

1. Information Security Policy (EV6)

1.1 Purpose

This policy establishes the framework for protecting MapRun's information assets, user data, and technology infrastructure.

1.2 Scope

This policy applies to all MapRun systems, including:

- Mobile applications (iOS and Android)
- Backend servers and cloud infrastructure
- User data and event information
- Administrative systems and tools
- Third-party integrations

1.3 Information Security Principles

Confidentiality: User data is protected from unauthorized access through encryption and access controls.

Integrity: Data accuracy is maintained through validation, backup, and change control processes.

Availability: Services are designed for reliability with monitoring and redundancy where practical.

1.4 Data Protection

1.4.1 User Data

- Personal information limited to name, email, and GPS tracking data during events
- GPS data stored only for event results and analysis
- User data encrypted in transit (TLS) and at rest
- Users can request data deletion via support@maprun.net

1.4.2 Access Control

- Administrative access restricted to authorized personnel only
- Multi-factor authentication required for backend systems

MapRun - FNE Enterprises

- Access logs maintained for security monitoring
- Regular review of access permissions (quarterly)

1.5 Security Controls

1.5.1 Technical Controls

- Encryption: TLS 1.2+ for data in transit, AES-256 for data at rest
- Authentication: Secure password policies, MFA for administrators
- Network Security: Firewall rules, restricted ports, VPC isolation
- Application Security: Input validation, secure coding practices

1.5.2 Monitoring

- System logs reviewed weekly for suspicious activity
- Automated alerts for security events
- Regular security assessments

1.6 Third-Party Services

MapRun uses the following third-party services with appropriate security controls:

- Cloud hosting (AWS/similar) with encryption and access controls
- Payment processors (Stripe, PayPal) - PCI DSS compliant
- Map services (Google Maps) - API key restrictions applied

1.7 User Responsibilities

Users must:

- Keep their device security up to date
- Not share account credentials
- Report security concerns to support@maprun.net

1.8 Policy Review

This policy is reviewed annually and updated as needed to address new risks or technology changes.

MapRun - FNE Enterprises

2. Business Continuity Plan (EV7)

2.1 Purpose

This plan ensures MapRun can continue to provide essential services during disruptions and recover quickly from incidents.

2.2 Scope

This plan covers:

- Mobile application availability
- Backend server services
- Event data and results
- User support services

2.3 Critical Services

2.3.1 Priority 1 - Essential (Recovery within 24 hours)

- Mobile app downloads (via App Store/Google Play)
- Event selection and download to allow participation in an event
- GPS tracking functionality during events
- Event results upload and display

2.3.2 Priority 2 - Important (Recovery within 72 hours)

- Event creation and management tools
- Historical results database
- User support email

2.3.3 Priority 3 - Non-critical (Recovery within 7 days)

- Website updates
- Statistics and analytics
- Documentation updates

2.4 Risk Assessment

MapRun - FNE Enterprises

Risk	Likelihood	Mitigation
Server failure	Medium	Automated backups, cloud redundancy
Cloud provider outage	Low	Multi-region deployment capability
Key personnel unavailable	Medium	Documentation, shared knowledge
Cyber attack	Low	Security controls, monitoring
Payment system failure	Low	Multiple payment providers

Table 1: Business continuity risk assessment

2.5 Response Strategy

2.5.1 Communication Plan

Internal Communication:

- Primary contact: peter@maprun.net
- Backup contact: Strategenics Pty Ltd support
- Status updates via team communication channel

External Communication:

- User notifications via MapRun website
- Email to event organizers as needed
- Social media updates for major incidents

2.5.2 Alternative Operations

During disruptions:

- Mobile apps continue to function offline for GPS tracking
- Results can be uploaded when connectivity restored
- Paper-based backup methods available for events
- Manual result processing capability maintained

2.6 Key Contacts

Role	Contact
Primary Administrator	support@maprun.net
AWS Support Partner	Strategenics Pty Ltd https://strategenics.atlassian.net/servicedesk/customer/user/login?destination=portals

