Appendix A

Excerpt from 2014 Software Sector Meeting Summary
Software Sector Item 3, Software Protection/Security

3. Software Protection/Security

Source:
NTEP Software Sector

Background:
The sector agreed that *NIST Handbook 44* already has audit trail and physical seal, but these may need to be enhanced.

From the WELMEC Document:

> **Protection against accidental or unintentional changes**
> Metrologically significant software and measurement data shall be protected against accidental or unintentional changes.
>
> **Specifying Notes:**
> Possible reasons for accidental changes and faults are: unpredictable physical influences, effects caused by user functions and residual defects of the software even though state of the art of development techniques have been applied.
>
> This requirement includes consideration of:
>
> a) Physical influences: Stored measurement data shall be protected against corruption or deletion when a fault occurs or, alternatively, the fault shall be detectable.
>
> b) User functions: Confirmation shall be demanded before deleting or changing data.
>
> c) Software defects: Appropriate measures shall be taken to protect data from unintentional changes that could occur through incorrect program design or programming errors, e.g. plausibility checks.
>
> **Required Documentation:**
> The documentation should show the measures that have been taken to protect the software and data against unintentional changes.
>
> **Example of an Acceptable Solution:**
>
> - The accidental modification of software and measurement data may be checked by calculating a checksum over the relevant parts, comparing it with the nominal value and stopping if anything has been modified.
> - Measurement data are not deleted without prior authorization, e.g. a dialogue statement or window asking for confirmation of deletion.
> - For fault detection see also Extension I.

The sector continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTETC Software Sector Meeting. The information requested by this checklist is currently voluntary, however, it is recommended that applicants comply with these requests or provide specific information as to why they may not be able to comply. Based on this information, the checklist may be amended to better fit with NTEP's need for information and the applicant's ability to comply.

The California, Maryland and Ohio laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

The Maryland laboratory had particular questions regarding 3.1 and 5.1. The information for 3.1 could be acquired from an operator's manual, a training video, or in-person training. The items in 5.1 were confusing to the evaluators. The terminology is familiar to software developers, but not necessarily others. It was indicated that manufacturers were typically quick to return the filled out questionnaire, but he didn't know how his laboratory was supposed to verify that it was true. Generally, the laboratories wouldn't be expected to verify things to that level. For example, if the manufacturer states that a checksum is used to ensure integrity, the laboratories wouldn't be expected to evaluate the algorithm used.

The intent was to see whether the manufacturer had at least considered these issues, not for evaluators to become software engineers. Perhaps a glossary or descriptive paragraphs might be added to assist the evaluators for if the manufacturer has questions for the evaluators.

OIML makes use of supplementary documents to explain the checklist they use. Below are links:
http://www.oiml.org/publications/D/D031-e08.pdf
http://www.welmec.org/latest/guides/72.html
http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf

WELMEC document 2.3 is the original source for our checklist, but it's been significantly revised and simplified. Mr. Payne, Maryland Department of Agriculture, is going to review the other documents and come up with some suggestions for the checklist. Mr. Roach, California Division of Measurement Standards, is going to begin using the checklist. The international viewpoint is that any device running an operating system is considered to be Type U. Mr. Roach mentioned that they're having lots of problems with "skimmers" stealing PIN's. Is there some way they can detect this?

Mr. Lewis, Rice Lake Weighing Systems, Inc., mentioned that he liked Measurement Canada's website. When answering similar questions, different pages would appear, based on answers to those questions: http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm00573.html

At the 2011 NTETC Software Sector Meeting, the laboratories were polled to obtain any feedback on the use of the checklist. Maryland attempted to use this checklist a few times. They had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with the Maryland evaluator didn't always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

At the 2013 meeting, it was reported by the labs that attempts to use the current checklist did not meet with many difficulties. The checklists were given to the manufacturers to fill out, and that seemed to work rather well. Minor modifications were made to clarify certain confusing areas or eliminate redundancy (Note the text above includes the updates made in 2013).

Discussion:

The labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

During the discussion, Ed Payne (NTEP lab,MD) said that his impression is that this is at least making some of the manufacturers think about security, which they hadn't necessarily done in the past.

It was indicated that some more or better examples may be helpful to manufacturers, and that more guidance is needed. Clearer instructions could be part of the checklist, or it could be a separate document. The Sector would like additional feedback specifically regarding what portions of it are causing confusion.

Due to proprietary issues, the labs can't simply give us direct feedback from the companies they interact with. Darrell Flocken volunteered to obtain information from the labs, aggregate it, and remove any potential proprietary information issues.

The checklist as updated during the 2014 meeting:

1. Devices with Software

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **The manufacturer should indicate whether it's solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal?** **AND** □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. **With the seal intact, can you change the software?** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **It may also include things like name, part number, CRC, etc.** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.** □ Yes □ No □ N/A

2. Programmable or Loadable Metrologically Significant Software

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with no access to the operating system and/or programs possible for the user. This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.

3.1. Check whether there is a complete set of commands (e.g., function keys or commands via external interfaces) supplied and accompanied by short descriptions. □ Yes □ No □ N/A

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. Complete this section only if you replied No to 1.1.

4.1. Check whether a checksum or equivalent signature is generated over the machine code of the metrologically significant software (program module(s) subject to legal control Weights and Measures jurisdiction and type-specific parameters). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.2. Check whether the metrologically significant software will detect and act upon any unauthorized alteration of the metrologically significant software using simple software tools (e.g., text editor). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

5. Software Interface(s)

5.1. Verify the manufacturer has documented:

5.1.1. **If software separation is employed, t**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.1.2. **For software that can access the operating system or if the program is accessible to the user, t**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.1.3. The functions of the metrologically significant software that can be accessed ~~**via the protective software interface**~~. □ Yes □ No □ N/A

5.1.4. The **metrologically significant** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

Conclusion:

The Sector discussed examples, such as the upgrade of application programs and how these changes would affect audit trails and version numbers. It should be clear that if the upgraded software doesn't affect anything metrologically significant, then it's irrelevant for the purposes of this checklist. On the other hand, if it does affect metrologically significant functions or parameters, it should be tracked and/or identified somehow.

The revised checklist will be reviewed and further edited as required, and the updated version can be sent to the labs.