Examination of the Swiss Internet Voting System

Audit scope 2a (development process)

Follow-up audit (round 3)

2023-08-04 / v1.0FINAL

Work performed for:

Swiss Federal Chancellery
Political Rights Section
Federal Palace West Wing
3003 Bern

Contact information

SCRT SA
Rue du Sablon 4
1110 Morges
Switzerland

T: +41 21 802 64 01
E: info@scrt.ch

Authors and contributors

Antonio Fontes	Lead examiner

Version history

Version	Author	Date	Version
1.0Draft	Antonio Fontes	2023-04-17	Draft version
1.0Final	Antonio Fontes	2023-08-04	Final version

Management summary

Scope and objective of the examination

During the period August 2021 – December 2022, SCRT carried out two successive security audits of the Swiss Post's e-voting system against a subset of requirements set forth in scope 2a (development processes) of the Federal Chancellery's ordinance on e-voting .

The first round of examination (round 1) resulted in the reporting of six non-compliances (findings), resulting in a proposition of eleven recommendations.

At the conclusion of the second round of examination (round 2), seven of the initial eleven recommendations were still either pending implementation or being implemented, thus leaving four ordinance requirements (findings) unsatisfied.

The objective of this third round of examination was to follow-up on the findings and recommendations that remained following the second round of examination.

Methodology

The examiners looked for evidence of effort to comply with the criteria not yet fulfilled during the first two examination rounds. They conducted reviews of the personnel in charge of the e-voting system's development and development processes (interviews) and reviewed the relating evidence (e.g., policies, procedures and processes, specifications, documentation, reports, etc.).

The third round of examination was performed during April 2023.

Results

As a result of this third-round examination, the examiners confirm the implementation of six out of seven recommendations that remained after the round 2 examination. One (1) final ordinance requirement remains partially unsatisfied:

- Requirement 17.2 (test coverage): tools to perform automated and semi-automated runtime security testing are still in the process of being deployed and integrated in the e-voting system development lifecycle.

The severity of this non-compliance is still classified as *moderate*. This is explained by the presence of multiple alternative measures still in place to detect errors or vulnerabilities in the e-voting system at runtime (i.e., user/voter reporting, bug bounty program, and penetration tests upon each release).

Final note

The examiners note the significant effort invested by Swiss Post to address and resolve almost all findings and recommendations issued by the examiners, in particular the specification and deployment of new processes, and the improvement of existing ones, within the e-voting project systems engineering activity.

The recommendation associated with the last incomplete finding requires the adoption and the integration of new tooling (automated runtime application testing software) into the e-voting system development and release pipeline. While the vetting and selection process has been completed, selected employees are being trained to use the tool both adequately and efficiently prior to its final integration into the project.

All things being equal otherwise and assuming the remaining recommendation will be implemented within reasonable delay, the examiners consider the outcome of the examination to be generally positive.

The examiners conclude this summary by thanking Swiss Post again, and more particularly those who have been personally involved in this follow-up audit, for their cooperation during the interview and workshop sessions, and for the transparency demonstrated throughout the examination.

Table of content

1 Context

1. Electronic voting (hereafter referred to as: e-voting) was introduced in Switzerland through multiple pilot schemes from 2004 onwards. A total of 15 cantons made e-voting possible in over 300 trials, until early 2019. Two implementations were available: the system made by the canton of Geneva and the system operated by the Swiss Post (initially developed by Scytl). In June 2019, the canton of Geneva announced the withdrawal of its e-voting system with immediate effect. It was followed in July of the same year by the announcement by Swiss Post of the withdrawal of its e-voting system from operation to focus on improving the solution. Since then, e-voting is not possible in Switzerland.

2. In June 2019, the Swiss Federal Chancellery (hereafter: Federal Chancellery) was commissioned by the Federal Council to redesign a new trial phase, using "e-voting systems, which are fully verifiable" [1]. This redesign of the trial phase focuses on four objectives:

1. Further development of the e-voting systems
2. Effective controls and monitoring
3. Increased transparency and trust
4. Stronger connection with the scientific community

3. A taskforce was set up to make proposals for the future of internet voting. To that end, the Federal Chancellery invited experts from academia and industry to engage in a broad dialogue on internet voting in Switzerland. After this dialog, the Federal Chancellery and the cantons published a final report on the redesign and relaunch of internet voting trials, with a catalogue of measures [2].

4. The Federal Council took note of the final report and commissioned the Federal Chancellery to amend the legal bases of the Confederation. In April 2021, the Federal Council opened a consultation procedure on the amendment to the legal bases, which was drafted by the Federal Chancellery. The redesign includes both a partial revision of the Ordinance on Political Rights (PoRo) [3] and a complete revision of the Federal Chancellery Ordinance on Electronic Voting ("VEIeS", or "OEV") [4]. The OEV specifies, among others, the requirements for authorising electronic voting, including the technical and administrative controls for approving an e-voting system[1].

5. The Federal Chancellery issued an audit concept for the examination of Swiss internet voting systems [5] defining the foundations for assessing the compliance of electronic voting systems with the draft OEV and its annex [6], as per chapter 26 of the annex of the draft OEV, and for obtaining recommendations for improvements. The audit concept was subsequently

[1] The criteria for the examination reported in this document is built on a subset of these controls.

revised to version 1.5 in September 2022 [7], without regressions or significant changes to the scope 2a (development processes).

6. SCRT was mandated by the Federal Chancellery to assess the compliance of the Swiss Post's revamped e-voting system against some of the requirements of the draft OEV. The present report focusses on the examination of the perimeter defined as follows in the audit concept: *Scope 2a - Development process*.

7. In May 2022, the Federal Council enacted the partially revised Ordinance on Political Rights (PoRo) [8], which became applicable from Jul. 1st 2022. The totally revised Federal Chancellery Ordinance on Electronic Voting (OEV) [9] came into force on the same date.

8. A second assessment was conducted in mid-September 2022 to follow-up on the findings raised in the initial audit report [10].

9. In March 2023, the Federal Council granted three Swiss cantons basic licences for resuming trials with online voting in federal votes with a limited part of the electorate. The licences are valid until May 2025 and a first popular vote involving the e-voting system was held in June 2023 [34].

10. A third assessment was conducted in April 2023 to follow-up on the findings raised in the second audit report [11].

2 Methodology

11. The methodology to assess the requirements remains unchanged from the first- and second-round examinations. Readers of the round 1 [10] and round 2 [11] examination reports are invited to skip chapters 2 and 3, and jump to chapter 4 (Follow-up examination results).

2.1 General process

12. The examination was based on SCRT's information systems audit methodology. The process specifies four-phases depicted in the figure below:

Figure 1 - Process

2.2 Collection of evidence

13. As a general principle, the examiners aimed at acquiring two types of evidence for each requirement. Types of evidence included documents (e.g., documentation, test reports, written instructions, etc.), statements (e.g., obtained during plenary sessions or during interviews), and demonstrations (e.g., tools, scripts, configurations or process material shown during interviews).

14. Part of the examination included reviewing documents classified as confidential by Swiss Post and thus not released to the public. Motives for not disclosing these documents to the

public included either or both the a) preservation of the confidentiality of business processes deployed at the organisation level and which may confer Swiss Post a competitive advantage on other actors, and b) the preservation of confidentiality of operational data (e.g., risk control, infrastructure operations, etc.). Swiss Post confirmed to the examiners that these documents remain accessible to the Cantons.

