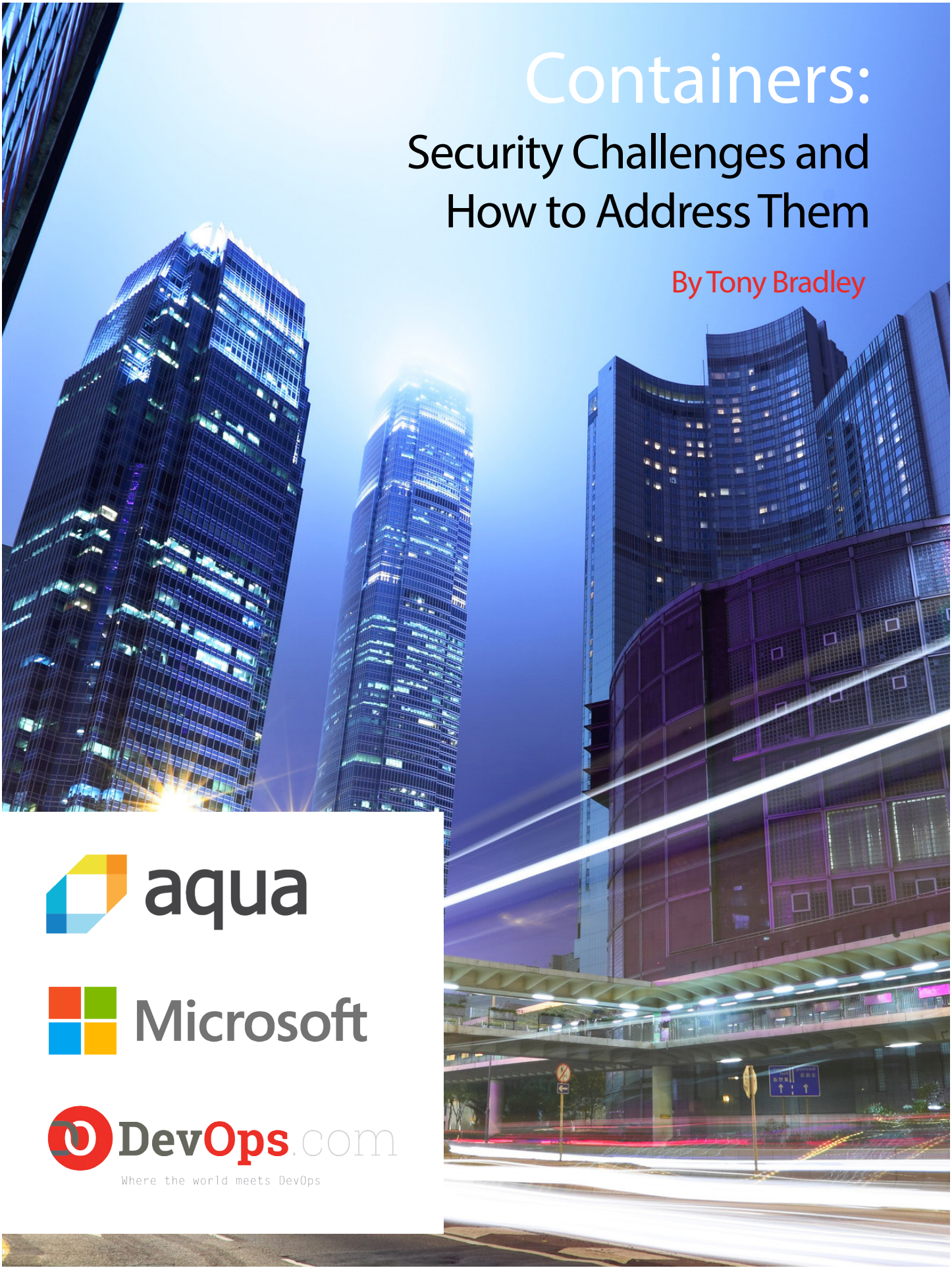


Containers:

Security Challenges and How to Address Them

By Tony Bradley

aqua

Microsoft

DevOps.com

Where the world meets DevOps

Table of Contents

Containers: Security Challenges and How to Address Them 3

Security Issues Introduced by Containers 4

Security Principles for Protecting Your Container Environments 6

Harden the Host Environment 9

Steps to Secure Your Containerized Applications 9

Wrapping Containers with Security 12

Use Containers with Confidence and Peace of Mind 13

About the Author 13

Containers: Security Challenges and How to Address Them

By Tony Bradley

Containers recently exploded onto the IT scene, but they aren't truly new. The concept of containers has existed for years within Linux, but Docker and the rise of DevOps have propelled containers from obscure technology to mainstream acceptance to business imperative in a matter of just a few short years.

