Security Incident Response Procedure

Document ID: SEC-IRP-001
Version: 1.0
Effective Date: January 16, 2026
Owner: Quality Management
Classification: Internal

---

1. Purpose

This procedure establishes a structured approach for detecting, responding to, and recovering from security incidents affecting the PSScript platform. It ensures consistent handling of security events while minimizing impact and enabling continuous improvement.

---

2. Scope

This procedure applies to: - All security incidents affecting PSScript infrastructure - All team members who discover or respond to security incidents - Third-party integrations and services - Development, staging, and production environments

---

3. Definitions

Term	Definition
Security Incident	Any event that compromises confidentiality, integrity, or availability of systems or data
Security Event	An observable occurrence relevant to security (may or may not be an incident)
Incident Commander	Person responsible for coordinating incident response
CAPA	Corrective and Preventive Action
RCA	Root Cause Analysis
MTTR	Mean Time to Recovery

---

4. Incident Classification

4.1 Severity Levels

Level	Name	Description	Response Time	Examples
S1	Critical	Active exploitation, data breach, system compromise	Immediate (< 15 min)	Active attack, credential leak, ransomware
S2	High	Significant vulnerability, potential data exposure	< 1 hour	Auth bypass, SQL injection, privilege escalation
S3	Medium	Limited vulnerability, no active exploitation	< 4 hours	XSS, CSRF, misconfiguration
S4	Low	Minor issue, minimal risk	< 24 hours	Info disclosure, deprecated protocols

4.2 Incident Categories

- **Access Control**: Unauthorized access, authentication bypass, privilege escalation
- **Data Breach**: Unauthorized data access, exfiltration, exposure
- **Malware**: Virus, ransomware, cryptominer, backdoor
- **Denial of Service**: Service disruption, resource exhaustion
- **Injection**: SQL injection, command injection, code injection
- **Configuration**: Misconfiguration, exposed credentials, insecure defaults
- **Supply Chain**: Compromised dependencies, malicious packages

---

5. Incident Response Phases

Phase 1: Detection & Identification

Objective: Recognize and confirm security incidents

Detection Sources: - [ ] Automated monitoring alerts (logs, metrics) - [ ] Security scanning tools (Trivy, TruffleHog) - [ ] User/customer reports - [ ] CI/CD pipeline security checks - [ ] Third-party vulnerability disclosures - [ ] Penetration testing findings

Identification Checklist:

```
□ What systems/data are affected?
□ When did the incident start?
□ Is it ongoing or contained?
□ What is the initial severity assessment?
□ Who discovered it?
```

Action: Create GitHub issue using Security Vulnerability template

---

Phase 2: Containment

Objective: Limit the scope and impact of the incident

Immediate Containment (S1/S2):

```
# 1. Isolate affected services
docker-compose stop <affected-service>

# 2. Revoke compromised credentials
# Rotate API keys, tokens, passwords immediately

# 3. Block malicious IPs (if applicable)
# Update firewall/WAF rules

# 4. Preserve evidence
docker logs <container> > incident-logs-$(date +%Y%m%d-%H%M%S).tx
```

Short-term Containment: - Deploy temporary patches or workarounds - Enable enhanced logging/monitoring - Implement additional access controls - Notify affected users if required

Evidence Preservation: - [ ] Capture system logs - [ ] Document timeline of events - [ ] Screenshot/record anomalies - [ ] Preserve affected artifacts

---

Phase 3: Eradication

Objective: Remove the threat and underlying vulnerability

Eradication Steps: 1. **Identify root cause** - What vulnerability was exploited? 2. **Develop fix** - Create and test patch 3. **Code review** - Security-focused review of fix 4. **Deploy fix** - Follow standard deployment procedures 5. **Verify remediation** - Confirm vulnerability is resolved

Verification Checklist:

```
□ Vulnerability scan shows clean
□ Exploit attempt fails
□ Security tests pass
□ No residual malicious artifacts
```

Phase 4: Recovery

Objective: Restore normal operations safely

Recovery Steps: 1. **Validate systems** - Confirm integrity of restored services 2. **Restore from backup** - If data corruption occurred 3. **Monitor closely** - Enhanced monitoring for 48-72 hours 4. **Gradual restoration** - Phased service restoration if needed

Recovery Verification: - [ ] All services operational - [ ] Data integrity confirmed - [ ] Security controls functioning - [ ] No signs of persistence

---

Phase 5: Post-Incident Activities

Objective: Learn from the incident and prevent recurrence

5.1 Root Cause Analysis (RCA)

RCA Template:

```
## Incident Summary
- **Incident ID:** SEC-YYYY-NNN
- **Date:** YYYY-MM-DD
- **Severity:** S1/S2/S3/S4
- **Duration:** X hours

## Timeline
| Time | Event |
|------|-------|
| HH:MM | Initial detection |
| HH:MM | Containment started |
| HH:MM | Fix deployed |
| HH:MM | Recovery complete |

## Root Cause
[Detailed description of the underlying cause]

## Contributing Factors
1. [Factor 1]
2. [Factor 2]

## Impact
- Systems affected:
- Data exposed:
- Users impacted:

## Resolution
[Description of the fix implemented]
```

