December 2019

PROCESSING AND DELIVERING SECURITY SIGNALS FROM GUESTS TO HOSTS IN VIRTUALIZED ENVIRONMENTS

Michael Halcrow

Thomas Garnier

Brandon Baker

Eric Biggers

Maya Kaczorowski

See next page for additional authors

Inventor(s)
Michael Halcrow, Thomas Garnier, Brandon Baker, Eric Biggers, Maya Kaczorowski, Hovik Manucharyan, and Konstantin Stepanyuk

PROCESSING AND DELIVERING SECURITY SIGNALS FROM GUESTS TO HOSTS IN VIRTUALIZED ENVIRONMENTS

ABSTRACT

This disclosure describes techniques that leverage memory organization in virtual machines and their hosts to emplace code that protects against malware. Malware detection instrumentation is emplaced in guest kernel space, which is relatively privileged and better protected than other guest memory spaces. Malware behavioral analysis logic, which classifies a guest process as benign or malign, is emplaced in host ring 3 space, to take advantage of the virtualization boundary.

Even if unaffected by the attack, the protected kernel may still not be able to quickly communicate knowledge of the attack to the malware behavioral analysis logic, which resides in the host. This is because such communication normally travels through guest userspace, which may be compromised. This disclosure further describes techniques that enable the guest kernel to communicate sensitive information to the host while bypassing guest userspace, e.g., by using a virtio-vsock channel.

KEYWORDS

- Cloud security
- Container security
- Virtual machine
- Virtualization boundary
- Malware detection
- Intrusion detection
- Code instrumentation

- Malware behavioral analysis

BACKGROUND

In remediating malware attacks on virtual machines, time is of essence. By detecting and reporting behavior that is indicative of a security attack, the virtual machine platform, e.g., a cloud computing provider, can help incident response teams identify and remediate attacks as they happen. Protection against malware is achieved using two coordinating modules, e.g.,

- malware detection instrumentation, that observes workload behavior, and
- malware behavioral analysis logic, that analyzes the captured behavior to identify instances of malicious behavior and to classify a workload as benign or malign.

Malware detection instrumentation uses deep knowledge of the computational environment in order to accurately measure workload behavior. Malware behavioral analysis logic can take the form of machine learning models with attendant training.

A malware attack can have serious consequences for the owner of the workload running on the virtual machine, e.g., disruption of computational tasks, compromise of sensitive data, abuse of computational resources, etc. Malware can target and compromise anything in the vicinity of the workload. Components that are subject to attack and compromise can include the malware detection instrumentation and the malware behavioral analysis logic itself.

Currently, cloud providers do not include platform-native intrusion detection techniques. Therefore, cloud computing users have to rely on the use of third-party technologies. However, third-party intrusion detection systems lack the ability to deploy a guest/host privilege separate framework, unlike managed cloud providers.

