National Defense-ISAC

Software Security Automation: Security Controls Evaluation Criteria

August 19, 2020

Authors:
Marion Belden
Chris Breeden
Mike Heim
Paul Keim
Jeff Malovich
John Munsch
Waldemar Pabon
Allen Ott

Reviewers:
Marion Belden
Jose Vega
Renee Stegman

Table of Content

Executive Summary

Securing code as well as the dependencies used in software requires a combination of security controls to support good cyber hygiene and a cohesive cyber protection strategy. Organizations who create software need to secure their Software Development Lifecycle (SDLC) by implementing and requiring good cyber hygiene and security controls. Secure code development requires the implementation of specific security checks at different stages in the SDLC. To provide proper coverage, securing code requires the implementation of multiple application security tools, which helps proactively identify weaknesses and risk. The development process encompasses an ecosystem of code as well as dependencies supporting a successful provisioning of features as part of the software engineering practice.

With such a diverse set of security controls, how does your organization know if the right tool was selected for the right job? Understanding the organization's requirements as well as the features needed to enforce security compliance becomes an important step in the security control selection process. Furthermore, as more organizations move to agile environments where automation and integration become key ingredients of the cyber protection strategy, any potential evaluation needs to consider integration, detection, and protection as well as compliance requirements. Traditional tool selection processes, without a cohesive strategy, could translate into a time-consuming exercise which fails to fulfill key requirements needed to support an accurate evaluation. Complementing the ND-ISAC's previous work, ND-ISAC Software Security Automation: A Roadmap Toward Efficiency and Security White Paper, this Security

Controls Evaluation Criteria offers a practical approach for selecting the products your organization will need to implement its security automation roadmap. This white paper provides a quantitative approach to evaluate each security tool and align the needs of the organization with a common set of features and capabilities available in the different security controls.

To help organizations with their security control selection process, this white paper provides a quantitative, criteria-based approach through the implementation of a grading requirement framework, The content of the paper focuses on integration, detection, protection, and compliance requirements and common feature sets offered by security tool vendors to fulfill the organizational needs. Regardless of the characteristics of your organization and its processes, selecting the right tools to satisfy these requirements and enable security automation is a key component in securing your SDLC and improving the organization's cyber hygiene strategy.

Introduction

Objective

Organizations who create software need to secure their Software Development Lifecycle (SDLC) by implementing and requiring good cyber hygiene and security controls. Secure code development requires the implementation of specific security checks at different stages in the SDLC. To provide proper coverage, securing code requires the implementation of multiple application security tools, which helps to proactively identify weaknesses and risk. The development process encompasses an ecosystem of code as well as dependencies supporting a

successful provisioning of features as part of the software engineering practice. Securing the code as well as the dependencies used in software requires a combination of security controls to support a good cyber hygiene strategy.

Such a strategy will help development teams minimize the number of vulnerabilities present in code. A vulnerability is a hole or weakness in the application, which can be a design flaw, or an implementation buy, that allows an attacker to cause harm to the stakeholders of an application.

The table below represent the technologies required to secure software at the different stages of the SDLC and that are part of an organization's application security controls.

References to specific vendor products in this white paper are provided as examples for informational purposes only and do not represent an endorsement of a product or solution.

Application Security Technologies Definitions

Technology	Description
Static Application Security Testing (SAST)	Conducts white box testing, performing analysis of source code for security vulnerabilities early in the software development process as part of the Integrated Development Environment (IDE), the commit or build process in a Development Operations (DevOps) methodology.
Software Composition Analysis (SCA)	Provides detection capabilities for security vulnerabilities in third-party components.
Dynamic Application Security Testing (DAST)	Conducts black box testing to detect vulnerabilities associated with the application behavior by evaluating content from user/attacker perspective while application is running.
Runtime Application Self Protection (RASP)	Provides detection and protection capabilities during runtime through instrumentation by monitoring an application's behavior and context of the behavior.
Interactive Application Security Testing (IAST)	Analyzes code by scanning for security vulnerabilities while the application is being tested (automated or manual test). This security control runs outside of the continuous integration continuous delivery (CI/CD) pipeline.

With such a diverse set of security controls, how does your organization know if the right tool was selected for the right job? Understanding the organization's requirements as well as the features needed to enforce security compliance becomes an important step in the security control selection process. Furthermore, as more organizations move to agile environments where automation and integration become key ingredients of the cyber protection strategy, any potential evaluation needs to consider integration, detection, and protection as well as compliance requirements.

Traditional tool selection processes, without a cohesive strategy, can translate into a time-consuming exercise which fails to fulfill key requirements needed to support an accurate evaluation. To reduce associated software risks and weaknesses, organizations need a cyber protection strategy that includes a defense in depth approach and implementation of security controls within the SDLC. This white paper provides a quantitative approach to evaluate each security tool against the needs of the organization. The evaluation framework provides a common set of features and capabilities to assist organizations with the evaluation of different software security tools.

Audience

The audience for this white paper includes security engineers, lead software engineers, product managers, senior managers and executives responsible for the selection, implementation, and integration of software security automation tools in the organization.

Structure of the paper

This paper introduces the importance of selecting the right tool to secure software and minimize risk to the organization. The evaluation framework needed to simplify the selection and evaluation process is presented and discussed. Common requirements are presented in a series of categories providing a feature structure to group them. Finally, this white paper provides a set of tool specific requirements which are structured in a series of feature categories; following the same structure presented in the common requirements section.

Software Security Automation: Security Controls Evaluation Criteria

Requirements Evaluation Process

This section of the white paper explains how to use the Category tables within each section to evaluate, rate, and select a security automation tool. Each organization needs to consider their specific environment and business requirements in order to select an appropriate automation tool. Use the tables from the Common Categories and Requirements section and tables from the appropriate tool section to identify your organization's requirements. As outlined below, assign a weight to each category and individual criteria in order to objectively select the appropriate automation tool for your organization.

