19

How to trust systems

A. Jøsang, F. Van Laenen[@], S. J. Knapskog and Joos Vandewalle[§]
Department of Telematics
The Norwegian University of Science and Technology
N-7034 Trondheim, Norway.

Abstract

The owners and users of distributed systems need to trust components of the system from a security point of view. In this paper we investigate the possible methods for establishing trust in the security features of an IT product or system.

Keywords

Trust, security, trusted systems, assurance, security evaluation

1 INTRODUCTION

Security evaluation is an example of a well established method for establishing trust in implemented system components. The method is based on a set of evaluation criteria like e.g. TCSEC (USDoD, 1985), ITSEC (EC, 1992), CC (ISO, 1996) or similar, and accredited evaluation laboratories which perform the evaluation under supervision of a national authority. A successful evaluation leads to the determination of an assurance level which shall reflect to which degree the TOE or system component can be trusted. It must be recognised that evaluation assurance does not represent the users own trust in the actual system component, but rather a recommendation from a supposedly trusted authority. The evaluation assurance is thus only one of several factors supporting the user's trust in the product.

In this paper we will discuss methods to establish trust in systems. Our goal is to

[@] A part of this research was done while the author was student at KUL, Belgium, and visiting NTNU, Trondheim.
[§] Department of Electronics, Katholieke Universiteit Leuven, Belgium

determine the most relevant factors which can contribute to the establishment of trust, and to find out how these factors can be combined to produce an overall perception of trust. Further, we will look at the ISO evaluation criteria (ISO, 1996) and suggest new elements not yet included in the present version. There is a slight distinction to be made between simply determining the most correct level of trust and wanting to increase it, and we will briefly describe how these aspects are related.

When studying IT security one is forced to take the human psychology into concideration. The traditional purpose of IT security is to prevent breaches of confidentiality, integrity and availability by implementing threat countermeasures expressed as technical aspects of the IT-system. The purpose of the countermeasures is to generate trust which is a human phenomenon. Trust would allow users to use a system in ways which they otherwise would avoid, so that in practice the system becomes more valuable and a more powerful tool.

2 DEFINITION OF TRUST RELATIVE TO IT SECURITY

Trust is a very general concept which can be used in almost any context. For the purpose of IT security, it is desirable to give trust a more specific meaning which may be useful in formal modelling of security.

The main rationale behind IT security is that some agents in a given situation may attempt an attack on the system, and security is supposed to prevent such attacks to succeed. The existence of malicious behaviour in general is not only the reason to have IT security, but in fact also as a necessary condition for trust (Jøsang, 1996).

Malice is here defined as a combination of dishonest and crooked behaviour, which implies both lying and breaking the rules (law, contract etc.). Similarly we define benevolence as a combination of honest and straight behaviour, which implies telling the truth[1] and respecting the rules.

A human would be trusted if believed to be benevolent, and distrusted if believed to be malicious. It may be true that there is a finite number of factors which determine whether a human will behave in a benevolent or malicious way, but it is extremely hard to determine all these factors, and therefore the behaviour of a human is hard to predict. For all practical purposes, whatever the underlying mechanism may be, we will call the human mechanism which decides between benevolent and malicious behaviour *the free will*, and we designate agents possessing this type of free will as *passionate*. We define trust in a passionate agent as *the belief that it will behave without malicious intent.*

Algorithms, protocols, software, hardware can hardly be characterised as passionate or having a free will, but they can still be trusted. We will call this type of agent *rational* as opposed to passionate. Because a rational entity has no free will, it is not expected to be malicious or benevolent. What exactly is being trusted is that its behaviour can be completely and uniquely described, and that it will resist any attempt

[1] In the sense: doing what you say you will do

of malicious manipulation by an malicious agent. Thus there is a third party involved in addition to the trusting party and the trusted rational entity, namely the possibly malicious agent. We therefore define trust in a rational entity as *the belief that it will resist attacks from malicious agents.* In the rest of the paper we mainly focus on the second type of trust because it applies to systems.

Even with the narrowing of the meaning of trust in systems through the definitions above, trust still has a broad meaning. In order to give it an even more precise meaning we use *trust purpose* which expresses what exactly the trusted system is being trusted for. The security of a system may consist of several components, such as integrity and confidentiality on a general level, and key generation and certificate verification on a more detailed level. The trust purpose can reflect these components.

