8.AA Use of unchecked data from an uncontrolled or tainted source [???]

8.AA.1 Description of application vulnerability

This vulnerability covers a general class of behaviours, the identification of which is referred to as 'taint analysis'.

Whenever a program gets data from an external source, there is a possibility that that data may have been tampered with by an attacker, attempting to induce the program into performing some damaging action. Such data is called 'tainted'

The general principle should be that before tainted data is used, it should be checked to ensure that it within acceptable bounds or has an appropriate structure, or otherwise can be accepted as untainted, and so safe to use.

8.AA.2 Cross reference

TBD

8.AA.3 Mechanism of failure

The principle methods of failure are:

- Use of the data in an arithmetic expression, causing the one of the problem described in section 6
- Use of the data in a call to a function that executes a system command
- Use of the data in a call to a function that establishes a communications connection

8.AA.4 Avoiding the vulnerability or mitigating its effects

Different mechanisms of failure require different mitigations, which also may depend on how the tainted data is to be used:

- Tainted data used in an arithmetic expression may need to be tested to ensure that it doesn't cause arithmetic overflow, divide by zero or buffer overflow
- Integer data used to allocate memory or other resources should be checked to ensure that it won't cause resource exhaustion
- Strings passed to system functions should be checked to ensure that they are well formed and have an expected structure (e.g. see [RST] Injection)

This vulnerability is described as 'data from an uncontrolled source', as a distinction may need to be drawn between data from outside the program, but which is still trustworthy, and that that comes from a source that could credibly be modified by an attacker. For example, data read from a file may be regarded as trustworthy (untainted) if the file is inside a firewall, but potentially tainted if it is from a more generally accessible location.