User Trust Levels and Their Impact on System Security and Usability

Henryk Krawczyk[1] and Paweł Lubomski[2]

[1] Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Gdańsk, Poland
hkrawk@eti.pg.gda.pl
[2] IT Services Centre, Gdańsk University of Technology, Gdańsk, Poland
lubomski@pg.gda.pl

Abstract. A multilateral trust between a user and a system is considered. First of all we concentrate on user trust levels associated with the context-oriented CoRBAC model. Consequently, there were computed user profiles on the basis of its implementation in the information processing system "My GUT". Furthermore, analysis of these profiles and the impact of user trust levels on system security and usability have been discussed.

Keywords: context, system security, CoRBAC model, trust levels, user profile, system usability

1 Introduction

Let us consider two situations with respect to multilateral trust, i.e. from the system to the user and vice versa (see Fig. 1). From the system side the trust is estimated on the basis of the observed user behavior. Generally typical user behavior satisfies some rules, which mostly do not change over time. If the system detects a change, it is likely that somebody pretends to be a valid user. In such a situation the system trust to the user is reduced. In this case some extra actions must be taken by the system to verify the user. For example, the system may ask the user for some more detailed information or fire some stronger security mechanisms. This increases the chance that the system becomes more secure. However, frequent user verification takes more time and therefore the system usability and user satisfaction decreases. It is important to find some balance between the increase of system security, and system usability.

Similarly, let us consider the parallel situation when the user has some doubts concerning trust in the system's security. Then the user needs some tests to verify the system. These tests should be periodic and should be executed by an external auditor. In addition, the system must be reliable and produce the correct results of computations. High usability of the system also increases the user trust in the system.

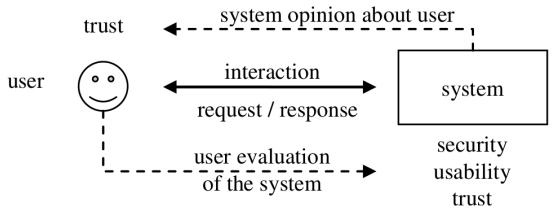

Fig. 1. The considered situation in the paper.

In the paper we consider a system with three options of system security:

1. a security scenario $secs_1$ with basic security mechanisms
2. a security scenario $secs_2$ with strong security mechanisms
3. a security scenario $secs_3$ with context-oriented security CoRBAC mechanisms [9]

In other words, we consider three variants of security scenario for the system, while functionality is the same. The first one (labeled $secs_1$) is a traditional security scenario with no context-oriented mechanisms. Let us assume there is only a basic security mechanism (e.g. password check on access). The second one (labeled $secs_2$) is the opposite case – it has strong security mechanisms, so that the security level of the system is much higher. As an example of these mechanisms, there could be a CAPTCHA mechanism [2], two-factor authentication or cryptographic signing, but the mechanisms are still not context-oriented. Let us assume that $secs_2$ has cryptographic signing, because it is the strongest security mechanism. The third one (labeled $secs_3$) is a system scenario with context-oriented security mechanisms implemented. These mechanisms could be similar to the second ones, but they are fired only in some situations, determined from the context.

We introduce the security levels of these variants of the system security – labeled respectively $secs_1$, $secs_2$, $secs_3$. According to [10] the following inequalities apply:

$$sec_1 < sec_3 < sec_2 \tag{1}$$

As an example we consider the IT system implemented in our university called "My GUT". Using the information about behaviour of over 46 thousand "My GUT" system users we introduced a four-level user trust model, and on the basis of that we tested both sides of the trust relationship. We assess that the increase of security in the third type of system does not decrease significantly the system usability. This property is the most important in the practical use of the system with CoRBAC mechanisms.

2 User and System Trust

It is a known fact that a human being is the weakest point in the security of a system [15, 1]. On the other hand, the higher the level of user confidence in

the system, the more secure the system is [13]. Pahnila, Siponen and Mahmood argue that sanctions for users not complying with the security policy have a insignificant impact on the level of system security. A more appropriate direction is to build user awareness, and using the security mechanisms which are possibly the least intrusive for them. But they should be noticeable to users so that users, could believe that the system is secure [7, 3].

As mentioned earlier, there are two aspects of the mutual trust between the user and the system. One describes how a human being trusts the system as its user, the other one shows how the system trusts its user.

The first aspect of trust is how the system trusts the user. On the basis of the level of this trust, the system enforces weaker or stronger security mechanisms in CoRBAC security model [10]. The model assumes that the more trustworthy the user is the more s/he can do in the system. As a result, the level of system security is growing up, because only highly trusted users can interact with the system [6]. This fact also affects the user's opinion of the system security, and finally the user trust in the system.

