is common for many organizations to be unaware of hardening guides, thus they leave their device configuration settings in an insecure posture. An inverted model known as a loosening guide should replace such hardening guides and explain which changes users should make while also listing the resulting security risks. These guides should be written by security practitioners who can explain the tradeoffs in clear language to increase the chances of them being applied correctly.

Rather than developing hardening guides that list methods for securing products, the authoring organizations recommend software manufacturers shift to a secure by default approach and providing "loosening guides." These guides explain the business risk of decisions in plain, understandable language, and can raise organizational awareness of risks to malicious cyber intrusions. Security tradeoffs should be determined by the customers' senior executives, balancing security with other business requirements.

RECOMMENDATIONS FOR CUSTOMERS

The authoring organizations recommend organizations hold their supplying software manufacturers accountable for the security outcomes of their products. As part of this, the authoring organizations recommend that executives prioritize the importance of purchasing secure by design and secure by default products. This can manifest through establishing policies requiring that IT departments assess the security of software before it is purchased, as well as empowering IT departments to push back if necessary. IT departments should be empowered to develop purchasing criteria that emphasize the importance of secure by design and secure by default practices (both those outlined in this document and others developed by the organization). Furthermore, IT departments should be supported by executive management when enforcing these criteria in purchasing decisions. Organizational decisions to accept the risks associated with specific technology products should be formally documented, approved by a senior business executive, and regularly presented to the board of directors.

Key enterprise IT services that support the organization's security posture, such as the enterprise network, enterprise identity and access management, and security operations and response capabilities, should be seen as critical business functions that are funded to align with their importance to the organization's mission success. Organizations should develop a plan to upgrade these capabilities to leverage manufacturers that embrace secure by design and secure by default practices.

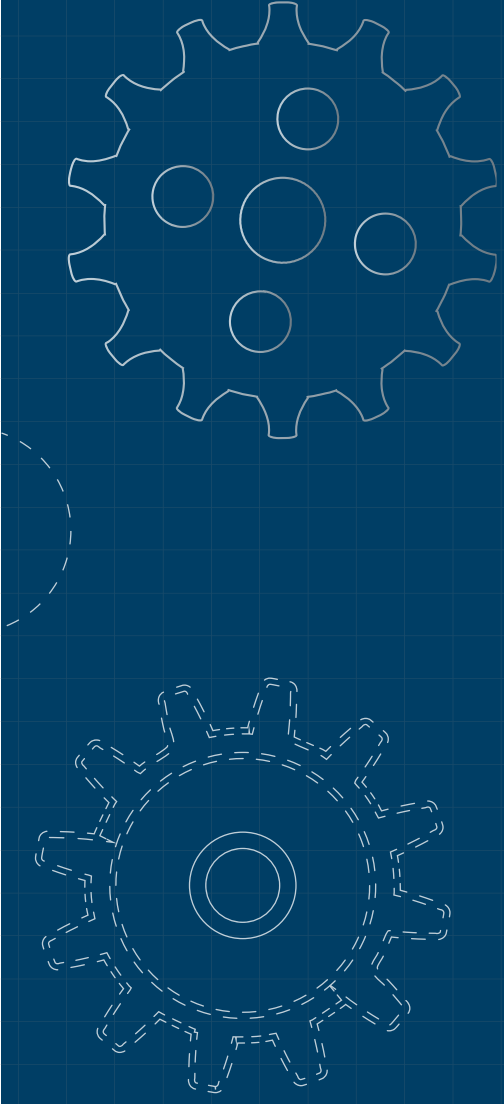
Where possible, organizations should strive to forge strategic relationships with their key IT suppliers. Such relationships include trust at multiple levels of the organization and provide vehicles to resolve issues and identify shared priorities. Security should be a critical element of such relationships and organizations should strive to reinforce the importance of secure by design and secure by default practices in both the formal (e.g., contracts or vendor agreements) and informal dimensions of the relationship. Organizations should expect transparency from their technology suppliers about their internal control posture as well as their roadmap towards adopting secure by design and secure by default practices.

In addition to making secure by default a priority within an organization, IT leaders should collaborate with their industry peers to understand which products and services best embody these design principles. These leaders should coordinate their requests to help manufacturers prioritize their upcoming security initiatives. By working together, customers can help provide meaningful input to manufacturers and create incentives for them to prioritize security.

When leveraging cloud systems, organizations should ensure they understand the shared responsibility model with their technology supplier. That is, organizations should have clarity on the supplier's security responsibilities rather than just the customer's responsibilities.

Organizations should prioritize cloud providers that are transparent about their security posture, internal controls, and ability to live up to their obligations under the shared responsibility model.

