A reversed approach to security - How to evaluate software and its security (practical session)

Alexandre Dulaunoy

12th February 2005

Contents


```
"You can't evaluate a man by logic alone."
-- McCoy, "I, Mudd", stardate 4513.3
```

1 Introduction to evaluating software

As you seen in the previous courses, software is one of the major sources of risks in various information infrastructure. Existing systems use a large volume of different software components having different kind of interaction in order to communicate. Inter dependencies between components are important, they trust each others when they are operating. If one of the software

component is compromised, you have facing a big issue for the overall infrastructure. When you are building new or updating information systems, you must choose the right software to limit and reduce the associated risks... One important thing to do is to evaluate the security of a software.

1.1 Steps for evaluating software

Evaluating software can be quite tricky as the level of information available can be very different is the software is proprietary or free software. But you must use all the possible ways to evaluate software. There is more than one way to evaluate a software and its security.

1.1.1 Selecting candidates

This can be seem obvious but they are various way to select potential software candidate. You can make an RFP (Request for Proposal) to select software from vendors/integrators, browse the Internet, ask other users of equivalent software.

1.1.2 Functionalities

Major discussions on selecting software are focusing on the functionalities. Often more functionalities that you get means that the complexity of the software is growing. More and more line of codes can be a source of bugs and possible vulnerabilities in the software. You have to carefully make a selection that fit your requirements without exaggerating the functionalities that you would like to have/propose.

1.1.3 Legal perspective

If you are planning to evaluate software, legal aspect can be very important. For example, if you are looking at the license for the software. Is the license permits software evaluation ? reverse engineering ? Do you have access to source code ? can you legally recompile the application ? Are they responsible for security issue ? do they provide some kind of guarantee ?

1.2 Security perspective

A lot of parameters can be considered useful to evaluate the security aspect of a software. This list is not limited but can be a starting point when you are evaluating software for production usage.

1.2.1 Security advisory/disclosure

All software contains bugs and security vulnerabilities (be aware of vendors claiming the opposite), you must ensure that the vendor provides means to

inform you of security issues in their software.Do they have a contact for security issues ? Do they provide patches in a short time period ? Is the vendor providing resolution to security issues ? Do they make documentation about the vulnerabilities in their software ? Do they provide credits to people finding bugs ?

1.2.2 Software Distribution

How is the vendor handling the distribution of their software ? Do they provide cryptographic signature of their files ? Do they rely on untrusted software ? Is update on production system easy or applicable ?

1.2.3 Documentation

Documentation often shows the overall "quality" of a software and can be an excellent factor. Documentation is not only the user documentation of the software but this could also include release notes, security notes or specific comments on how to implement the software.

2 Practical Session

2.1 Selecting an MTA

2.1.1 Qmail

2.1.2 Postfix

2.1.3 Sendmail

3 Bibliography

- How to Evaluate Open Source Software / Free Software (OSS/FS) Programs, David A. Wheeler,`http://www.dwheeler.com/oss_fs_eval.html`
- Snake Oil Warning Signs: Encryption Software to Avoid, Matt Curtin, `http://www.interhack.net/people/cmcurtin/snake-oil-faq.html`