Compute Cloud Security: Co-Resident Watermarking Schemes

Adam Bates, Masoud Valafar, Hannah Pruse, Joe Pletcher, and Kevin Butler
OSIRIS Lab, Computer & Information Science Department, University of Oregon

Cloud computing is transforming how computing services are provided and maintained. It allows businesses to only pay for the resources they need, and to quickly scale up services if demand suddenly rises. This saves companies from needing to purchase and maintain private datacenters, decreasing the up-front cost of doing business. The key to these services is virtualization, which allows physical machines to be multiplexed into many guest virtual machines.

However, virtualization leaves customers vulnerable to the actions of others allocated to the same physical machine. Research has demonstrated attacks that allow exploit this co-residency, but past work has focused on hypervisor software vulnerabilities that could eventually be patched. Our work demonstrates that even if other channels for establishing co-residency are removed, we can determine whether an adversarial guest is co-resident with a targeted server through observation of network traffic. We use concepts from network flow watermarking to determine whether we can establish co-residency based on the target’s traffic responses.

To test this attack, we created a miniature cloud with two co-resident virtual machines (VMs) on a local server. We connected a second Client workstation over an isolated local network. During each trial, the Client measured packet arrivals from the Server’s network flow in 500ms intervals. These measurements were sorted into two groupings: the periods of Flooder activity and inactivity. We then compared these two samples using the $\chi^2$ statistical test. If the $\chi^2$ test rejected the null hypothesis, then our samples represented two distinct distributions. This indicates that the Flooder altered the Server’s performance, and that the VMs were indeed co-resident.

Our first set of trials used the Xen hypervisor. We first tried the test on an isolated network, then over the Internet by connecting a Client at Georgetown University to our cloud in Eugene, Oregon. The graphs above show that the traffic rates are greatly reduced when the Flooder was active, which the $\chi^2$ test confirmed.

We repeated the attack using the VMWare ESXi hypervisor. Our results again show two distinct distributions of data. The increased variance between the distributions is an artifact created by ESXi’s fair CPU scheduling algorithm. From these results, we demonstrate the hypervisor independence of the co-resident watermarking scheme.

Beyond multi-tenancy, co-resident watermarking can be used as a general-purpose covert channel. It could be used to perform load measurement on victim instances, leaking information about a victim company’s business. It could also be used to infiltrate data or even as an anonymous control channel for botnet command. Based on our first trial, we attempted to transmit a 1024-bit secret key over the network flow channel. Our trials demonstrate a bitrate of 1.91bps, which we used to transmitted the key in 8 minutes and 57 seconds.

To further test the resiliency of the watermark, we increased the number of VMs on the Server. We observed the effect of the Flooder VM diminishes as the number of guests increased. However, the $\chi^2$ test still detected watermark signature with up to 6 total VM guests.

The adversary we consider in this work wishes to locate a victim running a web Server instance in the cloud. First, the adversary registers as a legitimate cloud user and launches many of her own instances. We refer to these as Flooders. With some probability, one of these Flooders will be launched co-resident to the target Server. From a Client computer outside the cloud, the adversary initiates a web session with the target Server, then tells each Flooder to take turns injecting activity onto the network. If a Flooder is able to inject delay into the Client-Server web session, then we know the Flooder and Server to be co-residents.

To further test this attack, we created a miniature cloud with two co-resident virtual machines (VMs) on a local server. We connected a second Client workstation over an isolated local network. During each trial, the Client measured packet arrivals from the Server’s network flow in 500ms intervals. These measurements were sorted into two groupings: the periods of Flooder activity and inactivity. We then compared these two samples using the $\chi^2$ statistical test. If the $\chi^2$ test rejected the null hypothesis, then our samples represented two distinct distributions. This indicates that the Flooder altered the Server’s performance, and that the VMs were indeed co-resident.

Our first set of trials used the Xen hypervisor. We first tried the test on an isolated network, then over the Internet by connecting a Client at Georgetown University to our cloud in Eugene, Oregon. The graphs above show that the traffic rates are greatly reduced when the Flooder was active, which the $\chi^2$ test confirmed.

We repeated the attack using the VMWare ESXi hypervisor. Our results again show two distinct distributions of data. The increased variance between the distributions is an artifact created by ESXi’s fair CPU scheduling algorithm. From these results, we demonstrate the hypervisor independence of the co-resident watermarking scheme.

For further information
Please contact Adam Bates (amb@cs.uoregon.edu).