Abstract
This ESG Technical Review documents ESG’s evaluation and analysis of how HCL AppScan can help developers continuously secure applications using policies defined by security specialists. We also evaluate how AppScan can easily be integrated into CI/CD pipelines and support other aspects of DevSecOps initiatives to provide continuous application security at scale.

The Challenges
According to ESG research, IT still isn’t getting easier. Nearly two-thirds (64%) of organizations say IT is more complex compared with two years ago. This is driven by higher data volumes, new security and privacy regulations, and the increase in the number of applications that leverage modern architectures. At the same time, the threat landscape is becoming increasingly dangerous, as malicious actors focus their energy on developing more sophisticated, targeted attacks. ¹

As organizations have become more aware of cybersecurity threats, and general security best practices have become well understood and well documented within the AppDev community, organizations are incorporating cybersecurity principles and tools into DevOps pipelines and methodologies. While usage of application security (AppSec) tools has increased steadily in the past 5 years, 62% of organizations still only leverage these tools on less than half of their apps (see Figure 1).²

Figure 1. Use of Application Security Testing Tools

Thinking about all your organization’s applications, whether developed in-house or by third parties, approximately what percentage is protected by some form of application security testing tool, such as dynamic application security testing, static application security testing, etc.? (Percent of respondents, N=220)

62% of organizations test less than half of their applications

Source: Enterprise Strategy Group

² Source: ESG Master Survey Results, Application and Email Security Trends, September 2019.
The Solution: AppScan

HCL designed AppScan to reduce organizational risk by helping developers rapidly identify and remediate application vulnerabilities in every phase of the development lifecycle. AppScan can be deployed on-premises or as SaaS on the cloud, and provides a suite of testing technologies including static (SAST), dynamic (DAST), interactive (IAST), and open source testing to increase the security of standalone, web, and mobile applications.

AppScan supports DevSecOps by integrating directly into the software development lifecycle, including inline execution during automated build and deploy pipelines, and feedback and remediation inside the integrated development environment (IDE). The solution also includes management capabilities to enable regulatory, compliance, and security professionals to continuously monitor and maintain the security posture and compliance of applications.

The benefits of incorporating AppScan into a DevSecOps model include:

- **Actionable reporting**—identified vulnerabilities include fix recommendations, simplifying developers’ efforts to secure applications.
- **Reduction in noise**—machine learning analysis improves accuracy, reduces false positives, and helps prioritize vulnerabilities and fixes to maximize ROI.
- **Current vulnerabilities knowledge**—continuous updates ensure that AppScan can identify vulnerabilities associated with the most recent attacks.
- **Better governance**—predefined security testing policies help achieve compliance with regulations such as PCI DSS, HIPAA, GDPR and industry standards and benchmarks including OWASP Top 10, SANS 25, and more.
- **Open source risk mitigation**—identify and manage risk associated with open source components integrated into applications.

AppScan integrates into four main aspects of DevSecOps:

- **Policy definition**—security and regulatory professionals define and associate policies with an application.
- **Scan and analyze**—developers use static analysis to “shift security left,” to identify vulnerabilities in code early in the CI/CD pipeline. Dynamic and interactive testing similarly help developers to identify vulnerabilities in running applications.
- **Vulnerability remediation**—developers review and fix vulnerabilities. AppScan’s machine learning accelerates remediation by grouping and prioritizing related vulnerabilities and identifying potential fixes and fix locations in source code.
- **Reporting**—security and regulatory professionals and organizational management can continuously monitor the security and compliance of their applications.

ESG Validated

ESG’s evaluation proceeded through these four aspects of AppScan activities in a modern DevSecOps CI/CD pipeline, starting with defining and associating security policies with an application. First, ESG defined an application for security testing using AppScan on Cloud’s user interface. During the application definition process, ESG assigned the application to an asset group, which provides a measure of access control. ESG also defined the business impact, choosing from critical, high, medium, and low categories. AppScan uses this information to enable security professionals to better understand and prioritize vulnerabilities.

An advanced option section enabled ESG to provide more information for management, such as the business unit, the app’s URL, description, technology, and personnel involved, including the tester, business owner, and key developers.
Next, as shown in Figure 2, ESG associated security policies with the application. AppScan’s security policies are a method to specify which security issues the developers should care about, and the policies contain definitions of regulations and potential vulnerabilities. AppScan includes numerous predefined policies for industry standard benchmarks and regulations including HIPAA, PCI DSS, GDPR, and more, and users can define their own policies.