"Containers and the container ecosystem provide the automation tools and processes for doing DevOps," explains Todd Fine, CEO of Wintellect. "Containers also support microservice architecture by providing a simple, lean way to deploy applications. The drive behind these two paradigms is to improve efficiency and security around application development and operations."

"Containers and the container ecosystem provide the automation tools and processes for doing DevOps."

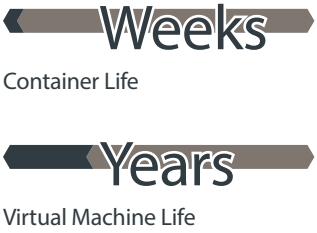
— Todd Fine
CEO,
Wintellect

Containers have a significant impact on productivity and efficiency. Businesses can spin up hundreds to thousands of containers per host on demand—each booting in as little as 1/20th of a second as Google claims to have already achieved. IT environments spins up a fraction of that in VMs per host, and at a significantly slower speed. Containers enable companies to increase server utilization, decrease overhead management, ease application portability and enable micro services.

A number of problems companies have from a variety of different perspectives can all be solved with containers. A container is a compact, portable software package that includes everything needed for an application to run—code, system tools, libraries—without extraneous or unnecessary code or services.

Containers are similar in concept to virtual machines. However, containers are much smaller and more efficient. In order for an application to run in a virtual machine, the virtual machine runs on a guest operating system, and it requires a hypervisor to manage on a server. Containers are much smaller and more efficient because they share the operating system kernel with other containers on a host via API calls.

Measurement of Container Life vs. Virtual Machine Life



Patrick Lang, program manager of Windows Containers for Microsoft, describes containers as another form of virtualization that is focused specifically on the needs of the application. He notes that with containers it's easy to identify what is in your application and know that it is always going to work the same way. "It's a better way of virtualizing applications, and it's fundamentally better than virtual machines."

Fine says, "The life of a container is typically measured in weeks versus virtual machines whose lives can extend to years. It is not unheard of to have software running on system that are even decades old. Shorter patch cycles can reduce security risks that stem from unpatched, unmaintained, and unsupported software."

Beyond being more compact and efficient, containers also enable applications to be deployed consistently across different operating system environments. The container includes the runtime code needed for the application to run—including any configuration files, dependencies, libraries. Containers eliminate the differences from different operating systems or OS distributions so the application will run consistently no matter where the container is deployed.

Security Issues Introduced by Containers

There is a general lack of awareness of existing container security concerns and best practices. Organizations need to understand the security issues that arise due to the differences in how VMs and containers function. Enterprises need to prepare for the glut of additional files that need protection with containers and the unwieldy nature of third-party libraries that containers use. Businesses must also consider configuration mistakes including those that grant root status to containers or simply make containers overly complex. Most importantly, organizations that adopt containers need to accept responsibility for security them, and should expect to keep tabs on new container vulnerabilities as the industry discovers them.

While the sharing of the host operating system kernel is one of the primary benefits of containers, it is also the crux of the problem when it comes to security concerns with containers. The lack of proper isolation between containers and the kernel during runtime means that a vulnerability that exists in the shared OS kernel can be leveraged to gain access to or exploit the containers.

Fine stresses, "Containers provide abstraction at the kernel level rather than the hardware level that virtual machines provide. The concern as it relates to containers is that a compromised kernel can enable unwanted access to all the containers running on a particular container hosts."

Privileged containers are also an issue. Running privileged containers is a vulnerability in and of itself. "Privileged containers run as root. If a malicious user or workload escapes in a privileged container, the container will then run as root on that system," says Dustin Kirkland, manager of Ubuntu Product & Strategy for Canonical.

> "Privileged containers run as root. If a malicious user or workload escapes in a privileged container, the container will then run as root on that system."
>
> — Dustin Kirkland
> Manager of Ubuntu Product & Strategy, Canonical

Another element of concern when it comes to securing containers is their inherently volatile and dynamic nature. Hundreds or thousands of containers can be created or destroyed in an instant to scale with demand. They are often short-lived and have dynamic IP addresses. The sheer volume of potentially vulnerable endpoints makes identifying and resolving security issues a Herculean task.

