dynatrace

The next generation of cloud application security

In a world where everything is code, our security approach needs to change.

The world has changed. Has your security kept pace?

As Microsoft's CEO Satya Nadella is fond of saying: "Every company is now a software company." So, if your organization is like most, it's likely you are spending more on software development than in the past, and you are using a wider range of application technologies than ever before.

Your organization probably has a mix of traditional applications running on virtual machines, along with more modern applications running in containers orchestrated by Kubernetes. Some of your applications may leverage microservices which cut across cloud boundaries. As a result, your attack surface is more complex than ever.

Roles and responsibilities are also changing. Increasingly, your DevOps teams are being asked to shoulder much of the burden for application security. As a result, they have a stronger voice in deciding which security tools they want to use.

And the pace of software delivery is increasing. Some of your DevOps teams are envisioning the day when they can deliver new features on a weekly or even a daily basis.

Against this backdrop, consider:

- **Has your security approach changed** to accommodate new technologies and ways of working?
- **Are old security products and approaches still being used** and, if so, are they achieving the optimal outcome for new ways of working?
- **Is your approach working well for you,** or is it reflecting the fact that you haven't taken steps to modernize?

CISOs have been asking these questions, or ones like them, ever since the term "DevSecOps" was coined. Stated simply, DevSecOps is the seamless integration of security testing and protection throughout the DevOps lifecycle. And that is the focus of this white paper.

This whitepaper explores the various ways application security tools and processes need to change in order to address the agility needs of organizations that are implementing modern software methodologies. The goal is to help organizations better understand how they can evolve their security approach to better align with the directions they are heading, the risks they are exposed to, and the high level of automation that modern application development practices demand.

Software is eating the world

We have all seen the rapid changes in the world of software development. Organizations everywhere are trying to radically transform their businesses using software technologies. Modern applications are being used to drive revenue, stay connected with customers, and keep employees productive. Thus, the phrase "software is eating the world" is now part of our common lexicon and produces over 500,000 Google search results.

In our most recent survey of CIOs:

89%

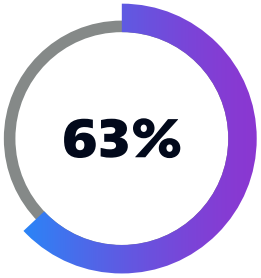
said their digital transformation had **accelerated in the last 12 months.**

58%

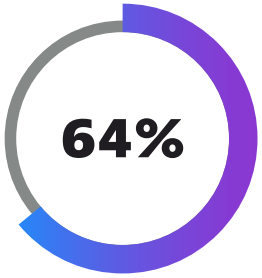
said the **speed of digital transformation would continue to increase.**

Unfortunately, greater speed usually has a negative impact on security.

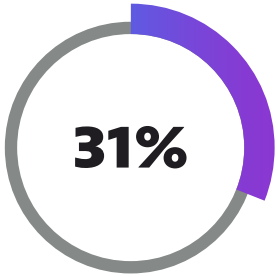
This became evident in our recent survey of 700 CISOs around the world:



of CISOs say new ways of working, such as DevOps and Agile development, have made it more difficult to detect and manage software vulnerabilities.



of CISOs say developers don't always have time to scan for vulnerabilities in their code and apply a fix before the application is pushed to production.



of CISOs say application and DevOps teams often don't work with the security team to avoid slowing down.

Four technologies enabling the acceleration

Underpinning this acceleration are several new technologies and methodologies that security professionals need to be aware of when designing a security architecture:

Containers, running in combination with orchestration systems such as **Kubernetes,** provide a more reliable and more automated way to deploy and scale applications than ever before. Containers can spin up and down quickly. The average lifespan for some containers is just minutes.

Unfortunately, the speed and relative opacity of containers are problematic for traditional security tools. In our recent CISO survey, 62% said container runtime environments have made it more difficult to detect and manage software vulnerabilities.

Open-source software is now commonly used to speed the development of custom applications. In fact, the average application consists of 70% open-source components, almost double what it was five years ago.* These reusable chunks of code are great for speeding application development, but they often contain built-in security vulnerabilities.

DevOps teams and automated workflows are increasingly being used to produce and deliver software more quickly and with higher quality. The traditional siloed functions of development, QA, and operations are now being merged into a single team. Developers are now responsibile for the quality of their releases, and they have the tools they need to monitor their applications at runtime. This creates fast feedback cycles and avoids inefficiencies such as when information is scattered across different teams.