15. Unless specified otherwise, written evidence collected and reviewed during the examination is referred in the bibliography of this report, with public links whenever possible. Some sources, which were not made physically available to the examiners but shown on-screen during live interviews with the examinees, are cited without reference.

16. Swiss Post provided the examiners an internal document mapping each Federal Chancellery requirement with one or more corresponding documented evidences [12].

2.3 Requirements

17. The examiners specified three states for the requirements, as follows:

- *Pass* - The proposed evidence allows the examiners to consider that the requirement is met.
- *Partial miss or P.Miss* – The proposed evidence allows the examiners to conclude that the requirement is partially met.
- *Miss* - The proposed evidence either allows the examiners to conclude that the requirement is not met (explicit miss), or that there is insufficient evidence to reach another conclusion (implicit miss).

2.4 Findings

Definition

18. During the first-round examination, the examiners raised a finding when evidence provided by the examinee did not provide satisfying assurance that the requirement was met (implicit miss) or when the evidence provided explicitly indicated that the requirement was not met, or partially met (explicit miss).

19. New findings were not identified during the second-round examination. Any reference to findings in this document refers to findings already reported in the first-round report.

Severity of findings

20. The examiners specified three severity levels, as follows:

- *High severity* - The finding identifies a failure to produce evidence of satisfying a requirement.

- *Moderate severity* - The finding identifies a partial failure to produce evidence of satisfying a requirement.
- *Low severity* - The finding refers to a significant opportunity for improvement or optimisation.

21. Readers should note that the severity indicated in this report only reflects the opinion of the examiners and could be subject to re-evaluation by relevant parties.

Recommendations

22. For each finding, one or more recommendations were issued during the first-round examination. A finding is considered as resolved or closed once all its associated recommendations are implemented.

2.5 Assumptions

Trustworthiness of statements

23. The examiners assume that the examinees were completely honest and transparent when providing answers to the examiners' assessment questions. Although several proofs of testing were shown, no observation of the actual implementation of the OEV's requirements within the e-voting system was carried out to verify the accuracy of the examinees' statements.

Trustworthiness of security measures

24. The examiners assume that the security measures described in the documents provided as evidence in the context of the present examination are implemented and are effective. No verification of the actual implementation effectiveness of the OEV's requirements within the e-voting system (e.g., security testing, vulnerability assessment, penetration test, etc.) was carried out within the scope of this examination to verify the accuracy of the statements made in the security documents.

Non-regression

25. The examiners assumed that the Federal Chancellery requirements, for which satisfying evidence of implementation was provided by Swiss Post during the first and second rounds of examination, were still met during the third round of examination.

3 Examination criteria

26. The audit concept for the e-voting examination [13] specifies several assessment scopes (e.g., cryptographic protocol, software, infrastructure, etc.) , each encompassing a subset of the requirements specified in the Federal Chancellery ordinance on electronic voting published in April 2022 [9].

27. This chapter enumerates the requirements for the scope 2a ("assess the development processes").

3.1 Effect of the revised ordinance on initial results (impact assessment)

28. A revised ordinance on electronic voting entered into force on 1st July 2022 [9]. Changes between the revised version and the version used during the first-round examination are summarized as follows[2]:

- Requirement 8.12 (transparent communication of known flaws): renamed to 8.13.
- Requirement 24.1.19 (configuration list): requires an additional information item labelled as "commit history".

29. The examiners assessed the impact of the changes above and concluded as follows:

- Regarding requirement 8.13: impact is considered null.
- Regarding requirement 24.1.19: the configuration management system used by the examinee to produce the artifacts can produce a history of commits to the source code (including the source code of the CI/CD pipeline itself), along with their author, as part of its feature set. Impact is also considered null.

3.2 Third round examination criteria

30. No significant changes were made to the criteria (e-voting concept [7]) for the third round examination of the software development process. Consequently, the examiners limited the criteria for the third examination to the requirements that had not been met, or partially met, during the second-round examination (findings and associated recommendations from the second-round examination report).

31. The illustration below summarizes this situation:

[2] This list shows changes to the requirements that are part of the assessment scope 2a. Changes to the ordinance that affected other examination scopes are not listed here.

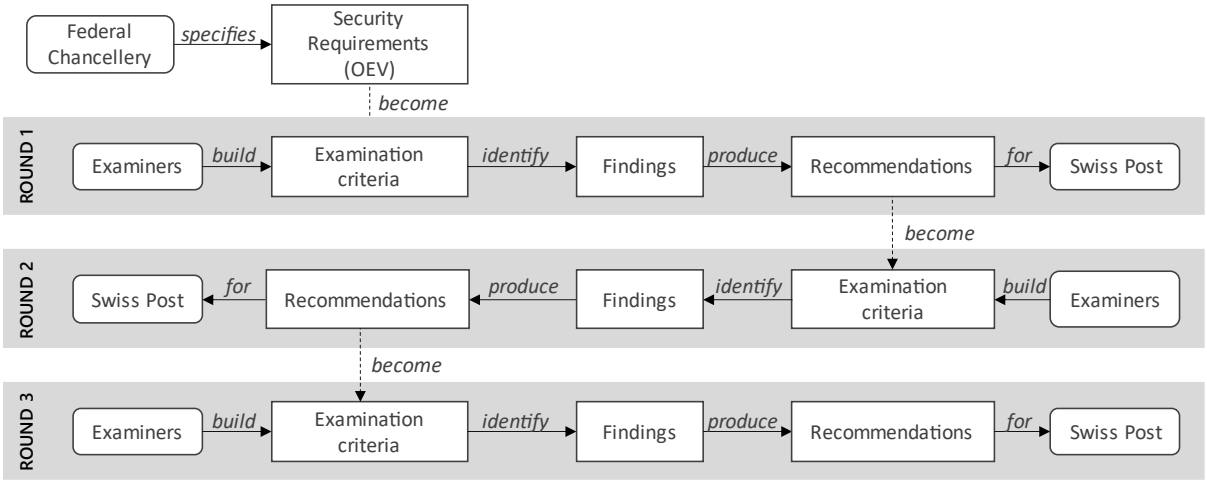

Figure 2 – Examination criteria for the third-round examination.

3.3 Federal Chancellery requirements

32. The following tables enumerate the 17 requirements that constitute the criteria for scope 2a ("assess the development process") of the audit concept for the examination of the Swiss internet voting systems [13], grouped by topic[3].

Development process and lifecycle requirements

Key	Requirement
24.1.1	A life cycle model is defined. The life cycle model: » is used for the development and maintenance of the software (a); » provides for the necessary controls during the development and maintenance of the software (b); » is documented (c).
24.1.2	A list must be made of the development tools used and configuration options chosen for the use of each development tool.
24.1.3	The documentation for the development tools includes: » a definition of the development tool (a); » a description of all conventions and directives used in the implementation of the development tool (b); » a clear description of the significance of all configuration options for using the development tool (c).
24.1.4	The implementation standards to be applied must be specified.

Table 1 - E-voting requirements: lifecycle

[3] Topic names and the regrouping of requirements were chosen by the examiners and may not necessarily reflect the Federal Chancellery's vision.

