INSIGHTS 16 min read

Share

Twitter

LinkedIn

by Naor Shmuel Published on 08/27/2020 Last updated on 06/18/2024

Protecting your Kubernetes platform from multiple attack vectors with the K8SHIELD™ Framework

As many know, the MITRE ATT&CK framework provides a threat matrix that guides administrators, developers, DevOps, security teams, and others in protecting their networks, systems, and endpoints from undesirable access and manipulation. However, this framework currently supports attacks primarily related to these platforms: Linux, Windows, and macOS-based attacks, as well as general cloud (Infrastructure as a Service (IaaS), for example) environments.

The MITRE ATT&CK has offered us tools that are fundamental in defending ourselves by giving us access to critical information about how these attacks might be performed, and how we can protect against them. But what about Kubernetes?

What we are missing is a MITRE ATT&CK matrix that is interpreted for the Kubernetes environment, a matrix that connects the dots and provides the missing security context for Kubernetes security best practices. We need a matrix that can elaborate on the different security checks and exploit information, including descriptions of the techniques used to manipulate these security pitfalls in the Kubernetes context. Visualizing the relationship between these, and how they connect to your specific deployment is vital in building the best defense strategy possible.

Understanding the MITRE ATT&CK Framework

Microsoft recently took this one step further by providing us with some guidelines for developing a parallel threat matrix. With the Cisco Cloud Native Security Solution, we've brought this matrix to life. We've taken the concepts presented by Microsoft and the theory of a threat-based model from MITRE and implemented a matrix that is tailored for Kubernetes, helping our users actively detect potential threats in their Kuberetes clusters but also to create, implement and monitor their defense strategies and the security of their applications and deployments. After all, what is MITRE Att&ck if not a solid opportunity for you to enhance your cloud native security?

With our K8SHIELD™ Framework, we've also released a graphical view that connects the dots for you with the familiar ATT&CK matrix, displaying the risks and their applicability to deployed clusters. The matrix actively shows you how you're affected by the known attack vectors and where you can strengthen your defenses enabling you to take relevant action in real time. With this matrix fully integrated into our solution, and our end-to-end holistic approach to security, handling Kubernetes has never been easier.

A cloud-native holistic approach to security with the MITRE ATT&CK Framework

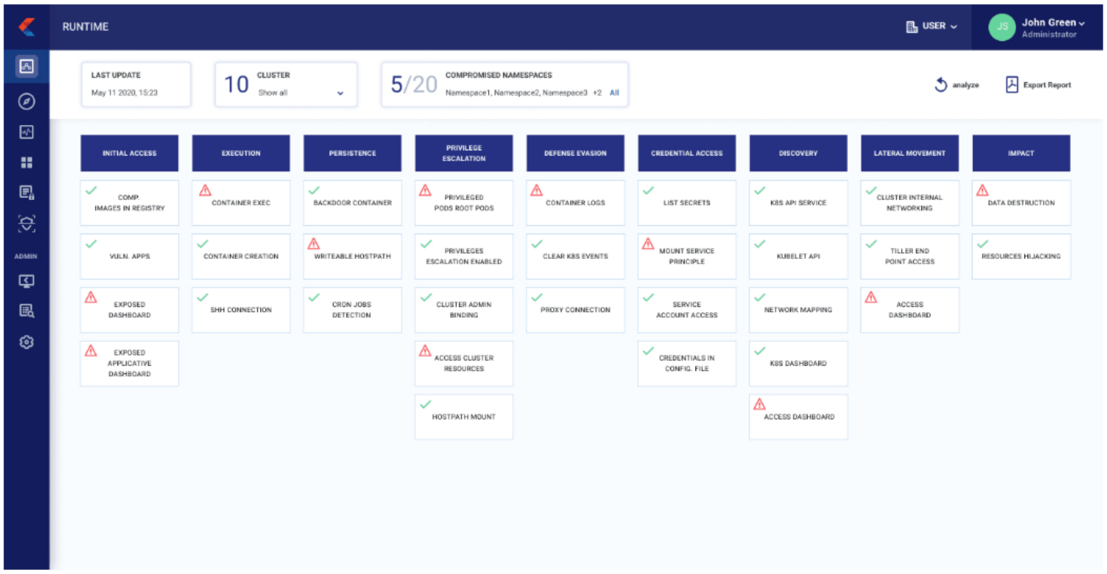
A cloud-native holistic approach to security with the MITRE ATT&CK Framework

As more and more networks are moving in their entirety to the cloud, so is Kubernetes being adopted faster and faster. The Cloud Native Computing Foundation (CNCF) recently reported that 1.7 million developers use Kubernetes!