Process steps to evaluate and select the appropriate automation tool based on your organization's requirements:

1. Identify Categories
2. Select and Specify Criteria
3. Assign Category Weight
4. Assign Individual Criteria Weight
5. Distribute Request to Vendors
6. Rate Responses

Identify Categories

First, identify the categories needed for the security tool under evaluation. Many categories are common across the Application Security Technology stacks (refer to the Common Categories and Requirements section). However, not all categories are applicable to all stacks. For example, the CI/CD Pipeline Integration Category is important to SAST and SCA, but unrelated to RASP.

Select and Specify Criteria

Criteria are questions related to how the tool satisfies a specific requirement. Select and specify criteria needed within each of the identified categories dependent on your organization's requirements. Your organization may not need all criteria identified, or your organization may need to modify criteria details dependent on the organization's environment and/or customer base.

Assign Category Weight

Assign a weight to each category identified for evaluation; the total of all category weights should add up to one. The purpose of assigning a weight to each category is to identify the categories that are most important in the decision process for the tool under evaluation. The category weighting is used in conjunction with the individual criteria weight, rating, and score to select a tool. When applying category weights, determine which category is more important than the other categories and give it a higher weight. Depending on your company's organizational structure, the technical evaluation may not include a Licensing Category, leaving the

responsibility with your procurement department to determine how a given vendor's licensing model maps to your organization.

Example Category Weights:

(Note: Organizations must determine which categories are more important than the others, dependent on the tool under evaluation. Below is just an example):

Category	Related Criteria Questions relate to:	Weight
Integration	Plugins and/or capabilities that support executing scans from CI/CD pipelines.	.15
Application Program Interface (API) Capabilities	Ability to automate management of projects, users, scan results, token and policy configuration, via an API.	.12
Identity and Access Management	Ways in which the tools support authentication and access management.	.11
Detection Capability	Process/Capability of tool to detect vulnerabilities and minimizes false positives/false negatives.	.10
Programming Language Support	Ability to support the programming languages used in the organization to meet enterprise needs.	.13
Reporting	Ranging from individual scan results to roll-up metrics.	.09
Compliance	Process the tool uses to support maintaining compliance industry and government regulations/standards.	.04
Support	Vendor maturity, support options, vendor support personnel location.	.01
Licensing	Questions relating to how the licensing model works; per developer, per site etc.	.25
Total		1.0

Assign Individual Criteria Weight

After specifying criteria questions for a given category, weight each criterion within each individual category, applying a higher weight to more important criteria. The criteria weights assigned within each category need to add up to one.

Example Criteria Weights:

(Note: In some cases, you may include Informational Questions which provide context related to another question. Use a weight of 0.0 for informational questions).

	Rationale	Weight
Criteria Question 1	Rationale related to question.	.15
Criteria Question 2		.12
Criteria Question 3		.16
Criteria Question 4		.10
Criteria Question 5		.09
Informational Question		0.0
Criteria Question 6		.07
Criteria Question 7		.06
Criteria Question 8		.08
Criteria Question 9		.11
Criteria Question 10		.05
Total		1.0

Distribute Request to Vendors

Remove all rationale, weight, rating, and score columns from the evaluation spreadsheet that is sent to the vendors and distribute a request for proposal (RFP) or request for information (RFI) to each vendor. The criteria questions are shared with vendors as part of the RFP, the assigned rationale, weighting, rating and score are never shared with bidders. Only share assigned rationale, weighting, rating, and score information within your organization.

Rate Responses

Rate each vendor's criteria response by assigning a score of 0, 1, 2, 3, or 4 based on how closely the vendor's response aligns with the requirement.

0- product does not meet the requirement

1- product partially meets requirement

2- product meets requirement in a sub-optimal way

3- product meets requirement

4- product meets and significantly exceeds expectations in how the requirement was implemented.

Example						
Category	Criteria	Rationale	Vendor Response	Weight	Rating	Score
Licensing	Question 1	*The purpose of this question is to determine how the product....*		.15	1	.15 (weight x rating)
Licensing	Question 2			.12	3	.36 (weight x rating)
Licensing	Question 3			.16	2	.32 (weight x rating)

Remove all rationale, weight, rating, and score columns from the evaluation spreadsheet that is sent to the vendor.

If you have a team of analysts reviewing vendor responses, the rating is the average of the consolidated analyst rating for each requirement. If you have 4 or 5 analysts, they will all evaluate each requirement, identifying the average per requirement.

E.g. 3 reviewers provide Ratings; 3,2 and 3, Score = (weight x (3,2,3)/3)

Common Categories and Requirements

This section provides a compilation of common categories and associated requirements to include in the evaluation process of all security control tools (SAST, SCA, DAST, IAST, RASP). The common categories and requirements set the context for the evaluation process of the different security controls. These requirements aid with the selection effort needed to satisfy the provisioning of a basic set of features/capabilities to support the organization cyber hygiene strategy. The common requirements for tool selection are broken down within the following categories:

- Licensing
- API
- Integration
- Support
- Detection Capability
- Identity and Access Management
- Compliance
- Reporting
- Programming Language Support

Each of these categories provides common requirements to consider regardless of which security control is evaluated. Organizations should address the level of importance associated with each requirement based on their unique needs. Refer to the Requirements Evaluation Process for an explanation of how to properly weight each requirement below.

Category: Licensing					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How is the licensing cost calculated? Is it per scan, per scanner, per application, per project, lines of code, etc.?	Evaluate each one of the cost structures provided by the vendor to select an approach aligned with current and future budget constraints. Prefer the solution that has the best licensing model for your organization.				
Is there a path for the price model to scale without becoming very expensive?	Prefer solutions which provide cost scaling flexibility.				
Are you able to combine different pricing models?	Prefer tools with multiple pricing models as they will support a cohesive alignment with the organization cost strategy.				

