Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 6 (2011) pp. 415–423
© MIR Labs, www.mirlabs.net/jias/index.html

Security for ICT Collaboration Tools

Gerben Broenink, Geert Kleinhuis and Frank Fransen

TNO, department: Security,
Eemsgolaan 3, 9727 DW, Groningen, The Netherlands
{Gerben.Broenink, Geert Kleinhuis, Frank Fransen}@tno.nl

In order for collaboration tools to be productive in an operational setting, an information base that is shared across the collaborating parties is needed. Therefore, a lot of research is done for tooling to create such a common information base in a collaboration tool. However, security is often not given a lot of attention. In this paper we argue that security is a necessary part of collaboration systems. We identified and categorized security issues in a collaboration system, MiReCol, and recognized a new group of security issues that apply to collaboration tooling in general. Those new issues are related to the fact that several different authorized users are using the collaboration tool together. In those situations, the threat exists that an unauthorized user can view confidential data. We have researched possible countermeasures against these new threats. Some possible countermeasures are already being researched, and are mentioned in this paper. One of them, 'labeling and release' has been worked out in more detail, to research what the consequences are of this countermeasure.

We conclude that security is a necessary part of an ICT collaboration tool, and depending on the case, security countermeasures have to be implemented. Most security issues can be handled with existing techniques. However, to protect confidentiality in a multi-user environment, extra techniques have to be developed.

Keywords: Security, Collaboration, Confidentiality, Mixed Reality, Threats, Countermeasures

I. Introduction

Collaboration is an expensive and necessary activity. For almost all design decisions and calamity handling some kind of collaboration is needed. Therefore, collaboration tools are developed, to make collaboration easier and cheaper. In most collaboration tools, security is not given a lot of attention [2, 3, 4, 7, 8]. The three security properties (Confidentiality, Integrity and Availability), often do not get much attention in collaboration tools; sharing information is often a key part of the tooling, so confidentiality seems to be less important. And because most tools are tested in an environment where availability is not an issue, availability is not taken into account. The third security property, integrity, is often taken for granted, or is considered less important.

In this research, we have looked at possible security issues which arise when ICT collaboration tools are used. This is done by analyzing plausible security issues in the MiReCol system.

A. Related research

In the field of ICT collaboration tooling, not much attention is given to the security aspects of the collaboration tooling. Aim of most research projects is to increase the collaboration functionality, and less on securing the tools. However, some other research projects are found.

Walter-Franks e.a. [14] studied detection of users by infrared sensors. This can be used to detect that users are in the proximity of the table and when users are walking around the table. Each individual user can be tracked.

Agerwall e.a. [1] proposed a model to gain trust in a collaborative environment, in an incremental way. Instead of only configuring rights in the registration phase, this model supports a dynamic trust level, which allows a user to gain trust during one session, or several sessions. In this way, the natural human-to-human trust relations are mimicked.

Tolone e.a. [13] studied the access control requirements for collaborative systems, and investigated the current access control models, and mentioned their shortcomings. Finally they described criteria for comparing different models.

Kim e.a. [9] constructed an overview of authentication methods on tabletops. Their overview contains several pin-authentication methods, graphical passwords and pressure passwords.

The IR Ring [11] is developed to authenticate users touches on a multi-touch display. It is a ring-like device, which can be used to authenticate the touches of the different users on a single display. It is a user friendly way to authenticate touches. It can, however, not be used to identify all users, because only the users who touch the display and wear a ring are identified.

These projects address specific security issues in the collaboration field, however, they do not give a general overview of security issues in collaboration tools.

Earlier research [12, 16] mentions the authorization problems, however their focus is on collaboration tooling where multiple users are collaborating by using their own user interface (a single-user user interface). In our approach, we focus on several users collaborating on a common user interface (a multi-user user interface).

B. MiReCol (Mixed Reality for Collaboration)

To research the security issues of a collaboration tool, we researched MiReCol (Mixed Reality for Collaboration). MiReCol is a work-in-progress concept of an ICT collaboration tool that combines models from distinct domains to present a combined mixed reality, in which a user can get one interface to all the models. The program started because distinct domains existed which all had their own models, and own systems. Therefore, there was a desire to be able to combine all those models into one tool. For example, when modeling a new housing development, the tool should

