Open for Innovation
KNIME

KNIME Software Development Framework & Security

An Overview

Table of Contents

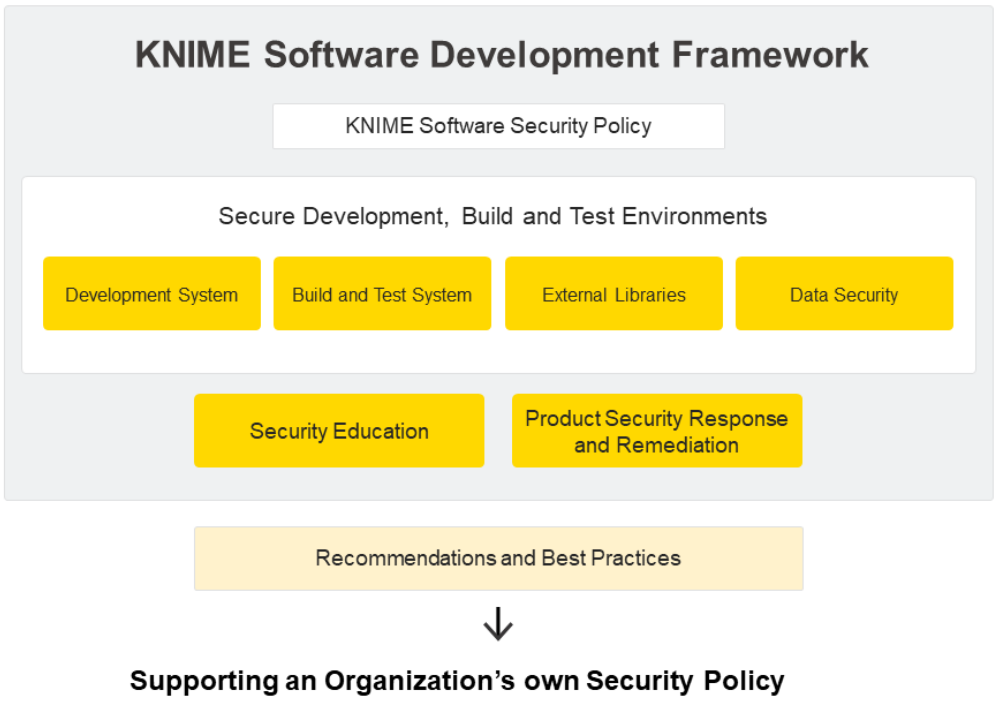
KNIME Software Development Framework & Security

KNIME firmly believes in open source and the power of the community. Our philosophy is to maintain and develop an open source platform containing all the functionality that any individual might require and to continue delivering extended functionality through our own work and that of the KNIME community. KNIME complements the open source KNIME Analytics Platform with licensed commercial software for increasing productivity and enabling collaboration.

To support our development goals, KNIME has implemented rigorous software engineering standards, a key part of these standards focuses on ensuring all aspects of software security. Our security policy includes a structured approach for exchange and the communication of security related topics with our extensive user community.

The extensive capabilities and choices available with KNIME Software means that each organization will want to implement KNIME Software capabilities based on their business as well as security requirements. To assist, KNIME provides software capabilities and best practices from our customers so that an organization can implement and test their own security compliance regime on their usage of KNIME Software.

This document gives an overview of the KNIME Software Development Framework and Security approach.


KNIME Software Development Framework
KNIME Software Security Policy
Secure Development, Build and Test Environments
Development System
Build and Test System
External Libraries
Data Security
Security Education
Product Security Response and Remediation
Recommendations and Best Practices
Supporting an Organization's own Security Policy

KNIME Software Security Policy

In the continuous development of our standards and guidelines for security, we take into consideration industry best practice, standards and guidance including:

- Atlassian Trust and Security Standards
- AWS Well Architected Framework - Security Pillar
- Center for Internet Security (CIS)
- Common Architectural Weakness Enumeration (CAWE)
- Common Attack Pattern Enumeration and Classification (CAPEC)
- Common Weakness Enumeration (CWE)
- GitHub Managing Security Standards
- International Organization of Standardization (ISO)
- Internet Engineering Task Force Standards
- Jenkins Security Guidelines
- Microsoft Security Development Lifecycle (SDL)
- National Institute of Standards and Technology (NIST)
- OWASP Software Assurance Maturity Model
- OWASP Top 10 Project
- SEI CERT Secure Coding Standards

Security Education

KNIME provides training for software developers and others involved in the software development process not only in the Software Security policy but also training in and access to our development environments designed to help detect and resolve common weaknesses and exposures in real time as they are writing code. Our security culture includes regular formal and informal exchanges.

