Mechatronics security by design

Peter Thorne, Managing Director, Cambashi Ltd.

Engineers responsible for mechatronics development have always known that it's not just PCs that can suffer from malware. A study in 2011[1] used experiment rather than theory to identify vulnerabilities of in-vehicle automotive systems. Not only was this a strong reminder of the seriousness of the issue, but the study also established that prior physical access to the car was not necessary. Remote exploitation of the automotive software was feasible via a range of external interfaces – from malformed MP3 files on a CD to interception of the wireless connections used by tire pressure monitoring systems.

One learning point from this study was the observation that "…virtually all vulnerabilities emerged at the interface boundaries between code written by distinct organizations."

If you work with software for mechatronic devices, don't take this as a call to bring all the development in-house and rewrite all the code from scratch. It will be necessary to use existing, external software stacks that are established, proven and perhaps even certified for the capabilities they provide. But perhaps it is worth revisiting your plans for how you build the linking parts.

We all know that many exploits depend on finding something that is 'allowed', but not planned for – that is, the system 'accepts' an event or input that was not part of the specification. So, look again at the specifications of the external software components you are using. They probably come with very clear documentation of their interfaces –timing, data values, required call sequences to make things work correctly and so on. But what response is defined when timing, data values or call sequences are not as specified? These questions apply equally to situations where your software is taking the lead; and also to situations where your software is responding to data and events collected and identified by the third party software. The issue is whether unexpected conditions can cause unplanned and unpredictable responses.

To be certain these possibilities cannot provide pathways that malware can use, it is necessary for these ‘abnormal conditions’ to be impossible. Think carefully about the word ‘impossible’. Most of us assume it means ‘impossible in normal operation’ – that’s a dangerous interpretation in this context. So, if these conditions might occur, then the response of the system must be specified. This means specifying requirements for all the conditions that can’t / won’t / shouldn’t happen. Many of these can probably be grouped together, so the effort involved may not be quite as bad as you think. But without those requirements, there won’t be any logic to reject problem inputs, and your test people won’t have any reason to build tests that cover or sample these conditions. So the bad guys will be able to find a way in.

References

[1]: “Comprehensive Experimental Analyses of Automotive Attack Surfaces”, available without registration from http://www.autosec.org/publications.html [1].

Bio

Peter Thorne is Managing Director, Cambashi Ltd. He is responsible for consulting projects related to the new product introduction process, e-business, and other industrial applications of information and communication technologies.He works closely with IT vendors, providing them with independent expertise on matching their products and service to real industry needs. He has applied information technology to engineering and manufacturing enterprises for more than 20 years, holding development, marketing and management positions with both user and vendor organizations. Immediately prior to joining Cambashi in 1996, he headed the UK arm of a major IT vendor's Engineering Systems Business Unit, which grew from a small R&D group to a multi-million dollar profit center under his leadership.

Peter holds a Master of Arts degree in Natural Sciences and Computer Science from Cambridge University, is a Chartered Engineer, and a member of the British Computer Society.

Links:

[1] http://www.autosec.org/publications.html