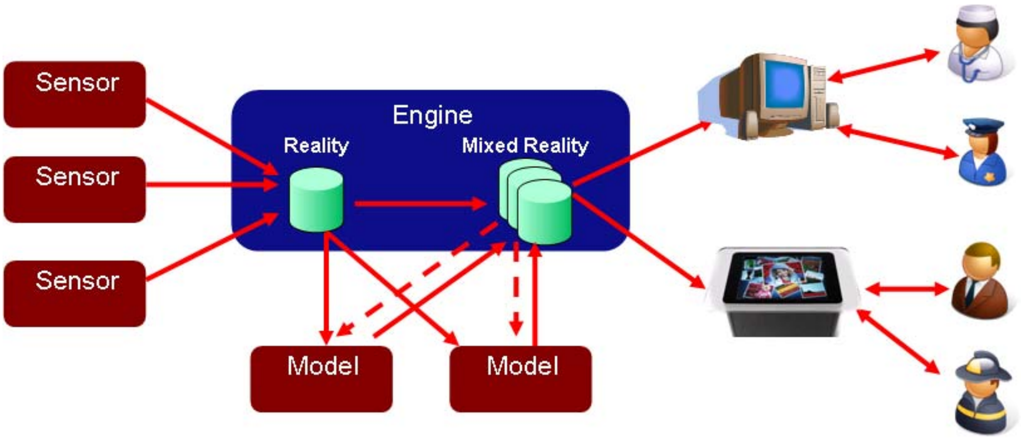

Figure 1: A functional view on MiReCol

be able to calculate the effects on the environment, using one model, and calculate the effect on the traffic density using another model. In this way, the user should be able to see all the effects of one decision at once. To create such a tool, an open architecture is needed to create the possibility that each collaborator could add their own models or sensors. Note that most collaboration tools focus on a common user interface which can be used for collaboration. However, in this concept the user interface is just a part of the project. There is also collaboration needed between the models and the sensors.

The MiReCol tool is far from finished; however there is a high level architecture. In Figure 1 a functional view on MiReCol is given. At the left, different kinds of sensors produce data which is stored in a database in MiReCol, together with all the other source data. Connected to MiReCol are several models which can process the source data. After a model has calculated its results, the results are stored back in MiReCol, together with all the other result/source data. It is possible that the data iterates this loop several times for different models. When the final result data is available, it is send to the user interface (In Figure 1, a web server and a multi-touch-table), where several people can see it together, and discuss about it.

C. *Aim of our research*

The aim of our research is to identify security issues of the MiReCol concept. We started by selecting two use cases in which the MiReCol system could be used. After that, possible security issues are identified, and finally we have analyzed the new collaboration specific issues and researched possible solutions to these new issues.

In Chapter two, use cases which are used for the security threat assessment are explained. After that, in Chapter three, attention is given to the newly found security issues. In Chapter four followed by the possible solutions to the issues. Finally in Chapter five, we present our conclusions.

II. Use cases

In this chapter we introduce two typical use cases of the MiReCol concept. The two use cases are: the planning phase of a building for a new embassy in The Hague, and a Flood control tool to calculate the strength of dikes and calculate which dike will break first, during a storm. For both use cases, we estimated the importance of the three commonly accepted security properties: confidentiality, integrity and availability.

A. *Use case: Embassy*

In this use case, plans are being made to build a new embassy in The Hague. A friendly nation wants to build a new embassy, and has to collaborate about the plans with the government, the police, fire brigade and the residents' association. Each member in this collaboration has its own

Table 1: Sessions in the embassy case

Session	Participants	Views	Confidential	Availability	Integrity
1	Mayor Police department Security agencies	Security view Traffic view	Yes	No	Yes
2	Mayor Residents association	Traffic view Noise pollution view 3D view	No	No	Yes

Table 2: Session in the flood control case

Session	Participants	Views	Confidential	Availability	Integrity
1	Regional water authority Government Military	Dike strength view Weather view Evacuation view	No	Yes	Yes
2	Regional water authority Government Firm of contractors	Dike strength view	No	No	Yes

concerns; the fire brigade wants the new embassy to be fire safe, the police department wants the new building to be easy to protect, for example during protests and state visits. The residents' association wants to have a friendly neighborhood. During public sessions all collaborators will sit together and discuss the different options. The police may require roadblocks to prevent cars from getting to close to the building, and the residents' association may demand that the roadblocks look friendly (e.g. no concrete blocks but flower tubs). The advantage of MiReCol is that all collaborators can try their preferences, and see what the consequences are. The collaborators will discuss public information during some of the sessions. However, it is possible that the police want to make plans about how they can protect a visitor during a state visit (e.g. by using roadblocks or snipers). The police might want to determine what the optimal use of security resources is (e.g. roadblock positions). It is clear that the police department does not want to share this information with the residents' association, and therefore, two separate sessions are organized. In Table 1 both sessions are shown, as can be seen, there is a significant difference between both sessions, both use different views and have different confidentiality levels. However, the decisions made in session one can have consequences for session two, and the other way around.

