DAYBLINK

Container Security

A Panacea Becomes a Vulnerability

Introduction

Over the past 5 years, container-based architecture has become the unquestionable standard for nearly all applications. Yet, less than half of organizations classify their security strategy for containers as Intermediate or Advanced. In addition, executives cite security as the top concern for why containers have not been more widely adopted by their firms. For many companies, containerization may be the doorway to digital transformation as it enables enterprises to develop, deploy, and deliver their applications faster — providing greater agility and efficiency over traditional software development methodologies. However, the many benefits of containers will not be realized without improved security practices that assuage decision-makers' security fears and allow them to comfortably provide the green light to containerized production workloads. This paper considers the challenges that enterprises face as they adopt this new technology and offers suggestions to building a better container security strategy.

What is a Container?

A container is a construct designed to package and run an application or its components on a shared operating system. Containers are isolated from other containers and share the resources of the underlying OS, allowing for efficient restart, scale-up, or scale-out of applications across clouds.[1] Prior to containers becoming a popular option for virtualizing applications, the de facto choice was to deploy applications on virtual machines (VMs).

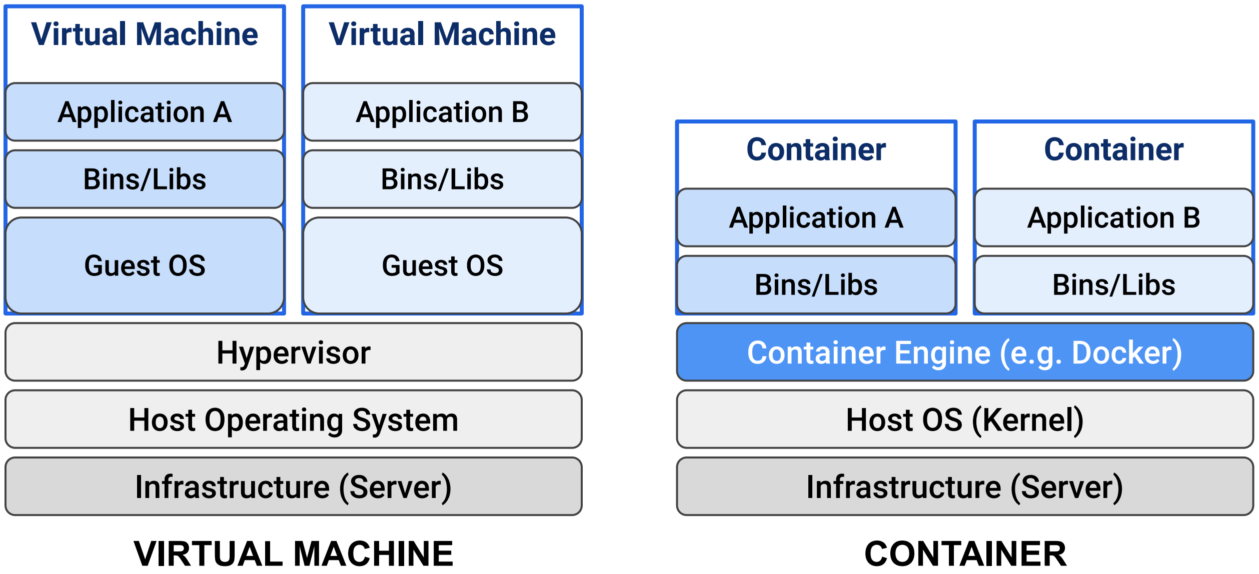


Figure 1

As represented in the comparison above, a virtual machine abstracts the OS from the underlying server hardware providing the ability to run multiple isolated machines with different operating systems and applications on the same server. Similarly, containers also package applications and their dependencies into a single unit. However, they do so by abstracting the application from the OS and allowing multiple isolated applications to share a single OS, referred to as the “kernel.” Because the kernel is shared, each container image only needs to include what is required to run an application (code, runtime, system tools, system libraries, settings, etc.), whereas a single VM would need all of these components in addition to its own OS. This difference makes containers more lightweight than VMs — meaning they can be deployed in a fraction of the time, need fewer hardware resources, and shorten the application development lifecycle.[2]

The Rise of Containers

These are only a few of the benefits driving the rapid shift to containers. Numerous recent surveys of IT professionals, from engineers to senior leadership, indicate that their enterprises are moving at a lightning pace to adopt the technology at scale.

