Project Submission

General Information

1.1. Name of Project

COCONUT-SVSM

1.2. Project Description

The COCONUT-SVSM is an implementation of a Secure VM Service Module for confidential computing virtual machines (CVMs). It runs within the TCB of the CVM and aims to support three operational modes:

1. **Service Module Mode:** In this mode, the COCONUT-SVSM runs alongside an operating system that is fully enlightened to CVM operation and could run natively as a CVM without additional support. However, by running at an elevated privilege level compared to the rest of the CVM operating system, the COCONUT-SVSM can provide additional services that require privilege separation for security and robustness purposes (such as an implementation of a virtual TPM), services that are required by the underlying architecture to run outside of the context of the guest operating system (such as emulation of an APIC for secure delivery of virtual interrupts), or services that enhance the operation of an existing operating system by executing outside of the resource domain of the OS (such as emulation of a UEFI variable store).
2. **Paravisor Mode:** In this mode, the COCONUT-SVSM operates as a trusted Virtual Machine Manager inside the trust boundary of the CVM, and handles virtualization services required by the guest operating system that would otherwise have to be handled by a host VMM, in addition to implementing all of the services available in Service Module mode. Such operation can provide transparent emulation of devices or other architectural primitives that would be expected of an operating system that has not been enlightened to execute in a CVM context, minimizing the code changes required to make an operating system function in a CVM environment and broadening the scope of workloads that can execute in a CVM.
3. **Service VM Mode:** In this mode, the COCONUT-SVSM runs in a CVM without an accompanying guest operating system, creating a lightweight environment for hosting services that may need to be made available to other CVMs on the system. Such an environment is a CVM in and of itself, but with a dramatically reduced resource footprint compared to a full operating system.

With its ability to provide secure services, the COCONUT-SVSM aims to become a platform to execute host hypervisor services within the trusted context of the CVM and reducing the interface between the hypervisor and the CVM OS. It acts as a proxy between the hypervisor and the CVM OS to reduce the need to harden the CVM OS from malicious hypervisor input.

The COCONUT-SVSM currently runs in AMD SEV-SNP environments. Long-term, the objective is to support other CVM architectures, such as Intel TDX or ARM CCA, though the Service Module and Paravisor modes will only be available on architectures that support multiple privilege context within a single CVM boundary (such as Intel TDX Partitioning or AMD SEV-SNP VM Privilege Levels).

The project was started in early 2022 to implement a Secure VM Service Module for AMD platforms. It was first publicly announced at the OC3 conference on March 15th, 2023. Since then it attracted numerous contributors from a variety of entities and doubled its size as measured by lines of code. With the code its vision also grew and the COCONUT-SVSM now aims to become a platform for providing secure services to CVM OSes, supporting a broad variety of hardware platforms, host hypervisors and services.

1.3. How does this project align with the Consortium's Mission Statement

The project helps the adoption of confidential computing by providing a platform for executing hypervisor services within the trusted context of a CVM. With services provided from a trusted environment the attack surface of the CVM OS towards the host hypervisor is reduced, making the overall CVM more secure.

Also services providing emulations for devices carrying security sensitive state (e.g. RTMR emulations or a Trusted Platform Module) can only be securely provided to CVM OSes by running them in a TEE. These services are needed to provide runtime attestation capabilities on some TEE platforms, where COCONUT-SVSM drives further adoption of confidential computing by making the TEE software stack fully attestable.

With support for different operational modes the COCONUT-SVSM is flexible enough to be used with unmodified and enlightened CVM OSes or TCB architectures without built-in memory isolation. Especially in paravisor mode running an unenlightened OS the COCONUT-SVSM can broaden the scope of possible workloads that can run in a CVM environment.

1.4. Project website URL

N/A. The website will at some point appear at https://coconut-svsm.dev/

1.5. Social media accounts

N/A

Legal Information

2.1. Project Logo

N/A - No project logo yet.

2.2. Project license.

COCONUT-SVSM is dual-licensed under MIT or APACHE-2.0.

2.3. Existing financial sponsorship.

No existing financial sponsorship except the time its contributors are allowed to spend on the project by their employers.

2.4. Trademark status.

COCONUT-SVSM is not trademarked.

2.5. Proposed Technical Charter

Please see attached Technical Charter document.

