Shielded VM

Shielded VM offers verifiable integrity of your Compute Engine VM instances, so you can be confident your instances haven't been compromised by boot- or kernel-level malware (https://en.wikipedia.org/wiki/Malware) or rootkits (https://en.wikipedia.org/wiki/Rootkit). Shielded VM's verifiable integrity is achieved through the use of Secure Boot (/security/shielded-cloud/shielded-vm#secure-boot), virtual trusted platform module (vTPM) (/security/shielded-cloud/shielded-vm#vtpm)-enabled Measured Boot (/security/shielded-cloud/shielded-vm#measured-boot), and integrity monitoring (/security/shielded-cloud/shielded-vm#integrity-monitoring).

Shielded VM is the first offering in the Shielded Cloud initiative. The Shielded Cloud initiative is meant to provide an even more secure foundation for all of Google Cloud (GCP) by providing verifiable integrity and offering features, like vTPM shielding or sealing (https://en.wikipedia.org/wiki/Trusted_Computing#SEALED-STORAGE), that help prevent data exfiltration (/security/data-loss-prevention/preventing-data-exfiltration).

This topic describes Shielded VM, for information about how to modify Shielded VM options, see Modifying Shielded VM options (/compute/docs/instances/modifying-shielded-vm).

Secure Boot

Secure Boot helps ensure that the system only runs authentic software by verifying the digital signature of all boot components, and halting the boot process if signature verification fails.

Shielded VM instances run firmware which is signed and verified using Google's Certificate Authority, ensuring that the instance's firmware is unmodified and establishing the root of trust (https://www.uefi.org/sites/default/files/resources/UEFI%20RoT%20white%20paper_Final%208%208%2016%20%28003%29.pdf) for Secure Boot. The Unified Extensible Firmware Interface (UEFI) 2.3.1 firmware (https://www.uefi.org/sites/default/files/resources/UEFI_Secure_Boot_in_Modern_Computer_Security_Solutions_2013.pdf), securely manages the certificates that contain the keys used by the software manufacturers to sign the system firmware, the system boot loader, and any binaries they load. Shielded VM instances use UEFI firmware.

On each boot, the UEFI firmware verifies the digital signature of each boot component against the secure store of approved keys. Any boot component that isn't properly signed, or isn't signed at all, isn't allowed to run.

If this occurs, the VM instance's serial console (/compute/docs/instances/interacting-with-serial-console) log will have an entry containing the strings **UEFI: Failed to load image** and **Status: Security Violation**, along with a description of the boot option that failed. To troubleshoot the failure, disable Secure Boot by using the instructions in Modifying Shielded VM Options (/compute/docs/instances/modifying-shielded-vm) so that you can boot the VM instance, diagnose and resolve the issue, then re-enable Secure Boot.

Virtual Trusted Platform Module (vTPM)

A vTPM is a virtualized trusted platform module (https://trustedcomputinggroup.org/trusted-platform-module-tpm-summary/), which is a specialized computer chip you can use to protect objects, like keys and certificates, that you use to authenticate access to your system. The Shielded VM vTPM is fully compatible with the Trusted Computing Group (TPM) library specification 2.0 (https://trustedcomputinggroup.org/tpm-library-specification/) and uses BoringSSL (https://boringssl.googlesource.com/boringssl/), which is FIPS 140-2 L1 (https://csrc.nist.gov/publications/detail/fips/140/2/final) validated.

