

Experiment No 08

AIM

To perform Static Application Security Testing (SAST) using Snyk integrated with Jenkins and generate comprehensive vulnerability reports for a sample application.

TOOLS & TECHNOLOGIES

| Tool/Technology | Purpose | Version |
|-----------------|--|------------|
| Jenkins | Continuous Integration/Delivery automation | Latest LTS |
| Snyk | Vulnerability scanning and dependency analysis | Latest |
| Maven | Build automation and project management | 3.x |
| GitHub | Source code management and version control | Latest |

THEORY

DevSecOps & Security Integration

DevSecOps is a development paradigm that integrates security practices across the software development lifecycle (SDLC). It follows a **Shift-Left** methodology, detecting vulnerabilities early in the process—making fixes faster and less expensive.

Key Principles:

- **Automation-First Approach:** Automated security testing within CI/CD
- **Continuous Monitoring:** Real-time security oversight
- **Collaboration:** Dev, Sec, and Ops teams work together, not in silos
- **Risk-Based Decision Making:** Security priorities determined by risk assessments

Static Application Security Testing (SAST)

SAST is a **white-box** testing methodology analyzing source code, bytecode, or binaries **without execution**, providing:

- Early vulnerability detection
- Code-wide coverage
- Compliance and secure coding practices

SAST Process:

1. Code Parsing: Transforms code to abstract syntax tree (AST)
2. Data Flow Analysis: Tracks data movement
3. Control Flow Analysis: Examines execution paths
4. Pattern Matching: Locates known vulnerabilities
5. Rule Engine: Applies rules to detect risks

Vulnerability Types Detected:

- Code logic flaws: XSS, SQLi, buffer overflow
- Unsafe and outdated dependencies
- Insecure configurations
- Input validation errors
- Auth & access control issues

Snyk: Developer-first SAST Tool

- Database of 700,000+ vulnerabilities
- Real-time updates via multiple sources
- Integrates with GitHub, Jenkins, Maven, npm, etc.
- Produces risk-prioritized reports and actionable fixes

Severity Classification:

| Severity Level | CVSS Score | Risk Level | Action Required |
|----------------|------------|------------|-------------------|
| Critical | 9.0–10.0 | Extreme | Immediate fix |
| High | 7.0–8.9 | High | Fix in 30 days |
| Medium | 4.0–6.9 | Moderate | Fix in 90 days |
| Low | 0.1–3.9 | Low | Fix as convenient |

PROCEDURE

METHOD 1: Snyk Dashboard Integration

Step 1: Account Setup

- Go to snyk.io, register, authenticate via GitHub/Google/email

The screenshot shows the Snyk dashboard interface. On the left, there's a sidebar with navigation links: Organization (humayunk01), Dashboard, Projects (selected), Integrations, Members, and Settings. Below the sidebar, there are notifications for Product updates, Help, and Humayun Khan. The main content area has a header 'Secure your dependencies with Snyk' and a sub-header 'Scan your projects to get started.' It features three main sections: 'Monitor deployed apps', 'Protect your source code', and 'Monitor local projects'. Each section contains a list of features and a 'Browse integrations' or 'Add projects' button. At the bottom right, there's a link to 'Full documentation for Snyk CLI'.

Step 2: Repository Integration

- Add a GitHub project, grant access, configure settings (scan frequency, notifications)

The screenshot shows the Snyk Integrations page. The sidebar includes Organization (humayunk01), Dashboard, Projects, Integrations (selected), Members, and Settings. The main area is titled 'Source Control' and lists various integration options: All Integrations, Source Control (GitHub, GitHub Enterprise, GitHub Cloud App, GitHub Server App), GitLab, Bitbucket Server, Bitbucket Cloud App, and Bitbucket Cloud. The GitHub integration card is highlighted with a red border and labeled 'Configured'. A search bar at the top says 'Search integrations...'.

snyk | Which GitHub repositories do you want to test?

sprintbootwebapp

Personal and Organization repositories

Humayunk01

sprintbootwebapp

We are unable to search repositories where you have been added as a contributor on GitHub

Settings

Add custom file location (optional)
To add a dependency from a non-default path, add it below:

