**Electronic Voting Platform – Part Two: Security Solutions and Cost‑Benefit Analysis**

**Overview**

Part Two of the assignment involves choosing the right security solutions to 20 requirements that have been identified in Part one. In every requirement, the group must come up with possible solutions, evaluate them based on their pros and cons, and select the most appropriate solution through a cost-benefit evaluation. This report defines that process. A summary table indicates the proposed solutions and the approach chosen in each requirement and the rationales were provided in detail backed by reputable sources.

**Summary of Requirements and Proposed Solutions**

The following table provides a summary of possible solutions to each requirement (using keywords to save space) and the solution selected. Cons and pros are given in detail in the next section. Chosen solution means the solution that was selected as the most effective one after considering the benefits of security, the impact that it has on the user, the complexity and cost of its implementation.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Requirement (concise title)** | **Potential solutions (keywords)** | **Selected solution** |
| 1 | Strong voter authentication | username/password; digital ID; biometrics | username/password with identity proofing |
| 2 | Multi‑factor authentication | SMS OTP; TOTP app; hardware token; biometrics | TOTP app for voters |
| 3 | Secure communication (TLS/SSL) | TLS 1.3/1.2; non‑encrypted HTTP | TLS 1.3/1.2 with strong ciphers |
| 4 | Encryption of stored votes | database encryption (AES); homomorphic encryption; no encryption | database encryption (AES) with key management |
| 5 | End‑to‑end vote integrity (digital signatures) | digital signatures; MACs; blockchain | digital signatures on ballots |
| 6 | Voter privacy and anonymity | mix‑nets; homomorphic encryption; pseudonyms | mix‑net or homomorphic tallying |
| 7 | Role‑based access control (RBAC) | RBAC framework; hard‑coded roles; attribute‑based access control | standard RBAC framework |
| 8 | Comprehensive audit logging | centralised SIEM; local log files; DB logs | centralised SIEM with retention |
| 9 | Verifiability | receipt with cryptographic proof; zero‑knowledge proofs; none | cryptographic voter receipts & public audit proofs |
| 10 | Protection against DoS attacks | DDoS mitigation service; rate‑limiting; auto‑scaling | DDoS mitigation + rate‑limiting |
| 11 | Secure API input validation | parameterised queries; stored procedures; input escaping; WAF | parameterised queries + allow‑listing |
| 12 | Secure coding and reviews | static analysis tools; code reviews; developer training | static analysis + peer code reviews |
| 13 | Patch & vulnerability management | regular patch schedule; automated updates; containerisation | regular patching and vulnerability scanning |
| 14 | Incident response & recovery plan | formal IR plan; ad hoc response; outsourced SOC | formal incident response plan |
| 15 | Data backups & disaster recovery | full + incremental backups; real‑time replication; tape backups | weekly full + daily incremental backups |
| 16 | Continuous monitoring & anomaly detection | SIEM/IDS; manual log review; outsourced SOC | SIEM/IDS with anomaly detection |
| 17 | Password & session policies | strong complexity rules; passphrases; passwordless; session timeout | strong password rules + session timeout |
| 18 | Malware & antivirus protection | real‑time antivirus; network scanning; none | real‑time antivirus & endpoint protection |
| 19 | Privacy notice & consent | comprehensive notice & consent; minimal notice; none | comprehensive privacy notice & consent |
| 20 | Secure key management | hardware security module (HSM); software keystore; environment variables | HSM with key encryption & rotation |

**Detailed Rationale for Each Requirement and Solution**

**1 – Strong voter authentication**

**Potential solutions:**

* **Username/password** – Issue each voter a unique username and strong password; verify identity through voter registration information.
* **Digital identity (government ID)** – Use government‑issued digital certificates or verified ID to authenticate the voter.
* **Biometrics** – Require fingerprint or facial recognition to verify identity.

**Pros and cons:**

* *Username/password* is cost‑effective and familiar to users. However, passwords alone can be compromised, so this approach relies heavily on strong password policies[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#:~:text=Implement%20Proper%20Password%20Strength%20Controls%C2%B6) and may not sufficiently mitigate credential theft.
* *Digital identity* provides strong assurance of voter eligibility but requires integration with government identity systems and may raise privacy concerns. It can be complex to deploy for all voters.
* *Biometrics* offer a high‑assurance factor but increase hardware costs and raise concerns over the storage and protection of biometric data.

