Implementing security measures within a threat intelligence sharing platform

➢ **Tool : MISP (Malware Information Sharing Platform)**

1. Security Assessment of Existing Platform Architecture

a. Authentication & access control

⇨ Method supported : MISP supports various authentication mechanisms, including :

- Username/Password: Utilizes bcrypt hashing for secure password storage.
- API Keys: Each user can generate unique API keys with specific permissions.
- SSO/LDAP: Integration with identity providers like LDAP and SAML2 is possible through plugins.
- MFA: Multi-Factor Authentication can be implemented via reverse proxies like Authentik or Keycloak .

⇨ Role-based access control (RBAC) : MISP offers customizable RBAC, allowing administrators to define roles and permissions tailored to organizational needs.

b. Encryption mechanisms

⇨ Data in Transit: MISP enforces HTTPS for secure communication using SSL/TLS certificates.

⇨ Data at Rest: While MISP doesn't provide native encryption for data at rest, it's recommended to implement database-level encryption (e.g., TDE in MariaDB) to protect stored data.

⇨ Email Encryption: Supports GPG and S/MIME for encrypting and signing notifications, enhancing the confidentiality of shared intelligence .
⇨ Event Signing: Introduced in MISP 2.4.156, a new synchronization event signing mechanism ensures the integrity and authenticity of shared events .

c. Vulnerability assessment

⇨ Known Vulnerabilities: Previous versions of MISP have had vulnerabilities, such as:

- CVE-2020-28947: XSS in template elements.
- CVE-2021-25324: Stored XSS in galaxy cluster view.
- CVE-2021-27904: Information disclosure due to incorrect sharing group associations .

⇨ Mitigation: Regular updates and patches are released to address identified vulnerabilities. It's crucial to keep MISP instances up-to-date to mitigate security risks.

d. Data privacy & compliance

⇨ GDPR Compliance: MISP assists organizations in complying with GDPR by implementing data minimization practices. Users are guided to include only necessary data when reporting vulnerabilities .
⇨ ISO/IEC 27010:2015 Compliance: MISP aligns with this standard, which focuses on information security management for inter-organizational and inter-sector communications. It includes cryptographic techniques for information sharing and requires logging of internal dissemination of shared information .

⇨ Terms & Conditions: MISP allows customization of Terms & Conditions, enabling organizations to define clear guidelines for sharing and coordinating on threat intelligence.

e. Security best practices

⇨ Regular Updates: Ensure MISP is running the latest stable release to benefit from security patches and new features.
⇨ Secure Configuration: Implement security best practices, such as disabling unused features, enforcing strong password policies, and restricting access to the MISP instance.
⇨ Monitoring & Logging: Utilize MISP's logging subsystems to monitor user activities and detect potential security incidents.
⇨ Community Engagement**:** Participate in the MISP community to stay informed about emerging threats and share insights on improving the platform's security.

2. Strengthening Data Encryption

a. Secure transport protocols

⇨ Goal : protect data in motion between clients, server, and integrated tools.
⇨ Steps to implement TLS on MISP :

- Use HTTPS (SSL/TLS)
    - Install MISP behind apache or Nginx
    - Use let’s encrypt or a commercial CA for TLS certificates
    - Enforce TLS 1.2 + only, disable TLS 1.0/1.1 and weak ciphers

- Configure strict transport settings
    - In your web server add strict=transport-security headers
    - Redirect all HTTP traffic to HTTPS
- Test TLS strength
    - Use tools like : SSL labs and testssl.sh for CLI-based scanning

b. Encrypt data at rest with AES-256

⇨ Goal : protect stored threat data and logs from unauthorized access.

⇨ Options for data-at-rest encryption in MISP :

- Database-level encryption
    - Use MariaDB or MySQL with TDE (Transparent Data Encryption)
    - Encrypt sensitive tables (e.g., events, attributes, logs) with AES-256
- Filesystem encryption
    - Use linux LUKS, dm-crypt, or eCryptfs to encrypts the volume storing :
    - MYSQL/MariaDB data directory
    - /var/www/MISP
    - Log and backup directories
- Encrypt backups
    - Encrypt backup archives using gpg or openssl

c. Secure key management system

⇨ Goal : handle encryption keys securely – ensure availability, confidentiality and auditability.

