boxboat | 2021 White Paper

IMPROVING YOUR SECURITY POSTURE ON THE CONTAINERIZATION JOURNEY

When you make changes in your development workflow or technology, your security practices must also adjust. The move to containers is no different.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Whenever you make changes in your development workflow or the technology you use, your security practices have to adjust as well. The move to containers is no different. While containerized applications are neither inherently more or less secure than a legacy or VM-based application, they require a different approach to security. The risk is that as organizations move to containers they will keep the same security practices — and ultimately learn the hard way that doesn't work.

With proper planning, however, organizations can keep their applications secure both during and after the move to containers. Here's what you need to know to increase your organization's security sophistication on your container journey.

CONTAINER SECURITY IS DIFFERENT

Even though securing containers is not radically different from securing VMs, the paradigm shift that comes with moving to a containerized system leaves many engineers and security professionals totally confused about how to best approach container security. Especially when compared to legacy monoliths, though, there are some important differences. For one, container security doesn't rely on perimeters and firewalls to the same extent that monoliths do, instead focusing on configurations and access controls. "Where people get into trouble is that they forget that the container itself represents a new attack surface, as do the orchestrators that usually work with containers, whether that's Kubernetes or other types of orchestrators," explained Cindy Blake, senior security evangelist at GitLab. Here are some specifics about container security that teams don't always understand early on in their container migration.

> "Where people get into trouble is that they forget that the container itself represents a new attack surface, as do the orchestrators that usually work with containers, whether that's Kubernetes or other types of orchestrators."

CONFIGURATIONS. Configuration management is a critical part of container security. This can include whether or not a container runs as root (almost never a good idea) and if there are any limits on resource usage. Configuration is key to keeping containers secure but can also be overwhelming, because there are so many potential knobs to turn.

This is complicated by the fact that the defaults in Docker are insecure. By default, the container will run as root and have no resource usage limits. Secure configurations depend on developers actively changing these insecure defaults — or an automation tool changing the configurations. When security incidents do happen, it's often because of configuration errors in either the container, the orchestrator or the cloud provider.

CONTAINER IMAGES. When a developer builds a container image, he or she is generally assembling the image from pre-made components rather than creating something new from scratch. This helps speed up the image build process, but it can also lead to two security problems. First of all, ideally container images should contain only what is necessary for the container to carry out its function, but when developers use pre-made images they often contain unnecessary libraries and code. This increases the potential attack surface on the image.

COMPONENTS OF A STRONG SECURITY POSTURE IN CONTAINERS

So what exactly makes your container security posture secure?
Here are some general best practices:

- ☑ Use trusted base images
- ☑ Do not run containers as root and do not allow containers to acquire new privileges
- ☑ Use minimal base images
- ☑ Use a secrets management tool and don't store secrets in images
- ☑ Scan images throughout the entire lifecycle, from uploading to a registry to runtime
- ☑ Use automation tools to ensure configuration best practices are applied consistently organization-wide

Second, container images have to come from trusted sources, because the final image security is only as strong as each of its component parts. Setting organization-wide governance policies on acceptable image sources — as well as ensuring complete image scanning throughout the application lifecycle — is the best way to keep container images secure.

Finally, secrets should never be stored in container images — but sometimes are. Don't make this mistake.

ACCESS CONTROLS. Role-based access controls (RBAC) are an important part of the security posture both for limiting access to the image registries that you use to store finished container images as well as controlling access to applications during runtime. Access controls apply both to users and to the containers themselves, and both should always be given the lowest level privileges that still allow them to do their job effectively.

Access controls are more complex in a containerized environment, which increases the risk of errors if the process is handled manually.

NETWORK AND STORAGE. Because of the way containers and microservices work, containerized applications have higher network traffic and different ways of consuming storage. Many security breaches are related to unsecured data, either in the storage itself or during transit. Setting up data encryption that doesn't impact performance is important — because you don't want developers to intentionally not set up encryption so that the application's performance won't suffer.

RUNTIME. Perhaps one of the biggest differences between container security and VM or monolith security is in the runtime monitoring. Containers are ephemeral, which means that if the monitoring system isn't collecting data in real time, that data will be lost forever.

