Modeling and Assessing Coercion Threats in Electronic Voting

Riccardo Longo[1[0000−0002−8739−3091]], Majid Mollaeefar[1[0000−0002−0277−3029]], Umberto Morelli[1[0000−0003−2899−2227]], Chiara Spadafora[2[0000−0003−3352−9210]], Alessandro Tomasi[1[0000−0002−3518−9400]], and Silvio Ranise[1,2[0000−0001−7269−9285]]

[1] Fondazione Bruno Kessler, Center for Cybersecurity, Trento, IT
{rlongo,mmollaeefar,umorelli,altomasi}@fbk.eu
[2] Department of Mathematics, University of Trento, IT chiara.spadafora@unitn.it

Abstract. Electronic voting holds the potential to increase voter participation and streamline election processes, but its broad use is limited by many challenges, e.g., ensuring system security and usability. One of the most difficult threats to counter is coercion; i.e., the ability to monitor and force voters' actions. This paper proposes a methodology to assess an e-voting system's resistance to coercion by gathering the security properties that an e-voting solution should offer from both academia and regulation, and adapting the Microsoft STRIDE and LINDDUN threats and the OWASP Risk Rating Technologies to the e-voting scenario.

Keywords: E-Voting · Threat Modeling · Coercion Resistance · Risk Analysis

1 Introduction

Electronic voting (e-voting) is currently adopted by 36 countries worldwide (15 allowing voting via the Internet) and experimented with different technologies by 27 countries [10]. However, significant risks remain of internal and external threats – e.g., manipulated elections and state-sponsored attacks – and the impact of a compromise would be severe.

The properties that e-voting solutions need to satisfy in order to be trustworthy have long been the subject of academic research [6,8,2,17,11,14,20] and of efforts by the Council of Europe (CoE)[3]. Drawing on the experience of participating countries and organisations with direct experience or specialised knowledge of e-voting, CoE proposed in 2014 and updated in 2017 a set of recommendations and adoption guidelines on the core aspects of e-voting [7].

One of the well-established approaches to verifying the security properties of systems is threat modeling, a systematic process used to identify potential security and privacy threats to the system, as well as its risk of compromise and

[3] An international organisation including 46 countries, from within and without the EU, and promotes among others human rights, democracy and free elections.

possible mitigations. In the context of remote voting, such as Internet voting solutions that do not require the presence of voters in controlled voting environments like polling stations, coercion is one of the most important and complex threats to model. Coercion attacks have been the subject of extensive studies (e.g., [11]), and modeling and demonstrating coercion resistance is challenging.

In this paper, we propose a novel methodology for performing a threat analysis on e-voting systems, and we summarize the results of its application to assess the risks posed by a coercer to an Internet voting solution specifically designed to be coercion-resistant [4,13]. Our research includes the following contributions:

- A mapping of CoE properties and principles to verifiable voting system properties from academic literature (Section 2.1, Table 1);
- The adaptation of STRIDE methodology [19] and LINDDUN framework [12] to the e-voting scenario;
- The identification of the most relevant attackers for an e-voting system, with particular focus on the attacker that we call coercer;
- The tailoring of the likelihood factors of the OWASP Risk Rating methodology [15] to the e-voting scenario, together with new impact factors based on a new methodology to compute the risk.

We focus the analysis reported in this work on the coercer, its capabilities and the risk of damaging identified properties. Coercion is of particular importance for remote voting solutions as voting takes place in an unsecured environment. For an overview of the capabilities of the remaining identified attackers, we refer the reader to the complementary material [1].

In Section 2 we outline our main contributions on the novel methodology proposed to perform a threat analysis on e-voting systems; Section 3 reports on the results of the application of this methodology to a coercion-resistant voting protocol [4] and Section 4 draws some conclusions and discusses future work.

2 Proposed Methodology for the Threat Analysis of an E-Voting System

In this Section, we present our novel methodology for performing a threat analysis on e-voting systems. First, we provide some background on the properties of a secure e-voting protocol (Section 2.1), then we explain the threat frameworks (Section 2.3) and the main attackers (Section 2.4). In Section 2.5 we explain our adaptation of the risk rating methodology to the e-voting scenario.

2.1 Properties of a Secure E-Voting System

A secure e-voting system should aim at enforcing the following properties: *correctness* (CO) [11], *fairness* (FA) [7], *vote privacy* (VP) [21], individual verifiability (i.e., *cast-as-intended verifiability* (CAIV) [8], *recorded-as-cast verifiability* (RACV) [14], *tallied-as-recorded verifiability* (TARV) [17]) and *universal verifiability* (UV) [21]. These properties are ubiquitous in literature, so we move their description to Appendix A.1. Here we highlight two other main properties: coercion resistance, which is the main focus of our analysis, and eligibility verifiability which we slightly adapted:

- *coercion resistance* (CR) [11]: voters cannot prove whether or how they voted, even if they can interact with the adversary while voting;
- *eligibility verifiability* [6,20], which we divide for a more precise analysis into two properties that anyone should be able to verify:
  - EV1: all valid votes have been cast by eligible voters;
  - EV2: all valid votes have been cast by distinct voters.

However these properties do not properly cover the whole CIA triad of information security (i.e., Confidentiality, Integrity, Availability), particularly availability is quite overlooked. Therefore, we propose to consider two new properties:

- *right to vote* (RTV) - eligible voters are able to cast valid vote;
- *successful completion* (SC) - the election process reaches the end, publishing the result of the tallying.

The second contribution of our methodology is to group the aforementioned properties into three main aggregated properties (see Table 1), in order to render the results more comprehensible and easier to visualize. The first aggregated property, namely *Functional and Security Requirements*, refers to the fact that a voting system should allow an election to be carried out properly, allowing all and only the eligible voters to select their choices and record those correctly without disclosing them before the final tally. The second aggregated property, *Vote Freedom*, concerns the ability of a voter to express their preference without undue influence. Finally, *Verifiability* refers to the feasibility of checking every step of the election process and assessing its correctness.