The second aspect of trust is how the user trusts the system. A user opinion is based on the following aspects [5]:

- the functionality level of the system,
- the quality level of the system (meant as error prone level),
- the responsiveness of the system interface (especially during high load),
- the quality and reliability of the data being processed and stored in the system,
- human imagination about the system security [3].

There is one more parameter which we want to take into account for the considered system: usability (labeled, for introduced security scenarios, respectively $usab_1$, $usab_2$, $usab_3$). It can be defined as a human being's convenience in using a system. As is mentioned above, this parameter has also a big impact on the user trust in the system.

Implemented security mechanisms impact on usability in the following ways:

- the more, or the stronger security mechanisms involved, the lower the usability, as a user has to do some extra activities (e.g. typing a password multiple times),
- the more security mechanisms, the lower usability, because the responsiveness of the system decreases.

On the basis of these assumptions the following inequalities are true for the considered system:

$$usab_1 > usab_3 > usab_2 \tag{2}$$

Introducing the context-oriented CoRBAC model, and user trust levels, we claim that:

$$usab_1 \approx usab_3 \tag{3}$$

3 Context-oriented Security Model

Users interact with the system through an interface located in an environment. This environment is described by many parameters which define the context of this activity [4, 11, 12]. The context may be considered as a certain period of time in which the user action is performed, a physical or logical localization of the user, a general state of the system, a relation between the user and the utilized data, a user interaction scenario, the kind of the device being used, etc. [14]. The ability to integrate contextual information makes the role-based security model flexible.

3.1 Context Definition

According to [9] context (C) can be defined as a set of different context parameters (CP):

$$c_i \in C{=}CP_1{\times}CP_2{\times}CP_3{\times}\cdots{\times}CP_z \tag{4}$$

Context parameter (CP_j)i s a finite, discrete set of possible values of the parameter:

$$CP_j{=}\left\{cp_{j1},\, cp_{j2},\, cp_{j3},\, \ldots,\, cp_{jk_j}\right\},\;\; k_j{=}|CP_j|\,,\;\; j{=}\,1,\;2,\;\ldots,\,z \tag{5}$$

There are some assumptions:

1. The set of context parameter CP_j contains all possible values of this parameter.
2. Sets of context parameters CP_j can be of different size:
$$|CP_a| = k_a, \quad |CP_b| = k_b, \quad k_a = k_b \vee k_a \neq k_b \tag{6}$$
3. The values of the context parameter (the elements of CP_j) are separable - in any given situation only one element of the set describes the context parameter:
$$\forall cp_{ja} \in CP_j, \quad \forall cp_{jb} \in CP_j, \quad cp_{ja} \neq cp_{jb} \tag{7}$$

Some context parameters have a continuous character (e.g. time). During the context analysis they are clustered into certain groups, e.g. days of the week. The same situation applies to context parameters with discrete but numerous values, e.g. set of IP addresses. They are clustered into subsets of a specific netmask.

An example of the context parameters considered in this paper and their possible values is as follows:

- CP_1 – logical localization of user – a set of the following elements: cp_{11} = internal network, cp_{12} = campus network, cp_{13} = external network (Internet),
- CP_2 – time of user's activity in the system – a set of elements: cp_{21} = weekday, cp_{22} = Saturday, cp_{23} = Sunday.

Then $z = 2$, $|CP_1|{=}\,3$, $|CP_2|{=}\,3$, $|C|{=}\,9$, e.g. $c_1 = (cp_{11},\, cp_{21})$ = (internal network, weekday).

3.2 Idea of CoRBAC Model

The CoRBAC model has been presented and discussed at an international congress of IT security experts and practitioners [10, 9]. It is based on wide context acquisition and its analysis. To do that, we arbitrarily adopted 4 trust levels of the system towards the user: TL1, TL2, TL3, TL4, where the highest level is TL4 (user is the most trustworthy) and the lowest level is TL1 (user is the least trustworthy). The user trust level is determined on the basis of a user profile built on the history of the user interaction in the system, pointing to a current context value at each access request. This process is precisely explained in the next section. The initial trust level is TL4. If we want stronger security at the beginning of the user interaction in the system there should be assumed a lower initial trust level (e.g. TL1).

Figure 3.2 presents a trust level calculation mechanism of each user access request verification. Let us assume that for user u the current context accompanying the x-th access request is $c_x(u)$.