MapRun - FNE Enterprises

Cloud Hosting Support	Via provider portal
Payment Processor Support	Stripe/PayPal support lines

Table 2: Key contact information

2.7 Testing and Maintenance

- Backup restoration tested quarterly
 - Contact information verified annually
 - Plan reviewed and updated annually
 - Lessons learned documented after incidents
-

MapRun - FNE Enterprises

3. Disaster Recovery Plan (EV8)

3.1 Purpose

This plan provides specific procedures for recovering MapRun systems and data following a disaster or major system failure.

3.2 Scope

Recovery procedures for:

- Backend servers and databases
- Application infrastructure
- User data and event results
- Administrative access and tools

3.3 Recovery Objectives

System/Service	RTO (Recovery Time)	RPO (Data Loss)
Mobile apps (stores)	0 hours*	N/A
Backend API servers	4 hours	1 hour
Event database	4 hours	1 hour
Results processing	8 hours	24 hours
Website	24 hours	24 hours

Table 3: Recovery time and data loss objectives

*Apps already distributed via app stores remain functional

3.4 Backup Strategy

3.4.1 Data Backup

- Database backups: Daily automated backups, retained for 30 days
- User data: Continuous replication to backup region
- Event files: Weekly backups, retained for 12 months
- System configurations: Version controlled, backed up daily

MapRun - FNE Enterprises

3.4.2 Backup Storage

- Primary backups: Cloud storage (same provider, different region)
- Secondary backups: Offline storage updated monthly
- Backup encryption: AES-256 encryption at rest
- Backup testing: Quarterly restoration tests

3.5 Recovery Procedures

3.5.1 Server Failure Recovery

Step 1: Assessment (15 minutes)

- Identify affected systems
- Determine failure cause
- Notify key stakeholders

Step 2: Backup Server Activation (30 minutes)

- Launch replacement server from cloud template
- Configure networking and security groups
- Verify connectivity

Step 3: Data Restoration (2 hours)

- Restore latest database backup
- Verify data integrity
- Apply transaction logs to minimize data loss

Step 4: Service Restoration (1 hour)

- Update DNS records
- Test application functionality
- Monitor for issues

Step 5: Verification (30 minutes)

- Confirm all services operational
- Verify data accuracy
- Notify users of restoration

3.5.2 Database Corruption Recovery

1. Stop application access to database
2. Identify corruption extent and cause

MapRun - FNE Enterprises

3. Restore from most recent clean backup
4. Apply transaction logs where possible
5. Validate data integrity with sample checks
6. Resume service and monitor closely

3.5.3 Complete Infrastructure Loss

1. Activate alternate cloud region
2. Deploy infrastructure from configuration backups
3. Restore data from most recent backup
4. Reconfigure DNS and networking
5. Test functionality thoroughly before going live
6. Communicate estimated restoration time to users

3.6 Data Recovery Priorities

Priority 1 (Critical):

- Current event data (last 7 days)
- User account information
- Recent results (last 30 days)

Priority 2 (Important):

- Historical event data
- Historical results database
- Event configurations

Priority 3 (Desirable):

- Analytics data
- System logs beyond 30 days
- Archived event files

3.7 Roles and Responsibilities

Role	Responsibilities
System Administrator	Execute recovery procedures, system restoration
Database Administrator	Data restoration, integrity verification
Communications Lead	User notifications, status updates
Testing Lead	Verify functionality post-recovery

MapRun - FNE Enterprises

Table 4: Disaster recovery team roles

3.8 Post-Recovery Actions

- Document incident details and timeline
- Conduct root cause analysis
- Update recovery procedures based on lessons learned
- Review and improve backup/monitoring strategies
- Report to stakeholders

3.9 Plan Testing

- Tabletop exercises: Semi-annually
 - Backup restoration tests: Quarterly
 - Full recovery simulation: Annually
 - Plan review and update: Annually or after major changes
-

MapRun - FNE Enterprises

4. Incident Response Plan (EV9)

4.1 Purpose

This plan establishes procedures for detecting, responding to, and recovering from security incidents affecting MapRun services.