**Chosen solution:** A **username/password system combined with identity proofing** during voter registration. Voters will create strong passwords that meet complexity guidelines (minimum length, composition rules and periodic rotation)[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#:~:text=Implement%20Proper%20Password%20Strength%20Controls%C2%B6), and identity proofing will verify that the digital account corresponds to a real voter. This balances security, cost and usability.

**2 – Multi‑factor authentication (MFA) for voters**

**Potential solutions:**

* **SMS one‑time password (OTP)** – Send a code via text message that the user enters after their password.
* **TOTP (Time‑based One‑Time Password) application** – Use an authenticator app (e.g., Google Authenticator) that generates a time‑based code.
* **Hardware token** – Provide voters with physical tokens (e.g., FIDO key) that generate codes.
* **Biometric second factor** – Use fingerprint or face recognition as the second factor.

**Pros and cons:**

* *SMS OTP* is simple to deploy but vulnerable to SIM‑swapping attacks and message interception; it may also incur per‑message costs.
* *TOTP app* offers stronger security and is inexpensive because most users have smartphones; however, it requires voters to install an app.
* *Hardware tokens* provide very strong security but are costly to distribute and can be lost or damaged.
* *Biometric second factor* enhances assurance but requires hardware and may raise privacy and accessibility concerns.

**Chosen solution:** Implement **TOTP‑based MFA** for voters. The OWASP MFA cheat sheet recommends requiring MFA for all users and offering TOTP options[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html#:~:text=Quick%20Recommendations%C2%B6). TOTP provides a good balance of security and cost: codes are generated on the user’s device, reducing the risk of interception, and there is no reliance on SMS infrastructure.

**3 – Secure communication (TLS/SSL)**

**Potential solutions:**

* **TLS 1.3/1.2 with strong cipher suites** – Configure the web server to support only TLS 1.3 and TLS 1.2, disable older protocols, and use GCM ciphers; enable fallback protection (TLS\_FALLBACK\_SCSV).
* **Non‑encrypted HTTP** – No encryption; data is transmitted in plaintext.

**Pros and cons:**

* *TLS 1.3/1.2* provides confidentiality, integrity and authentication for data in transit. The OWASP TLS cheat sheet states that secure server configuration should support TLS 1.3 and TLS 1.2, disable old protocols and use strong ciphers[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html#:~:text=Introduction%C2%B6).
* *Non‑encrypted HTTP* is insecure: data can be intercepted or modified; thus it is unsuitable.

**Chosen solution:** Configure the voting platform to **use TLS 1.3 (with fallback to TLS 1.2)** and strong cipher suites, while disabling SSL and older TLS versions. This ensures confidentiality and integrity of communications[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html#:~:text=Introduction%C2%B6).

**4 – Encryption of stored votes**

**Potential solutions:**

* **Database encryption (AES)** – Encrypt the vote database at rest using symmetric cryptography (e.g., AES‑256). Keys are stored securely.
* **Homomorphic encryption** – Encrypt votes such that they can be tallied without decryption; results are decrypted only at the end.
* **No encryption** – Store votes in plaintext.

**Pros and cons:**

* *Database encryption* is relatively easy to implement and protects data at rest; however, decryption is required during tallying. It relies on secure key management.
* *Homomorphic encryption* allows calculations on encrypted data, preserving privacy during tallying, but is computationally intensive and complex to implement for large elections.
* *No encryption* exposes sensitive data and is unacceptable.

**Chosen solution:** Use **database encryption (e.g., AES‑256)** to encrypt votes at rest and protect against database compromise. Homomorphic encryption could be considered for future upgrades, but due to high complexity and performance overhead, standard encryption is a practical choice for the prototype. Proper key management will ensure confidentiality of the stored votes.

**5 – End‑to‑end vote integrity (digital signatures)**

**Potential solutions:**

* **Digital signatures** – Voters sign their ballots using a private key; the server verifies the signature to ensure the ballot is authentic and untampered.
* **Message authentication codes (MAC)** – Use symmetric keys to compute a MAC for each ballot.
* **Blockchain storage** – Store each vote as a transaction on a blockchain, using the chain to provide immutability and integrity.

**Pros and cons:**

* *Digital signatures* provide strong integrity and non‑repudiation: the voting guide notes that digital signatures ensure that ballots come from eligible voters and that votes are not tampered with[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=Digital%20signature%20These%20provide%20verification,being%20used%20to%20unlock%20them). However, key distribution and management are required.
* *MACs* are efficient but rely on a shared secret; if the key is compromised, integrity is lost.
* *Blockchain* offers immutability, but adding blockchain complexity may not align with election requirements and can reduce scalability.

