Confidential Space Security Review

Google Inc
Version 1.0 – December 1, 2022

Prepared By
Viktor Gazdag
Oliver Brooks
Sebastian Chlopecki

Prepared For
Acacia Tena

1 Executive Summary

Synopsis

During the summer of 2022, Google engaged NCC Group to conduct a security assessment of the Confidential Space product. The system provides a confidential computing environment that allows cloud customers to run workloads in the cloud that can be attested to run a specific payload with high assurances that the workload was not and cannot be tampered with. Container images and source code were provided for various components of the Confidential Space platform.

Overall, no security vulnerabilities were identified in the in-scope components of the Confidential Space product during the allotted testing time.

Assessment Description

NCC Group reviewed the design and architecture of the proposed Confidential Space deployment. During this time NCC Group consultants:

- Reviewed engineering and product documentation, such as data flow diagrams, product requirements, deployment scenarios, functional and design specifications.
- Analyzed the environment's security model and trusted boundaries to determine whether it adequately meets security objectives and requirements.
- Identified possible weaknesses in the existing design such lack of encryption, inadequate auditing and logging, over permissive permission, lack of separation of duty, lack of assurance of measured boot and missing security controls in the underlying networks, applications, or operating systems.

For the host and hardening element of the engagement, NCC Group consultants performed numerous checks against the infrastructure:

- Performed various checks including auditing the open ports on the host and the container, checking for inappropriately permissive network ingress/egress filtering, outdated Linux kernels, common privilege escalation vectors on the host and checking for common Linux misconfigurations within the container and on the host.
- Consultants checked for the presence of at rest disk encryption and performed thorough checks on the filesystem mounts to ensure that there was no way for sensitive data to be persisted on the host after a workload finishes executing and shuts down.
- Additionally, consultants checked for overly permissive configuration on the GCP service account which starts the VM, potential vectors to abuse the GCP Metadata Service to escape the workload container, general Docker container escapes from workload to host, and the presence of hardcoded credentials on the host filesystem.
- Finally, consultants spent some time performing threat modeling activities with the product development team to establish whether there were any ways that the underlying GCP infrastructure could be hardened.

For source code review, NCC used both automated tools and manual review to: identify known vulnerabilities and outdated dependencies, identify and verify enforcement of security controls, review application logic for errors, check data processing and validation, and to generally verify that the stated design goals are being met by the implementing code.

The source code review involved identification of security-critical and other high-priority areas based on identification of user-controlled inputs, associated data processing and security implications in the larger system. In many cases there is not sufficient time available to perform a full manual review of all the available source code.

This manual analysis was augmented with some automated analysis tools across the entire codebase – this is useful for bulk identification of simple issues, but does not uncover more complex problems and is therefore not a major focus of the assessment.

During the web service assessment the consultants used proven non-invasive testing techniques to quickly identify any weaknesses. The service is assessed from several perspectives, including with no credentials, user credentials, and privileged user credentials. The following common high level test cases were performed such as unvalidated input, broken access control, injection flaws, improper error handling, session management and transport security.

This is followed by an in-depth and thorough assessment of in-scope web services to ensure that correct configuration and recommended practices have been followed to minimise client exposure.

Consultants investigated for authentication weaknesses such as bypassing authentication with spoofed tokens and authentication with replay of authentication information. Input validation cases included finding limitations of defined variables and protocol payloads, data length and type, and serialized format. Validation testing included ensuring strong types, length, and data format. Recorded session information was replayed to attempt to fool services and ensure session management was working intended.

Furthermore, transport security weaknesses were checked, including insufficient certification chain validation and weak cipher suite configuration. The client-server communication channel was tested to ensure it was adequately secured for its intended use.

Scope

NCC Group's evaluation included:

- **Container hardening review**: A review of the container in which the customer workloads will be run.
- **Attestation Verifier Service**: The API that exposes methods that correspond to back-end actions to verify attestations and receive OIDC tokens.
- **Attestation Agent**: A small open-source Golang application which communicates with the Attestation Verifier service.
- **Application Design Review**: The design and architecture of the Confidential Space platform based on the following documentation:
  - Confidential Space Container Launcher
  - Confidential Space VM Image Measurements
  - Confidential Space VM Image Design
  - Confidential Space PRD & User Interactions
  - Attestation Verifier Service API
  - Authenticated Encryption of Confidential Space VM Image Stateful Partition
  - Replies from Google engineers

Testing was performed on a Private Preview version of the product on Google Cloud Platform.