3 ADEQUATE TRUST OR MAXIMUM TRUST

Trust should be based on objective evidence. Irrational trust is not based on objective evidence, but for instance on faith or a vague and fuzzy feeling which can not be rationally justified, and sometimes this trust can persist despite evidence of the contrary. This type of trust may be valuable in other situations, but can be risky for IT security. Even if trust ultimately is subjective, it will be an advantage if as many users as possible have a common trust based on the same objective evidence. One would normally expect the manufacturers to trust the systems they produce in the same way. However, there are indications that this is not always the rule.

A manufacturer which knows about vulnerabilities in the system it produces may not want to reveal this information for several reasons. A more or less legitimate reason can be that by publishing this information, the potential attackers may learn as much about the system's vulnerabilities as the legitimate users. After all, if nobody knows about a vulnerability, it will not be exploited. However, this strategy can be very dangerous, and if an attacker can discover and exploit a weakness which already was known by the manufacturer, the consequences can be serious.

Recent successful attacks on smart cards have shown that the cards can be very vulnerable. (Boneh *et al*, 1996), (Andersen and Kuhn, 1996). The manufacturers must have known about some of these vulnerabilities because of the extreme secrecy with which they have surrounded the manufacturing process. The security of smart cards depends to a large extent on the confidentiality of the hardware design and the supposed difficulty of physical inspection or manipulation. This was made very clear when an attempt to perform a security evaluation of a smart card failed due to the manufacturers' "fear" of having their products evaluated (EC, 1994), (Jøsang, 1995).

Humans are often irrational, and so is trust. This may be a minor problem for the user himself, because at least to some degree it is his own choice. On the other hand, this situation can be dangerous for the manufacturers because they may no longer be able to repair damaged trust with objective evidence. One good example of this phenomenon is the fuss surrounding the FDIV flaw in Intel's Pentium microprocessor

during 1994. This flaw would only cause extremely sparse errors but the general public's trust in the processor and also in Intel as a manufacturer dropped dramatically. Only when Intel announced a no-questions-asked return policy did the company manage to regain the public's trust, and interestingly, even though everybody could have the old Pentium version replaced, many chose not to do so, which indicates that simply knowing that they could have their processor replaced was enough to restore the trust in it.

System manufacturers naturally wish to generate the highest possible public trust in their products whereas the users real trust may not always be based on objective evidence. For the general benefit of both the users and the manufacturers, it would be desirable if manufacturers always revealed, if necessary in a controlled way, all security relevant evidence which the users objectively should know, and users should try to base their trust mainly on objective evidence.

4 THE KNOWLEDGE PARADIGM OF TRUST

Many professions provide services in an environment of relative ignorance, and Smithson (Smithson, 1988) describes civil engineering and justice as two such examples. Civil engineering is very explicit when it comes to error in the estimations of load and structural strength. The practice of justice on the other hand has extensive relevancy criteria for the evidence presented in court. These two approaches reflect the different types of evidence which these two professions work with. IT security professionals also have to deal with relative ignorance, and for this purpose it will be useful to determine the nature of the evidence at hand and then determine the best practical way to use it. The two next sections give a general description of the types of evidence relevant for trust.

4.1 Direct evidence

A user of a system can never obtain perfect knowledge of the system he uses, nor of the external or internal threats, and he is therefore unable to exactly determine the system's security. By gathering as much knowledge as he can about the system he will get an idea or a belief about the security, or in other words, he will gain a certain trust in the system. In this perspective, security can be understood as an idealistic goal for the system designers, whereas trust represents the users' actual imperfect knowledge about how successful the designers have been in reaching their idealistic goal. This situation of less than complete knowledge will always persist, and the problem we are facing is how to handle it.

A good assessment of a system's security in its environment will be based on evidence from many different sources. A distinction can be made between direct and indirect evidence. By direct evidence we mean the evidence resulting from an

investigation of the system through for instance a security evaluation, and through direct experience with the system. By indirect evidence we mean mediated evidence such as for instance recommendations and advice. The direct evidence can be grouped into system evidence, environment evidence and security incident
evidence. This grouping fits well with a typical model of risk analysis like for instance the model employed by CRAMM[2] (CCTA, 1991). This is illustrated in Figure 1.

