LexOS Production Security Checklist

🔒 Comprehensive Security Hardening for H100 GPU Infrastructure

This checklist ensures that your LexOS production deployment meets enterprise-grade security standards.

Pre-Deployment Security

1. Infrastructure Security

- [ ] **Kubernetes Cluster Hardening**
- [ ] RBAC enabled and configured with least privilege
- [ ] Pod Security Standards enforced (restricted profile)
- [ ] Network policies implemented
- [ ] Admission controllers configured (OPA Gatekeeper recommended)
- [ ] etcd encryption at rest enabled
- [ ] API server audit logging enabled
- [ ] Kubelet security configuration hardened
- [ ] **Node Security**
- [ ] OS hardening applied (CIS benchmarks)
- [ ] Unnecessary services disabled
- [ ] SSH access restricted and key-based only
- [ ] Firewall rules configured (iptables/ufw)
- [ ] Intrusion detection system deployed
- [ ] Regular security updates scheduled
- [ ] **GPU Node Specific**
- [ ] NVIDIA driver security updates applied
- [ ] GPU access restricted to authorized containers only
- [ ] MIG (Multi-Instance GPU) configured if needed
- [ ] GPU memory isolation verified

2. Container Security

- [ ] **Image Security**
- [ ] Base images from trusted sources only
- [ ] Container images scanned for vulnerabilities (Trivy, Snyk)
- [ ] No secrets embedded in images
- [ ] Images signed and verified
- [ ] Minimal base images used (distroless preferred)
- [ ] Regular image updates scheduled

- [ ] **Runtime Security**
- [ ] Containers run as non-root user
- [ ] Read-only root filesystem where possible
- [ ] Security contexts properly configured
- [ ] Capabilities dropped (ALL) and only required ones added
- [ ] seccomp profiles applied
- [ ] AppArmor/SELinux policies configured

3. Network Security

- [ ] **Network Segmentation**
- [ ] Network policies implemented for all namespaces
- [ ] Ingress traffic properly filtered
- [ ] East-west traffic encryption (service mesh)
- [ ] DNS policies configured
- [ ] Private container registry access only
- [ ] **TLS/SSL Configuration**
- [ ] Valid SSL certificates installed
- [ ] TLS 1.2+ enforced (TLS 1.3 preferred)
- [ ] Strong cipher suites configured
- [ ] Certificate rotation automated
- [ ] HSTS headers enabled
- [ ] Certificate transparency monitoring

Application Security

1. Authentication & Authorization

- [ ] **Identity Management**
- [ ] Strong password policies enforced
- [ ] Multi-factor authentication (MFA) enabled
- [ ] JWT tokens with proper expiration
- [ ] Refresh token rotation implemented
- [ ] Session management secure
- [ ] Account lockout policies configured
- [ ] **API Security**
- [ ] API rate limiting implemented
- [ ] Input validation on all endpoints
- [ ] SQL injection protection
- [ ] XSS protection headers
- [ ] CSRF protection enabled
- [ ] API versioning strategy implemented

2. Data Protection

- [ ] **Encryption**

- [ ] Data at rest encryption (database, storage)
- [ ] Data in transit encryption (TLS)
- [ ] Encryption key management (KMS)
- [ ] Database connection encryption
- [ ] Backup encryption enabled
- [ ] **Data Privacy**
- [ ] PII data identification and protection
- [ ] Data retention policies implemented
- [ ] GDPR compliance measures
- [ ] Data anonymization for non-prod environments
- [ ] Audit trails for data access

3. Secrets Management

- [ ] **Kubernetes Secrets**
- [ ] All secrets stored in Kubernetes secrets (not ConfigMaps)
- [ ] Secrets encrypted at rest in etcd
- [ ] External secrets operator configured (if applicable)
- [ ] Secret rotation policies implemented
- [ ] No hardcoded secrets in code or configs
- [ ] **External Secrets**
- [ ] HashiCorp Vault or similar KMS integrated
- [ ] API keys rotated regularly
- [ ] Database credentials managed externally
- [ ] SSL certificates managed via cert-manager
- [ ] Service account keys rotated