**Chosen solution:** Implement **digital signatures** on ballots: each voter’s device signs the vote using a private key; the server verifies with the corresponding public key. Digital signatures ensure end‑to‑end integrity and prove that ballots originate from legitimate voters[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=Digital%20signature%20These%20provide%20verification,being%20used%20to%20unlock%20them).

**6 – Voter privacy and anonymity**

**Potential solutions:**

* **Mix‑nets** – Encrypt votes and pass them through a series of servers that shuffle and re‑encrypt them, breaking the link between voter identity and the encrypted vote. The UK Engage report notes that mix‑nets anonymise ballots by shuffling them so the link between voter and vote is removed[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=Mix,are%20safely%20decrypted%20and%20are).
* **Homomorphic encryption** – Use homomorphic encryption to allow tallying of votes without ever decrypting individual ballots.
* **Pseudonyms/tokens** – Assign a pseudonymous token to each voter to decouple identity from the vote; the token is used to retrieve the ballot.

**Pros and cons:**

* *Mix‑nets* provide strong anonymity and have been used in end‑to‑end verifiable systems, but they require multiple independent servers and careful implementation.
* *Homomorphic encryption* ensures that ballots are never decrypted individually, but the cryptosystem is computationally expensive and may limit performance.
* *Pseudonyms* are simpler to implement but may still allow correlation between voters and votes if tokens are compromised.

**Chosen solution:** Use **mix‑nets or homomorphic tallying** to anonymise ballots. Mix‑nets are selected for the prototype because they provide strong anonymity by shuffling and re‑encrypting ballots[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=Mix,are%20safely%20decrypted%20and%20are). Homomorphic encryption could be evaluated for future iterations.

**7 – Role‑based access control (RBAC)**

**Potential solutions:**

* **RBAC framework** – Implement a standard RBAC system that defines roles (admin, auditor, voter) and assigns permissions accordingly.
* **Hard‑coded roles** – Embed access logic directly into the application code.
* **Attribute‑based access control (ABAC)** – Use attributes (e.g., clearance level, department) to determine access.

**Pros and cons:**

* *RBAC framework* offers clear separation of duties and is easy to manage; it supports least‑privilege principles[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html#:~:text=Enforce%20Least%20Privileges%C2%B6).
* *Hard‑coded roles* are simple but difficult to maintain and audit, increasing the risk of privilege escalation.
* *ABAC* provides more granularity but is more complex to implement and administer.

**Chosen solution:** Use a **standard RBAC framework** with clearly defined roles and privileges. The OWASP Authorization cheat sheet recommends enforcing least privilege, denying access by default and thoroughly reviewing authorization logic[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html#:~:text=Enforce%20Least%20Privileges%C2%B6)[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html#:~:text=Deny%20by%20Default%C2%B6). RBAC meets these guidelines and balances granularity and maintainability.

**8 – Comprehensive audit logging**

**Potential solutions:**

* **Centralised SIEM** – Forward logs from application servers, firewalls and endpoints to a Security Information and Event Management (SIEM) system for correlation and long‑term storage.
* **Local log files** – Store logs on each server.
* **Database logging** – Write logs into a database table.

**Pros and cons:**

* *Centralised SIEM* allows aggregation and correlation of logs, early detection of suspicious activity and retention of critical logs for regulatory requirements. CISA guidance recommends sending logs to a central SIEM, correlating them and retaining critical logs for at least a year[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=evidence%20of%20their%20activities,of%20one%20year%2C%20if%20possible).
* *Local log files* are easy to implement but can be deleted or tampered with if the server is compromised; they also make correlation difficult.
* *Database logging* centralises logs but may slow database performance and requires additional security.

**Chosen solution:** Implement **centralised audit logging via a SIEM**, aggregating logs from all components. Logs should be retained for at least one year and protected from unauthorized access[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=evidence%20of%20their%20activities,of%20one%20year%2C%20if%20possible).

**9 – Verifiability (cast‑as‑intended, recorded‑as‑cast)**

**Potential solutions:**

* **Cryptographic voter receipts** – Provide voters with a receipt containing a cryptographic commitment to their ballot that they can later use to verify that their vote was recorded correctly without revealing the vote.
* **Zero‑knowledge proofs** – Use zero‑knowledge proofs or proofs of correct shuffling to demonstrate that the votes were counted correctly.
* **No verifiability** – Provide no mechanism for voters to verify their vote.

