Niagara Enterprise Security

企业级安全保障体系与最佳实践白皮书

FIPS 140-2 | PKI 证书管理 | 企业审计 | TLS 1.3 | 组件签名

FIPS 140-2 验证加密	—	双因素认证 · AES-256 · PBKDF2
PKI 公钥基础设施	—	X.509证书 · 智能卡 · 自动续期
企业级审计追踪	—	全量 Audit Log · 防篡改 · SOC合规
组件化安全架构	—	代码签名 · 模块认证 · 最小权限

IT 合规体系 — Enterprise Security Standards

FIPS 140-2 | PKI 证书管理 | Enterprise Audit | TLS 1.3

1 FIPS 140-2 验证加密引擎

- RSA SecurID 双因素认证（FIPS 140 级别）
- 256-bit AES 传输与存储加密
- PBKDF2 密钥派生（SCRAM-SHA-256 握手协议）
- Cipher Suite 组策略自定义
- CryptoService 模块符合美国政府标准

2 PKI 公钥基础设施

- X.509 证书双向设备身份验证
- 内置 CA 证书信任存储管理
- 证书到期自动监控预警
- 每台设备独立证书（JACE / Edge）
- 证书吊销与更新工作流

3 企业审计与合规追踪

- 全量 Audit Log — 追踪每一次操作
- 审计处理失败自动告警
- 日志防篡改存储，符合 SOC / ISO 要求
- 密码强度强制：10位+大小写+数字
- 失败锁定防暴力破解
- 会话并发控制 + 临时账户自动过期

资料来源：Tridium FIPS 140 Configuration Guide | Hardening Guide 2023 | PKI Operations Manual

数据安全 — Code Signing | TLS | IAM | Audit Trail

供应链安全 | 全链路加密 | 身份认证与权限 | 审计与灾备

1 代码签名与供应链安全

- Workbench GUI 一键签名，无需命令行操作
- 签名链：CA证书（机构）→ 签名证书（身份）→ 模块
- CA证书导入 Niagara Trust Store 验证
- 未签名模块拒绝加载（安全策略强制执行）
- Module Manager 追踪版本/依赖/签名信息
- 签名即身份证明 + 完整性保证

2 端到端 TLS 1.3 加密

- 三通道加密：Fox TLS / Web TLS / Platform TLS
- HTTPS-Only 模式：拒绝所有明文 HTTP
- BACnet/SC 基于标准 PKI 的 TLS 1.3 加密
- oBIX / REST API 强制 HTTPS + SCRAM-SHA-256
- Cipher Suite 企业级自定义策略
- 兼容 Qualys 等第三方安全扫描工具

3 身份认证与访问控制（IAM）

- SAML 2.0 单点登录 — 一次认证，访问全部站点
- LDAP / Active Directory 企业目录对接
- RSA SecurID 双因素认证（FIPS 140 级别）
- RBAC 角色权限：精确到视图/菜单级
- 最小权限原则 — 超级用户不超过 2 人
- 程序对象必须超级用户授权才能执行

4 审计追踪与数据保护

- 全量 Audit Log：谁、何时、做了什么、从哪里
- 审计失败自动告警（可对接工单系统）
- 报警历史 oBIX 接口查询与确认
- 远程加密 Web Backup (.dist) → 本地 + NAS + 云端
- SRAM 掉电自动恢复 + NiMH 电池持久化
- 3-2-1 备份策略：3份/2介质/1异地

资料来源：Tridium Code Signing Guide | SSL Connectivity Guide | SAML SSO Guide | oBIX Authentication Guide

可维护性 — 组件化架构 | 已签名模块库 | 可复用 | 版本管理

告别万行大程序：小型规模块一键拖拽组装，每个模块签名验证

1 组件化替代单块大程序

- 每个模块 = 独立 .jar 文件，职责单一
- 模块间通过类型化 Slot 连接（输入/输出/配置）
- 从 Palette 调色板拖拽即可组装逻辑（可视化编程）
- 没有万行大程序 — 全是小组件拼装
- 每个模块可独立升级，不影响整站运行
- 跨项目复用已签名的组件库

2 代码签名确保信任链

- 每个模块必须签名才能部署到生产环境
- 签名链：CA证书 → 签名证书 → 目标模块
- 未签名模块 = 直接被安全策略拒绝
- Module Manager 记录版本、依赖、签名信息
- 团队成员共享签名组件库，无需重复造轮子
- 签名即审计：谁签了什么模块、什么时候

3 自动化灾备与恢复

- Web Backup 远程触发备份，无需现场操作
- 三重存储：本地磁盘 + NAS + 云端 (GitHub)
- SRAM 电容备份：断电后 RAM 数据自动恢复
- NiMH 电池持久化关键运行数据
- 单脚本并行备份多个站点
- 定时自动备份 + 邮件通知

传统单块程序 vs 组件化架构

Monolithic vs Component-Based Architecture

结构	一个文件几万行代码	独立 .jar 模块+类型化接口
复用性	复制粘贴到每个项目	共享 Palette 组件库
安全性	无法签名，全量访问	每个模块签名+权限管控
升级	替换整个程序，风险极高	独立升级单个模块
测试	手工测试，牵一发全身	每个组件可独立验证
协作	单人瓶颈，无法并行	多人并行开发不同模块
审计	谁改了什么？	签名历史+版本号可追溯

核心结论

Niagara 的组件化架构从根本上解决了"万行大程序"问题。每个组件都是经过签名、有版本号、可独立替换的积木块——支持多人并行开发、CI/CD 流水线、企业级安全治理。

资料来源：Tridium Module Development Guide | Module Manager | Code Signing Guide | Web Backup Guide

集成与开发 — REST API | BQL/SQL | SNMP | Grafana | CI/CD

对接现有 IT 工具栈 | 开放协议 | 无厂商锁定

1 REST API 与数据对接

- oBIX REST API：读写点值、查询历史、报警管理
- BQL (Baja Query Language)：SQL 风格实时数据查询 SELECT/WHERE
- 自定义 REST 端点 (BRestApiService)：任意 App 直连
- JSON/XML 输出，Web 前端可直接消费
- Node-RED / Python / Postman 等任何 HTTP 客户端
- CSV/ODBC 导出，对接 Tableau / Power BI / Excel

2 IT 基础设施完美兼容

- SNMP v2c/v3：对接企业网管（Nagios / Zabbix）
- MQTT：对接 IoT 云平台（AWS IoT / Azure IoT Hub）
- Modbus TCP/RTU：通用工业设备协议
- BACnet/IP & BACnet/SC：楼宇设施标准协议
- LDAP / Active Directory：企业统一身份源
- Syslog：审计日志对接 SIEM（Splunk / ELK）

3 监控与可视化

- Grafana：通过 REST API 或 InfluxDB 桥接展示数据
- Prometheus：自定义 Exporter 采集 Niagara 指标
- Kibana / ELK：报警历史与审计日志可视化
- oBIX Web Service：浏览器实时仪表盘（无需插件）
- Px Views：Workbench 内零代码 HMI 面板
- Report.px：定时 PDF/Excel 报表导出

4 开发者工作流与 CI/CD

- Java / BajaScript SDK：IntelliJ / Eclipse 开发模块
- Maven / Gradle 构建管道编译打包 Niagara 模块
- GitOps：站点配置（bog/px）存 GitHub 版本管理
- OpenClaw AI Agent：自然语言描述自动生成逻辑
- agent-browser 无头测试框架：Niagara UI 自动化测试
- Docker：容器化 Niagara 站（dev / staging / prod）

对接你现有的 IT 工具栈 — 不需要推倒重来