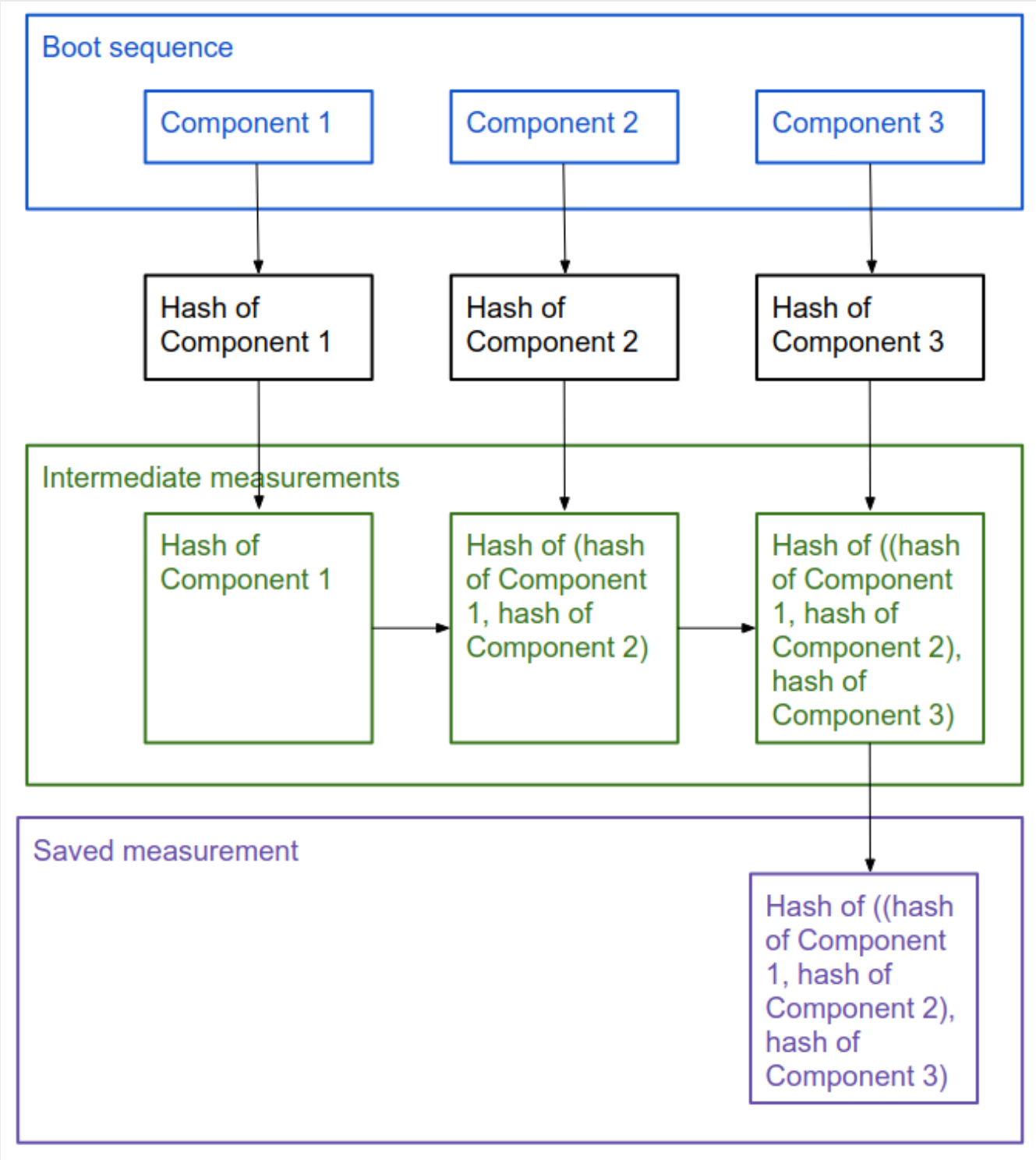
The Shielded VM vTPM enables Measured Boot (/security/shielded-cloud/shielded-vm#measured-boot) by performing the measurements needed to create a known good boot baseline, called the *integrity policy baseline*. The integrity policy baseline is used for comparison with measurements from subsequent VM boots to determine if anything has changed.

You can also use the vTPM to protect secrets through shielding or sealing (https://en.wikipedia.org/wiki/Trusted_Computing#SEALED-STORAGE). See the Go-TPM (https://github.com/google/go-tpm) project on GitHub for Go language examples that illustrate how to use a vTPM for this purpose.

Measured Boot

During Measured Boot, a hash of each component (for example, the firmware, bootloader, or kernel) is created as the component is loaded, and that hash is then concatenated and

rehashed with the hashes of any components that have already been loaded, as illustrated below:



This information identifies both the components that were loaded, and their load order.

The first time you boot a VM instance, Measured Boot creates the integrity policy baseline from the first set of these measurements, and securely stores this data. Each time the VM instance boots after that, these measurements are taken again, and stored in secure memory until the next reboot. Having these two sets of measurements enables integrity monitoring (/security/shielded-cloud/shielded-vm#integrity-monitoring), which you can use to determine if there have been changes to a VM instance's boot sequence.

Integrity monitoring

Integrity monitoring helps you understand and make decisions about the state of your VM instances.

Integrity monitoring relies on the measurements created by Measured Boot, which use platform configuration registers (https://link.springer.com/chapter/10.1007/978-1-4302-6584-9_12) (PCRs) to store information about the components and component load order of both the integrity policy baseline (a known good boot sequence), and the most recent boot sequence.

