PREVENT CRYPTOJACKING BY SECURING KUBERNETES

A disturbing new trend is on the rise in public cloud security breaches. Attackers are not just stealing sensitive data, but now they're also hijacking compute power in insecurely configured Kubernetes clusters to mine for cryptocurrency. Tesla, Aviva, and other companies had their Kubernetes clusters on AWS used for cryptocurrency mining, in addition to potential sensitive data leakage. Cryptojacking is clearly becoming a new and emerging threat.

We ask key questions on why this threat is becoming so widespread with Kubernetes container clusters on AWS and present three steps to prevent it.

Why is Kubernetes a platform of choice for cryptojacking?

Container technologies, such as Docker and Kubernetes, have been phenomenal in improving developer productivity. With lightweight portable containers, packaging and running application code is effortless. However, while developers and applications can benefit from them, many organizations have knowledge and governance gaps, which can create security gaps.

How widespread is the footprint of Kubernetes on AWS?

According to a recent study, 63% of Kubernetes stacks run on AWS. The widespread usage of Kubernetes clusters on AWS, coupled with their management complexity and insecure configurations, can leave the door open for attackers to use it for cryptocurrency mining.

What is the potential impact?

The obvious impact can come in the form of higher public cloud bills. Even worse, these gaps can lead to a multi-stage attack where a Kubernetes breach can also lead to compromising sensitive keys, data, and machines beyond the cluster itself. This lateral movement can be a big concern in enterprises where hundreds and even thousands of containers are being provisioned every week.

Three Steps to Secure Kubernetes Clusters

We present 3 ways an enterprise can secure their Kubernetes clusters

1. Blindspot detection – find all Kubernetes clusters
2. Assessing security and hardening the Kubernetes clusters
3. Continuous monitoring, automation and remediation

With these three steps, cryptojacking threats can be completely avoided. Let's dive into each of these three steps.

I. Blindspot Detection - Find All Your Kubernetes Clusters

One of the first challenges enterprises will face is to detect Kubernetes clusters running in AWS. Shadow IT is very much alive today, and even if IT knows about the EC2 servers that have been provisioned, there's a good chance they don't know all of the software installed on those servers. To secure their Kubernetes clusters, step number one is find out where they exist. Discovery tools, such as BMC Discovery, can quickly discover Kubernetes clusters in AWS as shown in the figure below.

