Using Trust on the Internet

Stjepan Groš, Marin Golub, Vlado Glavinić
Faculty of Electrical and Computing Engineering
University of Zagreb
Unska bb, 10000 Zagreb, Croatia
E-Mail: {stjepan.gros, marin.golub, vlado.glavinic}@fer.hr

Abstract— **Trust is a concept we heavily use in everyday life because it allows us to cope with the complexity of interactions with other people. This concept is also implicitly used throughout the Internet, both in the infrastructure and in the end-user services. Since security problems on the Internet are becoming more and more serious, and thus present a threat to the further growth of the Internet, there is ongoing research that tries to turn trust into the first class component of the Internet architecture. In this paper we review concept of a trust and we give short review of the security problems on the Internet. We argue that those problems are fundamental in nature and, using current mechanisms, will never be solved appropriately. As a potential solution to this problem we discuss architecture modifications, related protocols and procedures that would allow trust to be an explicit part of the Internet. The goal of this paper is to form a base for further research on the trust issues of the Internet.**

Index Terms— **security, trust, reliability, internet, trustworthiness**

I. Introduction

Security breaches on the Internet are constantly being reported by different media. No matter if it's some kind of a DoS attack, or theft of privacy information, it is clear that they want go away any time soon. This is, at a first glance, very strange as we have very good cryptographic algorithms without known flaws. Furthermore, there are good protocols for the use on the Internet that also don't have known vulnerabilities. Finally, there are best practices and recommendations that are known to be very effective in increasing security of devices and systems.

To get an idea on how to solve this problem, it is very interesting to look how people cope with the uncertainty in the *real world*. As it turns out, we heavily rely on *trust* and trusting other people to behave in a good way. Of course, there are also other means of maintaining this state, primary by punishing those that don't behave appropriately and also, by punishing non-punishers. The key point in this interpersonal relationships is that trust is dynamic and subjective, i.e. not predetermined and fixed in a time.

It seems that trust, if introduced into the Internet in some way, could be very important mechanism that would improve security and made Internet more usable on a longer time scale. This is, actually, not something new, because if we look more carefully, we'll conclude that trust is in some form already used on the Internet. Furthermore, there were research projects on trust issues[1], though majority concentrated on e-Commerce and similar applications.

That security and trust are becoming more serious as the time goes on, even threating Internet's growth, become so obvious that Internet Society[2] issued a call for the participation in a discussion about trust and the future of the Internet[3]. The meeting was held on October 9th, 2007, and conclusion was that trust, along with identity on the Internet was important enough to be declared major strategic initiative.

So, we may ask ourselves, what is *trust*? This term is used in wide variety different contexts, and thus, has many meanings making it very hard to define. One of our goals is to review definitions of trust and to either select most appropriate one, or to generate a new one.

The ultimate goal of this paper is to lay down a foundation and goals that will serve as some kind of a guide lance into making Internet a more secure place by making trust a first class concept. Furthermore, we envision a framework on the Internet, consisting of architecture associated protocols and appropriate API, like GSSAPI[4], that will allow different applications to leverage benefits of trust.

This paper is structured as follows. In Section II we describe problems with current security. Then, in Section III we define and describe the concept of trust. In Section IV we show how trust could be used in the Internet to make it more secure and trustworthy place. The paper finishes with conclusions and future work in Section VI.

II. Current security problems

In [5] the *security* is defined as a system condition in which system resources are free from unauthorized access and from unauthorized or accidental change, destruction and loss. Obviously, the security is a certain state of the system which has to be continuously maintained and verified. The way security is implemented is to enumerate threats, evaluate risks and apply protective measures and procedures. Different, well known, mechanisms are used to accomplish security, e.g. physical protection, cryptographic functions, access controls. The maintenance is accomplished by constantly monitoring that implemented measures and procedures are enforced. There is multitude of different techniques that allow security to be established and maintained, but, there are no mechanisms to measure effectiveness of the applied techniques.

The majority of the current security problems are consequences of operational errors, or malicious attempts, caused by humans. For example, distributed denial of service attacks are mounted from multiple machines possibly with spoofed IP addresses. Attacks with the spoofed IP addresses can be stopped by applying proper filters on network edges [6]. If addresses are not spoofed, then there is high probability that attacks come from compromised hosts that are again consequence of inadequate maintenance of hosts. As another example, spam related issues can also be traced to unsecured mail servers or carelessness of hosting companies and their service providers. We can conclude that humans are very important parameter in security equation, so economic and psychology factors have to be consulted when developing or applying security mechanisms.

