REQUIREMENTS AND EVALUATION TECHNIQUES FOR ONLINE-VOTING

Melanie Volkamer[1], Robert Krimmer[2]

Abstract – Online-Voting is more and more in use, and even in political elections as the current Estonian elections (March 2007) show. Currently these systems are not either not evaluated or the evaluation reports are not comparable, even if each of them is consistent and contains trustful and convincing arguments. The problem is the leak of unified rational requirements and evaluation techniques. Therefore, after having presented both in [11] for voting devices, we present in this paper a list of requirements Online-Voting shall ensure and describe how to evaluate the different categories of requirements

1. Introduction

Several Online Elections trials took place in the past ten years all around the globe in various forms and on different political levels. Some of the online elections where even legally binding. The decision to choose a specific approach or even a concrete system was based on a couple of different facts. In general, the choice was not only made by the responsible election authority (REA). Rather the REA was supported by either a group of security experts, a security company, or a security civil service that analysed the system. These groups used different approaches for their analysis and focused on different aspects: eg, concentrating on the cryptographic protocol or a more general testing of the software. Thereby each group defined their ”own” re-quirements (mainly deduced from the five election principles) the system should be evaluated against. Thus, the level of details of the requirements definition differs a lot. Moreover, the available list of requirements for Online Elections has in common that a concise definition of the intruder and trust model is missing. Consequently, the evaluations are not transparent to the public and it is not possible to compare different systems based on the evaluation reports Thus, this paper is a first step to come closer to a standardised Online Election System evaluation. As a starting point we need to define a complete, consistent and stable set of requirements (see section 5). For this paper we concentrate on functional and security requirements (so called security objectives) and narrow the objective of assessment down (see section 2). The formalism to describe this set is according to the Common Criteria Methodology [1] and is based on already existing sets of requirements (see section 3). The next step the requirements are classified in different categories according to the proposed evaluation technique.

[1] University Passau, Institute for IT-Security and Security Law, Innstraße 43, D-94032 Passau, Germany, melanie.volkamer@uni-passau.de

[2] Competence Center for Electronic Voting and Participation (E-Voting.CC), Pyrkergasse 33/1/2, A-1190 Vienna, Austria, r.krimmer@e-voting.cc

2. Object of Assessment

An Online Election system contains software and procedures before, during and after the elections. In the pre election phase, the generation of the election register and the ballot take place, depending on the authentication system the authentication tokens need to be generated and send to the voter. Moreover, the system needs to be configured (eg, the number of allowed candidates to be chosen). In the election phase, the voter casts his vote and the e-votes are collected and stored. In the post election phase, the election outcome is computed and depending on the system, voter or universal verification procedures can take place. Because of space reasons, we assume that the voter and only voters have the proper authentication tokens and all the configurations are made accurately (eg, no voters are missing in the election register, the candidates are listed in the right order and the e-ballot box is empty). Never the less we are aware of the problem, that intrusions to the pre election phase can violate already the election secrecy as well as change the outcome at the end. The other way round it is difficult to cover those e-voting systems where the voter already get an anonymous authentication token. Moreover, we are not discussing as detailed in [10]. In addition, we skipped the topic disputations, eg, voters not having received authentication tokens, coming to a negative verification result, who cannot log on the system, or cast a vote.

3. Requirements Definition

The requirements below are derived from the German list of requirements [12] (which are defined as a Protection Profile based on the Common Criteria) as an expansion of the requirements proposed in [11] for voting devices. While the main idea, methodology and the terminology is taken from [11], the new classifications of evaluation methodologies are related to [12]. We use the same terminology as described in the glossary of [11] and the key words SHALL and SHOULD are to be interpreted as described in RFC 2119.Words and phrases from the glossary appear ·like this·. The most important items are displayed in figure 1. We distinguish ·voting software on the voting server site· and ·voting server·. While the requirements containing ·voting software on the voting server site· demand the software itself to ensure it, the requirement containing ·voting server· can be either ensured by the ·voting software on the voting server site· or the environment of the software. Moreover, we are talking only about one voting server to keep the requirements more generic. In an instantiation to a specific system it can be replace to the particular component. Analogous to [11], each requirement is associated with at least one of the election principles. We add the category [dp] to classify requirements related to data protection. As already mentioned in [11], certain terms can only be defined by the ·responsible election authority·. They appear underlined.