Limitations

The assessment had the following limitations when consultants conducted the review of the application:

- Out of scope for Application Design Review: side-channel attacks, attacks with physical access to the CPU, attacks involving workload author collaboration.

Strategic Recommendations

As the Confidential Space product is designed to be an effective Trusted Execution Environment (TEE) for customers who emphasize security and compliance, best practice advice should be given to customers that details how to securely manage Workload Identity Pool (WIP) configuration, service account permissions and container launch policies.

Include a patching policy timeline to address image related vulnerabilities that affects the Confidential Space Virtual Machine (VM) to prevent any vulnerabilities that would allow an attacker either to gain access to the VM and control its runtime.

2 Dashboard

Target Data		Engagement Data	
Name	Confidential Space	**Type**	Native Application Review, Web Service Assessment, Architecture Review, Host Hardening Review
Type	Desktop Client / Web Application / Host Review / Container Review / Architecture Review	**Method**	Black-box and Code-Assisted
Platforms	Golang	**Dates**	2022-08-15 to 2022-10-18
Environment	Production	**Consultants**	3
		Level of Effort	68 Person-Days

Targets

Attestation Agent	A small open source Golang application which communicates with the Attestation Verifier web service. Code was sourced from the master branch of https://github.com/google/go-tpm-tools/ (git log commit eb06ae8ac6ff49dc16e482fb56d49612273eb8ec, tag: launcher/v0.3.9, tag: cmd/v0.3.9)
Attestation Verifier Service	A blackbox security assessment of the web service which provides an OIDC token when supplied with a valid claim. The web service is hosted at https://confidentialcomputing.googleapis.com
VM and Container Hardening Assessment	A review of the “Confidential-Space” GCP Compute Engine image, in order to determine whether there are any areas of the VM which could be further hardened to prevent compromise/leakage of decrypted PII data.
Architecture Review	A highly detailed, security focused review of the Confidential Space architecture.

Critical High Medium Low Informational

3 Architecture Review

This section evaluates the security decisions made by Google in designing the Confidential Space platform.[1] NCC Group first identified design patterns which are common among secure networks and environments, then the design of the Confidential Space components were compared against the best practices in each of those areas.

The proposed environment's goal was to allow a third party operator to access aggregated data via a trusted workload in an environment that attests its boot chain environment, including container launcher configuration and container image measurements. The workload running by the Confidential Space Virtual Machine's (VM) container uses the attestation to access Trusted Party sensitive data, as the Trusted Party uses the attestation claims to verify the integrity of the Workload. The workload access and processes the sensitive data and then the result (aggregated data) is made available for the operator. This has meant the operator could access only the aggregated data (and not the sensitive data) after the remote attestation (RA) was successful, the workload received the short-lived token and made the aggregated data available.

The scope excludes physical-access-based and side-channel, instead focuses on practical cases. Attack scenarios that depend on the Workload Author and the Trusted Party security posture or security settings are also out of scope.

The following points were taken from one of the documents shared by Google as requirements to be fulfilled by the project to allow a wide range of use cases by using Confidential Collaboration environment:

- An execution environment that excludes the workload administrator from:
  - Having access to the data
  - Modifying or interfering with the behavior of the workload
- An attestation verifier service that can vouch for:
  - The identity and state of the workload
  - The hardened execution environment and its state
  - The identity (proxy via service account) that the workload uses

All points that are listed above seemed to be adhered to by the proposal.

Architecture

Participants

The following personas defined by Google were participating in the environment:

- Workload Author: develop the code for running the workload and create the container image that runs the workload binary. Both the workload binary and container image are accepted and trusted and are allowed to run by the operator.
- Operator: run the workload that access and process the sensitive data. The result is an aggregated data that can be accessed by the operator.
- Trusted Party: in charge of managing data encryption keys, KMS and Workload Identity Pool (WIP).
- Third Party Auditor: audit the workload image and code written by the Workload Author.

1. Introducing Confidential Space: https://cloud.google.com/blog/products/identity-security/announcing-confidential-space

Work Flow

The operator can only access the aggregated data when the binary finished processing the sensitive data. The binary runs in the only container within the confidential virtual machine and the VM is verified by the remote attestation service.

