CHAPTER 8

The CIA Triad

8.1 Motivation

At time of writing (December 2017) a question has recently arisen within the German electrotechnical standardisation community about whether the “CIA triad” of terms from traditional cybersecurity should be taken into the International Electrotechnical Vocabulary, IEV, document IEC 60050 [9]. The triad consists of the terms

- Confidentiality
- Availability
- Integrity

It is worth looking at these terms to see what kinds of concepts they represent. The concepts are in fact very variable.

8.2 Availability

Let us start with a simple case: *availability* is already in the IEV [9, Definition 192.01.23]:

> *192.01.23*
> *availability (of an item)*
>
> *ability to be in a state to perform as required*

> *Note 1 to entry: Availability depends upon the combined characteristics of the reliability (192-01-24), recoverability (192-01-25), and maintainability (192-01-27) of the item, and the maintenance support performance (192-01-29).*

Note 2 to entry: Availability may be quantified using measures defined in Section 192-08, Availability related measures.

We can query whether this is quite the right definition required for, say, the technical considerations in Chapter 4, but let me leave this aside for now. It is clear from this definition that availability is a system property.

8.3 Integrity

Integrity has been considered at length in Chapter 5. In summary, there seem to be many different notions of integrity, used for different purposes. It would surely be reasonable to distinguish them, and put them all in the International Electrotechnical Vocabulary if any one of them is to go in.

I remark again that the various definitions of integrity render it a system property or properties (my proposed definitions), or a sociotechnical property (the IFIP WG 10.4 definition), or not a property but a number [10, Subclause 3.5.4].

8.4 Confidentiality

Confidentiality is interesting. Unlike the other two properties, for which one could claim they are objective properties of a technical system[1], *confidentiality* appears to be a sociotechnical concept, dependent upon not just the objective system properties but upon properties of the environment in which the system operates and the use to which it is put by people. Consider the following example.

In the 1970's and 1980's, quite a lot of work in computer security was put into the concept of *multilevel secure* (MLS). The general idea of an MLS system was that it consisted of files plus an operating system which manipulated those files through

[1] although I did note in Chapter 5 that the IFIP WG10.4 definition of integrity makes it a sociotechnical concept, as do other conceptions.

read and *write* commands. The files were partitioned into an ordinal series of classes[1], as were the users of the system. A user in class C was to be able to read any file of class C or below, and able to write any file of class C or above. This property was known as *write-up, read-down*. It and variants such as the SRI model [8] and the Bell-LaPadula model [1, 2] of multilevel security are discussed in [3, Chapter 5 and Section 20.3.1.1] but details need not concern us here.

Suppose you have a multilevel secure system S which has been, shall we say, formally proven to fulfil a MLS security model such as the SRI model or Bell-LaPadula model. Then, we might think, confidentiality is assured. But in fact confidentiality is dependent upon more than pure system properties, as follows. Suppose we have a set of users, and these users are assigned to the various classes as befits their security clearances. Then the (informal) confidentiality of the system is technically assured, by hypothesis. Now, inadvertently give to all those users the access credentials to the highest class. Then all files become readable and writable by all users. Confidentiality is violated. But it has been violated through an act that has nothing to do with the system as technically defined.

It follows that confidentiality is not a technical system property, but a sociotechnical property. It is a property of the system along with access policies and controls that might not physically have anything to do with the physical computer system at all. It is irreducibly sociotechnical.

8.5 Some Considerations on Property Type

Does it make a difference what type of property is considered? In some cases, it certainly does. Protocols (strict procedures using strictly-defined data types) are devised by computer security engineers for accomplishing necessary security tasks, for example authenticating a system user in order to grant himher an appropriate system role, or authenticating a message in order to bequeath full trust in its contents. Users of personal computers might be familiar with the PGP suite, originally developed by Philip Zimmermann [11], software which "plugs in" to certain mail clients (for

[1] An ordinal series is a well-founded linear order. Here it is just a finite linear order – the classes are finite, since the computer systems being considered were finite, and any finite linear order is well-founded.

example the Enigmail “plug-in” for Mozilla Thunderbird [7]) and makes use of public infrastructure (certification authorities and key repositories) in order to authenticate messages from its users, and to maintain the confidentiality of messages between sender and receiver, avoiding a “Man in the Middle” (MITM) being able to read a message in transit. It uses a series of protocols devised by Whitfield Diffie and Martin Hellman which use a Public Key Infrastructure (PKI) [6] (also briefly recounted in [3, Section 9.3.1]). Diffie and Hellman won the ACM Turing Award for their work.