4.2 Scope

This plan covers:

- Security breaches and data compromises
- Malware and cyber attacks
- Unauthorized access attempts
- Service disruptions from malicious activity
- Privacy violations

4.3 Incident Classification

Severity	Description	Response Time
Critical	Active breach, data exposure, service down	Immediate (< 1 hour)
High	Attempted breach, vulnerability discovered	4 hours
Medium	Suspicious activity, minor security issue	24 hours
Low	Policy violation, informational alert	72 hours

Table 5: Incident severity classification

4.4 Incident Response Team

Primary Responders:

- Incident Lead: System Administrator
- Technical Lead: Developer/Technical Support
- Communications Lead: Management/Support Team

Contact Information:

- Primary: support@maprun.net
- Emergency: +61 419 612 369

MapRun - FNE Enterprises

4.5 Response Process

4.5.1 Detection and Identification

Detection Sources:

- Automated monitoring alerts
- User reports
- Security logs review
- Third-party notifications

Initial Assessment:

- Verify incident is genuine (not false positive)
- Classify severity level
- Identify affected systems and data
- Activate response team

4.5.2 Containment

Immediate Actions (Critical/High Incidents):

1. Isolate affected systems if necessary
2. Block malicious IP addresses or accounts
3. Preserve evidence (logs, snapshots)
4. Prevent further damage or data exposure
5. Document all actions taken with timestamps

Short-term Containment:

- Apply temporary fixes or workarounds
- Increase monitoring on affected systems
- Change compromised credentials
- Notify relevant stakeholders

4.5.3 Eradication

1. Identify root cause of incident
2. Remove malware or malicious code
3. Close security vulnerabilities
4. Patch systems and update configurations
5. Verify threats are fully removed

Document Version: 1.0

Effective Date: February 18, 2026

Review Date: February 18, 2027

Owner: FNE Enterprises (ABN 16 169 645 217)

MapRun - FNE Enterprises

4.5.4 Recovery

1. Restore systems from clean backups if needed
2. Gradually return services to normal operation
3. Monitor closely for recurrence
4. Verify data integrity
5. Restore user access

4.5.5 Post-Incident Review

Within 7 days of incident resolution:

- Document complete incident timeline
- Analyze what worked well and what didn't
- Identify improvements to procedures or controls
- Update security measures and policies
- Share lessons learned with team

4.6 Communication Procedures

4.6.1 Internal Communication

- Incident team notified immediately upon detection
- Regular status updates during response (hourly for critical incidents)
- Management briefed on critical/high incidents within 2 hours

4.6.2 External Communication

User Notification Required When:

- Personal data has been compromised
- Service disruption exceeds 4 hours
- User action required (password reset, etc.)
- Regulatory notification obligations exist

Notification Timeline:

- Initial notification: Within 24 hours of confirmation
- Updates: As significant developments occur
- Final notification: Within 7 days of resolution

Communication Channels:

MapRun - FNE Enterprises

- Email to affected users
- Website banner notification
- Social media updates for widespread issues

4.7 Specific Incident Scenarios

4.7.1 Data Breach

1. Immediately restrict access to affected systems
2. Determine scope: what data, how many users
3. Secure evidence for potential investigation
4. Notify affected users within 24 hours
5. Provide guidance on protective measures
6. Assess regulatory reporting requirements

4.7.2 Ransomware Attack

1. Isolate infected systems immediately
2. Do not pay ransom
3. Assess backup availability for recovery
4. Report to law enforcement if appropriate
5. Restore from clean backups
6. Identify and close infection vector

4.7.3 Unauthorized Access

1. Lock compromised accounts immediately
2. Review access logs to determine scope
3. Change all administrative credentials
4. Review and strengthen access controls
5. Monitor for further unauthorized attempts
6. Notify affected users if data accessed