Software security documentation requirements

Key	Requirement
24.1.20	Software development security documentation includes: » a description of the physical, procedural, personnel, and other security measures necessary to protect and ensure the integrity of the design and implementation of the software in its development environment (a); » evidence that the security measures provide the necessary level of protection to preserve the integrity of the software (b).

Table 2 - E-voting requirements: software security documentation

Quality assurance requirements

Key	Requirement
24.5	Regular and objective checks are carried out to ensure that the processes carried out and the associated work products comply with the description of the processes, standards and procedures to be implemented (a). Deviations are followed up until they are corrected (b).

Table 3 - E-voting requirements: quality assurance

Configuration management system requirements

Key	Requirement
24.1.14	The software is provided with a unique identification.
24.1.15	The configuration management documentation includes: » a description of how configuration items are identified (a); » a configuration management plan describing how the configuration management system will be used in the development of the software and the procedures that will be followed for the adoption of changes or new elements (b); » evidence that the procedures for adoption provide for adequate review of changes for all configuration items (c).
24.1.16	The configuration management system: » uniquely identifies all configuration items (a); » provides automated measures to ensure that only authorised changes are made to configuration items (b); » supports the development of the software through automated procedures (c); » ensures that the person responsible for accepting the configuration item is not the same person who developed it (d); » identifies the configuration items that make up the security functions (e); » supports verification of all changes to the software using automated procedures, including logging of the author and the date and time of the change (f); » provides an automated method for identifying any configuration items that are affected by a change to a particular configuration item (g);

	» can identify the version of the source code on the basis of which the software is generated (h).
24.1.17	All configuration items are inventoried in the configuration management system.
24.1.18	The configuration management system is used in accordance with the configuration management plan.
24.1.19	A configuration list is created that contains the following items: » the software, » evidence of the checks required to ensure security compliance, » the parts that make up the software, » the source code, » the commit history[4], » reports on security flaws and on the status of their correction (a). For each element relevant to security functions, the developer is named (b). Each element is uniquely identified (c).

Table 4 - E-voting requirements: configuration management system

Testing requirements

Key	Requirement
17.1	The functions relevant to the security of the system (security functions) are tested. The tests are documented with test plans and expected and actual test results. (a) The test plan (b): » specifies the tests to be performed; » describes the scenarios for each test, including any dependencies on the results of other tests. The expected results must show the results that are expected if the test is successfully executed. (c) The actual results must be consistent with expected results. (d)
17.2	An analysis must be made of the test coverage. This includes evidence that: » the tests defined in the test documentation match the functional specifications of the interfaces (a); » all interfaces have been fully tested (b).
17.3	An analysis must be made of the depth of testing. This includes evidence that: » the tests defined in the test documentation match the subsystems related to security functions and modules that play a role in ensuring security (a); » all subsystems related to the security functions mentioned in the specifications have been tested (b); » all modules that play a role in ensuring security have been tested (c).
25.13.3	The integration tests cover all modules.

[4] The requirement for a commit history was added in the revised ordinance.

25.13.4	The software tests cover all modules.

Table 5 - E-voting requirements: testing

Transparency requirement

Key	Requirement
8.13[5]	Known flaws and the need for action associated with them are communicated transparently. **: identified as 8.12 in the round 1 examination criteria.*

Table 6 - E-voting requirements: transparency

Systematic correction of flaws requirements

Key	Requirement
24.4.1	Processes are defined for the correction of flaws. The processes include: » documentation of specific aspects, in particular with regard to the traceability of flaws for all versions of the software, and of the methods used to ensure that system users have information on flaws, corrections and possible corrective actions (a); » the obligation to describe the nature and impact of all security flaws, information on the status of work to find a solution and the corrective measures adopted (b); » a description of how system users can make reports and enquiries about suspected flaws in the software known to the software developers (c); » a procedure requiring a timely response and automatic dispatch of security flaw reports and appropriate corrective actions to registered system users who may be affected by the flaw (d).
24.4.2	A process is defined for handling reported flaws (a). This process ensures that all reported and confirmed flaws are corrected and that the procedures for correction are communicated to system users (b). It provides for arrangements to ensure that the correction of security flaws does not give rise to new security flaws (c).
24.4.3	Policies must be defined for the reporting and correction of flaws. These include: » instructions on how system users can report suspected security flaws to the developer (a); » instructions on how system users can register with the developer to receive reports of security flaws and the corrections (b); » details of specific contact points for all reports and inquiries on security issues concerning the software (c).

Table 7 - E-voting requirements: systematic correction of flaws

[5] Identified as requirement key 8.12 in the round 1 examination report, the requirement remains unchanged.

3.4 Additional criteria: secure systems development lifecycle

33. A central part of the work consisted in evaluating the development process put in place by Swiss Post for its e-voting system. To limit potentially ambiguous interpretations of what could qualify as the OEV requirement "life cycle model, which provides for the necessary controls during maintenance and development of the software" [6, p. 33], the examiners derived an interpretation of "necessary controls" based on the following references:

- Security software lifecycle requirements and assessment procedures (v1.0), PCI [14],
- Application software security controls, Critical security controls (v.8), CIS [15],
- Fundamental practices for secure software development (third edition), SAFECode [16].
- SAMM - Software assurance maturity model assessment tool (v1.5), OWASP [17].

34. The characterisation of "necessary controls" is summarised as follows:

Organisational measures:

- The organisation has appointed a security champion within each development team, which, among others, acts as a security liaison and leader between the development team and the organisation's security structures,
- The organisation established interfaces between the development team and the organisation's information security structures and with external security advisor(s),
- The organisation established interfaces with its incident response structure,
- The organisation ensures that personnel involved in the design, construction or attestation of the system attended role-based security awareness and training.

Operational enablement measures:

- A catalogue of threats, with their respective controls or countermeasures, and status (e.g., mitigated, not mitigated, etc.), is documented and maintained,
- Security requirements are documented,
- Secure design principles or secure architecture baseline requirements are documented and integrated in the development process,
- Changes to the system are subject to a threat assessment (e.g., threat modelling, abuse cases, attacker stories, etc.) aimed among others at identifying potential and relevant threats and identifying appropriate countermeasures,
- Coding guidelines, or equivalent, are documented. In particular, they propose standardised responses to well-known causes of risk and error (e.g., input canonicalization and validation, output encoding, command interpreter query parameterisation, filesystem access, database access, protected storage of sensitive data or secrets, etc.),
- High-risk code is identified as such and subject to extended review (e.g., manual review or testing),

- Vulnerability management is integrated and performed throughout the entire development process with the support of adequate tools and processes,
- Source code, including all relevant adjacent artifacts, released to customers is centralised, versioned, and protected from unauthorised access.