There are also two different security concerns resulting from how containers are typically developed and managed. DevOps and containers rely heavily on open source components. Sometimes that means running open source tools or within a container, and sometimes that means pulling in snippets or elements of code from a larger open source project to perform a specific function or service.

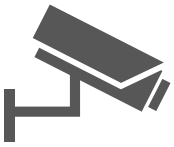
In either case, there is some risk involved. The use of un-vetted open source tools or components exposes the container ecosystem to risk. The other issue is just trying to keep up with identifying and mitigating new vulnerabilities in existing containers as they are discovered in the open source tools they utilize.

The volume of containers and volatility of the container environment are another concern. In a DevOps world of automation, continuous integration and continuous delivery, developers are constantly creating, modifying, and updating container images. Keeping up with the rapid pace of change can be daunting when it comes to managing security risk.

It is difficult to project how container breaches will scale when the public record of successful breaches is sparse or nonexistent, but all the right elements for catastrophe are there. "Modern application container software,

which is not yet even 3 years old, still has some immaturity challenges, including software bugs and vulnerabilities," says Jay Lyman, Research Manager for Cloud Management and Containers at 451 Research.

Security Principles for Protecting Your Container Environments



Use Active Detection

Securing containers has to be viewed and managed holistically. As much as 80 percent of the workload in containers is dedicated to the application, but it's important to protect the infrastructure—the operating system and container engine—as well as the application itself.

When it comes to computer and network security, early detection is crucial. The earlier in the process you can identify and mitigate a security risk, the easier it will be and the less impact it will have on the application or infrastructure.

With early detection as a foundational goal, there are three driving principles for protecting your container environments.

1. Use preventive measures. Harden your containers and container environment. Use least privilege access and minimize the potential attack surface to reduce your overall exposure to risk.

2. Use active detection. You can't just implement a container environment, deploy containers and forget about it. The container ecosystem is volatile by nature, so you must employ active security measures and monitoring to ensure the containers are always protected.

3. Use active response. Detecting issues is only half the battle. You also need tools and processes in place that enable you to actively respond. You need to stop malicious activity wherever possible and mitigate issues to limit damage elsewhere.

Amir Jerbi, CTO at Aqua Security, strongly emphasizes the need for active monitoring and response as opposed to relying solely on preventive measures like vulnerability management or host hardening. "The analogy is like thinking that your apartment is safe because you locked the door and shut the windows, without using home monitoring or an alarm system – but unlike a burglary, IT security breaches can go undetected for months without proper monitoring," cautions Jerbi.

Container Security **Made Simple**

Aqua Security helps you navigate the complex world of containers with a platform that automates and simplifies application security in containerized environments, whether on-premises or in the cloud.

Vulnerability Management
Automatically scan images, integrate with your CI/CD tools

Advanced Threat Mitigation
Detect anomalies and stop multiple attack vectors

Runtime Protection
Enforce container isolation and network nano-segmentation in production

Visibility and Compliance
Gain visibility into container and user activity, with full audit trail