The Council of Europe provides 49 recommendations for e-voting systems, organized in 8 high-level, technology-independent principles [7]. Following the aforementioned aggregation of the properties of an e-voting system, we identify the applicable CoE recommendations, which can be grouped in the same way. In Table 1, for each aggregated property, we list the constituent properties and associate the corresponding CoE recommendations and principles. Note that we consider some principles and properties to be out of the scope of a security assessment; for instance, recommendations under Principle I: Universal Suffrage concern human-computer interface issues to address ease of use and accessibility.

2.2 System Characterization

The methodology we developed to assess e-voting systems aims to evaluate how attackers may affect the properties presented in Section 2.1. As usually done for threat analysis, we consider four phases: *System Characterization*, *Threat Modeling*, *Risk Analysis*, and *Mitigation Suggestions*.

The first one aims at understanding the assets to protect and the security assumptions of system components (which indirectly limit the capabilities of

Table 1. Mapping of aggregated properties to the constituent properties from Sec. 2.1 and the corresponding CoE recommendations and principles on e-voting systems.

Aggregated Property	Constituent Properties	CoE Principles	CoE Recommendations
Functional and Security Requirements	SC, EV1, RTV, CO, FA	II: Equal suffrage	7, 8
		IV: Secret suffrage	24
		VIII: Reliability and security	45
Vote Freedom	CR, VP	III: Free suffrage	10
		IV: Secret suffrage	19, 23, 25, 26
Verifiability	CAIV, RACV, TARV, UV, EV2	II: Equal suffrage	9
		III: Free suffrage	15, 16, 17, 18
		V: Regulatory and organizational	30
		VIII: Reliability and security	49

attackers). Once the assets have been identified, the analysis continues by considering the possible threats to the assets (Section 2.3) and the relevant attackers with their attack vectors in the different e-voting phases (Section 2.4). Finally, a *Risk Analysis* (Section 3.4) quantifies the risk of the identified threats, providing a priority for the process of risk mitigation (an example is given in Section 3.5).

The System Characterization lays the groundwork for understanding the system's architecture, its components, and the data flow, which are crucial for identifying potential vulnerabilities and threats in subsequent phases. It also helps to scope the analysis via security and trust assumptions on the system components.

The first step is to identify the assets, i.e., any valuable element that requires protection from potential threats. This may include data or services that, if compromised, could lead to adverse outcomes. It is also useful to identify who can access the assets and when. Note that this characterization is highly dependent on the specific protocol under analysis, so we invite to tailor it accordingly.

Regarding the security and trust assumptions to be used in the analysis, some elements could be deemed out of scope and therefore not included[4]:

- The integrity and reliability of the authentication of voters via their digital identities, and the supportive infrastructure and communication channels.
- The integrity and reliability of the public key infrastructure which checks identities behind public keys and distributes the keys reliably to all parties.
- The security of the employed cryptography; encryption is secure, signatures are unforgeable, and zero-knowledge proofs are sound and complete.
- All entities can access a fairly reliable clock that gives a common time.

[4] Note that these assumptions are also in line with a security assessment of a public service infrastructure with trusted entities, and accessible via a mobile application.

– The integrity and reliability of the store used by voters to download the voting application.

2.3 Threat Categorization: STRIDE and LINDDUN for E-Voting

Currently, there is not a standard framework for assessing threats to an e-voting system, therefore our third contribution is to adapt the STRIDE [19] methodology and the LINDDUN [12] framework to the e-voting scenario, as recently proposed also by [9]. The categories considered in STRIDE are: Spoofing (SP), Tampering (TA), Repudiation (RE), Information Disclosure (ID), Denial of Service (DS), and Elevation of Privilege (EP); LINDDUN instead considers: Linking (LN), Identifying (IF), Non Repudiation (NR), Detecting (DT), Data Disclosure (DD), Unawareness and Unintervenability (UU), and Non Compliance (NC). A full description of the categorization adapted to e-voting can be found in Appendix A.2, while in the following we present only those explicitly mentioned by our proposed methodology in Section 3.

Tampering (TA): maliciously change or modify data stored or in transit. This can be performed on any of the communication channels or by compromising one of the web servers or the voters' devices.

Information disclosure (ID): read data stored or in transit without the necessary permissions. This results in leaking sensitive data, e.g., by compromising one of the entities or listening on communication channels.

Denial of Service (DS): deny access to resources, such as by making a web server temporarily unavailable or unusable.

Non-Repudiation (NR): being able to attribute an action to a voter, for instance in case of an over-the-shoulder attack. This is the core threat category for coercion scenarios.

Detecting (DT): deduce the involvement of a voter through observation, e.g., while listening on the communication channel between the voter's device and the web servers. This threat category is also crucial in coercion scenarios.

We consider *Unawareness and Unintervenability* and *Non-Compliance* out of scope for our threat analysis methodology. In fact, awareness is raised before voting begins through channels that lie outside the scope of the protocol in use, and compliance needs to be assessed against regulations and guidelines applicable to each concrete instance. Indeed, compliance with individual frameworks has been the subject of research for individual systems [16].

2.4 Attackers

Attackers are entities that are motivated to find and exploit vulnerabilities to achieve malicious goals according to their abilities and opportunities. We identify the most relevant attackers for an e-voting system, with their capabilities:

1. *Coercer*: this attacker aims to manipulate the outcome of an election by forcing voters to vote in a specific way or abstain from voting. They can, for instance, observe the voter and request recordings of their actions.

2. *Network Attacker*: this attacker has the ability to snoop and tamper with all Internet traffic, albeit with limited capabilities of DOS.
3. *Internal Attacker*: this attacker can corrupt and control some trusted entities (for example some of the entities responsible for tallying ballots).
4. *Device Cracker*: this attacker compromises the device that the voter uses to vote and perform verifications.
5. *Curious Authority*: this attacker can snoop but not tamper with everything managed by the authority responsible for voter authentication.

In our proposed methodology one should analyze, for each attacker, which threats it could pose to the identified assets according to the attack vectors at their disposal and the disruptions that enable further threats.