High-performing teams are leveraging automation in everything they are doing. But if security tooling is not adequately automated, it will be seen as "friction" and be shunned by developers.

Hybrid multicloud environments, while not exactly a new technology, are now the dominant operating environments for most organizations. Our research shows that modern applications, especially those that leverage microservices, commonly extend across more than one cloud boundary, which is sometimes called a "hybrid cloud". Over three quarters (78%) of organizations tell us they are currently using multicloud or hybrid cloud environments to run their applications.

Unfortunately, traditional security tools often do not operate across operating boundaries. They are siloed, as described in the following section.

30% of global security decision-makers whose firm experienced an external breach said that **the attack exploited a software vulnerability.**

"The State Of Application Security, 2021", Forrester Research, March 23, 2021.

Are traditional approaches to security appropriate for today's high-speed environments?

When viewed against this backdrop of change, it's clear traditional approaches to application security are struggling. Here are the most common failure points:

Slow speed

Traditional security products were not designed for today's high-speed environments. They take a while to produce results — hours, or days if you count the time spent coordinating between the development team that wrote the code and the security team that operates the security products and delivers the reports.

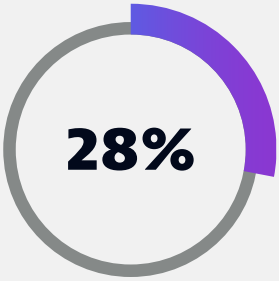
Another aspect of slow speed is scanners that operate periodically. Research conducted by IDC indicates that over half of enterprises scan their production environment for vulnerabilities no more frequently than monthly.[1] This is why the #1 priority of organizations who say they implement DevSecOps is to increase the frequency of code scans. They know that the speed of their containers vastly outpaces the speed of their security tools (see sidebar).

Lack of automation

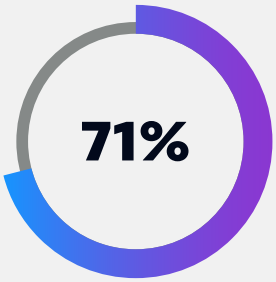
Many traditional security products need to be configured separately for each application. This lack of automation is a jarring mismatch with the automated world DevOps teams live in. This is one of the reasons why so many application developers say they intentionally avoid using security tools, and even avoid working with security teams.

Lack of automation is also a problem for security staff. They can never be sure every application has been properly instrumented and is being properly monitored. Just to answer the question "How are we doing?" or "Does a new vulnerability announcement affect us?" requires a lot of manual work. And, given the shortage of security talent and staffing, this extra work is particularly burdensome.

When speed kills security[2]



of CISOs say **application teams sometimes bypass vulnerability scans** to speed up software delivery



of CISOs are **not completely confident that applications have been fully checked for vulnerabilities** before going live in production.

[1] IDC, DevSecOps Adoption, Techniques, and Tools Survey, Doc #US47597321, April 2021

[2] Precise, automatic risk and impact assessment is key for DevSecOps, Dynatrace, 2021

Siloed and limited viewpoint

Most security products have a siloed view of vulnerabilities, and thus, they lack an integrated view of risk. This requires you to utilize multiple products — different products for different environments — and then stitch things together. This problem has been getting worse in recent years, not because security products are getting worse, but because operating environments have been getting more complex and applications more distributed, both of which exacerbate the problem of silos.

An example of a siloed security product is one that is sold by a specific public cloud provider. If your application is comprised of microservices that communicate across cloud boundaries, that security product will not be able to assess your entire application or understand what it connects to.

An example of a limited scope security product is a vulnerability scanner that can't see inside running containers to detect which libraries are actually being used by the application. As a result, the scanner produces either no information or generates many false positives.

This problem applies even to the latest generation of "cloud workload protection platforms" that were designed with containers in mind. They are only able to scan container images when they are at rest in registries or look at the Docker file manifest. Without knowing which libraries are being used by the application in runtime, and how the cloud workload protection platforms cannot accurately distinguish between a potential vulnerability and a real risk.

Comparing the characteristics of modern development environments to common security tools, several areas of incompatibility are apparent:

Modern technologies	Challenges
Containers	Short lifespan makes it hard for traditional security tools to monitor them in production environment, resulting in visibility gaps. Opaque to many traditional security tools which can't observe inside the application at runtime. Relying on just an inspection of the Docker file results in many false positive vulnerability detections.
Open-source software	Static application scanning tools can't identify which parts of the open-source software packages are actually used by the application, or how they are used. This results in many false positive vulnerability detections.
Rapid software development and release cadences	Slow security tools require developers to wait for results, which delays the release. Or, security tests will be intentionally skipped to remain on schedule. False positives waste time and frustrate developers.
Hybrid multiclouds	Many security tools can't see beyond cloud boundaries; thus, they can't give you a complete picture of your application, and don't let you enforce security policies consistently across boundaries.