Given the complexity of Kubernetes, however, and the new threat matrix, guidance is needed for developers, DevOps, CISOs, and more in collaborating and creating safer applications for their cloud-native deployments.

Cisco Cloud Native Security Solution offers a comprehensive knowledge base of curated information that provides a clear picture of your security posture. Our framework and matrix, modeled after the MITRE ATT&CK® Framework, but designed specifically for the Kubernetes ecosystem, allows for the identification of attack patterns specifically within Kubernetes clusters, including pattern analysis, remediation suggestions, and detailed reports – throughout the entire lifecycle, starting with continuous integration and through to production.

While there are similarities between the tactics, overall the tactics and the techniques to access a K8s network differ from those used to access traditional networks – often greatly. Based on that assumption, and after significant research, we created this Kubernetes-specific matrix:



LAST UPDATE
May 11 2020, 15:23

10 CLUSTER
Show all

5/20 COMPROMISED NAMESPACES
Namespace1, Namespace2, Namespace3 +2 All

INITIAL ACCESS	EXECUTION	PERSISTENCE	PRIVILEGE ESCALATION	DEFENSE EVASION	CREDENTIAL ACCESS	DISCOVERY	LATERAL MOVEMENT	IMPACT
✓ COMP. IMAGES IN REGISTRY	⚠ CONTAINER EXEC	✓ BACKDOOR CONTAINER	⚠ PRIVILEGED PODS ROOT PODS	⚠ CONTAINER LOGS	✓ LIST SECRETS	✓ K8S API SERVICE	✓ CLUSTER INTERNAL NETWORKING	⚠ DATA DESTRUCTION
✓ VULN. APPS	✓ CONTAINER CREATION	⚠ WRITEABLE HOSTPATH	✓ PRIVILEGES ESCALATION ENABLED	✓ CLEAR K8S EVENTS	⚠ MOUNT SERVICE PRINCIPLE	✓ KUBELET API	✓ TILLER END POINT ACCESS	✓ RESOURCES HIJACKING
⚠ EXPOSED DASHBOARD	✓ SHH CONNECTION	✓ CRON JOBS DETECTION	✓ CLUSTER ADMIN BINDING	✓ PROXY CONNECTION	✓ SERVICE ACCOUNT ACCESS	✓ NETWORK MAPPING	⚠ ACCESS DASHBOARD	
⚠ EXPOSED APPLICATIVE DASHBOARD			⚠ ACCESS CLUSTER RESOURCES		✓ CREDENTIALS IN CONFIG. FILE	✓ K8S DASHBOARD		
			✓ HOSTPATH MOUNT			⚠ ACCESS DASHBOARD		

Leveraging the new K8s matrix to plan and monitor your deployment

Kubernetes is fast becoming the industry standard in cloud-native container orchestration, but simultaneously it's also super complex and not everyone really understands it well. On top of that, managing its security with rapid deployments and constant changes involves many different roles collaborating together, which has its obvious advantages, but can make things even more complicated. This is especially true when dealing with technology that at its core is so advanced. And, despite the shift-left movement, it turns out that not all developers are even entirely aware that this is what they're using! This all makes it hard to manage security for Kubernetes.

Understanding Kubernetes security starts with understanding how Kubernetes can be breached. Our new K8s matrix brings a complete picture to every role involved in protecting the security of your organization's Kubernetes clusters like that of the original MITRE ATT&CK® matrix. By drawing a hypothetical parallel between the traditional operating system and the Kubernetes cluster, we offer a model that is immediately digestible. Furthermore, by drawing these similarities, the new model, like the already familiar more traditional model, also provides the technical detail of how attacks are performed, alongside an explanation as to what attackers aim to achieve by performing said technical steps.

