Security implications of structure*

Jeremy Bryans and Budi Arief

University of Newcastle upon Tyne

1 Introduction

Computer security is an important issue in determining the dependability of computer systems. It becomes even more crucial when we talk about *computer-based systems (CBS)*, where we take into consideration the roles played by the human actors (or human components) involved in the system.

In this chapter, we begin to explore the security of complex CBS (sometimes called *socio-technical systems*). We do this by putting forward a common structuring abstraction for technical systems (that of *component-based systems*), then extending this abstraction to computer-based systems, in order to take into account the socio-technical structure of the system.

2 Security basics

Security is about protecting assets. In a computer system, these assets are things like information, processing power or hardware. In a computer-based system, this list must be extended to include more ethereal notions, such as trust. Traditionally, the ways in which the assets of any system may be compromised are frequently grouped into three aspects: *confidentiality*, *integrity* and *availability*.

In order to avoid security compromises, certain security measures are usually applied to systems. It is virtually impossible to have a "totally secure system" without sacrificing the usability of that system. Security measures are therefore employed to provide an acceptable level of protection, based on the purpose of the system and the perceived security threats that this system is going to face.

* This document is an extract of a full chapter from *Structure for Dependability: Computer-Based Systems from an Interdisciplinary Perspective*, pp. 217-227, Springer, 2005.

Security can be seen as an "all or nothing" property. Attackers must be kept from having any impact on the system whatsoever. In most cases, however, security protection is composed of several structured layers, protecting different levels of the system. This is comparable to James Reason's Swiss Cheese model [9], where each layer provides protection from certain types of attacks but has weaknesses (represented as holes) against other types (see Fig. 1). Security breaches happen when holes on these layers are aligned, allowing attackers to penetrate every layer of protection.

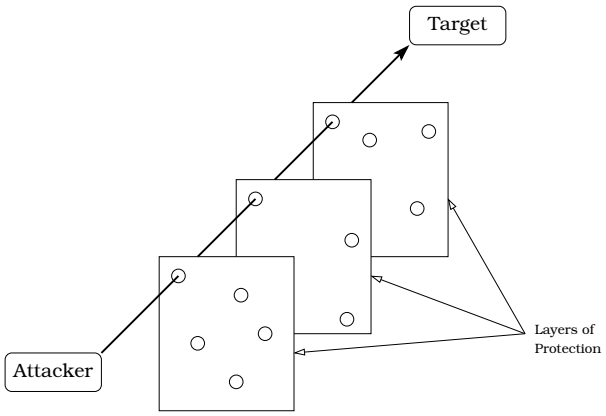

Fig. 1. Description of a successful attack within Reason's Swiss Cheese model

Fault-tolerance is about error detection, containment and recovery. Error containment can be seen in structural terms, with potential sources of error within a system being contained by the (hardware or software) structures. Error containment is about not letting errors out. The same applies in reverse for security: it is about not letting malicious "errors" in.

These two approaches (all-or-nothing and fault-tolerance) lead to the development of quite different systems. Thinking about the two approaches in structural terms can help to understand the resultant systems. Any physical thing protected by "all-or-nothing" security will have big, strong, obvious defences. A fault-tolerance mindset leads to "absorbent" security. A physical thing protected by absorbent security will be surrounded by a number of layers of security. These will be designed to protect against different types of threats, so that an enemy finding it easy to break one layer will be likely to have difficulty in breaking another. Rather than defending the outer boundary of the system at all costs, an attack is "absorbed".

3 Structure and socio-technical security of CBS

Computer systems are built up of *components*, which may themselves be systems. In many cases, these components are *commercial-off-the-shelf (COTS)* products with their own identifiable function. They also have *boundaries*, and some means of interaction with surrounding components or systems [8].

When these components are combined into one system, the boundary of the resultant system encompasses all the boundaries of its components, which could lead to some interesting security issues. Security is about protecting boundaries around these components, both from having private information flowing from "inside" to "outside" and from attacker (could be outsider or insider) gaining full access (read, write, modify, delete) to the information.

The interconnected nature of these components makes it more difficult to secure the whole system. This is because it opens up the boundary to another level, where a breach in one component might lead to further breach in other components or even the whole system. When we consider socio-technical systems, issues on boundaries still exist, but we now have to consider a very different form of component: people. This immediately leads to two new types of interface or boundary to consider: the *person–machine* boundary and the *person–person* boundary.

When we consider people as components, the assumption that they have the same functionality or purpose breaks down. People have more than one purpose. Furthermore, people are able to change their behaviour (and indeed purpose) according to the situation in a way that programmed components cannot.

When we dig a little deeper, a more difficult problem arises. This has been hinted at earlier, and is the question of motive. A legitimate user within a socio-technical system may have many complex and even contradictory motives, and even a single motive may result in opposite behaviours.

People also tend to be creative: they find work-arounds to certain restrictions that were enforced to improve security. In most cases, improving security means more effort or less flexibility to people. They do not always like the idea, and if they can find a way to bypass this and make their life easier, they will do so.

People can potentially improve system security. They are able to observe anomalous behaviours, which might be otherwise left un-noticed by a machine. On the down side, humans could act as the weakest link when it comes to security. Human nature and tendency – for example their willingness to help others or their predictable behaviour under pressure – are often exploited by attackers through social engineering [5, 6].

4 Conclusion

Human involvement in any system is unavoidable, and will critically influence the structure and security of the system, making it unpredictable and therefore

hard to study. To understand how these socio-technical systems behave, we need to better understand the behaviour of people. This will lead to a better design of security measures in term of usability and effectiveness. As a result, the risk of human components bypassing or rendering the security measures useless through their careless actions could be reduced.

Another way to improve the security of computer-based systems is by making the human components aware of the importance of sound security practices and the havoc that security breaches could bring. It is very common – if not mandatory – for new employees to undergo safety training or induction. This could be extended to include *security induction*, where new employees are made aware of the organisation's security policies.

References

1. Anne Adams and Martina Angela Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.
2. Anne Adams, Martina Angela Sasse, and Peter Lunt. Making passwords secure and usable. In *Proceedings of HCI'97 People and Computers XII*, pages 1–19. Springer, 1997.
3. Rebecca Gurley Bace. *Intrusion Detection*. Macmillan Technical Publishing, 2000.
4. Denis Besnard and Budi Arief. Computer security impaired by legitimate users. *Computers & Security*, 23(3):253–264, May 2004.
5. Brian Hatch, James Lee, and George Kurtz. *Hacking Linux Exposed: Linux Security Secrets & Solutions*. Osborne/McGraw-Hill, 2001.
6. Kevin Mitnick and William Simon. *The Art of Deception: Controlling the Human Element of Security*. Wiley, 2002.
7. David Powell and Robert Stroud (Editors). Conceptual model and architecture of MAFTIA. Technical Report MAFTIA Deliverable D21, Project IST-1999-11583, January 2003.
8. Brian Randell. Dependability, structure and infrastructure. Technical Report CS-TR 877, University of Newcastle, Nov 2004.
9. James Reason. *Human Error*. Cambridge University Press, 1990.
10. Robert Reeder and Roy Maxion. Error analysis of a security-oriented user interface. Technical Report 872, Newcastle University Computing Science, 2004.
11. Martina Angela Sasse, Sasha Brostoff, and Dirk Weirich. Transforming the weakest link - a human computer interaction approach to usable effective security. *BT Technological Journal*, 19(3):122–131, 2001.
12. Herbert A. Simon. *Models of Man*. Wiley, New York, 1957.
13. Clifford Stoll. *The Cuckoo's Egg*. Doubleday, 1989.