DESCRIPTION

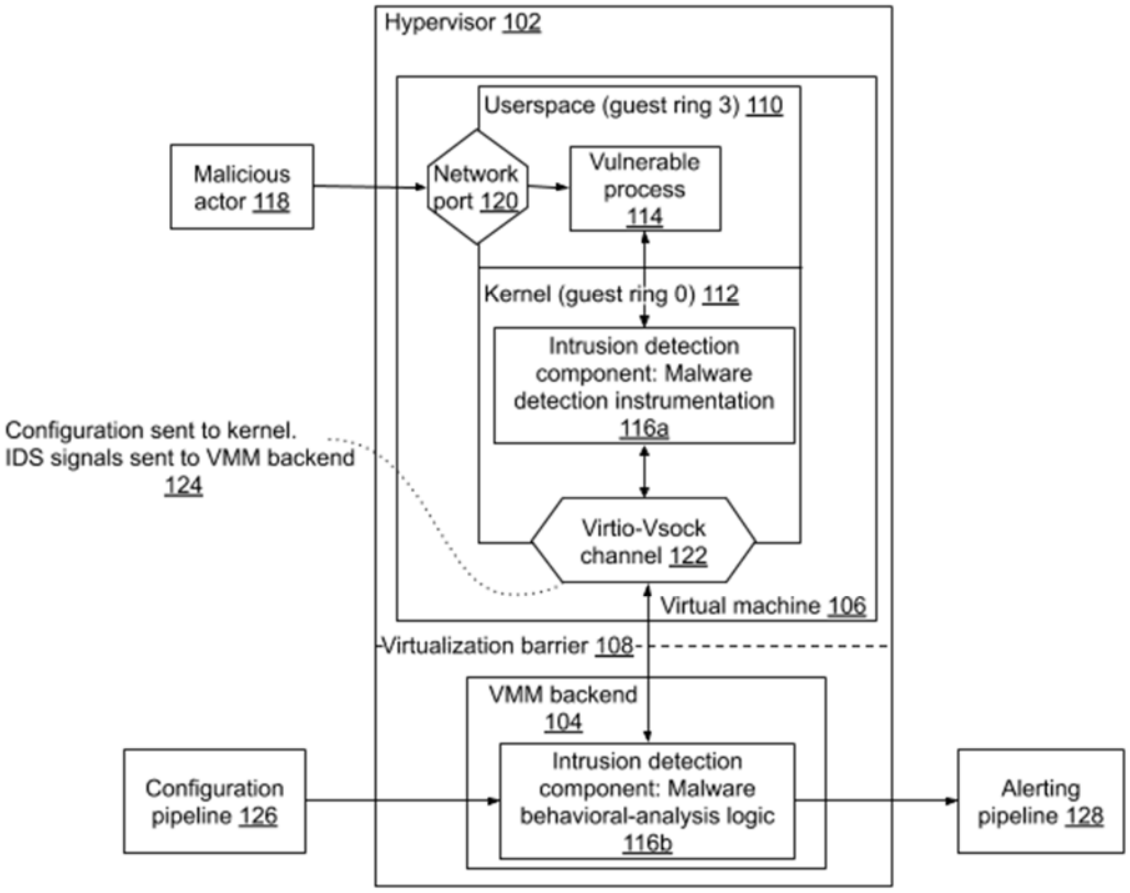

Fig. 1: Processing and delivering security signals from guests to hosts in virtualized environments

Fig. 1 illustrates processing and delivering security signals from guests to hosts in a virtualized environment, per techniques of this disclosure. A hypervisor or virtual machine monitor (VMM) 102 runs on a host machine, and enables the spawning of one or more virtual machines or guests 106. The hypervisor includes a VMM backend 104 which runs on the host and manages the virtual machines. A virtualization barrier 108 separates the virtual machines from the VMM backend.

Memory in a virtual machine is organized into spaces of differing privilege. For example, kernel space 112, also known as guest ring 0, has the highest privilege and is highly protected;

userspace 110, also known as guest ring 3, has a lower privilege and has lower protection; etc. Similarly, the host memory is divided into host ring 0 (highest privilege and protection), host ring 3 (lowest privilege and protection), etc. The virtual machine includes one or more network ports 120.

In an example threat scenario, a malicious actor 118 attempts to corrupt the virtual machine by transporting malware via a network port. The VM workload, e.g., software applications executed by a cloud computing customer, runs in userspace (guest ring 3). The workload often includes software that has security vulnerabilities that can be exploited. An exploit can lead to arbitrary and malicious operations in userspace. The workload is therefore referred to as a vulnerable process 114.

The techniques of this disclosure leverage the relatively high protection of the guest kernel-space, further protect the guest kernel-space by locking it down, and emplace within it an intrusion detection component (116a). The intrusion detection component thus works behind a security boundary, the userspace-kernel boundary, that can be exploited only if there is also a vulnerability in kernel space, which is relatively less likely.

