REF: EB-2016-0032

To Whom It May Concern:

In 2010 the world received a glimpse of how devastating cyber security could be. The Stuxnet worm damaged the Iranian nuclear power program and provided a window into the kinds of programs state actors had been waging for years. Unlike big name corporate breaches of years prior, this breach proved to the general public that credit cards and identity theft were not the most pressing concerns of cyber security: human lives were potentially at stake.

In the years that followed, the public continued to be hammered with headlines of named security incidents like Heartbleed, WannaCry, and Petya. Governments and standards organizations around the world began to build cybersecurity standards to address the clear, growing threats.

Some of these standards or frameworks, like the NIST CyberSecurity standard, formed the basis for other security

programs. While the content of these standards vary, they all have a major impact on shaping security programs within companies. We find many controls across these disparate standards: security policy patch management, and incident management to name just a few.

Broadly, however, they fail to address one of the most critical sources of cybersecurity risk: vulnerable software. The risk of vulnerable software from the supply chain is real. Stuxnet, WannaCry, Heartbleed, Petya and Heartbleed all had root causes in software flaws. Nearly all vulnerabilities listed in MITRE's Common Vulnerability & Exposure (CVE) database - a list of well known security flaws that form the basis for most hacks - are rooted in software flaws. Shockingly, nearly all of these flaws are preventable with mitigations listed in the companion Common Weaknesses Enumeration database. Most standards deal with this risk by stressing that organizations should have processes and tools to mitigate the risk of vulnerable software. One example is patch management, which is a systematic process to update software when vendors release updates which contain fixes to security holes. These kinds of processes are notoriously difficult

for organizations to execute on, which leads to a significant number of breaches. Organizations then need to rely on a whole host of additional controls to detect and block attacks in real time or work quickly to respond to a breach after it occurs. The composition of an average cybersecurity program is largely to make-up for shortcomings in software. No single control is a silver bullet, and as breach after breach shows us, even all the security controls in aggregate are often not enough to stop real damage. Yet, regulatory standards and best practices continue to place strong emphasis on these controls with almost no emphasis on secure software development. The widely referenced NIST Cybersecurity Framework , which forms the basis for the OEB's Cyber Security Framework, describes a large number of key activities as part of a cybersecurity program but makes no mention of secure software. It's entirely possible to be compliant with this framework without having a software security program or reviewing the software security practices of your software vendors.

The downstream effect is that there is little incentive for most software vendors to improve their secure development practices. Data shows that instead of holistically incorporating security into

their software development processes, most organizations - including software vendors- simply rely on automated scanning solutions that cover a small fraction of real risk. Their customers aren't asking and compliance standards aren't mandating them to build security into software, so they aren't doing it.

There are several industry standards that software vendors could look to for taking a holistic approach to software security: the ISO 27034, Microsoft's SDL or the Building Security In Maturity Model (BSIMM). Yet apart from the largest vendors, very few do. Most people in information security haven't even heard of the ISO 27034 standard, and very few vendors have adopted it because there is no customer or regulatory pressure to do so. Chief Information Security Officers (CISO) put secure development 14th out of their top 17 priorities, which effectively means it doesn't get any attention at all. This is despite clear evidence that secure development processes are cost effective and significantly reduce risk. In the domain of control systems, the IEC 62443-3-3 provides detailed system security requirements which include both software and hardware level controls.

Ten years ago David Rice wrote Geekonomonics: an entire book about the economic issue of vulnerable software. In the following decade, not much has changed. Clearly, market forces alone will not solve the problem. The people who write and update cybersecurity standards, laws and frameworks to pay attention to this significant problem. These documents form the basis for most cybersecurity programs worldwide. We need the people who audit and enforce these standards to increase their emphasis on security of in house software development practices and make sure that companies are reviewing the secure development posture of their vendors. While we will never produce perfect software, we can make a significant reduction in the number of security incidents by preventing well-known software security flaws. Without making these critical changes, we can be sure that the problem of cybersecurity will continue to grow exponentially.

Rohit Sethi, COO
Security Compass
rohit@securitycompass.com
1-888-777-2211, x.102