This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 643964.

This work was supported (in part) by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0091.

Sehen: Secure Yet Efficient Virtual Network Embedding in a Multi-Cloud Environment

Max Alaluna

User-centric management of security and dependability in clouds of clouds
“We all live every day in virtual environments, defined by our ideas.” [Michael Crichton]
The concept

virtual network 2

virtual network 1

virtual network 3

supercloud

Network Hypervisor

User-centric management of security and dependability in clouds of clouds
Motivation

virtual network 1

virtual network 2

Tenants (End users)

Private Data center

Network Hypervisor
Motivation

virtual network 1  virtual network 2

Tenants (End users)

Private Data center (totally dependent on a cloud provider)

Network Hypervisor
Motivation

Tenants
(End users)

Private
Data center
(totally dependent on a cloud provider)
Motivation

Tenants
(End users)

Private
Data center
(totally dependent on a cloud provider)
Our very high-level idea: use multiple hybrid clouds
High-level view of the solution

virtual network 1

virtual network 2

Multi-cloud Network Hypervisor

Tenants (End users)

Public cloud A

Public cloud B

Private cloud

08 Fevereiro 2017
High-level view of the solution

Tenants (End users)

Multi-cloud Network Hypervisor

Public cloud A

Public cloud B

Private cloud
High-level view of the solution

Multi-cloud Network Hypervisor

Tenants (End users)

Public cloud A

Public cloud B

Private cloud
Motivation

Current platforms share a few characteristics:

• target the datacenter of a single cloud provider;

  ➢ Problem: single point of failure (SPOF).
Motivation

Current platforms share a few characteristics:

- target the datacenter of a single cloud provider;

  ➢ Problem: single point of failure (SPOF).

- do not consider security and dependability in their design.
Current platforms share a few characteristics:

• target the datacenter of a single cloud provider;
  ➢ Problem: single point of failure (SPOF).

• do not consider security and dependability in their design

• requires full control over the infrastructure
  ➢ Problem: targets only private cloud settings (not leveraging from existing public cloud infrastructure)
Objectives / goals

We propose to build a network hypervisor that offers its tenants:

• scale out **full network virtualization**;

  ➢ any topology
  ➢ any address scheme
  ➢ isolation between tenants
Objectives / goals

We propose to build a network hypervisor that offers its tenants:

• full network virtualization;

• security and dependability requirements;
  ➢ tolerate cloud outages
  ➢ enable various levels of security (of virtual nodes & links)
  ➢ enable various levels of dependability (of virtual nodes & links)
Challenge

Efficient & Secure Virtual Network Embedding (VNE) in a multi-cloud scenario;
Motivation Secure VNE

1. Cloud outages
2. Corrupt cloud insider
3. Needs to comply with privacy legislation
4. Increase resilience and availability
Challenge

Efficient & Secure Virtual Network Embedding (VNE) in a multi-cloud scenario;
Challenge

**Efficient & Secure** Virtual Network Embedding (VNE) in a multi-cloud scenario;

**BUT,**

what is **Virtual Network Embedding?**
Network Virtualization

1. Allows multiple heterogeneous networks (specified by different users) to run on a shared infrastructure.
Network Virtualization

1. Allows multiple heterogeneous networks (specified by different users) to run on a shared infrastructure.

2. Virtual Network Embedding: Finding an effective mapping of the virtual nodes and links onto the substrate network.
Virtual Network Embedding

• Finding an effective mapping of the virtual nodes and links onto the substrate network.
Virtual Network Embedding

• Finding an effective mapping of the virtual nodes and links onto the substrate network.
Virtual Network Embedding

• Finding an effective mapping of the virtual nodes and links onto the substrate network.
Virtual Network Embedding

• Finding an effective mapping of the virtual nodes and links onto the substrate network.
Virtual Network Embedding

- Finding an effective mapping of the virtual nodes and links onto the substrate network.

08 Fevereiro 2017
Navtalk
Secure VNE - Model

Basic requirements

Nodes: CPU

Links: bandwidth
Secure VNE - Requirements security demands

Communications (virtual links)
- Confidentiality & Integrity
  - LC0 – Default
  - LC1 – Authenticity, Integrity
  - LC2 – Confid. & Auth., Integrity
- Availability
  - LA0 – Single path
  - LA1 – Replicated paths

Computations (virtual nodes)
- Trusted Executions
  - NT0 – Container
  - NT1 – VM
  - NT2 – Secure VM
- Availability
  - NA0 – Single virtual node
  - NA1 – Replication (another cloud)
  - NA2 – Replication (same cloud)

Trust Domains (clouds)
- CT0 – Public Cloud
- CT1 – Trusted Public Cloud
- CT2 – Private Cloud
Secure VNE – previous work

• We have developed a MILP formulation to solve the SecVNE problem.

• The objective function of our formulation had three goals, to minimize:
  1. the sum of all computing costs;
  2. the sum of all communication costs; and
  3. the overall number of hops of the substrate paths for the virtual links.
Secure VNE – previous work

• We have developed a MILP formulation to solve the SecVNE problem.

