SAFECode response to NTIA RFC on Promoting Stakeholder Action Against Botnets and other Automated Threats

SAFECode is pleased to have this opportunity to provide our comments in response to the NTIA Request for Comments on Promoting Stakeholder Action Against Botnets and Other Automated Threats (RIN 0660-XC035).

About SAFECode

The Software Assurance Forum for Excellence in Code (SAFECode) is a non-profit organization dedicated to increasing trust in information and communications technology products and services through the advancement of effective software assurance methods. SAFECode is a global, industry-led effort to identify and promote best practices for developing and delivering more secure and reliable software, hardware and services.

The members of SAFECode create and deliver systems and software that are used in a wide variety of environments ranging from desktop applications to critical infrastructures to consumer and enterprise security. While the software that SAFECode members create is used in a large number of different ways, in today's world of connected systems, the need for security is a common thread across all of our members and their customer environments.

One of SAFECode's missions is to bring member companies' software security experts together to create practical guidance that organizations and individuals can use to create more secure software. Over the last several years, SAFECode has released concrete, actionable guidance documents and over fifteen online training courses as free downloads on its website.

Our response to the RFC deals primarily with software security assurance, and addresses the RFC's question about fostering prompt adoption of secure development practices. By software security assurance, we refer to the steps that development organizations (including personnel from decision makers to individual software developers) take to make their software resistant to attacks, and to respond effectively to discovered vulnerabilities.

Given the current state of software technology, it is not feasible to create software that meets users' functional requirements and is free from all vulnerabilities. But software security assurance makes a significant difference: without development organizations' commitment to software security assurance, the need for response to vulnerabilities and attacks arises more frequently, and the vulnerabilities that must be addressed are more serious.

Software Security Assurance

Over the years since the formation of SAFECode, our members have identified a set of common principles that apply to software security assurance:

1. Secure development is an organizational commitment and a holistic process, not an afterthought or a task that gets delayed until the very end.
2. There is no one-size-fits-all approach to software security assurance.
3. Despite differences, common secure development practices shared across the industry have proven both practical and effective.

4. Providing transparency of software assurance processes and practices helps customers and other key stakeholders manage risk effectively.
5. Sharing information about processes and practices supports developers' efforts to advance software security assurance and positively impacts the security and reliability of the technology ecosystem.
6. Software security assurance should become a part of any software engineering education program – it should not be confined to education of security specialists.

We believe that the government should make it clear that secure development is expected of suppliers of systems that will be connected to the Internet. In some cases, for example applications in critical infrastructures or national security systems, it may be appropriate for government to mandate secure development and to require attestation or certification of conformance with international consensus standards. In response to such an expectation or requirement, individual development organizations will need to establish their own software security assurance processes, and consistent with the second principle listed above, those processes will vary from organization to organization as a function of specific products, technologies, and development processes. However, any effective secure development process has these common attributes:

1. It reflects the application of software security best practices such as those identified in SAFECode's "Fundamental Practices for Secure Software Development, 2nd Edition"[i].
2. As mentioned above, the specific practices applied by a development organization are tailored to its specific products, technologies, and development processes.
3. The secure development process is reflected in delivered software. For example, the development organization uses tools that constrain the kinds of software code that it produces, and that find potential security vulnerabilities that the organization then remedies before it delivers software to customers.
4. The secure development process is continually improved in response to the discovery of software vulnerabilities. The development organization not only fixes specific vulnerabilities, it also updates its processes, tools, and developer training so that similar vulnerabilities do not occur in other software that it delivers to customers.

While large organizations, such as many of the members of SAFECode, can be expected to create and sustain secure development processes tailored to their needs, smaller organizations should be expected to take advantage of free guidance documents, such as those released by SAFECode, in creating their secure development processes. Such organizations can also use commercial, free, or open source tools that help ensure the security of software during development. There may be a need for a "lightweight" software security assurance process that smaller organizations that develop low-risk products of limited functionality can apply, but it is unclear how to distinguish the organizations and functionality where such a process would be appropriate. SAFECode would be willing to work with NIST and other industry organizations to help clarify such distinctions and create an appropriate process.

Assessing a Software Security Assurance Process

Given the wide variation in technology products and development processes, customers and regulators may ask how they can determine whether a particular development organization is actually implementing an effective software security assurance process. Such a determination may be especially

challenging for consumers who lack the technical sophistication – or interest – to delve into a development organization's process.

Technically sophisticated customers or government agencies may wish to apply the guidance in SAFECode's "Principles of Software Assurance Assessment"[ii]. This guide points out that organizations that implement secure development processes will be likely to release documentation describing those processes, and that customers should be able to determine whether the process described reflects the common attributes summarized above.

