MI-LXC (Mini-Internet Testbed) for Network Security Training and Security Tools Demonstration

François Lesueur
francois.lesueur@univ-ubs.fr
@FLesueur / @flesueur@mastodon.social

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

Pass The SALT, July 2022

Université Bretagne Sud, IUT Vannes (Info), IRISA (CASA)

Agenda

- A short introduction (slides)
- A first-steps demo
- Hands-on !

Now is a good time to download the VM
(link on the PTS schedule)

`https://flesueur.irisa.fr/mi-lxc/images/milxc-debian-amd64-1.4.2.ova`

MI-LXC

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 network and security practical works (in French, free, on the Github page)
  - Certification Authorities (ACME)
  - Network intrusion
  - Network segmentation
  - IDS
  - DNS
  - Mail
  - MitM
  - LDAP
  - ...

A reference topology simulating a *mini-internet*

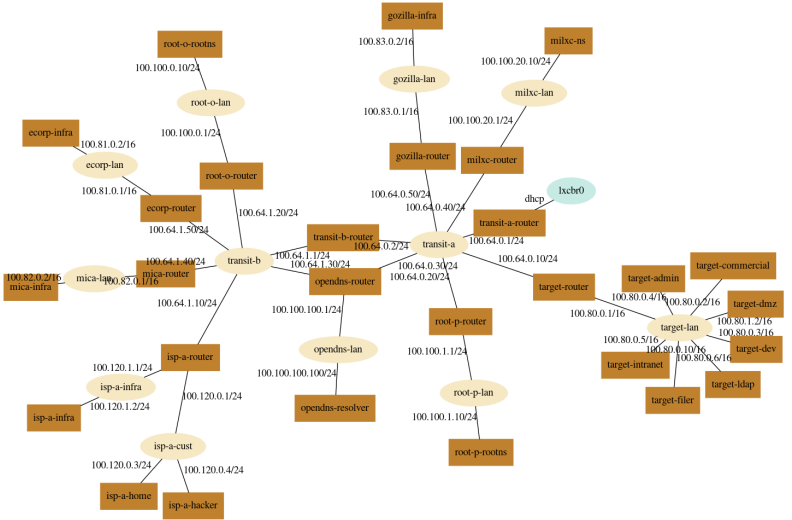
What is simulated ?

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, . . .)

Topology

- 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...



A framework to build virtual infrastructures

Underlying technologies

- LXC (vs Docker, VM) for the hosts
- Lib LXC (vs Vagrant) for manipulating LXC
- Python for the framework (with LibLXC bindings)
- Bash (vs Ansible, Puppet, etc.) for the provisionning of the hosts
- JSON for the specification of the topology
- Linux (vs Windows, BSD, etc.) for the hosts
- (vs Kathara, Marionnet, GNS3, Labtainers, Hynesim, Terraform, CyberRanges)

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

Result

Some figures

- 28 containers, 11 AS, 12 network bridges, 7GB 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 from the web server
- CA integration in the trust store of the browser editor
- Browser update on the web client

Remaining risk

- Attack during the certification

Intrusion Scenario

Objectives

- Understand a multi-step attack workflow
- Privilege escalations (system, network)

- Bruteforce wiki + reverse-shell upload
- Mail spoofing
- Social engineering
- Lazagne
- Nmap
- Network pivot
- Profit !

Network Segmentation

Objectives

- Learn iptables/nftables
- Design a segmented network architecture and a policy matrix

Constraints

- Centralized authententication (LDAP, from IMAP, filer, desktops)
- SMTP, IMAP, DNS, etc. $\rightarrow$ DMZ
- Internal servers (filer, intranet)
- Desktop, admin workstations
- Should add a VPN...

IDS

Objectives

- Discover NIDS/HIDS/Collection/Aggregation
- And that we only find... what we're looking for !

- NIDS : Suricata
  - Bruteforce (HTTP 403 errors)
  - A #!/bin/sh in a packet
  - Internal nmap
- HIDS : OSSEC
  - Bruteforce (Apache logs)
  - Uploaded file (reverse shell) creation
- Collection : Prelude/Prewikka
  - Centralization
  - Correlation

Today's menu

- (Brief) Intro (done)
- First steps demo
- Tutorial (`https://github.com/flesueur/mi-lxc/blob/master/doc/TUTORIAL.md`) :
  - Learner : Learning network/security (sysadmin)
  - Designer : Specification of an infrastructure (JSON + bash)
  - Developper : extending the framework (Python)
  - $\Rightarrow$ Let's start at II.3 !

What's next ?

What is working ?

- This infrastructure with several 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) ?
- Other OS (Windows via VM) ?

What's really coming ? Functionnalities splitting !

The framework: SNSTER

- System and Network Simulator for hipsTERs
- Python framework + collection of templates
- A fast-prototyping tool for different (custom) topologies

⇒ Will be hosted at `https://www.snster.net`

The Mini-Internet topology: MI-LXC

- ∼ the `groups/` subfolder
- Will use the externalized SNSTER

MI-LXC (Mini-Internet Testbed) for Network Security Training and Security Tools Demonstration

François Lesueur
francois.lesueur@univ-ubs.fr
@FLesueur / @flesueur@mastodon.social

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

Pass The SALT, July 2022

Université Bretagne Sud, IUT Vannes (Info), IRISA (CASA)