tufin

SecureCloud™

Security Posture Report

SecureCloud account: Demo

28 May 2020, 04:20 PM

Table of Contents

Public Cloud Summary

3	6,309	13,694	1
CLOUD ACCOUNTS	ASSETS IN VIOLATION	TOTAL ASSETS	ACCOUNTS FAIL CIS BENCHMARK

Risky Ports

1,748 out of 13,694 assets
have high-risk ports

Overly Permissive Access

3,227 out of 13,694 assets
have overly permissive access

Policy Violations

✓ **0 out of 13,694 assets**
have cloud vendor settings that don't comply with your policy

Tag Usage

✓ **0 out of 13,694 assets**
without tags

CIS Benchmark Public Cloud

10 out of 57 tests
fail the CIS benchmark

Kubernetes Summary

7	770	8,729	3
CLUSTERS	WORKLOADS IN VIOLATION	TOTAL WORKLOADS	CLUSTERS FAIL CIS BENCHMARK

Workload Security

85 out of 8,729 workloads
have a security context that violates your risk configuration

Kubernetes Vulnerabilities

7 out of 7 cluster
have known vulnerabilities

Container Vulnerabilities

770 out of 8,729 workloads
have containers with score B and lowe

Permissive Network Policies

3 out of 7 cluster
have permissive Kubernetes network policies

CIS Benchmark Kubernets

70 out of 133 tests
fail the CIS benchmark

Public Cloud Account Details

Green_dev

3,521 ASSETS IN VIOLATION	**4,560** TOTAL ASSETS	**Failed** CIS BENCHMARK

Violation Type	Assets	Violation Details
Risky ports	698	RDP (195 assets) SSH (568 assets)
Permissive source	2,174	Any (42 assets) 10.0.0.0/8 (2,174 assets)
Permissive service	987	Any (987 assets)
Policy violations	--	--
Tag Usage	--	--
CIS benchmark	--	10 out of 19 tests fail

Green_prod

352 ASSETS IN VIOLATION	**4,574** TOTAL ASSETS	**Passed** CIS BENCHMARK

Violation Type	Assets	Violation Details
Risky ports	352	Telnet (235 assets) POP3 (65 assets) Microsoft-DS (235 assets) RDP (155 assets) Net-Bios (13 assets) SSH (297 assets)
Permissive source	65	Any (16 assets) 10.0.0.0/8 (65 assets)
Permissive service	--	--
Policy violations	--	--
Tag Usage	--	--
CIS benchmark	--	--

Green_stag

2,436 ASSETS IN VIOLATION	**4,560** TOTAL ASSETS	**Passed** CIS BENCHMARK

Violation Type	Assets	Violation Details
🟥 Risky ports	698	RDP (235 assets) SSH (485 assets)
🟥 Permissive source	2,176	Any (635 assets) 10.0.0.0/8 (2,098 assets)
🟥 Permissive service	988	TCP:0-65535 (988 assets)
🟩 Policy violations	--	--
🟩 Tag Usage	--	--
🟩 CIS benchmark	--	--

Kubernetes Cluster Details

C_dev

357	**1,247**	**3**	**Failed**
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

Violation Type	Workloads	Violation Details
Kubernetes Vulnerabilities	--	1 High 2 Low
Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (12 workloads)
Container Vulnerabilities	357	Score F (52 workloads) Score C (112 workloads) Score B (193 workloads)
Permissive Network Policies	--	12 Namespaces
CIS benchmark	--	10 out of 19 tests fail

C_prod

12	**1,247**	**3**	**Failed**
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

Violation Type	Workloads	Violation Details
Kubernetes Vulnerabilities	--	2 Medium
Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (13 workloads)
Container Vulnerabilities	12	Score B (12 workloads)
Permissive Network Policies	--	12 Namespaces
CIS benchmark	--	10 out of 19 tests fail

C_stag

12	1,247	1	Failed
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

Violation Type	Workloads	Violation Details
Kubernetes Vulnerabilities	--	1 High
Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (10 workloads)
Container Vulnerabilities	12	Score C (12 workloads)
Permissive Network Policies	--	12 Namespaces
CIS benchmark	--	10 out of 19 tests fail

C_stag B

38	1,247	3	Failed
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

Violation Type	Workloads	Violation Details
Kubernetes Vulnerabilities	--	2 Low
Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Capability NET_ADMIN (8 workloads) Capability NET_RAW (8 workloads) CPU unlimited (8 workloads) Privileged (8 workloads) Privileged escalation (8 workloads) Read-only OS mount (8 workloads) Root allowed (12 workloads) Unmasked ProcMount (8 workloads)
Container Vulnerabilities	36	Score F (12 workloads) Score C (12 workloads) Score B (12 workloads)
Permissive Network Policies	--	--
CIS benchmark	--	10 out of 19 tests fail

CA_dev

125	**1,247**	**3**	**Failed**
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

	Violation Type	Workloads	Violation Details
🟥	Kubernetes Vulnerabilities	--	3 Low
🟥	Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (10 workloads)
🟥	Container Vulnerabilities	124	Score C (124 workloads)
🟩	Permissive Network Policies	--	--
🟥	CIS benchmark	--	10 out of 19 tests fail

CA_prod

12	**1,247**	**3**	**Failed**
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

	Violation Type	Workloads	Violation Details
🟥	Kubernetes Vulnerabilities	--	2 Low
🟥	Workload Security	12	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (8 workloads)
🟥	Container Vulnerabilities	12	Score B (12 workloads)
🟩	Permissive Network Policies	--	--
🟥	CIS benchmark	--	10 out of 19 tests fail