To learn more, visit www.aquasec.com

contact@aquasec.com +1 (415) 946-4058

aqua

Scale and ship your app faster: apply your knowledge today

Build your first container using Docker on Hyper-V

Free, hands-on virtual lab

No download or installation required

Visit aka.ms/trycontainers

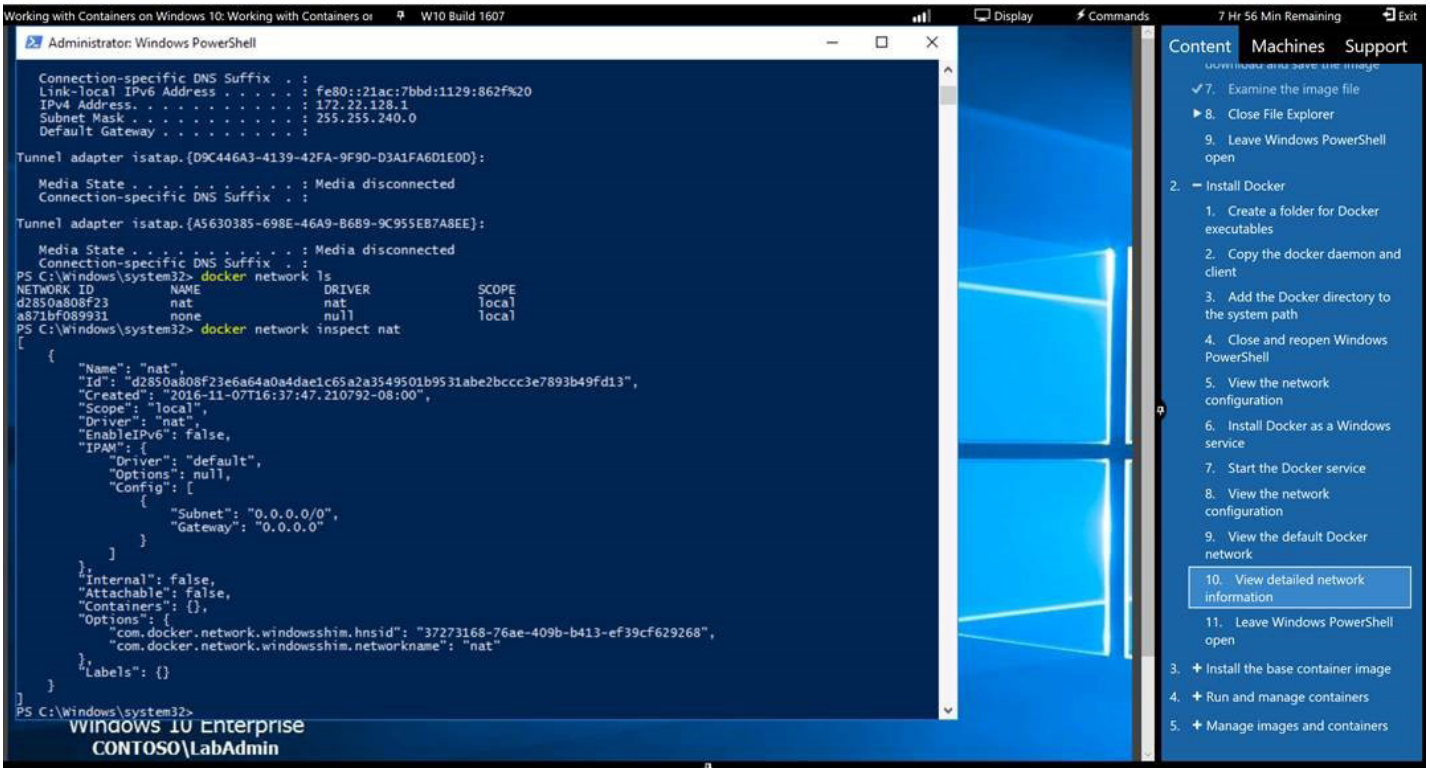
Now that you've learned how to use containers securely, why not put your knowledge into action? See firsthand how containers can support faster software delivery and help your team accomplish more with your valuable time and resources.

Apply what you've learned

Microsoft has created a free, hands-on, virtual lab that walks you through the basics of installing, deploying and managing a Docker container. You'll also learn the fundamentals of incorporating Docker on Hyper-V into your current development plans.

It's an easy way to get started with containers – and there's no setup or installation required. In less than an hour, you can build, deploy and manage your first container.

Start your free training today.



Microsoft

Jerbi also stresses the need for monitoring that is container-aware. "Containers add a layer of obscurity that reduces visibility. You have an operating system running a container engine, which in turn runs containers. The OS is not aware what containers are running—it only sees the container engine. The container engine knows what containers are running, but has no clue what the containers are actually doing. So, if you're running a host-based security tool to monitor the OS, you will not see what containers are running and what they're doing."

Harden the Host Environment

The first step in running a secure container ecosystem is to make sure host environment is hardened to minimize exposure to risk.

The first step in running a secure container ecosystem is to make sure host environment is hardened to minimize exposure to risk. For starters, you should ensure that the operating system and container engine that the containers rely on is fully patched and up to date. Make sure all of the packages and libraries used in your container images are also patched and up to date. For additional guidance, you can use the Docker CIS Benchmark to harden your Docker environment. The Docker CIS Benchmark provides guidelines you can use as a checklist to make sure your Docker container environment is following defined security best practices.