In [11], the requirements are divided into the following categories: **system requirements**, **operational requirements** and **assurance requirements**. System requirements are further distinguishes between **functional**, **security**, **auditing** and **usability requirements**.

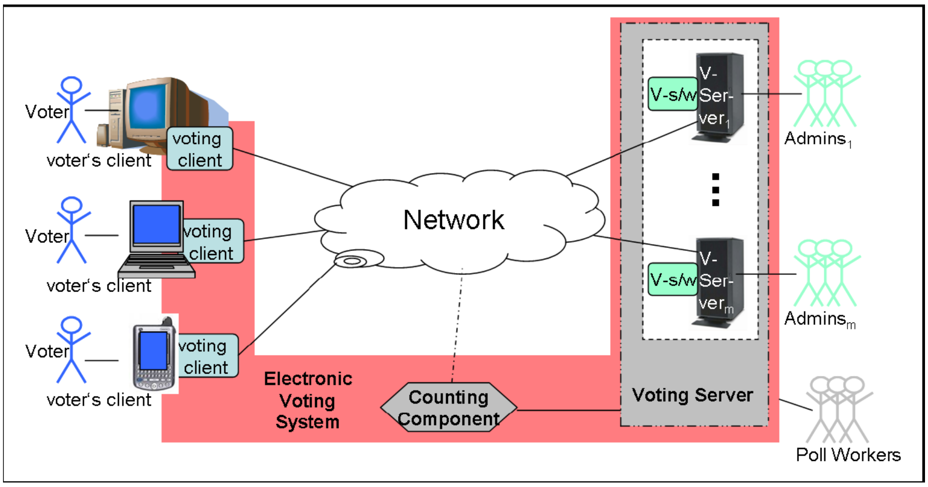

Fig. 1: Components of the e-voting system

The usability, auditing, operational and assurance requirements are very similar for ·*voting devices*· and ·online voting·. Because of space reasons we do not repeat those requirements with the small changes but concentrate on the functional and security requirements which we call Security Objectives (O). These are re-categorised according to the Common Criteria [1] speech: Security Objectives mitigating against specific Threats (T) (see section 5.1) and Security Objectives based on Organisational Security Policies (P) (see section 5.2). We assign possible threats to a security objective while we left out the definition of organisational security policies because they are similar to the corresponding security objective.

4. Methods of Assessments

The categorisation into the two kind of security objectives is relevant for the applied evaluation techniques. While those deduced from organisational security policies can easily be checked (does the corresponding functionality exists?) there is more effort necessary to decide about the others. Here the definition of the expected intruder model is essential, including the intruder's capabilities, his knowledge and his goals: With respect to his goals we distinguish between breaking the election secrecy, faking the election outcome, computing intermediate results and changing the ballot. With respect to his capabilities and knowledge be distinguish between

- **outside** intruders: those who use any public available information and can read on the network. Note: There is a difference between these information available during the polling period and those after. The second category of information can be used to compute intermediate results but this is uninteresting after the election) and
- **inside** intruders: those who can change software and data and who can decrypt data (knowing corresponding keys). In addition they know all the outsider information. These are either administrators, the developers of the software or the one hosting the voting server.
- a single malicious voter or a single malicious administrator.

After having defined all these threats the evaluation task starts. There are two possibilities for an e-voting system to overcome a threat: Either the e-voting system provides the functionality or the e-voting system needs to be supported by the environment to do so. For instance the e-voting system might only ensure the one-voter-one-vote principle under the additional organisation assumption that the voter does not forward his authentication token to someone else to give this person the possibility to vote in his behalf. Another example is that the e-voting system might only be receipt-free under the assumption that the voter's client is trustworthy. This fits very well for threats provoked by an outside intruder or a malicious voter. The evaluator would here come up with a list of assumptions which needs to hold in order to fend the threats. The responsible election authority has then to ensure that the assumptions are given in the environment where the e-voting system is used. Otherwise the e-voting system should not be used. With respect to the evaluation whether threats by inside intruders are fended we need to be careful because we are talking only about one voting server in the security objectives. If there would be only one entity running the whole election then such a single entity has to be trusted with respect to all security objectives. Thus, a common approach is to split the voting server into several sub entities such that compromising security objectives requires compromising two or more entities. To evaluate corresponding security objectives, it is essential to analyse the number of entities needed to be compromised in order to successfully attack the e-voting system according the described threat. To do so we introduced the term (k_1+...+k_m **out of** n_1+...+n_m)^i**-resilient**: **A** system is called k-**resilient w.r.t. a security objective**, $k \geq 1$, if at least k entities need to be corrupted to compromise the security objective. So, if a system relies on a single trusted entity with respect to a particular security objective it is just 1-resilient w.r.t. this security objective.