Category: API					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How does the tool expose tool features through the API? How much functionality is exposed?	Prefer tools exposing the features needed to be supported in your DevOps pipeline.				
Are APIs available to expose tool data? If yes, what data is and is not available? Which APIs are supported?	Prefer tools that securely allow the download scanning report data. Such a capability will support any business intelligence effort to provide better visibility into the overall organization compliance level.				
How are the APIs protected? Is the tool using API Keys, JavaScript Object Notation (JSON) Web Tokens, etc.?	Prefer tools protecting the APIs with API keys.				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How does the tool integrate with DevOps Agile workflows?	Prefer solution which supports the integration (e.g., use of plugins) into DevOps workflows without the need to create custom code for the integration to work.				
How is the tool able to track which artifact is currently published to production, and which artifacts were scanned as part of development, but were never released to production?	This feature is useful for integration into asset management tools and vulnerability reporting tools (applies to all security controls except RASP).				
Does the tool offer an integrated development environment (IDE) plugin for analysis during development? What IDEs are supported?	Important for the Shift Left perspective, allowing earlier integration and detection capabilities (applies to all security controls except RASP).				
What CI/CD tools (e.g. GitLab, Jenkins, Azure DevOps, Bamboo, etc.) can be integrated for automated scanning?	Prefer tools which use plugins to integrate their capabilities into CI tools.				
Does the CI/CD integration kick off a scan in a 'fire-and-forget' manner, or is there a mechanism for having the build pipeline await scan completion and pass/fail the build based on results?	Prefer ability to have pipeline await scan results, enabling automation of security checks (applies to all security controls except RASP).				
Does the CI/CD integration provide automated blocking of deployments of unscanned or flawed builds?	Prefer solution providing control to deploy into different environments.				
Can defects be pushed into work backlogs within an Application Lifecycle Management (ALM) system? What ALMs are supported? (e.g. GitLab, Azure DevOps, Jira, etc.)	Prefer option to push high-level summary tickets, rather than creating one ticket per issue. Avoid spamming the backlog.				

Does the tool offer integrations with industry standard Security Information and Event Management (SIEM) solutions? (Splunk, ArcSight, etc.) (via API, via add-on/application, etc.)	Prefer options that allow security pertinent data to be sent to SIEM tools for monitoring and reporting. This depends on organizational guidelines or capabilities.				
How does the tool integrate with GRC or other reporting mechanism? If not, where are results tracked for risk & resolution?	Prefer solution supporting the integration with GRC or other reporting mechanisms providing risk visibility to the organization.				

Category: Support					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What product support is available? Is email, phone, and online support provided?	Prefer solution where the vendor support mechanisms align with the needs of the organization.				
What is the solution's release cycle? (e.g. quarterly, annually, etc.)	Prefer solution with a well-established release cycle. Mature vendors will provide updates on a quarterly/bi-yearly basis. Vendors with no planned patching schedules for their solution should be carefully evaluated.				
Does the solution release cycle include updating the supporting information (knowledge center, API documentation, etc.)?	Prefer solutions providing comprehensive documentation, which should include APIs, knowledge center, etc. (if applicable).				
Is documentation pre-released in advance of the release?	Prefer solutions providing updated documentation in advance or when the latest changes are released.				
Is 24x7 support offered as part of the regular maintenance?	For critical/Tier 1 applications, prefer vendors providing 24x7 support. For RASP, this may be a critical requirement as application production may be impacted by RASP outages.				
Is product documentation available	Prefer vendors providing online documentation and videos in addition to downloadable				

in any of the following ways? Online Downloadable File Video/Audio	documentation. Clarify if documentation is generally accessible to your entire organization without additional user registration and without a shared account. If there is some authentication required for access to documentation, that access should be integrated with your enterprise single sign on.				
What formal product training is available?	Prefer vendors that provide options for product training depending on the needs of your organization.				

Category: Detection Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool detect common injection and advanced attacks? If yes, what industry standards are supported (e.g. OWASP® Top 10, CWE/SANS Top 25, etc.)?	Prefer solution providing detection for industry standards such as the OWASP® TOP 10, CWE/SANS Top 25, etc.				
What default policies are provided by the tool? What common certifications or regulatory requirements (e.g., Payment Card Industry Data Security Standard (PCI DSS), Health Insurance Portability and Accountability Act (HIPAA), etc.) are supported?	'Policy' here refers to collections of CWEs or other findings grouped together to make a coherent baseline that development teams should be attempting to address. Understanding the default policies implemented in a tool helps evaluation of implementation time and startup cost.				
How does the tool implement custom policies?	Understand if the tool uses CWEs or other vulnerability enumeration to identify and correlate findings.				
Is it possible to apply multiple policies to one application profile (e.g. apply the PCI DSS policy, the .NET policy, and the HIPAA policy to one profile)?	Prefer solution allowing the configuration of multiple policies for a single application profile.				

Does the tool auto-detect the application language and customize the policies applied based on that?	Prefer solution providing a manually tuned or auto-detection capability for the application language used and corresponding policies.				
Does the tool support the ability to re-classify risk ratings of specific findings or finding categories?	Prefer solution providing the ability to re-classify risk ratings of specific findings or finding categories.				
How are policies applied to applications? Are they applied globally, or can they be assigned to portfolio groups, or specifically to single application profiles?	Prefer solution allowing to apply policies either globally, or groups, or to a specific application.				
Does the tool support re-scanning old application versions with updated rulesets?	Understand and identify the support for a historical forensics' assessment.				
Does the tool support custom filter creation for findings?	Custom filters allow teams to mass-mark known false positives in the codebase, reducing the overall amount of consistent work needed.				
What defect validation occurs before presented to a tester? How is the logic explained to the tester?	Prefer solution which thoroughly explains to a tester or developer why a particular request and/or response indicates a vulnerability (applies to all security controls except RASP).				
Does the tool provide customers an early warning of vulnerabilities before they are officially accepted in the National Vulnerability Database (NVD)? (This may apply to most of the security controls but not all.	Tool should aggregate multiple vulnerability intel sources to warn of new vulnerabilities in advance of being published in the NVD, thereby reducing the window of exposure. Vendors should be forthcoming about their vulnerability research capabilities and vulnerability intel feeds incorporated into the product.				
What remediation advice is provided?	Provide the identified vulnerabilities with remediation steps and advice about configuration changes, component upgrade details, and code change requirements.				