DISCLAIMER



Resources

CISA

- CISA's SBOM Guidance
- CISA's Cross-Sector Cybersecurity Performance Goals
- Guidelines on Technology Interoperability
- CISA and NIST's Defending Against Software Supply Chain Attacks
- The Cost of Unsafe Technology and What We Can Do About It | CISA
- Stop Passing the Buck on Cybersecurity: Why Companies Must Build Safety Into Tech Products (foreignaffairs.com)
- CISA's Stakeholder-Specific Vulnerability Categorization (SSVC) Guidance
- CISA's Phishing Resistant MFA Fact Sheets
- Cyber Guidance for Small Businesses | CISA

NSA

- NSA's Cybersecurity Information Sheet on Memory Safety
- NSA's ESF Securing the Software Supply Chain: Best Practices for Suppliers

FBI

- Understanding and Responding to the SolarWinds Supply Chain Attack: The Federal Perspective
- The Cyber Threat - Response and Reporting
- FBI's Cyber Strategy

National Institute of Standards and Technology (NIST)

- NIST's Digital Identity Guidelines
- NIST's Cyber Security Framework
- NIST's Secure Software Development Framework (SSDF)

Australian Cyber Security Centre (ACSC)

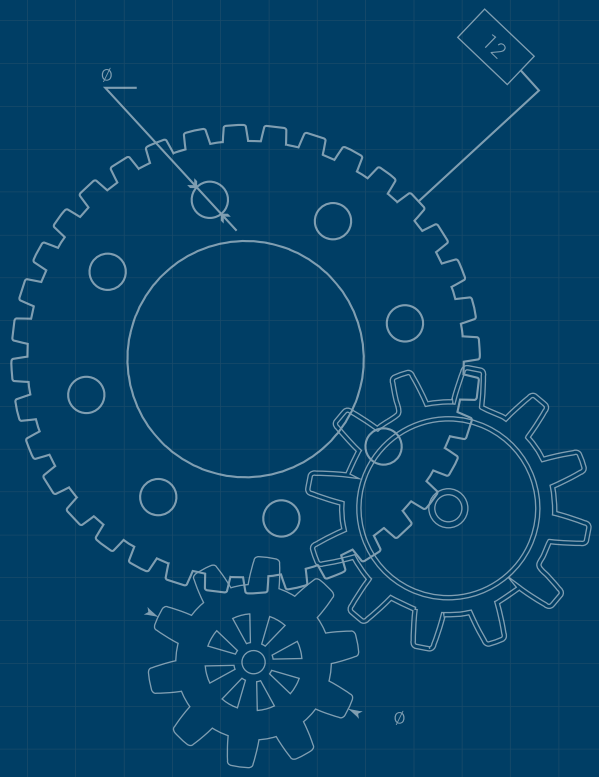
- ACSC's IoT Code of Practice Guidance for Manufacturers

The United Kingdom's National Cyber Security Centre (UK)

- The UK's Cyber Assessment Framework
- The UK NCSC's Secure Development and Deployment guidance
- The UK NCSC's Vulnerability Management guidance
- The UK NCSC's Vulnerability Disclosure Toolkit
- University of Cambridge's CHERI
- So long and thanks for all the bits - NCSC.GOV.UK

Canadian Centre for Cyber Security (CCCS)

- CCCS's Guidance on Protecting Against Software Supply Chain Attacks
- Cyber supply chain: An approach to assessing risks
- Canadian Centre for Cyber Security's CONTI ransomware guidance



Germany's Federal Office for Information Security (BSI)

- The BSI Grundschutz compendium (module CON.8)
- The international standard IEC 62443, part 4-1
- State of IT-security in Germany report, 2022
- BSI practices of web application security

Netherland's National Cyber Security Centre

- NCSC-NL's Mature Authentication Factsheet

Japan's National Center of Incident Readiness and Strategy for Cybersecurity (NISC)

- Japan's National Cybersecurity Strategy

Japan's Ministry of Economy, Trade and Industry (METI)

- Guide of Introduction of Software Bill of Materials (SBOM) for Software Management
- Collection of Use Case Examples Regarding Management Methods for Utilizing OSS and Ensuring Its Security

Cyber Security Agency of Singapore

- Technical Advisory on Secure API Development
- CSA SingCERT Vulnerability Disclosure Policy
- CSA SingCERT Incident Response Checklist
- CSA SingCERT Incident Response Playbooks
- CSA Security by Design Framework
- CSA Security by Design Framework Checklist
- CSA Guide to Cyber Threat Modelling
- CSA Cybersecurity Labelling Scheme

Other

- How Complex Systems Fail
- The New Look in complex system failure

REFERENCES

[1] https://csrc.nist.rip/publications/history/ande72.pdf

[2] https://www.cisa.gov/sbom and SBOMs references in TR 03183-2 https://www.bsi.bund.de/dok/TR-03183

[3] Juran on Quality by Design by J.M. Juran, 1992.