We focus here on evaluating the coercer, since it is the most peculiar within the context of e-voting. This attacker monitors voters, with the goal of ensuring that they comply with the coercer's instructions. For example, a mobster may compel someone to vote for a specific candidate, or to not vote at all. We model this monitoring ability with two attack vectors:

- *over-the-shoulder*: the attacker can observe the voter while it interacts with the voting device;
- *social engineering* the attacker can deceive and manipulate voters to obtain information or influence their actions. With a little stretch from the normal definition, in order to streamline the analysis, we include threatening voters.

2.5 Rating the Risk of Threats

Another core contribution of our work is the adaptation of the OWASP risk-rating methodology [15] to the e-voting scenario. As mentioned before, our main attacker is a coercer. Other kinds of attackers are easier to analyze with standard OWASP methodology.

While we followed and adapted OWASP criteria and scale for likelihood factors, we propose a completely novel approach for impact factors which considers how the aggregated security properties from Section 2.1 are affected, in order to account for the specificity of the e-voting scenario. The adaptation of the OWASP Likelihood factors to the e-voting scenario can be found in Appendix A.3.

Impact To evaluate the impact, we observed that the OWASP guidelines are not tailored for an e-voting solution since they have been developed for completely different scenarios (mainly cyber attacks targeting a company's IT infrastructure). We, therefore, propose a new approach by measuring the impact of successful attacks on the aggregated e-voting properties presented in Section 2.1 (see Table 1). Below, we provide the **impact factors** (scale 1 to 9) and how to estimate their value.

- *Functional and Security Requirements.* An eligible voter cannot cast a vote, but manages to solve the issue (1); votes can be modified or cancelled but the issue can be solved (3); a few eligible voters are unable to vote or votes can be modified, a few ineligible voters can vote (5); multiple eligible voters are unable to vote, multiple ineligible voters can vote, several votes can be modified, results are disclosed ahead of time (7); several voters are unable to conclude the voting process (9).

- *Vote Freedom.* Evasion strategies unlikely to be detected (1); evasion strategies detected if the voter does not follow the recommendations (2); evasion strategies of a specific voter detected before voting phases (3); knowledge that a specific voter cast a vote [5] (5); knowledge of content but not validity of a vote cast by a specific voter (6); knowledge of validity but not content of a vote cast by a specific voter (7); knowledge of content and validity of a vote cast by a specific voter (9).
- *Verifiability.* A single verification does not work, the issue can be detected and immediately solved (1); a single verification does not work, the issue can be detected and solved but not immediately (3); a single verification is missing or does not work (4); a single verification has been falsified and this goes unnoticed (6); tally proofs are missing, some voters can cast multiple valid votes (7); tally proofs have been falsified and this goes unnoticed, or many voters can cast multiple valid votes (9).

We consider an oversimplification to compute an overall impact as the average of the impact factors, since the three aggregated properties are damaged differently by different attackers. Therefore we propose to use a weighted average, giving more consideration to the aggregated properties affected by more attacks with non-zero impact value.

Risk For each threat, we compute its associated *risk* separately for each aggregated property. The risk value of the attack is the multiplication of its likelihood by its impact. We divide the risk into 6 levels according to a set of thresholds in line with the intervals from OWASP: the risk is *None* if it is exactly 0, *Very Low* if it is below 3.5, *Low* if it is less than 10.0, *Medium* if it is below 24.0, *High* if it is less than 47.0, and *Critical* if it is 47.0 or greater.

Finally, we propose to aggregate the results by dividing the attacks according to the phase during which they can be performed, thus giving some insight into which parts of the system are more vulnerable.

3 Application: Analysis of a Coercion-Resistant E-Voting Protocol

In this Section, we apply the threat modeling methodology developed in Section 2 to evaluate the Internet voting system described in Section 3.1 and illustrated in Figure 1, focusing specifically on the threats posed by a coercer. This approach will allow us to assess how our methodology addresses potential vulnerabilities and threats within the context of this specific e-voting scenario.

[5] Excluding if the attacker tells the voter to do so.

3.1 A coercion-resistant E-Voting Protocol

The e-voting protocol under analysis [4,13] derives from *Civitas* [5], improved with the techniques of [3,18] to have a vote tally linear in the number of votes, with threshold construction of the voting credential derived from [22]. This protocol has been developed in the context of a research project partly funded by the Italian national mint (Istituto Poligrafico e Zecca dello Stato) and adapted to the Italian electoral law for Italian citizens voting from abroad. Voters authenticate themselves via their national digital identity (eID) to obtain a voting credential from the responsible authorities in order to cast ballots. The concept of voting credentials has been introduced by [11]. These credentials allow voters under the influence of a coercer to express their true votes while pretending to comply with the coercer's demands. Each voting credential comprises a private and a public part and will be used to validate cast ballots. When the voter is, or fears to be, subject to a coercion attack, they can autonomously create a decoy credential indistinguishable from the real one. This credential will not validate the corresponding ballot when votes are tallied. The coercer cannot understand if a ballot has been built with a decoy or valid credential, since this distinction emerges only when the votes are tallied after all ballots have been shuffled. To enhance usability of the scheme, in [4,13] the credential is given to the voter in the form of a six-digit PIN. When inserted during the voting phase, this PIN unlocks the valid credential needed to cast a valid vote. To create a decoy credential, it is sufficient to set up a decoy PIN. The decoy credential is then delivered to the voter in the same way as the valid one.

The correctness of the PIN can be verified via a Designated Verifier Non-Interactive Zero-Knowledge Proof (DVNIZKP), which proves the correctness of the associated credential. If a decoy PIN is set up, a forged proof is created to verify the decoy credential.