• The **objective function** of our formulation had three goals, to minimize:

\[
\min \quad \beta_1 \sum_{(i,j) \in E^V} \sum_{u,v \in N^S} \alpha_{u,v} \ w_{f_{u,v}}^{i,j} \ sec^S(u,v) \\
+ \beta_1 \sum_{u,v \in N^S} r_{l_{u,v}} \ sec^S(u,v) \\
+ \beta_2 \sum_{i \in N^V} \sum_{v \in N^S} \cpu^Y(v) \ w_{n_{i,v}} \ sec^S(v) \ cloud^S(v) \\
+ \beta_2 \sum_{v \in N^S} r_{n_v} \ sec^S(v) \ cloud^S(v) \\
+ \beta_3 \sum_{(i,j) \in E^V} \sum_{u,v \in N^S} w_{l_{u,v}}^{i,j} \\
+ \beta_3 \sum_{(i,j) \in E^V} \sum_{u,v \in N^S} b_{l_{u,v}}^{i,j} \quad (1)
\]
Secure VNE weakness

1. MILP problems are NP-complete: do not scale for a large network

2. The MILP solution has many constraints to solve both, working and backup, VNE in one step: could be simplified
Efficient & Secure VNE

We propose a **heuristic** to solve the problem.
Efficient & Secure VNE

We propose a **heuristic** to solve the problem.

**Sehen**: Secure Yet Efficient Virtual Network Embedding in a Multi-Cloud Environment
Sehen: A Secure Yet Efficient VNE in a Multi-Cloud Environment

1. A two step heuristic
   a. First: node mapping based on the degree & available resources of nodes, & available bandwidth of its links by means of a “Utility function”.
1. A two step heuristic
   a. First: node mapping based on the “Utility function”
      * CPU: size of the node
      * Bandwidth: width of the link
1. A two step heuristic
   a. First: node mapping based on the “Utility function”

   Illustrative Network

   Random choosing
1. A two step heuristic
   a. First: node mapping based on the “Utility function”
1. A two step heuristic
   a. First: node mapping based on the “Utility function”

   **Random choosing**

   Possibly NOT the best choice
1. A two step heuristic
   a. First: node mapping based on the “Utility function”
1. A two step heuristic
   a. First: node mapping based on the “Utility function”
1. A two step heuristic

   a. First: node mapping based on the “Utility function”

\[
U(x) = \%R_N \left( \sum_{e \in \{\text{nodes}\}} \%R_E \right) \cdot \left[ \log(n) + 1 \right] \cdot (\text{sec or cloud})
\]

\( R_N \): Residual capacity of the substrate node

\( R_E \): Residual capacity of the substrate link

\( n \): # of links in a specific node

\( \text{sec} \): node security level

\( \text{cloud} \): cloud security level
1. A two step heuristic
   a. First: node mapping based on the “Utility function”
   b. Second: link mapping based on MCF linear programming

2. Repeat for Backup VNE nodes&links

3. Evaluate against 2 naive heuristics, the precursor heuristic (FG-Heu) and the most common VNE algorithm (D-Vine)
   a. FR-Heu performs random node mapping in a full random way
   b. PR-Heu apply random mapping after excluding nodes that do not attend requirements
Evaluation – 25 nodes (Optimal solution)

1. “20”: indicates 20% of VN request demands Backup Virtual Network
2. “H”: indicates 70% of virtual nodes demands security
3. “L”: indicates 30% of virtual nodes demands security
SecVNE: Secure Virtual Network Embedding in a Multi-Cloud Environment

Evaluation – 25 nodes (Optimal solution)

Acceptance ratio about 95% without considering security
SecVNE: Secure Virtual Network Embedding in a Multi-Cloud Environment

Evaluation – 25 nodes (Optimal solution)

Acceptance ratio about 95% without considering security

Acceptance ratio between 86% and 92% considering security
Evaluation – 25 nodes (Optimal solution)

With 50 nodes the optimal algorithm ran for more than 150 minutes and did not present a result.

SecVNE: Secure Virtual Network Embedding in a Multi-Cloud Environment
SecVNE: Secure Virtual Network Embedding in a Multi-Cloud Environment

Evaluation – 25 nodes (Optimal solution)

With 50 nodes the optimal algorithm run for more than 150 minutes and did not present a result.

It doesn’t scale
Evaluation – 50 nodes

1. “50”: indicates 50% of VN request demand Backup Virtual Network
2. “H”: indicates 70% of virtual nodes demand security
3. “L”: indicates 30% of virtual nodes demand security
Evaluation – 50 nodes

1. “50”: indicates 50% of VN request demand Backup Virtual Network
2. “H”: indicates 70% of virtual nodes demand security
3. “L”: indicates 30% of virtual nodes demand security
Evaluation – 50 nodes

1. “50”: indicates 50% of VN request demand Backup Virtual Network
2. “H”: indicates 70% of virtual nodes demand security
3. “L”: indicates 30% of virtual nodes demand security
4. D-Vine and FG-Heu do not considers security
Evaluation – 50 nodes

1. “50”: indicates 50% of VN request demand Backup Virtual Network
2. “H”: indicates 70% of virtual nodes demand security
3. “L”: indicates 30% of virtual nodes demand security
4. D-Vine and FG-Heu do not considers security
Conclusion

1. Very good acceptance ratio comparing to other algorithms
Conclusion

1. Very good acceptance ratio comparing to other algorithms

2. Considers security and availability
Conclusion

1. Very good acceptance ratio comparing to other algorithms

2. Considers security and availability

3. Time around 30 min to run 1,000 requests: it DOES scale
Questions ?
Conclusion

Thank you !!!