B. *Use case: Flood control*

The Dutch are continuing their fight against the water. If all the dikes would break, half of the country would be flooded, because half of the country is below see level. Therefore, designing, building and monitoring dikes are important activities. To monitor a dike, a variety of sensors can be used. Traditionally, dikes are inspected by hand, and different inspectors report their findings to a central point. However, there are also several automated techniques to inspect a dike. All these findings must be combined to get an overall view of the status of the dikes. In case of a calamity (e.g. a huge storm is coming) this combined view can be used to calculate which dikes need emergency repairs and which do not. It is also possible to predict how long it will take before a dike breaks, and how many people need to be evacuated. In Table 2, two possible sessions are shown. One session is an emergency situation in which the government, the regional water authority and the military try to secure the dikes, or evacuate the people. Another possible session is a session between the regional water authority, the government and a firm of contractors, to discuss how existing dikes should be strengthened, and new dikes should be build.

III. Security issues

In the previous chapter, we have seen that collaboration tools can be put into action in different cases, and those cases put different requirements on the security features of the used collaboration tool. To be able to construct those requirements, an overview of possible security issues is needed. Therefore a threat assessment has been conducted to identify those security issues.

A. *General security issues*

In many aspects, a collaboration tool is not different to any other software program. Therefore, many security issues which apply to information systems in general also apply to collaboration tools. Some examples of these security issues with respect to collaboration tools are:

- Due to malfunction of a sensor or malicious tampering with one or more sensors attached to the tool, could result in incorrect input of sensor data. In other words, the integrity of the source data is affected.
- Unauthorized access to the database can cause information stolen tampered with, and/or deleted. In other words, the confidentiality, integrity and/or availability of the data could be breached. Moreover, when the database contains personal data, then unauthorized access may cause an unauthorized invasion of privacy.
- By eavesdropping on the communication the confidentiality of the information that is transferred could be breached. Again this may also result in an unauthorized invasion of privacy.
- Configuration errors could for instance result in problems on the interfaces between the sensors and the collaboration tool (e.g. the sensor measures the temperature in degree Celsius and the collaboration tool expects the temperature to be in Fahrenheit). The integrity of the source data may thus be affected.
- Power failure, system defects, loss of communication, fire, floods, etc. are common causes for affecting the availability of an information system. These also apply to collaboration tools.

In the appendix a more extensive overview of security issues we identified for the MiReCol system are presented.

Most of the security issues we identified are well known within the information security community. And although dealing with some of these security issues may even be a hard challenge for particular environments, security solutions do exist. In the assessment of the security of the MiReCol

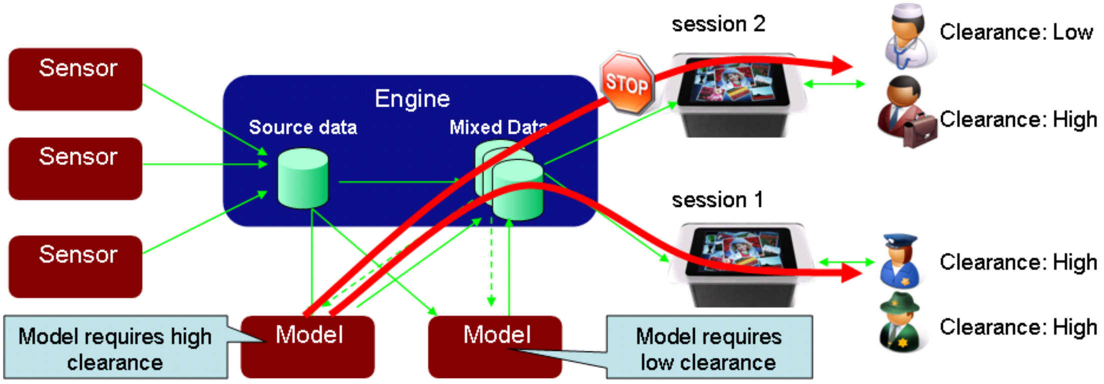

Figure 2: More complex access control

system, we did however also identify two security issues that are not that well known. In the following two sections these two security issues will be introduced.

B. *Access control gets more complex*

In some of the use cases we have seen that confidentiality can be important in collaboration environments. One of the important security functions to protect confidentiality is access control. In many cases access control is regulated by techniques like: requiring a user to enter his name and password, or to use his smartcard. However, in collaboration tools these general countermeasures can not easily be implemented. Collaboration tools often use multi-touch tables or beamers to involve all users in the collaboration. In this way, multiple users are using one user interface. As an effect there is no easy mechanism to identify and authenticate all users that are participating in the collaboration session.

An easy solution would be to demand the session leader to identify himself, and use his credentials to determine the rights for the current session. However, when multiple users have different rights, this could lead to conflicting situations. For example in

Table 1, the mayor could be the session leader in both sessions. However, in session one confidential data may be viewed, and in session two confidential data may not be viewed. It can be concluded that the rights for a session are not solely dependent on the session leader, but on the combination of all users. To prevent unauthorized users access to confidential data, rights have to be assigned based on the least privileged user. And to determine the rights of the least privileged user, all users have to be identified. In Figure 2, an example is given of model data which may flow to one session; however the same data may not flow to another session, while there is at least one user with a high clearance in both sessions.

