Supporting Security Sensitive Tenants in a Bare-Metal Cloud

Problems with Existing Cloud Offerings

1. A Virtualization-based shared hardware offering is prone to side-channel, covert-channel, hyperjacking, etc.
Problems with Existing Cloud Offerings

2. Cloud orchestration softwares have huge trusted computing base (TCB) and a massive attack surface.
3. **Limited visibility and control over implementation and operation; tenants needs to trust non-maliciousness and competence of the provider**
Problems with Existing Cloud Offerings

4. Adheres to one-size-fits-all security solutions for operational efficiency
Problems with Existing Cloud Offerings

Bare-Metal clouds overcome the problems faced by virtualized offerings but are prone to firmware-based attacks and still possess other public cloud problems (2, 3 and 4).
Is it Possible to Architect a Cloud that...

- Is appropriate for even the most security-sensitive tenants?
- Doesn’t require the tenants to fully trust the provider?
- Doesn’t impact tenants with less stringent security requirements or who are willing to trust the provider for their security?
Bolted: An Architecture for Secure Bare-Metal Cloud Service
Isolation Service

Attestation Service

Provisioning Service

Allocate a node and move it into a quarantined state where node is isolated

1

Allocate a node and move it into a quarantined state where node is isolated

2

Download bootloader and client side attestation software

Download bootloader and client side attestation software

3

Attest Node’s Firmware

Attest Node’s Firmware

4

If Attestation passes: move the node to tenant’s enclave

If Attestation passes: move the node to tenant’s enclave

5

Provision the node with tenant’s OS and applications

Provision the node with tenant’s OS and applications

6

If Attestation fails: moves the node to rejected pool

If Attestation fails: moves the node to rejected pool

Tenant Secure Pool

Rejected Pool

Free Pool
Allocate a node and move it into a quarantined state where node is isolated

Download bootloader and client side attestation software

If Attestation passes: move the node to tenant’s enclave

If Attestation fails: move the node to rejected pool

Provision the node with tenant’s OS and applications

Bare Metal Imaging (BMI)

- Allocates nodes
- Creates networks
- Controls provider’s switches
- Provides VLAN-based isolation
- Must be deployed by the provider
- Remote attestation and key management system
- Delivers tenant’s kernel, initrd, encryption keys, ...
- Serves Images from network-mounted boot drives
- No need to scrub the node before releasing it
- Network booted node only fetches the part of the image it uses

Bolted Implementation

Bolted Implementation
Answering different security needs of different tenants
Minimizing the trust in the provider
Network Encryption

- To protect against provider
- Securely bootstrapped through Keylime
● Limits the access to tenants’ remotely stored data including the provider
● Encrypted data on local disk with ephemeral keys stored only in memory
● Securely bootstrapped through Keylime
What about the firmware?
What about the firmware?

- BIOS, UEFI, … are huge
  - Vulnerable to attacks; potentially enabling tenants to modify FW
  - No way for tenant to inspect FW
What about the firmware?

- LinuxBoot: A stripped down linux firmware
  - Open source
  - Deterministically built
What about the firmware?

- Bolted works with either UEFI or LinuxBoot
  - With UEFI, download LinuxBoot runtime (Heads) as execution environment for Keylime client
  - We have burned Heads into a small number of servers
Boot Time

- Dell R630 server
  - 2 Xeon E5-2660 v3 2.6 GHz
  - 256 GB RAM

Higher Security with Performance!!!
The Cost of Minimizing Trust on the Provider

Performance Degradation (%)

MPI

Applications

LUKS
IPSec
LUKS+IPSec

EP  CG  FT  MG
● HIL
  ○ https://github.com/cci-moc/hil
● BMI
  ○ https://github.com/cci-moc/ims
● Keylime
  ○ https://github.com/mit-ll/python-keylime
● LinuxBoot
  ○ https://github.com/osresearch/linuxboot
Concluding Remarks

- It is possible to measure all components needed to boot a server securely
- Small Microservices; most can be deployed by tenants and not in TCB
  - Minimizing trust in the provider
  - Provider does not need to deploy a global security policy
- Supporting even the most security sensitive tenants
- Tenants can make the cost/performance/security tradeoff