



KUBERNETES • 3 MIN

Kubernetes webhook used by

Continuer sans accepter

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

Paramètres du cookies Tout accepter

Accueil > Blog > Kubernetes > Kubernetes webhook used by attackers

14 December 2023

For an attacker, the main step after compromising a system is to establish persistent access to it. It means that even if you remove his initial access, he could easily come back thanks to the multiple backdoors he installed in your system. Some hackers are even specialized in selling backdoors on the darknet. This threat is also present in your **Kubernetes** clusters.

SOMMAIRE

- Microsoft Kubernetes Threat Matrice
- What are Admission Controllers?
- Using admission webhook to implement a backdoor
- Protect your Kubernetes cluster

Vous avez un projet Cloud ?

[Contactez-nous →](#)

Partager → [in](#) [f](#) [t](#)

Conclusion

Microsoft Kubernetes Threat Matrice

Microsoft released a **threat matrix** for Kubernetes based on the MITRE ATT&CK framework. There are multiple ways to establish persistence in a **Kubernetes cluster** but in this article, we will deep dive into the technique involving malicious admission controllers.

What are Admission Controllers

Kubernetes Admission Controller

Admission controllers intercept Kubernetes API requests and can modify or reject them before any operation occurs. The admission controller operation occurs before the persistence operation. In this article, we will focus on webhooks. A webhook receives the kube API request and can modify or reject it. There are two types of webhooks :

- The **mutating webhook** is triggered after the authentication step. Its role is to intercept the request, modify it or not, and respond with the patched request. For example, you can add labels, add a sidecar container, change the docker image used, and so on.
- The **validating webhook** is triggered at the end of the flow. It receives the request and can respond if the request is accepted or rejected but can't modify it. For example, you can only validate a deployment that doesn't run a privileged container, check the signature of a container, and so on.

Using admission webhook to implement a backdoor

Continuer sans accepter

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

Paramètres du cookies

Tout accepter

Now, imagine that you are an **attacker** and you compromised a Kubernetes cluster. Your next step would be to deploy backdoors in order to sell it or to come back if your initial access is discovered. A simple way to do this would be to deploy a backdoored container. You can do this manually, with a deployment, for example. But there is a big chance of attracting the attention of a cluster user.

As myself being a **Kubernetes** user, `kubectl get pods` is probably one of the commands I use the most. And this kind of deployment is suspicious because it has been created by a user with a high privilege. So, let's try something clever: what if a cluster user can inject a backdoor for you? One way to achieve this would be to use a mutating webhook in the cluster.

Now, we won't create any pod manually. We will create a mutating webhook in the cluster that each time a new pod is created, it injects a backdoor into it. As we saw earlier, `mutatingwebhookconfiguration` is a good candidate for this. Let's use the sidecar container to inject a backdoored container in all pods.

[Continuer sans accepter](#)

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

[Paramètres du cookies](#)[Tout accepter](#)

After the installation of this webhook, every pod of your **Kubernetes cluster** will have a backdoored sidecar container. So the attacker will be able to come back to your Kubernetes cluster through any of the pod deployed. Of course, it is very noisy to inject a backdoor into every pod.

We can modify the **mutating webhook** to adjust the injection rate or target some specific pods to get something less obvious. This technique has a lot of advantages for the attacker. You can deploy whatever you want as a sidecar container from backdoor to cryptominers. But the most important interest is stealth. The cluster is itself deploying backdoor without your intervention.

We can also imagine a system where the webhook is pointing to an external server. In this case, the attacker can change the behavior of the webhook on the fly. He can remain silent for months and activate the

payload only when he needs it. That’s why detecting this kind of attack can be really hard.

Protect your Kubernetes cluster

A good **RBAC** is not easy to set up and maintain in your Kubernetes cluster and it is not always enough to prevent this kind of attack. Indeed, this attack can be deployed by a malicious cluster user, who can create a webhook from scratch.