So how should organizations ensure that they can maintain a highly secure environment while migrating to containers? We've helped hundreds of companies adopt containers, and there are common best practices when it comes to ensuring a continually secure environment during and after containerization. We are technologists and we work with technologists, but we've found that excessive focus on technology and tools is the biggest security mistake companies make. Effective container security starts with cultural and organizational changes. Tools are important, but they can't replace a cohesive security strategy. Here's what that means.

START WITH A CULTURE OF SECURITY

Whether you're on a monolith, virtual machines or containers, your security posture is not determined by the tools you use, but by the cultural and organizational norms put in place. Tools are just that — a tactical tool to help make security easier, faster and more consistent. If key stakeholders don't believe that security is important, the best tools in the world won't keep the application safe.

Everyone knows that security breaches are embarrassing, expensive and harmful to the organization in the long run. But making the changes to ensure that they don't happen can be challenging — more challenging than simply purchasing a tool.

START WITH THE VISION

Since we're talking about the move to containers, now is a good time to mention that *no one should move to containers just for the sake of moving to containers*. Make sure there's a clearly articulated reason or reasons for migrating to containers — and those reasons should include both technical and business factors.

Container security should be a part of that vision — it has to be a critical aspect of the migration's overall success. When container security is framed as a key part of the overall container strategy rather than a last-minute imposition or a meaningless box to check it will be easier to get the organization-wide buy-in that leads to a successful container security strategy.

OBTAIN BUY-IN

When we work with companies on their container migration, it's common for software engineers to say things like, "Don't bring in the networking guys, they'll just open up a can or worms," or "Don't bring in the security guys, they don't understand containers." These types of comments are self-sabotage and can make the migration exponentially more challenging and risky, from a security standpoint.

To be fair, many of the security pros might not understand containers — the same can probably be said for many of the engineers on the team, too (more on that later).

To ensure they get buy-in, organizations need to start the container journey with involvement from every team that will be impacted by the migration. This includes networking, storage, compliance, security, infrastructure, development and business leadership. This is one reason an overarching vision is so important. When the migration champions have a vision to share with other teams — one that appeals not to narrow team goals like decreasing friction in the

CI/CD pipeline but that speaks to larger strategic goals like delivering applications faster than competitors — it's easier to get everyone on board.

As an organization works on a migration strategy, representatives from all relevant teams should be consulted. It's very rare for someone to have a concern that can't be addressed, and the sooner in the migration process questions about security, compliance or networking are resolved the faster the organization can start getting value from containers. In addition, it also reduces the risk of cut corners leading to compliance or security lapses.

At the strategy phase, everyone should also be aware of how this migration will impact departmental budgets. People get anxious about change, and they get especially anxious about changes they fear could erase their jobs. Part of the container migration can include changing the way that the company manages budgets, too. Instead of a siloed approach, in which there's one funding source for development, another for testing and yet another for logging and monitoring, a more integrated budgetary approach can help organizations apply the right financial resources at the right time.

> "You might have a really good engineer there, a great IT person there. There might even be someone internally who is a recognized thought leader in the space. But this kind of siloed knowledge isn't enough."

THE SKILLS GAP IS A SECURITY GAP

When companies first start to migrate, they generally do have some internal expertise with containers and other cloud native technologies, but that knowledge exists in a few pockets rather than throughout the organizations. "You might have a really good engineer there, a great IT person there," explained Will Kinard, CTO at BoxBoat. "There might even be someone internally who is a recognized thought leader in the space." But this kind of siloed knowledge isn't enough to ensure the migration will be a success and the applications will stay secure.

Up-skilling the entire engineering department should be a major focus of any container migration strategy. Again, it's important to think broadly about who will be impacted by the move to cloud native and what functions need to interact with the new technology. "You need to have the knowledge to know that you have an exposure, and then you have to have the skills to resolve

that exposure," Blake said. Not even knowing where potential security vulnerabilities might be lurking is a huge and often underestimated part of the security skills gap.