**Pros and cons:**

* *Cryptographic receipts* allow voters to confirm that their ballot is recorded and counted as cast without revealing the vote. The UK Engage report emphasises “cast‑as‑intended, recorded‑as‑cast and counted‑as‑recorded” as critical verifiability properties[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=anonymised,Counted%20as%20registered).
* *Zero‑knowledge proofs* provide strong assurance but may be difficult for voters to understand and require advanced cryptography.
* *No verifiability* undermines trust in the election system.

**Chosen solution:** Implement **cryptographic receipts** and publish public proofs (e.g., hash chains, proof of shuffling) that allow voters and auditors to verify the “cast‑as‑intended, recorded‑as‑cast” property[uk-engage.org](https://uk-engage.org/wp-content/uploads/2024/07/End-to-End-Verifiable-Election-Systems.pdf#:~:text=anonymised,Counted%20as%20registered). This balances transparency and complexity and enhances voter confidence.

**10 – Protection against denial‑of‑service attacks**

**Potential solutions:**

* **DDoS mitigation service** – Contract with a third‑party provider to detect and filter malicious traffic before it reaches the voting servers.
* **Rate‑limiting and load balancing** – Implement rate‑limiting at the application and network layers; use load balancers to distribute traffic.
* **Auto‑scaling** – Deploy the application in a cloud environment that can scale up resources during high load.

**Pros and cons:**

* *DDoS mitigation service* provides specialised protection against volumetric attacks; however, it incurs ongoing cost.
* *Rate‑limiting and load balancing* reduce the impact of application‑level attacks and help distribute load but may not handle very large attacks alone.
* *Auto‑scaling* helps absorb traffic spikes but cannot protect against network saturation.

**Chosen solution:** Deploy a **multi‑layer approach**: contract a **DDoS mitigation service** and implement **rate‑limiting and load balancing**. CISA’s DDoS spotlight recommends using vulnerability scanning services, defense‑in‑depth strategies and DDoS mitigation providers, and integrating DDoS response into incident plans[cisecurity.org](https://www.cisecurity.org/insights/spotlight/election-security-spotlight-ddos-attacks#:~:text=). Combining these measures ensures availability during election periods.

**11 – Secure API input validation**

**Potential solutions:**

* **Parameterized queries** – Use prepared statements with bound parameters for database queries.
* **Stored procedures** – Encapsulate database logic in stored procedures.
* **Allow‑list input validation** – Validate and sanitise input based on expected patterns.
* **Web application firewall (WAF)** – Deploy a WAF to filter malicious requests.

**Pros and cons:**

* *Parameterized queries and stored procedures* prevent SQL injection by separating data from code[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html#:~:text=,to%20test%20for%20the%20issue).
* *Allow‑list input validation* ensures only expected input is accepted[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html#:~:text=,66).
* *WAF* provides an additional layer of protection but does not replace proper coding practices.

**Chosen solution:** Use **parameterized queries** and **allow‑list input validation** for all APIs. OWASP’s injection cheat sheet lists prepared statements, stored procedures, allow‑list input validation and escaping as primary defenses[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html#:~:text=,to%20test%20for%20the%20issue)[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Injection_Prevention_Cheat_Sheet.html#:~:text=,66). While a WAF can be added later, secure coding is the foundation of this requirement.

**12 – Secure coding and code reviews**

**Potential solutions:**

* **Static analysis tools** – Use automated tools to detect vulnerabilities during development.
* **Peer code reviews** – Conduct manual code reviews focusing on security issues.
* **Developer security training** – Provide training on secure coding standards.

**Pros and cons:**

* *Static analysis tools* identify common vulnerabilities early but may produce false positives and require tuning.
* *Peer code reviews* foster knowledge sharing and catch logic flaws but require time and discipline.
* *Developer training* builds long‑term security awareness but may not catch immediate flaws.

**Chosen solution:** Combine **static analysis tools** with **peer code reviews** and **developer security training**. This layered approach encourages secure coding practices and detects vulnerabilities early.

**13 – Patch and vulnerability management**

**Potential solutions:**

* **Regular patch schedule** – Apply security patches on a defined schedule (e.g., monthly) and within vendor‑recommended timelines.
* **Automated updates** – Configure systems to automatically install updates.
* **Containerisation** – Use container images with known, patched components; rebuild images regularly.