- Local key store
  - Store keys in a restricted-access directory
  - Use chmod 600 permissions and restrict OS-level access
- Use a dedicated KMS
  - hashiCorp vault
  - AWS KMS
  - Azure key vault
  - GCP cloud KMS
- Rotation and auditing
  - Rotate keys regularly
  - Log key access and usage
  - Implement alerts for unauthorized access attempts

3. Enhancing User Authentication and Authorization

a. Implement multi-factor authentication

⇨ MISP does not natively support MFA, but can be implement it via :

⇨ Reverse proxy with MFA support

- Use a proxy in front of MISP.
- Ex. Keycloak + MISP
- Set up keycloak with MFA

- Use SAML2 or OIDC to federate login to MISP
- Protect the /users/login endpoint via keycloak

b. Fine-grained access control

⇨ MISP has bult-in role-based access control that allows :

Role	Permissions
Org Admin	Full control over their organization's data
Site Admin	Full system-wide access
User	Limited to view/edit/create based on org and sync rules
Read-only	Can only view, not modify
Automation	API-only users with limited UI access

⇨ Customize roles :

- In the UI : administration > roles
- You can restrict access to specific features
- Limit visibility to specific tags or distribution levels

⇨ Best practice :

- Assign minimum privileges based on actual responsibility.
- Audit role settings regularly.

c. Integrate with SSO

⇨ MISP supports SSO through SAML 2.0 using SimpleSAMLphp.
⇨ Setup steps :

- Install simpleSAMLphp on the same server or a separate auth node
- Configure MISP to use SAMl authentication

- Update config.php to enable SAML
- Link attributes to MISP users

- Connect to your idP

4. Improving access control for threat intelligence data

a. Review & update ACL in MISP

⇨ What to review :

- Who can view, edit, publish, sync, or delete threat data
- What roles exist and what each role can/cannot do
- Whether distribution settings and sharing groups are used correctly

b. Implement granular access control

⇨ Objective : restrict read/write/delete/sync access based on role, organization, data type, and trust level.

⇨ How to do it :

- Use role permissions

Role Type	Description
User	Access to org-specific data
Org Admin	Manages users and data within one org
Site Admin	Full platform access
Read Only	Cannot create/edit
Automation	API-only access

⇨ Limit distribution via tags

- Use TLP or custom tags
- Restrict visibility based on distribution value:
    - 0 : your organization only
    - 1 : this community only
    - 2 : connected communities
    - 3 : all communities
    - 4 : sharing group

c. Enforce access policies for trusted parties only

⇨ User Vetting

- Only onboard vetted users from trusted orgs
- Require admin approval for all new accounts

⇨ Audit Logs

- Enable full audit logging: Administration > Logs
- Monitor data exports, login attempts, API use

⇨ Restrict API Access

- Assign API keys per user
- Disable unused keys and restrict via IP range or gateway

⇨ Limit Sync Partners

- Only allow synchronization with known, signed, and trusted instances
- Enable event signing to verify authenticity (available since MISP 2.4.156)

5. Implementing Data Anonymization and Privacy Measures

i. Anonymization Techniques for MISP

a. Data Masking

- Replaces real PII with fake but realistic data (useful for demo/test environments).
- Example:
  - john.doe@example.com → user123@example.com

b. Tokenization

- Replaces sensitive data with a non-sensitive placeholder (token), mapped securely in a vault.
- Used when data must be restored later (e.g., for LEA use).

c. Hashing

- One-way transformation (e.g., SHA-256) for data that never needs to be reversed.
- Ensure use of salted hashes to avoid rainbow table attacks.

d. Generalization

- Reduce the precision of data. Example: sharing just the first two octets of an IP address (192.168.*.*).

e. Suppression

- Remove PII fields completely from records when they are not necessary.

ii. Implementing in MISP

a. Using MISP Taxonomies and Object Templates

- Define attributes containing PII with specific object templates.
- Enforce tagging with [TLP:AMBER], [TLP:RED] or custom tags indicating sensitivity.

b. Create a Data Sanitization Module

- Develop a MISP PyMISP script or a module (pre-sharing or post-receipt) to:
  - Detect fields like email, ip-src, ip-dst, comment, etc.
  - Apply masking, tokenization, or removal based on rules.
- Use the MISP Event filters and object templates to define which attributes need sanitization.

c. Leverage MISP Sharing Groups and Distribution Levels

- Use distribution settings:
  - This community only, Connected communities, etc.
- Define sharing groups based on trust levels and sensitivity.

d. Log Access and Anonymization Activities

- Enable audit logging in MISP.
- Monitor who accesses what data, when, and from where.

iii. Integration and Automation

a. Integrate with DLP and SIEM Tools

- For alerts if sensitive data is shared without anonymization.

b. Automate PII Scanning

- Use Python scripts or modules integrated with PyMISP to automatically scan events and flag/transform PII.

c. Compliance and Data Subject Rights

- Store anonymized and original data separately if required for lawful bases.
- Provide deletion and access logging features to support data subject rights.