ADOPTION BY THE NUMBERS

89%

is the projected 2-year increase in container adoption by the end of 2020

(RedHat) [3]

49%

of organizations have containerized over a quarter of their applications (up 8% in 6 months)

(StackRox) [4]

41%

of companies invested over $500,000 in container technology in 2019

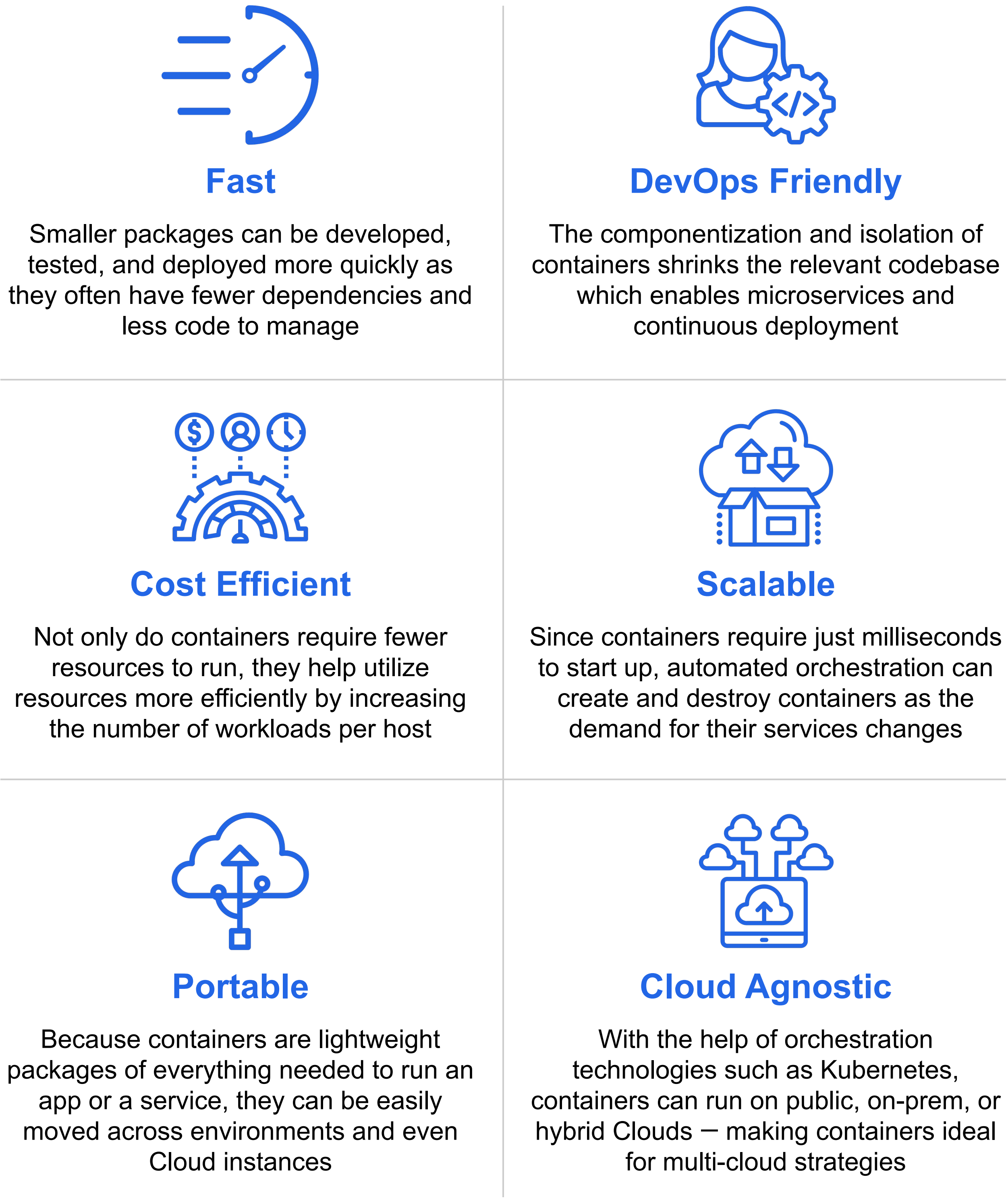
(Portworx & Aqua) [5]

