**1. GitHub Advanced Security (GHAS)**

**What it does**:  
GitHub Advanced Security is a built-in feature for GitHub Enterprise that helps teams find and fix security issues early in the development process. It includes static code analysis, secret scanning, and dependency review, all natively integrated with GitHub repositories.

**Why it's useful for pipelines**:

* **Code Scanning (Static Analysis)**: Analyzes source code to detect security flaws like SQL injection, buffer overflows, or logic errors before the code is deployed.
* **Secret Scanning**: Detects accidentally committed credentials or tokens in your codebase and blocks them from reaching production.
* **Dependency Review**: Identifies vulnerabilities in open-source packages before code is merged into main branches.

**Value to the pipeline**:  
This tool shifts security left by integrating security checks directly into pull requests and build workflows. Developers can fix issues before they reach production, reducing remediation time and risk.

**2. Snyk**

**What it does**:  
Snyk focuses on securing the components *outside* your source code — mainly open-source dependencies, container images, and infrastructure-as-code (IaC) configurations.

**Why it's useful for pipelines**:

* **Open Source Vulnerability Scanning**: Detects known vulnerabilities in third-party libraries used in the application.
* **Container Security**: Analyzes Docker images to uncover security flaws in the operating system or software packages.
* **Infrastructure as Code (IaC) Scanning**: Examines ARM templates, Terraform files, etc., for misconfigurations that could expose cloud resources.

**Value to the pipeline**:  
Snyk ensures that what you're building with — your containers, libraries, and infrastructure — is as secure as the code you write. It provides real-time feedback in CI/CD and can block vulnerable deployments.

**3. StackHawk**

**What it does**:  
StackHawk is a dynamic application security testing (DAST) tool. It scans your **running application or API** for vulnerabilities that could be exploited in a real-world environment.

**Why it's useful for pipelines**:

* **Runtime Security Testing**: Identifies issues like XSS (Cross-Site Scripting), broken authentication, insecure APIs, and misconfigured endpoints.
* **Post-Deployment Scans**: Tests the actual application once it’s deployed in a dev/staging environment, catching vulnerabilities that static analysis tools might miss.
* **API Security**: Especially useful for microservices and GraphQL/REST APIs to detect attack surfaces.

**Value to the pipeline**:  
While code and dependency scans are preventive, StackHawk simulates real attacks on the deployed app, helping ensure it’s actually secure in production-like environments.

**Why All Three Are Needed Together**

| **Tool** | **Stage** | **Type of Scan** | **What It Protects** |
| --- | --- | --- | --- |
| GHAS | Pre-build | Static code + secrets | Vulnerabilities in the source code, hardcoded secrets |
| Snyk | Build | Dependency, Container, IaC | Vulnerabilities in libraries, container base images, cloud infrastructure |
| StackHawk | Post-deploy | Dynamic (DAST) | Real-world issues in live applications and APIs |

Using all three provides **end-to-end security coverage**:

* GHAS ensures the code written is secure.
* Snyk ensures what the code runs on (libraries, containers, cloud) is secure.
* StackHawk ensures what’s deployed and running behaves securely.