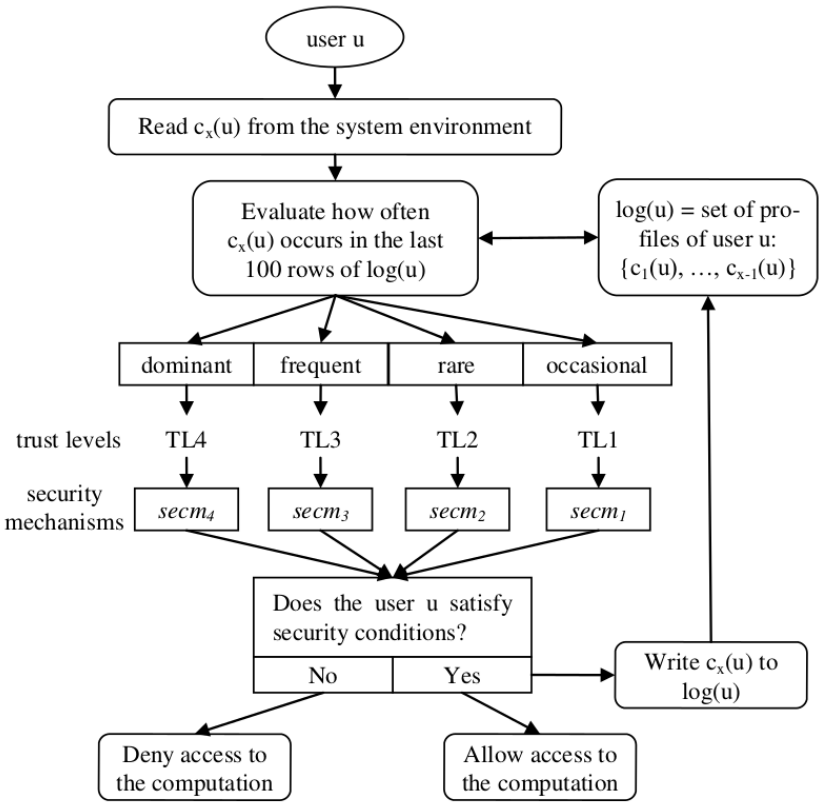

Fig. 2. Algorithm of security scenario $secs_3$ for x-th access request of user u

During the user interaction in the system, before each user access request in the system, the context is analyzed and the appropriate trust level is chosen. Next, the adequate security mechanism is fired.

The security mechanisms (*secm*) are some extra actions which the user is forced to do to prove that s/he is authorized. For TL4 it could be a password-based authentication. For TL3 it could be authentication based on both a password and CAPTCHA [2]. Two-factor authentication could be a security mechanism being forced on TL2. Computation on TL1 could be guarded by cryptographic signing of the user login request.

We can conclude that the security scenario $secs_1$ is similar to security scenario $secs_3$ working on trust level TL4. The second considered security scenario $secs_2$ with very strong mechanisms is similar to security scenario $secs_3$ working on trust level TL1. Security scenario $secs_3$ dynamically chooses the trust level and the corresponding security mechanism on the basis of a current context value $c_x(u)$ and user profile log(u). It is done for each user (u) and for each user access request separately.

4 User Profiles

We have analyzed the activity of over 46 000 users of "My GUT" system [8] within a nearly 4-year period. We took into account mainly security logs, whose first goal is to allow forensic analysis in case of any possible security incidents.

The analysis is based on the statistically unusual behavior of a user, taking into account context consisting of two context parameters. We have taken two assumptions – two limits of the frequency metric: 1% and 5% of the user activity. Below the first mentioned limit an activity is very suspicious, so the trust level is the lowest (TL1) and, according to the CoRBAC model, a user has to prove strongly that s/he is the right one. These mechanisms are the most inconvenient to the user. Between this and the next limit there is the second trust level (TL2). The second limit is assumed also arbitrarily, but we also consider one more limit – at the level of 10%. We assume that activity performed more often than one out of ten activities is not suspicious enough. This way, we assume three limits of frequency metric (defined later), and on the basis of them, four trust levels (TL1 to TL4).

We have adopted one simplification: we have split user interaction into smaller scenarios corresponding to the user sessions from login to logout request. We have analyzed only login access requests.

The analysis consists of two aspects:

- how many users are under suspicion,
- how many times the trust level is reduced – how often the security mechanisms need to take action.