Cisco Cloud Native Security Solution offers a uniquely comprehensive solution, with the synergy offered by our comprehensive threat-based matrix that provides the context necessary to enable understanding and take correct action and our risk analysis tool that predicts threat potential, allowing DevSecOps teams to prioritize their tasks based on the actual risk level. This combination is fundamental in aiding collaboration between the different technical roles, each of which naturally has a different perspective and therefore needs different details in order to take action. Accordingly, the model accounts for:

- **Tactics** – the attack vector; the ultimate objective of an attacker
- **Techniques and sub-techniques** – the methods used to achieve said objectives
- **Documented attacks** describing how adversaries achieved these tactics by using the associated techniques
- **Recommendations for remediation**

Let's drill down into each of the tactics to see what it's really all about.

Initial access

Initial access is the first vector point an attacker attempts to bypass – accessing the container undetected. In Kubernetes, undetected access can be achieved by exploiting:

Compromised images in the registry

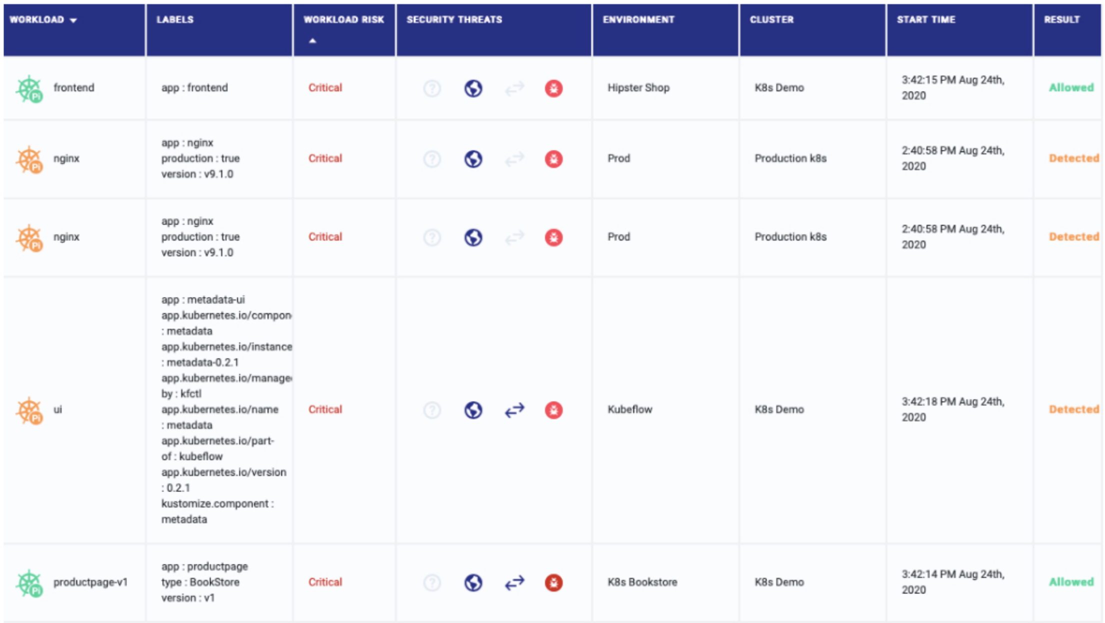
A compromised image is an image that was modified/changed after it was uploaded to the images registry. Attackers who manage to access images registries can modify images to include vulnerable elements that can be exploited later when the image is used to build a container. Compromised images don't pass through the CI tests/checks process, but given their location in the images registry they can be perceived as trusted images.A prerequisite step to avoid compromised images is to restrict users to deploy containers from unauthorized images registries. Restricted registries usage can minimize the potential of compromised images usage. To mitigate the risk, users need to whitelist images which pass through the CI pipeline. When images deviate from their baseline they are perceived as compromised images. Following are the deviation rules:

- Before images are pulled, validating rules defined by users in a registry policy, whitelisting specified registries only
- Ensuring only the images pushed to the registry through dedicated CI plugin are allowed based on a specified hash, labelling all other images to be blocked
- During runtime validate the image doesn't change based on Kubernetes procedures (e.g. init containers)
- Images are scanned monitored on an ongoing basis

Vulnerable applications

Vulnerable applications can be easily exploited with techniques such as remote code execution (RCE) by hacking simple network misconfigurations for services such as the load balancer, external name, or the external IP address. In fact, RCE is so common that it has recently become a top priority of the Kubernetes.io bug bounty program this year. To defend against such attacks, based on your configurations, all communication to and from any identified vulnerable pods can be completely blocked throughout its entire lifecycle.