How we get to DevSecOps: shift-left and shift-right

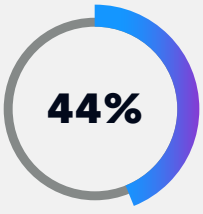
Since about 2015, the main approach to improving security for high-velocity DevOps teams has been to "shift-left" security, which means conducting security assessments early in the software development lifecycle. The urge to shift security left was caused by the following realizations:

- The increasingly ephemeral nature of containers left no time for traditional vulnerability scanners to identify vulnerabilities in the production environment.
- The immutable nature of containers meant that patching needs to be done by developers, not IT operations staff.
- It is far easier (cheaper) for a developer to fix a vulnerability if they find it sooner rather than later.

All of this was good. But now, enterprises are realizing it was not wise to completely abandon the production environment. We've seen many successful attacks against Kubernetes environments — from the malicious images that were inserted into Docker hub, to the attacks against Azure and Tesla, all stemming from "cryptojacking". As a result, 44% of enterprises are now planning to adopt new runtime security controls over the next 12–24 months.[3]

This renewed emphasis on the production environment is called **"shift-right"** security and is important for the following reasons:

- **Production is where most attacks happen,** so it makes sense to find new ways to monitor your running containers that are opaque to traditional security tools.
- **Scanning a static image, either in a repository or in a development environment, can't give you the same rich insights** you can get if you observe the application running in production. For example, you don't get to see what libraries are actually called, how they are used, whether a process is exposed to the Internet, or whether a process interacts with sensitive corporate data.
- **Most organizations do not have a perfect set of homogeneous CI/CD pipelines.** Some applications might go through the prescribed security testing during preproduction, but others may slip through without adequate testing. For example, maybe an important deadline left too little time for security testing. Or maybe an application was an off-the-shelf application that wasn't touched by your developers. These applications appear in production and represent an attack surface that needs to be assessed.
- **New zero-day vulnerabilities are discovered after an app has been deployed into production.** They lurk out there, exposing you to risk.



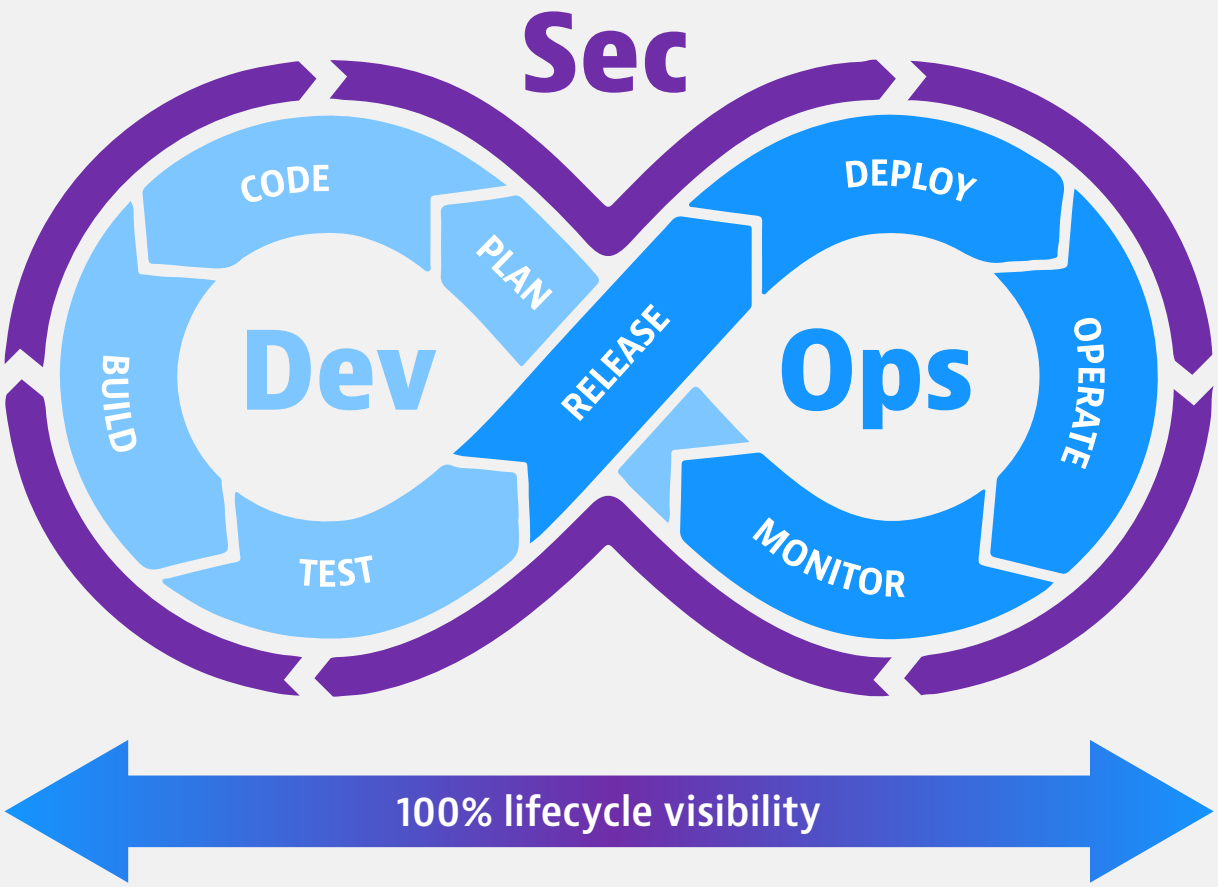
of enterprises are planning to adopt new runtime security controls over the next 12–24 months.