4.7.4 DDoS Attack

1. Activate DDoS mitigation through hosting provider
2. Identify attack patterns and sources
3. Block malicious traffic where possible
4. Communicate service disruption to users

MapRun - FNE Enterprises

5. Monitor for service restoration
6. Review and enhance DDoS protections

4.8 Evidence Preservation

For all security incidents:

- Preserve system logs and audit trails
- Take snapshots of affected systems before changes
- Document all actions with timestamps
- Maintain chain of custody for evidence
- Store evidence securely for potential investigation
- Retain for minimum 90 days or longer if required

4.9 Continuous Improvement

- Review and update plan quarterly
 - Conduct tabletop exercises semi-annually
 - Update contact information as changes occur
 - Incorporate lessons from actual incidents
 - Stay informed of emerging threats
-

MapRun - FNE Enterprises

5. Patch Management Standards (EV12)

5.1 Purpose

This document establishes standards for timely identification, testing, and deployment of security patches and system updates for MapRun infrastructure.

5.2 Scope

This process covers:

- Server operating systems (Linux/Unix)
- Database systems
- Web servers and application servers
- Third-party libraries and dependencies
- Mobile application frameworks
- Cloud infrastructure components

5.3 Patch Management Principles

Timely Updates: Security patches applied based on severity and risk.

Testing First: Patches tested in development before production deployment.

Minimal Disruption: Updates scheduled during maintenance windows when possible.

Documentation: All patches tracked and documented.

5.4 Patch Categories and Timelines

Severity	Description	Target Timeline
Critical	Active exploits, zero-day vulnerabilities	24-48 hours
High	Serious vulnerabilities, no active exploit	7 days
Medium	Moderate risk, no immediate threat	30 days
Low	Minor issues, feature updates	Next cycle (90 days)

Table 6: Patch severity and deployment timelines

MapRun - FNE Enterprises

5.5 Patch Management Process

5.5.1 Identification

Monitoring Sources:

- Operating system security advisories
- Database vendor security bulletins
- Framework and library vulnerability databases
- Cloud provider security notifications
- CERT/CVSS vulnerability reports

Weekly Review:

- Check all monitoring sources
- Identify applicable patches
- Assess severity and risk
- Prioritize patches for deployment

5.5.2 Assessment

For each identified patch:

1. Determine if patch applies to MapRun systems
2. Assess vulnerability severity (CVSS score)
3. Evaluate risk of exploitation
4. Identify affected systems and services
5. Determine dependencies and prerequisites
6. Plan testing approach

5.5.3 Testing

Development Environment Testing:

- Deploy patch to development server
- Test core application functionality
- Verify API operations
- Check mobile app connectivity
- Test event creation and results processing
- Monitor for errors or performance issues

Test Duration:

MapRun - FNE Enterprises

- Critical patches: 4-12 hours minimum
- High patches: 1-2 days
- Medium patches: 3-5 days
- Low patches: 1 week

5.5.4 Approval

Approval Authority:

- Critical/High patches: System Administrator
- Medium/Low patches: Technical Lead
- Emergency patches: Can be expedited with post-approval review

Approval Criteria:

- Testing completed successfully
- Rollback plan prepared
- Maintenance window scheduled (if needed)
- Stakeholders notified (for service impact)

5.5.5 Deployment

Pre-Deployment:

- Take system backup/snapshot
- Verify rollback procedure
- Notify users if service interruption expected
- Stage patch files and deployment scripts

Deployment Process:

1. Place service in maintenance mode if required
2. Apply patch following vendor instructions
3. Restart services as needed
4. Verify services operational
5. Test critical functionality
6. Monitor system logs for errors

Post-Deployment:

- Verify patch applied successfully
- Monitor system performance for 24 hours
- Document deployment details and time
- Remove maintenance mode