Cisco Cloud Native Security Solution scans your deployment to detect public-facing applications providing details about any applications that may be at risk due to existing vulnerabilities based on their severity levels, recommending external communication be blocked.

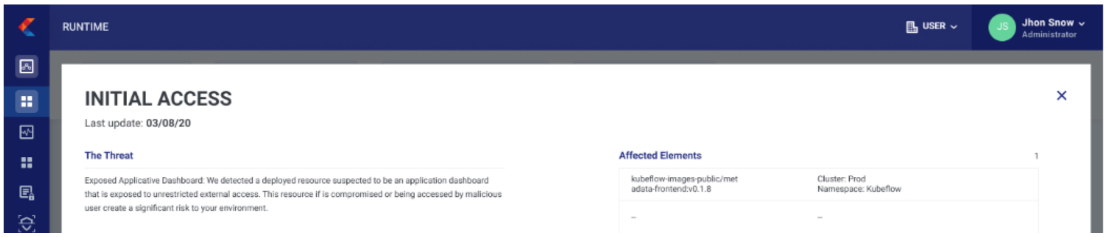


WORKLOAD ▾	LABELS	WORKLOAD RISK ▴	SECURITY THREATS	ENVIRONMENT	CLUSTER	START TIME	RESULT
frontend	app : frontend	Critical	? 🌐 ⇄ ●	Hipster Shop	K8s Demo	3:42:15 PM Aug 24th, 2020	Allowed
nginx	app : nginx production : true version : v9.1.0	Critical	? 🌐 ⇄ ●	Prod	Production k8s	2:40:58 PM Aug 24th, 2020	Detected
nginx	app : nginx production : true version : v9.1.0	Critical	? 🌐 ⇄ ●	Prod	Production k8s	2:40:58 PM Aug 24th, 2020	Detected
ui	app : metadata-ui app.kubernetes.io/compon : metadata app.kubernetes.io/instance : metadata-0.2.1 app.kubernetes.io/manage by : kfctl app.kubernetes.io/name : metadata app.kubernetes.io/part-of : kubeflow app.kubernetes.io/version : 0.2.1 kustomize.component : metadata	Critical	? 🌐 ⇄ ●	Kubeflow	K8s Demo	3:42:18 PM Aug 24th, 2020	Detected
productpage-v1	app : productpage type : BookStore version : v1	Critical	? 🌐 ⇄ ●	K8s Bookstore	K8s Demo	3:42:14 PM Aug 24th, 2020	Allowed

Exposed dashboards

As a major interface used to manage the entire Kubernetes deployment, anything that can be done using kubectl can be done using the dashboard – and this makes the dashboard a highly sensitive point in your network. Known vulnerabilities can lead to privilege escalation and arbitrary code execution.

If the dashboard is exposed, it can be manipulated by remote cluster management. To avoid this, it's vital to install updated versions of the dashboard (v1.10.1 and greater), while protecting its services with proper configuration (for example, the LoadBalancer or NodePort), and users should be allowed access only when they have a service account token or a kube config file. Cisco Cloud Native Security Solution detects exposed Kubernetes dashboards and applicative dashboards deployed in Kubernetes clusters and alerts with recommended remediation steps.