Additional problem faced by the Internet users and operators is that there is no assurance of security measures applied by different entities. In other words, when communication takes place between two end points, those end points neither have information on security of all the elements taking part in the communication, nor they have any way of changing communication path based on security requirements.

Even worse, reports sent to abuse addresses of different Internet providers usually have no effects or they are of a limited scope. Furthermore, there are Internet providers driven only by profit who don't do enough in order to protect the rest of the Internet from their malicious customers. There are already monitoring activities that try to asses from where attacks originate, e.g. [7]. The idea is that if the rest of the Internet reacts on such ISPs by lowering their ability to connect they would start to take care of malicious traffic. Unfortunately, this mechanism is not used much in practice.

It is clear that measures to increase security decrease the freedom of participants. In other words, security is about restricting what people can do, and usually, people don't want to be restricted. Thus, people are inclined to use those measures and, generally, resist them. *Perfect* security (in a given context) is cumbersome, and cumbersome technology is deployed and operated incorrectly and insecurely [8].

Furthermore, the entity that had security breach doesn't always bear the economic cost of a breach and thus it is not motivated to invest more into the security.

Measures to protect security of the system are based on some assumptions. In other words, we have to assume that something is secure per se, and thus we can rely on it when building security of the whole system. For example, if we are designing security for Internet access in some company, we have to assume that the equipment belonging to the company is secure. Otherwise, there is no way to design working system. Obviously, in this particular case, with the spread of mallware, these assumptions are very weak. Another, very good example, are laptops. They present a significant challenge to security of an organization since they are not physically protected while the user takes them home or on a trip.

Assumptions made in those cases are sometimes very simplified as there is no way to cope with them. This additionally makes security on the Internet a problem.

III. The concept of trust

We heavily use the word *trust* in everyday life and, intuitively, it's clear what is meant by it. The primary purpose of the trust is to allow us to simplify the complexity of the environment in which we live.

Research of the trust was first done in sociology, psychology and philosophy. When trust was transfered into computer science, it was primary as a mean to solve some problems with mobile and/or intelligent agents. Gradually, this term started to be used for other purposes, e.g. it's now regularly used in e-Commerce, different social networks, peer-to-peer networks and different kind of forums.

There are several definitions of trust, two of which we'll mention here. The first definition is given by Diego Gambetta, and it's frequently cited in the literature:

> Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent

> or group of agents will perform a particular action, both before he can monitor such action (or independently or his capacity ever to be able to monitor it) and in a context in which it affects his own action. When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him. Correspondingly when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so.

This definition specifically states that trust is *subjective*. This makes it very hard be used in in computing environment and thus, a mean to make it more formal is necessary.

Another definition of trust is given by IETF[5]:

1) (I) /information system/ A feeling of certainty (sometimes based on inconclusive evidence) either (a) that the system will not fail or (b) that the system meets its specifications (i.e., the system does what it claims to do and does not perform unwanted functions). (See: trust level, trusted system, trustworthy system. Compare: assurance.)
2) (I) /PKI/ A relationship between a certificate user and a CA in which the user acts according to the assumption that the CA creates only valid digital certificates.

IETF differentiates between two uses of term trust and so divides definition into two parts, depending in which context the term is used. But both contexts basically say the same, with PKI being more specialized.

A. Trust related terms

Trust, though the fundamental, is not the only term that appears in papers. There are also few others that we list in this subsection.

Trust is a relationship between *truster* and *trustee*, i.e. truster believes trustee will do specified action. Note that in this paper we sometimes use the term *end point* instead of trustee or truster, which can be either human or some machine.

When we say for someone or something that it's *trustworthy* it means that we trust in this person or thing.

The very important concept related to a trust is the *reputation*. Reputation can be positive or negative and the more positive it is, we trust more to the given entity, likewise, the more negative it is the more untrustworthy is the entity.

It's simply impossible to have enough information to be able to asses everyone's trustworthiness! In that cases, we have to rely on third party to help us, so we introduce *recommendations*.

B. Determining trust

What both definitions we cited basically say is that trustor believes that the trustee we'll perform it's task with certain probability. It is also interesting to note that probability that system will function as intended is not depended only on it's security, but also on availability, reliability, etc. More generally, we can say that trust reflects the system's dependability.