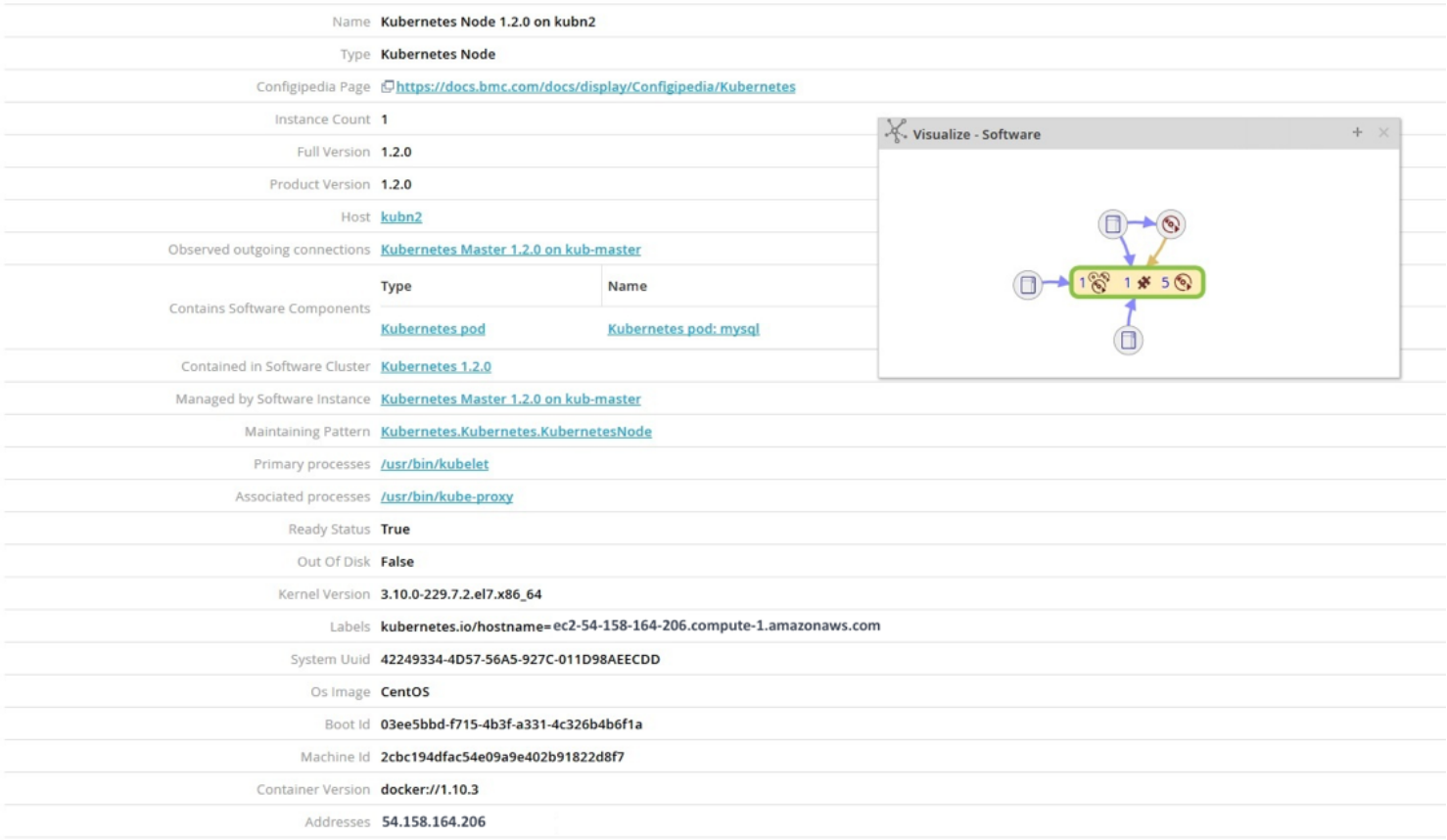

Figure 1. BMC Discovery can find Kubernetes cluster blindspots

II. Assess and Harden your Container Stack

After Kubernetes clusters are discovered, enterprises must define security policies to enforce that these clusters and their workloads are properly secured. As seen in Figure 2, there are many layers in a container stack, and all of these layers need to be hardened. Hardening of this stack is especially important for Kubernetes, since many of the Kubernetes installers default to developer-friendly, but insecure configurations. For example, there are the well-known kops security issues.