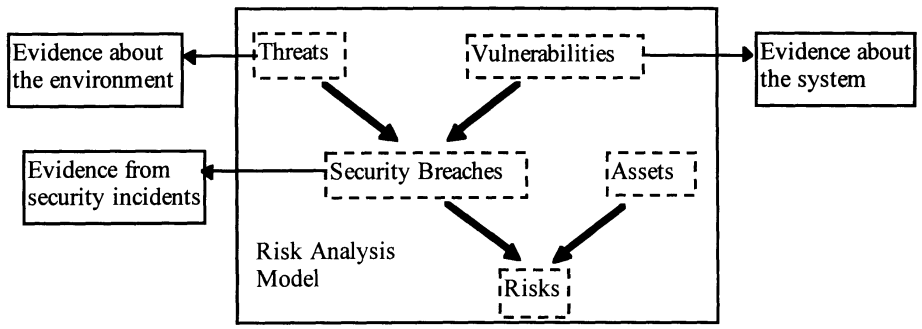

Figure 1 Direct security evidence.

According to the risk analysis model, the analysis and combination of evidence about the threats (environment) and about the vulnerabilities (system) generates likelihoods of security breaches, but only the operation of the system can show whether any of these breaches will materialise as security incidents. A materialised security breach is the worst that can happen, and it is considered better for an IT system to have too high security than to suffer from breaches. However, one of the purposes of the risk analysis is to be able to balance the cost of security with the losses due to security breaches. A direct consequence of this is that "sufficient" trust is the actual goal for a practical system.

4.2 Indirect evidence

We intuitively tend to trust somebody if he is trusted by others whom we already know and trust. This principle is extensively used, also for security services, e.g. when an entity must rely on trusted third parties, key distribution centres or certificate issuers to establish trust in IT- systems.

When a user is unable to collect direct evidence to form his trust in a system, either because he does not have access to it or because he does not possess the expertise to evaluate it, he depends on indirect evidence. A mediating agent can recommend in a backward direction that the following entity in the chain can be trusted, and so forth

[2] CCTA Risk Analysis and Management Methodology)

until the intended target entity is reached. This kind of recommendation can be formal and explicit like for instance the issuing of an evaluation certificate, or implicit like when a user simply trust a system because other users whom he trusts are using the same system and seem to trust it.

Figure 2 is an illustration of one possible way for a user to view a system of which he is unable to obtain direct evidence. The evidence he has direct access to may come from other users, a certification authority, consultants, experts etc. Their evidence may in turn come from the evaluators or the manufacturers who have access to the direct evidence.

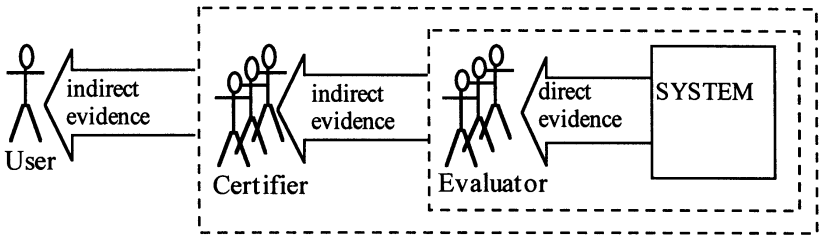

Figure 2 Indirect security evidence.

Figure 2 is not supposed to be a general model, but simply to illustrate the flow of evidence in the case of security evaluation. As a matter of fact, the user will usually have access to various types of both direct and indirect evidence. He may for instance obtain direct evidence from security incidents during operation and from assessing the threats in the system's environment. In addition to the security evaluation report, he may receive indirect evidence through security advisory reports. The difference between direct and indirect evidence is that the first is obtained by direct observation of the system and its environment, and the latter is mediated through passionate agents which can be humans, human organisations or the public in general. The user must therefore take into account potential malicious or irrational behaviour on the side of the mediating agents.

Strictly speaking, it is not correct to consider trust itself as flowing between the mediators. A recommendation from a trusted party to trust a particular system is in reality a piece of evidence which the user himself has to consider together with other relevant pieces of evidence, resulting in a subjective belief which constitutes his trust in the system.

