Cloud computing security
Exploring information leakage in third-party compute clouds

Yangchun Fu

November 18th, 2011

Outline
1. Introduction
2. EC2 cloud cartography
3. Co-residence
4. side-channel cross-VM attack
5. Conclusions

Third-party cloud computing
On-demand computing outsourcing, share third-party infrastructure
- Amazon’s EC2
- Microsoft’s Azure
- Rackspace’s Mosso

Risk from cloud provider
- new trust relationship
  customers must trust their cloud provider to respect the privacy of their data and the integrity of their computations.

Threat from VM placement
- multiplex the virtual machines of different customers upon the same physical hardware.
- The adversary may be able to place the malicious instance on the same physical machine where the victim instance reside through legal process. Then lunch certain cross-VM attacks
Two steps

1. Placement: adversary arranging to place their malicious VM on the same physical machine as that of a target customer
2. Extraction: extract confidential information via a cross-VM attack.

Terminology

- Instance: a running OS image
- Co-resident: instances that are running on the same physical machine

Research questions

- Where is an instance located?
- How to determine if two instances are co-resident?
- How to launch an instance that will be co-resident with other user's instance?
- How to exploit cross-VM information leakage once co-resident?

Outline

1. Introduction
2. EC2 cloud cartography
3. Co-residence
4. Side-channel cross-VM attack
5. Conclusions

Amazon EC2 service

- Virtual machine monitor: Xen
- Run Linux, FreeBSD, OpenSolaris and Windows as guest
- A privileged virtual machine called Dom0 is configured to route packets for guest images and report itself as a hop in traceroutes.

More about Amazon EC2 service

- Configuration parameter:
  - Contain 3 availability zones with separate power and network connectivity
  - 5 instance types: m1.small, c1.medium, m1.large, m1.xlarge, c1.xlarge.
more about Amazon EC2 service

**Goal**
understand where potential target are located in the cloud and the instance creation parameters need to attempt establishing co-residence of an adversarial instance.

**Cloud Cartography**

**Hypothesis**
different availability zones likely to correspond to different internal IP address ranges and may be true for instance types as well. Thus mapping the use of the EC2 internal address allows an adversary to determine which IP address correspond to which creation parameters.

**Empirical measurement study**

**Two Data sets**
- launching a number of EC2 instance of varying types and surveying the resulting IP address assigned.
- enumerating public EC2-based web servers and translate public IP to internal IP

**Instance placement parameters**

**Experiment 1**
iteratively launching 20 instances for each of the 15 availability zone/instance type pairs.

![Graph 1](Image)

[Credit: Figure from Thomas's paper, CCS 2009]
different availability zones use different IP regions.

**Experiment 2**
launching 100 instances(20 of each type) in zone 3 for two account A and B.

![Graph 2](Image)

[Credit: Figure from Thomas's paper, CCS 2009]
The same instance type within the same zone use similar IP regions even for different accounts.
Enumerate public EC2-based web servers

**Experiment**
1. Utilize WHOIS queries, identify 4 distinct IP address prefix associated with EC2.
2. Perform a TCP connect probe on port 80, result in 11315 responsive IPs.
3. Perform a TCP port 443 scan, result in 8375 responsive IPs.
4. Translate the public address into an internal EC2 address, result in 14054 potential targets.

**Goal**
Map the 14054 potential target into instance type and availability zone.

**The data we have?**
4499 sample instances, which has 997 unique internal IPs and 611 unique Dom0 IPs associated with these instances.

**Outline**
1. Introduction
2. EC2 cloud cartography
3. Co-residence
4. Side-channel cross-VM attack
5. Conclusions

**Detecting co-residence**
- Co-residence checking algorithms
  - Internal IP address are close (within 7)
  - Obtain and compare xen Dom0 address, sending TCP SYN and set TTL small: VM -> Dom0 -> VM
  - Network latency: small packet round-trip times.

**Observations**
- On EC2, VMs can be co-resident only if they have identical creation parameters (region, availability zone, instance type)
- Cloud-internal IP address assigned to VMs are strongly correlated with their creation parameters.
- Use EC2's DNS service to map public IP address to private internal address.

**Mapping algorithms**
- All IP from a /16 are from the same availability zone.
- A /24 inherit any included sampled instance type.
- A /24 containing a Dom0 IP address, we associate this /24 address the type of the Dom0's associated instance.
- All /24's between consecutive Dom0 /24's inherit the former's associated type.
Achieving co-residence

Overall strategy
- Derive target's creation parameters
- Create similar VMs until co-residence is detected.

Improvement
- Exploiting EC2's sequential assignment strategy
- Trigger new creation of new VMs by the victim by inducing load (e.g., RightScale)

Experiment
- A single victim instance was launched
- 20 probe instances were launched by another account
- Co-residence checks were performed.

Result

What is side-channel attack
- Cross-VM information leakage due to sharing of physical resources like CPU's data caches, memory bus, network access.


### Cache Attacks

**Example**
- CPU contains small, fast memory cache shared by all applications.
- If only attacker access memory, it is served from the fast cache.
- If victim also access memory, the cache fills up and attacker notices a slow-down.
- From this, attacker can deduce the memory access patterns of victim.

### Measuring Cache Usage

**Prime+Trigger+Probe technique**
- Allocate buffer $B$, enough to fill significant portion of caches.
- Prime: Read $B$ at cache line size to ensure it is cache.
- Trigger: Busy loops until the CPU's cycle increase by a large value.
- Probe: Measure the time it takes to read $B$.

### Estimating Traffic Rates

Performance cache load measurement, and sent http requests at different rates.

![Graph showing performance cache load measurement](credit: Figure from Thanos's paper, CCS 2009)

### Conclusions

- Identify the risk from sharing physical infrastructure using third-party cloud computing.
- Suggested countermeasure: let user choose their VM placement policy.

### Outline

1. Introduction
2. EC2 cloud cartography
3. Co-residence
4. Side-channel cross-VM attack
5. Conclusions