[3]ESG Research Report, The Maturation of Cloud-Native Security Securing Modern Applications and Infrastructure, March 2021.

Don't misunderstand us. We believe shift-left security is great. We are only saying it's a mistake to think the development end of the software development lifecycle is the only portion worth monitoring. Just as the DevOps infinity loop spans the entire software development lifecycle, good security also needs to span the entire SDLC.

Gone are the days when you could focus on your application and then leave the rest to a firewall. Today, a threat could potentially come from a service that's connected to another service which is connected to a third service.

Good security needs to span the entire software development lifecycle.



A cloud security checklist

Given the rapid changes we are seeing to technology and process workflow, what are the ideal characteristics of an enterprise application security program? Here's a checklist.

Fast, automated deployment and fast results
The security tool needs to run with no manual steps, no configurations, no custom scripts, etc. It needs to be present to provide information even to those application developers who actively avoid using security tools for fear it will slow them down, and to IT teams running COTS applications that your developers never touched. Only with 100% automated deployment everywhere can you truly feel confident in the information your system is giving you.

Wide scope
The security tool should function across all types of compute environments including containers, Kubernetes, serverless, PaaS, and traditional VMs.

All environments
The security tool should be able to assess applications running in hybrid cloud and multicloud environments. By reaching across boundaries, the security tool can properly understand the transitive dependencies and the "chain of risk" that occurs with modern microservice-based applications.

Full lifecycle
As discussed above, security tools need to be able to work across the entire software development lifecycle, in both pre-production and production environments. Static image scanning is not sufficient. Runtime visibility is required.

Proven low impact and stability
The security (e.g., the agent) product should place minimal demands upon the workload, and it should not disrupt the stability of the application.

Observability and contextual awareness
The security product needs to be able to see inside each workload to understand how each library is being used in order to distinguish a theoretical vulnerability from a real one. It also needs to see outside each workload to understand whether the vulnerability is exposed to an attacker. Finally, it needs some way to understand the importance of each asset, in order to measure the potential impact to the organization should the vulnerability be attacked and compromised.

Developer acceptance
Everything about the security program — which includes both products and processes — ultimately need to be accepted by the developers who will be tasked with remediating the issues that are found.

The convergence of performance monitoring and application security monitoring

In 2017, Gartner analysts Cameron Haight and Neil MacDonald analyzed what would happen if application performance monitoring platforms were used for security monitoring purposes. They believed a single agent could be used for both purposes, resulting in greater efficiency, less potential for system instability, and greater contextual awareness.[4]

This is now happening. To respond to the need for security that spans both production (shift-right) and pre-production (shift-left), Dynatrace has added security to its existing observability platform. The same agent that can provide deep observability for container performance can do the same for container security issues. This agent can provide rich information such as what libraries are called, how they are used, whether a process is exposed to the Internet, and whether an application or service interacts with sensitive data. This is much richer information than traditional security scanners or behavioral anomaly tools have been able to deliver in the past.