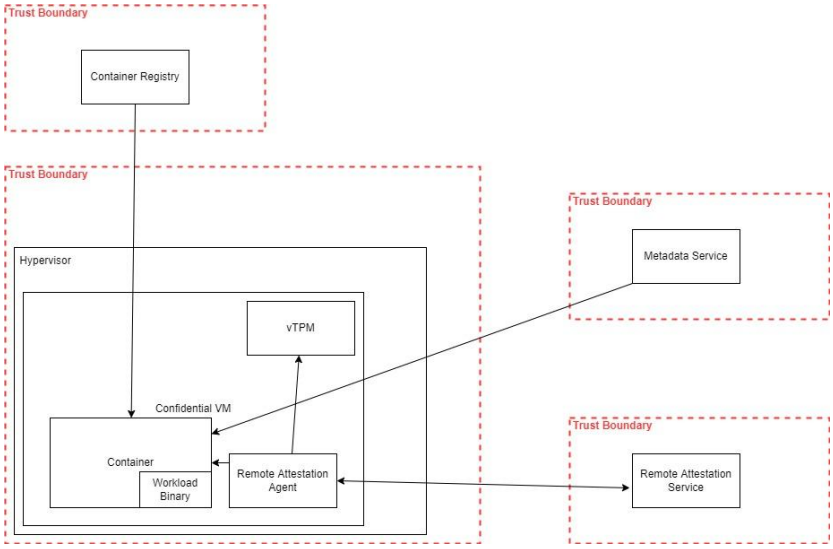

Figure 1: Components and communications in the workflow

The following high level process happens when the operator wants to get access to the data:

- Operator chooses a virtual machine with confidential computing and secure boot, then selects the Confidential Space image created by Google
- An Attestation Key (AK) certification (signed by Google) and an AK are generated when the VM is created in virtual Trusted Platform Module (vTPM)
- The virtual machine boot component and additional parameters (kernel binary, grub.cfg, grub commands, file system hashes) are measured and stored in the event log. If the measurement is correct then the container launcher is started.
- The container launcher pulls the image from the container registry and reads the environment variables from Metadata Service (MDS)
- The container launcher checks if the image and environment variables are compliant with the image launch policy created by the workload author
- If it is compliant then start the container image launch and environment variables are measured and Container-Optimized OS (COS) event log is saved for remote attestation
- The attestation agents send a request to the remote attestation verifier service for the nonce
- The remote attestation verifier service generates a nonce, keeps a record of it and send it to the attestation agent
- The attestation agents send the attestation quote (including nonce+PCR(Confidential Space image measurements+container launcher measurements)+boot log+COS Event Log) signed with the AK to the remote attestation verifier service (confidentialcomputing.googleapis.com)
- The remote attestation verifier service verifies the AK certification, the signature, checks the nonce and compares the Platform Configuration Registers (PCR) values with the replayed COS Event Log and Boot Log
- The PCR values and the relevant data in the logs are compared against a database that stores the base and gold values for PCR values and the event data (managed by Google)

- If everything passed then the remote attestation verifier service generates and sends back the Google signed OpenID Connect (OIDC) token to the agent including the derived attestation claims
- If the validation of the quote fails, the container exits and the requirements will not match the WIP policy
- The agent saves and stores the Google signed OIDC token that expires after 1 hour in a file that is mounted to the container
- Launcher binary starts the container
- The remote attestation agent sends the Google signed OIDC token to the WIP Security Token Service to receive the STS token that can be used for impersonating a service account (to decrypt data) for 1 hour
- The WIP Security Token Service will validate the OIDC token's attributes against the required conditions such as container image hash, swname etc
- The operator input (including environment variables) is also compared against the WIP configuration on top of the workload author launch policy

Analysis

The integrity of the work flow is tightly controlled and measured by vTPM, the Workload Author and the Google services in multiple layers. The confidential computing, workload author launch policy, the Confidential Space image, the container launcher, WIP Security Token Service and remote attestation work together throughout the work flow to provide the requirements for the operator to access the sensitive data.

There are limited opportunities where the workload can submit any input and the boot process is measured and checked by multiple services. The first layer of defense to provide integrity is the confidential computing with shielded VM options (secure and measured boot with vTPM), where the second layer is the hardened Confidential Space image with minimal attack surface. The third layer is the launch policy that specifies how the container will be launched. The fourth layer is the WIP configuration that includes what images are allowed to run and WIP Security Tokens service that limit the attacks and act as a last check for getting a token.