Monitoring & Logging Security

1. Security Monitoring

- [ ] **Threat Detection**
- [ ] Runtime security monitoring (Falco)
- [ ] Anomaly detection configured
- [ ] Intrusion detection system active
- [ ] Vulnerability scanning automated
- [ ] Security event correlation
- [ ] **Compliance Monitoring**
- [ ] CIS benchmark compliance monitoring
- [ ] Policy violations alerting
- [ ] Configuration drift detection
- [ ] Compliance reporting automated

2. Audit Logging

- [ ] **Comprehensive Logging**
- [ ] Kubernetes audit logs enabled
- [ ] Application security logs captured
- [ ] Authentication events logged
- [ ] API access logs maintained
- [ ] Database access logs enabled
- [ ] **Log Security**
- [ ] Log integrity protection
- [ ] Centralized log management
- [ ] Log retention policies
- [ ] Log access controls
- [ ] Log encryption in transit and at rest

Mobile App Security

1. Application Security

- [ ] **Code Protection**
- [ ] Code obfuscation enabled
- [ ] Anti-tampering measures implemented
- [ ] Root/jailbreak detection
- [ ] Debug detection and prevention
- [ ] Certificate pinning implemented
- [ ] **Data Protection**
- [ ] Local data encryption
- [ ] Secure storage for sensitive data
- [ ] Biometric authentication support
- [ ] Session timeout configured
- [ ] Screen recording prevention

2. Communication Security

- [ ] **API Communication**
- [ ] Certificate pinning for API calls
- [ ] Request/response encryption
- [ ] API key protection
- [ ] Token-based authentication
- [ ] Secure WebSocket connections

Operational Security

1. Access Control

- [ ] **Administrative Access**
- [ ] Privileged access management (PAM)

- [ ] Just-in-time access for sensitive operations
- [ ] Administrative actions logged and monitored
- [ ] Separation of duties implemented
- [ ] Regular access reviews conducted
- [ ] **Service Accounts**
- [ ] Minimal permissions for service accounts
- [ ] Service account token rotation
- [ ] Unused service accounts removed
- [ ] Service account activity monitored

2. Incident Response

- [ ] **Preparation**
- [ ] Incident response plan documented
- [ ] Security team contact information updated
- [ ] Escalation procedures defined
- [ ] Communication templates prepared
- [ ] Recovery procedures tested
- [ ] **Detection & Response**
- [ ] Security monitoring alerts configured
- [ ] Automated response procedures
- [ ] Forensic capabilities available
- [ ] Backup and recovery procedures tested
- [ ] Post-incident review process defined

Compliance & Governance

1. Regulatory Compliance

- [ ] **Data Protection Regulations**
- [ ] GDPR compliance implemented
- [ ] CCPA compliance measures
- [ ] Data processing agreements in place
- [ ] Privacy policy updated
- [ ] Consent management implemented
- [ ] **Industry Standards**
- [ ] SOC 2 Type II compliance (if applicable)
- [ ] ISO 27001 alignment
- [ ] NIST Cybersecurity Framework adoption
- [ ] Industry-specific regulations addressed

2. Security Governance

- [ ] **Policies & Procedures**
- [ ] Information security policy

- [ ] Acceptable use policy
- [ ] Data classification policy
- [ ] Incident response procedures
- [ ] Business continuity plan
- [ ] **Training & Awareness**
- [ ] Security awareness training for all staff
- [ ] Phishing simulation exercises
- [ ] Secure coding training for developers
- [ ] Regular security updates communicated

Continuous Security

1. Regular Assessments

- [ ] **Security Testing**
- [ ] Penetration testing (quarterly)
- [ ] Vulnerability assessments (monthly)
- [ ] Code security reviews
- [ ] Configuration audits
- [ ] Third-party security assessments
- [ ] **Monitoring & Metrics**
- [ ] Security KPIs defined and tracked
- [ ] Security dashboard implemented
- [ ] Regular security reports generated
- [ ] Trend analysis performed
- [ ] Continuous improvement process

2. Updates & Maintenance

- [ ] **Patch Management**
- [ ] Regular security updates applied
- [ ] Patch testing procedures
- [ ] Emergency patching procedures
- [ ] Patch compliance monitoring
- [ ] Rollback procedures tested
- [ ] **Security Reviews**
- [ ] Monthly security reviews scheduled
- [ ] Quarterly security assessments
- [ ] Annual security audits
- [ ] Continuous threat modeling
- [ ] Security architecture reviews