Category: Identity and Access Management					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What Identity Providers (e.g., Active Directory, Azure, etc.) does the solution support?	Understand what identity providers are supported to plan for integration into your environment.				
Does the solution support Single Sign On (SSO)? If so, what SSO technologies Security Assertion Markup Language (SAML), Open Authorization (OAuth), etc.) does it support?	Single Sign-On simplifies access to the tool and can also serve as a control point for Multi-Factor Authentication (MFA).				
What is the tool's authorization model? What default user roles exist? Can custom user roles be created? Can role-based-access be scoped to a single application and a portfolio of applications? Can the security tool's roles be mapped to roles used in other application development lifecycle tools?	The eventual goal of security automation and orchestration is to move to a self-service model, streamlining the developer experience with security, and reducing the overall workload for the security engineer.				
Does the tool natively support Multi-Factor Authentication?	If SSO will not be supported in the environment, native MFA provides a useful additional control preventing unauthorized access.				

Category: Compliance					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool provide updated security dashboards for standards compliance: PCI DSS, HIPAA, OWASP® Top 10, SANS/CWE Top 25? List supported compliance items.	Prefer solution providing coverage for industry standard compliance needed for your organization (PCI, OWASP®, etc.).				
Can the tool trigger different responses: from initiating an automated approval workflow to failing the build? If yes, provide supported response types.	Prefer solution allowing different responses during the build process.				
Can the tool ensure an application meets policy guidelines (e.g., no critical or high vulnerabilities associated with application)? If yes, describe implementation.	Prefer solution supporting the enforcement of policy guidelines.				

Category: Reporting					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How can individual teams review corresponding security vulnerability findings?	One of the key aspects of security control tools is the ability to provide self service capabilities to development teams. Prefer solution where each team can have access to specific security issues.				
How does the tool provide the total number of vulnerabilities associated with a project?	Prefer solution providing reports with security vulnerabilities associated with the specific project to support a better remediation plan.				
How does the tool provide detailed security guidance and remediation advice on vulnerabilities?	Prefer tools providing information about security vulnerabilities and also providing guidance on how to fix the security vulnerabilities.				
What scoring system does the tool use to	Understand what differences might exist between the tool's				

evaluate the risk of findings? Does the tool support custom scoring definitions?	risk calculations and organizational risk calculations.				
Are all of the tool's reporting functions available through the tool's API?	Prefer tools with reporting integrations that support any organizational dashboards, security information and event system (SIEM), or other centralized logging systems.				

Category: Programming Language Support					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Which of the following programming languages are supported by the tool? Python / C# Java / C/C++ JavaScript /TypeScript / Visual Basic (VB .Net) Hypertext Preprocessor / VBA R / Objective-C Swift / Matlab Kotlin / Go Ruby / Scala Rust / Dart Ada / Lua Perl / Cobol Groovy / Haskell	Prefer solution providing coverage of common programming languages. The list provided is not complete but is generally in order of popularity based on languages used in open source projects at the time of this writing. Ensure languages used in your organization are supported.				
Which of the following common programming frameworks are supported? .NET Framework / Java Server Faces (JSF) Spring / Sling Struts / Django jQuery / Angular React / Rails Grails / Sinatra Flex / Flask ASP / Hibernate Entity Framework	Ensure the tool covers programming frameworks used throughout your organization. A variety of frameworks are listed as examples, but this list is not exhaustive.				

Use the common categories and requirements provided in this section in conjunction with the requirements for each specific security control presented in the rest of the white paper.

Tool Specific Criteria

This section provides a compilation of requirements specific to each security control (SAST, SCA, etc.). To successfully evaluate a tool, organizations need to combine in their evaluation process the requirements presented in the Common Categories and Requirements section as well as the specific requirements associated with each security control. This section will cover specific requirements for the following security controls:

- Static Application Security Testing
- Dynamic Application Security Testing
- Software Composition Analysis
- Runtime Application Self Protection
- Interactive Application Security Testing

Static Application Security Testing

SAST tools evaluate the application's source code or compiled artifacts in order to identify common patterns indicative of issues. Because SAST examines the code directly, the detection rules, filters, and scanning capabilities will vary depending on the languages and frameworks used by the development teams. Thus, it is important to select a tool that provides maximum coverage of your programming environment.

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What is the standard deployment of the tool? Persistent scan servers, containerized scanners created whenever an artifact is generated, or a different model?	Understanding the different deployment models will better indicate how the tool will be able to be deployed in your environment and fit in with your existing technology environment.				
For Software as a Service (SaaS) or other managed services: How long are artifacts stored within the tool? How are the artifacts protected when stored and in transit?	For applications that are valuable intellectual property, a cloud based SAST tool may not be appropriate. Even if one is selected, some artifacts may need to be purged immediately after finishing their scan.				
Does the tool support customization of notifications and other automated messaging?	The ability to customize notifications and other messaging can help to reduce the signal-to-noise ratio presented to developers, as well as presenting appropriate remediation and mitigation steps at every opportunity.				
Does the tool support developer sandbox scans for pre-	The limitations of the tool around how much pre-production scan support is provided are important to plan around for development teams that produce a high volume of release candidate builds.				

production testing? If so, what limitations exist around the number of sandbox scans for any given application?					
How long are scan results stored for? Are they exportable into a long-term storage format?	Having the historical records of applications available is useful for forensic analysis and having them in a searchable format allows for building of long-term metrics and success stories.				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool have an integrated Software Composition Analysis component?	Combining scanning components reduces the amount of cross-tool finding correlation needed.				
Does the tool have an integrated DAST or IAST component?	Combining scanning components reduces the amount of cross-tool finding correlation needed.				