The e-voting protocol is divided into five phases (see Figure 1):

1. *Setup.* The protocol parameters are chosen, and the public keys are generated (1a and 1b) and published (1c). Given some election parameters (1d), the registration authorities (Registration Tellers) cooperatively generate a voting credential for every eligible voter and publish the public parts (1e).
2. *Enrollment.* Using a voting device updated with the current election parameters (2a), each voter authenticates to the voting platform via their eID (2b), and the Electoral Roll (2c) checks their status as eligible voters. Then, the voting device requests to the RTs the voter's credential, which is delivered after a random waiting period (2d). The authorities also send a DVNIZKP, so that voters can verify the validity of the PIN shown by their device (2e).
3. *Credential Management.* Each voter can set up one or more ruse PINs (3a) and verify them with the decoy DVNIZKP (3c). Each voter can also request to receive again their valid PIN (3a) and/or verify a PIN with the DVNIZKP (3c) (multiple times). Both ruse PINs and reminders (3b) are delivered and displayed exactly as in phase 2 (2d, 2e).
4. *Voting.* Each voter expresses their preferences on their voting device and validates them by inserting a PIN (4a). The device creates an encrypted

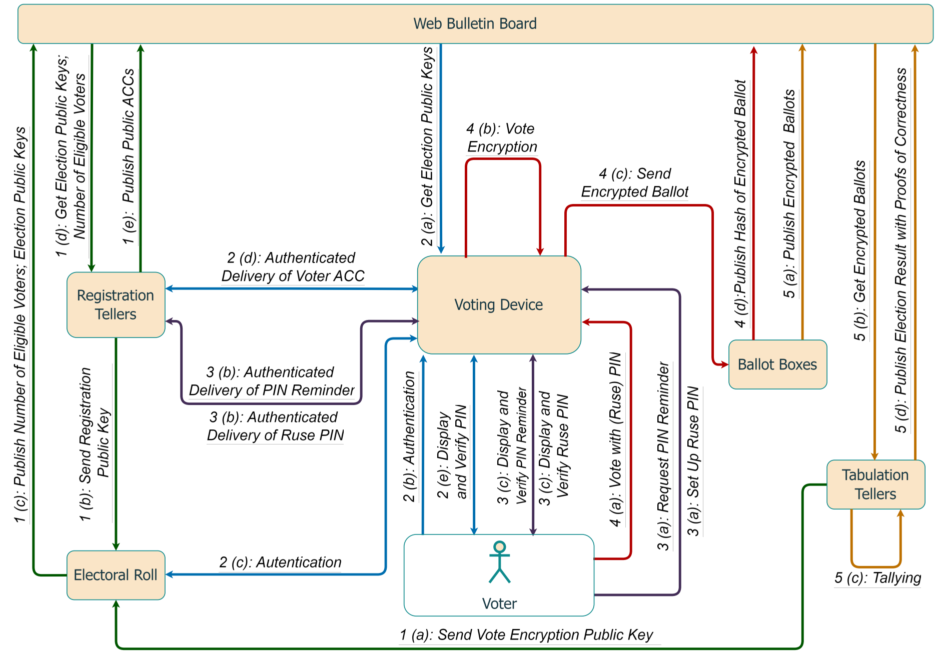
Web Bulletin Board
1 (c): Publish Number of Eligible Voters; Election Public Keys
1 (d): Get Election Public Keys; Number of Eligible Voters
1 (e): Publish Public ACCs
Registration Tellers
1 (b): Send Registration Public Key
Electoral Roll
2 (d): Authenticated Delivery of Voter ACC
3 (b): Authenticated Delivery of PIN Reminder
3 (b): Authenticated Delivery of Ruse PIN
2 (c): Autentication
2 (a): Get Election Public Keys
4 (b): Vote Encryption
4 (c): Send Encrypted Ballot
Voting Device
2 (b): Authentication
2 (e): Display and Verify PIN
3 (c): Display and Verify PIN Reminder
3 (c): Display and Verify Ruse PIN
4 (a): Vote with (Ruse) PIN
3 (a): Request PIN Reminder
3 (a): Set Up Ruse PIN
Voter
1 (a): Send Vote Encryption Public Key
4 (d):Publish Hash of Encrypted Ballot
5 (a): Publish Encrypted Ballots
Ballot Boxes
5 (b): Get Encrypted Ballots
5 (d): Publish Election Result with Proofs of Correctness
Tabulation Tellers
5 (c): Tallying

Fig. 1. Simplified Diagram of the E-Voting solution under analysis. The five phases are represented in different colours: *1-Setup* in green, *2-Enrollment* in blue, *3-Credential Management* in purple, *4-Voting* in red, and *5-Tallying* in orange.

ballot (4b) and casts it (4c). For confirmation, a hash is published on a Web Bulletin Board (WBB) (4d). Each voter can simulate multiple votes by using their decoy PINs or express their real preference with their valid PIN. Re-voting is allowed; only the last ballot cast with the valid PIN will count in the final tally.

5. *Tallying.* When the voting period ends, the encrypted ballots are released by the Ballot Boxes (5a). Duplicate, malformed and invalid ballots are set aside after being verifiably mixed by the tallying authorities (Tabulation Tellers). The remaining ballots are fetched (5b) and counted (5c), and proofs of correct tally execution are published (5d).

More details on the e-voting protocol can be found in [4,13].

3.2 Application: System Characterization

In addition to the common assumptions presented in Section 2.2, we also assume the integrity of the Web Bulletin Board. The data contained on the WBB is assumed to be stored and backed up (and replicated) using a write once, read many model. This implies that everyone access the same (updated) version.

For the analysis of the e-voting protocol, particular attention is paid to the threats posed by coercion attacks. Following the protocol assumptions, we suppose that the coercer can surveil the communication channels between the voting device and the other entities, sniffing encrypted traffic, but the monitoring and surveillance capabilities of the coercer are limited to discontinuous time frames.

The list of all assets of the protocol under analysis is available at [1]. Table 2 presents an excerpt for the *PIN*, the *Vote*, and the *communication channel* between the voter device and the ballot box. This includes, together with a short description, the identification of both the entities involved in the protocol (see Figure 1) that could access the asset and of the phase in which this can be done.

Table 2. An excerpt of identified assets. The *Access* column lists the entities that can access the asset, in the phases indicated between parentheses.