INITIAL ACCESS

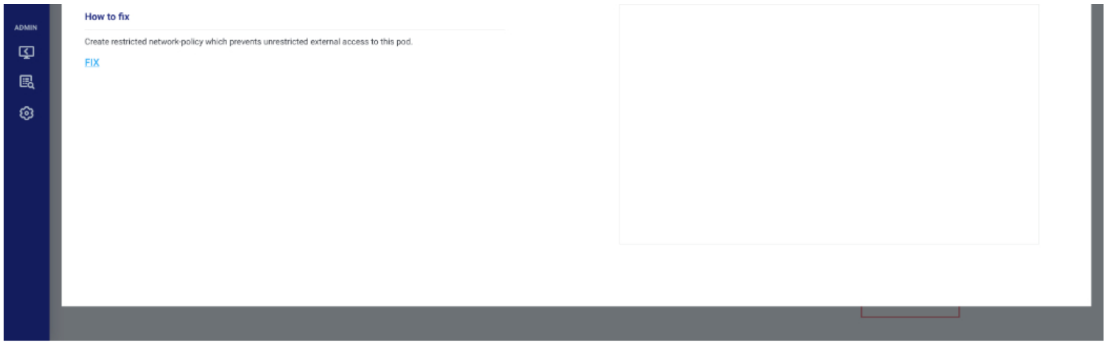
Last update: **03/08/20**

The Threat

Exposed Applicative Dashboard: We detected a deployed resource suspected to be an application dashboard that is exposed to unrestricted external access. This resource if is compromised or being accessed by malicious user create a significant risk to your environment.

Affected Elements 1

kubeflow-images-public/metadata-frontend:v0.1.8	Cluster: Prod Namespace: Kubeflow
–	–



How to fix

Create restricted network-policy which prevents unrestricted external access to this pod.

Execution

After accessing the network, the attacker manipulates open doors to execute malicious code. In Kubernetes, malicious code can be executed by exploiting container creation and execution and SSH events.

Container execution

Once an attacker finds a way to gain permissions for execution, they will be able to execute any commands they want from within the containers. Cisco Cloud Native Security Solution has always monitored your deployments for unauthorized container creation and execution, and now also enables granular policies to control execution of specific target commands such as patch requests, interactive shells, reverse shell, bash etc. and interactive input by blocking stdin in execution. Attackers with permissions to deploy a pod or a controller in the cluster (DaemonSet, ReplicaSet and Deployment for example) can create new resources through which to then execute malicious code. In fact, CVE-2019-1002100 shows just how dangerous patch request permissions can be, for example. Hardened API rules and deployment policies can protect you against this. Cisco Cloud Native Security Solution identifies all of the users in your network with an RBAC that enables escalated privileges so that you can audit these details on a regular basis and limit each and every user to the principle of least privilege. Additionally, users can white list cluster events, thereby preventing the execution of all other unrecognized events.

SSH connections

By default, port 22 is set for SSH servers and root login is allowed, meaning anyone can connect to port 22 using the root user, particularly if password authentication is enabled and the root password is weak. SSH connections to a K8s deployment are useful for remotely connecting to clusters, but if misconfigured, can be used to illegally access and attack the container. To protect against this, our solution detects open SSH ports on pods, and alerts or blocks these pods based on your predefined policy.

Persistence

Persistence is the tactic of maintaining a foothold within the object under attack. In Kubernetes, undetected persistence can be achieved as the result of these techniques:

Writable hostPath and hostPath mount

hostPaths are useful for securing storage space so that new containers can be mounted. Writable hostPaths however can be used to gain persistence by attackers by loading their own malicious applications, for example. hostPath mount can be used to achieve privilege escalation (further described below). For this reason, Kubernetes offers a pod security policy, including a rule to whitelist hostPaths. If this is not configured, however, all hostPaths are automatically allowed. Cisco Cloud Native Security Solution analyzes your pod policies in runtime and guides you in correcting and hardening your configuration in order to ensure you're limiting privileges for hostPath.

CronJob

CronJob is used to schedule jobs and as such, if accessed by an attacker, can be used to schedule execution of malicious code and perform cryptomining manipulations. With proper RBAC (role-based access control) policies that tightly control deployed pods and CronJobs, this can be prevented. It's important to check and recheck such policies starting with the CI and on an ongoing basis as part of deployment and runtime. Cisco Cloud Native Security Solution regularly scans your configurations, identifies problematic roles, ensures CronJobs can only be created within clusters by authorized users and recommends creation of a cluster events policy.

Privilege escalation

Once access is obtained, attackers seek ways to increase privileges further. In Kubernetes, privilege escalation can be achieved by exploiting:

- Privileged and root pods
- Privilege escalation enabled
- Cluster administration binding
- Access to cluster resources

- Hostpath mount

With improper defenses, an attacker can escalate privileges by spinning a new container, creating cluster bindings, or by way of hostPath mount, thereby being able to also retrieve cloud resource data with a simple HTTP call from a workload to the cloud metadata. One such example of a known privilege escalation vulnerability is CVE-2018-1002105, proving the necessity for handling permissions with care.