Attestation measures:

- Change requests are subject to standardised or routine security verification (e.g., security checklist in definition of ready).
- New code is tested for well-known vulnerabilities and errors (aka, source code review, static analysis, etc.) as well as existing code (to mitigate regressions) prior to release.
- Third-party components are vetted against well-known threats prior to being integrated into the system.
- Third-party components are inventoried and monitored for known issues or vulnerabilities.
- The runtime is tested for well-known vulnerabilities and errors (e.g., dynamic/runtime application security testing) prior to release.
- The security of the final system, both in its entirety and its individual high-risk components, is regularly tested by independent actors through adequate methods (e.g., external penetration testing, bug bounty, 3rd party expert review, etc.).
- Releases and all associated artifacts are certified, and their authenticity can be independently verified (e.g., code or binary signing, etc.),

3.5 Additional criteria: inclusion of third-party components

35. Due to the extensive use of third-party components in the e-voting system, the examiners also derived an interpretation of "necessary controls" for the use and inclusion of third-party components as part of the software engineering process. The following reference was used to derive requirements:

- Managing security risks inherent in the use of third-party component, SAFECode [18].

36. The characterisation of "necessary controls", in the context of third-party components, is summarised as follows:

Organisational measures:

- The organisation conducts a standardised risk assessment prior to integrating a third-party component into the system.

Operational enablement measures:

- Threats derived from the use and inclusion of third-party components in the software application are identified and documented, and adequate countermeasures or controls implemented wherever relevant and/or necessary.

Attestation measures:

- Third-party components are tested for known vulnerabilities or malicious activity, both prior to their inclusion in the system and after (monitoring).

4 Follow-up examination results (November 2022)

37. This chapter summarizes the observations and conclusions reached by the examiners following the second review of the remediation steps taken by Swiss Post to remediate the findings identified during the initial examination.

38. For each finding, the following content is included:

- The description of the finding, as initially reported during the initial examination.
- Updates on the status of findings following the second-round examination.
- Results from the third-round follow-up examination. This only applies to findings that were marked as *in progress* or *pending* following the second-round examination.

4.1 F-01 Insufficient integration of security in the software development lifecycle

Status prior to follow-up examination (round 1 and round 2)

Key	F-01
Title	Insufficient integration of security in the software development lifecycle
Requirement ID(s)	24.1.1
Requirement(s)	A life cycle model is defined. The life cycle model: » is used for the development and maintenance of the software (a); » **provides for the necessary controls during the development and maintenance of the software (b);** » is documented (c).
Severity	Moderate
Rationale	The examiners noted the accrued effort invested by the examinee to build a secure e-voting cryptographic protocol. However, the same level of effort was not observed equivalently on the overall system's development lifecycle of the solution [10, Para. 25]. In particular, the examiners noted the following: » the examinee invested a large amount of effort, controls and activities to secure the solution, and more particularly the e-voting protocol, but remained unable to position itself in terms of excellency and maturity on what is generally considered state-of-the-art in terms of secure systems engineering [10, Para. 25];

	» the security testing strategy focused on post-code flaw/vulnerability detection measures (e.g., code review, public reviews, bug bounties, penetration testing by third parties, etc.) while [10, Para. 26], thus potentially failing to identify threats, flaws and vulnerabilities at earlier stages of the development lifecycle; » a lack of formal training, in particular amongst architects and developers of the solution [10, Para. 29];
Recommendations	R-01 - Formalise the integration of security development lifecycle guidance or best practices into the e-voting system's development process. R-02 - Ensure e-voting system personnel and stakeholders have received adequate role-based training on secure systems engineering. R-03 - Integrate threat modelling, or equivalent, in early stages of the development process. R-04 - Establish a baseline set of security principles or rules for each phase of the development lifecycle (e.g., requirements, architecture and/or design, coding, testing, build, deployment, etc.).
Round 2 examination results	R-01 – status updated (completed) R-02 – status updated (completed) **R-03 – status updated (in progress)** R-04 – status updated (completed)

Table 8 - Finding F-01 (insufficient integration of security in the software development lifecycle)

Follow-up examination (round 3 – April 2023)

Evidence

39. New or updated evidence was provided or shown:

- » Document: Threat modelling [19]
- » Document: 2023-03-23 threat modelling session report [20]
- » Document: 2023-02-02 threat modelling session report [21]
- » Live personnel interview

Assessment

40. On the third recommendation (R-03):

- » At the organization level, the examinee is still deploying threat modelling across its system engineering activity.
- » Within the e-voting project, the process has reached continuous improvement stage. Appointed security champions are responsible for establishing reference threat models for each major component of the e-voting system along with the architecture and development teams. They are also responsible for conducting threat modelling workshops upon request from an architect or developer. Workshops occur monthly and involve at least one security champion and a rotating subset of architects and developers. Findings are turned into work items in the issue tracking and project

management platform (Jira) for further processing, which may involve investigating and/or mitigating a potential issue.

- The threat identification approach is still in its early maturity stage (e.g., use of system-centric threat identification methods such as STRIDE[22]) and being tuned to better align with the organization's constraints and culture (e.g., agile threat modelling activity). Opportunities for improvement were discussed during the interviews. These included the use of more exhaustive threat identification and enumeration techniques, such as STRIDE+LM [23] and LINDDUN [24], the use of threat prioritization methods, and techniques for industrializing the overall threat modelling process (e.g., threat modelling tools, threat modelling as code, threat catalogues and protection profiles for well-known patterns, etc.).
- The examiners were provided with reports from past threat modelling sessions, summarizing the overall session stages and deliverables produced. The existence of findings and their integration in the change management system could be verified.
- In terms of the e-voting system's threat model, the examiners observed that the e-voting project now operates with its own threat model. As expected, the threat model includes the threat catalogue specified by the Federal Chancellery in its e-voting ordinance [9, App. 13].

41. Based on the evidence shown to the examiners, the implementation of recommendation R-03 is considered satisfying and marked as completed.

4.2 F-02 Conflicting/ambiguous attribution of security responsibilities

42. Satisfying evidence for the implementation of the recommendations derived from the finding F-02 was found during the second round of examination. For additional information, see conclusions of the second-round examination report [11].

4.3 F-03 Insufficient protection measures against malicious third-party components

Status prior to follow-up examination (round 1 and round 2)

Key	F-03
Title	Insufficient protection measures against malicious third-party components
Requirement ID(s)	24.1.1[6]
Requirement(s)	A life cycle model is defined. The life cycle model: » is used for the development and maintenance of the software (a); » **provides for the necessary controls during the development and maintenance of the software (b);** » **is documented (c).**
Severity	Moderate
Rationale	The examiners noted the following: » The examinee deployed a software composition analysis (SCA) tool, which allowed the team to be alerted when a third-party component (TPC) is publicly identified as vulnerable or compromised [10, Para. 43]. » However, other classes of threats, such as TPCs that have been either compromised or built maliciously without knowledge of the community or the vendor, and TPCs that contain unreported vulnerabilities, were still lacking mitigating controls [10, Para. 43], [10, Para. 44]. » The examinee had not yet formalized the threat model on which decisions to adopt or reject TPCs in the e-voting platform were or would be based.
Recommendations	R-06 - Establish a reference threat model for the use of third-party components in the e-voting system, maintained with the status of implementation of chosen controls and countermeasures. R-07 - Establish a security vetting process for the selection of new third-party components, and the review of existing ones, embedded in the e-voting system.
Round 2 examination results	**R-06 – status unchanged (missing or incomplete)** **R-07 – status updated (in progress)**

Table 9 - Finding F-03 (Insufficient protection measures against malicious third-party components)

[6] A typo in the first-round examination report incorrectly mentioned requirement 24.1.15 (configuration management system documentation) in addition to requirement 24.1.1.