6. Integrating Security Tools and Threat Intelligence Sharing Protocols

i. Integration of SIEM systems with MISP

⇨ **Goals :**

- Real-time ingestion of threat intel into SIEM
- Use MISP as a threat intelligence source for correlation and alerting
- Feedback loop form SIEM to MISP

⇨ Common integration methods

SIEM	Method of Integration	Notes
Splunk	MISP App for Splunk, API, PyMISP	Real-time ingestion via REST API
IBM QRadar	STIX/TAXII integration or MISP connector	Use IBM's Threat Intelligence Platform (TIP)
Elastic SIEM	Logstash + PyMISP or direct API	Enrich log data with MISP attributes
Microsoft Sentinel	Custom Logic Apps, API Integration	Use REST and STIX feeds

⇨ **Steps for Integration**

- Enable MISP REST API and secure it via API keys and HTTPS.
- Configure PyMISP or use native connector in the SIEM platform.
- Set up **s**cheduled polling or push for:
    - IOC ingestion
    - Sightings reporting
    - Event updates
- Normalize data formats (JSON, STIX 2.1) for SIEM compatibility.
- Add mapping logic (e.g., map MISP "ip-dst" to SIEM "destination.ip").

ii. Use of industry-standard protocols: STIX & TAXII

⇨ **MISP and STIX/TAXII**

- MISP supports:
    - STIX 1.1, 2.0, 2.1
    - TAXII 2.1 (client and server) via plugins

⇨ **Implementation Guide**

- Enable TAXII support in MISP:
    - Use misp-taxii-server
- Configure collection-to-event mapping
    - Example: "APT Threats" → MISP tag APT
- Connect SIEM/TIP as a TAXII client
    - Pull feeds or subscribe to specific collections

- Ensure format compatibility
    - Validate exported data using stix2 Python lib or with SIEM feed validation tools

iii. Secure Data Exchange Mechanisms

⇨ **Security Controls**

- Enable HTTPS with strong TLS (≥1.2)
- Use API key rotation policies
- Limit API scope using role-based access control (RBAC) in MISP
- Enable audit logging of all exports/imports
- For outbound feeds:
    - Restrict by IP, token, or sharing group
    - Sign data where possible (e.g., PGP)

7. Incident response and reporting

i. Components of the IRF

component	Function
Monitoring layer	Detect anomalous or unauthorized activity in MISP
Alerting system	Trigger notifications to security teams
Investigation tools	Centralize logs, sightings, and correlated threat data
Response tools	Track incident handling, user actions, IOC updates
Reporting module	Summarize incidents, vulnerabilities, response outcomes

ii. Monitoring & detection

⇨ Enable MISP Platform Monitoring

- MISP Audit Logs:
  - Tracks logins, API calls, object modifications.
  - Available in Administration > Audit logs.
- System Logs:
  - Web server logs (/var/log/apache2/access.log)
  - OS security logs (/var/log/auth.log)
- Enable MISP Logging to SIEM
  - Send logs to Splunk, QRadar, or ELK for centralized monitoring.

⇨ Detectable Anomalies

- Multiple failed logins (brute force attempts)
- Large data exports from non-authorized roles
- Sudden tagging or removal of sensitive IOCs
- Creation or deletion of events outside work hours

iii. Automated alerting setup

⇨ Option 1: Use MISP + SIEM

- Route audit logs to a SIEM platform.
- Set up detection rules:
  - Excessive IOC download events
  - New admin account creation
  - Direct modification of TLP-tagged data
- Alert via email, Slack, or ticketing system.

⇨ Option 2: Lightweight Monitoring with Fail2Ban or OSSEC

- Monitor /var/log/auth.log and MISP API access.
- Block or alert on repeated failed logins.

8. Security and Compliance Documentation

i. Security controls documentation framework

Control Area	Details to Document
Encryption	TLS version, cipher suites, data-at-rest encryption, certificate details
Multi-Factor Authentication (MFA)	Providers used, enforcement policies, recovery steps
Role-Based Access Control (RBAC)	Defined roles, permissions mapping, least privilege strategy
Audit Logging	What is logged, retention policy, log review schedule
Incident Response	IR policy, escalation matrix, detection tools, post-incident reporting
Backup & Recovery	Frequency, encryption, test restore procedures
Data Handling (PII)	Anonymization/tokenization controls, user data lifecycle, GDPR DSR workflows

ii. Compliance checks & reviews

⇨ **Targeted standards**

- GDPR : data minimization, rights to access/erasure, consent/logging
- ISO 27001 : annex a controls

iii. Security configuration guide for MISP deployment

⇨ **Operating System**: Harden Linux (disable unused services, patching schedule)
⇨ **Firewall Rules**: Allow only HTTPS, restrict MISP API/IP access
⇨ **SSH Access**: Disable root login, enforce key-based authentication
⇨ **Access Controls**

- RBAC in MISP:
  - Define roles: Analyst, Publisher, Admin, Read-only
  - Restrict event creation, deletion, sharing
- MFA:
  - Integrate with IdP (e.g., LDAP/SAML + Duo/OTP)
  - Enforce via application gateway or web portal

⇨ **Encryption & Certificates**

- TLS ≥1.2 with strong cipher suites
- Auto-renewing certificates (e.g., Let's Encrypt with Certbot)
- Encrypt MISP database and backup storage (e.g., LUKS, GPG)

⇨ **Secure MISP Settings**

- Disable insecure feeds or import modules
- Enable audit logging and log forwarding
- Configure “Warning Lists” to prevent importing known safe IOCs

⇨ **Backup & Recovery**

- Automate database and config backups (daily)
- Store offsite and verify via checksum
- Document restore procedures