There is no perfect way to protect your Kubernetes cluster. You can deactivate **webhooks** because the attacker can still create a webhook such as the `nginx-ingress controller` because the user able to create a webhook because the user able to create a webhook probably also able to modify valid webhooks. The `nginx-ingress controller` probably the best way to deal with this attack. The `nginx-ingress controller` can notify you when it detects a mutating webhook. The `nginx-ingress controller` gives you a clue that your Kubernetes cluster is under attack. In this case, inspecting webhooks might be a good way to detect it. You can add a notification every time a mutating webhook is created but it might be really noisy.

Conclusion

The strength of this attack might also be a good way to detect it. The attacker introduces a differential between what we want to deploy and what is really deployed by your Kubernetes cluster. This differential can be handled by the **gitops approach**. Indeed, by using tools like **ArgoCD**, you can check if what you have in your Kubernetes cluster is actually what you asked to deploy from your repo. With this mechanism, no mutating webhook would be able to modify your pods to install backdoors in it; otherwise, ArgoCD will correct the deployment.

Continuer sans accepter

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

Paramètres du cookies

Tout accepter



Alexandre Hervé

Alexandre is a SecOps Engineer at Theodo Cloud. He assesses the security level of cloud infrastructures and helps protect them against malicious behaviors. He enjoys CTF and playing the piano.

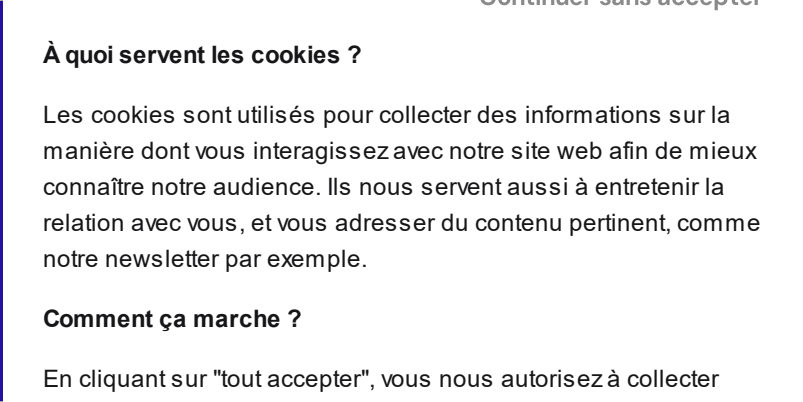
Articles similaires



TECHNOLOGY • 4 MIN

How to master network policies in a Kubernetes cluster?

Learn how to easily restrict network traffic between your pods in a kubernetes cluster using Network Policies.



TECHNOLOGY • 3 MIN

How to use taints, tolerations, and node affinities to isolate workloads in a Kubernetes cluster?

This article explains how to use taints, tolerations, and node affinities to isolate workloads in a Kubernetes cluster.



TECHNOLOGY • 3 MIN

How can an attacker use malicious admission controllers to settle in your Kubernetes cluster without you being aware of it?

This article explains how can an attacker use malicious admission controllers to settle in your Kubernetes cluster without you being aware of it?

Continuer sans accepter

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

Paramètres du cookies

Tout accepter

Vous avez un projet ?

Audit, test d'intrusion, sécurisation ? Nous sommes à votre écoute

Nous contacter

Theodo Cloud Security,
Expert en Cybersécurité à Paris
1 rue de Saint-Pétersbourg
Paris 75008

Copyright © 2021 Theodo Cloud
Security

GLOBAL

À propos

Rejoignez-nous

Blog

Contact

OFFRES

Audit de sécurité

Test d'intrusion

Sécurisation

Cloud & infrastructure

CONFIDENTIALITÉ

Mentions légales

Gestion des cookies

Continuer sans accepter

À quoi servent les cookies ?

Les cookies sont utilisés pour collecter des informations sur la manière dont vous interagissez avec notre site web afin de mieux connaître notre audience. Ils nous servent aussi à entretenir la relation avec vous, et vous adresser du contenu pertinent, comme notre newsletter par exemple.

Comment ça marche ?

En cliquant sur "tout accepter", vous nous autorisez à collecter ces données, et nous nous engageons les protéger. Vous pouvez également paramétrer les cookies auxquels nous aurons accès. Ou sinon, vous pouvez tout simplement cliquer sur "Continuer sans accepter". Nous vous laissons le choix pour vos données personnelles !

Paramètres du cookies

Tout accepter