5 HOW TO PRACTICALLY DETERMINE TRUST IN SYSTEMS

We will here try to describe a suitable framework for using the types of security relevant evidence described in section 4 in order to determine trust in systems. Like in civil engineering and in justice, where structures are modified and verdicts are changed if

they are proven wrong by new evidence, the methodology for determining trust should be dynamic and take into consideration new evidence such as security incidents and new threats.

5.1 A measurement unit for trust.

Risk analysis methodologies usually operate with a quantifiable measure of risk. In case of CRAMM (see section 4.1) this is obtained by assigning a value to each threat and vulnerability and from this determine some sort of probability for each type of security breach which can occur. The probability value is finally multiplied with an asset value to produce a measure of risk. In practice the risk levels obtained after the first round of analysis are often outside any reasonable measure. For that reason, a second round of analysis is often necessary where the threat and vulnerability values are adjusted to give a more reasonable result. The reason for the abnormal results from the first round of analysis is that it is in fact difficult to determine a level of threat or vulnerability by a single value. Risk analysis methodologies were partly developed because it seemed a logical thing to do, partly also because a measure of risk is a managerial requirement. One purpose of risk analysis is to give the management an impression of quantifiable knowledge.

The idea of formally determining trust in computer systems was first developed within the military community and later spread to the commercial sector. TCSEC (USDoD, 1985) which were the first security evaluation criteria, defined a grading of 7 assurance levels, and when other criteria followed, they also tended to use 7 assurance levels as an attempt to maintain compatibility with TCSEC. When considering the large amount and the complex nature of the evidence which an evaluator must consider, there is nothing which logically indicates that this can be translated into a discrete value such as an assurance level. Again, it must be recognised that the idea of determining an assurance level for systems is as much a managerial requirement as it is a natural characteristic of a system. For this reason it should be recognised that the definition of 7 assurance levels is very ad hoc. On the other hand, a more rich and thereby complex classification of assurance could easily become useless because users would not be able to understand it. That is in reality the dilemma we are facing: The simpler the classification of trust becomes, the less it is able to reflect the diverse aspects of security, but on the other hand, the richer the classification, the less useful it becomes.

Evaluation assurance can not be directly translated into trust and only provides indirect evidence of a system's security. We argued that a single assurance level, attractive as it may be, in fact may blur the total picture of a system's security more than it clarifies it. The picture does not get any simpler when in addition other types of evidence have to be considered, thus indicating that trust can not be reasonably measured as a single value. Approximations may be used if a particular situation requires it and if users find it convenient, and for that purpose the problem is to find the most appropriate way of doing it.

5.2 A framework for determining trust based on objective evidence

Four sources of objective evidence which a user should consider for determining the security of a system are listed below:

1. **Evaluation assurance.** Whenever this type of evidence is available it will reflect a thorough analysis of the system. This can be very valuable, but can also generate unjustified trust if other types of evidence are ignored.
2. **Assessment of environment and threats.** The user or owner of a system should always make an assessment of the security threats present in the system environment. The system security should be reconsidered each time new threats are discovered,
3. **Security advisory reports.** This type of evidence gives up-to-date information about newly discovered security vulnerabilities, and should be closely monitored, even more so because also attackers usually have access to these reports.
4. **Security incidents.** A long period without security incidents does not necessarily lead to increased trust, but a security breach should directly influence the trust, overruling all other evidence including evaluation assurance.

The four types of evidence are quite different, and it may not be evident how they can be combined. We will suggest a method or framework which we believe suitable for the operators of a system. It assumes trust to be dynamic in function of new evidence, and takes the evaluation assurance level as a basis for the highest reachable trust level under normal operation conditions. Degradation of trust can for instance be caused by new evidence such as advisory reports on vulnerabilities. The elimination of the vulnerabilities may re-establish the trust. A distinction can be made between trust in the correctness of a system and trust in its effectiveness in the actual environment, in order to reflect the corresponding two types of evaluation assurance. Evidence of a fault in the system will for instance lead to degraded trust in the correctness, whereas evidence of a new threat may lead to degraded trust in the effectiveness. On a general level there should be defined grades of degradation from the normal security level, and on a more detailed scale, how much degradation each particular evidence type should cause. Upgrading of the security level may be done by eliminating or countering the factors which caused the degradation. By careful collection and consideration of evidence in this way, a system operator can keep a good overview of the system's security level which should be directly reflected in the operators' trust in the system. If their actual subjective trust deviates from the assessed security level, it can only mean that the operators consider factors or evidence not included in the assessment method.

