MI-LXC: A Small-Scale Internet-Like Environment for Network Security Teaching

François Lesueur\textsuperscript{1,2} (francois.lesueur@insa-lyon.fr / @FLesueur)
Camille Noûs\textsuperscript{2} (camille.nous@cogitamus.fr / @NousCamille)

https://github.com/flesueur/mi-lxc

ETACS, August 17, 2021

\textsuperscript{1}INSA Lyon, Département Télécommunications, Services et Usages, CITI, DynaMid
\textsuperscript{2}Laboratoire Cogitamus
Problem Statement and Objective

- *Network* security requires a (large) network of hosts
- Creating, maintaining and distributing this large network is difficult

**Core requirements**

- Lightweight: Should run on students’ laptops
- Maintainable: Should be manageable by a (very) small team
- Representative: Should allow to run realistic scenarios

**Existing solutions**

- Network-centered (GNS3, Mininet, …) do not fit realistic services
- Docker-based (Labtainers, Kathara, …) do not simulate whole operating systems
- VM-based cyberranges (ADLES, KYPO, …) require large resources
**Mini-Internet using LXC ?**

- A framework to build virtual infrastructures
  - *Infrastructure-as-code*
  - LXC containers
  - Maintainable, versionnable, SLOC-scalable, lightweight

- A reference topology simulating a *mini-internet*
  - Core services: DNS, SMTP, HTTP, …
  - BGP routing among independent AS
  - A prerequisite to practice network/internet security

- Some security practical works (in French)
  - Certification Authorities (ACME)
  - Network intrusion
  - Network segmentation
  - IDS
A framework to build virtual infrastructures
Topology specification

Target infrastructure specification

- Global topology in *global.json*
- AS local topologies in different *local.json*
- Bash provisioning for each host

Template mechanism

- AS templates
- Host templates
A reference topology simulating a *mini-internet*
What is simulated?

**Internet roots* (personal view...)

- Interconnection of Autonomous Systems (AS)
- Through multi-path routing (transit, peering, BGP)
- Using some standardized protocols (BGP, HTTP, SMTP, ...)
- In an orchestrated/federated organization (IANA, ICANN, IETF, ...)

- **11 AS (transit + edge)**
- **BGP routing**
- **Alternative DNS root**
- **An internal TLD (.milxc)**
- **Some DNS zone xyz.milxc**
- **SMTP, IMAP, HTTP**
- **Graphical mail clients**
- **Suricata, OSSEC, Prelude, SmallStep CA...**
Some figures

- 28 containers, 12 network bridges, 6GB HDD, 2GB RAM
- 1000 Python lines (framework), 1000 Bash lines (provisioning), 300 JSON lines (topology)

So it is...

- Versionable
- SLOC-scalable
- Lightweight
- Maintainable
Training examples
HTTPS / CA

**Attack model**
- HTTP connection
- BGP hijacking (or DNS, MitM)

**ACME CA deployment**
- CA generation (Smallstep)
- Certificate request
- CA integration in the trust store
- Browser update

**Remaining risk**
- Attack during the certification
What’s next?

What is working?

- This infrastructure with 4 trainings
- Quite stable (thanks to all my students ;-) )
- Licensed under AGPL: https://github.com/flesueur/mi-lxc

Perspectives

- New scenarios?
- Some (legit) background noise?
- Some other security tools (MISP, hunting)?
- Some other network tools (netem, dynamips)?
- Other OS (Windows via VM)?
MI-LXC: A Small-Scale Internet-Like Environment for Network Security Teaching

François Lesueur\textsuperscript{1,2} (francois.lesueur@insa-lyon.fr / @FLesueur)

Camille Noûs\textsuperscript{2} (camille.nous@cogitamus.fr / @NousCamille)

https://github.com/flesueur/mi-lxc

ETACS, August 17, 2021

\textsuperscript{1}INSA Lyon, Département Télécommunications, Services et Usages, CITI, DynaMid

\textsuperscript{2}Laboratoire Cogitamus