**Pros and cons:**

* *Regular patch schedule* ensures timely remediation of vulnerabilities; however, manual processes can lead to delays.
* *Automated updates* reduce administrative overhead but may introduce downtime or compatibility issues if applied without testing.
* *Containerisation* allows consistent deployment and easier patching but requires container management expertise.

**Chosen solution:** Implement a **regular patch schedule** supported by **vulnerability scanning**. CISA emphasises that patch management reduces the likelihood of compromise[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=enterprise%20networks%20and%20strengthen%20election,practices%20cover%20the%20following%20topics). Automated updates may be used for non‑critical components after testing.

**14 – Incident response and recovery plan**

**Potential solutions:**

* **Formal incident response (IR) plan** – Develop a documented plan with defined roles, contact information, incident classification, escalation procedures and communication channels.
* **Ad hoc response** – Respond to incidents informally without a pre‑defined plan.
* **Outsourced security operations centre (SOC)** – Contract with a third‑party SOC to handle incidents.

**Pros and cons:**

* *Formal IR plan* provides clear guidance and ensures a coordinated response. CISA recommends establishing incident response plans and identifying thresholds for actions[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=,Patch%20management%20policies).
* *Ad hoc response* leads to confusion and delays when incidents occur.
* *Outsourced SOC* can provide expert support but may not be necessary for smaller elections and may increase cost.

**Chosen solution:** Develop a **formal incident response and recovery plan** that defines roles and contact information and integrates DDoS and other cybersecurity incidents into the plan[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=,Patch%20management%20policies)[cisecurity.org](https://www.cisecurity.org/insights/spotlight/election-security-spotlight-ddos-attacks#:~:text=). The team may consult external experts for guidance but will retain control over response procedures.

**15 – Data backups and disaster recovery**

**Potential solutions:**

* **Weekly full backups and daily incremental backups** – Perform a full backup regularly (e.g., weekly) and incremental backups daily to capture changes. CIS recommends combining full and differential or incremental backups because full backups are slower but faster to restore, while incremental backups are faster but slower to restore[cisecurity.org](https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-backups#:~:text=A%20backup%20is%20a%20copy,for%20data%20protection%20and%20recovery).
* **Real‑time replication** – Mirror data to a remote site in real time.
* **Tape backups** – Store backups on physical tapes kept off‑site.

**Pros and cons:**

* *Full + incremental backups* balance backup time, storage and restore speed. They provide resilience against ransomware and data loss[cisecurity.org](https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-backups#:~:text=A%20backup%20is%20a%20copy,for%20data%20protection%20and%20recovery)[cisecurity.org](https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-backups#:~:text=Backups%20are%20necessary%20due%20to,abide%20by%20data%20retention%20requirements).
* *Real‑time replication* offers minimal data loss but may replicate corrupted data and is costly.
* *Tape backups* provide an offline copy but require manual handling and slower recovery.

**Chosen solution:** Adopt **weekly full backups combined with daily incremental backups** and store copies off‑site. This approach aligns with CIS recommendations[cisecurity.org](https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-backups#:~:text=A%20backup%20is%20a%20copy,for%20data%20protection%20and%20recovery) and ensures data can be restored quickly in the event of malware, hardware failure or natural disasters[cisecurity.org](https://www.cisecurity.org/insights/spotlight/ei-isac-cybersecurity-spotlight-backups#:~:text=Backups%20are%20necessary%20due%20to,abide%20by%20data%20retention%20requirements).

**16 – Continuous monitoring and anomaly detection**

**Potential solutions:**

* **Security Information and Event Management (SIEM) with intrusion detection** – Deploy a SIEM platform that analyses logs in real time and triggers alerts for anomalies.
* **Manual log review** – Periodically review logs by hand.
* **Outsourced SOC** – Use a managed detection and response service.

**Pros and cons:**

* *SIEM with intrusion detection* provides automated monitoring and can quickly identify suspicious activity. CISA encourages establishing baselines for host and network activity and investigating anomalies[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=Establish%20a%20Baseline%20for%20Host,and%20Network%20Activity).
* *Manual log review* is resource‑intensive and may miss subtle patterns.
* *Outsourced SOC* offers expert monitoring but may be expensive and require data sharing.

**Chosen solution:** Implement **continuous monitoring via a SIEM with anomaly detection** to establish baseline activity and detect deviations[cisa.gov](https://www.cisa.gov/news-events/news/best-practices-securing-election-systems#:~:text=Establish%20a%20Baseline%20for%20Host,and%20Network%20Activity). Alerts will be integrated into the incident response plan.