Asset	**Description**	**Access**
PIN	Code that unlocks a voting credential	Voting Device (2, 3, 4), Voter (2, 3, 4)
Vote	Preference expressed by the voter	Voting Device (4), Voter (4)
VD-BB channel	Communication channel between the Voting Device (VD) and the Ballot Box (BB)	Voting Device (4), Ballot Box (4)

3.3 Application: Threat Modeling

Following the methodology, in this second phase we start by considering, for each asset identified in Section 3.2, the consequences of the threats it could face. Afterwards, for each identified threat, we categorize it with the adapted STRIDE and LINDDUN (see Section 2.3). The result of this categorization is presented in Table 3, which outlines the number of compromised assets. Subsequently, we refine our analysis by marking in which phases the threat could be posed

Table 3. Number of assets impacted by STRIDE and LINDDUN threats.

	STRIDE						**LINDDUN**				
	SP	**TA**	**RE**	**ID**	**DS**	**EP**	**LN**	**IF**	**NR**	**DT**	**DD**
Assets	14	114	8	100	80	14	10	6	33	20	4

and which attack vectors could be used to exploit it. Finally, we identify the disruption caused by successful attacks, and whether they generate other threats.

For example, consider the asset *PIN*. A *leak* of the PIN, i.e., when the coercer learns it, may lead to a coercion attack. In fact, if the valid PIN is known, the attacker may force the voter to cast a vote complying with the given instructions and, while observing them voting, determine whether the voter is actually casting a valid vote. This threat can be categorized by STRIDE as an *information disclosure* (ID), and by LINDDUN as *non-repudiation* (NR) because the coercer could determine whether their instructions have been followed.

The phases in which the voting device displays the PIN to the voter are the enrollment (phase 2), and whenever a reminder is requested in the credential management (phase 3). Note also that, when the voter sets up a *ruse PIN* (phase 3), the voting device displays it exactly as in the previous cases. Finally, the PIN is visible on the voting device for a short period of time when it is typed in during voting (phase 4). Therefore the coercer can learn the PIN at these times with an *over-the-shoulder* attack. However, since the coercer cannot continuously monitor the voter, and a ruse PIN is always displayed exactly as the valid PIN, the attacker is not sure of the validity of the PIN learned. The PIN could also be leaked through social engineering techniques, such as pretending to be the official voting authority support service, but also in this case there is a possibility that actually a ruse PIN is leaked instead of the valid PIN.

Note that compromising the voting device (e.g., with a malicious app or malware) could also lead to a PIN leak. However, this attack vector is not among the capabilities of the coercer (but can be performed by the *device cracker*).

As an example, in Table 4, we present an excerpt of the results, the full analysis can be found at [1]. Additionally, to give an intuition of how the coercer threatens the e-voting system, we summarize in Table 5 which type of threats it can pose in each phase.

3.4 Application: Risk Analysis

In our analysis, it emerged that, for the coercer, there are a total of 44 attacks. However, only 3 have a non-zero impact value for *Functional and Security Requirements*; 40 have non-zero impact for *Vote Freedom*, and 3 for *Verifiability*. Following the methodology presented in Section 2.5, we assigned for each aggregated property the weights of 0.068 (i.e., $\frac{3}{44}$), 0.909 ($\frac{40}{44}$), and 0.068 ($\frac{3}{44}$) respectively.

To give an example of the risk analysis, let us consider once again the threat "leak of the PIN". Regarding the skill level, we consider it to be low but not null, so it is a 7 according to OWASP. The motive is high, but we scale the value down to a 7 since the observed PIN can be a ruse one. For the opportunity, some access or resources are required so it is a 7, and the exploit is easy, so it is a 5. Regarding the attack detection, if the coercer asks directly for the PIN obviously the voter is aware of it. Meanwhile, an over-the-shoulder attack is sneakier. Considering that voting operations are sensitive, so they are not usually performed in public spaces, we mediate the two cases with an average of 3. Finally, the leak of a PIN does not impact *Verifiability* or *Functional and Security Requirements*, while for

Table 4. Excerpt of Threat Analysis. The *Cat.* column categorizes the attack according to STRIDE and LINDDUN methodologies, listing all relevant types.

Asset	Threat	Cat.	Phases	Attack Vectors	Disruptions
PIN	leak	ID, NR	2, 3, 4	over-the-shoulder, app compromise, social engineering	coercion attack[1]
Vote	tampering	TA, DS	4	app compromise, malware on app	tampering of encrypted ballot[2]
VD-BB channel	DOS	DS	4	channel DOS	inability to cast a vote
VD-BB channel	snooping	DT	4	channel sniffing	coercion attack[3]

[1] Only if the attacker can reliably establish the validity of the PIN.
[2] If the voter performs the individual verifiability checks, the tampering is detected.
[3] If the attacker instructed the voter to abstain, it may detect a defiant ballot casting.

Table 5. Assessment of the phases impacted by STRIDE and LINDDUN threats posed by the Coercer: 🟠 indicates that the phase is impacted while 🟦 signifies that the phase is not impacted by the corresponding threat.

	STRIDE						LINDDUN				
Phase	**SP**	**TA**	**RE**	**ID**	**DS**	**EP**	**LN**	**IF**	**NR**	**DT**	**DD**
1	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦
2	🟦	🟠	🟦	🟠	🟠	🟦	🟠	🟠	🟦	🟠	🟦
3	🟦	🟠	🟦	🟠	🟠	🟦	🟠	🟠	🟠	🟠	🟦
4	🟦	🟠	🟦	🟠	🟠	🟦	🟠	🟠	🟠	🟠	🟦
5	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦

Vote Freedom the worst case is that with an over-the-shoulder attack the coercer detects a vote casting (impact 5), but considering all the other cases, we scaled it down to a 4. This example, together with the snooping of the channel between the voter device and the ballot box, are presented in Table 6.

Table 6. Example of Likelihood and Impact Assessment.
Legend: S = Skill Level, M = Motive, O = Opportunity, E = Ease of Exploit, D = Evasion of Attack Detection, R = Functional and Security Requirements, F = Vote Freedom, V = Verifiability.

