Consumer Software Labeling Program (CSLP)

Jeff Williams, Veteran Cybersecurity Industry Leader - jeff.williams@contrastsecurity.com

Long ago (2004), I proposed creating "Software Facts" for software. Later (2010), I analyzed a variety of different labeling regimes to learn what might work for security. For this position paper, I once again scoured the literature for labeling studies. There are dozens of informative examples here, including labeling programs covering food, menus, tobacco, drugs, sweeteners, music, movies, cars, vehicles, energy efficiency, vending machines, cleaning products, alcohol, GMOs, recyclables, and more.

However, while there are an overwhelming number of studies, they are all very specific to a particular context. The unfortunate outcome is an explosion of disparate and often apparently contradictory results. To make sense of it all, I strongly recommend a close reading of the survey of the literature by Bonros and Constantato. It's literally impossible to provide anywhere near a comprehensive set of recommendations in a two page position paper. Nevertheless, given the characteristics of the software market: massively asymmetric information, high competition, and complex supply chains, I offer the following observations and recommendations for NIST to consider for a Consumer Software Labeling Program (CSLP).

NIST - Take a Broad View of Consumer Software. Consumer software isn't just programs that run on a desktop computer. Most of the risk to consumers comes from software running on remote servers, largely web apps and web APIs that connect to other backend systems like databases and mainframes. Consumers need labels on this software so they can make informed decisions when choosing providers.

NIST - Your Goal Is Influencing the Software Market. NIST should focus CSLP on resolving the asymmetric information failure in the software market. Currently, consumers generally can't tell the difference between secure software and insecure software, and therefore won't pay extra for security. Ultimately, this creates a "market for lemons" where all software is insecure. A CSLP can help to correct this market failure by ensuring all players in the market have the same access to security information about software.

NIST - Make Labels Mandatory. NIST should take the bold stand that all software producers ***must*** disclose basic information about the security of their software. Mandatory labeling is justified because the overall societal benefit will dramatically exceed the cost of breaches to both enterprises and individuals. Studies show that only high quality products participate in voluntary labeling, and consumers tend to choose unlabeled, weaker products, rather than higher quality labeled products.

NIST - Let Producers Create Their Own Labels. The speed and scale of modern software development make the idea of creating centralized testing/certification programs impossible. However, producers creating inaccurate labels may be exposed to both market and legal repercussions.

NIST - Focus on Software Producers Not Consumers. Informing consumers is great -- but improving the security of software for everyone changes everything. Many studies show that disclosing information through a label results in producers improving their products. Engineering, marketing, sales, and legal teams simply won't allow products to be sold with obviously negative information on the label. This

effect happens more quickly than changes in consumer behavior, and would represent a huge advance in the software industry. This is what we need!

NIST - Provide Context for Processes and Testing. The labels should reflect the whole picture. Imagine a company claiming they did great security testing, but their threat model only covered clickjacking. I wouldn't trust a label that doesn't create some kind of line of sight spanning a strong threat model, great security architecture, strong defenses, a safe supply chain, and runtime protection.

NIST - Target a Very High Level Label. Based on other labeling regimes and the nature of the software and security market, we believe that a very high level label will be effective with consumers and still influence producers. High level labels will also minimize the new criteria required and make it easier for producers to adopt and produce. Additional detail could be added in the future.

NIST - Reflect the Assurance Case. Ideally, labels would reflect a balanced view of all the aspects of a software assurance case. For example, one high level label concept could use simple letter grades for each of these six key assurance argument pillars:

- Did you identify all the expected threats?
- Does your software have defenses against those threats?
- Did you verify that those defenses are correct and effective?
- Can your software detect attacks and defend against them?
- Did you verify the security of your software supply chain?
- Are there any unresolved security issues in any part of the software?

NIST - Avoid Including "Indirect" Assurance Evidence. Labels should be based largely on "direct" evidence that measures the software itself in some way. "Indirect" evidence about the people, processes, and technology used in development doesn't really tell you very much about whether the software itself is actually secure. The label should be based on the assurance evidence that is produced by this "software factory," not the existence of the factory itself.

NIST - Include Both Claims and Warnings. Some software deserves warnings as part of the label. If software contains serious vulnerabilities, whether from custom code or dependencies, their existence should be revealed as part of the label.

NIST - Make Labels Obvious. Security labels won't change anything if people don't see them. Different types of software may require different label placement in order to get seen. Consider mandating a standard URL pattern, such as /security, that always contains security label information, perhaps in a machine-readable format. Additional and supporting information can be provided in addition to the label. For example, the expected threats could be described along with details about defenses and testing for each.