If more entities are involved it is generally easier to find a collusion of corrupted entities. For example, a 1-resilient system in which any of 10 involved entities can be corrupted is worse than a 1-resilient system in which any of 5 involved entities can be corrupted. Therefore, we propose to determine for all possible threats the total number of entities n and call it k **out of** n-**resilient w.r.t. a security objective**.

Moreover, we have to take into account that, eg, in case of 2 -resilient, it might happen that for one entity you can choose one out of x and for the second one you can choose one out of y. To display this in our result, we use (k_1+...+k_m **out of** n_1 +...+n_m). For some security objectives there might be different possibilities to violate it either by entity A and B or by corrupt entity C and D. In order to also cover this case, we propose (k_1 + ... + k_m **out of** n_1 + ... + n_m)^i**-resilient**.

We have not yet clearly defined what we call a sub entity of the voting server. Such an entity contains software, hardware and operating system in use on the server as well as the persons in charge (including the developers of the software, persons setting any configuration at the server as well as the administrators and poll-workers operating at this server).

Throughout it is assumed that the cryptographic primitives used are secure; for instance, it is assumed that the security provided by the symmetric ciphers used by the system cannot be broken other than by getting direct access to the secret keys.

5. Security Objectives

The security objectives are divided in those deduced from organisational security policies and those to fend threats.

5.1 Security Objectives related to Threats

O_1 [all] The *·voting server·* SHOULD be tamper-resistant. The *·voting server·* SHALL be tamper-evident.

T_1 An inside intruder tampers the *·voting server·* in order to

a. fake results (either by altering, adding or deleting *·votes·*),
b. compute intermediate results,
c. break the election secrecy,
d. prove the link between a particular *·voter·* and his *·vote·*,
e. get knowledge about personal data from the *·voters·*.

T_2 An outside intruder gets access to the *·voting server·* over the network in order to tamper it.

O_2 [all] The only application running on the *·voter's client·* SHOULD be the *·voting client·*. Where other application run in parallel they SHALL not interfere with the *·voting client·*.

T_3 Any intruder runs on the *·voter's client·* malware which either read the *·vote·* (break election secrecy), alter the *·vote·*, or read the authentication information to cast a *·vote·*(change the outcome).

O_3 [se] The *·e-voting system·* SHALL only provide receipt-free information to the *·voter·*.

T_4 An outside intruder can prove the link between a particular *·voter·* and his *·vote·*.

O_4 [se] The *·e-voting system·* SHOULD ensure that the *·voter·* is not able to construct a receipt proving his *·vote·*.

T_5 A malicious ·voter· uses all information sent to his *·voter's client·*, displayed on his *·voter's client·* and sent from his *·voter's client·* to construct a proof in order to sell his *·vote·*.

T_6 A malicious *·voter·* uses all information from above and uses intermediate results calculated on his *·voter's client·* to construct a proof in order to sell his *·vote·*.

T_7 A malicious *·voter·* cooperates with an intruder of the *·voting server·* in order to prove his decision.

O_5 [dp] The *·e-voting system·* SHALL ensure the data protection law with respect to the transmission of the *·identification data·*.

T_8 An outside intruder reading on the network gets knowledge about personal data from the *·voter·*.

O_6 [fr] The *·e-voting system·* SHALL not provide any information in transmitted protocol messages, which allow to construct the link between the a particular *·voter·* and his *·vote·*.

> *Appl. Note: The ·e-voting system· SHALL ensure that neither the ·vote· itself nor the number of chosen candidates (including an empty ballot), nor a spoilt ·vote· (eg, by using the length of the protocol messages) can be linked to a particular ·voter·.*

T_9 The outside intruder reading on the network breaks the election secrecy.