Follow-up examination (round 3 – April 2023)

Evidence

43. The following new or updated evidence was provided or shown:

- Document: evaluation criteria for libraires & frameworks [25]
- Document: third-party request report [26]
- Document: reference threat model for third-party components [27]
- Document: procedure for the approval of third-party dependencies and history of approvals [28]
- Live personnel interview

Assessment

44. On the first recommendation (R-06):

- The examinee has formalized a reference threat model for the use of third-party components, including third-party libraries and APIs. At time of examination, the reference threat model specifies a list of threat scenarios, for which mitigating measures or countermeasures are specified. These threats include:
  - Attacks on poorly built components (e.g., bad coding habits and errors),
  - Use of unauthorized components (i.e., incompatible licensing),
  - Developer account takeover (i.e., developers with poor security hygiene),
  - Poorly maintained or unmaintained components,
  - Components with an embedded logic bomb or backdoor.
- The threat model and the associated mitigations are used as reference material when requests for adding a new third-party component into the e-voting system are processed by the architecture board.

45. Based on the evidence shown to the examiners, the implementation of recommendation R-06 is considered satisfying and marked as completed.

46. On the second recommendation (R-07):

- The insertion of a new library or third-party component into the e-voting project is governed by an approval process led by organization-level architects. Developers and architects of the e-voting project are not given direct access from their development stations to community-maintained repositories (e.g., Maven Central) but to an organization-wide repository acting as an intermediary authority.
- To be added to the organization's repository, a motivated request (online form) must be submitted to an architecture review board who can add components to the organization-level repository.
- The criteria through which architects will approve or reject a component has been formalized and includes a set of both external (e.g., meta-data, attributes, activity, known vulnerabilities, etc.) and internal (e.g., source code) verifications. Details on the data used to reach a decision and the outcome of the verification are logged in a

third-party dependency request report. The inability to reproduce results upon subsequent approval requests was qualified as a weakness in the previous round of examination, this issue appears to have been resolved.

- The threat of dependency confusion attacks (DCAs) was discussed and is part of the threat model. DCAs originating from external repositories are mainly mitigated by developer-initiated library updates: versions are hard-coded client-side and updates from third-party repositories are not pushed into the project unless they have been formally approved. Additionally, repositories are configured to reject updates to internal modules from upstream repositories. DCAs originating from an internal attack (e.g., a disgruntled Swiss Post employee pushes a fraudulent dependency into the internal repository) are also part of the threat model and the mitigation strategy involves comparing internal builds and artifacts with those generated from control builds generated outside the trusted perimeter (several scenarios could result in a mismatch, amongst which a compromised build environment).
- Approval decisions are issued with an expiration in the case of e-voting components to ensure that they get renewed regularly.

47. Based on the evidence shown to the examiners, the implementation of recommendation R-07 is considered satisfying and marked as completed.

4.4 F-04 Insufficient security documentation

Status prior to follow-up examination (round 1 and round 2)

Key	F-04
Title	Insufficient security documentation
Requirement ID(s)	24.1.20
Requirement(s)	Software development security documentation includes: » **a description of the physical, procedural, personnel, and other security measures necessary to protect and ensure the integrity of the design and implementation of the software in its development environment (a);** » **evidence that the security measures provide the necessary level of protection to preserve the integrity of the software (b).**
Severity	Moderate
Rationale	The examiners noted the following: » A large amount of documentation had been made available, including documented procedures, architecture and design documents, features documentations and protocol specifications. » However, the examiners noted that, although the e-voting cryptographic protocol had been extensively documented, reviewed and tested, many other security-relevant aspects of the e-voting system had not yet been documented. In particular, the examiners had noted

	the absence of threat models, baseline architecture and design principles, coding guidelines, security testing procedures, code attestation procedures, and moreover, the absence of a central document that would tell third-party how the e-voting system, in general, is secured[7].
Recommendations	R-08 - Establish a central document that describes how security assurance in the e-voting system is obtained (e.g., e-voting security whitepaper).
Round 2 examination results	**R-08 – status updated (in progress)**

Table 10 - Finding F-04 (Insufficient security documentation)

Follow-up examination (round 3 – April 2023)

Evidence

48. The following new or updated evidence was provided or shown:

- Document: Security whitepaper [29]
- Live personnel interview

Assessment

49. On the recommendation (R-08):

- Following the second-round examination, the examiners had observed an incomplete coverage of topics deemed essential in a security whitepaper of the e-voting system, as it almost exclusively focused on describing the e-voting system development process, using the OWASP's software assurance maturity model (SAMM) [30] as a reference for the content. A second concern was raised about the exclusive focus of the whitepaper on content produced by the e-voting development team, which failed to reflect the overall security strategy put in place by Swiss Post. A third concern was raised on the risk of attempting to centralize both deeply technical (e.g., cryptographic protocols) and high-level (security principles) in a single document.
- A new version of the whitepaper was produced, which now addresses the following concerns:
  - Scope of the e-voting system security whitepaper,
  - Summary of the cryptographic protocol and pointers to detailed specifications,
  - Summary of data protection core principles and organizational measures,
  - Summary of the public scrutiny concept (community program, including the bug bounty and published documentation and code, penetration tests and public audits),

[7] Here, the examiners make (and made) a distinction between the security of the e-voting system (the sum of all things put together, in the eye of the user) and the security of the e-voting protocol itself, which are considered two distinct, although highly interconnected, concepts.

  - Summary of the secure development lifecycle (selected assurance model, separation of duties and 4-eyes principle, toolchain, in-house development concept, threat modelling approach and security champions program),
  - Summary of the network security strategy (including the multiparty deployment approach),
  - Summary of the operational security strategy (trusted builds and deployment, security scanning, staff reviews and vetting),
  - Summary of infrastructure security.
- » The document centralizes high-level information on several core topics pertaining to the security of the e-voting system and proposes pointers (links) to external documents when deeper technical content is needed or expected (e.g., links to the specifications of the various security protocols, detailed architecture documents, etc.).
- » While the document has not yet reached a completion level that the examiners would strictly qualify as complete (e.g., incomplete sections, some sections may benefit from proofreading), this requirement was aimed at ensuring that the foundations for establishing an e-voting security whitepaper were met as to facilitate the addition of new or revised content on a regular basis.

50. Based on the evidence shown to the examiners, the implementation of recommendation R-08 is considered satisfying and marked as completed.

4.5 F-05 Insufficient quality control over security attestation operations

Status prior to follow-up examination (round 1 and round 2)

Key	F-05
Title	Insufficient quality control over security attestation operations
Requirement ID(s)	24.5[8]
Requirement(s)	**Regular and objective checks are carried out to ensure that the processes carried out and the associated work products comply with the description of the processes, standards and procedures to be implemented (a).** » Deviations are followed up until they are corrected (b).
Severity	Moderate
Rationale	The examiners noted the following:

[8] A typo in the first-round examination report incorrectly mentioned requirement 24.1.15 (configuration management system documentation) in addition to requirement 24.5 (quality assurance).