Figure 2

These surveys only confirm what developers, DevOps, and security practitioners already know: containers are here to stay. The adoption rates indicate that containers are more than just an additional option when selecting how to virtualize — they are becoming a core technology to enable digital transformation. For corporations to become more innovative and capitalize on fast-changing customer demands, a nimble IT infrastructure is no longer a benefit but a requirement. The ability to pivot more quickly than competitors has never been more valuable and reinforced by the impact of the COVID-19 pandemic. Containers help build the foundation to establish this agility. They allow applications to be deployed and scaled faster, easily moved across public, private, and hybrid clouds, and accelerate the development lifecycle.

These, along with the other top benefits driving adoption, are shown in Figure 3.

Container Characteristics Driving Adoption



Fast

Smaller packages can be developed, tested, and deployed more quickly as they often have fewer dependencies and less code to manage

DevOps Friendly

The componentization and isolation of containers shrinks the relevant codebase which enables microservices and continuous deployment

Cost Efficient

Not only do containers require fewer resources to run, they help utilize resources more efficiently by increasing the number of workloads per host

Scalable

Since containers require just milliseconds to start up, automated orchestration can create and destroy containers as the demand for their services changes

Portable

Because containers are lightweight packages of everything needed to run an app or a service, they can be easily moved across environments and even Cloud instances

Cloud Agnostic

With the help of orchestration technologies such as Kubernetes, containers can run on public, on-prem, or hybrid Clouds — making containers ideal for multi-cloud strategies

Figure 3

Security Challenges Delaying Benefits

These benefits provide clear advantages over conventional software development and application deployments, enabling containers to be heralded as a panacea for developers and DevOps. However, container benefits can only be realized if containerized deployments are not delayed due to security concerns. In a recent StackRox survey, 44% of respondents indicated they have delayed a containerized deployment because of security risks or uncertainties.[4] The same survey captured that faster application development and release is the most valuable benefit of container environments, which is neutralized if deployments are stalled.

CHALLENGES BY THE NUMBERS

44%

of companies are limiting container adoption due to security concerns

(StackRox) [4]

94%

of IT professionals reported at least one type of container security incident or issue in the past year

(StackRox) [4]

47%

of organizations know they have vulnerable containers in Production

(Tripwire) [6]

Figure 4

While some organizations are delaying container deployments, others may be responsible for reinforcing security as a real concern. Eager IT professionals and executives who want to immediately reap the benefits offered by containers sometimes skip an in depth assessment of the organization's security posture. Even those who are risk-neutral may find themselves moving forward with containers in production that have vulnerabilities unknowingly, while others are ready to accept that risk. A container security report from 2019 found that 47% of respondents stated they have containers deployed in production with known vulnerabilities.[6] Like many new technologies, container security is only now being pushed to the left because it has become mainstream.

Vulnerabilities Generating Concern

As companies continue to deploy with known vulnerabilities, the practice is creating a culture that encourages reactive security over proactive security. To flip this script, it is important to begin by identifying the common threats and vulnerabilities that containers can create and understand the concerns driving container security strategies.

Top Threats and Vulnerabilities

- **Lack of Isolation:** In traditional virtualization technology, a compromised VM will not expose others since they run on separate OSs. Separation between containers is much thinner due to their shared kernel, allowing attackers to move horizontally if one container is compromised.

- **Misconfigurations:** While misconfigurations are not unique to containers, they are more common due to the increased complexity of container environments as well as the knowledge and skill gap that still exists in many teams.

- **Image Risks:** Images create numerous security challenges. Because they are a snapshot of an app, their components can become stale and quickly expose vulnerabilities. Images also require configuration and when configs are done incorrectly, attack vectors are opened. Additionally, since images can be easily ported and reused, developers are more likely to use non-compliant images.

Top Concerns

- **Inability to Assess Risk:** There are multiple points throughout a container's lifecycle where it is challenging to quickly determine risk.

