Nitro Enclaves

Overview

Nitro Enclaves (NE) is a new Amazon Elastic Compute Cloud (EC2) capability that allows customers to carve out isolated compute environments within EC2 instances [1].

For example, an application that processes sensitive data and runs in a VM, can be separated from other applications running in the same VM. This application then runs in a separate VM than the primary VM, namely an enclave. It runs alongside the VM that spawned it. This setup matches low latency applications needs.

The current supported architectures for the NE kernel driver, available in the upstream Linux kernel, are x86 and ARM64.

The resources that are allocated for the enclave, such as memory and CPUs, are carved out of the primary VM. Each enclave is mapped to a process running in the primary VM, that communicates with the NE kernel driver via an ioctl interface.

In this sense, there are two components:

1. An enclave abstraction process - a user space process running in the primary VM guest that uses the provided ioctl interface of the NE driver to spawn an enclave VM (that's 2 below).

There is a NE emulated PCI device exposed to the primary VM. The driver for this new PCI device is included in the NE driver.

The ioctl logic is mapped to PCI device commands e.g. the NE_START_ENCLAVE ioctl maps to an enclave start PCI command. The PCI device commands are then translated into actions taken on the hypervisor side; that's the Nitro hypervisor running on the host where the primary VM is running. The Nitro hypervisor is based on core KVM technology.

2. The enclave itself - a VM running on the same host as the primary VM that spawned it. Memory and CPUs are carved out of the primary VM and are dedicated for the enclave VM. An enclave does not have persistent storage attached.

The memory regions carved out of the primary VM and given to an enclave need to be aligned 2 MiB / 1 GiB physically contiguous memory regions (or multiple of this size e.g. 8 MiB). The memory can be allocated e.g. by using hugetlbfs from user space [2][3][7]. The memory size for an enclave needs to be at least 64 MiB. The enclave memory and CPUs need to be from the same NUMA node.

An enclave runs on dedicated cores. CPU 0 and its CPU siblings need to remain available for the primary VM. A CPU pool has to be set for NE purposes by an user with admin capability. See the cpu list section from the kernel documentation [4] for how a CPU pool format looks.

An enclave communicates with the primary VM via a local communication channel, using virtio-vsock [5]. The primary VM has virtio-pci vsock emulated device, while the enclave VM has a virtio-mmio vsock emulated device. The vsock device uses eventfd for signaling. The enclave VM sees the usual interfaces - local APIC and IOAPIC - to get interrupts from virtio-vsock device. The virtio-mmio device is placed in memory below the typical 4 GiB.

The application that runs in the enclave needs to be packaged in an enclave image together with the OS (e.g. kernel, ramdisk, init) that will run in the enclave VM. The enclave VM has its own kernel and follows the standard Linux boot protocol [6][8].

The kernel bzImage, the kernel command line, the ramdisk(s) are part of the Enclave Image Format (EIF); plus an EIF header including metadata such as magic number, eif version, image size and CRC.

Hash values are computed for the entire enclave image (EIF), the kernel and ramdisk(s). That's used, for example, to check that the enclave image that is loaded in the enclave VM is the one that was intended to be run.

These crypto measurements are included in a signed attestation document generated by the Nitro Hypervisor and further used to prove the identity of the enclave; KMS is an example of service that NE is integrated with and that checks the attestation doc.

The enclave image (EIF) is loaded in the enclave memory at offset 8 MiB. The init process in the enclave connects to the vsock CID of the primary VM and a predefined port - 9000 - to send a heartbeat value - 0xb7. This mechanism is used to check in the primary VM that the enclave has booted. The CID of the primary VM is 3.

If the enclave VM crashes or gracefully exits, an interrupt event is received by the NE driver. This event is sent further to the user space enclave process running in the primary VM via a poll notification mechanism. Then the user space enclave process can exit.

[1] https://aws.amazon.com/ec2/nitro/nitro-enclaves/ [2] https://www.kernel.org/doc/html/latest/admin-guide/mm/hugetlbpage.html [3] https://lwn.net/Articles/807108/ [4] https://www.kernel.org/doc/html/latest/admin-guide/kernel-parameters.html [5] https://man7.org/linux/man-pages/man7/vsock.7.html [6] https://www.kernel.org/doc/html/latest/x86/boot.html [7] https://www.kernel.org/doc/html/latest/arm64/hugetlbpage.html [8] https://www.kernel.org/doc/html/latest/arm64/booting.html