Dynamic Application Security Testing

DAST solutions test an application while it is running, rather than via source code analysis. A pure DAST tool conducts tests with a black-box approach, focusing only on the target application's inputs and outputs (rather than requiring the configuration of instrumentation like IAST). DAST is most commonly conducted by leveraging the Hypertext Transfer Protocol (HTTP) interfaces of a running web application; the DAST tool attempts to perform attacks on the application by sending malicious requests and analyzing the responses received to determine if

the attack was successful. Consideration should be given to the types of web applications (e.g. traditional dynamic web applications, applets, single-page applications) and web services (e.g. Representational State Transfer (REST), Simple Object Access Protocol (SOAP)) that a particular DAST solution can effectively analyze, the ease of analysis configuration, and the effectiveness and configurability of testing rules.

This section provides selection criteria specific to DAST solutions. Requirements are divided into categories pertaining to DAST specific requirements.

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Describe tool performance for users with a large amount of fine-grained permissions. What are the impacts on tool performance with such a configuration?	Especially for large companies; if a single user is not an administrator but has responsibility for scanning a large number of target applications, the tool should handle this scenario well and without marked performance degradation.				
How are scanning agents deployed and scaled?	Environment should be easily expandable to suit organization's scanning needs.				
How are scanning agents deployed in different environments (e.g., commercial cloud and on-premise network)? Is requiring a Hypertext Transfer Protocol Secure (HTTPS) connection between the administration server and the agent supported?	Prefer a solution where the product supports versatility and be able to dynamically test different target environments.				
If a server reaches capacity, can the environment easily be scaled up / load-balanced without the need to create isolated, entirely separate installations of the scanning tool? How is this supported?	One central instance of the tool (rather than separate instances created to ameliorate capacity issues) greatly reduces environmental complexity for both administrators and users.				

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool provide a vault or 'secret keeper' to securely store credentials used for testing? If yes, describe how this is accomplished.	Credentials should be securely stored in a way that other users cannot access them. Additionally, this assists users with tracking multiple sets of testing credentials.				
How are testing credentials treated? Are testing credentials treated as variables, rather than as just a part of the login script?	Secure storage of credentials. Additionally, allows users to update credentials without entirely recreating login steps.				
How does the tool intake Uniform Resource Locators (URLs) for scanning? Which of the following are supported? Manually input a homepage or landing page (allowing the scanning tool to crawl the target website) Capture URLs by browsing the target application with a browser extension? If so, which browsers are supported? Capture URLs by browsing (manual or automated) target application with built-in tool functionality or a separate utility set up on source scanner or users PC? IDE project file upload SOAP service's Web Service Definition Language (WSDL) / Web Application Description Language (WADL) upload HTTP Archive (HAR) file upload RESTful API Modeling Language (RAML) file upload Export from a 3rd-party tool (e.g. Postman, Fiddler, Burp, Swagger)	Prefer methods that minimize tool setup and interaction time (i.e., methods that leverage files developers may already have on-hand are preferable to those which require manual capture specifically for DAST setup). Sufficient crawling capabilities are important in the case that developers / testers do not capture all target URLs.				

What support does the tool provide for scanning single-page applications and JavaScript-heavy applications?	Single-page application (SPA) architecture and JavaScript-heavy applications in general should be robustly supported. Prefer solutions that support JavaScript analysis/parsing to ensure that the target application may be fully explored by the DAST tool's crawling capabilities.				
What methods are supported for target application authentication? Basic / New Technology LAN Manager (NTLM) Transport Layer Security (TLS) Certificate Form authentication Macro / Authentication Flow recording and playback Is custom login script recording supported for more complex target environments (e.g. enterprise SSO)? Is this capability supported directly within the tool User Interface (UI) or is a separate utility required? Does the tool provide a way to simulate Common Access Card authentication? How?	Prefer UI-based scripting for login over HTTP traffic replay as this can more accurately and dependably simulate user login. Prefer support for creating login scripting directly within the tool (i.e., in the web UI), rather than in a separate tool. Prefer support for easily reusing or sharing a login script.				
How is login / successful session in target application verified? If login credentials stop working (e.g., if they expire or if the target application goes down), how does the tool detect this and alert the user?	Login / session detection should be straightforward and easily verified / intuitive (it should be obvious to the tester when the authentication steps are or are not working). Prefer tools which will alert the tester if login credentials stop working, rather than silently failing.				
What API / web service scanning support is available? Compatibility with both REST and SOAP services using both Extensible	Prefer tools that can effectively test SOAP and REST APIs whether they produce / consume JSON, XML, or a mixture of both. Prefer solution which may allow developers to leverage				

Markup Language (XML) and JSON? Mutual TLS and Auth Header authentication supported? Allow import of a WSDL / Swagger file / Postman export for URL intake?	an already-existing list of API calls (such as a WSDL, Swagger file, or Postman export), rather than having to create a new capture for use in the DAST tool.				
Can scan 'templates' be created - pre-configure scans and essentially make them 'plug and play' for testers?	Prefer a scenario in which a tester can input a URL and credentials.				
Can a scan be reconfigured while running, or do reconfigurations require a stop and restart? If stop and restart are required, does this erase any progress the scan has already made?	Prefer solution which will allow mid-scan configuration. With a complex target application, scans can take a significant amount of time. If mid-scan reconfiguration is needed (e.g., credential expiration), it is preferable to be able to resume rather than restart.				
Does the tool offer an accurate estimation of how long a scan will take?	Useful for planning and monitoring of testing activities.				
How does the tool support scheduled scans or a scanning time window to be set on a per application basis?	Development teams may prefer for scans to be run off-hours and/or within a set amount time in order to avoid interfering with business of development activities.				
How configurable are scanning behaviors and tuning (e.g. number of HTTP requests per second, number of concurrent threads)?	Prefer solution which provides default values but offers tuning options - enables optimization of scan behavior to match target application's capabilities.				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool support fetching test credentials from an enterprise credential locker (e.g. CyberArk, CA PAM, BeyondTrust)? What credential lockers are supported?	Secure storage of credentials; ease of use.				