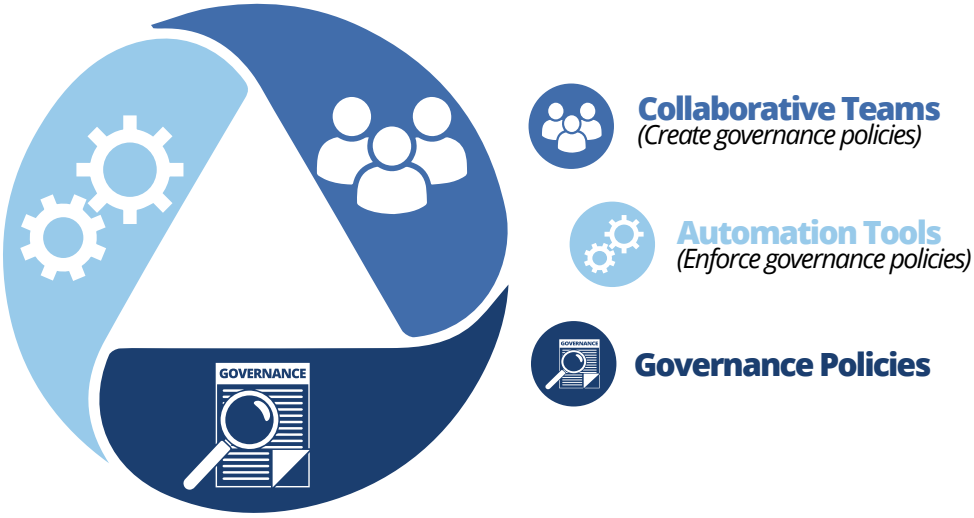
There are security implications for networking, storage and infrastructure as well as containers and microservices. It would be a mistake to only consider compute functions in your security strategy.

ADOPT TRUE DEVSECOPS

DevSecOps is more than just DevOps with security thrown in and more than shifting security left. Instead, it involves incorporating security into the entire application lifecycle, starting at the time code is created and running through the testing, deployment and production phases. DevSecOps also requires a profound organizational change in how security teams and DevOps teams collaborate — developers need to take more responsibility for security and security teams need to act more as advisors who cede some control over implementation. This is challenging for many organizations, especially given that most are not practicing true DevOps yet, either, and still have siloed development and operations.

DevSecOps also requires adopting the right tools that will facilitate collaboration without adding unnecessary complexity or tool overwhelm. When possible, it's generally better to have a single platform or pipeline where developers, security professionals and operators can see the entire application lifecycle without toggling between dashboards. When all members of a cross-functional team use the same tool to interact with the application it also decreases the risk of miscommunications between team members.

WHAT DOES SHIFTING SECURITY LEFT REALLY MEAN?



Organizations who are serious about container security need to get past buzzwords and lean in to the challenging changes that are necessary to keep applications security while also ensuring that the larger vision for containerized applications can become a reality.

INCREASE COLLABORATION

Developers, operators and security teams have to collaborate if organizations want to have a strong security posture. Unfortunately, there is a stereotype among developers that security pros are clueless people with clipboards making impossible demands. Security pros, on the other hand, often think of developers as careless code-spitters who write code without considering the security ramifications.

One problem is how developers are incentivized. In most organizations, developers are encouraged to be fast. Shipping more code, faster, is how they get promoted, get bonuses and get raises. Developer bonuses are rarely tied to whether or not a security vulnerability made it into production.

Security professionals, on the other hand, are rarely given bonuses based on how fast the developers ship their code, but rather how vulnerabilities are identified and remediated. If a security incident happens, they are more likely to face the consequences.

As a result, developers are generally very focused on speed, while security pros want everything to be perfectly secure. They tend to view these two goals as zero-sum.

When security is incorporated sooner in the process and security teams work as internal consultants instead of roadblocks, organizations can both deliver code faster and make it more secure. Creating team incentives that are shared between application team members can also help — both developers and security professionals can be evaluated on both how secure the application is as well as how quickly the team is able to deliver new functionality.

Shifting how everyone sees the relationship between security and development speed and changing incentive structures can go a long way towards improving the relationship between developers and security pros, ultimately leading to a stronger security posture.

EMBRACE AUTOMATION

Automation tools are critical to shifting security left and are also the only way that organizations can realistically expect to get both high development speed and tight security.