Each profile is computed in the following way:

1. For each user u, for each of their x-th access requests we assign a frequency metric describing the percentage of the previous access requests taking place

with given context value c_x (given type of localization and given type of day of week); we analyze only the 100 previous access requests of user u:

$$freq\left(access\_request_{u,x,c_x}\right) = \frac{\sum_{s=w}^{x-1} access\_request_{u,s,c_x}}{\sum_{s=w}^{x-1} access\_request_{u,s}} \cdot 100\%, \quad (8)$$

where:
$access\_request_{u,x,c_x}$ – x-th access request of user u, taking place with accompanying context value c_x,
$w = \begin{cases} x - 100, & x > 100 \\ 1, & x \leq 100 \end{cases}$.

2. If the frequency is lower than assumed limits, then the access request is marked as suspicious, and the trust level is reduced. In consequence this triggers stronger security mechanisms.
3. The first ten access requests of a user in the system do not change the trust level.

To clarify that, let us analyze the following example. Table 1 shows the log of access requests of one user pointing to their current context. There is assumed only one limit of mentioned-earlier frequency metric: "below 20%". This limit is assumed only for explanation of user profile building process. The profile is consequently updated after each positive access request.

Table 1. Example sequence of access requests for user u for x numbered from 1 to 13

x	$\mathbf{c}_x$	**Current frequency metric (%)**	**Below the 20% limit**
1	c_1	0	Yes (omitted)
2	c_1	100	No
3	c_1	100	No
4	c_2	0	Yes (omitted)
5	c_1	75	No
6	c_1	80	No
7	c_1	83,3	No
8	c_1	85,7	No
9	c_1	87,5	No
10	c_1	88,9	No
11	c_2	10	Yes
12	c_2	18,2	Yes
13	c_2	25	No

Let us analyze a few rows of Table 1. In the first row there is the first access request in the system. There were no requests of the considered user earlier, so the request in context c_1 amounts to 0% of previous requests. It exceeds the "below 20%" limit, but this value is omitted because of the rule that the first ten access requests do not change the trust level. Row 3 shows: access request in context c_1 in two access requests previous amounts to 100% and this does

not exceed the limit. In row 4 there is the first access request from the context c_2. The access request in context c_2 amounts to 0% of three previous access requests. That exceeds the limit but it is also omitted because of the rule of the first ten. Row 11 shows the first interesting case. It is the second time that the context c_2 occured. Access request in this context value amounts to 10% of ten previous access requests. That of course exceeds the limit and it is taken into account (triggers the stronger security mechanism). In row 13 the context c_2 occured for the fourth time. Access request in this context amounts to 25% of twelve previous access requests. That does not exceed the limit.

Figure 4 presents a graphical representation of the profile of user u, on x-th access request, taken in context value c_5. This profile is computed later than the requests presented in Table 1.

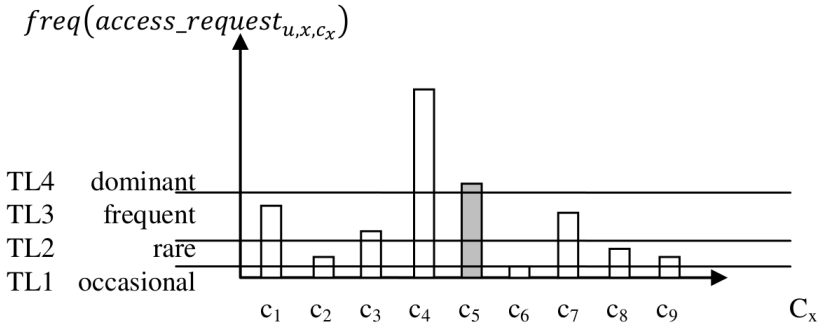

Fig. 3. Example user profile of x-th access request of user u, with accompanying current context $c_x = c_5$

We analyzed each user interaction log separately. For each user access request we built a user profile and assigned a corresponding trust level. Next we added the numbers of access requests taken in each trust level.

The result of this analysis is presented in Table 2. The numbers of requests belonging to intervals (m, n] when users were classified from m to n times in trust level TL_i. For each trust level there is a number and percentage of users that were classified in this trust level.

To summarize this analysis, about 70% of users were under suspicion. But it happened mostly from one to five times that the user was forced to take extra security actions, so these stronger security mechanisms could be slightly inconvenient to them. We have also analyzed the extreme cases (where required number of security actions amounted over 100). Those were the people who worked for a very long time in the administration section (which uses an internal network) and then changed their workplaces and positions, which affected the change of the network they used.

