## Statistics Based on the Attack

- **Total Compromised Packages**: 500+
- **Weekly Downloads Affected**: Billions
- **Attack Duration**: 9 days (before containment)
- **Geographic Impact**: Global
- **Sectors Affected**: All (universal npm usage)
- **Enterprise Vendors Hit**: CrowdStrike, multiple others
- **First AI-Assisted Supply Chain Attack**: Confirmed by Unit42

## Detection

### Phase 1: Immediate Threat Assessment

### A. Repository Scan

Search your GitHub Enterprise for these indicators:

*Malicious Repository Names:*

- Repository name: "Shai-Hulud"
- Description: "Shai-Hulud Repository"
- Content: data.json (base64-encoded stolen credentials)

*Migration Repositories:*

- Repository pattern: "*-migration"
- Description: "Shai-Hulud Migration"
- Status: Previously private repos made public

### B. Branch Analysis

Search Command: Look for branches named **shai-hulud** across all repositories

## C. Workflow Files

Critical Files to Find:

- .github/workflows/shai-hulud-workflow.yml
- .github/workflows/shai-hulud.yaml
- Any workflows connecting to webhook.site/bb8ca5f6-4175-45d2-b042-fc9ebb8170b7

# Phase 2: Deep Package Analysis

## A. Package.json Audit

*Scan all repositories for these compromised packages:*

High-Priority Packages:

- @ctrl/tinycolor@4.1.1, @4.1.2 (2.2M weekly downloads)
- @crowdstrike/* packages (any version after Sep 16, 2025)
- @nativescript-community/* packages
- Any package with postinstall: "node bundle.js"

B. File Hash Verification

*Malicious bundle.js Hashes:*

- 46faab8ab153fae6e80e7cca38eab363075bb524edd79e42269217a083628f09 (CrowdStrike variant)
- b74caeaa75e077c99f7d44f46daaf9796a3be43ecf24f2a1fd381844669da777
- dc67467a39b70d1cd4c1f7f7a459b35058163592f4a9e8fb4dffcbba98ef210c
- 4b2399646573bb737c4969563303d8ee2e9ddbd1b271f1ca9e35ea78062538db

Phase 3: Behavioral Detection (Ongoing)

A. Network Monitoring

*Monitor for connections to:*

- webhook.site (any subdomain)
- 5.199.166.1:31337 (reverse shell endpoint)
- Unusual GitHub API activity patterns

B. Process Monitoring

Watch for:

- trufflehog process executions
- Automated npm publish activities
- Unusual CI/CD pipeline triggers

# Mitigation

**Immediate Actions Required:**

1. Rotate ALL developer credentials immediately
2. Scan GitHub organization for malicious repositories
3. Check production applications for compromised packages
4. Implement emergency monitoring for ongoing attacks

## Priority 1 :

- GitHub Personal Access Tokens
- npm authentication tokens
- SSH keys used for deployment

## Priority 2 :

- AWS/GCP/Azure API keys
- Database connection strings
- Third-party service API keys
- CI/CD pipeline secrets

- GitHub Enterprise:
  - Enforce 2FA for all developer accounts (mandatory)
  - Enable GitHub Advanced Security features
  - Implement branch protection rules
  - Restrict workflow permissions
  - Enable audit logging and monitoring
- Development Workflow:
  - Migrate to Trusted Publishing (removes token dependency)
  - Implement dependency pinning
  - Set up automated security scanning
  - Add SBOM (Software Bill of Materials) generation
  - Enable real-time vulnerability alerts

## Priority 3 :

- Delete any "Shai-Hulud" repositories immediately
- Remove any "*-migration" repositories
- Delete "shai-hulud" branches from all repositories
- Remove malicious workflow files
- Review and revert any unauthorized commits since Sep 14, 2025

## Priority 4 :

- Identify all applications using npm packages
- Check package-lock.json for compromised packages
- Remove/downgrade affected packages to pre-attack versions
- Clear npm cache: npm cache clean --force
- Rebuild applications from clean sources
- Redeploy with verified clean packages

# Risks If Compromised

## Data Exfiltration

- **Stolen Credentials**: GitHub tokens, npm tokens, cloud API keys, SSH keys
- **Source Code Exposure**: Private repositories made public
- **Secret Harvesting**: Environment variables, configuration files, CI/CD secrets
- **Cloud Account Compromise**: AWS/GCP/Azure access via stolen keys

## Infrastructure Compromise

- **CI/CD Pipeline Takeover**: Malicious workflows continue running
- **Production Deployment Risk**: Compromised packages in live applications
- **Lateral Movement**: SSH keys enable network access
- **Persistent Backdoors**: Self-replicating worm continues spreading

## Business Impact Assessment

### Financial Risks

- **Regulatory Compliance**: GDPR, SOX, HIPAA violations
- **Data Breach Costs**: Average $4.45M per breach (IBM 2023)
- **Business Disruption**: Production downtime and recovery costs
- **Legal Liability**: Customer data exposure lawsuits
- **Reputation Damage**: Loss of customer trust and market value

### Operational Risks

- **Supply Chain Contamination**: Malicious code in customer deliverables

- **Developer Productivity Loss**: Credential rotation and system rebuilds
- **Customer Impact**: Applications with compromised dependencies
- **Vendor Relationships**: Trust issues with package ecosystem