Security professionals who work with containerized systems should not be manually changing configurations or monitoring runtime behavior. Their role is as an internal advisor to educate developers about security best practices, put the right automation tools in place and develop governance policies.

Any manual changes security professionals make should only be to security tools that create policies and guardrails through automation. This allows developers to focus on writing code while ensuring that they don't accidentally create security problems.

Critically, security automation needs to be incorporated into the entire development lifecycle, from writing code through end-of-life. "People need to understand the whole gamut of end-to-end security," Blake said. "Because a common mistake is to optimize one area and forget about the rest."

The right automation tools will also make collaboration easier by providing a single source of truth to use throughout the application lifecycle and will simplify the learning curve related to adopting containers.

CREATE GOVERNANCE POLICIES

In a collaborative DevSecOps approach, security professionals should focus on policy creation and tool selection, not giving developers a security checklist or manually changing configurations themselves.

Containerized applications are incredibly complex. Without both clear organizational governance policies and the automation tools to consistently enforce them, errors are inevitable. These errors will happen regardless of how conscientious the developers are or are not — there are simply too many knobs to turn for everyone to get it right every single time.

How to best apply security policies, the trade-offs that might be involved and prioritization are all questions that security professionals are best equipped to answer.

DON'T NEGLECT RUNTIME SECURITY

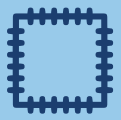
In all the discussion of 'shifting security left' it's important to remember that this shift is really an expansion of when security should be considered, not a true 'shift.' Traditional runtime security is still at least as important in containerized applications as it was in virtual machines or legacy apps.

Organizations need to have a strategy in place to address the following concerns:

Identifying and responding to CVEs in production

Monitoring for anomalous behavior

Collecting and storing metrics

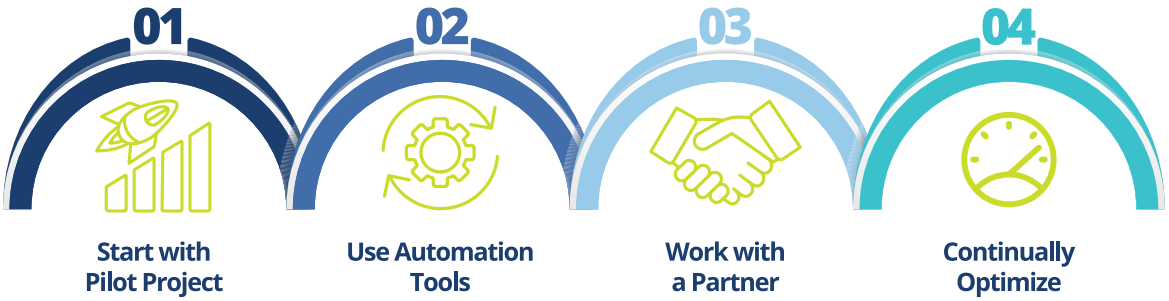


Automatic patching

Ideally, the DevSecOps incorporates development, security and operations. Again, there will be trade-offs to make when it comes to creating runtime security policies. Automatic patching, for example, helps ensure consistent security but also creates additional downtime risk. The right balance between operational risk and security risk is going to vary by organization and by individual application. Security and ops teams need to be able to communicate and collaborate so that they are making intentional choices about the right balance for each application.

HOW TO GET SECURITY RIGHT FROM THE BEGINNING

Organizations that are most successful in keeping their applications secure think about security from the very beginning of the containerization process, long before anything is going to be deployed. Here's how these organizations get security right.



START SMALL

Your entire organization is not going to become an elite DevSecOps performer overnight. However, you probably already have some pockets of expertise — use them.

The best approach to containerization involves starting with a small pilot project run with the best internal resources you have available. When you pick the right people and give them the resources they need to succeed — external support, budget and time — they can develop a process that works for the company's specific situation.

It goes without saying that this pilot team should include security specialists. In addition to working out how to move to containers, this team should also be figuring out the best way to collaborate between security, development and operations. This could include:

- Experimenting with communication methods
- Changing the metrics used for evaluating performance and setting compensation
- Using tools that facilitate collaboration and provide a single source of truth

Once the pilot project is successful, the organizational and technical changes can be gradually spread throughout the organizations.