Document Version: 1.0

Effective Date: February 18, 2026

Review Date: February 18, 2027

Owner: FNE Enterprises (ABN 16 169 645 217)

MapRun - FNE Enterprises

- Notify users of service restoration

5.5.6 Verification

- Confirm patch version installed
- Run vulnerability scans to verify fix
- Check system logs for issues
- Validate application functionality
- Monitor for unexpected behavior

5.6 Emergency Patching

For critical zero-day vulnerabilities with active exploitation:

1. Immediate assessment upon notification (< 2 hours)
2. Expedited testing (minimum safe testing only)
3. Management approval obtained quickly
4. Deploy to production within 24-48 hours
5. Enhanced monitoring post-deployment
6. Full testing conducted after emergency deployment

5.7 Mobile Application Updates

App Store Updates:

- Security updates prioritized for expedited review
- Users notified of critical updates via in-app messaging
- Forced update mechanism for critical security issues
- Update release notes include security information

Timeline:

- Critical: Submit within 48 hours, promote forced update
- High: Submit within 7 days, encourage update
- Medium/Low: Include in next regular release (30-90 days)

5.8 Third-Party Dependencies

Monitoring:

- Automated dependency scanning tools used
- Weekly check for library vulnerabilities

MapRun - FNE Enterprises

- GitHub security alerts monitored
- NPM/package manager advisories reviewed

Update Process:

- Vulnerable dependencies updated promptly
- Compatibility testing before deployment
- Pin versions to avoid breaking changes
- Document all dependency updates

5.9 Documentation Requirements

For each patch deployment, document:

- Patch identifier and description
- Vulnerability details (CVE number, severity)
- Affected systems
- Testing results
- Deployment date and time
- Deployed by (person responsible)
- Any issues encountered
- Verification results

5.10 Exceptions

When Patching May Be Delayed:

- Patch conflicts with critical system functionality
- Vendor advisory recommends waiting for revised patch
- Compensating controls implemented to mitigate risk
- System scheduled for replacement within 30 days

Exception Process:

- Document reason for delay
- Implement compensating controls where possible
- Set target date for patch application
- Review exception status weekly
- Management approval for delays exceeding target timeline

5.11 Maintenance Windows

MapRun - FNE Enterprises

Scheduled Maintenance:

- Regular window: 2nd Tuesday each month, 10 AM - 3 PM AEST (Midnight - 3AM UTC)
- Emergency window: As needed with 24-hour notice when possible
- Maintenance calendar published on website
- Users notified 48 hours in advance

During Event Season:

- Non-critical patches deferred to avoid event days
- Critical patches deployed off-peak hours
- Enhanced monitoring during high-usage periods

5.12 Metrics and Reporting

Tracked Metrics:

- Number of patches identified per month
- Average time to deploy by severity
- Number of overdue patches
- Exception rate and reasons
- Patch-related incidents

Reporting:

- Monthly summary of patching activity
- Quarterly review of patch compliance
- Annual assessment of patch management effectiveness

5.13 Review and Improvement

- Process reviewed quarterly
 - Metrics analyzed for compliance with timelines
 - Lessons learned from patch-related issues
 - Update procedures based on experience
 - Training for team on new tools or processes
-

6. Secure Software Development Lifecycle (EV13)

6.1 Purpose

This document defines the secure software development lifecycle (SSDLC) process for MapRun application development, ensuring security is integrated throughout the development process.

6.2 Scope

This process applies to:

- Mobile application development (iOS and Android)
- Backend API and server development
- Web interface development
- Third-party integrations
- Infrastructure as code

6.3 SSDLC Principles

Security by Design: Security considered from initial design through deployment.

Defense in Depth: Multiple layers of security controls implemented.

Least Privilege: Applications and users granted minimum necessary permissions.

Secure Defaults: Default configurations are secure configurations.