It is worth noting that a big portion of the security depends on the workload author and the trusted party to create the container image, develop the workload binary and limit the permissions of the service account in the WIP configuration.

Measurements

There are two boot measurements with Boot Log and COS Event Log that are sent in the quote by the remote attestation agent that is running in the virtual machine next to the container launcher. The measurements are utilizing AMD SEV (no SEV-ES support, but SEV-SNP will be supported in the future), vTPM (TPM 2.0 only) and verified in explicit remote attestation (RA) with Google service. The hashes of measured values (chain hashing and append-only) are stored in different Platform Configuration Registers (PCR) and sent with the boot measurement logs and COS Event logs for attestation.

The following two measurements and two logs are used in the remote attestation:

- Attested Confidential Space image boot
- Container binary launcher run the image
- Boot Log
- COS Event Log

Confidential Space Image Measurements

The following components are measured for integrity verification:

- Firmware
- GRUB
- Block of devices
- Systemd masked services
- Container launcher (indirectly measured)
- VM is configured with the Confidential Space keys (AKCert)

The firmware measures itself, AMD SEV, secure boot configuration, and EFI application's PE/COFF (shim, Grub).
GRUB measures the grub.cfg file hash, kernel binary hash and all the commands. The commands include dm-verity root hash, masked systemd services, and cloud-init in NoCloud mode.

The firmware event log is stored in PCR0 (CRTM, SEV), and PCR1 contains boot order and boot entries. The PCR4 contains the UEFI boot manager code, while the PCR5 disk's GUID partition table. PCR7 stores if the Secure Boot is enabled and PCR8 has the GRUB command line, kernel command line and modules command line. PCR9 contains all files read, including grub.cfg, kernel. PCR13 contains the container launcher information.

The Confidential Space image builds on the Container-Optimized OS (COS) image (LTS Linux distribution) developed and released by Google [2] and further improves the security of the image by disabling cloud-init and unnecessary packages and by enforcing the launch of only one container and integrity-protect on the on-disk resources used by the Confidential Space container. The COS image has a smaller attack surface than average virtual machine images as it has no package manager, less software installed and no third party kernel module or driver installation allowed. It comes with security defaults (sysctl disabled ptrace, ssh, GCE guest agent, cloud-init and unprivileged BPF) and includes a locked down firewall. The COS image is based on Chromium OS, an immutable root file system with verified boot, stateless configuration files in /etc folder (that can be set at runtime, but doesn't persist across reboots), a security-hardened kernel (Integrity Measurement Architecture (IMA), Audit, Kernel Page Table Isolation (KPTI), and some Linux Security Modules (LSMs) from Chromium OS. The number of Confidential Space images and kernels that can be used in the Confidential Space platform is limited and released by Google.

The Confidential Space image is running on a GCE Shielded VM by default, booting with UEFI and vTPM, but Secure Boot is not enabled by default, so the operator needs to enable it to pass the attestation verification. COS also has its own Secure Boot hierarchy that signs the loaded binaries in the boot chain that will be in the quote and are checked.

The COS file system can be broken down into three types:

- root partition (read-only mount)
- stateful partition (writeable, encrypted, persist reboot)
- stateless partition (writable, but not persist reboot).

The stateful partition is configured via the kernel command line which is measured and checked. The Confidential Space image goes one step further and encrypts the stateful partition with integrity.

2. Container-Optimized OS documentation: https://cloud.google.com/container-optimized-os/docs

There is a separate feature called integrity monitoring and the outcome of the confidential VM policy check (successful and failed) are logged in Cloud Logging service and can be viewed in the Logs Explorer (earlyBootReportEvent and lateBootReportEvent log entries with policyEvaluationPassed value). The service account that is used with the debug Confidential Space image would have Logs/Writer permission for creating logs.

Base Values

Name	Description	Value
ConfidentialTech	VM is running AMD SEV	AMD_SEV
GCEFirmwareVersion	Shielded features	1
SecureBootEnabled	Secure Boot is enabled	true
Policy's PermittedKeys	Permitted Keys	database
Policy's ForbiddenKeys/ Hashes	Forbidden Keys	database

Golden Values

Name	Desription	Value
Kernel Binary	Kernel binary hash value	sha256
grub.cfg		sha256
GRUB commands	List of well-known and allowed commands	list
Kernel command line	dm-verity root hash, systemd masked services, cloud-init referencing OEM partition	string
rootfs verity hash	root file system hash value	sha256
OEM verity hash	OEM partition hash value	sha256

The boot measurement log contains all the firmware and boot events and is using the PCR0-9 to parse and replay the event log.