H100 GPU Specific Security

1. GPU Security

- [ ] **Hardware Security**
- [ ] GPU firmware updated
- [ ] Secure boot enabled (if supported)
- [ ] Hardware attestation configured
- [ ] Physical security measures
- [ ] GPU access logging
- [ ] **Workload Isolation**
- [ ] MIG partitioning configured securely
- [ ] GPU memory isolation verified
- [ ] Compute isolation between workloads
- [ ] GPU resource quotas enforced
- [ ] Cross-tenant isolation verified

2. AI/ML Security

- [ ] **Model Security**
- [ ] Model integrity verification
- [ ] Model access controls
- [ ] Training data protection
- [ ] Inference request validation
- [ ] Model versioning and rollback
- [ ] **Data Pipeline Security**
- [ ] Training data encryption
- [ ] Data lineage tracking
- [ ] Feature store security
- [ ] Model registry security
- [ ] Experiment tracking security

Security Validation

1. Automated Testing

```
# Run security scan
kubectl run security-scan --rm -i --restart=Never \
  --image=aquasec/trivy:latest -- \
  trivy k8s --report summary cluster

# Check pod security standards
kubectl run pss-check --rm -i --restart=Never \
  --image=kubesec/kubesec:latest -- \
  kubesec scan /dev/stdin < k8s/lexos-deployment.yaml

# Network policy validation
kubectl run netpol-test --rm -i --restart=Never \
  --image=nicolaka/netshoot -- \
  nc -zv lexos-api-service.lexos.svc.cluster.local 8000
```

2. Manual Verification

```
# Check RBAC configuration
kubectl auth can-i --list --as=system:serviceaccount:lexos:lexos-api

# Verify secrets are not in plain text
kubectl get secrets -n lexos -o yaml | grep -v "data:"

# Check security contexts
kubectl get pods -n lexos -o jsonpath='{.items[*].spec.securityContext}'

# Verify network policies
kubectl get networkpolicies -n lexos

# Check for privileged containers
kubectl get pods -n lexos -o jsonpath='{.items[*].spec.containers[*].securityContext.pr
ivileged}'
```

Security Incident Response

1. Immediate Response

```
# Isolate compromised pod
kubectl patch deployment lexos-api -n lexos -p '{"spec":{"replicas":0}}'

# Collect forensic data
kubectl logs -n lexos -l app=lexos-api --previous > incident-logs.txt
kubectl get events -n lexos --sort-by='.lastTimestamp' > incident-events.txt

# Block suspicious traffic
kubectl apply -f - <<EOF
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: emergency-isolation
  namespace: lexos
spec:
  podSelector:
    matchLabels:
      app: lexos-api
  policyTypes:
  - Ingress
  - Egress
EOF
```

2. Recovery Procedures

```
# Restore from clean backup
velero restore create incident-recovery-$(date +%Y%m%d) \
  --from-backup lexos-daily-clean-backup

# Rotate all secrets
kubectl delete secret lexos-secrets -n lexos
kubectl apply -f k8s/secrets-rotated.yaml

# Update all container images
kubectl set image deployment/lexos-api lexos-api=ghcr.io/lexhelios/lexworking:secure-$
(date +%Y%m%d) -n lexos
```

Checklist Completion

Sign-off Requirements

- [ ] **Security Team Approval**
- Security Officer: __ **Date:** _____
- DevOps Lead: __ **Date:** _____
- Platform Owner: __ **Date:** _____
- [ ] **Compliance Verification**
- Legal Review: __ **Date:** _____
- Compliance Officer: __ **Date:** _____
- Risk Assessment: __ **Date:** _____

Documentation

- [ ] Security assessment report completed
- [ ] Risk register updated
- [ ] Incident response plan tested
- [ ] Security runbooks updated
- [ ] Compliance documentation filed

🔒 **Security is not a destination, but a continuous journey. Regular reviews and updates of this checklist are essential for maintaining a strong security posture.**

Last Updated: August 2025
Version: 2.0.0
Classification: Internal Use Only