OH Agent and Edge Workload Runtime Security

Status: In Progress

Sponsor User: *<todo>*

Date of Submission: 03 Jul 2023

Submitted by: Rahul Jadhav

Affiliation(s): AccuKnox

<Please fill out the above fields, and the Overview, Design and User Experience sections below for an initial review of the proposed feature.>

Scope and Signoff: *(to be filled out by Chair)*

<Please fill out the Overview, Design and User Experience sections for an initial review of the proposed feature.>

Overview

A mailing list for this sub-group has been created at https://lists.lfedge.org/g/OpenHorizonWorkloadSecurity and you can subscribe to the meeting calendar there, or by sending an email to OpenHorizonWorkloadSecurity+subscribe@lists.lfedge.org

<Briefly describe the problem being solved, not how the problem is solved, just focus on the problem. Think about why the feature is needed, and what is the relevant context to understand the problem.>

OpenHorizon provides the ability to flexibly deploy edge workloads by providing its own orchestrating elements. As an edge service provider who uses OpenHorizon this provides immense flexibility in deploying and managing edge operations.

However, this flexibility comes at a tradeoff wherein the workloads deployed on edge might not necessarily be created by security savvy developers and might have vulnerability. The impact of such a vulnerability exploit can be immense since it can bring the edge to a halt, but more importantly, the attacker has the possibility of leveraging a security gap in one workload to target another workload on the same edge node since they are colocated. The edge workloads may contain sensitive data related to user and hence needs to be protected.

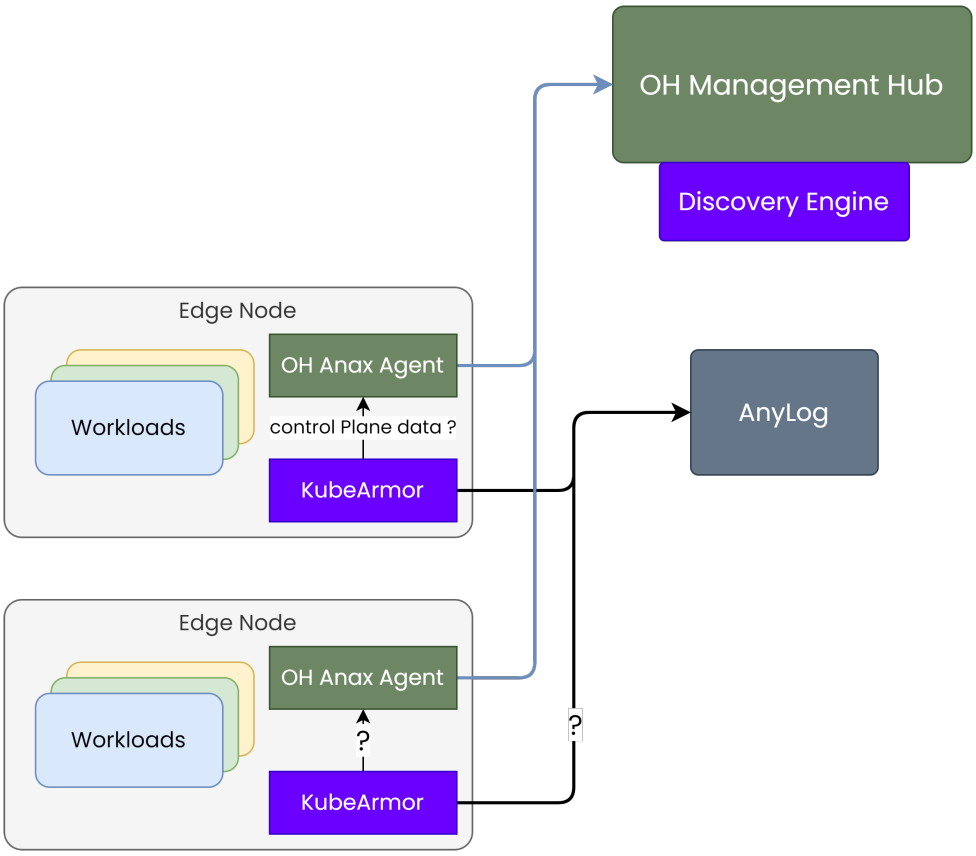
Furthermore, it is important for the edge administrators or service providers to have monitoring options for edge workloads. This could be needed further for compliance and regulatory purposes.

As an example, IEC 62443 standard defines following principles to be followed in the OT sector:

- Principle of least privilege: Provide edge node components and external interfaces only the required access and deny everything else.
- Defense in Depth: Multi layered defense techniques to delay or prevent a cyber attack in the industrial network
- Risk Analysis: Practice used to address risks related to production infrastructure, production capacity etc

Design

<Describe how the problem is fixed. Include all affected components. Include diagrams for clarity. This should be the longest section in the document. Use the sections below to call out specifics related to each aspect of the overall system, and refer back to this section for context. Provide links to any relevant external information.>



OpenHorizon-AnyLog Integration.drawio original draw.io file for any modifications if needed.

