Accountability in Hosted Virtual Networks

Eric Keller, Ruby B. Lee, Jennifer Rexford
Princeton University
VISA 2009
Motivation

• Trend towards hosted virtualized infrastructures
  – Enables companies to easily deploy new services
  – e.g., Amazon EC2

• Hosted virtual networks
  – *Infrastructure provider*: owns/maintains routers
  – *Service provider*: leases slices of routers
Understanding Security Threats

• **Service Provider wants**
  – Control software running exactly as written
  – Data plane forwarding/filtering as instructed
  – Data plane performing with QoS promised
  – Confidentiality/Integrity of data
  – Availability

• **Infrastructure Provider**
  – Doesn’t want to be unjustly blamed

• **Next: How are these possibly compromised**
Old model: Owning the router

- Entire platform is trusted
New model: Hosted (threat 1)

- Infra. Provider can tamper with control software,
- data plane configuration (HW router),
- data plane implementation (SW router)
New model: Shared (threat 2)

- Pink service provider can attack virtualization layer
- Possible competitor of Blue service provider
  - Affect operation of Blue service provider
Accountability

- Security threats lead to the need for accountability

- Accountable: Subject to the obligation to report, explain, or justify something; responsible; answerable [Random House]

- In hosted virtual infrastructure…
  – promised in the Service Level Agreement (SLA)
Outline of Approaches

• Detect
  – Network Measurement

• Prevent
  – Advances in Processor Architecture

• For each
  – Present solution possible today
  – Propose extension
Outline of Approaches

• Detect
  – Network Measurement

• Prevent
  – Advances in Processor Architecture

• For each
  – Present solution possible today
  – Propose extension
Monitoring SLA compliance

- Probe to determine:
  - Loss rates
  - Latency/Jitter
  - Path taken
- To know how DP supposed to act:
  - Log control messages (at boundaries)
  - Model network and replay logs
Extending the Interface Card

• Treat interface card as trusted (trusting vendor)

  • Enables performing measurement at each router
    – Reduces computation overhead
    – Improves accuracy
    – Improves amount of detail

• Enables independent verification
Outline of Approaches

• Detect
  – Network Measurement

• Prevent
  – Advances in Processor Architecture

• For each
  – Present solution possible today
  – Propose extension
Trusted Platform Module

• Recall what service provider wants
  – Control software running unmodified
  – Data plane acting as instructed
  – Data plane performing with correct QoS
  – Confidentiality/Integrity of data

• TPM: Chip on motherboard (on chip in future)
  – Encrypting storage
  – Attesting to integrity of system
TPM Limitations

- Does not protect against dynamic attacks
  - Can’t ensure software running unmodified

- Relies on chain of trust
  - Virtual machine verified by virtualization layer

- Implications
  - Can’t know if control processes started correctly and haven’t been modified
  - Can’t know if data plane acting as instructed with QoS
    (SW - Data plane is in virtualization layer)
    (HW – Configuration goes through virtualization layer)
  - Confidentiality of data not addressed
TPM needs physical separation

- Separate route processors (Logical routers)
- Remote control plane (4D, Ethane)
Security Enhanced Processor

• TPM relies on physical separation

• Instead – extend processor architecture
  – Confidentiality/integrity of data and software
  – Encryption/decryption to/from memory
  – Examples: SP[ISCA05], AEGIS[MICRO03], XOM[ASPLOS00]
  – Minimal extra circuitry

• None designed for hosted/shared environment

• None made good business case
  – So no (very limited) success
  – Market size of hosted virtualized infrastructures provides
    the incentive
Protecting Software and Data

- **Vendor installs private device key**
  - Write only

- **Service provider installs a secret key**
  - Encrypted with device’s public key
  - Sent to infrastructure provider to install
  - Write only

- **Service provider encrypts/hashes memory**
  - With secret key

- **Memory hashed and/or encrypted in main memory**
  - Decrypted/verified when cache line pulled in
  - Encryptedhashed when evicted
## What’s the right approach?

<table>
<thead>
<tr>
<th></th>
<th>Measure</th>
<th>+NIC</th>
<th>TPM</th>
<th>vm-SP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trust</strong></td>
<td>Other infrastructure providers</td>
<td>Vendor</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td><strong>Run-time complexity</strong></td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Confidentiality</strong></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Main downside</strong></td>
<td>Accuracy vs computation / storage tradeoff</td>
<td>Need to extend interface card</td>
<td>Requires physical separation</td>
<td>Need general purpose processor extension</td>
</tr>
</tbody>
</table>

- Virtual Mode-SP (extended processor) provides protection desired, minimal complexity, with business incentives to make it reality.
Conclusion

• A step toward realizing hosted virtual networks

• New business model leads to new security issues
  – Platform is hosted and shared

• Can use monitoring to detect violations

• Better to rearchitect routers to prevent violations

• Future work:
  – Virtual Mode-SP for hosted virtualized infrastructures
  – Explore implications of trusting the vendor
Questions