- **Staff Expertise:** To complete tasks faster, inexperienced cloud administrators and developers may shortcut security by removing protections, skipping configuration steps, or making unintentional mistakes that increase exposure.

- **Investment:** Despite investment in container security increasing substantially, 37% of companies feel that their strategy doesn't invest enough.[4]

- **Executive Understanding:** Executive teams may make misinformed decisions on container strategies without an in-depth understanding of the technology and the associated security risks.

Countermeasure Prioritization

The threats, vulnerabilities, and concerns outlined on the previous page are just a few of the considerations shaping container security strategy. To combat these, agencies and organizations, such as the National Institute of Standards and Technology (NIST) and the Center for Internet Security (CIS), provide guidelines on steps that companies can take to improve their container security posture. However, there is little guidance on where to concentrate time and effort to get the most value. The tradeoff between the benefits and risks provided by containers is evident, but what is the easiest, quickest, and cheapest path to slant the tradeoff favorably and begin deploying containers sooner with more confidence? Security organizations can use the example in Figure 5 to begin thinking through the prioritization of their initiatives. The effort and impact of the countermeasures mapped here could vary widely depending on the current and target states of your container security program.

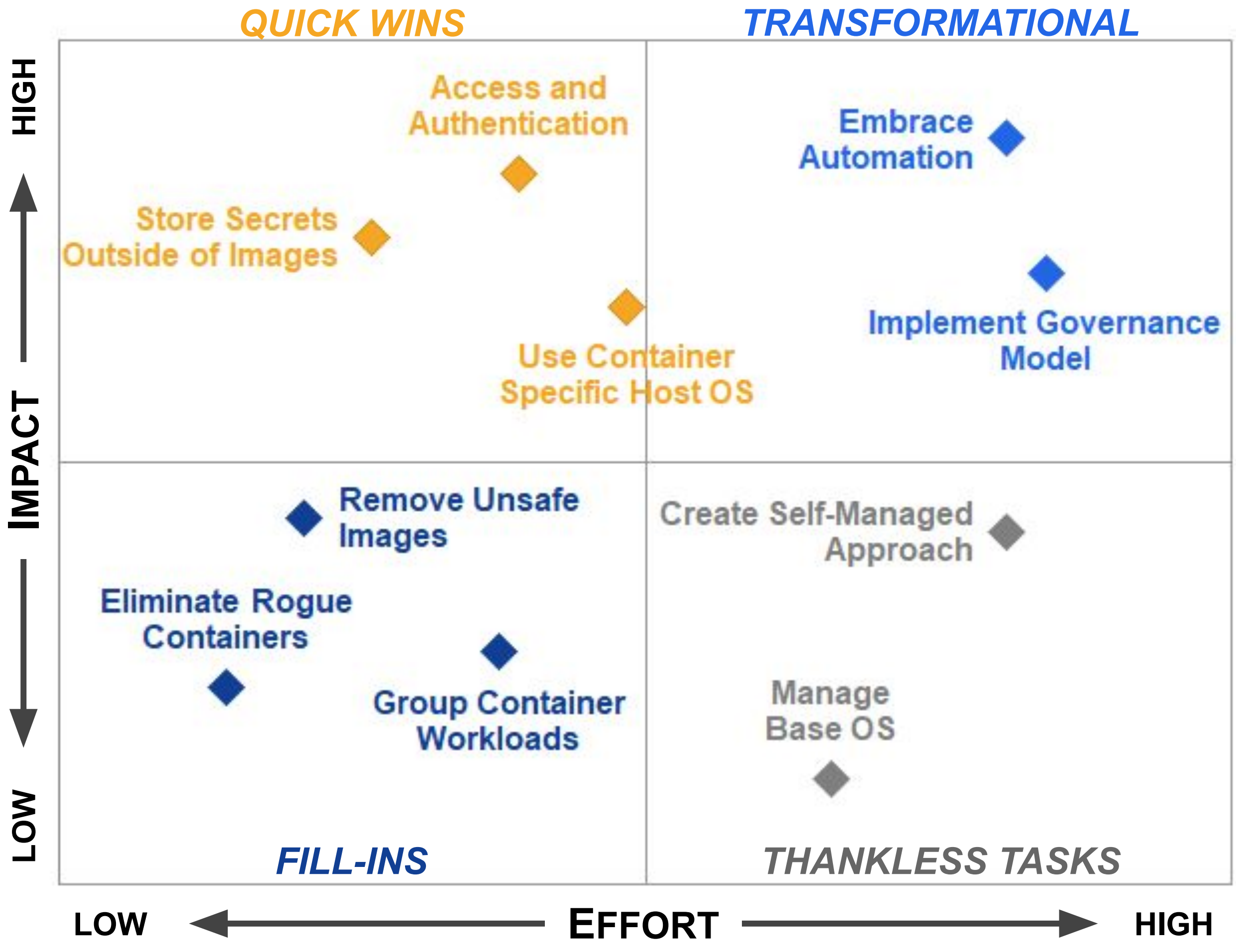


Figure 5

Countermeasures to Protect Containers

Each of the countermeasures listed in Figure 5 are detailed below to provide more insight. The countermeasures provided are not a comprehensive solution to a container security strategy. Instead, the goal is to present a framework for identifying the quickest path to securing your containers using these countermeasures as examples.[7]

Quick Wins

Access and Authentication: Improper or misconfigured access controls pose risks to image registries and orchestrators. For access to both, organizations should leverage single sign-on to their existing directory, practice least privilege principles, and ensure access and authentication is logged and audited.

Store Secrets Outside of Images: Visible secrets in images leave them susceptible to intruder theft in a container using the image or the image registry. Orchestrators like Kubernetes will manage secrets so that they can be secured and introduced at runtime securely. It is also possible to use an existing secrets management system and introduce them via an API call as needed.

Use Container-Specific Host OS: As many organization make the transition from using VMs to more container-based workloads, they do so without adopting new technologies and tools that are designed for containers. An OS designed for containers, such as Fedora CoreOS, will have fewer services enabled and file systems that are read-only which limits an attackers ability to gain access to the kernel and infiltrate other containers running on it. If application teams require general-purpose OSs, these can be secured by adding container-specific functionality to them. Regardless of the options implemented, this will substantially shrink your attack surface.

