// HALBORN

DRAFT

0xnodes CI/CD audit

DEVSECOPS AUDIT

Prepared by: **Halborn**

Date of Engagement: **January 28, 2021** - **January 28, 2021**

Visit: **Halborn.com**

DRAFT

DRAFT

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	1/28/2022	alex.yang

CONTACTS

CONTACT	COMPANY	EMAIL
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Alex Yang	Halborn	Alex.yang@halborn.com
Bryan recinos	Halborn	Bryan.Recinos@halborn.com
Ran Xing	Halborn	Ran.xing@halborn.com

DRAFT

EXECUTIVE OVERVIEW

DRAFT

1.1 INTRODUCTION

This report contains a detailed list of findings, highlighting the severity and impact of each one and certain proposed resolutions. The report will further guide businesses to improve their security policies, procedures, controls, and practices.

1.2 TEST APPROACH & METHODOLOGY

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

DRAFT

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

DRAFT

1.3 SCOPE

IN-SCOPE:

The security assessment was scoped to the following

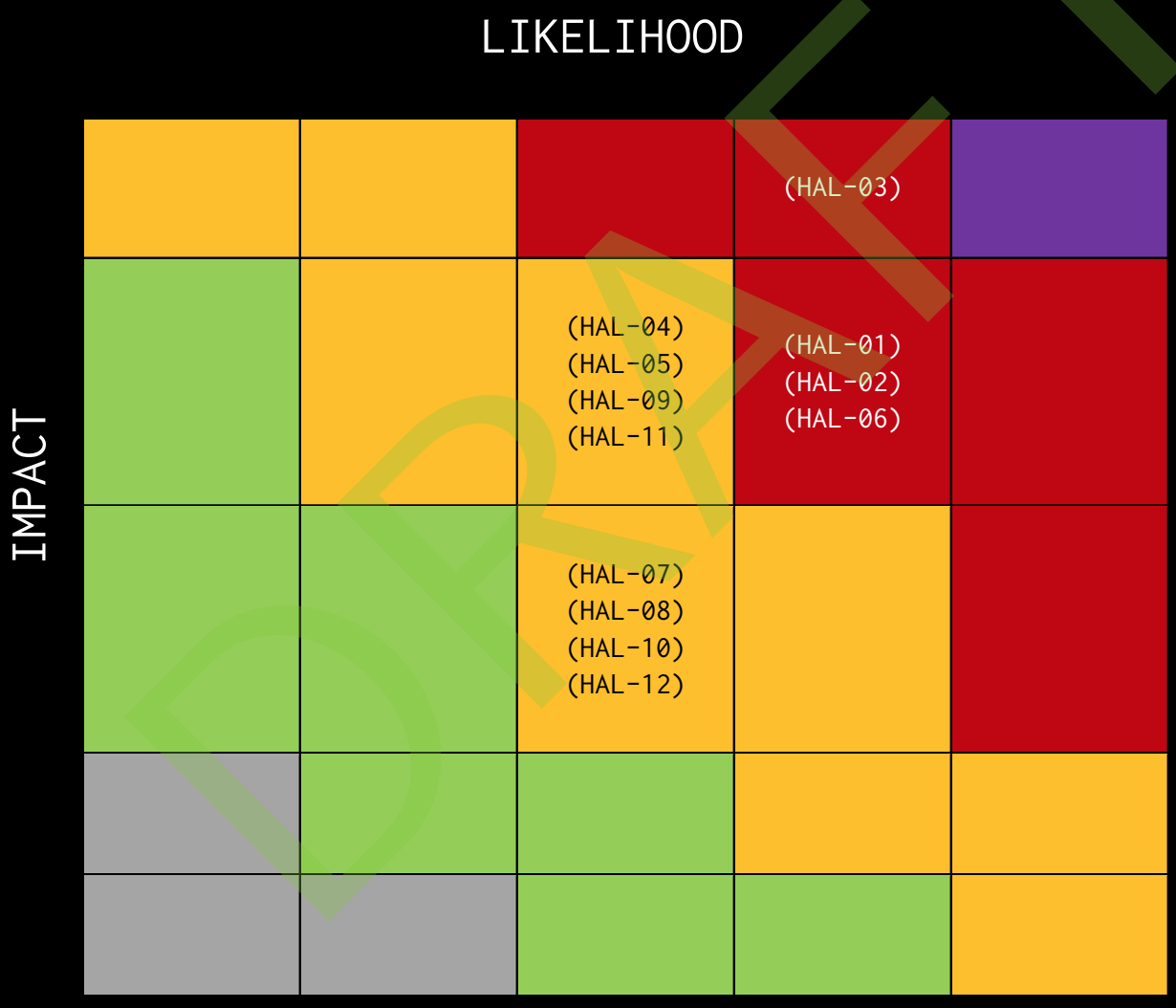
0xnodes github CI/CD workflow

GKE cluster

DRAFT

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	4	8	0	0



DRAFT

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - No code scanning	High	-
HAL02 - No secrets detection	High	-
HAL03 - No container scanning	High	-
HAL04 - No Github configured environment for deployment	High	-
HAL05 - Use openID connect when authentication	High	-
HAL06 - use self-hosted runners	Medium	-
HAL07 - use codeowners to monitor changes	Medium	-
HAL08 - use trust and stable actions version	Medium	-
HAL09 - GKE - Overly Permissive RoleBindings	Medium	-
HAL10 - GKE - No Pod Security Constraint Enforcement	Medium	-
HAL11 - GKE - No Network Security Policies	Medium	-
HAL12 - GKE - Insufficient Logging	Medium	-

DRAFT

FINDINGS & TECH DETAILS

DRAFT

3.1 (HAL-01) no code scanning - HIGH

Description:

Your code has a potentially dangerous attribute in a class, or unsafe code that can lead to unintended code execution. We should detect such vulnerability as early as possible. Static Application Security Testing should be used here.