Technical Information

3.1. High level assessment of project synergy with existing projects under the CCC

We currently assess two possible synergies of COCONUT-SVSM with existing projects under the CCC:

1. The COCONUT-SVSM project in the Service VM model can be used as a building block to port existing enclave-based projects under the CCC to virtualisation-based TCB hardware architectures. This broadens the scope of these projects and helps them to support a more diverse set of TCB hardware.
2. In any operational model the project can provide additional trusted services to enhance the attestation capabilities of trusted execution environments. This has possible synergies with projects under CCC which are targeted to attestation.

3.2. Describe the Trusted Computing Base (TCB) of the project.

COCONUT-SVSM is designed for virtualisation-based TCB hardware architectures and currently runs in AMD SEV-SNP environments. Support for other TCBs, e.g. Intel TDX, is planned.

3.3. Project Code of Conduct URL

https://www.contributor-covenant.org/version/2/1/code_of_conduct/

3.4. Source control URL

https://github.com/coconut-svsm/svsm

3.5. Issue tracker URL

https://github.com/coconut-svsm/svsm/issues

3.6. External dependencies

The COCONUT-SVSM project depends on a number of external Rust crates. Please see the table below.

Crate	License	Type
Bitflags	MIT OR Apache-2.0	Runtime
Gdbstub	MIT OR Apache-2.0	Optional Runtime
Gdbstub_arch	MIT OR Apache-2.0	Optional Runtime
Intrusive_collections	Apache-2.0 OR MIT	Runtime
Log	MIT OR Apache-2.0	Runtime
PackIt	MIT	Runtime

Crate	License	Type
Aes-gcm	Apache-2.0 OR MIT	Runtime
Memoffset	MIT	Build-time test
libfuzzer-sys	Apache-2.0 OR MIT	Build-time test

These direct dependencies have their own sub-dependencies as well. Below is a full list of Rust crates used by COCONUT-SVSM grouped by their license. The list was generated with the cargo license command and slightly updated to better reflect the SVSM-provided crates:

License	Crates
(MIT OR Apache-2.0) AND Unicode-DFS-2016	unicode-ident
Apache-2.0 OR MIT	aead, aes, aes-gcm, autocfg, bitflags, cfg-if, cipher, cpufeatures, crypto-common, ctr, ghash, inout, intrusive-collections, log, opaque-debug, polyval, proc-macro2, quote, syn, typenum, universal-hash, version_check
BSD-2-Clause	zerocopy, zerocopy-derive
BSD-3-Clause	subtle
MIT	generic-array, memoffset
MIT OR Unlicense	byteorder
SVSM managed (Apache-2.0 OR MIT)	packit, svsm, test

3.7. Standards implemented by the project

COCONUT-SVSM currently implements the following standards:
- GHCB Specification: https://www.amd.com/content/dam/amd/en/documents/epyc-technical-docs/specifications/56421.pdf
- SVSM Specification https://www.amd.com/content/dam/amd/en/documents/epyc-technical-docs/specifications/58019.pdf

3.8. Release methodology and mechanics

The project has not published releases yet and there are no processes defined for making a release.

3.9. Names of initial committers

- Carlos Bilbao (AMD)
- Claudio Carvalho (IBM)
- Tom Dohrmann (Hobbyist)
- Stefano Garzarella (Red Hat)
- Roy Hopkins (SUSE)
- Vasant Karasulli (SUSE)
- Jon Lange (Microsoft)

- Tom Lendacky (AMD)
- Thomas Leroy (SUSE)
- Carlos López (SUSE)
- Dan Middleton (Intel)
- Dov Murik (IBM)
- Cole Robinson (Red Hat)
- Jörg Rödel (SUSE)
- Nicolai Stange (SUSE)
- Oliver Steffen (Red Hat)

3.10. List of project's official communication channels

The COCONUT-SVSM community stays in contact via

- Project mailing list: svsm-devel@coconut-svsm.dev
- Weekly development calls every Wednesday at 9am PST

There is no Slack or similar communication channel yet.

3.11. Project Security Response Policy

Security bugs can be reported via security@coconut-svsm.dev. Besides that there is no process defined yet. We will define a security incident response process before making the first release.

3.12. Preferred maturity level

COCONUT-SVSM aims for the **Incubation** stage.

3.13. Any additional information the TAC and Board should take into consideration when reviewing your proposal.

N/A