Cisco Cloud Native Security Solution monitors the pods privileges by scanning their security context before they're deployed, and alerting users on potential risk due to misconfigured privileges. When deployed into the cluster, our unique Pod-Security Policies (PSP) are used to enforce correct privileges settings per container with the ability to use predefined profiles and granular policies (per pod as opposed to ServiceAccount in Kubernetes)..

Defense evasion

Just as it sounds, this tactic is the act of an attacker avoiding detection. To do this, the attacker attempts to hide their tracks and "stay under the radar". In Kubernetes, evasion can be achieved by exploiting:

- Container logs
- Clear K8 events
- Proxy connection: Just as many other configurations are included in the pod security and API policies, so can you protect your container and K8 event logs by properly preventing hostPath mount, blocking code execution, limiting permissions carefully for deletion of logs and events, and protecting your pods with hardened policies overall. Similarly, just as attackers attempt to access clusters by masking their IP addresses, so can Cisco Cloud Native Security Solution assist and guide you in implementing an IP whitelist to block all communication from unknown IP addresses.

Credential access

Credential access is when the attacker attempts to gain access to and steal passwords and other sensitive secrets and credentials. In Kubernetes, credentials can be stolen by exploiting:

- List secrets
- Mount service principle
- Service account access
- Credentials in the configuration file: These are easily manipulated vectors if secrets or application credentials are misstored in configuration files or if secrets and credentials are mishandled in other critical ways, such as by storing tokens improperly in the pod and then allowing unrestricted access to that pod. The API policies you implement should enforce read-only operations, read-only access to configuration files, block hostPath mounting, block the creation of webhooks that hook secrets and block mutating webhook access. Predefined API calls should be secured to restrict the potential of malicious activities on cluster resources that are not worker nodes/application pods. Additionally, dynamic backend logging should be implemented to properly track and monitor all events on the network. We recommend correct configurations and hardened policies, detects exposed secrets, and monitors for suspicious behavior during runtime to detect related attacks.

Discovery

Discovery is the set of actions the attacker takes in order to observe and learn about the network. In Kubernetes, discovery can be achieved by exploiting:

- K8 API services
- Kubelet API
- Network mapping
- Access K8 dashboard: All connected to the Kubernetes API service, direct access to your nodes and clusters, and the details of how the network is deployed overall, these vectors provide attackers with the tools they need to perform lateral movement and execute malicious activity. Cisco Cloud Native Security Solution recommends blocking connections between the cluster and the API server, ensuring it's only accessible from trusted subnets or even only from the virtual private cloud by configuring the necessary firewall rules. The solution discovers network traffic and reveals all network communications both in-cluster and between clusters across your complex environment. We also encrypt internal communication between services in the same cluster and maintain the security level for communication between services in different clusters, enforcing authorization policies on multi-cluster service communications.

Additionally, Cisco Cloud Native Security Solution verifies increased protection of the kubelet, and ensures a hardened granular API policy to block webhooks, and ensures compliance with CIS benchmarks. On top of this, we monitor your audit logs and alerts on detection of suspicious and unauthorized activity.

Lateral movements within Kubernetes

With lateral movements, adversaries continue to learn about the network, identify roadblocks that may present a problem for them, and also gain access to additional parts of the network. In Kubernetes, lateral movements can be achieved by exploiting:

gain access to additional parts of the network. In Kubernetes, lateral movements can be achieved by exploiting:

- Cluster internal networking
- Tiller endpoint access
- Access dashboard: It's important to always act according to the rule that if an attacker has gained access to a single container, they can use that as a foothold to gain broader network access, and to protect the multiple layers of your network accordingly. Typically, the additional access would then be gained by way of unprotected kubectl execution commands, and compromised or malicious images.

By observing best practices in handling and defending against initial access, privilege escalation, credential access, persistence, and discovery, you've already taken the right steps toward defense against lateral movement. Additionally, however, the tiller endpoint – part of your helm installation up to and including version 2 – must also be secured to provide the best defense. By default, it receives cluster admin permissions, and exposes the gRPC port from within the cluster without authentication. This enables attackers easy access without authentication as well. By protecting your tiller endpoint with sidecar injection, you can then block incoming communications as your method of defense. We monitor your tiller endpoint, recommending hardened security based on best practices, and also monitor port-forwarding, which is sometimes used in place of the –host option in the Helm CLI with this installation.