The successful attestation is parsed into MachineState and will contain GrubState (grub.cfg and command lines), KernelState (kernel command line and kernel hash) and GceInstanceInfo that ties the actual instance for remote attestation.

Analysis

The Confidential Space image builds on the COS image that already has limited or no attack surface and removes any remaining applications or services that are unnecessary for running the only container. The secure boot is preventing any modification in the boot process and immediately terminates the booting if any of the binary signatures are incorrect. The measurement of the boot process is checked by the remote attestation verifier service and the WIP policy. The vTPM keys are not exportable, the PCRs are overwritten with new hash values instead of modified and if one hash changes, the following hashes will change as well and the attestation will fail. The kernel command line and GRUB commands are measured, preventing any modifications. Accessing the virtual machine by the operator is not possible, because of the hardening and identity permissions. Data would only persist across boots if the operator could access the

container and its stateful partition, which is encrypted, because the root file system is read-only and the remaining type of partition is stateless.

Container Launch Measurements

The Confidential Space container will measure the container image reference, image digest, image id and environment variables, running Confidential Space image version, and Confidential Space container launcher running. The image author sets up launch policies based on the image reference, id and digest that are known hash values and which environment variables can be overwritten by adding image labels to the image.

The COS Event Log contains the following: image_reference, image_digest, image_id, args, env_var, restart_policy, cos_version, launcher_version and is using the PCR13 to parse and replay the event log. Workload Author can enforce launcher policies (image labels) for the operator as the launcher policy is part of the container image in the form of image labels (annotations). As an additional layer, if the workload author misconfigures the launch policy, the WIP Security Service Token could still deny the OIDC token when it compares these values against the WIP configuration and Common Expression Language (CEL) attribute mapping.

Analysis

The Confidential Space container launcher is only able to run one container that contains the workload binary, and it is the only way for an operator to interact with the VM, as all other methods are locked down (they provide avenues for operator tampering). For example, accessing the container via SSH is disabled. The container launcher checks for the required metadata service (MDS) variables (e.g., tee-image-reference) to launch and configure the workload. If these are not set or these values do not match with the workload author launch policy, the container launcher will not run and logged. These values are also compared later against the WIP configuration.

The launcher log can be found under the confidential-space-launcher log name in the Cloud Logging service. The log record includes information such as image digest, user account, error message, date and time.

Remote Attestation Flow

The software-based remote attestation will only ensure the integrity of the binaries, but not the execution and run-time as the behavior of a device at run-time can be changed without modifying the binaries. To prevent misuse, the operator is limited by not being able to access the container and the workload program that the workload author developed. In addition, the quote generation uses a server-side generated nonce (which cannot be used again) and a periodic token exchange with a limited lifetime of 1 hour. The attestation is done via secure channel and is encrypted at rest (AES-GCM and vTPM).

The agent connects to the remote attestation verifier service via HTTPS and requests a 128-bit nonce, generated and stored server-side, that expires after 1 hour. After receiving the nonce, the agent creates an attestation quote including the hashed values of the 14 PCRs, the server-generated nonce, some GCE instance information (GCEInstanceInfo), the Boot Log, COS Event Log, AKCert and signs it with the Attestation Key (AK). There is one trusted AK (as they are derived from the vTPM Endorsement Seed) per VM that lives in the vTPM (unexportable, signed by Google at VM creation time) and it is not rotated or updated for a given VM. This AK is trusted via the AKCert that is generated and verified on the server side. The Google AK/EK signing roots on the attestation verifier service is trusted.

The agent will only receive the OIDC token from the remote attestation verifier service if the TLS 1.2 connection is successful, the quote is correctly signed, the nonce is correct

and the hashes of the measured values are also matching with the Boot Log and COS Event Logs (that uses a different PCR). The agent will receive the corresponding STS token from the WIP if all the measured values compared to the base and golden values are matched. The WIP Security Token Service also checks that the claim is the same as it is in the WIP configuration. If the agent cannot communicate with the remote attestation server or any of the measured value checks fail, there will be no OIDC token generated. If an attempt was made to fake the remote attestation server, the connection would not happen as the TLS verification will fail.