O_7 [fr] The *·e-voting system·* SHALL ensure the confidentiality of the transmitted *·e-votes·* during the *·polling period·*.

> *Appl. Note: The ·e-voting system·* SHALL *ensure that neither the ·vote· itself nor the number of chosen candidates (including an empty ballot), nor a spoilt ·vote· (eg, by using the length of the protocol messages depending on the approach) can be deduced by reading all transmitted voting protocol messages.*

T_10 The outside intruder reading on the network computes intermediate results.

O_8 [all] The *·voting client·* SHOULD only communicate with the proper ·voting server·. The *·voting client·* SHALL clearly indicate to the *·voter·* whether it is connected with the proper *·voting server·*.

T_11 An outside intruder tries to redirect the *·voter·* to a faked *·voting server·* in order to

- take his authentication data to vote on behalf of the voter,
- alter his ·vote· before forwarding it to the proper *·voting server,*
- delete his *·vote·* and thus change the outcome,
- get knowledge about the *·voter's· ·vote·* and thus break the election secrecy.

O_9 [all] The *·e-voting system·* SHALL verify the authenticity, freshness and integrity of all transmitted protocol messages before processing them.

T_12 An outside intruder

- replays old protocol messages or
- sends new ones to the *·voting server·*

in order to add votes and thereby change the election outcome.

T_13 An inside intruder replays old protocol messages or sends new ones to the *·voting server·* in order to add votes and thereby change the election outcome.

O_10 [di] The *·e-voting system·* SHOULD store all but only authorized·evotes· (eg, cast only one *·vote·* per *·voter·*, in the proper format, and from a proper platform) in the *·e-ballot box·*. The *·e-voting system·* SHALL be capable to distinguish between *·authorized·* and *·unauthorized· ·e-votes·*.

T_14 An outside intruder replays protocol messages in order to cast several *·votes·* and thus to change the outcome.

T_15 A malicious *·voter·* intruder logs on the *·voting server·* again in order to cast a second *·vote·* and thus to change the outcome.

T_16 An inside intruder alters the transmitted *·e-vote·* on order to change the outcome.

O_11 [all] The *·voting server·* SHALL implement an access control policy which

- restricts all activities to particular *·user·*-roles
- requires physical presence.

T_17 An outside intruder gets access to the *·voting server·* without knowing or having the access tokens in order to tamper the *·voting server·*.

O_12 [all] The access control mechanism SHOULD only allow access to the *·voting server·* if at least two different *·users·* are logged in.

T_18 A malicious administrator abuses his access privileges to tamper data, hardware or software.

O_13 [di] The *·voting server·* SHALL protect the integrity and authenticity of *·e-votes·* after the *·polling period·*.

T_19 Any intruder tampers with *·e-votes·* after the *·polling period·* and before the *·counting phase·* to change the outcome.

O_14 [di] The *·counting phase·* SHALL verify the integrity and authenticity of *·e-votes·* separate,

T_20 The intruder changes the set of *·e-votes·* on the way to the *·counting procedure component·* in order to change the outcome.

O_15 [di] The *·e-voting system·* SHOULD protect the integrity of *·election·*data (at least including: *·votes·*, results and audit information) as soon as results are calculated.

T_21 The intruder changes the result after the results are calculated in order to change the outcome.

5.2 Security Objectives related to Organisational Security Policies

O/P_1 [all] The *·voting software on the voting server site·* SHALL not provide any functionality to calculate results during the *·polling period·*.

O/P_2 [di] The *·voting server·* SHALL provide the functionality to completely delete all data from previous *·elections·*

O/P_3 [tr] The *·voting software on the voting server site·* SHALL provide the functionality to read *·e-votes·* from the *·voting server·* into any *·counting procedure components·* in the 'election report' state.

O/P_4 [fr] The *·voting client·* SHOULD provide the functionality for the *·voter·* to

- change his *·selections·* before *·casting·*.
- ·spoil· his *·vote·*.
- easily cancel his *·voting process·* at any time.
- clear all his *·selections·*

O/P_5 [fr] The *·voting client·* SHOULD warn the *·voter·* when he tries to *·spoil·* his *·vote·* in one or more *·polls·*.