One challenge in hardening a container ecosystem is how to conduct a vulnerability scan. Credentials scans are more comprehensive, but require something like SSH credentials within the container. Running the SSH daemon within a container, however, is highly frowned upon.

Common container practice dictates that each container run a single process, but SSH would become a second process within the container. That process requires some sort of process manager, which in turn increases the complexity of the container. Perhaps the biggest deterrent, though, is that it would require developers to bake the key and credentials for SSH authentication into the container, which hinders image portability due to container protection issues, and would require developers to locate and kill all containers with the current version of SSH code in the event of an SSH server code vulnerability.

Steps to Secure Your Containerized Applications

"When a new vulnerability is discovered in packages used in images you've deployed, that event needs to automatically flag that you have containers running that are based on that image and prompt you to update or patch the code."

— Amir Jerbi
CTO,
Aqua Security

While the container environment is challenging from a security perspective, it's a manageable challenge. A few security measures, implemented well and managed effectively, can help you secure and protect your container ecosystem.

1. Implement vulnerability management as part of your container development life cycle. This goes back to early detection. By implementing effective vulnerability management throughout the container development life cycle, you improve your odds that you can identify and resolve security concerns before they become a more serious issue.

Jerbi clarifies the importance of monitoring beyond initial development, though, and also being able to quickly identify which containers in your environment are affected by new vulnerabilities. "Say you have scanned and approved an image as being free of vulnerabilities. When a new vulnerability is discovered in packages used in images you've deployed, that event needs to automatically flag that you have containers running that are based on that image and prompt you to update or patch the code."

2. Scan for vulnerabilities before pushing images to the registry. As a final check once container development is complete, you should perform a vulnerability scan on containers before pushing the images to the registry.

3. Continue scanning in the registry. New vulnerabilities are discovered all the time, so scanning for and identifying vulnerabilities is a continuous process. You should continue to scan container images in the registry both to identify any flaws that were somehow missed during development and to address any newly discovered vulnerabilities that might exist in the code used in the container images.

4. Map image vulnerabilities to running containers. Identifying issues in the container registry is one thing, but you also need to have a means of mapping vulnerabilities identified in container images to running containers so security issues can be mitigated or resolved.

5. Ensure that only approved images are used in your environment. There is enough change and volatility in a container ecosystem without allowing unknown containers as well. You should prohibit the use of unapproved container images, and have tools and processes in place to monitor for and prevent the use of unapproved container images.

"Public repositories like Docker Hub contain thousands upon thousands of images for almost every kind of software imaginable. However, most of these images are the work of users who published an image and have not maintained or patched it," cautions Fine. "Docker Hub does, however, mark some images as "official" images. These images are provided and vetted by software makers who provide fresh updates to these images and multiple version of the image. Using official images can help mitigate risks rather than using unofficial images."

Monitor Container Resource Activity

6. Only permit the use of approved registries. An extension of that step is to also only permit the use of approved container registries. Requiring the use of approved container registries reduces your exposure to risk by limiting the potential for unknown vulnerabilities or security issues to be introduced.

7. Ensure the integrity of images throughout the lifecycle. Part of managing security throughout the container life cycle is to ensure the integrity of the container images in the registry and as they are altered or deployed into production. Image signing or fingerprinting can be used to provide a chain of custody that allows you to verify the integrity of the containers.

8. Enforce least privileges in runtime. This is a basic security best practice that applies equally in the world of containers. When a vulnerability is exploited it generally provides the attacker with access and privileges equal to those of the application or process that has been compromised. Ensuring that containers operate with the lowest privileges and access required to get the job done reduces your exposure to risk.

Log All Container Administrative User Access for Auditing

9. Reduce the container attack surface by removing unneeded privileges. Along those same lines, you can also minimize the potential attack surface by removing any unused or unnecessary processes or privileges from the container runtime.

10. Whitelist files and executables that the container is allowed to access or run. Reducing the number of variables or unknowns allows you to maintain a more stable, reliable environment. Limiting containers so they can only access or run pre-approved or whitelisted files and executables is a proven method of limiting exposure to risk.

11. Enforce network segmentation on running containers. Maintain network segmentation or segregation between running containers to protect containers in one segment from security risks in another segment. Maintaining network segmentation may also be necessary for using containers in industries required to meet compliance mandates.