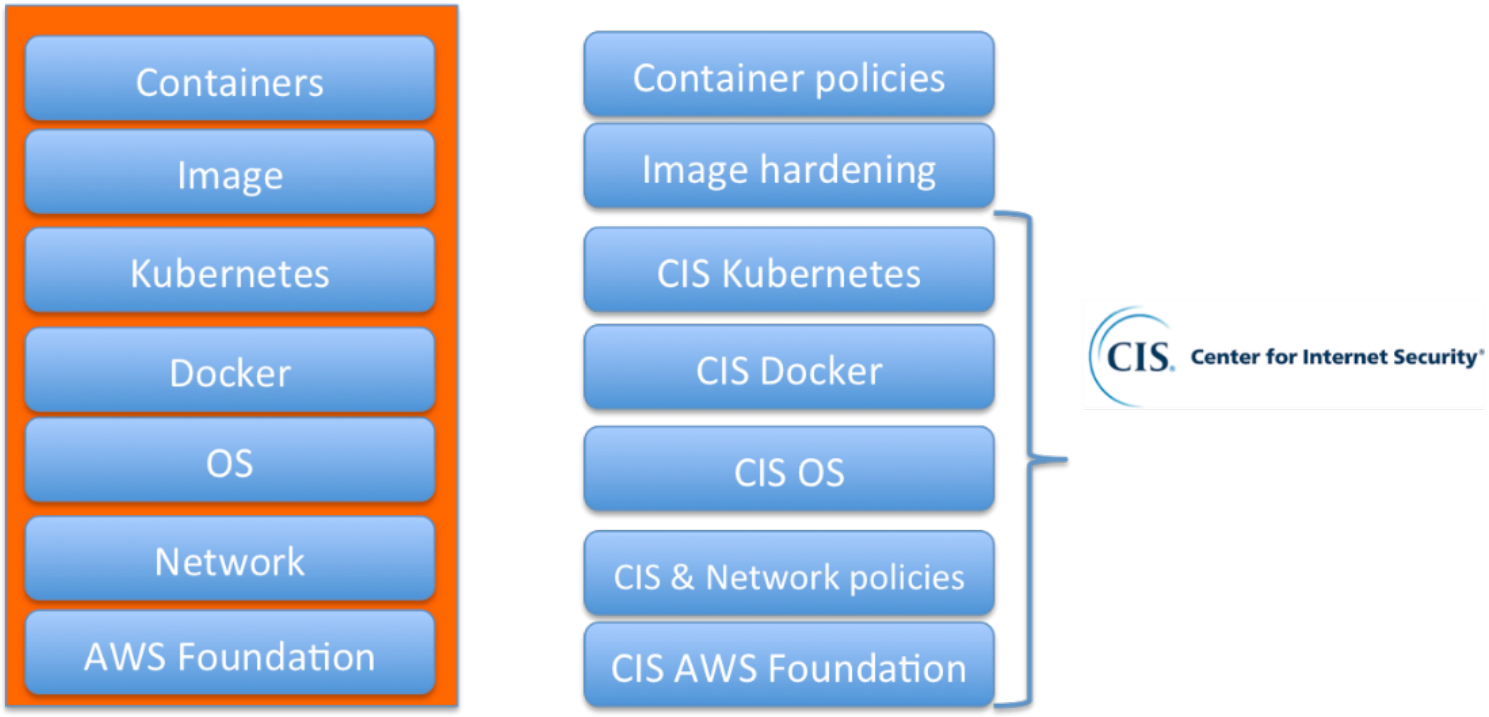

Figure 2. Defense in depth for container stack hardening

The lowest layers in the stack are the AWS foundation and network segmentation. In these layers, secure configurations should limit the blast radius by ensuring isolation of workloads. They should also implement standard public cloud security hygiene such as logging, monitoring, security groups, and IAM. The CIS AWS Foundation and CIS AWS Three-tier Web Architecture benchmarks define a solid set of controls to secure these layers. For the next three layers (OS, Docker runtime, and Kubernetes clusters), each of these layers have hundreds of configurations that also need to be secured. The CIS Operating System, Docker, and Kubernetes benchmarks are great resources for determining the specific configurations that need to be secure. If you would like help examining these configurations at scale, BMC's SecOps Policy Service can evaluate and harden all of those layers against their applicable CIS policies. A live report showing compliance violations of a Kubernetes master is shown in Figure 3 below.