5.2 CAPA (Corrective and Preventive Action)

Action Type	Description	Owner	Due Date	Status
Corrective	[Fix the immediate issue]			
Preventive	[Prevent similar issues]			

5.3 Lessons Learned

Post-Incident Review Meeting: - Schedule within 5 business days of incident closure - Include all responders and stakeholders - Document in `/docs/incidents/` directory

Review Questions: 1. What worked well in our response? 2. What could be improved? 3. Were our tools and processes adequate? 4. What training or resources are needed?

---

6. Communication

6.1 Internal Communication

Audience	When	Method	Content
Incident Team	Immediately	Slack/Call	Technical details, status
Leadership	S1/S2: Immediate, S3/S4: Daily	Email	Summary, impact, ETA
All Staff	After containment	Email	Awareness, any actions needed

6.2 External Communication

Scenario	Requirement	Timeline
Data breach affecting users	Notification required	Within 72 hours (GDPR)
Service disruption	Status page update	Immediately
Vulnerability disclosure	Coordinated disclosure	After fix deployed

Communication Templates: See `/docs/templates/incident-communications/`

---

7. Roles and Responsibilities

Role	Responsibilities
Incident Commander	Overall coordination, decision-making, communication
Technical Lead	Technical analysis, fix development, deployment
Security Lead	Threat assessment, evidence preservation, compliance
Communications Lead	Internal/external communications, status updates

---

8. Tools and Resources

8.1 Detection Tools

- GitHub Actions security scanning (Trivy, TruffleHog)
- Application logs (`/logs/` directory)
- Docker container health checks

8.2 Response Tools

```
# View container logs
docker-compose logs --tail=1000 <service>

# Check for suspicious processes
docker exec <container> ps aux

# Network connections
docker exec <container> netstat -tuln

# Database audit
psql -h localhost -U postgres -d psscript -c "SELECT * FROM pg_sta
```

8.3 Documentation

- Incident issues: GitHub Issues (Security Vulnerability template)
- RCA documents: `/docs/incidents/`
- Runbooks: `/docs/runbooks/`

---

9. Metrics and Reporting

9.1 Key Metrics

Metric	Target	Description
MTTD	< 1 hour	Mean Time to Detect
MTTR	S1: < 4h, S2: < 24h	Mean Time to Recovery
Incidents/Month	Trending down	Total incident count
Repeat Incidents	0	Same root cause recurrence

9.2 Quarterly Review

- Review all incidents from quarter
- Analyze trends and patterns
- Update procedures as needed
- Report to leadership

---

10. Training and Exercises

10.1 Training Requirements

Role	Training	Frequency
All developers	Security awareness	Annual
Incident responders	Incident response procedures	Semi-annual
Leadership	Executive briefing on security	Annual

10.2 Exercises

- **Tabletop exercises:** Quarterly
- **Simulated incidents:** Semi-annual
- **Full response drills:** Annual

---

11. Document Control

Version	Date	Author	Changes
1.0	2026-01-16	QMR	Initial release

Appendix A: Quick Reference Card

```
┌─────────────────────────────────────────────────────────┐
│           SECURITY INCIDENT QUICK REFERENCE             │
├─────────────────────────────────────────────────────────┤
│ 1. DETECT: Confirm the incident is real                 │
│ 2. CLASSIFY: Assign severity (S1-S4)                    │
│ 3. REPORT: Create GitHub issue (Security template)      │
│ 4. CONTAIN: Stop the bleeding                           │
│ 5. NOTIFY: Alert appropriate stakeholders               │
│ 6. ERADICATE: Remove the threat                         │
│ 7. RECOVER: Restore normal operations                   │
│ 8. REVIEW: Conduct RCA, document lessons learned        │
└─────────────────────────────────────────────────────────┘

SEVERITY RESPONSE TIMES:
  S1 Critical: Immediate (< 15 min)
  S2 High:     < 1 hour
  S3 Medium:   < 4 hours
  S4 Low:      < 24 hours
```

Appendix B: Incident Checklist

```
INCIDENT RESPONSE CHECKLIST

□ Incident detected and confirmed
□ Severity level assigned: S___
□ GitHub issue created
□ Incident Commander assigned
□ Stakeholders notified
□ Containment actions taken
□ Evidence preserved
□ Root cause identified
□ Fix developed and tested
□ Fix deployed to production
□ Recovery verified
□ Enhanced monitoring enabled
□ Post-incident review scheduled
□ RCA document completed
□ CAPA actions assigned
□ Incident closed
```

Generated 2026-01-16 21:23 UTC