The intrusion detection component comprises instrumentation that observes workload behavior and captures workloads and signals that show indications of corruption by malware. To optimize capture of workload behavior, the instrumentation can obtain information from deep within the workload, a task facilitated by its placement within the guest kernel. The instrumentation is by design relatively simple, and by emplacing it in the kernel, there is much less risk to exposing the behavior-capturing instrumentation to potential attackers. In this manner, the techniques herein provide the behavior capture instrumentation a measure of protection against an attacker.

Signals generated by the malware detection instrumentation are fed into malware behavioral analysis logic that classifies the behavior as benign or malicious. This logic, that analyzes the captured behavior, is relatively sophisticated, sensitive, and can take the form of machine learning models with attendant training processes. Placing the analysis logic in guest ring 0 risks exposing the logic to an attacker that can alter the behavioral signals emitted by the malware to evade future detection.

The techniques described herein leverage the relatively hard and privileged virtualization boundary between the guest and the host to emplace the sensitive and high-value behavior classification logic in host ring 3, e.g., in an intrusion detection component (116b) within the VMM backed module. The guest/host virtualization boundary is significantly more robust than the ring-3/ring-0 boundary in the guest.

The placements described herein of the malware protection modules leverage the privileged separation afforded by the layers of the stack to achieve deep insight into the behavior of the workload while simultaneously protecting the attack detection components in the event that the guest environment is compromised. In this manner, an attacker that manages to exploit the guest kernel only gets access to malware-detection instrumentation in guest ring 0, e.g., all the attacker knows is the signals being monitored for malware detection. This doesn't give an attacker actionable information about how the malicious behavior can be changed to evade detection.

The malware behavioral analysis logic, which is located within an intrusion detection component in host ring 3 space, is configured by a configuration pipeline (126), and sends its results to an alerting pipeline (128), which triggers an incident response team as necessary.

Placing the behavior classification logic in the host rather than the guest implies that the evaluation of behavioral signals incurs a guest-host transition known as a VM exit. VM exits incur additional CPU overhead (and a corresponding reduction in the time available to the guest workload) when data in memory is copied between a guest address space and a host address space. Thus, even if a guest kernel is affected by the attack, it may not be able to quickly communicate the knowledge of the attack the host. Also, such communication normally travels through userspace which itself can be compromised.

The techniques of this disclosure address the overhead of copying between guest and host address spaces by leveraging a zero-copy communications channel as follows. A virtio-vsock channel (122), based on the open standard by the same name, is built to serve as a high queries-per-second (QPS), low-latency communication link between the OS of the virtual machine and the VMM backend. The virtio-vsock channel carries intrusion detection signals from the malware detection instrumentation to the VMM backend, and configuration signals from the VMM backend to the guest-kernel (124).

To reduce the frequency of VM exits, communication can occur in a batched manner, whereby a buffer in guest ring 0 collects behavioral signals, and the contents of that buffer are provided on a periodic basis to the host ring 3 detection logic via a VM exit. In this manner, sensitive information is communicated between guest ring 0 (where the malware detection instrumentation is situated) and host ring 3 (where the malware behavioral analysis logic is situated) in a timely manner, and without traversing guest-userspace that may be compromised.

An alternate technique of guest-host communication is to make a VM exit for each behavioral signal that the guest ring 0 instrumentation collects. An alternative solution to the

problem of protecting sensitive behavioral classification logic from attackers is to host the detection logic inside an enclave that leverages secure computing or encrypted features of CPUs.

CONCLUSION

This disclosure describes techniques that leverage memory organization in virtual machines and their hosts to emplace code that protects against malware. Malware detection instrumentation is emplaced in guest kernel space, which is relatively privileged and better protected than other guest memory spaces. Malware behavioral analysis logic, which classifies a guest process as benign or malign, is emplaced in host ring 3 space, to take advantage of the virtualization boundary.

Even if unaffected by the attack, the protected kernel may still not be able to quickly communicate knowledge of the attack to the malware behavioral analysis logic, which resides in the host. This is because such communication normally travels through guest userspace, which may be compromised. This disclosure further describes techniques that enable the guest kernel to communicate sensitive information to the host while bypassing guest userspace, e.g., by using a virtio-vsock channel.