	» The examinee has put in place a robust quality assurance system, which ensures most, if not all, its operations and procedures are documented, and regularly improved. » It has implemented a large set of controls, including tools and processes, to produce security attestations and overall trust over the e-voting system. These include, among others, source code review tools, software composition analysis tools, public audits, external penetration testing, bug bounties, etc. » While these tools and controls are aimed at producing assurance about the security of the e-voting system, the effectiveness and correctness of operation of these tools was not governed by the quality assurance process in place.
Recommendations	R-09 - Establish procedures to confirm, review and improve the correct operation of security attestation measures, in particular automated measures.
Round 2 examination results	**R-09 – status unchanged (missing or incomplete)**

Table 11 - Finding F-05 (Insufficient quality control over security attestation operations)

Follow-up examination (round 3 – April 2023)

Evidence

51. The following new or updated evidence was provided or shown:

- » Document: testing of application security testing tools [31]
- » Live personnel interview

Assessment

52. On the recommendation (R-09):

- » The examiners observed the implementation of first measures and processes to assess the correct operation of its security attestation strategy.
- » More particularly, the e-voting development team started by establishing measures to assess and verify the completeness of coverage of the tools (e.g., static source code analysis, third-party component vulnerability scanning) used to automate security verifications[9]. The approach involved injecting canaries (artefacts specifically designed to trigger the activation of security alarms whenever they are detected) into various locations of the e-voting system (e.g., injecting vulnerable source code, injecting

[9] In this context, the examiners encouraged Swiss Post to assess the completeness of coverage of its security scanning tools by verifying, first, that the entire source code and all its dependencies were effectively scanned for vulnerabilities and errors (i.e., correctness of project scope definitions, no code left unchecked), and second, that the vulnerabilities for which Swiss Posts expects a high detection accuracy were indeed detected as such (i.e., correctness of scanning profiles, no false negatives).

known vulnerable third-party components, etc.) to confirm the correct execution and coverage of its security scanning tools (e.g., Fortify, X-Ray), and included comparing the results of its tools run internally with the results of external tools (e.g., comparing the results of an external dependency checker with the results of the internal dependency checker).

- A separate process ensures that the canaries are removed from release builds.
- Following completion of the first iteration of this process, the e-voting development team discovered that one of its security scanning tools could not verify certain portions of its source code and this could be attributed to a defect in the product. The team contacted the editor and filed a bug report, which was under review at time of the examination.

53. Based on the evidence shown to the examiners, the implementation of recommendation R-09 is considered satisfying and marked as completed.

Additional remarks

54. The requirement aimed at ensuring that the foundations for a continuous quality control and improvement process were met. Regular assessments and improvements of the quality control strategy are expected and thus highly recommended.

In the context of security attestation operations, this would likely consist in regularly reviewing the efficiency of the methods, tools and processes selected to verify the security of the e-voting system. For example, Swiss Post demonstrated to the examiners its ability to devise a quality control strategy to assess the completeness and correctness of its automated security scanning tools. The examiners hope that Swiss Post will not only extend these verifications to its entire security scanning tools portfolio, but also devise other strategies to also cover security attestation methods and processes (e.g., quality control of the threat modelling process, the design security approval process, and the third-party library approval process, etc.).

4.6 F-06 Insufficient security testing and attestation

Status prior to follow-up examination (round 1 and round 2)

Key	F-06
Title	Insufficient security testing and attestation
Requirement ID(s)	17.2
Requirement(s)	An analysis must be made of the test coverage. This includes evidence that: » the tests defined in the test documentation match the functional specifications of the interfaces (a); » **all interfaces have been fully tested (b).**
Severity	Moderate
Rationale	The examiners noted the following: » The examinee demonstrated strong evidence of security assurance along the entire lifecycle of the e-voting protocol. » However, the same coverage could not be observed for other areas of the e-voting system, especially parts designated as untrusted components per the e-voting protocol specification. » In particular, the examiners noted limited security assessment activities prior to entering the development (coding) phase and a lack of runtime security testing (prior to deployment).
Recommendations	R.10 - Establish procedures to review and/or validate design proposals generated in response to change requests, prior to entering the coding phase. R.11 - Establish or improve runtime/dynamic application security testing procedures executed prior to release. For relevant components, extend these procedures with additional fuzz testing.
Round 2 examination results	**R-10 – status updated (in progress)** **R-11 – status updated (in progress)**

Table 12 - Finding F-06 (Insufficient security testing and attestation)

Follow-up examination (round 3 – April 2023)

Evidence

55. The following new or updated evidence was provided or shown:

- » Document: Security checklist for new work items [32]
- » Document: Security guideline for the e-voting project [33]
- » Live personnel interview

Assessment

56. On recommendation R-10:

- Architects and developers are instructed to ensure that all submissions adhere to the rules set forth in internal security guidelines [33].
- A subset of the guideline is also specified in the form of a checklist [32] and is treated as a criteria to submit new work items into the system.
- The initiative aims at bringing structured security verifications into the design approval process and is expected to be enforced prior to entering development (definition of ready).

57. Based on the evidence shown to the examiners, the implementation of recommendation R-10 is considered satisfying and marked as completed.

58. On recommendation R-11:

- The e-voting team is in the process of deploying a dynamic application security testing (DAST) solution (Burp Suite) to perform attended and unattended runtime assessments of its HTTP-enabled components. The deployment of the platform is still ongoing with the training of the security champions team and is expected to reach operational status before second half of 2023.

59. Based on the evidence and the proposed action plan shown to the examiners, the implementation of recommendation R-11 is still considered to be in progress and aimed towards completion.

Additional remarks

60. While reviewing the content of the security guideline and its associated checklist, the examiners noted a potential occurrence of the "everything security checklist", a significantly large set of security architecture and coding requirements that architects and developers will hopefully review and validate upon each submission of a new work item. In the examiners' experience, this approach is unlikely to succeed in the long term. This is due to the security fatigue, that both developers and architects may eventually express, which becomes even more likely with infrequent updates. In this context, the examiners recommend resorting to very short checklists (2 to 5 items max.). This limitation will force the author(s) of the checklist to consider the utmost critical requirements to consider when submitting a new item for development rather than asking architects and developers to conduct a disguised security assessment each time they wish to mark a work item as ready for implementation.

61. The examiners also recommend considering:

- fuzz testing (or fuzzing), for components that interact with other protocols than HTTP and those that interact with other clients than web browsers.
- Consider auditing the security reviews and approvals issued prior to the development (e.g., definition of ready).