12. Monitor container activity and user access. Just as with any IT environment, you should consistently monitor activity and user access to your container ecosystem to quickly identify any suspicious or malicious activity.

13. Monitor container resource activity. In addition, monitor your resource activity like files, network, and other resources accessed by your containers. Monitoring resource activity and consumption is useful both for performance monitoring and as a security measure.

14. Log all container administrative user access for auditing. Maintain an accurate audit trail of administrative access to your container ecosystem, container registry, and container images. These logs may be necessary for auditing purposes, and will be useful as forensic evidence in the wake of any security incident.

Wrapping Containers with Security

"What's really important about Hyper-V Containers is that rather than trying to close existing holes now we can implement a solution that is secure by default and already meets compliance requirements."

— Taylor Brown
Principal Lead Program Manager, Microsoft

It's an unfortunate reality that security is often an afterthought—and that reality applies to containers as well. Following all of the advice above is great if you're starting from scratch developing new containers or implementing a fresh container ecosystem. Many organizations, however, have to face the challenge of securing existing containers that were already developed without security in mind.

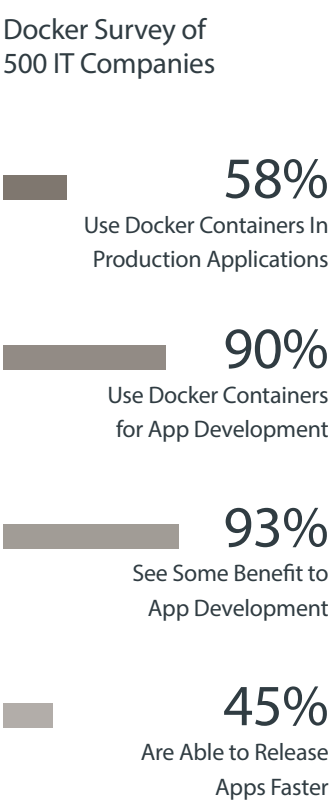
As security has taken center stage in the world of containers, a few solutions have emerged, like Microsoft Hyper-V containers. Hyper-V Containers are a sort of hybrid approach that take the lightweight footprint and modularity of containers and wrap them with the security of a virtual machine.

Taylor Brown, Principal Lead Program Manager at Microsoft, notes that Hyper-V Containers do not change anything about the development or deployment of containers. Developers can still create containers as they always have, or Hyper-V can be used to retroactively secure legacy containers that already exist. It is a decision that can be made at the time of deployment with the simple flip of a switch.

"What's really important about Hyper-V Containers is that rather than trying to close existing holes now we can implement a solution that is secure by default and already meets compliance requirements," says Brown.

Microsoft has developed a Docker virtual lab called Managing Docker on Windows to help you get started. The virtual lab walks you through setting up Docker in a Windows environment—from installing the necessary prerequisites to installing Docker. It also teaches you to create containers with and without Hyper-V isolation, and how to create layered container images from containers. If you're new to containers—this is an excellent place to start.

Use Containers with Confidence and Peace of Mind



Containers are here to stay, and will be a driving force for DevOps and application development for the foreseeable future. According to a Docker Survey of 500 IT companies, 58 percent of respondents reported using Docker containers in production applications. A full 90 percent are using Docker containers for app development. 93 percent report seeing some benefit to app development and 45 percent say they are able to release apps faster. On average, survey respondents are reporting a 13X increase in frequency of software releases. There are a number of contributing factors that containers provide that enable faster delivery.

In other words, containers are a permanent addition to the virtualization landscape. "Containers will open new opportunities for those who adopt them. They will reduce cost and the drag on both development organizations and ops infrastructure," says Randy Kilmon, VP of Engineering for Black Duck Software.

Containers provide a variety of unique benefits that simplify development and deployment of applications. They are an invaluable tool when used effectively, but part of using containers effectively is to understand the risks involved and take steps to ensure you're using containers securely.

Use the information in this paper and follow the steps outlined to secure your containerized applications so you can adopt containers with confidence and manage your container environment with some peace of mind.

ABOUT THE AUTHOR: Tony is a respected authority on technology. He has authored or co-authored a number of books, including Unified Communications for Dummies, Essential Computer Security, and PCI Compliance. Tony's work has appeared in PCWorld, CSO Online, Forbes, TechRepublic, and other print and online media sources.