	Likelihood					**Impact**		
Threat	**S**	**M**	**O**	**E**	**D**	**R**	**F**	**V**
PIN leak	7	7	7	5	3	0	4	0
VD-BB channel snooping	5	4	4	6	8	0	4	0

For each threat, we compute its associated risk separately for each aggregated property by multiplying the likelihood of an attack and its impact. To continue the previous example, PIN leak has a likelihood of 5.8, and a risk value of 23.2 (Medium) for *Vote Freedom*, while the other two are not impacted, hence are zero. This gives a total risk value of 21.2 (Medium).

In Table 7, we report the results obtained by analyzing the coercer and the overall risk. The numbers in the tables indicate how many successful threats with the selected risk level are posed in the corresponding phase (e.g., the coercer poses 1 threat to the property *Functional and Security Requirements* in phase 3 with risk level *Very Low*). It is important to note that some attacks may affect more than one phase and more than one property. Therefore, the total number of threats is not simply the sum but we account for the overlapping influence of certain threats on multiple phases and properties by ignoring intersections.

3.5 Application: Suggestions to mitigate the risks

Here we discuss the results reported in Table 7, giving some possible mitigations.

Vote Freedom. This is the most impacted property, as it should be expected given the nature of the attacker we are considering. The highest risks come from attacks during the voting phase which threaten the vote expressed by the voter. In fact, the vote is a critical asset, and its leak has a high impact even if the ballot itself is not valid. The reason behind this classification is that despite the possibility to employ evasion strategies to disguise the validity of a vote, it is not certain that these strategies have been employed, or whether they have been correctly executed. To have additional guarantees on their correct adoption, one of the strategies is to make all verification mechanisms mandatory in order to complete the voting process. However, this could reduce the usability of the service. It is therefore important to understand the trade-off between security and usability that works best in each specific context.

In general, all the attacks that impact this property reflect the fact that the voter may not properly follow some evasion strategy or the coercer may detect defiance by chance. The lesson learned is that to contrast a coercer it is

Table 7. Phase-by-phase breakdown of successful attacks posed by the *Coercer* along with associated risks, and both total and per phase average risk.
Legend: 🟢= Very Low Risk, 🟡= Low Risk, 🟠= Medium Risk, 🔴= High Risk, ⚫= Critical Risk, **A**= Average Risk.

	Functional and Security Requirements						**Vote Freedom**						**Verifiability**						**Total**					
Phase	🟢	🟡	🟠	🔴	⚫	**A**	🟢	🟡	🟠	🔴	⚫	**A**	🟢	🟡	🟠	🔴	⚫	**A**	🟢	🟡	🟠	🔴	⚫	**A**
1	0	0	0	0	0	⚪	0	0	0	0	0	⚪	0	0	0	0	0	⚪	0	0	0	0	0	⚪
2	0	0	0	0	0	⚪	0	**5**	**5**	0	0	🟠	0	0	**3**	0	0	🟠	**1**	**7**	**3**	0	0	🟡
3	**1**	0	0	0	0	🟢	0	**8**	**10**	0	0	🟠	0	0	**3**	0	0	🟠	**1**	**11**	**7**	0	0	🟡
4	**2**	0	0	0	0	🟢	0	0	**20**	**4**	0	🟠	0	0	**1**	0	0	🟠	**1**	**1**	**19**	**4**	0	🟠
5	0	0	0	0	0	⚪	0	0	0	0	0	⚪	0	0	0	0	0	⚪	0	0	0	0	0	⚪
Total	**3**	0	0	0	0	🟢	0	**8**	**27**	**4**	0	🟠	0	0	**3**	0	0	🟠	**1**	**11**	**24**	**4**	0	🟠

vital to deploy evasion strategies correctly, which requires great care and good instructions.

Functional and Security Requirements. The risks on this property derive from attacks where the coercer takes away the voter's voting device. Since it is sufficient to re-register with another device to solve the problem, the risk is very low. The other threats do not have an impact on this property.

Verifiability. To explain the attacks that impact the *Verifiability* we need to explain a few more details on the registration phase. When the voter registers to the e-voting system for the first time, they are supposed to safely back-up some information (e.g., via a credential manager), which is needed to verify the security and correctness of the initialization of further voting devices. If the coercer manages to destroy or corrupt this backup, then the voter will not be able to perform these verifications. The impact is medium (value 4) for all these attacks, the likelihood is also medium (around 4), so the risk level is medium.

4 Conclusions

Electronic voting systems are critical for enhancing accessibility and efficiency in modern electoral processes. However, their susceptibility to various security threats necessitates rigorous threat analysis to ensure their integrity and reliability. Particularly, the threat of coercion, a scenario where voters are pressured to vote in a certain way or disclose their voting choices, poses a significant challenge in maintaining the secrecy and voluntariness of votes. This makes conducting comprehensive threat analysis crucial for identifying and mitigating potential vulnerabilities that could undermine the democratic process.

In this paper, we presented a threat analysis methodology for e-voting systems, with a particular focus on coercion threats. Our approach adapts the STRIDE methodology, the LINDDUN framework and the OWASP Risk Rating Methodology to the e-voting scenario, and evaluates the impact of threats by using 3 groups of properties derived from academia and the Council of Europe, namely *Functional and Security Requirements*, *Vote Freedom*, and *Verifiability*.

Applying the proposed methodology to an Internet voting system specifically designed to be coercion-resistant, we identified 164 assets to be protected in 5 different phases; 5 different threat actors (including the coercer) and 442 possible threats. The threat modeling for the coercer, which is performed both manually and automatically as part of the methodology, quantifies the likelihood and impact of 43 different threats (and 8 combined ones, the most effective); the results highlight a negligible risk for threats against *Functional and Security Requirements*, 4 high-risk threats against *Vote Freedom* (and 27 medium ones), 3 medium-risk threats against *Verifiability*. As part of the methodology, we also provide possible mitigation measures to counter them.

In future work, we plan to extend the risk analysis of this Internet voting protocol for the remaining threat actors, and apply the methodology to other coercion-resistant e-voting systems to help securing them.