This issue can be subdivided into the following security challenges. First there is the (preferable unobtrusive) identification of all users in a collaboration. Secondly, there is the problem of determining the authorizing this group of users. These are two separate challenges, which have to be solved both.

C. *Information leakage between sessions*

In some of the use cases we have seen that confidentiality can be important in collaboration environments. Therefore it is important that high confidential data cannot be viewed in a less confidential session. However, in some cases it might be possible that high confidential data / models influence less confidential data / models. For example, in the embassy case, the security agencies might want to use snipers to protect visitors during a state visit. Security models can be implemented to require a line of sight between the embassy building, and the possible sniper positions. When a user in a public session wants to place a tree on this line of sight, the security model has to forbid this option. However, by objecting this option, an information leakage is created, because a user could start guessing why the system objected this option. In this case, there is no information leakage, but meta-information leakage. The line of sight has not been revealed, but the consequence of the line of sight requirement (forbidding the tree) has been revealed. The collaboration tool not only has to take care of the confidentiality of the data, but also of the confidentiality of the meta-data.

D. *Conclusions*

In collaboration tooling, security requirements should play an important role. However, most security issues are not new, and countermeasures are already known. In many aspects, security requirements of collaboration tooling are similar to security requirements of other applications. However, when collaboration tools are used in separate sessions, with different confidentiality levels, new security issues arises. The following two new security issues have been indentified:

1. Correctly and completely identifying and authenticating all users in a multi-user collaboration session
2. Prevent information leakage between different authorized sessions.

To counter these security issues existing security solutions have to be modified, or even new solutions have to be developed.

IV. Solutions

In the previous chapter, we identified different security issues of collaboration tools. Most issues are general security issues for all applications. However, two new security issues are found, for which new countermeasures are needed. Two possible solutions can already be found in literature, which can (partially) solve the recognized issues. Those solutions are: unobtrusive identification methods and labeling and release. The second solution is part of our own research. First we will give attention to unobtrusive authentication methods as they can be found in literature, secondly we will present the labeling and release solution in more detail.

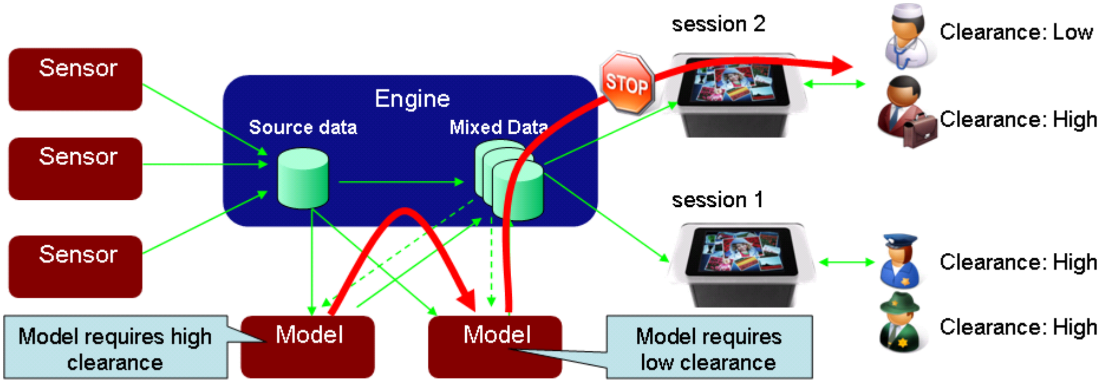

Figure 3: Information leakage between sessions

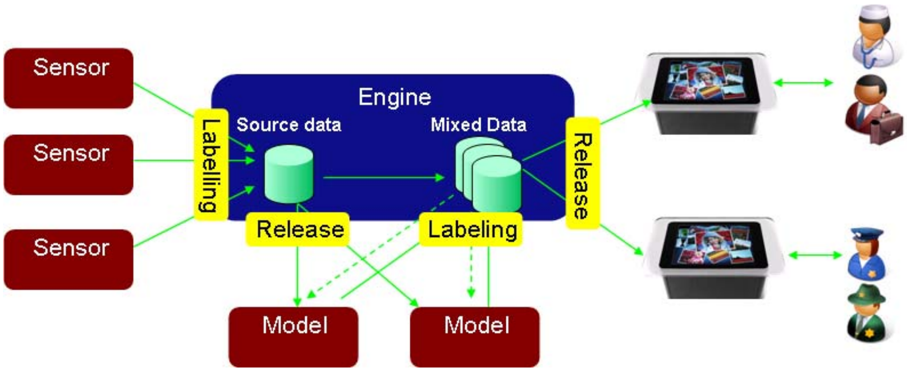

Figure 4: Label and release functionality

A. *Unobtrusive identification methods*

To solve the issue that access control gets more complex, unobtrusive authentication methods can be used. A collaboration tool has to identify and authenticate all users in the collaboration session, before it can decide which data can be viewed by the user. Most commonly used identification and authentication techniques only request the identity and credentials of a single user, and assume that there are no other users. Within the collaboration concept however, there can be more (different authorized) users at the same interface, at the same time. Therefore, the collaboration tool has to 'sense' the presence of its users, and request credentials of all users. We mention two research projects which develop solutions to this challenge. Note that unobtrusive identification solves only one of the two challenges we mentioned in paragraph 3B.