HCL recommends that developers start with a manual scan that covers all applicable policies to create a baseline, identifying the current state of the application’s security. Baseline scans enable the organization to identify and manage new vulnerabilities that are introduced after the date of the baseline scan.

The security professionals (who may also be developers) use their security expertise and an understanding of the application’s business impact to define what security vulnerabilities are a concern. For example, an internal corporate HR application may not need to be concerned about denial of service vulnerabilities, but it may need to be tested for SQL Injection and GDPR vulnerabilities.

Policies help separate roles and responsibilities between developers and security experts, while increasing collaboration and efficiency. Developers can focus on application development and vulnerability remediation while security experts maintain the responsibility to assess vulnerabilities and decide which are most critical to fix first.

**Figure 2. Associate Security Policies with an Application**

Next, ESG evaluated the scan and analyze aspect by integrating AppScan into the DevSecOps toolchain. ESG used the Jenkins automation server plug-in for AppScan. As shown in Figure 3, ESG specified both credentials and the application. ESG provided a name for the test and selected between static and dynamic testing.

ESG also had the option to specify additional parameters for static and dynamic scanning. For static scanning, ESG could scan both application code and third-party/open source components or limit the scan to just the application code.

Using the automation, ESG could create complex rules for passing or failing the build, such as non-compliance with application policies or exceeding a threshold for vulnerabilities with high severity. These rules enable application development to proceed in parallel with security vulnerability remediation and enable security experts to define security policies without having to edit the build automation.
Using the AppScan dashboard, we observed the status of the scan during the automated build process. The dashboard provided at-a-glance summary information of the identified issues, security scans, and compliance, along with a list of scans and issues. Using the dashboard, ESG quickly understood the current state of security of its application.

Figure 3. Integrating AppScan into the DevSecOps Pipeline with Jenkins

When the scan completed, ESG evaluated the vulnerability remediation aspect. HCL designed the AppScan dashboard for both the security expert and the code developer. As shown in Figure 4, using the “All Issues” tab on the dashboard, a security expert can review, triage, and prioritize the list of issues without having to look at the application source code. The dashboard includes quick-filter buttons for non-compliant and high/critical issues, and users can customize their own filters.

Figure 4. Scan Results

Clicking on the Details link brings up more details about the specific issue, as shown in Figure 5. The “Issue Details” popup window contains tabs that provide an overview, details, developer discussions, issue history, and recommended fixes.
Developers can add comments on the Discussion tab, and can update the status on the History tab, which acts as an audit trail. Changing the status to noise will keep the issue open but ignore the issue in future scans, accelerating the scan process.

Figure 5. Issue Details

HCL recommends that developers who utilize the AppScan results dashboard review issues using the “Issues by Fix Groups” tab. AppScan uses machine learning techniques to identify, categorize, and group related issues that typically have the same fix location in source code or the same fix technique applied in different locations. Using fix groups helps developers focus by removing extraneous information and can increase developer efficiency.

We selected the first fix group to view the details. As shown in Figure 6, AppScan identified and grouped four related SQL Injection issues. The fix group identified the location of each issue and provided a stack trace to help developers better understand where the issue occurred and how to fix the issue.

The AppScan machine learning engine also identified a potential fix location, increasing developer efficiency by pointing developers to the best location for rapid remediation. To further help developers, ML code attempts to find a common fix point, where one fix can address multiple issues in the fix group.

Figure 6. Fix Groups
To further enhance developer efficiency, developers can process AppScan results directly from their integrated development environment (IDE). ESG used the Eclipse plug-in to retrieve the list of issues from the previous scan. ESG then clicked on the SQL Injection fix group to review this issue, as shown in Figure 7.

All information and actions available in the AppScan console are also available through the IDE plug-in. Status updates are automatically propagated to AppScan on Cloud, maintaining consistency, and ensuring security professionals, developers, and management have the same view of application security.

HCL provides issue descriptions for an extensive list of coding languages and for Java, .NET, and C/C++, it additionally provides developers with the best fix locations. Clicking on a reference to a line of code in the issue automatically loaded the source file into the editor and moved the cursor to the line, enhancing developer productivity and accelerating issue remediation.