6.4 Development Lifecycle Phases

6.4.1 Phase 1: Requirements and Planning

Security Requirements:

- Identify sensitive data and functions
- Define security requirements (authentication, authorization, encryption)
- Assess compliance requirements (privacy laws, school safety standards)
- Identify threat scenarios and risks
- Document security acceptance criteria

MapRun - FNE Enterprises

Security Considerations:

- Data protection requirements
- User authentication and access control
- Privacy requirements for student/school use
- Integration security requirements

6.4.2 Phase 2: Design

Secure Architecture Review:

- Review system architecture for security weaknesses
- Apply secure design patterns
- Plan for secure data storage and transmission
- Design authentication and authorization mechanisms
- Identify trust boundaries and data flows

Threat Modeling:

- Identify potential threats using STRIDE methodology
- Assess attack surface and entry points
- Prioritize threats by risk level
- Define security controls to mitigate threats
- Document threat model and mitigation strategies

6.4.3 Phase 3: Development

Secure Coding Standards:

- Input Validation: Validate and sanitize all user inputs
- Output Encoding: Properly encode output to prevent injection
- Authentication: Use strong authentication mechanisms
- Session Management: Secure session handling and timeout
- Error Handling: Secure error messages, no sensitive data exposure
- Cryptography: Use approved algorithms and key management
- Logging: Log security events without sensitive data

Specific Practices:

- Use parameterized queries to prevent SQL injection
- Implement HTTPS/TLS for all network communications
- Store passwords with strong hashing (bcrypt, Argon2)
- Validate file uploads (type, size, content)

Document Version: 1.0

Effective Date: February 18, 2026

Review Date: February 18, 2027

Owner: FNE Enterprises (ABN 16 169 645 217)

MapRun - FNE Enterprises

- Implement rate limiting for APIs
- Use secure random number generation
- Avoid hardcoded credentials or secrets

Code Review:

- Peer review for all code changes
- Focus on security vulnerabilities during review
- Use security checklist for reviews
- Document review outcomes

Version Control:

- All code stored in Git version control
- Feature branches for new development
- Protected main branch requiring review
- Commit messages describe security-relevant changes

6.4.4 Phase 4: Testing

Security Testing Types:

- Static Analysis: Automated code scanning for vulnerabilities
- Dynamic Analysis: Runtime testing for security flaws
- Dependency Scanning: Check third-party libraries for known vulnerabilities
- Manual Testing: Security-focused test cases
- Penetration Testing: Annual testing by internal or external testers

Testing Activities:

Activity	Frequency
Automated static analysis	Every commit/build
Dependency vulnerability scan	Weekly
Manual security testing	Every release
API security testing	Every release
Penetration testing	Annually

Table 7: Security testing schedule

Vulnerability Management:

- Vulnerabilities triaged by severity
- Critical issues must be fixed before release

MapRun - FNE Enterprises

- High issues fixed within sprint or next release
- Medium/Low issues tracked and scheduled
- Retest after fixes applied

6.4.5 Phase 5: Deployment

Pre-Deployment Security Checks:

- Final security scan completed
- All critical/high vulnerabilities resolved
- Security test results reviewed
- Configuration hardening verified
- Secrets and credentials properly managed

Secure Deployment Process:

- Use infrastructure as code for consistency
- Apply principle of least privilege to services
- Enable security logging and monitoring
- Configure firewalls and network security
- Remove debug code and verbose error messages
- Verify TLS certificates and encryption

App Store Deployment:

- Follow platform security guidelines (Apple, Google)
- Request minimum necessary permissions
- Implement certificate pinning where appropriate
- Enable app transport security
- Code signing with protected keys

6.4.6 Phase 6: Operations and Maintenance

Ongoing Security Activities:

- Monitor security logs for anomalies
- Apply security patches promptly (see Patch Management)
- Conduct periodic security assessments
- Update dependencies regularly
- Review access controls quarterly

Incident Response:

- Follow Incident Response Plan (EV9)

Document Version: 1.0

Effective Date: February 18, 2026

Review Date: February 18, 2027

Owner: FNE Enterprises (ABN 16 169 645 217)