Category: Detection Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What information is presented to the tester for each defect? Full request / response? Is this easily replayable? Is this exportable in some way? Further demonstration of flaw Suggestions on how to fix flaw, links to outside resources Is there an exploit 'proof-of-concept?' E.g. Burp's CSRF PoC Generator	Prefer solution which provides most or all of this information and capability. With more information, it is easier for development teams to understand, evaluate, and remediate a defect.				
Can a defect be individually retested without rerunning the full scan? Does the tool support incremental or targeted finding scans? Can a request / test be manually edited and replayed?	Prefer solution with the ability to reliably retest a specific flaw without running an entirely new scan. Targeted retesting saves a great deal of time in comparison to rerunning a full scan. Prefer solution which allows manual editing and replay of a defect; allows tester to further understand extent of the defect and test fixes.				
Can testing rules and scan templates be tuned/customized? Can rules be customized based on tech stack? Does the tool make rule suggestions based on tech stack?	Prefer solution which will allows flexible configurability of testing rules - this both reduces false positives and shortens scanning time by eliminating unnecessary tests.				

How do testers configure custom rules? How do administrators package testing rules, and create templates, for use by testers? Any issues with tester permissions on certain rules or rulesets? Can scanning policy be customized to exclude certain findings or to perform a targeted scan for a particular issue (e.g. TLS compliance)? Does the solution offer a set of testing rules or type of scan that can comfortably be run on a Production web application?	Ensure that if users must be granted 'access' to rules or rule packages, that this is easy to configure via API and/or for a large group of users at once. Prefer solution which allows administrators to create scanning 'templates,' reducing the amount of configuration necessary for testers. Prefer solution which provides a well-rounded default scanning policy but allows full policy configurability and tuning. Prefer solution which offers a non-invasive scanning option to enable comfortable scanning of Production applications as needed.				
How is authorization boundary testing supported? (E.g., test to ensure that a standard user of the target app cannot access administrative functions or escalate privilege)	Prefer solution that offers support of authorization boundary testing.				
Can a tester put in multiple sets of test credentials on one scan? Test application using different identities with different roles	Prefer solution which allows multiple testing identities to be used on a single scan; otherwise multiple scans may need to be created for the same target application to test different roles.				
Are custom injection strings and/or word lists supported? If so, how?	Prefer solution which allows customization with organization-specific keywords.				

Software Composition Analysis

Software Composition Analysis (SCA), a relatively new technology, identifies security vulnerabilities and licensing flaws in third-party and open source components included in an

organization's applications. In addition to facilitating secure code development and providing software remediation guidance, SCA provides benefits that reduce unplanned work and lowers risk exposure across the business.

The 2020 Open Source Security and Risk Analysis (OSSRA) report identified a variety of weaknesses in open source security, risk, and legal (license) areas1. The report indicates that up to 99% of code in applications today include open source components. This open source code often includes vulnerabilities and/or licensing restrictions that can expose organizations to potential breaches and legal risks. Third party software also has vulnerability and licensing risks that can adversely affect organizations. The same report also found that 75% of the codebases contain vulnerable open source components and over 49% contained high-risk vulnerabilities (Synopsis, 2020).

Additionally, the report identifies audits that found that 68% contained components with license conflicts; 91% of codebases contained components that were more than four years out-of-date or had no development activity in the last two years. Using SCA in the development process would have identified the weakness described in the above OSSRA report, allowing organizations to evaluate and mitigate their open source quality, security, and licensing risks. In addition to vulnerability identification and remediation, SCA helps organizations achieve regulatory compliance.

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How is the solution deployed in the runtime environment? Instrumentation Agent Plugin	Understand the deployment method of the tool (e.g., plugin, Docker container(s), cloud-based, etc.) and environment configuration requirements (e.g., disk space, processors, random-access memory (RAM), operating system (OS)).				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How does the tool integrate with DevOps agile workflows? (e.g. Build tools, Package Managers, CI Servers.)	Prefer SCA tools that integrate seamlessly with repositories and build tools, package managers and CI servers as they provide developers with actionable data in the early stages of the lifecycle.				
How does the tool integrate with the following tools? Ticketing systems Developer Feedback Loop Solutions Security information and event management (SIEM)	Prefer SCA tools that provide native integration with defect management tools (e.g., Jira) to provide streamlined defect management workflow. Provide immediate feedback to developers by highlighting the precise source files, line numbers, and even subsections of lines that are affected by vulnerabilities or licensing issues. SCA tools that integrate into developer's IDE are preferable. Also, evaluate whether your SIEM tool can integrate its' threat data feed(s) for comparison with SCA tool output.				