But, for the moment, let us concentrate only on security. If the system is secure than we have high expectation that it will function as intended, and thus, we have very high trust in it. Conversely, if the system is insecure, then we don't trust it very much at all. Thus, we can conclude that *trust is a actually a measure of system's security*.

The main problem with determining how much to trust to a system is obviously determining how much secure it is! Further complication is that security of the system can not be analyzed in the isolation as environment influences it's security, and more importantly, it is the environment that makes identical system more or less secure. In other words, if the system is placed in secure environment, then it might be trustful. But, if the same system is placed in very hostile environment, then it's trust level might be negatively influenced.

In general, when any system is characterized, it's done by measurements and analysis of the system itself. The first problem with measurements is which parameters should be measured and in which units. After doing measurements we use given data to either explain current system's behavior or to predict it's future behavior. The following are two ways of determining trust level of a trustee:

1) *Direct experience*. In this method we constantly monitor and measure different parameters of the trustee. Based on the results of this process we determine it's trustworthiness.
2) *Recommendation*. It's not possible to anticipate all the systems we are going to communicate with and it's also not possible to measure and monitor all of them as this would create scalability problems. So, we can ask third party about it's experience with

the trustee. In this case we have to additionally take into the account the trustworthiness of the recommender.

IV. Trust on the Internet

The original Internet was designed with the implicit assumption that it will be used by trustworthy users and, as a consequence of this assumption, no security measures were designed in as the Internet was deemed trustworthy. While Internet grew different security measures were engineered in with the goal to make Internet the same level of trustworthiness as in the initial state. The main problem with this efforts is that it takes much resources to keep Internet's level of trust as it was in the beginning. This is further complicated by the fact that not everyone wants this level of security and, in the same time, has enough resources to achieve it. This clearly translates into conclusion that Internet doesn't represent a single trust domain. This is exemplified by different devices that divide network into the trusted zone and the rest of the Internet, e.g. firewalls.

To see how trust is already deployed on the Internet we can take as an example a process of buying something over the Internet. The first question we are confronted with is whether we are buying from the right shop. Then, do we pay to the right person and will this person correctly handle our credit card data? Also, will this person keep our private information confidential or will it sell them to the first buyer. When we bought the item, will it arrive?

But, the situation is more complex than previously described as we didn't take into account all the devices and organizations that our information goes through. There is implicit trust in them. Not only in those that are directly involved in data transfer over the network, but also indirectly, like our trust in CA provider that entities for which certificates it generates are properly validated.

The key problems that we analyze with respect to trust on the Internet in the following subsections are:

- Requirements
- Determining trust
- Architecture
- Communicating trust
- Trust decisions and use of the trust in the applications and protocols

A. Requirements

Before explaining how we envision making trust an explicit security mechanism on the Internet it is useful to enumerate requirements that we wish to satisfy. Our utmost goal is to allow Internet end points to use trust as a basis for their security decisions, like authentication mechanisms and strengths, and authorization. In effect, each environment can specify minimum acceptable trust level and in that way it can control how much risk it is prepared to take when communicating over the Internet.

The *first requirement* we set up is that the solution should be incrementally deployable and non-intrusive. In other words, we can not expect that the whole Internet will embrace trust technologies and thus we have to be prepared that there will be islands deploying our solution. If the solution is good it's probable that in a course of time places where trust is not used will form islands, i.e. minority.

Trust decision is a local matter and thus, we are not going to propose languages, in any form, that will be used to describe how decision is made. On the other hand, it is very probable that trust decisions will have to be made based upon the following parameters:

- Truster identity
- Trustee identity
- Action to be performed
- The level of trust on trustee

Our *second requirement* is that all four parameters have to be defined in such a way that it's possible by diverse systems to exchange information and to unambiguously base their decisions on those parameters.

The *third requirement* is to allow interoperable exchange of trust information between different entities on the Internet.

Finally, our *fourth requirement* is related to privacy issues. Our system should be designed in such way that privacy of participating parties can be preserved.

B. Determining trust level

If we are going to measure system's security then the result of the measurement process has to be some quantifiable value that can be compared. The obtained values make it possible to draw conclusions about system's security and allow systems to be compared to each other or to some predetermined requirements. Different measurement values can be used for this purpose, i.e. continuous with different ranges, discrete or symbolic. It's very probable that different applications would made better use of different values. What we believe is that certain number of different measurement systems should be standardized. Additionally, a system to translate between those systems could also be prescribed.

