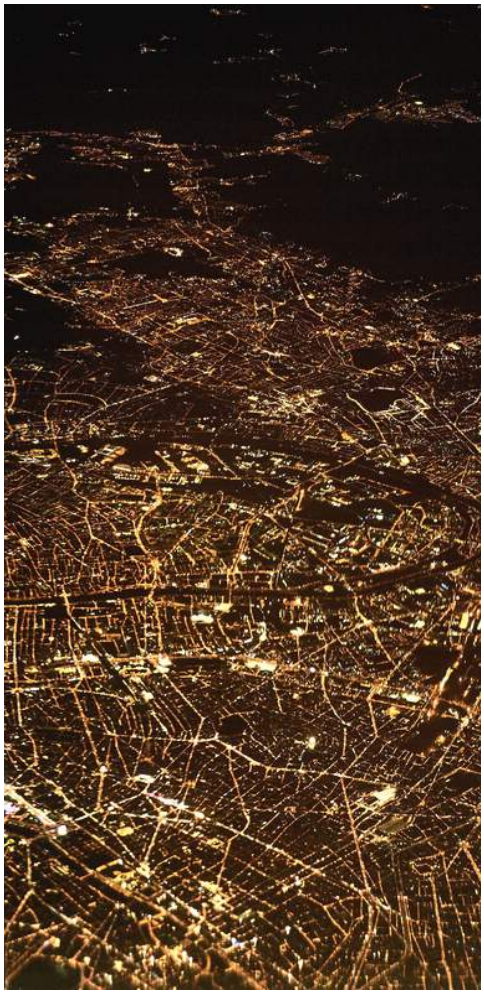
A SERIES ON KUBERNETES SECURITY

HOW TO DETECT RUNTIME THREATS IN KUBERNETES

INTRODUCTION



Kubernetes is one of the leaders in the container orchestration market. A recent survey by Cloud Native Computing Foundation (CNCF) suggests that 84% of companies are running Kubernetes containers in production. However, organizations running Kubernetes have also reported severe security threats in their respective container environment, with each threat linked to the container's lifecycle phase. These organizations often have to remediate vulnerabilities during the development phases and respond to threats during runtime to keep the impact to production and downtimes at a minimum.

RUNTIME THREATS

Organizations must be aware of runtime security threats at all times. Considering that these threats occur in real-time, it becomes incredibly challenging to deal with them. A compromised container cans another process, which can compromise other systems or containers. You can deal with such threats by regularly monitoring and conducting Kubernetes container threat detection activities. You should pay special attention to network traffic and restrict insecure and irrelevant communication, as determined by the network policies.

You can use what you've discovered through granular and robust monitoring to treat issues and stay in compliance with your Kubernetes security policies. But the work does not end here; you have to combine monitoring and visibility with your processes to facilitate lists that assist in identifying unforeseen issues and threats to the running container. Furthermore, you can also closely monitor new deployments to avoid the introduction of new vulnerabilities. Additionally, scanning container images for current security vulnerabilities also goes a long way in reducing the attack's surface.

WAYS TO DETECT RUNTIME SECURITY THREATS

To maximize Kubernetes' security levels against runtime threats, you should first be able to determine what these threats look like. Here's how you can do so:

LEVERAGE THE ABILITY OF THE NEW STACK

Although developers do not possess the required bandwidth to cater to security needs, they must include more data for their overall development process. The developers know the techniques that will run, binaries used, and the kind of interactions between containers. This data may be present in the DockerFile or a YAML file that explains how the system is orchestrated. This information should be automatically translated into the security profile to clear what the endpoint should and should not do. And, if anything apart from that happens, the system will be able to detect the same.

SET-UP A BASELINE

The term machine learning has gained a lot of traction lately. And this complex technology can help in dealing with Kubernetes security threats. In machine learning, you must have a certain baseline based on which the system learns, and the Kubernetes container can be an ideal candidate for this.

Identifying the baseline for how your application or container functions can help detect threats and vulnerabilities at runtime. You can set up your baselines by tracking configurations, resource utilization, network traffic, or process execution. You can then start detecting threats or potentially malicious activity much faster when you notice your container's behavior differing from its established baselines.

IDENTIFY AND REUSE THE SAFEST PIECES

When designing containers, start by ensuring that the images in use are from known and secure registries or from your own allow lists only.

You should also try and reuse images that have proven to be secure and made it through your vulnerability scans. Reusing your most secure images can help reduce attack surfaces. Since you're already aware that a bulk of the container you're designing contains a secure image, you'd only have to scan the few unknown pieces of your container.

VULNERABILITY SCANNING ON RUNNING DEPLOYMENTS

You should extend vulnerability scanning to running deployments in order to detect threats at runtime. The results from pre-deployment vulnerability scans can be used as a baseline to identify new and unknown vulnerabilities introduced in your containers. Scanning process activities for anomalies can also help detect threats that arise at runtime. Again, compare ongoing process activities with your process baselines to distinguish between known and unknown processes.

IMMUTABILITY

Leveraging the immutability of pods and containers can help nip runtime threats in the bud before they cause extensive damage to your deployment. When breached, immutability can help terminate breached pods and containers and replace them with a new pod. Looking out for such occurrences can help security teams identify when a threat occurs at runtime while keeping your deployment safe and available.

SKIP NOTHING, LOG EVERYTHING

Logs are still the single source of truth for everything that occurs within a computing system: all applications, infrastructure elements, and networking resources within a container record logs. Sending your logs to a log analytics platform can help you analyze and visualize what's going on within your container. By analyzing logs, you can gain insights into traffic patterns, visualize network activity, identify potential security threats, and pinpoint misconfigurations that can lead to potential vulnerabilities. Capable log analytics platforms can also help you set up alerts whenever it logs anything out of the ordinary or detects unwanted or unrecognized patterns in your container, helping you identify threats at runtime.

OBSERVE NETWORK TRAFFIC

Observing your active networking traffic and comparing it with what your Kubernetes network policies allow is another way to detect unnecessary or unexpected communication between your services. You can also use this practice to further strengthen your network policies by observing what communication your policies allow but isn't happening. This can help identify and plug loopholes within your network policies.

The difficulty of observing all network traffic across all your services can vary based on the size and complexity of your deployments. In some cases, it might make sense to depend on commercial Kubernetes security solutions that provide a holistic view of your network traffic while alerting you whenever an anomaly is detected.

KUBERNETES CONTAINER THREAT DETECTION SOLUTIONS

To completely secure your Kubernetes deployments and containers from runtime security threats, you must focus your security efforts on the development and deployment phases of your container. Doing so can significantly reduce the chances of security incidents that can bring down or adversely affect your containers.

From the best practices listed above, it's clear that you can exercise greater security for your containers by following simple practices and through effective logging and monitoring. To summarize, here are some recommendations to secure Kubernetes at runtime:

- Leverage information from the build and deploy phases to assess various routine activities during runtime to establish baselines. Use these baselines to compare and identify suspicious activity.
- Run vulnerability scans before deployment and during runtime in order to identify existing and new vulnerabilities.
- Observe network traffic to help differentiate between known and unknown communication.
- Log everything you can; ship your logs to a highly capable log analytics platform to analyze, visualize, and recognize threats and anomalies as they occur.
- Leverage the immutability of pods to limit the extent of a breach by quickly replacing compromised pods with new ones.
- Reuse trusted and time-tested images while building new containers.