EXPECT MORE FROM AUTOMATION TOOLS

A strong security posture in containers is simply not possible without extensive reliance on automation tools. But in most cases automation tools require customization and configuration to make them best fit your organizations workflows, architectures, policies and priorities. If you use these tools out-of-the-box, you'll only get about 80% of the potential value.

From the beginning, organizations should prioritize becoming experts in how to use their automation tools so that they can leverage those tools' capabilities to reduce the amount of manual effort required to secure their applications.

In addition, whenever possible organizations should try to consolidate tools. Using a single platform like GitLab to handle security automation instead of a dozen separate tools to handle individual security tasks decreases the learning curve for everyone in the organization and decreases the tooling complexity. When there are fewer moving parts, it's less likely that a critical step will slip through the cracks or that metrics won't be properly correlated because they require toggling between dashboards, leading to either missing a security incident or wasting time investigating a suspicious event that turns out to be benign.

GITLAB

GitLab is a complete, single source of truth DevOps platform that gives developers, security teams, operators and business leaders a single place to manage their application's lifecycle, including source code management, CI/CD, security and more. With GitLab, organizations can improve their development velocity, improve communication between teams and specialities and integrate security into the entire application lifecycle.

WORKING WITH A SECURITY-FOCUSED MIGRATION PARTNER

Working with a migration partner is a good way to get past the skills gap faster and avoid making mistakes at the beginning of your containerization process that could come back to haunt you months and years down the road. On the other hand, too many organizations find themselves stuck in analysis paralysis, unable to make decisions out of fear they'll make the wrong one.

Every migration to containers is unique. Each organization has a different tech stack, different security requirements and different priorities. There are plenty of resources available that outline general best practices, but those can't provide a complete roadmap that takes your organization's unique needs into account. "It's fairly easy to put together a vanilla reference architecture for how to migrate to containers from a monolith," explained Kinard. "But within the first day or two you

realize that is an academic exercise, because there are so many options. It can be very challenging to pick the right tool, the right framework, the right process, the right governance policies."

Working with a migration partner who has seen hundreds of migrations can help organizations get past the fear of making the wrong decision — and actually reduce the risk of that happening. In addition, a migration partner can draw on extensive experience to develop a tailored migration strategy that takes the organization's unique needs and priorities into account.

CONCLUSION

For most organizations, the biggest challenge when it comes to securing their applications during and after the move to containers and to adopting a DevSecOps approach is organizational and cultural, not technical — though there are also technical challenges and skills gaps that are not insignificant, either.

The most successful organizations work with a migration partner to help them identify the best ways to change their organizational culture and align incentives between security, operations and development while also helping them select the best tools and get the most out of those tools. Working with a partner eliminates much of the uncertainty about the containerization and DevSecOps journey, giving everyone more peace of mind and decreasing the container migration's time to value.

When they embrace automation tools and organizational change, teams can get both improved security and better development velocity.

HOW TO SELECT A MIGRATION PARTNER

Working with a migration partner can speed up your time to value and help set the organization up for long-term success with DevSecOps and containers. But not all migration partners are the same. Here are some factors to consider:

- ☑ Does the partner focus on just technology or do they also help with organizational and cultural change?
- ☑ Does the partner have experience working with security-conscious organizations, especially those in regulated industries?
- ☑ What technology partnerships does the partner bring to the table?
- ☑ Does the partner have experience building relationships between departments within companies and getting buy-in from department leaders?

BoxBoat Headquarters
7910 Woodmont Ave., Suite 230
Bethesda, MD 20814
202.810.9570
www.boxboat.com

ABOUT BOXBOAT

At BoxBoat, we specialize in helping security-focused organizations embrace the digital transformation and get maximum value out of containers. We facilitate the organizational, cultural and technological changes that accelerate innovation, improve development velocity and encourage all types of stakeholders to work collaboratively to build value. Our consulting approach is centered around building a cohesive strategy that involves collaboration between development, security and operations, making technology decisions based on each organization's specific needs and helping teams overcome the skills gap sooner.

Get in touch to see how we can help your organization.

info@boxboat.com