4.7 Summary of recommendations and statuses

Key	OEV key(s)	Finding	Recommendation	Status
R-01	24.1.1	F-01	Formalise the integration of security development lifecycle guidance or best practices into the e-voting system's development process.	Completed*
R-02	24.1.1	F-01	Ensure e-voting system personnel and stakeholders have received adequate role-based training on secure systems engineering.	Completed*
R-03	24.1.1	F-01	Integrate threat modelling, or equivalent, in early stages of the development process.	Completed**
R-04	24.1.1	F-01	Establish a baseline set of security principles or rules for each phase of the development lifecycle (e.g., requirements, architecture and/or design, coding, testing, build, deployment, etc.).	Completed*
R-05	24.1.1	F-02	Establish a security champion program and appoint a champion in each e-voting system development team.	Completed*
R-06	24.1.1	F-03	Establish a reference threat model for the use of third-party components in the e-voting system, maintained with the status of implementation of chosen controls and countermeasures.	Completed**
R-07	24.1.1	F-03	Establish a security vetting process for the selection of new third-party components, and the review of existing ones, embedded in the e-voting system.	Completed**
R-08	24.1.20	F-04	Establish a central document that describes how security assurance in the e-voting system is obtained (e.g., e-voting security whitepaper).	Completed**
R-09	24.5	F-05	Establish procedures to confirm, review and improve the correct operation of security attestation measures, in particular automated measures.	Completed**
R.10	17.2	F-06	Establish procedures to review and/or validate design proposals generated in response to change requests, prior to entering the coding phase.	Completed**
R.11	17.2	F-06	Establish or improve runtime/dynamic application security testing procedures executed prior to release. For relevant components, extend these procedures with additional fuzz testing.	In progress

Table 13 - Summary of recommendations and statuses

*: marked as completed during round 2 examination
**: marked as completed during round 3 examination

4.8 Summary of findings and statuses

Key	OEV key(s)	Finding	Severity	Recommendation	Remediation status
F-01	24.1.1	Insufficient integration of security in the software development lifecycle	Moderate	R-01, R-02, R-03, R-04	Resolved**
F-02	24.1.1	Conflicting / ambiguous attribution of security responsibilities	Moderate	R-05	Resolved*
F-03	24.1.1	Insufficient protection from risky third-party components	Moderate	R-06, R-07	Resolved**
F-04	24.1.20	Insufficient security documentation	Moderate	R-08	Resolved**
F-05	24.5	Insufficient quality control over security attestation operations	Moderate	R-09	Resolved**
F-06	17.2	Insufficient security testing	Moderate	R-10, R-11	In progress

Table 14 - Summary of findings and statuses

*: marked as resolved during round 2 examination
**: marked as resolved during round 3 examination

5 Examination results and update on requirements

5.1 Scope 2a (software development process)

Development process and lifecycle requirements

Key	Requirement	Finding(s)	Decision
24.1.1	A life cycle model is defined. The life cycle model: » is used for the development and maintenance of the software (a); » provides for the necessary controls during the development and maintenance of the software (b); » is documented (c).	F-01 F-03	PASS*
24.1.2	A list must be made of the development tools used and configuration options chosen for the use of each development tool.	-	PASS
24.1.3	The documentation for the development tools includes: » a definition of the development tool (a); » a description of all conventions and directives used in the implementation of the development tool (b); » a clear description of the significance of all configuration options for using the development tool (c).	-	PASS
24.1.4	The implementation standards to be applied must be specified.	-	PASS

Table 15 - E-voting requirements: lifecycle

Software security documentation requirements

Key	Requirement	Finding(s)	Decision
24.1.20	Software development security documentation includes: » a description of the physical, procedural, personnel, and other security measures necessary to protect and ensure the integrity of the design and implementation of the software in its development environment (a); » evidence that the security measures provide the necessary level of protection to preserve the integrity of the software (b).	F-04	PASS*

Table 16 - E-voting requirements: software security documentation

Quality assurance requirements

Key	Requirement	Finding(s)	Decision
24.5	Regular and objective checks are carried out to ensure that the processes carried out and the associated work products	F-05	PASS*

	comply with the description of the processes, standards and procedures to be implemented (a). Deviations are followed up until they are corrected (b).		

Table 17 - E-voting requirements: quality assurance

Configuration management system requirements

Key	Requirement	Finding(s)	Decision
24.1.14	The software is provided with a unique identification.	-	PASS
24.1.15	The configuration management documentation includes: » a description of how configuration items are identified (a); » a configuration management plan describing how the configuration management system will be used in the development of the software and the procedures that will be followed for the adoption of changes or new elements (b); » evidence that the procedures for adoption provide for adequate review of changes for all configuration items (c).	-	PASS
24.1.16	The configuration management system: » uniquely identifies all configuration items (a); » provides automated measures to ensure that only authorised changes are made to configuration items (b); » supports the development of the software through automated procedures (c); » ensures that the person responsible for accepting the configuration item is not the same person who developed it (d); » identifies the configuration items that make up the security functions (e); » supports verification of all changes to the software using automated procedures, including logging of the author and the date and time of the change (f); » provides an automated method for identifying any configuration items that are affected by a change to a particular configuration item (g); » can identify the version of the source code on the basis of which the software is generated (h).	-	PASS
24.1.17	All configuration items are inventoried in the configuration management system.	-	PASS

24.1.18	The configuration management system is used in accordance with the configuration management plan.	-	PASS
24.1.19	A configuration list is created that contains the following items: » the software, » evidence of the checks required to ensure security compliance, » the parts that make up the software, » the source code, » the commit history[10], » reports on security flaws and on the status of their correction (a). For each element relevant to security functions, the developer is named (b). Each element is uniquely identified (c).	-	PASS

Table 18 - E-voting requirements: configuration management system

Testing requirements

Key	Requirement	Finding(s)	Decision
17.1	The functions relevant to the security of the system (security functions) are tested. The tests are documented with test plans and expected and actual test results. (a) The test plan (b): » specifies the tests to be performed; » describes the scenarios for each test, including any dependencies on the results of other tests. The expected results must show the results that are expected if the test is successfully executed. (c) The actual results must be consistent with expected results. (d)	-	PASS
17.2	An analysis must be made of the test coverage. This includes evidence that: » the tests defined in the test documentation match the functional specifications of the interfaces (a); » all interfaces have been fully tested (b).	F-06	**P. MISS**

[10] The requirement for a commit history was added in the revised ordinance.

17.3	An analysis must be made of the depth of testing. This includes evidence that: » the tests defined in the test documentation match the subsystems related to security functions and modules that play a role in ensuring security (a); » all subsystems related to the security functions mentioned in the specifications have been tested (b); » all modules that play a role in ensuring security have been tested (c).	-	PASS
25.13.3	The integration tests cover all modules.	-	PASS
25.13.4	The software tests cover all modules.	-	PASS

Table 19 - E-voting requirements: testing

Transparency requirements

Key	Requirement	Finding(s)	Decision
8.13[11]	Known flaws and the need for action associated with them are communicated transparently. **: identified as 8.12 in the round 1 examination criteria.*	-	PASS

Table 20 - E-voting requirements: transparency

Systematic correction of flaws requirements

Key	Requirement	Finding(s)	Decision
24.4.1	Processes are defined for the correction of flaws. The processes include: » documentation of specific aspects, in particular with regard to the traceability of flaws for all versions of the software, and of the methods used to ensure that system users have information on flaws, corrections and possible corrective actions (a); » the obligation to describe the nature and impact of all security flaws, information on the status of work to find a solution and the corrective measures adopted (b); » a description of how system users can make reports and enquiries about suspected flaws in the software known to the software developers (c); » a procedure requiring a timely response and automatic dispatch of security flaw reports and appropriate corrective actions to registered system users who may be affected by the flaw (d).	-	PASS

[11] Identified as requirement key 8.12 in the round 1 examination report, the requirement remains unchanged.

24.4.2	A process is defined for handling reported flaws (a). This process ensures that all reported and confirmed flaws are corrected and that the procedures for correction are communicated to system users (b). It provides for arrangements to ensure that the correction of security flaws does not give rise to new security flaws (c).	-	PASS
24.4.3	Policies must be defined for the reporting and correction of flaws. These include: » instructions on how system users can report suspected security flaws to the developer (a); » instructions on how system users can register with the developer to receive reports of security flaws and the corrections (b); » details of specific contact points for all reports and inquiries on security issues concerning the software (c).	-	PASS

Table 21 - E-voting requirements: systematic correction of flaws

*: decisions followed by an asterisk indicate decisions updated during the third-round examination.

