Editorial: Trust and Security Must Become a Primary Design Concern in Embedded Computing

As I write this editorial, a lot has been happening around us. Apple's secure enclave processor firmware has been decrypted; face recognition-based secure identification of iPhone X has been hacked, with relatively inexpensive masks, apparently; the Uber data breach compromised the privacy of 57 million people; the Equifax breach compromised even a larger number of people; and new Android Trojan malware were discovered in the Google Play store. These are but a few examples that have been published in the media. A lot more is happening and being suppressed. News reports suggest that Uber might have paid hackers to suppress a data breach. Other reports indicate that even an Indian telecom provider might have paid hackers for similar reasons. In India, a cloud of suspicion over electronic voting machines, popularly known as EVMs, have gathered after a few reports surfaced on all votes automatically going to a specific party. Even today, reports of that sort of compromise are coming in, as there is a hotly contested regional election taking place in India today. While the Election Commission of India has denied any such possibilities, they have not been forthcoming in offering experts to open these EVM machines and probe the hardware, firmware, and possible attack surfaces, leaving a section of politicians and the public suspicious. Given that Twitter bots and social media bots have been unleashed by the millions to sway public opinions, creating and propagating fake news and challenging the normal democratic processes, it is a scary new world that we are faced with today. Continually extending our digital footprint and moving all political, business, financial, social, and transactional activities to the digital world in search of efficiency, are we not exposing ourselves to an unknown future?

Why are these relevant to embedded computing? This is a journal of embedded computing, and in my editorial comments, I always seem to be obsessed with cyber security over all other challenges germane to embedded computing. Is it intentional to attract more submissions in the embedded security domain? My answer is that, subconsciously, possibly, but there is a good logical reason for this fear and concern. Furthermore, most devices that we find vulnerable, the same devices that affect our lives and society, are embedded computing devices—be it mobile phones, voting machines, automobiles, avionic controls, Internet of Things (IoT)-based automation, sensors and control systems in utilities, or medical devices such as pacemakers or insulin pumps.

While a large-scale data breach usually concerns IT systems, even there we saw a tangled web of cyber physical sensor network security affecting IT security. Take, for example, the Target data breach. The network for HVAC system monitoring was breached and became the conduit to propagate malware to point-of-sales machines, which led to the stealing of credit card information while being swiped for authorization.

Compounding these vulnerabilities and mounting threats, insider threats are also increasing through social engineering and other human-centric processes. A high-privilege system administrator might compromise individuals unless the system is designed to be robust against insider attacks. We have evolved our IT systems with the assumption of trusted central authority, or a trusted third party, and only became aware of the large percentage of internal attack incidents in major breaches.

Our Computing stack has also been designed to trust a lower abstraction layer—and only in the last two decades or so, we became conscious enough to evolve the notion of “trusted computing.” The notion of a “root-of-trust” anchored boot strapping of trusted computation is a powerful one—and that is where the enclave processor, hardware security module (HSM), Intel Software Guard Extensions (SGX), Cisco network enclave, and similar concepts originated.

However, these enclave-based security architectures need to be proven secure, and while the recent hack of the firmware encryption of the Apple enclave processor does not compromise the private keys or the enclave’s security, it seems to have created a lot of concern. The existence of Trojan malware laced apps in Google Play has raised similar concerns. Notwithstanding the sandboxing in Android, if an unpatched version of Android is on the mobile, even the dirty Copy on Write (COW) vulnerability in the Linux kernel could break the sandboxing. So, the upgrading of Android to the most recent version is becoming very urgent for all users.

In terms of internal attackers, thwarting malicious behavior by privileged users is finding some solutions in block chain technology. Immutable logging with public verifiability can provide some deterrence to malicious activities by a privileged user. Also, using block chain to implement a distributed Domain Name Service (DNS) or a distributed Public Key Infrastructure (PKI) could be seen as an alternative to dependence on a trusted third party—when the trust in third parties, especially in the hands of untrusted entities, are at an all time low.

There was a time when we trusted the hardware and Operating system, and we only assumed applications being downloaded or purchased were untrustworthy. From there, today, we cannot trust any vendor or manufacturer, given the outsourcing nature of the business and the increasing number of threats and enhanced attack surfaces. Even with the proliferation of IoT devices, we cannot even trust the devices around us, as we saw in the case of Mirai Botnet, given the vulnerable devices of questionable security. We cannot trust our medical devices—our pacemakers or insulin pumps. We are unable to trust our democracy when executed through electronic voting.

It is, thus, high time that embedded security takes precedence over optimization—be it cost, power, or latency. However, the choice cannot always be made as real-time requirements often get in the way. Therefore, new methods and techniques for trust anchoring are necessary. Sandboxing, authentication, security certificates, anti-malware scanning—none of these provide enough security.

Even though we get a reasonable number of embedded security-based submissions in this journal, I think out-of-the-box solutions are required, and more provable security solutions are needed.

Finally, this is the first issue of Volume 17. This is slightly delayed because we had a very special online issue publishing all accepted papers of ESWEEK 2017. This included all papers in the three major conferences, including EMSOFT, CODES + ISSS, and CASES. The program committees worked very hard to act as guest editors for the journal to select the papers, having two rounds of reviews. This was the first time ESWEEK publication model was “journal only”—in order to avoid duplication of content in conference proceedings and journals, and saving the scarce resource of review bandwidth by reviewers to review papers twice, once for the conference and again for the journal—albeit with a 30% increase in material. This special issue was published in October 2017 right before the ESWEEK conference in Seoul, South Korea.

So now we are ready with the first issue of Volume 17. Recall that we are now going bi-monthly, unlike quarterly in the past. This means there will be 6 issues of Volume 17 onward.

Volume 17:1 has two special issues—one on battery-free sensing and communications technologies, and the other with invited best articles of ESWEEK 2016. I thank the guest editors of both

these special issues for the enormous effort they have put in to get the submissions reviewed and make the selections. We also have many regular articles on a variety of topics.

Sandeep K. Shukla

Indian Institute of Technology Kanpur

December 9, 2017

Editor-in-Chief