**17 – Password and session policies**

**Potential solutions:**

* **Strong password complexity rules** – Enforce minimum and maximum lengths, require a mix of characters and restrict reuse.
* **Passphrases** – Allow long passphrases for increased memorability.
* **Passwordless authentication** – Use WebAuthn or FIDO2 for passwordless logins.
* **Session timeout and re‑authentication** – Terminate sessions after inactivity and require re‑authentication for sensitive actions.

**Pros and cons:**

* *Strong password rules* are easy to implement but can lead to password fatigue; they must be balanced with usability. OWASP recommends minimum and maximum lengths, composition rules and transmitting passwords only over TLS[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#:~:text=Implement%20Proper%20Password%20Strength%20Controls%C2%B6).
* *Passphrases* improve memorability and often encourage longer passwords.
* *Passwordless authentication* reduces password fatigue but may be complex to deploy and overlaps with MFA.
* *Session timeout* ensures that inactive sessions are closed, reducing the risk of session hijacking.

**Chosen solution:** Enforce **strong password policies** (minimum length, complexity, rotation) and implement **session timeouts**. Require passwords to be transmitted only over TLS[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html#:~:text=Implement%20Proper%20Password%20Strength%20Controls%C2%B6). The team may allow passphrases to improve usability.

**18 – Malware and antivirus protection**

**Potential solutions:**

* **Real‑time antivirus and endpoint protection** – Install antivirus software on all servers and workstations; enable real‑time scanning and regular updates.
* **Network scanning and intrusion prevention** – Deploy network‑based intrusion prevention systems (IPS).
* **No antivirus** – Rely on other controls.

**Pros and cons:**

* *Real‑time antivirus* detects and blocks malicious files and behaviours; however, it requires regular updates and may impact performance.
* *Network scanning and IPS* complement endpoint protection by monitoring network traffic for malicious patterns.
* *No antivirus* leaves systems vulnerable to malware.

**Chosen solution:** Deploy **real‑time antivirus and endpoint protection** on all servers and integrate it with network‑based intrusion detection. This layered approach helps detect malware and maintain system integrity.

**19 – Privacy notice and consent for data collection**

**Potential solutions:**

* **Comprehensive privacy notice and opt‑in consent** – Provide clear information about what data is collected, why, how it is used and retained; require explicit consent.
* **Minimal privacy notice** – Provide limited information; rely on implied consent.
* **No privacy notice** – Provide no information.

**Pros and cons:**

* *Comprehensive notice and consent* enhances transparency and complies with data protection laws; however, it may require additional effort to draft and maintain notices and consent forms.
* *Minimal notice* may not meet legal requirements and can reduce voter trust.
* *No notice* risks violating privacy laws and eroding trust.

**Chosen solution:** Provide a **comprehensive privacy notice** and obtain **explicit consent** from voters regarding data collection and processing. Although not a technical control, this measure supports legal compliance and builds trust in the system.

**20 – Secure key management**

**Potential solutions:**

* **Hardware Security Module (HSM)** – Store and handle cryptographic keys inside tamper‑resistant hardware; encrypt keys with other keys of equal or greater strength; provide integrity protection.
* **Software keystore** – Store keys in encrypted files or key stores on the server.
* **Environment variables** – Store keys in environment variables on the server.

**Pros and cons:**

* *HSM* provides the highest level of security by isolating keys and performing cryptographic operations inside protected hardware; OWASP’s key management cheat sheet recommends encrypting keys with another key of equal strength and storing them in an HSM or cryptographic vault[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html#:~:text=2,based%20algorithms)[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html#:~:text=1,than%20the%20keys%20being%20protected).
* *Software keystores* are easier to deploy but may be vulnerable if the server is compromised; keys should still be encrypted and integrity‑protected.
* *Environment variables* are convenient but not secure for sensitive keys.

**Chosen solution:** Use an **HSM or secure cryptographic vault** to store and manage keys. Keys should be encrypted using a key‑encryption key of equal or greater strength and backed up securely[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html#:~:text=2,based%20algorithms)[cheatsheetseries.owasp.org](https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html#:~:text=1,than%20the%20keys%20being%20protected). Proper key rotation and access control must be enforced.

**Conclusion**

The section offers a sensible choice of solutions to all 20 security requirements of the Electronic Voting Platform. Both chosen solutions balance security with cost and practicality with references to the trusted sources, including OWASP, CISA, CIS and UK Engage. These solutions will be used to lead the implementation of the group in later stages of the project.