1) *HUMABIO*

The HUMABIO program [5] covers the concept of an unobtrusive multimodal biometric authentication. Within the HUMABIO program several multiple biometric sensors are combined to identify humans. E.g. a combination gait information, height information, voice and face recognition could be used in a pilot to identify employees in an airport.

2) *RFID badges*

RFID techniques can be used to identify and locate people [15]. This technique is more obtrusive. It requires that all users to wear an active RFID badge that is used to locate the users, and monitor when they enter the room where a collaboration tool is used to discuss confidential data. This technique is user friendly and easy to use. A malicious user can, however, easily bypass the system by not wearing his RFID badge. In addition, locating users in a building by means of actives badges can also be seen as less privacy friendly.

B. *Label and release data*

To solve the multi-user access control that prevents leakage of confidential information to a less confidential session, labeling and releasing of data is a possible solution. To guarantee that confidential data is not sent to unauthorized users, all outgoing data can only be released if it has been checked whether it might be released or not. This is called a release functionality; it requires that some (part of a) system is responsible that only authorized users can request confidential data [10]. In Section 2 this release functionality is explained. The release functionality needs to check the confidentiality level of the data. This task becomes easier when all data in the system contains a label, which specifies the confidentiality level of the data. This can be achieved by a labeling functionality, which labels all incoming data. This function will be explained first in Section 1. How the functionality can be incorporated in the MiReCol functionality is shown in Figure 4. Finally, we will present the consequences of labeling and release in Section 3.

1) *Labeling functionality*

The aim of a labeling function is to classify all incoming data with a label, which can be attached to the data. When a labeling mechanism classifies all incoming data, we can be sure that all data in the database has a corresponding label. A label contains meta-data, for example, all data which is produced by a surveillance camera could be labeled as 'confidential' because it contains privacy information which only may be viewed in case of a calamity. Other information, like a street map, may be labeled as 'public', meaning that everybody may view the data. When a collaborator wants to join collaboration, and connect his sensors and models, he can decide about the label he wants to give to their data, and who is allowed to view the data. The labeling mechanism can classify the data based on the source of the data, and the policy which has been formulated by the collaborator.

The labeling mechanism can 'sign' the label, so the receiver of the label can verify the correctness of the label, and ensure that no one has modified the label after it has been created. However, signing the label would have several consequences for the architecture, and it depends on the importance of the information in the system whether it is valuable to implement the signing of labels. In Section 3 we will elaborate this point.

2) *Release functionality*

It has to be guaranteed that only data which is allowed to be transported to the receiver (a user interface, or a model) can leave the system. To guarantee this, release functionality can be implemented. Release functionality is placed at the border of the system, so it can check all leaving data. The release function will check the confidentiality level of the data (for which it can use the label) and the confidentiality level of the session. When the user is authorized to get the data, the data is released. Otherwise, the data is blocked.

The level of security mechanisms in the release module depends on the impact when classified data is revealed to the wrong people. For example, when a company would go bankrupt when the information is leaked, it is useful to add security mechanisms in the release module. However, when it is only unfortunate that information is leaked, it is not valuable to add security mechanisms.

3) Consequences

Each security mechanism has it's consequence on the entire system. In this way, each mechanism has its price, which has to be paid to increase the security of the entire system. In this paragraph we will elaborate the consequence of the labelling and release mechanisms, and also give some options to increase or decrease the security level and the consequences of the measures.

Public Key Infrastructure

When a label is created by a labelling authority, meta-data is connected to the original data. However, when the label is transported to another system or entity, it is impossible to verify that the labelling authority created the label, and no-one has modified the label since. Therefore, a labelling authority could 'sign' the label with a digital signature. However, such a signature would require a Public Key Infrastructure (PKI) to be installed. The concept of a PKI is a well known and proven technology, it is also a complex technology; each collaborator has to trust the certificate authority and the certificate authority has to distribute the public key's trough the network. So, it is a well known, but also an expensive technology. The alternative is to not sign the labels, and store the labels in a place where every-one has read-only rights, and the labelling authority has write-rights.

Storage of labels

There are two important issues with the storage of the labels. The first is the fact that the collaborators have to trust that only the labelling authority can write labels, and no-one can change them. This is especially important when the labels are not signed.