Any mediating agent must be considered to be passionate. The case is different from trusting the security of a rational system which means trusting its ability to resist

malicious attacks. The possibility that the evaluation facility itself is subject to malicious attacks in order to undermine the ongoing security evaluations is remote.

6 ASSURANCE COMPONENTS IN THE COMMON CRITERIA

6.1 Pedigree and credentials to ease evaluations

The concept of pedigree in IT Security (Van Laenen, 1995) can be useful in the sense that both direct and indirect experience with the security of the products from a given manufacturer may influence the final trust that the user will have in the security of future products from the same manufacturer. In the commercial market oriented world, this kind of trust building is probably even more common than the more formal product evaluation activities. Still, there may be a role to play for an evaluation facility, but not as evaluator of the security features of every single product, but as a process evaluator, checking on the manufacturer's security credentials in analogy to a quality assurance (QA) process. Several aspects may be included in a manufacturer's security credentials, e.g. the company as a whole, the development processes used within the company, what QA mechanisms are implemented, and the credentials of the individuals making up the company. Some of these aspects may be objective, and readily evaluateable, while others may be more subjective, quite subtle distinctions between trustworthy and non-trustworthy companies or organisations.

An experimental Assurance Class, called Credentials and Pedigree (ACP) is proposed. It contains (for now) only one Assurance Family, named Process Quality Label (ACP_PQL). The assurance components and their elements are described along the guidelines given in (ISO, 1996). The Assurance Family consists of 5 assurance components, one of which is covering the fact that the developer has earned an ISO 9000 certificate, and the other four covering the developer's consistent use of the Capability Maturity Model, in four hierarchically ordered levels.
Table 1 shows what an assurance component built on ISO 9000 may look like. The description does not pretend to be all-encompassing.

Assurance element ACP_PQL.1.1D states that the developer has to have an ISO 9000 certificate before the start of the development of the product (TOE).

Table 1 **Assurance Component ISO 9000 Certificate.**

ACP_PQL.1 ISO 9000 Certificate

Objectives This component makes sure that the plants of the manufacturer have a process with the proper quality, according to the ISO 9000 family, thus inferring that they are capable of producing a secure TOE with the specified level of assurance.

Threats Not all the plants of the manufacturer have the capability to produce a TOE with a good quality assurance.

Dependencies The ISO 9000 family

Developer Action Elements

ACP_PQL.1.1D	**The developer will have an ISO 9000 certificate before the start of the development of the TOE**
ACP_PQL.1.2D	**The developer will register all the plants that are involved in the development of the TOE**
ACP_PQL.1.3D	**All the subcontractors involved in the development of the TOE will have an ISO 9000 certificate**
ACP_PQL.1.4D	**The plants of the developer and all the subcontractors will be certified against all relevant parts of the ISO 9000 family**

Content and Presentation of Evidence Elements

ACP_PQL.1.1C	**The evidence elements will contain the ISO 9000 certificate for all involved plants**

Evaluator Action Elements

ACP_PQL.1.1E	**The evaluator will check that all the involved plants of the developer were certified before the start of the development of the TOE**
ACP_PQL.1.2E	**The developer will check that all the involved plants are certified to all relevant parts of the ISO 9000 family**
ACP_PQL.1.3E	**The evaluator will check that all subcontractors are certified to all relevant parts of the ISO 9000 family**

6.2 Flaw Remediation

Trust will be influenced by the presence of known flaws in the IT product you buy. The quick and appropriate remediation of found flaws therefore is of major importance. (On the other hand, flaw remediation can be made redundant if a perfect flaw avoidance scheme is invented.)

Not-the-First Version

It is quite common to have more trust in an existing system that already proved itself, than in a new system. The reasoning behind this is simple: newly added functionality will probably also mean newly added flaws.