Category: Detection Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool provide a mechanism for automated identification and verification of vulnerabilities?	Ensure the tool provides vulnerability descriptions, remediation guidance, license information, and potential policy violations so the problem(s) can be addressed in the early stages of development.				
What remediation advice is provided?	Provide the identified vulnerabilities with remediation steps and advice about configuration changes, component upgrade details, and code change requirements.				
Does tool produce fixes to the code to reference a safe version of an open-source component?	Prefer tools that provide an option to automatically remediate vulnerabilities by creating pull requests to upgrade to a current version (or one that complies with company policy). Additionally, some products offer custom patches when an acceptable version of a component is not available.				
How does the tool provide alerts for newly discovered vulnerabilities that affect previously scanned software?	Provide continuous monitoring capability on repositories for newly discovered security/licensing/vulnerability issues and generate alerts and policy enforcement to proactively identify/eliminate risks.				
Can the tool scan binary code in addition to source code?	Depending on your environment, select an SCA tool that can scan virtually any software, including desktop applications, mobile applications, and embedded system firmware.				

Category: Policy & Compliance					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Automatic license policy enforcement to cross reference every open source component found in your code with your organizational software license policies?	SCA can ensure that all applications and use of code comply with the associated license policies of the applications and code. This compliance insurance is critical, as many different types of licenses exist in the software industry, and each has a different set of legalities and operational agreements.				
Assurance that code snippets are identified, and where appropriate, corresponding licenses are acquired and included?	Snippets of code pulled from a component still carries with it license compliance obligations. Snippet scanning identifies fragments of code that match a larger component, allowing you to view the associated license information, and ensures it is included in the BoM.				

Category: Reporting					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What out-of-the-box reports are provided that describe the application's risk?	Ensure that the SCA tool's pre-defined reports provide the level of detail required to assess the risk associated with the application(s).				
What inventory reporting is available? Does inventory reporting include all open source components, and all direct and transitive dependencies?	Prefer SCA tools that generate an inventory report of all open source components (including direct and transitive dependencies) so you can ensure license compliance and manage security vulnerabilities.				
Can the tool generate reports on how fast developers remediate known vulnerabilities and license risk?	Evaluate whether SCA tool calculates remediation metrics (e.g., MTTR) on resolving an application's licensing/vulnerability issue(s) to help gauge the efficiency of remediation efforts.				
BoM compare feature that highlights what has changed over time; produce an accurate, up-to-date inventory of their	SCA tools can generate a complete software BoM that tracks open source and third-party components of code. SCA tools identify known security vulnerabilities, associated licenses,				

open source versions and patch status?	and code quality risks (including patch status) of those components.				
What industry specific formatted reports are supported? (e.g. CycloneDX, SPDX, SWID.)	There are a several different software bill of materials (SBOM) specifications available; including CycloneDX, SPDX, and SWID. Ensure criteria supports the specification(s) required by your organization.				

Runtime Application Self Protection Tool Specific Criteria

Similar to IAST, RASP is an agent-based tool instrumented into the application's runtime. There, it monitors the runtime behavior for potential attacks using a similar signature-based detection ruleset to a Web Application Firewall (WAF). While there is no real automation mechanism for creating virtual patches via these tools (as the rules will need to be custom created for each situation), the APIs and implementation of the patches can be automated.

This section provides selection criteria specific to the RASP security control. Requirements are divided in categories pertaining to RASP specific requirements.

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How is the solution deployed in the runtime environment? **Instrumentation (agent)** **Plugin** **Replacement of Libraries/Java Virtual Machine (JVM)**	To minimize environment complexity, prefer solution allowing the deployment of agents through instrumentation in the runtime environment.				
What level of tuning is required for the tool to start providing detection and blocking?	Some tools require policy configuration before any detection/blocking is possible. Prefer solution that balance the provisioning of default configurations which help with the				

	detection/blocking early on without extensive configuration effort.				
What level of complexity is associated with upgrades to the RASP tool? High Medium Low	Prefer low levels of complexity with tool upgrades to minimize the cost associated with the operational nature of the security control.				
Can the solution support applications deployed in different environments? On-Premise Containers Platform-as-a-Service (PaaS) Infrastructure-as-a-Service (IaaS)	Prefer solution providing coverage for most, if not all, of the environments.				

Category: Performance					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
What increase % in response latency should be expected from the tool?	RASP will impact the application performance due to the technology approach used to detect attacks in memory. Prefer solution which provide a low latency percentage (less than 10% of increase in response times). Solutions claiming zero latency overhead in the response should be carefully validated. Recommendation: Conduct a performance test of the solution, if possible, to properly evaluate the impact of having the RASP tool in monitoring mode vs. blocking mode. Hardware specifications and workload, along with the number of rules and policies to evaluate could affect the performance of the application.				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Can the solution integrate with other DAST and SCA solutions? Which solutions? How do they integrate?	Prefer solution which can integrate with the security controls used by your organization to provide SCA and DAST scanning capabilities.				
What external tools can integrate with your tool and how? Specify specific names of supported tools. Ticketing System Developer Feedback loop solutions SIEM	Customize this criteria to include names of specific tools your organization requires for integration.				

Category: Detection Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How does the tool identify and verify vulnerabilities?	Prefer solution providing a centralized approach toward the collection and presentation of vulnerabilities.				
How does the tool detect the use of vulnerable 3rd party libraries?	Prefer solution capable of detecting 3rd party libraries vulnerabilities.				
How can policies be created to detect attacks against specific frameworks (tailor detection)?	Some solutions provide mechanisms to create your own policies. Prefer solution providing the flexibility of not just the customization of existing policies but also the creation of new ones.				
What capability if provided by the tool to create custom rules?	Prefer solution that allow the creation and modification of custom rules.				
Does the tool record a log entry in the Dashboard per detection?	Prefer solution logging each vulnerability entry to a centralized Dashboard provided by the tool.				
Does the tool record a log entry in the tool logs per detection?	Prefer solution recording vulnerability findings in the tool logs to facilitate integration with SIEM solutions.				