Select a repository... /path/to/file.ext

Exclude folders (Supported for Snyk Open Source and Snyk Container only, optional)
Specify the names of the folders that you want to exclude from the search (maximum 10 folders)

humayunk01 > Projects > Import Logs

Import Logs

Today

Humayunk01/sprintbootwebapp (master)
Import triggered at 19:54:20

| Project | Status |
|---------------------------|-----------------|
| pom.xml | Project created |
| Snyk Code supported files | Project created |

View project

View project

29 September 2025

> Humayunk01/springbootwebapp (main)
Import triggered at 18:13:54

> Humayunk01/springbootwebapp (main)
Import triggered at 14:51:37

> Humayunk01/springbootwebapp (main)
Import triggered at 14:47:45

humayunk01 > Projects

All projects

Add filter

Group by targets

Sort by highest severity

Targets

Humayunk01/sprintbootwebapp

| Project | Imported | Tested | Issues |
|---------------|--------------|--------------|----------------------|
| M pom.xml | a minute ago | a minute ago | 10 C 116 H 76 M 17 L |
| Code analysis | a minute ago | a minute ago | 0 C 2 H 0 M 0 L |

Ready to import another project?

Secure your entire stack with Snyk

Add projects

Step 3: Scan Execution

- Snyk detects dependencies and scans automatically
- Generates vulnerability report, ranks by severity

The screenshot shows the Snyk Targets interface. At the top, there's a search bar labeled 'Search targets'. Below it, a summary of issues is displayed: 10 Critical (C), 116 High (H), 76 Medium (M), and 17 Low (L). A prominent red box highlights the 'View report' button. The main area shows a project named 'HumayunK01/springbootwebapp' with two entries: 'pom.xml' and 'Code analysis', both imported and tested 3 minutes ago.

Step 4: Report Analysis

- Review vulnerability details (CVE, CVSS, affected package, fix info)

The screenshot shows the 'Issues Detail' report. On the left is a sidebar with 'Reports' selected. The main area has a heading 'Issues Detail' with a 'Change Report' dropdown. It includes filters for 'Issue Status: Open', 'Project Target: HumayunK01/springbootwebapp', and 'Project Origin: github'. The 'Issues' section displays 'Total Issues' and 'Unique Vulns' both set to 0. The 'Issues by Severity' section shows counts for Critical, High, Medium, and Low vulnerabilities, all at 0. A red box highlights these sections.

The screenshot shows the 'Vulnerabilities Detail' report. The sidebar shows 'Reports' is selected. The main area has a heading 'Vulnerabilities Detail' with a 'Change Report' dropdown. It includes filters for 'Issue Status: Open', 'Issue Type', and 'Add filter'. The 'Issues' section displays 'UNIQUE VULNS', 'TOTAL ISSUES', and 'VULNERABILITY' all set to 0. The 'CONFIGURATION' and 'LICENSE' sections also show 0. A red box highlights the 'Issues' section.

METHOD 2: Jenkins + Snyk Plugin Integration

Step 1: Jenkins Setup

- Install plugins: Snyk Security, Maven, GitHub
- Configure Maven, JDK as global tools

The screenshot shows the Jenkins 'Manage Jenkins / Plugins' interface. A search bar at the top contains 'snyk'. Below it, a sidebar has 'Available plugins' selected. In the main area, a card for 'Snyk Security 5.0.1' is highlighted with a red box. The card includes the 'Install' button, which is also highlighted with a red arrow. Other details shown include 'Name: 1', 'Released: 3 mo 22 days ago', and 'Health: 93'.

The screenshot shows the Jenkins 'Manage Jenkins / Tools' interface under 'Maven installations'. A card for 'MAVEN_HOME' is shown, with its name and path ('C:\Program Files\apache-maven-3.9.11') filled in. There is an unchecked checkbox for 'Install automatically'. At the bottom are 'Save' and 'Apply' buttons.

Step 2: Snyk Configuration

- Add API token in Jenkins credentials
- Set Snyk installation

Snyk

Name: Snyk

Install automatically ?

Install from snyk.io

Version: latest

Update policy interval (hours): 24

OS platform architecture: Auto-detection

Add Installer ▾

Save Apply

New credentials

Kind: Snyk API token

Scope: Global (Jenkins, nodes, items, all child items, etc.)