Transformational

Embrace Automation: Containers and Kubernetes have a large number of options when it comes to configuration, meaning the risk of human error increases. As cloud providers put more and more power in the user’s hands, the door is opened for misconfiguration which was the most frequent cause of breaches in 2019. The majority of configuration errors can be avoided by implementing automation.

Implement Governance Model: As more companies elect to use multiple public cloud providers along with on-prem and hybrid clouds, the number of security models and controls has increased greatly. While containers can be ported across clouds easily, a safe container in one provider may not be safe in another. Having a governance model in place that applies to all clouds will help teams architect their applications to be safe regardless of the cloud instance where they are deployed. That being said, an appropriate governance rule may require applications deployed via containers to have their architecture reassessed when moving from one cloud to another to ensure compliance.

Fill-Ins

Remove Unsafe Images: Even if new images are checked for vulnerabilities prior to being added to secured registries, they can inevitably become unsafe as they age. To ensure that these images are not deployed, an automated clean up of stale images based on timestamps, labels, or naming conventions should be scheduled. This requires that operational processes are established and followed by image developers.

Group Container Workloads: In addition to using container-specific host OS, it is best practice to avoid running containerized and non-containerized workloads on the same host. It becomes more complex and challenging to implement countermeasures specific to containers when hosts have mixed workloads. As an additional step, the container workloads can be further segmented based on their characteristics. Where possible, containers should share a kernel only when they have the same purpose, sensitivity level, and threat posture.

Eliminate Rogue Containers: If an environment has containers that are unbeknownst to security teams, it is possible they will be missed during vulnerability scans or have been misconfigured. This is found most commonly in development environments where teams launch containers frequently as tests. To avoid the creation of untracked containers, access controls should be implemented in all environments (dev, stage, and prod) and container deployment activities and user identities must be captured and audited to identify containers that have not been destroyed after creation.

Thankless Tasks

Create Self-Managed Approach: Despite staff training and knowledge being a top concern for many professionals, many organizations will elect to self-manage components of their container environment to account for unique IT systems or processes. It is generally more valuable and requires much less effort to leverage managed services and tools that are available.