MapRun - FNE Enterprises

- Analyze security incidents for root cause
- Implement fixes and preventive measures
- Update security controls based on lessons learned

6.5 Security Tools and Resources

Development Tools:

- Static Analysis: SonarQube, ESLint security plugins, or similar
- Dependency Scanning: Dependabot, or similar
- Secret Scanning: git-secrets, TruffleHog, or similar
- IDE Plugins: Security linters and real-time analysis

Security Resources:

- OWASP Top 10 (Web and Mobile)
- OWASP ASVS (Application Security Verification Standard)
- CWE Top 25 (Common Weakness Enumeration)
- Platform security guidelines (Apple, Google, AWS)

6.6 Third-Party Components

Selection Criteria:

- Active maintenance and security updates
- Good security track record
- Reputable source/maintainer
- Open source with community review preferred
- License compatibility

Ongoing Management:

- Maintain inventory of third-party components
- Monitor for security advisories
- Update promptly when vulnerabilities disclosed
- Evaluate alternatives if component abandoned
- Document component purpose and usage

6.7 Security Training

Developer Training:

- Secure coding practices training annually

MapRun - FNE Enterprises

- Platform-specific security guidance (iOS, Android)
- Common vulnerability awareness (OWASP Top 10)
- Security tool usage training
- Incident response procedures

Training Resources:

- Online security training platforms
- Security documentation and guidelines
- Security community resources and blogs
- Conference talks and webinars

6.8 Security Champions

Role:

- Advocate for security in development process
- Provide security guidance to development team
- Stay current on security threats and best practices
- Coordinate security testing and reviews
- Liaise with external security resources

Responsibilities:

- Lead threat modeling sessions
- Review security test results
- Coordinate penetration testing
- Maintain security documentation
- Promote security awareness

6.9 Secrets Management

Credential Protection:

- Never commit credentials to version control
- Use environment variables for configuration
- Implement secrets management solution (AWS Secrets Manager or similar)
- Rotate credentials regularly
- Use different credentials for each environment

API Keys and Tokens:

- Restrict API keys by IP or domain

MapRun - FNE Enterprises

- Use short-lived tokens where possible
- Implement key rotation process
- Monitor API key usage
- Revoke keys immediately when compromised

6.10 Security Metrics

Development Metrics:

- Number of security vulnerabilities by severity
- Time to fix vulnerabilities
- Security test coverage
- Dependency vulnerability count
- Code review completion rate

Process Metrics:

- Security training completion
- Threat models created
- Penetration test findings
- Security incidents related to code
- Policy compliance rate

6.11 Compliance and Auditing

Code Auditing:

- Quarterly security-focused code reviews
- Annual comprehensive security assessment
- Audit logs maintained for all deployments
- Track adherence to secure coding standards

Documentation:

- Security requirements documented for each feature
- Threat models maintained and updated
- Security test results archived
- Vulnerability tracking and resolution documented
- Process compliance evidence maintained

6.12 Emergency Security Updates

MapRun - FNE Enterprises

Process for Critical Security Issues:

1. Immediate assessment and priority assignment
2. Fast-track development and testing
3. Expedited security review
4. Rapid deployment following minimal safe testing
5. Enhanced post-deployment monitoring
6. Post-incident review within 7 days

Communication:

- Notify users of critical security updates
- Provide clear guidance on update urgency
- Transparent communication about resolved issues
- Credit security researchers appropriately

6.13 Continuous Improvement

- Review SSDLC process quarterly
- Incorporate lessons from security incidents
- Update based on new threats and vulnerabilities
- Adopt new security tools and practices
- Benchmark against industry standards
- Solicit feedback from development team

6.14 Process Review

This SSDLC process is reviewed and updated:

- Quarterly for minor updates
 - Annually for comprehensive review
 - After significant security incidents
 - When adopting new technologies or platforms
 - Based on changes to compliance requirements
-