Cryptographic protocols such as these consist of sequences of actions which are intended to achieve a specified series of properties. So-called action logics are formal logics (consisting of a syntax and inference rules) in which actions can be formulated, as well as properties, and in which it is possible formally to prove (derive from assumptions using the inference rules) that certain rigorously-formulated properties are achieved (or not) by certain sequences of actions. Action logics have been at the centre of at least five Turing Awards (to Robert Floyd, C. A. R. Hoare, Robin Milner, Amir Pnueli, and Leslie Lamport). The so-called BAN logic (named after the initials of its three proponents) has been used for many decades for the formal verification of security protocols [4, 5]. Formal verification works as follows: the protocols are formulated in the syntax of BAN logic, and the inference rules are used to show that, if the protocol executes from a given system state (the “preconditions”) formulated in BAN-logic syntax, the desired resulting system state (the “postconditions”) is attained. Formal verification of this sort is essentially a mathematical process, and is not necessarily easy. However, such methods constitute the only way known so far in which desired system properties can be confidently shown without exception to be attained by executing a protocol from a given system state. Computer security protocols generally need to be exceptionless, and for this reason computer security has been at the forefront of formal and mathematical verification methods for four decades.

Proving protocols correct using a logic such as BAN logic is only part of the problem of verifying protocols, of course. One must also show the assumptions are valid. For example, many protocols use a *nonce*, a unique message transmitted once and that cannot be replayed. Typical practical (presumed) nonces are the Transaction Authentification Number (TAN), used in many on-line banking protocols. A BAN-logic inference assumes a nonce is indeed a nonce, but of course whether the TAN really does satisfy the properties required of a nonce is a separate matter which must be investigated. Many so-called “replay” attacks succeed because supposed-nonces used in an implemented protocol are guessable from previous transactions. So there are many steps to verifying a practical procedure, not only verifying whether the protocol succeeds, but also verifying that it has been appropriately implemented to fulfil the assumptions of its verification.

Formal verification, as well as validation of assumptions, is in theory applicable when the properties to be shown are system properties. With sociotechnical properties, a difficulty arises in verifying the results of human actions, for human agents do not necessarily perform perfectly according to the requirements of a protocol. Thus, for example, no matter how technically accomplished the protocol, a confidential-message recipient can destroy confidentiality if heshe reads out the contents loudly in full hearing of the rest of the office staff. Such possibilities make it almost impossible to devise rigorous methods to validate sociotechnical properties. The best that can be done is to verify purely technical subprotocols, and to devise pre- and postconditions for human actions, which can then be written explicitly into human procedures and requirements for human participation in the system.

For validation and verification purposes, then, properties formulated as pure system properties are very helpful. Sociotechnical properties are harder to validate because of the vicissitudes of human agency.

8.6 Summary

I have considered the CIA triad and remarked that they appear to be different kinds of property: a system property, or a sociotechnical property of system + human/operator/authority environment. Availability appears to be a technical system property, integrity can be a pure system property but there are also appropriate sociotechnical definitions, and confidentiality appears to be irreducibly sociotechnical. Validating and verifying protocols for pure system properties is, although by no means easy, very much more amenable to success than validation of sociotechnical properties.

Bibliography

[1] D. Bell and L. LaPadula, *Secure Computer Systems: Mathematical Foundations*, Technical Report MTR-2547, Vol. 1, March 1973, MITRE Corporation, Bedford, MA.

[2] D. Bell and L. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation*, Technical Report MTR-2997, Rev. 1, March 1975, MITRE Corporation, Bedford, MA.

[3] Matt Bishop, *Computer Security*, Addison-Wesley, 2003.

[4] Michael Burroughs, Martín Abadi and Roger Needham, *A Logic of Authentication*, DEC SRC Research Report 39, 1989, revised 1990. Available as http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-39.pdf, accessed 2017-12-06.

[5] Michael Burroughs, Martín Abadi and Roger Needham, *A Logic of Authentication*, ACM Transactions on Computer Systems 8(1):18-36, February 1990. Available as http://www.cs.cmu.edu/ dga/15-712/F07/papers/Burrows90.pdf.

[6] Whitfield Diffie and Martin Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory 22(6):644-654, November 1976. Available as https://www-ee.stanford.edu/ hellman/publications/24.pdf, accessed 2017-12-06.

[7] (No author) *Enigmail*, available from https://www.enigmail.net/index.php/en/

[8] Richard J. Feiertag, Karl N. Levitt, and Lawrence Robinson, *Proving Multilevel Security of a System Design*, 6th ACM Symposium on Operating System Principles, pp57-65, ACM Press, December 1977.

[9] International Electrotechnical Commission, IEC 60050, *International Electrotechnical Vocabulary*, on-line at www.electropedia.org .

[10] International Electrotechnical Commission, IEC 61508-4:2010, Functional safety

of electric/electronic/programmable electronic safety-related systems: Part 4 – Definitions and abbreviations, IEC, 2010.

[11] Philip Zimmermann, *WWW Home Page*, at https://philzimmermann.com/EN/background/index.html, no date.