Integrity monitoring compares the most recent boot measurements to the integrity policy baseline and returns a pair of pass/fail results depending on whether they match or not, one for the early boot sequence and one for the late boot sequence. Early boot is the boot sequence from the start of the UEFI firmware until it passes control to the bootloader. Late boot is the boot sequence from the bootloader until it passes control to the operating system kernel. If either part of the most recent boot sequence doesn't match the baseline, you get an integrity validation failure.

If the failure is expected, for example if you applied a system update on that VM instance, you should update the integrity policy baseline. Updating the integrity policy baseline sets the baseline to the measurements captured from the most recent boot sequence. If it is not expected, you should stop that VM instance and investigate the reason for the failure.

You can view integrity reports in Cloud Monitoring, and set alerts on integrity failures. You can review the details of integrity monitoring results in Cloud Logging. For more information, see Monitoring Integrity on Shielded VM Instances (/compute/docs/instances/integrity-monitoring).

Integrity monitoring events

Shielded VM creates log entries for the following types of events:

- `clearTPMEvent`: Identifies if the vTPM has been cleared, which deletes any secrets stored in it. This doesn't affect any aspect of Shielded VM, so you will only care about this if you use the vTPM to shield sensitive data as described in Virtual Trusted Platform Module (vTPM) (/security/shielded-cloud/shielded-vm#vtpm).
- `earlyBootReportEvent`: Identifies whether the early boot sequence integrity check passed, and provides details on the PCR values from the baseline and the most recent boot sequence that were compared to make that determination.
- `lateBootReportEvent`: Identifies whether the late boot sequence integrity check passed, and provides details on the PCR values from the baseline and the most recent boot sequence that were compared to make that determination.
- `setShieldedInstanceIntegrityPolicy`: Logged each time you update the integrity policy baseline.
- `shutdownEvent`: Logged each time the VM instance is stopped.
- `startupEvent`: Logged each time the VM instance is started. The interesting information in this event is the `bootCounter` value, which identifies how many times this instance has been restarted.
- `updateShieldedInstanceConfig`: Logged each time you enable or disable one of the Shielded VM options.

The typical event progression you see in the logs is `startupEvent`, `earlyBootReportEvent`, `lateBootReportEvent`, and eventually `shutdownEvent`, all with the same `bootCounter` value to identify them as describing the same VM instance boot sequence.

If you update the integrity policy baseline in response to an expected integrity failure on a VM instance, you will see additional `earlyBootReportEvent` and `lateBootReportEvent` events that describe the new integrity policy baseline measurements. The following example shows the expected sequence:

- `startupEvent`
- `earlyBootReportEvent` that compares original baseline to latest boot sequence (passes)
- `lateBootReportEvent` that compares original baseline to latest boot sequence (fails)
- `setShieldedInstanceIntegrityPolicy` when you update the integrity policy baseline (/compute/docs/instances/integrity-monitoring#updating-baseline), which sets the baseline to the measurements captured from the latest boot sequence

- `earlyBootReportEvent` that compares new baseline to latest boot sequence (passes)
- `lateBootReportEvent` that compares new baseline to latest boot sequence (passes)

Cloud Identity and Access Management authorization

Shielded VM uses Cloud IAM for authorization.

Shielded VM operations use the following Compute Engine permissions:

- `compute.instances.updateShieldedInstanceConfig`: Allows the user to change the Shielded VM options on a VM instance.
- `compute.instances.setShieldedInstanceIntegrityPolicy`: Allows the user to update the integrity policy baseline on a VM instance.
- `compute.instances.getShieldedInstanceIdentity`: Allows the user to retrieve endorsement key information from the vTPM.

Shielded VM permissions are granted to the following Compute Engine roles:

- `roles/compute.instanceAdmin.v1`
- `roles/compute.securityAdmin`

You can also grant Shielded VM permissions to custom roles (/iam/docs/creating-custom-roles).

Organization policy constraints for Shielded VM

You can set the `constraints/compute.requireShieldedVm` organization policy constraint (/resource-manager/docs/organization-policy/org-policy-constraints) to `True` to require that Compute Engine VM instances created in your organization be Shielded VM instances.

Learn how to set the `constraints/compute.requireShieldedVm` constraint in Using boolean constraints in organization policy (/resource-manager/docs/organization-policy/using-constraints#boolean-constraint). You must be an organization policy administrator

(/resource-manager/docs/organization-policy/using-constraints#add-org-policy-admin) to set a constraint.

What's next

- Learn about retrieving the endorsement key from the vTPM (/security/shielded-cloud/retrieving-endorsement-key).
- Learn about one approach to automating responses to integrity monitoring events (/security/shielded-cloud/automating-responses-integrity-failures).

Last updated 2020-04-09.