Manage Base Operating Systems: Another initiative that is sometimes more work than it is worth is managing the components of base OSs. The majority of vulnerabilities found at the OS level can be mitigated by using container-specific OSs. Companies should first ensure that generic OSs are not used for containers since they will make this task unnecessarily complex and time consuming. If container-specific OSs are being used, organizations should take advantage of vendor-provided and recommended updates to ensure that the OS components have the latest security features installed.

Final Thoughts

For most organizations, it is imperative to increase adoption of containers in order to stay ahead of competitors. However, many are moving forward with haste rather than caution, leaving production containers unguarded. Under opposite conditions, there are companies that are not able to take advantage of the benefits containers provide because security fears make their security teams and leaders nervous.

The key is to understand container technology, your acceptable level of risk, and how to rapidly strengthen your security posture. This endeavor is nontrivial. Even mature organizations can struggle to recognize the implications of the tradeoffs associated with container security decisions. While this paper only discussed a handful of the many countermeasures you can take to protect containers, there are many others that may be worthwhile listed in publications such as NIST 800-190. Once a list of possible countermeasures is understood, determining which to implement first is dependent upon the required effort and expected impact. These metrics may vary widely for each activity depending on the gap between current and target states, but determining the priority of countermeasures is critical. Without focusing on the highest value and easiest to implement initiatives, organizations will find themselves delaying the benefits of containerized architecture, or worse: deploying containers at risk.

References

(1) Karmel, A., Chandramouli R., Iorga M. (2016). *NIST Definition of Microservices, Application Containers and System Virtual Machines* (NIST Special Publication 800-180). U.S. Department of Commerce, National Institute of Standards and Technology. https://csrc.nist.gov/publications/detail/sp/800-180/draft

(2) Google Cloud Platform. (2020). *What are containers and their benefits?* Google LLC. https://cloud.google.com/containers

(3) Dawson M. (2018, December 18). Red Hat Global Customer Tech Outlook 2019: Automation, cloud, & security lead funding priorities. *Red Hat Blog*. https://www.redhat.com/en/blog/red-hat-global-customer-tech-outlook-2019-automation-cloud-security-lead-funding-priorities

(4) StackRox Security. (Winter 2020). *The State of Container and Kubernetes Security.* StackRox, Inc. https://security.stackrox.com/rs/219-UEH-533/images/State_of_Container_and_Kubernetes_Report.pdf

(5) Portworx and Aqua Security. (May 2019). *2019 Container Adoption Survey*. Portworx, Inc. & Aqua Security Software, Inc. https://portworx.com/wp-content/uploads/2019/05/2019-container-adoption-survey.pdf

(6) Tripwire. (January 2019).*Tripwire State of Container Security Report.* Tripwire, Inc. https://www.tripwire.com/-/media/tripwiredotcom/files/white-paper/tripwire_state_of_container_security_report.pdf

(7) Souppaya, M., Morello, J., Scarfone K. *Application Container Security Guide* (NIST Special Publication 800-190). U.S. Department of Commerce, National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf

About the Authors

Zachary White is a Senior Consultant within DayBlink's Cybersecurity Center of Excellence specializing in private, public, and hybrid cloud strategy, cloud security, and cloud application deployments. He is based in the Vienna, Virginia office.

Shelby Balius is a Manager within DayBlink's Cybersecurity Center of Excellence specializing in Organizational Transformation. She is based in the Vienna, Virginia office.

Justin Whitaker is a Partner and Practice Lead of DayBlink's Cybersecurity Center of Excellence and is based in the Vienna, Virginia office.

About DayBlink

In today's cybersecurity environment, the threat landscape is rapidly evolving. It's outpacing the current defensive resources and skill sets of most corporations – meaning many companies are falling victim to attacks by malicious agents. The way we do business is also changing – with more data stored, living in the cloud, and constantly demand on the go. Breaches can mean losing clients and customers overnight.

DayBlink works with clients to improve their security posture. We assess threats and vulnerabilities, identify organizational risk, prioritize remediation efforts, and implement solutions to secure IT environments and critical assets from sophisticated cyber-attacks.

DAYBLINK

dayblink.com/services/cybersecurity/

cybersecurity@dayblink.com

866.281.4403