The second issue is the fact that it must be possible to retrieve the label of each data object. This will require some system to organise the labels and data objects in such a way to retrieve both. There are several options to do this:

- Store data object and label in a database. In this way, the label can just be another column in a database table. It will also make rights management on labels a lot easier. However, it has also some drawbacks. It will require the data itself to be formalized enough to be stored in a database. A second drawback is that the label and the data cannot easily be transported to another party when this party is not using the same database model.
- A second option can be used when the original data is stored in a data object. (for example, an XML file) In this case, the label can be added in the original file. An advantage of this option is that all existing storage mechanisms can stay in place. However, this option also has several drawbacks. First the original file type must be expandable. Secondly, for each file type a separate format must be used.
- A third option is to store the label in a separate file, next to the original data object. The advantage is that is can be used by each file type, whatever the original file type might be. However, a drawback of this option is that some kind of content management system is needed to organize all those separate files.

In a complex environment, storage of labels will be complex to. There are several options to store the labels, and they all have their advantages and disadvantages.

Policy management

A labelling mechanism as well as a release mechanism requires a policy which describes their functionality. A labelling mechanism has to be aware under what circumstances a data object should be labelled with which label. And a release mechanism should be aware when a data object might be released. Therefore policies have to be written and distributed among the mechanisms. However, when the number of mechanisms increases (when multiple data sources and models are used, the number of mechanisms increases) it will be harder to maintain a consistent set of policies. Therefore, some kind of policy management is needed to detect and solve conflicts in policies and maintain a consistent set of policies.

Balancing measurements and consequences

As mentioned before, each measurement has its consequences and its price. Depending on the importance of the collaboration, a balance between the measurements and the accepted risks has to be found. For each type of collaboration the amount of consequences people are willing to take to accept to counteract on the risks will change. As a result each type of collaboration will find another balance.

V. Conclusions and future research

Collaboration is an expensive and complex activity. To support this process, more and more collaboration tools are developed. Collaboration tools often provide efficient communication between the collaborations, and shares data between them. However, security issues are often not given a lot of attention. In this paper we have showed that security can be an issue in collaboration tooling. Most possible security issues are collaboration tools are common security issues, which have common countermeasures. However, two new security issues are found, and described. Those issues are: the more complex access control, and the information leakage between sessions. Those issues arise in a multi-user user interface, multiple different authorized users collaborate with one user interface.

There are already research projects to develop techniques which possible can counteract those issues, those techniques are gait, face and voice recognition. However those techniques are not mature yet. Therefore, more research is needed to identify users in an unobtrusive way, so the access control problem can be solved.

More research is also needed to develop techniques which can prevent information leakage between collaboration sessions. To prevent information leakage, labeling and release techniques can be used, however with the current techniques it is practically impossible to prevent meta-data leakage.

Appendix: Overview of security issues

During the project, we focused on identifying general security threats to the MiReCol system. The identified threats are grouped in six categories. In the following paragraphs, those categories are discussed.

A. Integrity of source data

Several sensors are generating data to the MiReCol platform, which is used by the models. It is critical for MiReCol that the generated source data has a sufficient level of integrity. A lack of integrity would cause integrity errors in the model results. There are several possible causes for a lack of integrity in the source data:

S1. Malicious tampering with sensor or sensor data: A malicious owner of a sensor could tamper with the sensor or the sensor data. This would cause the MiReCol engine to store malicious data in the database.

S2. Defect sensor. A sensor could be malfunctioning, and therefore generating wrong data.

S3. Interface problem. When the interface between a sensor and MiReCol is not clearly determined or incorrectly configured, possible errors arise. E.g. when a temperature sensor measures the temperature in Fahrenheit, and MiReCol expects the temperature to be measured in Celsius, the integrity is violated.

S4. Unauthorized access to the database. When an unauthorized user gets access to the MiReCol engine, and is able to change the data in the database, he can cause an integrity problem.

All mentioned issues are visualized in Figure 7a.

B. *Integrity of model data*

Like source data, a lack of integrity of model data causes MiReCol to generate erroneous results. Again, there are several possible causes of the lack of integrity:

M1. Malicious tampering with model or model data: A malicious owner of a model could tamper with the model or the model data. This would cause the MiReCol engine to store malicious data in the database.

M2. Incorrect model. A model could be malfunctioning, and therefore generating wrong results.

M3. Interface problem. When the interface between a model and MiReCol is not clearly determined or incorrectly configured, possible errors arise.

M4. Unauthorized access to the database. When an unauthorized user gets access to the MiReCol engine, and is able to change the data in the database, he can cause an integrity problem.

M5. Lack of integrity of input data. When there is a lack of integrity of the input data, a model is unable to calculate correct results.

All mentioned issues are visualized in Figure 7a.

C. *Confidentiality*

When confidential or privacy sensitive data is used, this data must be protected. There are several issues concerning the confidentiality of the data:

C1. Unauthorized access to the database. When an unauthorized user gets access to the MiReCol engine, and is able to read confidential data in the database, he can cause a confidentiality problem.

