# Security in distributed telemetry and control networks

Fred Baker <fred@cisco.com>

The questions asked on the web page for the workshop were as follows:

- What techniques for issuing credentials have been deployed?
- What extensions are useful to make existing security protocols more suitable for smart objects?
- What type of credentials are frequently used?
- What experience has been gained when implementing and deploying application layer, transport layer, network layer, and link layer security mechanisms (or a mixture of all of them)?
- How can “clever” implementations make security protocols a better fit for constrained devices?
- Are there lessons we can learn from existing deployments?

and it was stated that this was a non-exhaustive list. Personally, I am glad it's non-exhaustive, as I suspect that it is starting from some assumptions (certificates) that may not be consistent with the most useful practice.

From my perspective, it is difficult to take a position on the security issues and mitigation techniques requirements of a generalized application in the absence of a threat model for it. There are some things that can be obviously stated regarding security in any application and any network, but it is difficult to get specific on the issues and mitigation techniques without first getting specific about the application and the network it is embodied in.

With that caveat, let me first take note of the CIA (Confidentiality, Integrity, Availability) and the ISO 7498-2 models. The former is commonly used and rolls the concepts of Authentication and Authorization into the concept of Availability. The latter was pointed out to me by Steve Kent, and defines five security services: confidentiality, authentication, integrity, access control (which I call "authorization"), and non-repudiation. I tend to think of both of those models as indicating the questions a security service must ask and answer, but I will then go on to point out that there are at least four layers in the communication model, and those questions have to be asked and answered at each.

If I were to phrase the questions taking layers into account, I might ask

- Why is your system permitted to communicate in my network?
- Why is your machine capable of and authorized to communicate with mine?
- Why is your application capable of and authorized to communicate with mine?
- After the communication event has passed, why are you able to access and understand the information that was exchanged?

The first question is the kind of question answered by IPsec in its tunnel mode, firewalls, PANA, and IEEE 802.1X. It may or may not involve a confidentiality component - one might encrypt the communications on a WiFi SSID - but it certainly involved authentication and authorization. The second is the kind of question addressed by network layer authentication (IPsec) and Transport Layer Authentication (TCP-AO). The third is the kind of question addressed by TCP-AO in another sense, TLS, or https. The last, applicable with medical or billing records, is the stuff of signed or encrypted data - RPKI with SBGP, or similar algorithms in application layer data, or mechanisms based on chain of trust.

In dealing with, to pick on one, confidentiality, one has to ask the question at each layer because different matters may be confidential. In medical records, for example, the content of the record is confidential (and so encrypted within a database or provided with other forms of access control) for privacy reasons, but access to the record may also be confidential for the same or different reasons. For example, if I have encrypted a heart patient's records and so concealed a diagnosis of heart disease, but don't conceal the access to those records by a heart surgeon, their content may as well be in the open - it's not that hard to guess what the issue is.

From my perspective, since there are different questions and different forms of the same questions at different layers, it is not sufficient to apply one tool and dismiss security as a problem. One needs an adequate threat analysis, and one needs an adequate response to the issues it raises. It is not necessary to use all of the possible mitigation techniques in each layer in each scenario, but if the question has not been asked and reasonably answered, there may be a security flaw.

As to "things to learn from current deployments", I think the security of bluetooth and Skype bears looking at. What we want is something that will allow a typical low-tech homeowner deploy a relatively secure system easily will be a procedure-based admission in which the device is told to "become discoverable", learn a new neighbor, and then become "undiscoverable" again. Under the hood, it might perform a Diffie-Helman exchange or otherwise build a reasonably strong credential that is shared by the two devices and unknown to the user. This is obviously not a "perfectly secure" system, but in practice it seems adequate for most residential uses.