In IoT We Trust?

Voas/Kuhn/Laplante

(*for IEEE IoT Newsletter*)

No.

IoT is an acronym comprised from three letters: (I), (o), and (T). But the Internet (I) has never been highly associated with the term 'trust.' Identity theft, false information, a breakdown in personal privacy, and so many other negative features of (I) cause some people to avoid the Internet altogether. But for most people, avoidance is not an option. The other letter of importance here is the one associated with 'things' (T). Similar trust concerns occur for (T). Why? Because the 'things' carry their own baggage of trust concerns and the interactions between 'things' exacerbate these concerns. In short, and from a trust standpoint, the IoT is an untrustworthy backbone with untrustworthy things attached -- a perfect storm [1].

In this short article, we'll review an abbreviated list of trust challenges that we foresee as increased adoption transforms the IoT into another ubiquitous technology just as the Internet is. These challenges are in no specific order, and are by no means a full set.

To begin, what do we mean by 'trust?' We will not use a formal definition, but rather a variation on the classical definition of reliability. Hence, we consider trust to be the probability that the intended behavior and the actual behavior are equivalent, given a fixed content and environment. For example, we can expect a trusted set of behaviors for a car operating on roadway (we cannot, however, expect such a set of behaviors for a car operating in a lake). This informal definition works well for both 'things' and systems of 'things'.

While subtle, we have just listed three key applications of trust: (1) trust in a 'thing', (2) trust in a system of 'things', and (3) trust that we are dealing with an appropriate environment and context. This brings us to another key ingredient related to trust -- the ability for the set of behaviors to be *bound*. Bounding does not apply to (1), but it does apply to (2) and (3). For (2), NIST offered a Special Publication (NIST SP 800-183) titled 'Networks of Things' [5] to discuss one way to bound a specific system of 'things' such that various metrics and measures of security and reliability could be assessed. The approach in NIST SP 800-183 was simple: define classes of 'things', essentially as building blocks. For trust application (3), bounding an environment is a difficult challenge, but unfortunately, necessary. Rarely will it ever make sense to make a claim such as: *this system of 'things' works perfectly for any environment, context, and for any anomalous event that the system can experience*.

So, let's look at a handful of trust-related issues that the reader may not yet have considered.

1. There is no universally accepted definition for IoT. Does any noun preceded by 'smart' (e.g. 'smart toy' or 'smart house' or 'smart city') define IoT? Clearly not, but you would not know it from the way many people talk about IoT. Assessing and measuring trust for an entity that is not defined is problematic.

2. Heterogeneity is a trust issue. Getting ‘things’ to connect and interoperate with other ‘things’ from other vendors is non-trivial. Heterogeneity of products and services from thousands of different vendors is terrific from a competition standpoint, but connecting a diverse set of components is rarely easy.
3. Getting the intended behavior is a trust issue. Even if we have no difficulty gluing ‘things’ to ‘things’, this only solves the architecture problem. It does not suggest that these composed ‘things’ will exhibit the intended composite behavior that we desire. Hardware and software components may or may not work well when integrated, depending on whether they were the right components to select, whether they had the proper security and reliability built-in, and whether the architecture/specification was correct. (Note there are subtle differences among integration, interoperability, compatibility, and composability.) Consider the following scenario: A hacked refrigerator's software interacts with an app on a person’s smartphone, installing a security exploit that can be propagated to other applications with which the phone interacts. The user enters their automobile and their phone interacts with the vehicle’s operator interface software, which downloads the new software, including the defect. Unfortunately, the software defect causes an interaction problem (e.g., a deadlock) that leads to a failure in the software-controlled safety system during a crash, leading to injury. The potential for this chain of events to occur demonstrates why interoperability is so challenging regarding identifying and mitigating risks and assigning blame when something goes wrong [4].
4. Certification of a product (not process or people) is a ‘grand challenge’ for just about anything, regardless of whether hardware, software, or system. And IoT is no exception [2]. IoT certification is nearly impossible unless the environment of a system of ‘things’ is bounded. (And to bound would be easier if we had a definition.) Also, consider the cost to certify a ‘thing’ relative to the value of that ‘thing’. Is certification an option for IoT-enabled systems? If so, who does it? Who certifies the certifier? What criteria are used? What does it cost? Are the benefits worth it with respect to time-to-market and cost-to-vet? What is the lifespan of a ‘thing’? Also, you must consider composability. Are the other ‘things’ in the system certified? If not, why not? Even if all ‘things’ are certified, that does not mean they will interoperate well (correctly) in a given environment. Certifying ‘things’, as standalone entities, does not solve the fundamental problem of trusting a system that resides in a specific environment. And what about third party limited warranties – do they still apply when components are interconnected?
5. IoT testing is a concern [3]. You can test ‘things’, systems of ‘things’, and subsystems of ‘things.’ You can test them in artificial environments or operational environments. In operational environments, systems of ‘things’ may only be bound-able for mere instants in time, therefore testing is problematic. Testing systems-at-rest is easier than testing systems that are reorganizing themselves in real-time and at massive scale. If you are testing a system of ‘things’ that relies on Internet connectivity, realize that the Internet at at any given time is different than the Internet even a millisecond later. This property of

constantly changing configurations may of course also hold for relatively small networks of things, isolated from the full Internet. Furthermore one of the biggest problems for the reliability of a system of 'things' during operational usage is data anomalies propagating through the system. This eventuality suggests that some form of off-nominal or fault injection testing should be considered, which is expensive.

6. IoT quality / security / reliability /etc. are unfortunately and mostly consumer concerns since regulated systems, e.g., commercial aircraft, already have rules for how to specify and test. Will IoT ever reach that level of maturity? And which 'ility' is most important? Reliability, Security, Privacy, Performance, Resilience, etc? Other trust considerations here include: (1) did you use a faulty or subpar architecture? (2) are you able to mitigate third party defective 'things' for which you have little or no control? (3) were the highest quality 'things' used and if so did you over-engineer and spend too much or were the lowest quality 'things' employed simply to save money?
7. External leased data is a concern – it may come from sensors owned and controlled by vendors and this data may be received by your system of 'things' at a time of their choosing and with an integrity level of their choosing. Will SLAs protect you? Are you able to mitigate faulty interfaces and communication protocols? Are you confident in your wireless service providers? And what about data tampering and data integrity? How secure is your data from accidental problems or malicious tampering, delay, or theft?

In summary, there are many IoT trust issues, of which we mentioned only a handful. But compromising even one of these trust issues destroys overall trust in the system. We are working on a more comprehensive listing to be incorporated in a new NIST publication later in 2018.

References

[1] I. Bojanova and J. Voas, 'Trusting the Internet of Things', Guest Editor Intro, *IEEE IT Pro*, October 2017

[2] J. Voas and P. Laplante, 'IoT's Certification Quagmire', *IEEE Computer*, April 2018

[3] J. Voas, R. Kuhn, and P. Laplante, 'Testing IoT-based Systems', 12th IEEE International Symposium on Service-Oriented System Engineering, March 26-29, 2018, Bamberg, Germany

[4] J. Voas and P. Laplante, The IoT Blame Game, *IEEE Computer*, Year:2017, Volume: 50, Issue: 6

[5] NIST SP 800-183, 'Networks of 'Things'', J. Voas, 2016