Secure Development, Build and Test Environments

Our internal development and build/test systems are based on Atlassian Bitbucket for development and Jenkins from the Continuous Development Foundation for build and testing.

Development System

We follow the recommended best practices for development promoted by Atlassian and its community, including encrypted security and role/code responsibility best practices. We are also a full member of the Atlassian community and are automatically notified of any security issues surrounding the development system we use.

The open nature of KNIME means that we use other open source libraries. A review of known issues and, when necessary, any potential security risk issues surrounding an open source library are first identified before accessing that library. Only stable, well documented versions of the open source libraries are included in the design process. When the decision is taken to include another open source library which could involve security related topics, KNIME documents that fact and monitors these open source libraries to ensure that any security related and relevant notifications can be assessed and, when required, incorporated in the KNIME Security Response and Remediation procedure. Each new version of those libraries is equally assessed before inclusion in KNIME Software.

Build & Test System

Our build and testing system is based on Jenkins from the Continuous Development Foundation. It is an in-house implementation. No code ends up in release code unless our strict 6-eye policy is followed. Only developers who have been respectively trained in our Security Education (see above) can surface code. A second set of eyes from a qualified reviewer is required to complete a series of low-level code checks, which include security considerations, before the code is merged into release or pre-release branches. Finally, a third set of eyes carries out manual checks to ensure the code serves its intended purpose.

Testing involves unit tests, full build tests, and workflow tests. The testing procedure is extremely advanced. At every level of testing, security tests are implemented as required. Workflow testing is particularly important, since this guarantees not only backward compatibility but also better replicates what a user can do with the software.

KNIME Community Testing

Further testing is provided by our large user community. KNIME Analytics Platform is open source and is available in source code not only from KNIME but also via GitHub. This means we benefit from community testing, which often includes penetration and other security risk assessment procedures. All feedback resulting from community testing is fed back into our development and security process.

External Libraries

In all cases, the architecture is designed so that every library version used within KNIME is completely documented. Subsequently, when a workflow is developed, executed, and saved on KNIME Business Hub, all aspects of exactly which package and library versions were used is documented. As of the KNIME Summer 2020 release, additional techniques are provided to easily extract all metadata from the executed workflows so that it can be available for additional compliance and risk assessment purposes.

Data Security

Important architecture and design criteria are motivated by data security. KNIME Software and node development by KNIME do not capture or return any data used within KNIME workflows to KNIME AG. No nodes return data to any 3rd party unless that is clearly documented as a part of the functionality of the node. An example of this type of functionality would be when a node passes data on to a database, a cloud service or a 3rd party package as a part of its defined functionality.

KNIME Software also does not capture or return any information about the user of KNIME Software to KNIME AG. With a user's actively enabled permission, anonymized statistics about node usage are returned to KNIME.

Users and organization do have the ability to include nodes and/or extensions to their version of KNIME Software that are outside of the control of KNIME. Identifying such nodes and extensions by an organization is possible and access can be controlled (see below).

Product Security Response and Remediation

KNIME has a well-defined product security remediation and response policy as well as a defined security team. No matter where a security issue or concern is raised, either from within our testing procedure or notification via an external source, first a thorough risk assessment is always performed in a timely fashion to determine the impact on KNIME Software.

If a risk or exposure is determined by the security team, a remediation plan is identified and all relevant customers are notified. In the case of KNIME commercial offerings, this includes a high priority email notification and recommendation to the individual designated as official contact within the organization.

All known past vulnarabiilites and remediations can be found at:

https://www.knime.com/security/advisories

Reporting a Security Vulnerability

We strongly encourage you to report potential security vulnerabilities to our security team first, before disclosing them in a public forum.

