# Malware

It is entered at no cost to a victim's system, but may become expensive to cope with it. What to do about it?

Forbid the load of any malware! Even recognizing malware requires to load it first. But this options means to completely isolate the system, Therefore it is not a really meaningful step in a time, when everybody is asking for connectivity.

Prevent malware from execution! This is my preferred approach. The first case of malware, I handled, happened in 1974 – so we are not talking about an outcome of the internet. Since the hack that left RSA and lots of their clients as victims, I developed attributes for a cyber-save hardware architecture.

This is, how „my“ option looks like:
Let your hardware care for cyber-attacks! It is capable of doing this chore after three weaknesses are taken care of: Distinctively separate between handling your system's software programs and the data, your system is designed to process. This separation has to be applied to

- Interfaces,
- Memory units, and
- Processors.

A simple version of such system is comprised of at least

- Two processors,
- Two sets of interfaces,
- Four separate memory units.

The processors are dedicated to software management and data processing.
The sets of interfaces have to be mutually incompatible, and serve software installation or data exchange exclusively.

The memory units hold different categories of data, and have differing hardware controlled access attributes concerning the mentioned processors.

The correct interconnection of these components leaves injected malware in a memory unit, to which no processor is capable to read instructions from. This renders malware isolated and prevents its execution by any of the processors. A demonstrator is available to show this functional relation.