Deployment Design

TODO

User Experience

<Describe which user roles are related to the problem AND the solution, e.g. admin, deployer, node owner, etc. If you need to define a new role in your design, make that very clear. Remember this is about what a user is thinking when interacting with the system before and after this design change. This section is not about a UI, it's more abstract than that. This section should explain all the aspects of the proposed feature that will surface to users.>

Deployment UX

1. Should we consider k8s mode of deployment or pure-containerized mode of deployment? KubeArmor works best with k8s mode of deployment and is the recommended mode. Having said that, the previous integration/demo/POC done with OH was in pure-containerized mode.
2. How would the deployment of KubeArmor on the target edge node happen? Will it be deployed as a separate workload with its own control plane or will it be integrated into the same control plane as that of OH?
    a. There is a value in keeping KubeArmor and associated tooling decoupled from Anax and OH Management Hub. This would allow independent updates and essentially the security should be considered as one more addon from the service provider side of things.
    b. The real challenge here is how would OH framework allow extensions to be built to integrate third party tooling?
3. Ship the hardening policies along with the KubeArmor installation.

Day2 Operations UX

1. How would the policy add/delete/list/modify work?
2. How would the recommended policies be shown to the user?
3. How would the SIEM tools integrations be done and at what point?
4. How would upgrade of KubeArmor be handled?

Use-cases to consider

<TODO: *Every security use-case could have a corresponding set of tags that could indicate the fulfilled compliance control, or attack framework (for e.g., MITRE) control fulfilled.*>

Observability & Monitoring use-cases

Security Event Monitoring:

1. File Integrity Monitoring: Any changes to the systems folders should be monitored/audited.
2. Reverse Shell execution
3. Use of security sensitive primitives: setuid(), setguid(),chmod(),chown(),
4. Updates to root certificates folder
5. Use of `kubectl exec` to gain shell access in the pod
6. Privilege escalation attempted
7. Monitor for external networks access
8. Suspicious IP detection (for e.g. using Feodo Blocked IP List)
9. Monitor for use of DGA (Domain Generation Algorithms) in the workload

Application Performance Monitoring:

1. Excessive CPU usage: >90% of CPU used consistently for > 2 mins
2. Excessive Memory usage: >80% of allocated memory used
3. ...

Goals

1. Install and run Open Horizon all-in-one, publish and deploy HomeAssistant and KubeArmor with test security policy
2. Demonstrate how to monitor the listed events and access the results

Deliverables

- Documentation allowing anyone to replicate the results of the goals listed above
- Demo video showing the results

Components

- Open Horizon - to deliver and manage running workloads
- KubeArmor - to monitor and enforce security policy on host and workloads
- HomeAssistant - example service

Protection: Hardening use-cases

Node Hardening:

1. Protect systems folders: Do not allow updates to kernel modules on the host.
2. Prevent root certificates updates

Workload/Pod/Container Hardening:

1. Protecting workload Secrets. Secrets could be injected in the workloads using volume mounts, environment vars, etc. Provide clear guidelines and specific tooling to secure such secrets.
2. Protecting sensitive assets mounted using volume mount points

Protection: Enforcing principle of least privilege

1. Network Segmentation and enforcing least privilege network access
2. Enforce Process Whitelisting
3. Enforce least permissive access to sensitive assets. All volume mount points can be considered sensitive assets.
4. Enforce least permissive process based network control. Only allow certain set of processes to do network communication.

Protection: Enforcing Network Protection

1. Enforce Ingress/Egress controls using CIDRSets, Domain names, Protocols/Ports
2. Auto Discover Network Protection rules.

Workload Forensics

1. Workload Process Monitoring
2. Workload Sensitive Asset access
3. External Network exposure for workloads
4. Ability to query forensics details for a specified time duration from past X days.

Other Topics:

1. Leveraging Confidential Computing for hardware based protections

charisse lu Security guidelines for workload creators - discussion

Command Line Interface

<Describe any changes to the hzn CLI, including before and after command examples for clarity. Include which users will use the changed CLI. This section should flow very naturally from the User Experience section.>

1. How to extend Anax cli and integrate with karmor cli? Can we expect the user to have two clis? Does Anax cli offer pluggable interfaces?
2. The policy add/delete/update/list should be handled through this cli.

External Components

<Describe any new or changed interactions with components that are not the agent or the management hub.>

Affected Components

<List all of the internal components (agent, MMS, Exchange, etc) which need to be updated to support the proposed feature. Include a link to the github epic for this feature (and the epic should contain the github issues for each component).>

Security

<Describe any related security aspects of the solution. Think about security of components interacting with each other, users interacting with the system, components interacting with external systems, permissions of users or components>

APIs

<Describe and new/changed/deprecated APIs, including before and after snippets for clarity. Include which components or users will use the APIs.>

Build, Install, Packaging

<Describe any changes to the way any component of the system is built (e.g. agent packages, containers, etc), installed (operators, manual install, batch install, SDO), configured, and deployed (consider the hub and edge nodes).>

Documentation Notes

<Describe the aspects of documentation that will be new/changed/updated. Be sure to indicate if this is new or changed doc, the impacted artifacts (e.g. technical doc, website, etc) and links to the related doc issue(s) in github.>

Test

<Summarize new automated tests that need to be added in support of this feature, and describe any special test requirements that you can foresee.>