Token: **API Key**

ID: Snyk-API-Token

Description:

Create REST API Jenkins 2.516.3

Global credentials (unrestricted)

+ Add Credentials

| ID | Name | Kind | Description |
|--------------------------------------|--|------------------------|----------------------------|
| tomcat-creds | humayun/***** (Tomcat Manager credentials) | Username with password | Tomcat Manager credentials |
| 2e2019db-82a2-4176-81cf-eddb9e1e2fa8 | HumayunK01/***** (GitHub Token) | Username with password | GitHub Token |
| snyk-api-token | snyk-api-token | Secret text | |

Icon: S M L

REST API Jenkins 2.516.3

Step 3: Pipeline Configuration

The screenshot shows the Jenkins 'New Item' configuration page. The 'Item name' field contains 'MavenSnykSpringBootApp'. The 'Item type' section shows four options: 'Freestyle project', 'Maven project' (selected), 'Pipeline', and 'Multi-configuration project'. A red box highlights the 'Maven project' option. At the bottom is an 'OK' button.

- Create Maven job, add Git repository

The screenshot shows the Jenkins job configuration page for 'MavenSnykSpringBootApp'. Under 'Source Code Management', the 'Git' option is selected. The 'Repository URL' field contains 'https://github.com/HumayunK01/sprintbootwebapp'. The 'Credentials' dropdown is set to '- none -'. A red box highlights the 'Repository URL' field and its associated input field. At the bottom are 'Save' and 'Apply' buttons.

- Set build goals (clean compile), add post-build 'Invoke Snyk Security Task' (target file: pom.xml, severity threshold: high, fail build on issues)

The screenshot shows the Jenkins 'Configuration' page for a project named 'MavenSnykSpringBootApp'. In the left sidebar, the 'Post Steps' section is highlighted with a red box. A modal window titled 'Add post-build step ^' is open, listing several options: 'Invoke Ant', 'Invoke Gradle script', and 'Invoke Snyk Security task'. The 'Invoke Snyk Security task' option is highlighted with a red box and has a red arrow pointing to it from below.

The screenshot shows the Jenkins 'Configuration' page for the same project. The 'Post Steps' section is again highlighted with a red box. In the main configuration area, under 'When issues are found', the 'Fail the build, if severity at or above' checkbox is selected. Below that, 'Snyk API token' is set to 'Snyk-API-Token', and 'Organisation' is set to 'Humayunk01'. Both of these fields are also highlighted with red boxes.

Step 4: Build & Reporting

- Manually or automatically trigger build
- Confirm Snyk scan, review console output
- Vulnerability report archived with build, synced with Snyk dashboard

The screenshot shows the Snyk dashboard for a project named 'Humayunk01'. A specific package, 'com.example.appdirect:springbootwebapp', is highlighted with a red box. The dashboard displays various details about the package, including its creation date ('Created Thu 2nd Oct 2025'), snapshot time ('Snapshot taken by cli 24 minutes ago'), and retest status ('Retest now'). It also shows imported by 'Humayun Khan', monitored on '02 October 2025, 20:04:44', and various metadata fields like Project Owner, Environment, Source, Hostname, Business Criticality, and Lifecycle, each with an 'Add a value' button.

humayunk01 > Projects > HumayunK01/sprintbootwebapp
M com.example.appdirect:sprintbootwebapp

Overview History Settings

Reset Search... 217 of 217 issues Group by none ▾ Sort by highest priority score ▾

ISSUE TYPE

- Vulnerabilities 209
- License issues 8

SEVERITY

- Critical 10
- High 114
- Medium 76
- Low 17

PRIORITY SCORE
Scored between 0 - 1000

Introduced through org.springframework.boot:spring-boot-starter-data-jpa@1.4.0 RELEASE Exploit maturity MATURE

Fixed in org.springframework:spring-beans@5.2.20, @5.3.18

Show more detail ▾ Learn about this type of vulnerability ▾

'FIXED IN' AVAILABLE Yes 207

CONCLUSION

Implementing SAST using Snyk and Jenkins within a CI/CD pipeline enabled early, automated detection of security vulnerabilities—finding 128 issues, including critical deserialization bugs.

Developers benefit from immediate actionable feedback, and security is now embedded across all stages of delivery, reducing costs and boosting the organization's security posture.

Key Outcomes:

- Early vulnerability detection reduces remediation costs & risks
- Seamless security integration for continuous delivery
- Actionable remediation, prioritization, and developer empowerment
- Continuous improvement through feedback and education

Future Enhancements:

- Integrate Dynamic Application Security Testing (DAST)
- Develop custom rules and advanced compliance dashboards
- Measure ongoing security improvement via KPIs