Kubernetes Container Created with Excessive Linux Capabilities

edit

This rule detects a container deployed with one or more dangerously permissive Linux capabilities. An attacker with the ability to deploy a container with added capabilities could use this for further execution, lateral movement, or privilege escalation within a cluster. The capabilities detected in this rule have been used in container escapes to the host machine.

Rule type: query

Rule indices:

- logs-kubernetes.*

Severity: medium

Risk score: 47

Runs every: 5m

Searches indices from: None (Date Math format, see also `Additional look-back time`)

Maximum alerts per execution: 100

- https://man7.org/linux/man-pages/man7/capabilities.7.html
- https://docs.docker.com/engine/reference/run/#runtime-privilege-and-linux-capabilities

Tags:

- Data Source: Kubernetes
- Tactic: Execution
- Tactic: Privilege Escalation

Version: 5

Rule authors:

- Elastic

Rule license: Elastic License v2

Investigation guide

Triage and analysis

Investigating Kubernetes Container Created with Excessive Linux Capabilities

Linux capabilities were designed to divide root privileges into smaller units. Each capability grants a thread just enough power to perform specific privileged tasks. In Kubernetes, containers are given a set of default capabilities that can be dropped or added to at the time of creation. Added capabilities entitle containers in a directory read and execute permission checks. NET_ADMIN - Perform various network-related operations. SYS_ADMIN - Perform a range of system administration operations. SYS_BOOT - Use reboot(2) and kexec_load(2), reboot and load a new kernel for later execution. SYS_MODULE - Load and unload kernel modules. SYS_PTRACE - Trace arbitrary processes using ptrace(2). SYS_RAWIO - Perform I/O port operations (iopl(2) and ioperm(2)). SYSLOG - Perform privileged syslog(2) operations.

False positive analysis

- While these capabilities are not included by default in containers, some legitimate images may need to add them. This rule leaves space for the exception of trusted container images. To add an exception, add the trusted container image name to the query field, kubernetes.audit.requestObject.spec.containers.image.

Setup

edit

The Kubernetes Fleet integration with Audit Logs enabled or similarly structured data is required to be compatible with this rule.

Rule query

edit

Framework: MITRE ATT&CK^{TM}

- Tactic:
  - Name: Privilege Escalation
  - ID: TA0004
  - Reference URL: https://attack.mitre.org/tactics/TA0004/
- Technique:
  - Name: Escape to Host
  - ID: T1611
  - Reference URL: https://attack.mitre.org/techniques/T1611/
- Tactic:
  - Name: Execution
  - ID: TA0002
  - Reference URL: https://attack.mitre.org/tactics/TA0002/
- Technique:
  - Name: Deploy Container
  - ID: T1610
  - Reference URL: https://attack.mitre.org/techniques/T1610/

VIDEO
Intro to Kibana

VIDEO
ELK for Logs & Metrics

Was this helpful? 👍 👎

Join us

Careers

Career portal

Trust & Security

Trust center

EthicsPoint portal

ECCN report

Ethics email

Investor relations

Investor resources

Governance

Financials

Stock

EXCELLENCE AWARDS

Previous winners

ElasticON Tour

Become a sponsor

All events