Recommendations:

We can enable SAST by enabling the Github advance security license or open source tool. Each PR should be tested properly to detect any vulneraility here.

***References**:
tfsec
SAST
synk

DRAFT

3.2 (HAL-02) no secrets detection - HIGH

Description:

To prevent any secrets pushed into the repo, we need to build a way to detect such behaviour.

Recommendation:

gitleaks can be used here to detect the secrets

References:
git-leaks

DRAFT

3.3 (HAL-03) no container scanning - HIGH

Description:

container scanning is useful to detect the potential vulnerability in advance.

Recommend:

We recommend to use GCR vuln scanning or build a workflow to scan it before we push to ECR.

References:
trivy-action

DRAFT

3.4 (HAL-04) No Github configured environment for deployment - MEDIUM

Description:

Environment protection rules can be applied for different environment. This can help to mitigate the risk of deploying to production without any approval.

Recommendation:

We recommend to create different environments in Github Actions.

References:
using environments for deployment

DRAFT

3.5 (HAL-05) Use openID connect when authentication - MEDIUM

Description:

If your GitHub Actions workflows need to access resources from a cloud provider that supports OpenID Connect (OIDC), you can configure your workflows to authenticate directly to the cloud provider. This will let you stop storing these credentials as long-lived secrets and provide other security benefits.

Recommendation:

We recommend to confiure OIDC in GCP

References:
OpenID connect to access cloud resources

DRAFT

3.6 (HAL-06) use self-hosted runners - HIGH

Description:

Using self-hosted runners for CI/CD to get better secure instead of using Github hosted runners.

Recommend:

We recommend to use self-hosted runners for CI/CD

References:
self-hosted-runner

DRAFT

3.7 (HAL-07) Use codeowners to monitor all changes - MEDIUM

Description:

You can use the CODEOWNERS feature to control how changes are made to your workflow files. For example, if all your workflow files are stored in .github/workflows, you can add this directory to the code owners list, so that any proposed changes to these files will first require approval from a designated reviewer.

Recommendation:

We recommend to enable CODEOWNERS to monitor changes as well as adjust the settings to get at least one approval and finish tests for PRs.

References:
code owners

DRAFT

3.8 (HAL-08) Use trust and stable actions version - MEDIUM

Description:

Github actions are very popular and we should consider to prevent the supply-chain attack to prevent any malicious one.

Recommendation:

We recommend to use verified actions and better to use stable one such as actions/setup-node@v2 instead of setup-node@master.

DRAFT

3.9 (HAL-09) Overly Permissive RoleBindings - MEDIUM

Description:

The external-metrics-reader (uas-hpa-external-metrics-reader) cluster role was found to have List permission on all resources in the kube-system namespace, an attacker compromising a resource bound to this role would be able to read all values, including secrets, within the namespace.

Recommendation:

We recommend to grant least privileges for this role.

DRAFT

3.10 (HAL-10) No Pod Security Constraint Enforcement - MEDIUM

Description:

While a none-root security context was observed on the realy-staging replicaset a number of security controls were missing. Additionally, this appeared to be hardcoded into the deployment rather than through an admission controller which may become cumbersome to maintain as additional deployments are introduced.
- Below is a none-exhaustive list of the missing controls:
Host namespaces
privileged
allowPrivilegeEscalation
readOnlyRootFilesystem
seLinux
seccomp (can be enforced as a security context)
linux capabilities (these should be restricted to only those that are required for the service to function)
(more info can be found here https://kubernetes.io/docs/concepts/policy/pod-security-policy/#host-namespaces)
Note: PSP's have been depreciated in the latest versions of Kubernetes in favour of custom admission controllers (OPA/Gatekeeper, K-Rail, Kyverno).
Recommendation
We recommend to apply the security policy for the deployment.

References:
pod-security-policy

DRAFT

3.11 (HAL-11) No Network Security Policies - MEDIUM

Description:

While Calico was in place no network policies were observed, in its current state this does not impose a risk,
however as more applications are added to the cluster segregation should be implemented where appropriate to restrict blast
radius of compromised pods.

Recommendation:

We recommend to appropriate rules should be applied for the namespace based or pod based.

References:
network-security-policy

DRAFT

3.12 (HAL-12) Insufficient Logging - MEDIUM

Description:

Fluentbit was deployed but there was no evidence that these logs were actively monitored.

Recommendation:

We recommend to add proper montioring for the k8s and logging via datadog or newrelic.

References:
datadog
newrelic

DRAFT

THANK YOU FOR CHOOSING

// HALBORN

DRAFT