After determining scales for measurement, the next step is to determine how exactly to assign values to different levels of the trustworthiness of the trustee.

The first method is by *direct experience*, i.e. we constantly monitor and measure different parameters of the trustee via the Internet. For example, we can regularly scan open ports of the trustee to see which ones are open. Then, based on the expected open ports, as well as those unexpectedly open, we associate certain trust value with this parameter. Furthermore, we can monitor changes of values and based on the expected and measured rate of change give additional value to the trust value. What can be measured and what values could be assigned to what is measured is a topic of a further research.

The second method is *recommendation*. It's obviously not possible to anticipate all the systems we are going to communicate with in order to track their trust level. There are two reasons for this, the first one is that it's hard to predict with whom we are going to communicate and thus we could be forced to monitor larger set of possible peers than it's actually needed. This creates scalability problems. So, we anticipate that third parties will communicate their experience and trust levels about trustee. In this case we have to additionally take into the account the trustworthiness of the recommender.

C. Architecture

Obviously, the Internet is a very diverse and complex system consisting of many different subsystems. To view and assess the trustworthiness of the Internet as a single entity is very unrealistic and thus, we have to break it down into smaller, more manageable, units. This also helps us to fulfill the first requirement, i.e. possibility of gradual deployment into the Internet. So, we propose the Internet to be subdivided into trust zones. These zones can be equal to existing autonomous systems, DNS zones or anything else. There is possibility that zones would have to be subdivided into smaller parts, potentially not visible outside of the given trust zone. For example, some ISP might wish to groups it's private users into one zone, while all of it's business customers into other zone.

Each zone has several components as shown in the Figure 1.

The main component of the architecture is a *trust server*. In each trust zone there is at least one trust server, but, for the security, scalability and similar purposes there could be any number of trust servers distributed on strategic points throughout the zone. The purpose of the trust server is to:

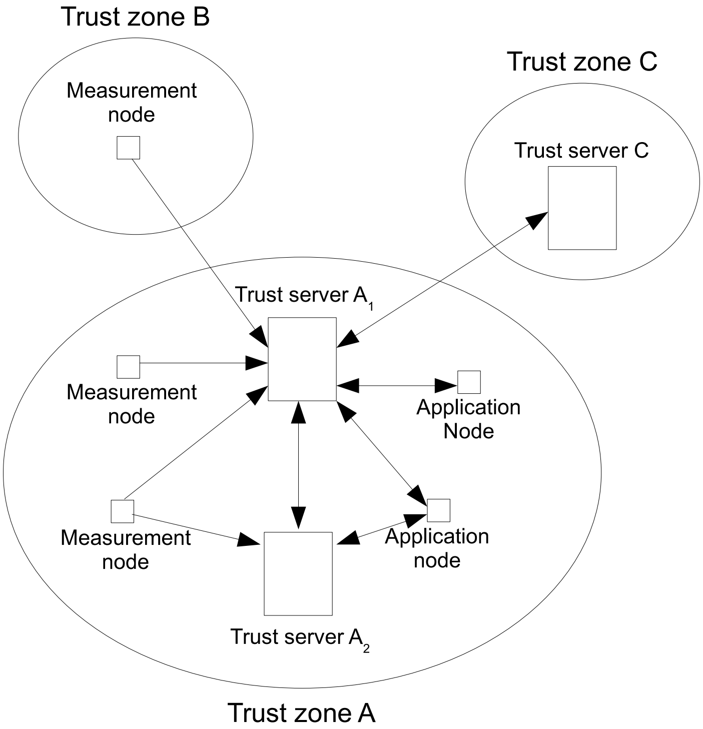

Fig. 1. Major components in the trust zone

1) Enumerate object for which it maintains trust levels
2) Collect and maintain information about trust levels of it's objects
3) Collect evidence of positive and negative behavior of it's objects
4) Disseminate trust information to interested parties

For a purpose of determining trust level of objects monitored by trust server there are measurement nodes spread through the trust zone, but also there could be measurement nodes in another zones of the interest. The purpose of the measurement nodes is to determine certain parameters of interest for the trust server. The reason for decoupling measurement from collection is to obtain different and independent values of certain parameters. The functionality of measurement nodes can be performed by specialized nodes, or the hosts and routers can have appropriate support and send updates to trust server. A single measurement node can send measured parameters to multiple trust servers.

