Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds

Ristenpart, Tromer, Shacham, and Savage

Presentation by Thomas Hines

High Level Questions

- Can we know where we are in the cloud?
- Can we know if we are on the same node as the victim?
- Can we intentionally be on the same node as the victim?
- Can we exploit being on the same node as the victim?

- (Spoilers) Yes

Threat Model

- Cloud Provider is trustworthy and not compromised
  - Trivial if not
- Attacker and Victim are both customers to a cloud provider

Case Study: Amazon EC2

- Remember this is 2009
- 20 instances per account
- Two regions: Europe and US
  - Three availability zones per region

Can we know where we are in the cloud?

- External IPs are segmented by zone/region
- Internal IPs are segmented by instance size
- Fairly simple to map out

- It would be possible to prevent this by scrambling IPs assigned to zones/regions/instance sizes, but this would make maintenance harder

Can we know if we are on the same node as the victim?

- The per node hypervisor counts as a hop for TCP traceroute
- Even easier, if the TCP traceroute is only one hop
- They confirmed the instances were on the same node by creating a disk usage covert channel

- Could prevent by disabling TCP traceroute or hiding hypervisor hop

Can we intentionally be on the same node as the victim?

- Must be in same zone, region, and instance size
- Brute force works surprisingly well
  - Launch many instances and check for co-residence
- Launch instance at the same time victim launches
  - Auto-scaling launches new instances in response to demand
    - Time launches to peaking demand
    - Artificially create demand

- Prevented by letting users choose to have exclusive instances
  - Insignificant extra cost for large users

Can we exploit being on the same node as the victim?

- Measure victim cache usage
  - Fill cache, then read cache after victim time slice
  - Cache based covert channel
- Estimate traffic rates of victim server
- Keystroke timing
  - Burst of activity on each keystroke on an idle instance

- Difficult to prevent except by eliminating co-residence

Prevention

- They suggest giving the users the option to eliminate co-residence
- xlarge instances are already one instance per node
- Users that require many small instances could require that they are the only instances on the node

Critique

- Nicely partitioned into four major questions
- The exploit section was a little weak
  - Mostly theoretical
  - They appear to have tried to steal keys and failed