O/P_6 [un] The *·voting client·* SHALL immediately transmit the *·e-vote·* to the *·voting server·*, when the *·voter·* has *·cast·* his *·vote·*.

O/P_7 [tr] The *·voting software on the voting server site·* SHALL provide feedback to the *·voter·* regarding the status of his *·vote·* (at least that his *·e-vote·* has been stored successfully in the *·e-ballot box·*).

O/P_8 [di] The *·e-voting system·* SHALL provide mechanisms to prevent data loss during normal operation, and in the case of exceptions, malfunctions and *·e-voting system breakdown·*.

O/P_9 [un] The *·e-voting system·* SHALL provide feedback in the form of error messages in the case of exceptions and malfunctions to the responsible *·administrators·* and *·poll-workers·*. Where a *·voter·* is in the *·voting process·* at that time he SHALL also get feedback.

O/P_10 [un] The *·e-voting system·* SHOULD prevent *·voter·* interaction in the case of exceptions and malfunctions.

O/P_11 [un] The *·e-voting system·* SHALL be capable to determine whether a particular *·voter·* cast a vote and his *·e-vote·* was successfully stored in the *·e-ballot box·* in the case of exceptions and malfunctions as well as after *·e-voting system breakdowns·*.

O/P_12 [un] The *·e-voting system·* SHALL be robust against power outage at the *·voting server·*, unexpected *·user·* activity, environmental effects (mechanical, electromagnetic, climatic, etc.) to the *·voting server·*, network problems, *·voter's client·* breakdowns and *·voting client·* breakdowns.

O/P_13 [tr] The *·voting server·* SHOULD not provide any information about the *·voting process·* except current state and the number of *·votes·* cast so far.

O/P_14 [se] The *·voting server·* SHALL delete any record of his *·voting process·* (other than the *·e-vote·* in the *·e-ballot box·*) from any memory of the *·voting server·*, when a *·voter·* completes the *·voting process·* (by *·casting·* his *·vote·* or cancelling).

O/P_15 [tr] The *·voting server·* SHALL be capable of producing comprehensive audit data.

O/P_16 [all] The *·voting server·* SHALL be capable of performing self-checks (including checking whether it is still available, the proper software is installed) and reporting about the results.

O/P_17 [all] The *·voting server·* SHOULD regularly perform automatic self-checks and report about the results.

O/P_18 [un] The *·e-voting system·* SHOULD be available during the whole *·polling period·*.

O/P_19 [fr] The *·voting client·* SHALL ensure that the ·voter's· *·selections·* are accurately represented in the *·e-vote·*.

O/P_20 [fr] The *·voting client·* SHALL ensure equality and accuracy of presentation of *·ballot·* options on any *·voter's client·*.

O/P_21 [fr] The *·voting clients·* SHOULD be compatible with any *·voter's client·*.

O/P_22 [di] The *·counting procedure·* SHALL accurately calculate results using the appropriate algorithm based on all (authorized) *·e-votes·* stored in the *·eballot box·* and only such *·e-votes·*.

O/P_23 [un] The *·voting server·* SHALL be capable of recording an adequate number of *·votes·*.

O/P_24 [fr] The *·voting device·* SHALL support an adequate number of voting options.

6. Related Work

The requirements defined in the paper are based on a list of requirements which one of the authors presented in [11] for Digital Recording Machines. This list is based on [7], [4], and [2], while it leaves Online-Voting specific requirements from [7] and [4] out.

The Council of Europe set up a working group in early 2003 to develop a set of standards for e-enabled voting that reflect member states' differing circumstances.

The standard was published in 2004 [4] and distinguishes between legal standards (covering the five election principles), procedural safeguards, operational standards, and technical requirements. It states the need for certification without giving any details about how such certification could be done.

The set of requirements for ”Online-Voting Systems for Non-parliamentary Elections” [7] (for networked polling-station elections) has been developed by the Physikalisch-Technische Bundesanstalt. It contains technical and organisational requirements and is based on an analysis of other available requirement catalogues.

The requirements are classified according to the different time intervals or classified as ”cross-sectional functions”. The catalogue does not describe the evaluation process or the evaluation depth.

Other relevant lists of requirements are the [6], [5], [3], [13], and [12].