Trust servers communicate using *recommendation protocol*. In the Figure 1 trust servers A_1 and A_2 in the same zone exchange information as well as trust servers A_1 and C in zones A and C. The protocol for the communication is the same, but the trust level that

A_1 has in A_2 will be higher that in C.

Finally, there are *application nodes*, or *trust clients* which are users of trust information. They connect to trust servers using *recommendation protocol* and obtain relevant data they use in their trust decisions. Finally, as trust clients are applications acting on behalf of the users and the applications use appropriate API to obtain all the necessary data. Trust clients can also submit their experience with the trust objects they worked with. If it's possible, then the experience also contains *evidence*. For example, evidence might be signed excerpt from the logs generated by the applications.

It's obvious that different trust zones will care for different objects and their trustworthiness, e.g. trust server can handle trustworthiness of the ISP's clients, or servers hosted by some Web hosting company. Thus we propose an interoperable specification language for description of objects.

D. Protocols

Protocols allow communication of trust parameters (objects, measurement, recommendations and evidences) between components of the trust architecture. There are plenty of choices of transport protocols and encodings. At the transport layer there is a choice of TCP or SCTP. We don't believe that connectionless protocols will be suitable for our purposes. Then, there is possibility of using a variant of HTTP protocol to transport application messages. But it seems that the variant of the protocol developed for exchange of Intrusion Detection messages[9] might be more suitable for our purposes. This is further in line with the selection of XML[10][11] for data encoding.

V. Making trust decisions

Trust decisions are explicitly out of scope of this work. The main premise for such decision is that there is no universally accepted language to specify trust policies, and very probably, there wont be such that would be suitable for all the applications. Thus we leave to implementors to define and implement languages that they find the most suitable for their needs.

What would benefit applications is standardized API for accessing trust subsystem that would allow applications to use trust for their purposes. This API could be modeled after GSSAPI[4].

VI. Conclusions

In this paper we gave overview of current security problems on the Internet, defined trust, and proposed how to use trust on the Internet in order to solve it's security problems.

Furthermore, we proposed architecture for the trust system that could be used on the Internet to implement *trust modulated transparency*[1] concept.

There is a lot of a work to be done before this system could be used on the Internet. First, and foremost, trust measures have to be specified. Then, protocols have to be defined. In order to be possible to assess trustworthiness of different objects, for each potential object parameters that can be measured have to be defined and a system should be specified on how those parameters affect trust measures. The proposed systems has to be verified against real-world experience. Finally, the security of whole system has to be analyzed in order to prevent successful attacks against the trust infrastructure.

References

[1] D. Clark, K. Sollins, J. Wroclawski, D. Katabi, J. Kulik, X. Yang, R. Braden, T. Faber, A. Falk, V. Pingali, M. Handley, and N. Chiappa, "New arch: Future generation internet architecture," http://www.isi.edu/newarch/iDOCS/final.finalreport.pdf, Dec. 2003.

[2] "Internet society (isoc) web page," Jan. 2008. [Online]. Available: http://www.isoc.org

[3] L. Lynch, "Call for participation: Trust and the future of the internet," Aug. 2007. [Online]. Available: http://www.isoc.org/isoc/general/trustees/headlines/20070809.shtml

[4] J. Linn, "Generic Security Service Application Program Interface Version 2, Update 1," RFC 2743 (Proposed Standard), Jan. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2743.txt

[5] R. Shirey, "Internet Security Glossary, Version 2," RFC 4949 (Informational), Aug. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4949.txt

[6] P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827 (Best Current Practice), May 2000, updated by RFC 3704. [Online]. Available: http://www.ietf.org/rfc/rfc2827.txt

[7] "Dshield; cooperative network security community," Jan. 2008. [Online]. Available: http://www.dshield.org

[8] R. Sandhu, "Good-enough security: Toward a pragmatic business-driven discipline," *Internet Computing*, vol. 7/1, p. 3, Jan. 2003.

[9] B. Feinstein and G. Matthews, "The Intrusion Detection Exchange Protocol (IDXP)," RFC 4767 (Experimental), Mar. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4767.txt

[10] P. Luthi, "RTP Payload Format for ITU-T Recommendation G.722.1," RFC 3047 (Proposed Standard), Jan. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3047.txt

[11] S. Hollenbeck, M. Rose, and L. Masinter, "Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols," RFC 3470 (Best Current Practice), Jan. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3470.txt