Category: Blocking Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How does the tool support the creation of policies to derive blocking decisions?	Prefer solution allowing the organization to create their own policies to determine when to block a specific threat vector.				
Does the tool provide predefined industry templates for the attacks that would be blocked?	Prefer solution providing templates for OWASP® Top 10, SANS 25, etc.				
What capability if provided by the tool to create custom rules?	Prefer solution that allow the creation and modification of custom rules for Virtual Patching.				
How is the blocking of a potential attack reported? Does it raise an exception in the underlying application or is there a standard response sent to the end user?	Prefer solution providing the flexibility to decide when to use a customized standard response to the end user and when to raise an exception. Solutions providing only the option to raise exceptions could have an impact in the overall application implementation; careful consideration should be exercised.				
Can policies be created to block attacks against specific frameworks (tailor blocking)?	Some solutions provide mechanisms to create your own policies. Prefer solution providing the flexibility of not just the customization of existing policies but also the creation of new ones.				
Does the tool record a log entry in the Dashboard per blocking?	Prefer solution logging each vulnerability entry to a centralized Dashboard provided by the tool.				
Does the tool record a log entry in the tool logs per blocking?	Prefer solution recording vulnerability findings in the tool logs to facilitate integration with SIEM solutions.				

Category: Alerts & Notifications					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Can the tool provide an alert when an attack is detected? If yes, how?	Prefer solution providing alert capabilities when a vulnerability is detected.				
Can the tool provide an alert when an attack is blocked? If yes, how?	Prefer solution providing alert capabilities when a vulnerability is blocked.				
Which mechanism is provided by the tool to support the creation of policies to derive when to send notifications?	Some solution will enable the relationship of policies and alert triggering associated with the violation of such policies. Prefer solution enabling the integration between policies and notifications.				

Interactive Application Security Testing Tool Specific Criteria

Interactive Application Security Testing is a hybrid of static, dynamic, and runtime application testing. Code is instrumented during development, then dynamically tested prior to production. Combining instrumentation with dynamic testing enables deeper code-level inspection and reporting similar to static analysis, while subjecting the software to actual attacks as it is running in test.

Consider the complexity and effort in instrumenting code specifically for security testing, and the labor involved in configuring and running the post-build dynamic test. Some IAST solutions may reduce time and effort in configuration and testing by injecting or hooking the runtime rather than instrumenting code and automating much of the dynamic test. These optimizations typically come at the expense of a less thorough analysis.

This section provides selection criteria specific to IAST solutions. Requirements are divided into categories pertaining to IAST specific requirements.

Category: Deployment and Configuration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
How is the solution deployed in the runtime environment? Instrumentation (agent) Plugin/Library Runtime Platform Injection	To minimize environment complexity, prefer solution allowing the deployment through runtime platform injection.				
Can the solution support applications deployed in different environments? On-Premise Traditional Data Center Containers-as-a-Service (CaaS) Platform-as-a-Service (PaaS) Infrastructure-as-a-Service (IaaS)	Prefer solution providing coverage for most, if not all, of the environments.				
What level of custom configuration is required for the tool to start providing detection?	Prefer solution that have strong pre-configured policies with easy customization capability.				
Is the solution enabled by functional testing and/or DAST tooling that is integrated with the IAST solution?	If your organization's functional testing is robust prefer tools that are exercised by normal functional testing so that additional activities don't burden the development teams. If the quality of functional testing is unknown or poor prefer a solution that also has DAST tooling that integrates with the IAST tool.				

Category: Integration					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Can the solution integrate with other security tooling solutions? Which solutions? How do they integrate?	Prefer solution which can integrate with other security tools, processes and methodologies within your organization.				
What external tools can integrate with the tool and how? Specify specific names of supported tools. Application Lifecycle Management (ALM) system Security Vulnerability Management Systems SIEM	Customize criteria to include names of specific tools your organization requires for integration.				

Category: Detection Capability					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Can the tool detect all OWASP® Top 10 Web Application Security Risks?	Only select a solution that can reliably detect all vulnerabilities that fall under OWASP® Top 10.				
Can the tool detect all CWE/SANS Top 25 Software Errors?	Prefer solution that can reliably detect most of the CWE/SANS Top 25 security vulnerabilities.				
Can policies be created and/or customized to detect more specific vulnerabilities or safely eliminate false positives?	Prefer solution that support scan customization of standard and/or organization scan policies.				
Does the tool support aggressiveness levels that allow for tuning of the false positive to true positive ratio?	Prefer tooling that supports aggressiveness level tuning.				
Does the tool do a good job of eliminating false positives from results?	Prefer a tool that has a good false positive to true positive ratio versus other competitors.				
Does the tool provide a mechanism for automated identification and verification of vulnerabilities?	Prefer solution that have a mechanism for automated verification of vulnerabilities such that it helps eliminate false positives.				

Category: Self Service					
Criteria	Rationale	Vendor Response	Weight	Rating	Score
Does the tool provide a Dashboard to support self-triaging of vulnerabilities by developers?	Prefer solution that have a web application dashboard and/or can be integrated into your organizations security vulnerability management system/dashboard to support developer team triaging of vulnerabilities.				
Does the tool provide detailed security guidance and remediation advice on vulnerabilities?	Prefer solution that provide remediation guidance.				

Conclusion

When it comes to application security controls and tooling, there are a myriad of options for every category of tool. With such diverse offerings available in the marketplace, the selection of the right security tooling can be a daunting undertaking with many factors to consider. The quantitative, criteria-based approach outlined by this white paper, in tandem with your organization's specific requirements, provides a framework for this process, focusing on integration, detection, protection, and compliance requirements and common feature sets offered by security tool vendors to fulfill these needs. Regardless of the characteristics of your organization and its processes, selecting the right tools to satisfy these requirements and enable security automation is a key component in securing your SDLC and improving your cyber hygiene strategy.

References

Synopsys. (2020, May 20). Open Source Security and Risk Analysis Report. Retrieved from https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2020-ossra-report.pdf?force_isolation=true