CHAPTER 6

An Example: Integrity and Buffer Overflow

Many think that, of the CIA triad, integrity is the most important for control systems. So I thought too.

I devised in Chapter 5 the notions of

functional integrity: the system is able to carry out its defined function; and

information integrity: the information provided to the system for it to carry out its function is veridical, and when transformed then veridically transformed.

It seems to be obvious that

- you need both concepts and
- they are different concepts.

Avizienis, Laprie, Randell and Landwehr [1, the update of the IFIP WG 10.4 vocabulary] say integrity is the

> *absence of improper system alterations*

IEC TS 62443-1-1:2009 [5] Subclause 3.2.38 says that data integrity is

> *the property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner.*

It also says (subclause 3.2.60) that integrity is the

> *quality of a system reflecting the logical correctness and reliability of the operating system, the logical completeness of the hardware and software implementing the protection mechanisms, and the consistency of the data*

> *structures and occurrence of the stored data*
> *NOTE In a formal security mode, integrity is often interpreted more narrowly to mean protection against unauthorized modification or destruction of information.*

whereas IEC 61508-4:2010 [10] subclause 3.5.4 says that safety integrity is the

> *probability of an E/E/PE safety-related system satisfactorily performing the specified safety functions under all the stated conditions within a stated period of time*

In the 1990's, US CERT used to publish vulnerabilities when they were discovered. Now Mitre CVE does this. In the 1990's, it is commonplace that 80% of the vulnerabilities published by CERT were buffer overflow vulnerabilities.

Two observations.

1. Having a buffer overflow vulnerability in your system does not violate any of the integrity definitions above.
2. Exploiting a buffer overflow vulnerability does not necessarily lead to loss of integrity of the system according to any of the above definitions except IEC 61508-4:2010 subclause 3.5.4, and mine.

Apropos IFIP WG 10.4: exploiting a buffer overflow involves no change in the system code itself.

Apropos IEC TS 62443-1-1:2009 subclause 3.2.28: exploiting a buffer overflow does not necessarily involve any violation of data integrity. A different sequence of commands is executed, starting with the command immediately after the supposed data buffer boundary entered as data and read when the stack is popped.

Apropos IEC TS 63443-1-1:2009 subclause 3.2.60: the "logical correctness of the operating system", the "logical completeness of the hardware and software" and the "consistency of the stored data" are all left unchanged by the exploitation of a buffer overflow.

However, apropos IEC 61508-4:2010 subclause 3.5.4: the flow of control of the program is changed, possibly to foreign code brought within the exploit, or possibly to another part of the installed program inappropriate for the current situation. This can be taken to alter the "probability" that the software will perform the safety function appropriate for the situation the system is now in, if indeed it is appropriate to perform a safety function in this situation

Apropos my definitions: the functional integrity of the program has changed with the exploitation of a buffer overflow. Foreign code, not matching the functional specification, might be executed, or a different part of the software inappropriate for the current situation might be thereby activated.

However, consider the following definition in IEC TS 62443-1-1:2009 subclause 3.2.16:

> *availability (performance)*
> *ability of an item to be in a state to perform a required function under given conditions at a given instant or over a given time interval, assuming that the required external resources are provided*

The availability of the program which the buffer overflow exploits is (probably) affected: the required function to be performed at the given instant at which the program counter jumps to the instruction entered as data and popped from the program stack is (likely) not performed, for that is the purpose of a buffer-overflow exploit.

Summary

Exploitation of a buffer overflow affects the availability but not the (performance) integrity or data integrity of a system according to IEC TS 62443-1-1:2009. It does not affect the integrity of a system according to Avizienis et al. (the IFIP WG 10.4 vocabulary). It directly affects the integrity of a system if the system is executing a safety function, according to IEC 61508:2010. It directly affects the functional integrity of a system according to Chapter 5.

Bibliography

[1] Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C., *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Trans. Dependable and Secure Computing, 1(1), Jan-Mar 2004.

[2] International Electrotechnical Commission, *IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations*, 2nd Edition, 2010.

[3] International Electrotechnical Commission, *IEC TS 62443-1-1:2009 Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models*, 2009.