Nevertheless, it is healthy not to automaticly regard the next version to be better simply because it is the next version, since the danger of adding a new flaw while remediating an old flaw is real. Also, if the next versions follow to quickly, this may indicate that the first version of the product was badly designed. Nevertheless, a next version will in general be better because the known flaws are removed from the product and the product has already proven itself through its use.

Time to Remediation

In some areas of IT, only known flaws are dangerous. As an example, a covert channel which cannot be exploited because it isn't discovered yet by any user, will not cause much harm. But as soon as one user knows how to handle the covert channel, it does become dangerous, and actually, a fortiori if the covert channel isn't known to the victim. If the covert channel is known to the public, the users can take precautions against the exploitation of it.

So, for some flaws, the time to remediation is most important. There are two ways to deal with this time to remediation: you can impose a maximum or a mean time to the IT producer. The maximum time is interesting to the user of the product, but may be very restrictive to the IT producer, and in the case of very severe flaws, may be seen as unreasonable. On the other hand, mean time to remediation may be to weak for the user.

6.3 Legal Aspects

Legal aspects can convert flaws into money loss for the IT producer, and therefore do have influence on the trust one can put in an IT product.

Liability and Guarantee

In most countries, it is possible to sue the producer if a product harms you in any way. Usually, the company will have to restore the damage, or pay for it. Damage may often not be restorable, like for example when data is lost or made public. If the IT manufacturer can be held liable for any damage the IT product does to the user, this will increase the users trust in the IT product.

If one tries to apply this in practice, several problems occur. First of all, this kind of assurance lies outside the traditional area of IT. Also, the laws may differ in different countries. This complicates the problem, and may restrict the use of this trust component.

Internal Liability

It may be a problem that the person responsible for a certain development project is transferred soon after the project is finished, so that in practice, no one can be held liable for any occurring flaws. It would be better to keep the person in charge responsible for the project, not only as long as its is being developed, but also after it is launched onto the market.

7 CONCLUSION

Trust expresses a belief and will always be a subjective notion. It is therefore meaningless to speak about solely objective trust. However it is possible to require trust to be based on objective and carefully collected evidence. We have described what

it means to trust a system and suggested how trust should be determined and what it should be based on. Formal security evaluation may play an important role for determining trust in systems, and we have given some comments on the Common Criteria which possibly constitute the latest and most modern criteria for that purpose.

8 REFERENCES

Anderson, Ross and Kuhn, Markus (1996). *Tamper Resistance - a Cautionary Note. http://www.ft.uni-erlangen.de/~mskuhn/tamper.html*

Boneh, D., DeMillo, R. A. and Lipton, R. J. (1996). *On the Importance of Checking Computations*. BELLCORE. *http://www.bellcore.com/SMART/index.html*

CCTA (1991). *CRAMM User's Guide (Version 2.0)*. The UK Central Computer and Telecommunications Agency.

EC (1992). *Information Technology Security Evaluation Criteria*. The European Commission.

EC (1994). INFOSEC investigation S2108: *Security Project for Evaluating* Smart Cards. The European Commission.

Van Laenen, F. (1995). *Pedigree and Credentials, Remediation and Legal Aspects to Gain Assurance in IT Products and Systems*. Master Thesis, KUL.

ISO (1996). *Evaluation Criteria for IT Security* (documents N 1401, 1402, 1403, 1404). ISO/IEC JTC 1/SC 27.

Jøsang, A. (1995). *The difficulty of standardizing smart card security evaluation.* Computer Standards & Interfaces 17(1995), pages 333-341.

Jøsang, A. (1996). *The right type of trust for distributed systems*. In Proceedings of the New Security Paradigms Workshop 96. ACM.

Paulk, M. C. (1994). *A Comparison of ISO 9001 and the Capability Maturity Model for Software*. Technical report, Software Engineering Institute, CMU/SEI-94-TR-12.

Smithson, M. (1988). *Ignorance and Uncertainty.* Springer Verlag.

Swaelens, G.J. (1992). *ISO 9000 Quality Standards in 24 Questions*. ISO 9000 News, 1, January 1992. Interview with MR J. E. Ware, Managing Director of BSI Quality Assurance and Chairman of ISO/CASCO

US DoD (1985). *Trusted Computer System Evaluation Criteria*. US Department of Defence.