The TLS connection provides a secure transport channel. The certificates and keys are the foundation for proving the integrity of the quote and measurements. The vTPM provides secure, unexportable key storage and enforces attestation key usage on TPM-generated data. The measurements securely stored in vTPM PCRs are not directly writable, instead only supporting the Extend operation. The server-side generated limited lifetime nonce provides freshness, proves that the claims are recent and prevents replay attacks. The nonce is stored without the quote at server-side for 1 hour and then removed. There is an approximately 10-queries-per-second (QPS) rate limitation provided by Google Cloud Control Frontend (CCFE) on generating a nonce and sending a quote for attestation.

Quote Validation

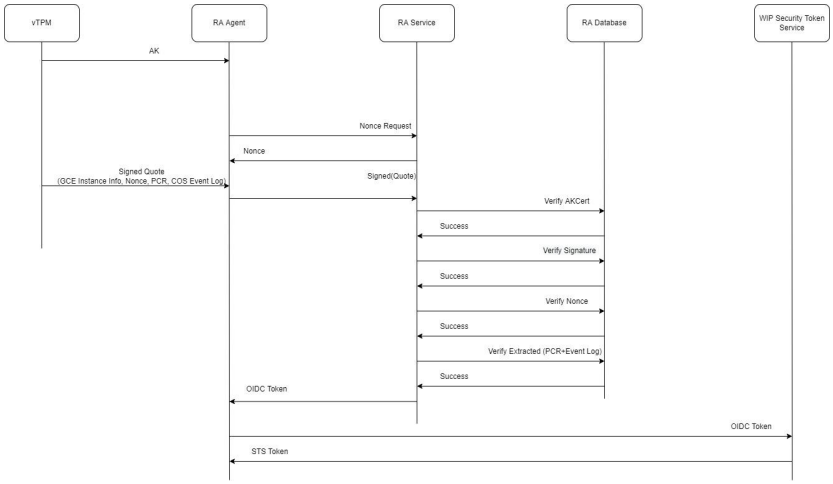

Figure 2: Quote validation and OIDC Token acquiring

The quote validation process will follow the steps below:

- The RA service validates the certificate chain on the client vTPM EK cert to ensure that it came from a Google signed vTPM.
- The RA service validates if the quote is signed by the AK and the correct nonce is used (preventing replay attacks by the client).
- The PCR values are matched against the expected values at server-side (base and golden values).
- The RA service validates that the vTPM event log produces the same set of PCR values that are in the quote (for both the Confidential Space image and container).
- If all of the checks are correct then an OIDC token will be sent to the RA agent.
- The OIDC token will be exchanged for an STS access token. All conditions for gaining access to the target resource are expressed and enforced at the WIP level (the conditions are defined for the WIP and enforced by the WIP Security Token Service).
- If all conditions are met, then a new credential is issued impersonating a configured service account with a 1-hour lifetime.

- The documentation didn't mention rate limitation on nonce generation or quote validation and, based on the answers from Google engineers, there are no rate limitations on the API call other than Google Anti-DDoS protection.

Analysis

The validation process prevents replay attacks with the short-lived server-side generated nonce and signing, plus the GCEInstanceInfo value ties the actual instance to remote attestation. As the nonce has 128-bit entropy and is protected by Google Anti-DDoS service, there is no notable risk in generating collisions, plus the challenge cannot be reused after it is expired.

The vTPM and signing certification is trusted by Google (root of trust) and the Quote signing key is verified during the process. Forging a key would not be verified by the remote attestation service as it is not signed by Google. If an attacker attempts to fake a remote attestation verifier server, the tokens issued by such an attacker-controlled server would not be signed with the correct signing key and cannot be used for Google services. The communication between services is going through a secure channel (HTTPS), providing confidentiality and additional integrity for the quote.

There is a separate feature called integrity monitoring and the outcomes of the confidential VM policy check (successful and failed) are logged in the Cloud Logging service and can be viewed in the Logs Explorer (earlyBootReportEvent and lateBootReportEvent log entries with policyEvaluationPassed value). The launcher log can be found under the confidential-space-launcher log name in the Cloud Logging service. The service account that is used with the debug Confidential Space VM would have additional Logs/Writer permission for creating logs.