C2. Eavesdropping the communication. When an attacker is able to eavesdrop the communication between a model and MiReCol, or between a sensor and MiReCol, a confidentiality problem arises.

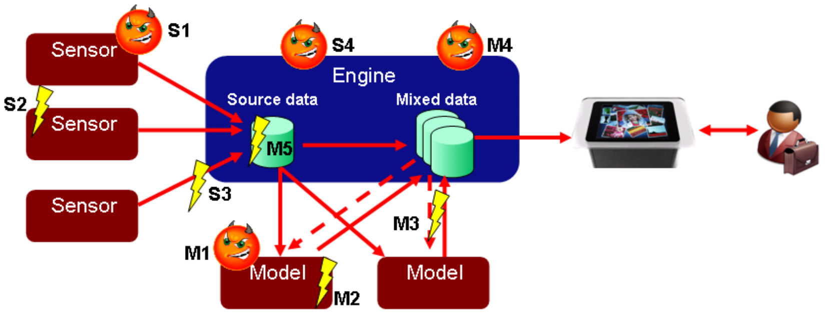

Figure 7a: Integrity issues

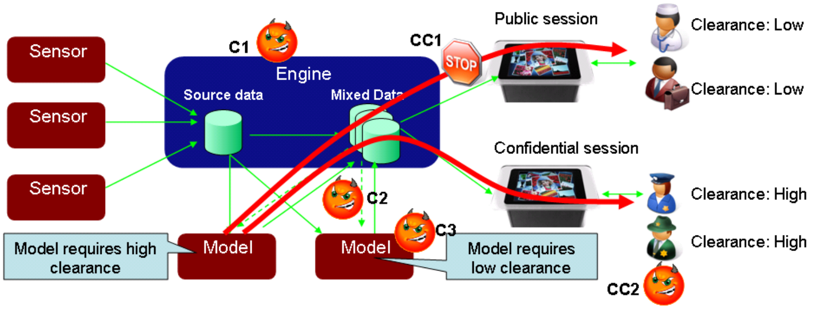

Figure 7b: Confidentiality issues

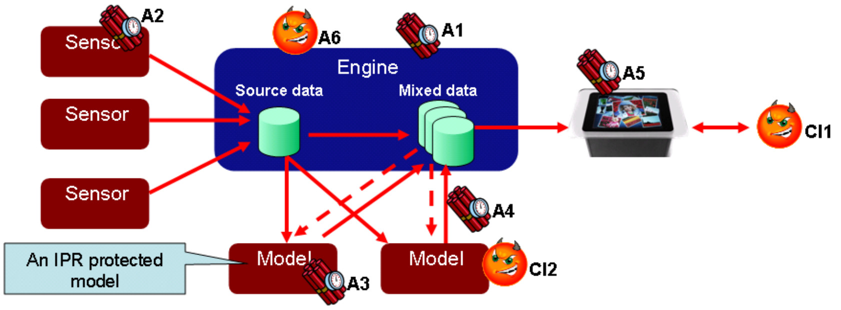

Figure 7c: Confidentiality en availability issues

C3. Malicious model. When a malicious model is processing confidential data, it can possibly leak this data to unauthorized parties.

All mentioned issues are visualized in Figure 7b.

D. Confidentiality in collaboration

There are some issues arising, because of the collaboration part of MiReCol. These issues arise, because multiple users are using one interface simultaneously. When not all users are authorized at the same level, the MiReCol engine has to decide whether it is allowed to display confidential data. There are two possible causes of this issue:

CC1. Confidential data is leaked during a session with different authorized users. During a session in which unauthorized people participate the engine should not allow confidential data to be shown.

CC2. Access control gets more complex. During a confidential session, confidential data can be showed by the system. If the system did not completely and/or correctly identify and authenticate users participating in the collaboration session the confidentiality may be breached.

These are the issues we have identified as new issues in a multi-user user interface. These issues are further explained in Section 3. The mentioned issues are visualized in Figure 7b.

E. Confidentiality of IPR

Some models which are connected to MiReCol might be protected by IPR. In these cases, another issue arises, namely the protection of IPR. There are two possible threats to the IPR:

CI1. A user accesses and copies all model logic. When a model is connected to the MiReCol system, a user can use the model. However, when the model is IPR protected, the user is not allowed to read all model data and rebuild an own copy of the model.

CI2. A malicious model accesses and copies another model. This threat is similar the threat CI1, however now the copying is done by another model, instead of a user.

All mentioned issues are visualized in Figure 7c.

F. Availability

MiReCol is connected to different sensors and models, which causes MiReCol to be dependent on these systems. When one of the other systems has availability issues, MiReCol itself has possible availability issues. An availability issue in MiReCol can be caused by the following reasons:

A1. Failing MiReCol engine. The MiReCol engine itself can fail, because of power failure, or a software bug.