References

1. Complementary material of threat analysis, https://docs.google.com/spreadsheets/d/14N2AhBPlWgespNwZJkzUSdZA93pQzpzWIjwAxWeqERo
2. Araújo, R., Ben Rajeb, N., Robbana, R., Traoré, J., Youssfi, S.: Towards practical and secure coercion-resistant electronic elections. In: Cryptology and Network Security. pp. 278–297. Springer Berlin Heidelberg (2010). https://doi.org/10.1007/978-3-642-17619-7_20
3. Araújo, R., Traoré, J.: A practical coercion resistant voting scheme revisited. In: International Conference on E-Voting and Identity. pp. 193–209. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-39185-9_12
4. Bitussi, M., Longo, R., Marino, F.A., Morelli, U., Sharif, A., Spadafora, C., Tomasi, A.: Coercion-resistant i-voting with short PIN and OAuth 2.0. In: E-Vote-ID 2023. Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn (2023), to appear, preprint available at https://eprint.iacr.org/2024/1398
5. Clarkson, M.R., Chong, S., Myers, A.C.: Civitas: Toward a secure voting system. In: 2008 IEEE Symposium on Security and Privacy. pp. 354–368. IEEE (2008). https://doi.org/10.1109/SP.2008.32
6. Cortier, V., Gaudry, P., Glondu, S.: Belenios: a simple private and verifiable electronic voting system. In: Foundations of Security, Protocols, and Equational Reasoning, pp. 214–238. Springer (2019)
7. Recommendation CM/Rec(2017)5 of the Committee of Ministers to member States on standards for e-voting, https://search.coe.int/cm/Pages/result_details.aspx?ObjectID=0900001680726f6f

8. Escala, A., Guasch, S., Herranz, J., Morillo, P.: Universal cast-as-intended verifiability. In: International Conference on Financial Cryptography and Data Security. pp. 233–250. Springer (2016). https://doi.org/10.1007/978-3-662-53357-4_16
9. de Farias, J.C.L.a.A., Carniel, A., de Melo Bezerra, J., Hirata, C.M.: Approach based on stpa extended with stride and linddun, and blockchain to develop a mission-critical e-voting system. Journal of Information Security and Applications **81**, 103715 (2024). https://doi.org/10.1016/j.jisa.2024.103715
10. International institute for democracy and electoral assistance (idea) homepage. Web Page, https://www.idea.int/, accessed: 2024-04-12
11. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: Towards Trustworthy Elections, Lecture Notes in Computer Science, vol. 6000, pp. 37–63. Springer (2010). https://doi.org/10.1007/978-3-642-12980-3_2
12. Linddun privacy threat types, https://linddun.org/threat-types, accessed: 2024-05-03
13. Longo, R., Morelli, U., Spadafora, C., Tomasi, A.: Adaptation of an i-voting scheme to italian elections for citizens abroad. In: E-Vote-ID 2022. University of Tartu Press (2022). https://doi.org/10.15157/diss/027
14. Müller, J., Truderung, T.: Caised: A protocol for cast-as-intended verifiability with a second device. In: International Joint Conference on Electronic Voting. pp. 123–139. Springer (2023)
15. OWASP: Risk rating methodology, https://owasp.org/www-community/OWASP_Risk_Rating_Methodology
16. Panizo Alonso, L., Gascó, M., Marcos del Blanco, D.Y., Hermida Alonso, J.A., Barrat, J., Aláiz Moreton, H.: E-voting system evaluation based on the council of europe recommendations: Helios voting. IEEE Transactions on Emerging Topics in Computing **9**(1), 161–173 (2021). https://doi.org/10.1109/TETC.2018.2881891
17. Popoveniuc, S., Kelsey, J., Regenscheid, A., Vora, P.: Performance requirements for End-to-End verifiable elections. In: 2010 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE 10). USENIX Association, Washington, DC (Aug 2010), https://www.usenix.org/conference/evtwote-10/performance-requirements-end-end-verifiable-elections
18. dos Santos Araújo, R.S.: On remote and voter-verifiable voting. Ph.D. thesis, Technische Universität Darmstadt (2008)
19. Shostack, A.: Threat modeling: Designing for security. John Wiley & Sons (2014)
20. Smyth, B., Ryan, M., Kremer, S., Kourjieh, M.: Towards automatic analysis of election verifiability properties. In: Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security. pp. 146–163. Springer (2010). https://doi.org/10.1007/978-3-642-16074-5_11
21. U.S. Election Assistance Commission: Voluntary Voting System Guidelines (VVSG) version 2.0 (02 2021), https://www.eac.gov/voting-equipment/voluntary-voting-system-guidelines
22. Wang, H., Zhang, Y., Feng, D.: Short threshold signature schemes without random oracles. In: Progress in Cryptology - INDOCRYPT 2005. pp. 297–310. Springer Berlin Heidelberg (2005). https://doi.org/10.1007/11596219_24

A Appendix

A.1 Description of the Properties that a Secure E-Voting System Should Satisfy

As outlined in Section 2.1, a secure e-voting system should be designed to uphold some properties. Here we give some more detail on those not explained in Section 2.1:

- *correctness* (CO) [11]: an adversary cannot preempt, alter, or cancel the votes of honest voters;
- *fairness* (FA) [7]: no information about how many votes each candidate has received can be learned until the voting results are published;
- *vote privacy* (VP) [21]: no one is able to know the content of a vote;
- *coercion resistance* (CR) [11]: voters cannot prove whether or how they voted, even if they can interact with the adversary while voting;
- individual verifiability, which is subdivided into:
  - *cast-as-intended verifiability* (CAIV) [8]: the voter can verify that the complete ballot (i.e. containing the intended vote) is correctly computed and cast;
  - *recorded-as-cast verifiability* (RACV) [14]: the voter can verify that the correct ballot is recorded for the tallying;
  - *tallied-as-recorded verifiability* (TARV) [17]: anyone can verify that all and only the recorded votes are tallied, and with the correct procedure;
- *universal verifiability* (UV) [21]: anyone can check that the published result of an election has been correctly computed;
- *eligibility verifiability* [6,20], which we divide for a more precise analysis into:
  - EV1: anyone can verify that all valid votes have been cast by eligible voters;
  - EV2: anyone can verify that all valid votes have been cast by distinct voters.
- *right to vote* (RTV) - eligible voters are able to cast valid vote;
- *successful completion* (SC) - the election process reaches the end, publishing the result of the tallying.