7. Future Work

For one subset of security requirements we proposed to analyse the system in order to determine the maximum number of entities which need to be corrupted to violate one of the defined requirements. In future, the advantages and disadvantages of small and large k's are going to discussed to point out the consequences, eg, generally, a (k +1)-resilient system is better than a k-resilient system. However, k-resilience (w.r.t. a security property) is not an absolute measure. This would also help the *·e-voting system·* to decide whether it fits with their situation. Moreover, the entity itself is going to be split in different sup entities, like the software provider, the person responsible for configuration and hosting the *·voting server·* as well as the administrators.

Another open task is the definition of requirements where at the moment, only assumptions are made (see section 2). The next step is then the application of the above set of requirements using the described evaluation techniques to systems in use.

8. Conclusion

The list of security objectives in section 5 covers the requirements from the other available sets of requirements. There for even if we cannot guarantee to provide a complete list, we at least ensured to cover all requirements we found in the literature. Similarly, we only extend the to be applied evaluation techniques: In section 4 we explained that the core application of the Common Criteria, should be extended by the computation of (k_1 + ... + k_m **out of** n_1 + ... + n_m)^i-**resilient** values for particular requirements in order to take different intruder models and intruder capabilities into account and therefore to make the evaluation results more precisely, understandable and thereby more transparent.

It is worth to mention that we need to add at least the following three requirements:

- **Usab** The *·voting client·* SHALL be easy to be installed and use on the *·voter's client·*.
- **Organ** The *·responsible election authority·* SHALL explain why the evaluation results meet the trust model of the *·election·* environment the *·e-voting system·* is used in.
- **Assur** The *·testing authority·* SHALL evaluate the crypto-protocol in use.

although we did not consider usability, organisational and assurance requirements in the first place.

9. Acknowledgments

Thanks to Berry Schoenmakers for useful and helpful discussions about the requirements and evaluation techniques. Funding for Melanie Volkamer's research visit at Technical University Eindhoven is provided by the DAAD (German Academic Exchange Service).

References

[1] Common Criteria for Information Technology Security Evaluation, Version 3.1, 2006.

[2] Bundesministerium des Inneren. Verordnung ¨uber den Einsatz von Wahlgeräten bei Wahlen zum Deutschen Bundestag und der Abgeordneten des Europäischen Parlaments aus der Bundesrepublik Deutschland. Bundeswahlgeräteverordnung (BWahlGV), (Vom 03.09.1975 (BGBl S. 2459) zuletzt geändert am 20.04.1999 (BGBl I S. 749)), 20.04.1999.

[3] Crown Copyright. e-Voting Technical Security Requirements, 2001.

[4] Council of Europe. Legal, operational and technical standards for e-voting. Recommendation Rec(2004)11 adopted by the Committee of Ministers of the Council of Europe and explanatory memorandum. Council of Europe, Straßburg, 2004.

[5] Die Schweizer Bundeskanzlei. Verordnung über die politischen Rechte, 2006.

[6] Gesellschaft für Informatik e.V. Wahlen 2005. Technical report, GI, 2005.

[7] Volker Hertmann, Nils Meissner, and Dieter Richter. Online Voting Systems for Nonparliamentary Elections - Catalogue of Requirements. Technical report, Physikalisch-Technische Bundesanstalt Braunscheig/Berlin, 8.5.2004.

[8] Engelbert Hubbers, Bart Jacobs, and Wolter Pieters. Ries - internet voting in action. In COMPSAC (1), pages 417–424. IEEE Computer Society, 2005.

[9] Micromata GmbH. Online-Wahlen für Verbände und Vereine, 2005.

[10] Melanie Volkamer and Robert Krimmer. Ver-/misstrauen schaffende Maßnahme beim e-voting. In Christian Hochberger and Rüdiger Liskowsky, editors, INFORMATIK 2006- Informatik für Menschen, Volume P-93, Pages 418–425. Lecture Notes in Informatics (LNI), 2006.

[11] Melanie Volkamer and Margaret McGaley. Requirements and evaluation procedures for e-voting. In ARES, 2007 (forthcoming).

[12] Melanie Volkamer and Roland Vogt. Protection Profile - Central Requirements for Online Voting Systems. Technical report, German Research Center for Artificial Intelligence, 2007.

[13] VoteHere. Network Voting System Standards, 2002.