Table 2. Number of different users who were classified from m to n times in trust level TL_i

Number of requests of a single user	**TL1** Number of users	Users (%)	**TL2** Number of users	Users (%)	**TL3** Number of users	Users (%)	**TL4** Number of users	Users (%)
0	12265	26.64	15037	32.66	10727	23.30	4922	10.70
(0 – 5]	33164	72.04	19261	41.84	12085	26.25	1655	3.60
(5 – 10]	513	1.11	7296	15.85	6418	13.94	1463	3.18
(10 – 20]	72	0.16	3498	7.60	7824	17.00	2660	5.78
(20 – 30]	15	0.03	643	1.40	4111	8.93	2324	5.05
(30 – 40]	3	0.01	176	0.38	2207	4.79	2105	4.57
(40 – 50]	0	0	66	0.14	1155	2.51	1832	3.98
(50 – 100]	2	0.01	51	0.11	1376	2.99	7250	15.75
over 100	0	0	6	0.01	131	0.28	21823	47.41

5 Conclusions

The analysis shows that for most of the users (about 70%) the forced stronger security mechanisms were almost invisible – max 10 times within a nearly 4-year period. So the usability can be considered as $usab_3 \approx usab_1$. However, it is true only when users do not change their behavior radically. The users who changed context very rarely were checked by stronger security mechanisms, which made them consider the system really secure (similar to [3]).

Context-oriented security scenario will not change the trust level when an attacker will make requests typical of user u (in the same values of context), which is (practically) unlikely.

We have also noticed that there was no significant difference in the analysis of 100 previous or all the previous to x-th access requests during the user profiles calculation.

In this paper we analyzed only two context parameters. There should be used a more complex context consisting of more context parameters. Then, user profiles will have a more uneven distribution of context values.

References

1. Balcerek, B., Frankowski, G., Kwiecień, A., Meyer, N., Nowak, M., Smutnicki, A.: Multilayered IT Security Requirements and Measures for the Complex Protection of Polish Domain-Specific Grid Infrastructure. In: Bubak, M., Kitowski, J., Wiatr, K. (eds.) eScience on Distributed Computing Infrastructure, pp. 61–79. Springer International Publishing (2014)
2. Bursztein, E., Martin, M., Mitchell, J.: Text-based CAPTCHA strengths and weaknesses. In: Proceedings of the 18th ACM conference on Computer and communications security - CCS '11. vol. 2011, p. 125. ACM Press, New York, New York, USA (2011)
3. Crawford, H., Renaud, K.: Understanding user perceptions of transparent authentication on a mobile device. Journal of Trust Management 1 (2014)

4. Cuppens, F., Cuppens-Boulahia, N.: Modeling contextual security policies. International Journal of Information Security 7(4), 285–305 (Nov 2007)
5. DeLone, W.H., McLean, E.R.: Information Systems Success: The Quest for the Dependent Variable. Information Systems Research 3(1), 60–95 (Mar 1992)
6. Dimmock, N., Belokosztolszki, A., Eyers, D., Bacon, J., Moody, K.: Using trust and risk in role-based access control policies. In: Proceedings of the ninth ACM symposium on Access control models and technologies - SACMAT '04. p. 156. ACM Press, New York, New York, USA (2004)
7. Furnell, S.: Usability versus complexity striking the balance in end-user security. Network Security (12), 13–17 (Dec 2010)
8. Gdańsk University of Technology: My GUT (2013), https://my.pg.gda.pl
9. Krawczyk, H., Lubomski, P.: CoRBAC kontekstowo zorientowany model bezpieczeństwa. Studia Informatica 34(3), 185–194 (2013)
10. Lubomski, P.: Context in Security of Distributed e-Service Environments. In: Proceedings of the Chip to Cloud Security Forum 2014. Marseille, France
11. Maamar, Z., Benslimane, D., Narendra, N.C.: What can context do for web services? Communications of the ACM 49(12), 98–103 (Dec 2006)
12. Mayrhofer, R., Schmidtke, H.R., Sigg, S.: Security and trust in context-aware applications. Personal and Ubiquitous Computing (Nov 2012)
13. Pahnila, S.P.S., Siponen, M.S.M., Mahmood, A.M.A.: Employees' Behavior towards IS Security Policy Compliance. 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07) (2007)
14. Sliman, L., Biennier, F., Badr, Y.: A security policy framework for context-aware and user preferences in e-services. Journal of Systems Architecture 55, 275–288 (2009)
15. Stanton, J.M., Stam, K.R., Mastrangelo, P., Jolton, J.: Analysis of end user security behaviors. Computers & Security 24(2), 124–133 (Mar 2005)