CB_dev

214	1,247	3	Failed
WORKLOADS IN VIOLATION	TOTAL WORKLOADS	KUBERNETES VULNERABILITIES	CIS BENCHMARK

	Violation Type	Workloads	Violation Details
🟥	Kubernetes Vulnerabilities	--	3 Low
🟥	Workload Security	13	hostNetwork (12 workloads) hostPID (12 workloads) Privileged (8 workloads) Root allowed (13 workloads)
🟥	Container Vulnerabilities	214	Score B (214 workloads)
🟩	Permissive Network Policies	--	--
🟥	CIS benchmark	--	10 out of 19 tests fail

Terms

Public Cloud

Term / Phrase	Description
Assets	An asset in SecureCloud is a server or cloud service in your cloud account such as a virtual machine or bucket storage.
CIS Benchmark Public Cloud	The CIS Benchmark for Public Cloud is a subset of the security controls developed by the Center for Internet Security (CIS) and adapted for Microsoft Azure and Amazon AWS. These controls have gained widespread acceptance as a baseline for public cloud security and can be downloaded from the CIS website.
Permissive service	Permissive service is where UDP or TCP ingress is allowed from any port (Min=0, Max=65535). This is configured in in Risk Configuration.
Permissive source	Permissive source is where the access allowes unlimited range of IPs (class A subnet mask e.g. 10.0.0.0/8) or any source (Source=ANY). This is configured in in the "Risk Configuration" section of the "Configuration" area.
Policy Violations	A policy violation is a case where access is allowed while going against the Cloud Policy.
Risky ports	List of ports that expose your environment to risk. the list is configured in the "Risk Configuration" section of the "Configuration" area.

Kubernetes

Term / Phrase	Description
Capability NET_ADMIN	Allows various network-related operations including interface configuration, administration of IP firewall, modifying routing tables and more.
Capability NET_RAW	Any kind of packet can be forged, which includes faking senders, sending malformed packets, etc., this also allows to bind to any address (associated to the ability to fake a sender this allows to impersonate a device, legitimately used for \"transparent proxying\" as per the manpage but from an attacker point-of-view this term is a synonym for Man-in-The-Middle). This should be prevented by dropping the NET_RAW capability.
Capability SYS_ADMIN	Never enable this capability - it is equivalent to root.
CIS Benchmark Kubernets	The CIS Benchmark for Kubernetesis a subset of the security controls developed by the Center for Internet Security (CIS) and adapted for Kubernetes . These controls have gained widespread acceptance as a baseline for public cloud security and can be downloaded from the CIS website.
Container vulnerabilities	SecureCloud scans your containers and gives each one a security score from A to F, according to the number and type of software vulnerabilities found, where A - the best score - represents the lowest risk and F - the worst score - represents the highest risk.

Term / Phrase	Description
CPU unlimited	The Container has no upper bound on the CPU resources it can use. The Container could use all of the CPU resources available on the Node where it is running. This should be prevented by setting the CPU limit at the container level or through a LimitRange at the namespace level.
hostIPC	Pod can share the host's IPC namespace.
hostNetwork	Pod can use the host's network namespace. This gives the pod access to the loopback device, services listening on localhost, and could be used to snoop on network activity of other pods on the same node
hostPID	Pod can use the host's process ID namespace. Note that when paired with ptrace this can be used to escalate privileges outside of the container (ptrace is forbidden by default).
Kubernetes vulnerabilities	Kubernetes Vulnerabilities are known vulnerabilities in Kubernetes implementations. The full list of vulnerabilities can be seen here on GitHub.
Memory unlimited	The Container has no upper bound on the amount of memory it uses. The Container could use all of the memory available on the Node where it is running which in turn could invoke the OOM Killer. Further, in case of an OOM Kill, a container with no resource limits will have a greater chance of being killed. This should be prevented by setting the memory limit at the container level or through a LimitRange at the namespace level.
Privileged	Container is privileged. Process in privileged containers are essentially equivalent to root on the host.
Privileged escalation	Process in this container can gain more privileges than their parent process. This should be prevented by setting 'allowPrivilegeEscalation: false'
Read-only OS mount	Container can read from a host OS volume.
Root allowed	Process in this container can run as root. This should be prevented by setting 'runAsNonRoot: true' and providing a non-zero runAsUser value or a USER value in the Dockerfile. It is advisable to select a value greater than 10000, as this reduces the likelihood that this value is already taken on the host system.
Running as root	Process in this container are running with UID 0 (root).
Unmasked ProcMount	Container has full access to host's /proc.
Unsafe sysctls	Pod can use unsafe sysctls. You should only allow unsafe sysctls for very special situations such as high-performance or real-time application tuning.
Security context	A security context defines privilege and access control settings for pods and containers.
Workload	Workloads are objects that set deployment rules for pods. Based on these rules, Kubernetes performs the deployment and updates the workload with the current state of the application. Workloads let you define the rules for application scheduling, scaling and upgrade.

Term / Phrase	Description
Workload security	Workload security refers to a pod or container having a security context that doesn't comply with your policy
Writable OS mount	Container can read and write to a host OS volume.
Writable root filesystem	Container has a writable root filesystem. An immutable root filesystem can prevent malicious binaries being added to PATH and increase attack cost. This should be prevented by setting 'readOnlyRootFilesystem: true'.

Legend

Term / Phrase	Description
✻	Any
⚙	Service
🖧	Subnet
⎈	Kubernetes Cluster
aws	AWS
▲	Azure
☁	Google Cloud