Only contact the security team to report undisclosed security vulnerabilities in KNIME software products and services and manage the process of fixing such vulnerabilities. We cannot accept regular bug reports or other security-related queries. We will ignore mail that does not relate to an undisclosed security problem in KNIME software products and services.

Details on reporting a suspected vulnerability can be found at:

https://www.knime.com/security/policy

Supporting an Organization's Own Security Policy

The very nature of the flexible, extensible KNIME Software means that the software can be used in many ways. KNIME Analytics Platform is a platform that provides the mixing and matching of over 6500 executable nodes. KNIME Business Hub complements the platform with capabilities for collaboration, automation, and execution of interactive applications as well as providing the capabilities of running analytics as a service. This can be done either onsite, in cloud, on Edge or as a hybrid implentation that gives the customer full choice and control of how to implement KNIME

Many nodes are built and maintained by KNIME. Trusted community extensions follow the exact same build and testing protocols as the rest of KNIME Software and are even built on our build system to guarantee tests are carried out and documented.

Note that community nodes as well as nodes that an organization may create itself or obtain from a third party are not checked by KNIME for security considerations. In addition, many standard extensions – such as those allowing Python, Java, JavaScript, R, Spark and in-DB Execution for example – bring the full range of those programming languages (with their own risk and security considerations) into KNIME. That means KNIME itself cannot perform penetration tests or risk assessment tests that can confirm any individual customer's usage of the software since every customer will mix and match those nodes in different ways.

KNIME customers perform their own penetration tests. To date, none of these penetration tests have yet to reveal anything where KNIME needed to take remedial or notification action surrounding KNIME software.

Recommendations

KNIME customers have provided feedback to help other customer who have security, risk assessment or penetration test requirements. These suggestions are summarized here.

1. Manage the exact configurations of KNIME installed not only on KNIME Business Hub (whether installed on own infrastructure or own Cloud) but also on clients by having a centrally maintained function (possibly in the IT department) that manages exactly which version, extensions, and nodes are accessible.

   This can include a centrally maintained version of KNIME Analytics Platform on client machines. Update sites can be disabled for users and preferences managed through KNIME Business Hub. Different versions can be created for different user groups depending on their qualifications and security clearances.

2. An important best practice is to capture and document all production workflows. A production workflow that is saved contains all information about its exact state at time of execution, including - when required - all data and intermediate data. This is best done via KNIME Business Hub, which provides the capabilities to not only version workflows but to compare/contrast changes done to workflows over time by specific users.

3. By default, KNIME supports many scripting languages and extensions from within the platform. If you already use scripting languages such as Python, Java, JavaScript, R or Spark then you will have security policies in place surrounding these languages. The same security and compliance policies can be invoked for KNIME when those extensions are used.

4. If in-database, SPARK or CLOUD (AWS, Google, Azure, etc.) related nodes and services are used from within KNIME then the same security policies surrounding the data bases can be invoked for KNIME when those DB extensions are used.

5. For **penetration testing** as well as satisfying data access or **data privacy requirements**, an assessment of each workflow and the types of nodes used can be done. For nodes that allow “open code” then a further investigation of those nodes for user-coded violations of security policy can be performed.

6. Have security-aware domain experts create KNIME Components for securely packaging and encrypting sequences (workflows) and making them available to less qualified users. In this way any critical data, systems, or capabilities can be effectively packaged and presented.

7. Share and instantiate such KNIME Components via KNIME Business Hub in all workflows so that the most up to date version is always included within other workflows using those capabilities. This also allows for a controlled interface via an organization's security access setup to those users who have the need to use but should not have direct access themselves.

8. Many **ISO standards** surrounding security, such as ISO 27001, recommend guidelines for how an organization should install and maintain its own IT infrastructure in a secure way. They do not apply to any specific software vendor or software itself. The information provided by KNIME should allow you to integrate KNIME within your security framework and meet ISO standards.

9. Work with KNIME Support early to set up and resolve any specific security and IT topics – KNIME can usually recommend a specific best practice for a specific approach or setup.

10. Most importantly: incorporate your own security policy into the training of how to use KNIME, its nodes and its extensions, and how to build secure workflows.