Dynatrace has built its Application Security Module within the Dynatrace Software Intelligence Platform. The advantages of this approach (as opposed to a bolt-on approach) include simplicity, efficiency, and stability. Dynatrace's existing customers do not need to deploy anything new; they have already deployed the Dynatrace OneAgent which can monitor applications for performance, reliability, and security. New customers need only deploy the Dynatrace OneAgent, which can be completely automated.

By leveraging Dynatrace's existing technologies, such as Smartscape and Davis AI, enterprise DevSecOps teams will be able to experience the following benefits:

Run fast and be secure. Our automated, real-time security aligns with your DevOps speed and cloud automation practices. No special coding, scripting, or configuration is needed. There's nothing a developer needs to do, which means there's nothing they could possibly forget to do.

Eliminate vulnerability blind spots. Our automated deployment provides real-time visibility and risk awareness to applications across their full lifecycle, from pre-production to production, in any operating environment. Dynatrace even provides visibility to third party apps and components which your developers never touched. If it runs, Dynatrace sees it.

Save up to 70% of the time your developers spend on remediation, allowing them to accelerate software delivery and reduce Mean Time to Remediation (MTTR). This time saving is achieved by the following:

[4]Gartner, "Application Performance Monitoring and Application Security Monitoring Are Converging", Cameron Haight and Neil MacDonald, 25 April 2017
(Note: The Gartner research report referenced herewith is archived and has been used only in the retrospective context, to the content.)

1. **Address risk, not vulnerabilities.** Instead of forcing security managers or developers to manually evaluate long lists of vulnerabilities to determine which ones to fix first, Dynatrace provides a prioritized list of the most important libraries to update. The Davis Security Advisor takes into account the following factors:
   - **Number of vulnerabilities** caused by each software library.
   - **Vulnerability severity,** which is based on the common vulnerability scoring system (CVSS) rating of each vulnerability and whether the relevant code is actually used at runtime in a way that exposes the vulnerability.
   - **Threat context,** which reflects whether there is a known public exploit for each vulnerability.
   - **Asset exposure,** which indicates whether the vulnerable code is communicating with the internet.
   - **Potential business impact,** which is determined by whether the processes are connected to sensitive data.

   Through this analysis, the Davis Security Advisor helps DevSecOps teams focus on the most problematic software libraries and open-source packages and remediate issues faster than ever before. These are actionable remediation recommendations at a level of sophistication never seen before. Completely automated.

2. **Remediation help.** Dynatrace Application Security includes an integration with Snyk's Intel Vulnerability Database which, for no extra charge, provides your developers with instant information about the required upgrades and/or potential workarounds to remediate each vulnerability. According to research sponsored by Snyk, this saves developers on average 8 hours per remediation.

Conclusion

As compute environments change, our approach to security needs to change too. The changes that have occurred over the past five years — shifting from traditional hosts to containers, and the rapid adoption of hybrid multicloud environments, plus the change from waterfall to Agile to DevOps — have put pressure on traditional security tools and resulted in blind spots and speed bumps.

Despite the recent efforts from security software manufacturers to modernize their products, surveys of enterprise security managers tell us that they are not impressed with the results. The products are still too slow, too manual, and too siloed. The information they produce is often riddled with false positives and, in many cases, the products still lack visibility into the right side of the DevOps environment — the production environment.

But now a radically new approach is possible, based on the convergence of observability and security. This convergence solves the blind spots and speedbumps DevSecOps teams have been experiencing with traditional security tools.

Existing Dynatrace customers already have the infrastructure in place to activate the application security functionality described above. New customers need only deploy the Dynatrace OneAgent to achieve complete performance, reliability, and security monitoring.

The value Dynatrace provides is not only faster and more automated results, but also fewer blind spots, better understanding of risks, more efficient remediation processes, and more confidence that the enterprise can deliver innovative software quickly without sacrificing security.

A radically new approach is possible, based on the convergence of observability and security.

About Dynatrace

Dynatrace provides software intelligence to simplify cloud complexity and accelerate digital transformation. With automatic and intelligent observability at scale, our all-in-one platform delivers precise answers about the performance and security of applications, the underlying infrastructure, and the experience of all users to enable organizations to innovate faster, collaborate more efficiently, and deliver more value with dramatically less effort. That's why many of the world's largest enterprises trust Dynatrace® to modernize and automate cloud operations, release better software faster, and deliver unrivalled digital experiences.

dynatrace.com blog @dynatrace

0714.21 13004_WP_USlet_cs