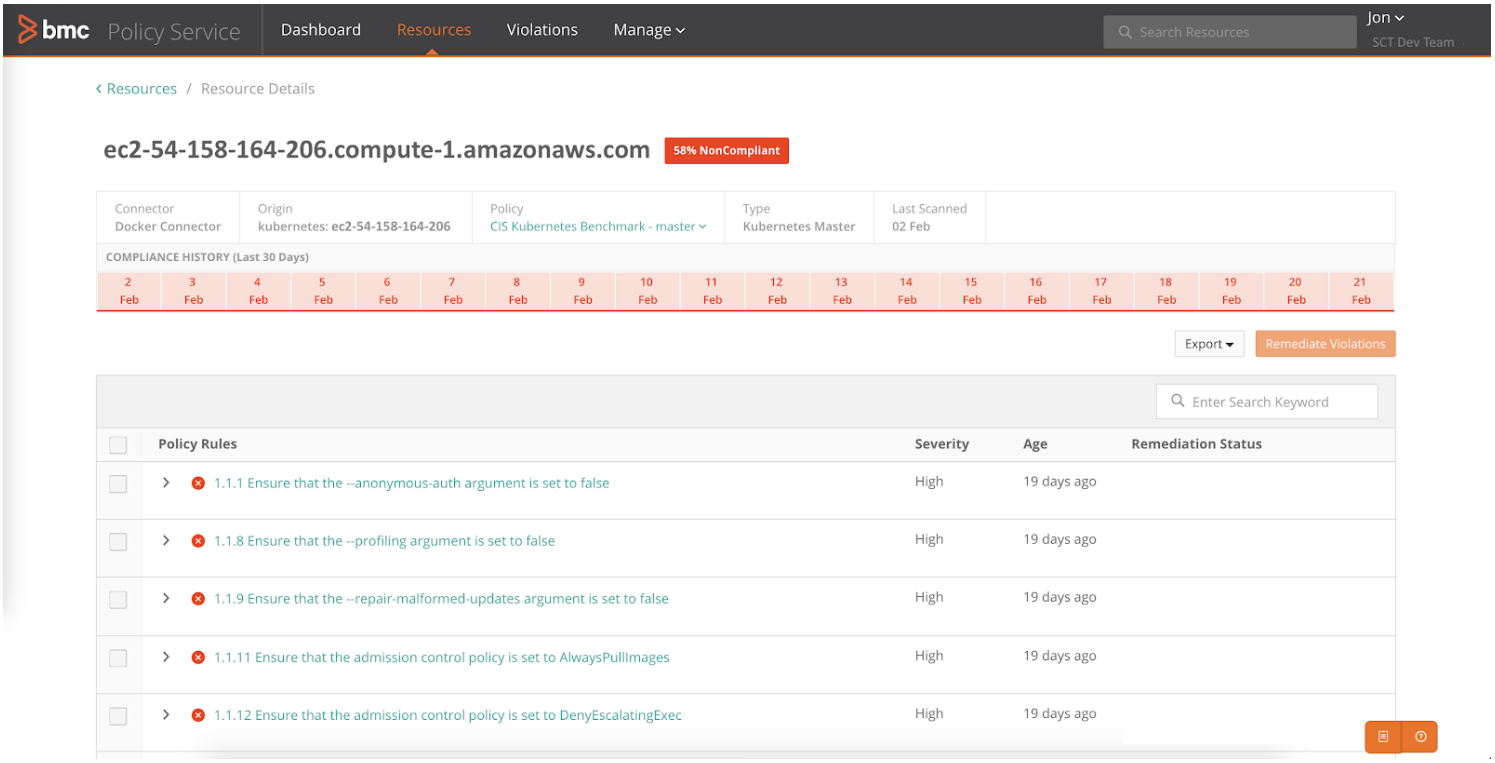

Figure 3. CIS Kubernetes on API server hardening with SecOps Policy Service

For example, section 1.1 of the Kubernetes CIS policy details securing the API server that Kubernetes runs, perhaps the most critical and central component through which all requests are made. There are many important security rules here. Other examples include: not allowing anonymous requests (1.1.1), not using the insecure defaults such as basic authentication or insecure tokens (1.1.2, 1.1.3), ensuring the use of HTTPS (1.1.4), etc. There are hundreds of more rules in securing both master and worker nodes, and the out-of-the-box CIS policies available in SecOps Policy Service make it fast and easy to assess for non-compliance.

III. Automation

Finally, enterprises should implement a policy automation solution, such as SecOps Policy Service, to continuously monitor, assess, and remediate container stacks. These tools can programmatically enforce the "security-as-code" policies. Policy automation addresses the automation, scale, and agility challenges to keep everything secure, since container stack is highly dynamic and changing frequently. Manually keeping track of all security is not humanly possible, hence the need to automate all these Kubernetes checks.

The Final Word

Container technology on AWS is being adopted at lightning speed, and we still firmly believe that neither public cloud nor Kubernetes is inherently insecure. However, the complexity of managing these container stacks with a strong security posture is increasingly challenging without a standard security policy (e.g., CIS Kubernetes, Docker, OS) and automation. All the cryptojacking breaches at Tesla, Aviva, and elsewhere could have been prevented. So we hope that the information in this article will help you prevent cryptojacking in your organization.