A.2 Detailed Description of STRIDE and LINDDUN for E-Voting

STRIDE is a framework developed by Microsoft that is used as a mnemonic for the following security threat categories: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privileges. It generally applies to any IT infrastructure, specifically for scenarios with large attack surfaces (such as Internet voting). However, in the case of e-voting systems, the *information disclosure* category does not enable the identification of all types of attacks on privacy that the voter or the system can suffer.

LINDDUN is a privacy-oriented framework that considers the following privacy threat categories: Linking, Identifying, Non-Repudiation, Detecting, Data

Disclosure, Unawareness and Unintervenability, and Non-Compliance. It aims to model risks associated with the links established between individuals and the data they generate. Of particular interest for the e-voting scenario are *linking* and *identifying*. These pose an inherent risk to any voting system, electronic or otherwise, in which voters must be authenticated as having the right to vote and uniquely distinguished from other voters, but the expression of their voting intentions must be unlinkable with their identities while being tallied and published in the final aggregate.

We now briefly integrate the description of Section 2.3 with the remaining threat categories from STRIDE and LINDDUN, adapted to the context of e-voting.

Spoofing (SP): pretend to be a trustworthy entity to gain unauthorized access to sensitive data. For instance, impersonate a voter (e.g., by gaining access to their device) to vote for a different candidate, or an authority service to intercept a voter's request for the anti-coercion credentials.

Repudiation (RE): perform prohibited operations without leaving traces. In the context of e-voting this translates to violating the verifiability of the system. More precisely, a repudiation attack manages to conceal evidence that a forbidden action has been performed or that some task has not been executed correctly; while the verifiability required from an e-voting systems prescribes that it is possible to check the correct execution of the protocol.

Elevation of privilege (EP): gain privileged access to resources in order to gain unauthorized access to information or to compromise a system.

Linking (LN): associate data items or voter actions to learn more about a voter or groups of voters. For instance, by associating a voting credential with the identity of a voter.

Identifying (IF): learn the identity of a voter, for instance in case of compromising a voter's device.

Data Disclosure (DD): excessively collect, store, process, or share voters' personal data.

Unawareness and Unintervenability (UU): when individuals are not sufficiently informed, involved, or empowered concerning processing of their personal data.

Non-Compliance (NC): when the system deviates from legislation, regulation, or from standards and best practices.

A.3 OWASP Risk Methodology: E-Voting adaptation of the Likelihood scale

The OWASP Risk Rating Methodology [15] is an approach to quantify the risk of security threats in order to make informed decisions. It evaluates the risk as the product of *likelihood* – i.e., how likely/easily a vulnerability is discovered and exploited by an attacker – and *impact* – i.e., material and non-material damage, such as the loss of data integrity or the reputation damage.

The provided factors to estimate the likelihood are connected with the worst-case threat agent (skill level, motive, opportunity, and size – i.e., from a specific developer to anonymous Internet users) and the exploited vulnerability (ease of discovery, ease of exploit, awareness, and intrusion detection – i.e., from active detection to not logged).

Likelihood To tailor the OWASP Risk Rating methodology to e-voting, considering the coercion scenario, we propose the following **likelihood factors** (scale 1 to 9):

- *Skill Level.* How technically skilled is a coercer (the lower it is, the easier it is to attack the voter): we consider a value of (8) if an asset is disclosed by physical observation or by simply requesting it from the voter; (7) if the coercer needs to access a voter's device or the asset is not visible in the set of actions to cast a ballot; (5) if the coercer is able to access a communication channel to/from the voter's devices; (3) if the attacker can perform combined attacks (such as learning voting intentions via social engineering and the vote expressed via an over-the-shoulder attack).
- *Motive.* How motivated is the coercer, in particular how well the attack could enable voter monitoring and control. We propose a rating of (2) if the monitoring is not specifically on ballot casting and has low probability of being successful, (4) if it still succeeds with low probability but directly linked to voting operations, (8) if the attack would give some direct control over voting capabilities. Intermediate values should be used to adapt for higher success probabilities or monitoring that allows the coercer to detect coercion evasion strategies (for example requesting a ruse PIN).
- *Opportunity.* What resources and opportunities are needed to perform the attack: we use (3) if the coercer needs to tamper with an external software supposed to be secure (e.g., a credential manager used by the voter); (4) if the attacker needs to know when the voter device will communicate (in order to attack); (5) if the attacker has to be able to attack a voter in two or more different time frames; (6) if the attacked asset can be retrieved by simply observing the voter but not during a compulsory operation (e.g. during an optional verification) (7) if the attacked asset is visible on the device during one of the operations necessary to cast a vote.
- *Ease of Exploit.* How easy is it for this group of threat agents to actually exploit this vulnerability. Here we follow the OWASP rating: theoretical (1), difficult (3), easy (5), automated tools available (9).
- *Evasion of Attack Detection* (named Intrusion detection by OWASP). We use (3) if the voter is promptly alerted or the attack is easily noticeable (e.g., they realise to have lost the voting device); (6) the voter may not realise to have been attacked if proper care has not been used (e.g., by not leveraging optional security mechanisms such as verification steps); (8) the attack can easily go unnoticed (e.g., in case a coercer passively listens on communication channels).

As previously said, since our main focus is the coercer, we do not quantify the *Size* factor. Considering its attack vectors (over-the-shoulder and social-engineering) we also do not consider *Ease of Discovery* (i.e., how easy an attacker discovers this vulnerability) and *Awareness* (i.e., how well known is this vulnerability to the attacker). Intermediate values (such as a skill level of 4) are assigned by comparison among similar attacks (or vulnerable assets), and lastly by using the values from the OWASP methodology (e.g., skill level 1 indicates no technical skills). The overall likelihood of each identified threat is computed as the average value of all the considered factors.