5.2 Summary of pending or incomplete requirements

Key	Requirement	Decision
17.2	An analysis must be made of the test coverage. This includes evidence that: » the tests defined in the test documentation match the functional specifications of the interfaces (a); all interfaces have been fully tested (b).	**P.MISS**

Table 22 – Summary of pending or incomplete requirements (summary)

5.3 Summary of remaining actions / recommendations

Key	OEV key(s)	Recommendation	Status
R.11	17.2	Establish or improve runtime/dynamic application security testing procedures executed prior to release. For relevant components, extend these procedures with additional fuzz testing.	**In progress**

Table 23 - Summary of recommendations and statuses

6 References

[1] "Reorienting eVoting and ensuring stable trial operation," *www.egovernment.ch*. https://www.egovernment.ch/en/umsetzung/schwerpunktplan/vote-electronique/ (accessed Oct. 21, 2021).

[2] Swiss Federal Chancellery, Political Rights Section, "Redesign and relaunch of trials - Final report of the Steering Committee Vote électronique (SC VE)." Nov. 30, 2020. Accessed: Dec. 06, 2021. [Online]. Available: https://www.bk.admin.ch/dam/bk/en/dokumente/pore/Final%20report%20SC%20VE_November%202020.pdf.download.pdf/Final%20report%20SC%20VE_November%202020.pdf

[3] Swiss Federal Chancellery, Political Rights Section, "Partial revision of the Ordinance on Political Rights and total revision of the Federal Chancellery Ordinance on Electronic Voting (Redesign of Trials)." Apr. 28, 2021. Accessed: Dec. 06, 2021. [Online]. Available: https://www.bk.admin.ch/dam/bk/en/dokumente/pore/Explanatory%20report%20for%20consultation%202021.pdf.download.pdf/Explanatory%20report%20for%20consultation%202021.pdf

[4] Swiss Federal Chancellery, "Federal legislation." https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/versuchsbedingungen.html (accessed Oct. 21, 2021).

[5] Swiss Federal Chancellery (FCh) - Political Rights section, "Audit concept for examining Swiss Internet voting systems - v1.3." May 18, 2021.

[6] Swiss Federal Chancellery, "Federal Chancellery ordinance on electronic voting (OEV)." Apr. 28, 2021. [Online]. Available: https://www.bk.admin.ch/dam/bk/en/dokumente/pore/OEV_draft%20for%20consultation%202021.pdf.download.pdf/OEV_draft%20for%20consultation%202021.pdf

[7] Swiss Federal Chancellery (FCh) - Political Rights section, "Audit concept for examining Swiss Internet voting systems - v1.5." Sep. 15, 2022. [Online]. Available: File reference: 431.0-2/5/12/16

[8] Swiss Federal Chancellery, Political Rights Section, "Partial revision of the Ordinance on Political Rights and total revision of the Federal Chancellery Ordinance on Electronic Voting (Redesign of Trials)." May 25, 2022. Accessed: Dec. 06, 2021. [Online]. Available: https://www.newsd.admin.ch/newsd/message/attachments/71705.pdf

[9] Swiss Federal Chancellery, "Federal Chancellery ordinance on electronic voting (OEV)." May 25, 2022. [Online]. Available: https://www.fedlex.admin.ch/eli/cc/2022/336/en

[10] A. Fontes, "Examination of the Swiss Internet voting system - Audit scope 2a - Development processes." Mar. 22, 2022. [Online]. Available: https://www.admin.ch/gov/en/start/documentation/media-releases.msg-id-88085.html

[11] A. Fontes, “Examination of the Swiss Internet voting system - Audit scope 2a - Development processes - Follow-up Audit (round 2).” Nov. 02, 2022. [Online]. Available: https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/ueberpruefung_systeme.html

[12] Swiss Post, “Post Audit Response to examination report by SCRT – Scope 2a Software (Internal document).” Jul. 29, 2022.

[13] Swiss Federal Chancellery (FCh) - Political Rights section, “Audit concept for examining Swiss Internet voting systems - v1.4.” Apr. 12, 2022. [Online]. Available: File reference: 431.0-2/5/12/16

[14] Payment Card Industry, “Secure software lifecycle (Secure SLC) requirements and assessment procedures - PCI-SSF v1.0.” Jan. 2019.

[15] Center for Internet security, “CIS Controls Version 8,” *CIS*. https://www.cisecurity.org/controls/v8/

[16] Software assurance forum for excellence in code (SAFECode), “Fundamental practices for secure software development - 3rd ed.” Mar. 2018.

[17] Watson, Colin, Lynch, Aidan, Coblentz, Nick, Keary, Eoin, and Deleersnyder, Seba, “SAMM Assessment toolbox v1.5 final.” OWASP, 2009. [Online]. Available: https://github.com/OWASP/samm/tree/master/Supporting%20Resources/v1.5

[18] Software assurance forum for excellence in code (SAFECode), “Managing security risks inherent in the use of third-party components.” 2017.

[19] Swiss Post, “Threat modelling (internal document).” Mar. 29, 2023.

[20] Swiss Post, “2023-03-23 threat modelling session (internal document).” E-voting, Mar. 29, 2023.

[21] Swiss Post, “2023-02-02 threat modelling session (internal document).” E-voting, Mar. 29, 2023.

[22] L. Kohnfelder and P. Garg, “The threats to our products,” *Microsoft Interface Microsoft Corp.*, vol. 33, 1999.

[23] M. Muckin and F. Scott C., “A Threat-Driven Approach to Cyber Security.” Lockheed Martin Corporation, Feb. 19, 2019.

[24] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, “A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements,” *Requir. Eng.*, vol. 16, no. 1, pp. 3–32, 2011.

[25] Swiss Post, “Framework / library evaluation (internal document).” Architecture & technology, Mar. 29, 2023.

[26] Swiss Post, “Third party dependency request: dependency-check-maven-plugin (internal document).” E-voting, Mar. 29, 2023.

[27] Swiss Post, “Threat reference model for third-party components (internal documents).” E-voting, Apr. 05, 2023.

[28] Swiss Post, “Third party dependency review process (internal document).” E-voting, Apr. 05, 2023.

[29] Swiss Post, “Security Whitepaper of the Swiss Post Voting System.” E-voting, Mar. 31, 2023.

[30] OWASP, “Software assurance maturity model (SAMM),” *OWASP SAMM*, Sep. 14, 2022. https://owaspsamm.org/ (accessed Dec. 08, 2021).

[31] Swiss Post, “Canary Tests for AST Tools (internal document).” E-voting, Mar. 29, 2023.

[32] Swiss Post, “Security checklist (internal document).” E-voting, Mar. 29, 2023.

[33] Swiss Post, “Hitchhiker’s Guide to Application Security (Internal document).” 2022.

[34] FCh, Swiss Federal Chancellery. “Trials with E-Voting.” Accessed July 31, 2023. https://www.bk.admin.ch/bk/en/home/politische-rechte/e-voting/versuchsuebersicht.html.