An emerging ISO Standard, ISO 27034 will provide a basis for independent certification of conformance with software security assurance best practices in the future. While the ISO 27034 conformance framework is not yet in place, a developer's attestation to conformance should carry some weight (see the discussion of government's role below).

A series of standards, technical reports, and related information released as ISA/IEC-62443 define procedures for implementing electronically secure Industrial Automation and Control Systems (IACS). These standards apply to asset owners, system integrators, security practitioners, and control systems manufacturers responsible for manufacturing, designing, implementing, or managing IACS. As IACS become interconnected and connected to the Internet, this guidance helps to ensure that parties involved in the supply chain and life cycle of a system work together to make and keep the systems secure.

Security testing tools and independent penetration testing may be beneficial, though both the conduct of such tests and the developers' responses to "false positives" from testing tools can be costly, and the tests themselves provide at best an imperfect assessment of products' security. It is not possible to "test security into" a product that has not been developed using a secure development process, but security testing (which is not the same as general functional testing) can identify specific security vulnerabilities, and the results of security testing can give an indication that a security assurance process has been used by showing the absence of repeated instances of common developer errors. The emerging work of the Cyber Independent Testing Laboratory (Cyber ITL)[iii] is promising, though not yet demonstrated at scale.

Using and Updating Software

Even if software is created using a mature and effective security assurance process, its security in use depends on the choices and actions of system integrators, administrators, and end users. These stakeholders play an important role in configuring software, managing security options, and making security-related decisions on a day-to-day basis.

One critical responsibility that is shared among developers, integrators, administrators, and end users has to do with updating. As we mentioned above, the experience of SAFECode members, and the industry in general, has demonstrated that it is not possible to achieve perfect security in software that is of size and complexity sufficient to meet the needs of real-world users. Thus, software must incorporate mechanisms to allow the application of security updates. Most software developers recognize this fact and provide a secure mechanism for installation of updates. Guidance to users should strongly discourage the use of software that cannot be updated.

It is important to be able to update all of the software components that may be incorporated in a product. For example, if a software application uses operating system software in a way that the

operating system developer did not anticipate, or makes changes to the operating system, it may be impossible to install a security update that eliminates a vulnerability in the operating system. Integrators and application developers should ensure that software they provide can be updated, should refrain from using software such as operating systems in unanticipated ways that "break" updating capabilities, and should consider how any modifications they must make will adapt to the installation of security updates.

Modern software applications frequently incorporate independently developed libraries or other components (commonly referred to as "Third Party Software"), and these components often require security updates as well. Software developers should carefully consider, and plan for security updates to the third-party components on which their software depends. This planning should also address potential needs to replace software components that are no longer supported. SAFECode has recently released a guide to help application developers understand and manage security risks due to the use of third-party components[iv].

The Role of Government

Today, the US government is largely silent about software security assurance and the importance of secure development. A government endorsement of practical guidance such as that contained in the Fundamental Practices and Third Party Component documents would help to send developers a message that secure development is both important and achievable.

ISO 27034, ISA/IEC-62443, and the work of the Cyber ITL represent emerging approaches that may lead to third-party certifications for secure development practices, and the government should monitor progress on these standards and processes. Today, none is sufficiently mature to be the basis of a government certification program, and there is no standard tailored to smaller development organizations and limited functionality products that pose a lower security risk.

Development organizations' descriptions of their secure development practices, including declarations of conformance with ISO 27034 or ISA/IEC-62443, can provide customers and the government with a degree of assurance that secure development practices are being applied. Recent FTC attention to the importance of product security is likely to ensure that developers do not make claims about secure development unless there are actually secure development programs behind such claims. It would be appropriate for NIST to participate in an effort to create a consensus standard that defined a software security assurance process for smaller development organizations and lower risk products.

---

[i] Fundamental Practices for Secure Software Development, 2nd Edition, SAFECode, 2011, available at http://www.safecode.org/wp-content/uploads/2014/09/SAFECode_Dev_Practices0211.pdf

[ii] Principles of Software Assurance Assessment, SAFECode, 2015, available at http://www.safecode.org/publication/SAFECode_Principles_for_Software_Assurance_Assessment.pdf

[iii] See http://cyber-itl.org/

[iv] Managing Security Risks Inherent in the Use of Third Party Components, SAFECode, 2017, https://www.safecode.org/wp-content/uploads/2017/05/SAFECode_TPC_Whitepaper.pdf