**Figure 7. AppScan in the IDE**

Next, as shown in Figure 8, using the AppScan console, ESG created a manual dynamic scan and reviewed the results. Using the manual scan creation wizard, ESG specified the URL to be tested and the testing environment, which controlled the aggressiveness of the test. For a staging or testing environment, AppScan will fill out forms automatically and execute JavaScript code to try to discover as much content as possible, while for a live production site, a more subtle approach will be used to minimize impact on the running application. ESG also configured test optimizations to control the depth and breadth, and thus duration, of the scan.

ESG created a focused dynamic analysis by capturing manual interactions with the website to drive the testing. ESG used an HCL plug-in for the Chrome browser to record interactions with the site. ESG then uploaded the recording to AppScan as part of the dynamic scan’s configuration stage.

ESG also specified whether the application was hosted on a private network or the public internet. For applications on public networks, HCL provides a signature file, enabling scans by authorized developers while preventing unauthorized users from abusing the site.

When the scan completed, ESG reviewed the results. Dynamic and static scans and results were available in various locations, such as the AppScan console, the IDE plug-in, and the AppScan Standard tool. When ESG scanned its application,
AppScan identified a cross-site scripting issue. The details popup included a capture of the website’s communications and a comprehensive discussion of various techniques to fix the issue.

**Figure 8. Dynamic Scans and Results**

Next, ESG evaluated the reporting phase. AppScan provides numerous reports and dashboards with various metrics. ESG could configure the graphs, filtering on key information such as business unit or issue severity. Figure 9 shows two dashboard graphs that ESG found particularly useful. Graphing security risk rating over time enabled ESG to understand the progression toward improving overall application security. Graphing top issues by type enabled ESG to understand the types of issues that were currently responsible for the greatest security risks.

**Figure 9. Reporting Dashboards**
Why This Matters

Integrating and automating application security testing into the DevSecOps methodology enables the identification and correction of cybersecurity vulnerabilities earlier in the application development lifecycle, which enhances security and increases efficiency. It also helps to alleviate challenges faced by skilled cybersecurity teams, which are often much smaller than development teams, as they try to keep up with the increasing pace DevSecOps demands.

ESG validated that HCL AppScan simplified and accelerated application security testing. With just a few clicks, ESG defined security policies and configured AppScan to test our application for a set of security vulnerabilities and compliance with benchmarks and regulations. ESG found configuring and running on-demand scans to be just as quick and easy, and results from static and dynamic analysis were presented quickly in a concise and consistent interface.

ESG validated that the AppScan machine learning engine accelerated the vulnerability remediation process by identifying and grouping related issues and attempting to identify a common fix point for multiple issues. Suggested fixes for each issue enhanced developer productivity.

ESG validated that automating AppScan as part of a DevOps CI/CD process through AppScan’s Jenkins and Eclipse plug-ins was easy and removed barriers that might prevent developers and security professionals from including application security testing as a part of the normal development process.

The Bigger Truth

Malicious actors are always on the lookout for new applications and are continuously creating new types of attacks that target previously unknown vulnerabilities. Addressing this reality is critical to business success; however, security often takes a back seat to functionality. Developers may also view security as an obstacle and are measured first and foremost on the speed at which they can deliver application functionality to end-users. The need to incorporate better security without inhibiting delivery speed highlights the need for integrating and automating application security testing into DevOps, often referred to as DevSecOps.

ESG conducted a technical review of HCL AppScan to evaluate its DevSecOps capabilities, and our evaluation revealed that:

- AppScan fosters cooperation between security professionals and developers. Security professionals can define which security issues need to be fixed without changing DevOps build processes, and developers can incorporate these definitions into their build processes without requiring specialized security expertise.
- AppScan helps security professionals and developers triage and prioritize vulnerabilities, minimizing noise.
- AppScan enhances developer productivity and accelerates vulnerability remediation by grouping related issues, attempting to identify common fix locations, and providing suggested fixes.
- AppScan is easy to integrate into DevSecOps pipelines, by using plug-ins for common automation servers and IDEs. Developers can leverage AppScan without changing their normal development workflows.

While ESG has validated that AppScan can be used for continuous application security via testing in a controlled environment, ESG encourages prospective customers to conduct their own evaluations to determine whether AppScan meets their specific business requirements and objectives.

If you wish to make continuous application security a core element in shrinking your organization’s attack surface and reducing potential avenues of compromise, ESG recommends that you evaluate how AppScan can be incorporated into your DevSecOps and CI/CD methodologies.