A2. Failing sensor. When a sensor is critical for some functionality, a failing sensor can cause a failing functionality.

A3. Failing model. When a model is critical for some functionality, a failing model can cause a failing functionality.

A4. Failing connections. When a connection is lost, all functionality which depends on this connection will fail.

A5. Failing multi-touch-table. When the multi-touch-table is failing, it will be impossible to display the results to the user.

A6. Attacker. When an attacker succeeds in a denial of service attack, he can cause availability issues.

All mentioned issues are visualized in Figure 7c.

References

[1] Agerwall, D. et. al. 2003, A new security model for collaborative environments. *Lawrence Berkeley National Laboratory:* LBNL Paper LBNL-52894. Retrieved from: http://www.escholarship.org/uc/item/0dc3q2sj

[2] Billinghurst, M. Kato, H., 1999. Collaborative Mixed Reality, *Proceedings of the first international symposium on mixed reality (ISMR '99)* Berlin, Germany, pp. 261-284.

[3] Booth, K. et. al., 2002, The “Mighty Mouse” Multi-Screen Collaboration Tool, *Proceedings of the 15th annual ACM symposium on User interface software and technology,* Paris, France, pp. 209-212.

[4] Brown, B. et. al., 2003. Lessons from the lighthouse: Collaboration in a shared mixed reality system. *Proceedings of the SIGCHI conference on Human factors in computing systems.* Florida, USA, pp. 577-584.

[5] Damousis, I.G. et al. 2007, Unobtrusive multimodal biometric authentication: The HUMABIO Project Concept, *EURASIP journal on advances in signal processing*, volume 2008, Article ID 265767, pp. 1-11

[6] Georgiadis, C. et. al. 2000, *Context and Role Based Hybrid Access Control for Collaborative Environments. In Proceedings of the fifth Nordic Workshop on Secure IT systems, Reykjavic, Iceland.*

[7] Hughes, C.E. et. al., 2005. Mixed Reality in Education, Entertainment, and Training. *IEEE Computer Graphics and Applications*. Vol. 25, no. 6, pp 24-30 Nov.-Dec. 2005.

[8] Kaufmann, H., 2003. Collaborative Augmented Reality in Education, *Imagina Conference 2003,* Monaco, Monaco.

[9] Kim, d., 2010. *Multi-Touch Authentication on Tabletops,* CHI 2010, Proceedings of the 28th international conference on Human factors in computer systems. pp 1093-1102.

[10] Paske, B.J. et. al. 2009, Information Labeling – Cross-Domain-Solutions, in *Intercom 2009*, volume 9, number 2, pp 47-50

[11] Roth, v. et. al. 2010, *The IR Ring: Authenticating Users' Touches on a Multi-Touch Display*, in Proceedings of the 23nd annual ACM symposium on User Interface Software and Technology, pp 259-262

[12] Sikkel, K. et. al. 1998, *User-Oriented Authorization in Collaborative Environments, in Proceedings of COOP'98, Cannes, 1998, pp. 175-183.*

[13] Tolone, W. et. al. 2005, A Access control in collaborative systems, in ACM Computing Surveys (CSUR), volume 37, issue 1, pp 29-41

[14] Walther-Franks, B. et. al.. 2008, User detection for a multi-touch table via proximity sensors. In Proceedings of the IEEE Tabletops and Interactive Surfaces. Amsterdam: IEEE Press.

[15] Yin, G. et. al. 2006, An Indoor Localization Mechanism Using Active RFID Tag, *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing,*

Author Biographies

Gerben Broenink was born in 1983. He received, in 2008, the masters degree Computer Science at the University of Twente. He joined TNO, where he is currently working for the security department, where he participates in scientific research projects for the government and the military. His field of expertise includes: network security, analyzing and developing the security of systems and services.

Geert Kleinhuis studied Computer Science at the Technical University Twente in Enschede. He graduated in 1989 after which he started working for the Royal Dutch Army as a computer engineer. In 1991 he joined KPN Research working in the area of Cryptology and Security. His areas of expertise include: Public Key Infrastructures (especially new security services based upon PKI), application of cryptography, Electronic Payment Systems, Internet security, intrusion detection systems, Risk Analysis and security management development. He also participated in the design and standardisation of security for systems as UMTS and PKI. In January 2002 he joined TNO working in the area of Telecommunication and Security. In his recent projects PKI, internet security in general and physical security are very important areas**.**

Frank Fransen received his master degree in Information Technology at the Technical University of Eindhoven in 1995. He is currently employed as a Senior Scientist in the security group of TNO. His work at TNO involves the study of emerging security technologies, analyze the security of systems, and advise organizations on the security of their systems. Main research topics are mobile network security, mobile phone security, security of smart cards / RFID / NFC systems, and sensor network security