Impact

With impact, the attacker destroys or steals information from the network – achieving the ultimate objective of the hack. In Kubernetes, impact can be achieved by destroying data or, alternatively, hijacking resources. The cryptojacking worm Graboid reported in October last year is a great example of the end game combined with other vectors as well. By adhering to the recommendations Cisco Cloud Native Security Solution makes and enabling the proper policies to be implemented based on recommendations and best practices in your defense against all of the other vectors, you protect yourself from this vector as well.

Start as early as possible

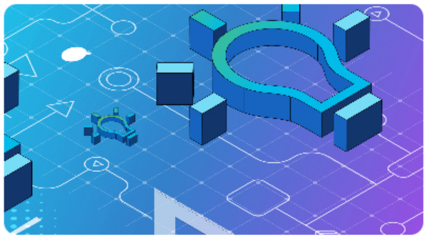
Cisco Cloud Native Security Solution delivers a Kubernetes-Native security platform including DevOps tools, CI/CD pipelines, and "security as code" that are robust, scalable, and portable. With this unique implementation, users work according to prioritized alerts combined with contextual explanations, guidance, recommended solutions and automation. This holistic view of the Kubernetes cluster risk level and dedicated mitigation suggestions for each attack pattern makes Kubernetes security clear for developers, DevOps and security teams alike, and as early as possible.

With this complete solution, you can collect, monitor, and visualize high-granularity metrics from all your containers at runtime. Directly from the Cisco Cloud Native Security Solution UI, analyze your current defense line, and drill down to get relevant information vital to properly managing your strategy: known vulnerabilities and exploits, risk assessment and prioritization, automatic fixes, and best practice advice and guidance.

To learn more about what Kubernetes security might look like in the world of cloud native security, visit our post on Kubernetes and multi-cloud security.

Driving Productivity and Improved Outcomes With Generative AI-powered Assistants
Mike Leone
PDF

Related articles



INSIGHTS

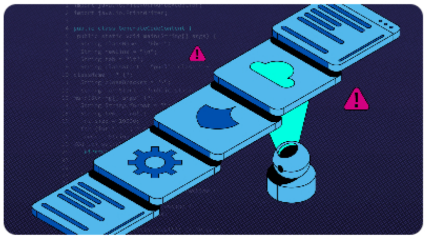
Platform engineering for cloud-native applications developments

CLOUD NATIVE KUBERNETES

PRODUCT

KubeClarity's cloud security tools: Architecture deep dive

CLOUD NATIVE KUBERNETES OPEN SOURCE OPENCLARIT



INSIGHTS

Top 15 software supply chain attacks: Case studies

CLOUD NATIVE KUBERNETES

-ER NEWSLETTER NEWSLETTER N

Subscribe to The Shift!

Get emerging insights on innovative technology straight to your inbox.

The Shift is Outshift's exclusive newsletter.
The latest news and updates on generative AI, quantum computing, and other groundbreaking innovations shaping the future of technology.

First name*

Last name*

Email*

Company name*

Job title*

Topics of interest*

□ Generative AI

□ Quantum computing

By submitting this form, you agree that Cisco may process your personal information as described in its Online Privacy Statement. Cisco may contact you with offers, promotions, and the latest news regarding its products and services. You can unsubscribe at any time.

☑ Yes, I agree

Subscribe now!

INITIATIVES

Programs

- Artificial Intelligence
- Quantum Labs
- Research
- Open Source

Events

- Upcoming & Recent Events

PORTFOLIO

Research

- GenAI
- Quantum

ABOUT US

Company

- About Us
- Our Team
- The Shift

Careers

- Job Openings

Connect

- Contact Us
- YouTube
- LinkedIn
- X

BLOG

Showcase

- GenAI
- The Breakdown
- Quantum Computing
- M2

Blog Categories

- Insights
- Inside Outshift
- Collaborations
- Product

Explore Cisco

Help Terms & Conditions Statement Cookies Trademarks