Mini-Internet using LXC (MI-LXC) : A first step towards a free CyberRange ?

François Lesueur
francois.lesueur@insa-lyon.fr
@FLesueur

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

Pass The SALT, July 2 2019

INSA Lyon, Département Télécommunications, Services et Usages,
CITI, DynaMid group

#whoami

Professional side

- Associate Prof at INSA Lyon
- Teacher and researcher on empowering infosec

Personal side

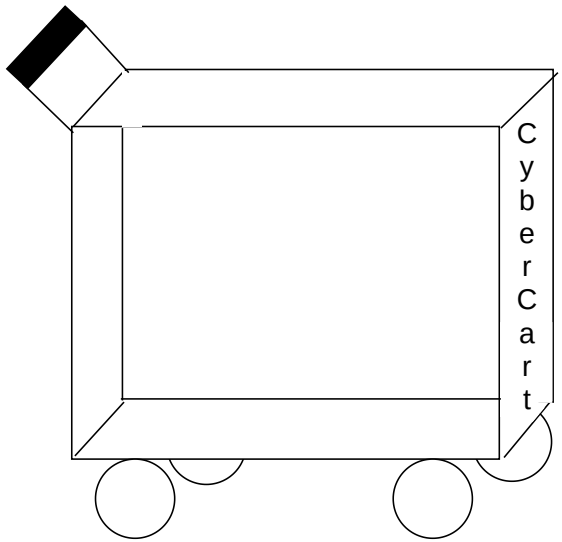
- Long time Debian GNU/Linux user
- Long time self-hosted too
- Half craftsman, half plumber

And on both sides...

Fear an oligopoly on knowledge/data possession/security

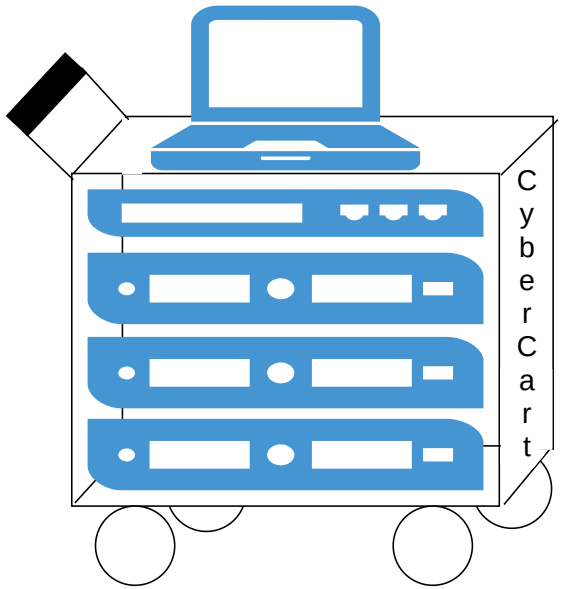
Cyberranges: Platforms to train people on realistic security scenarios

Some insights on cyberranges



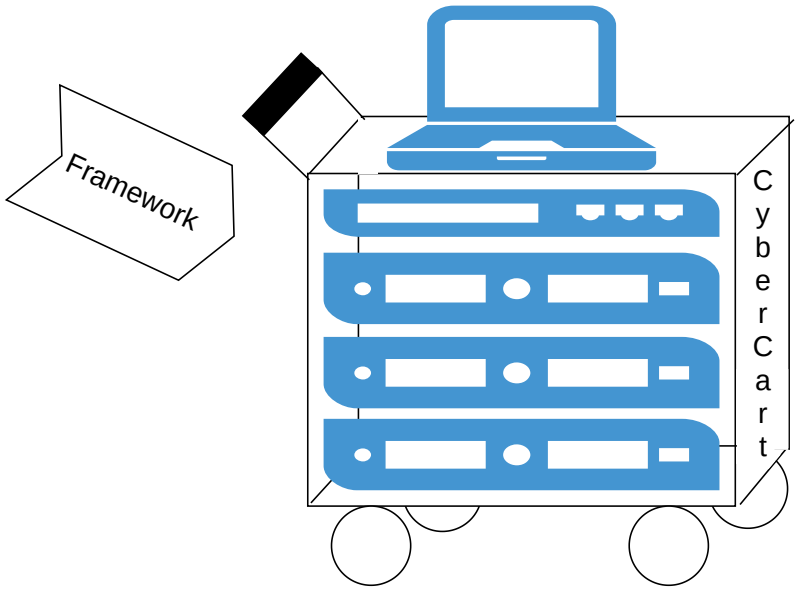
First you need a cart with some fancy name

Some insights on cyberranges



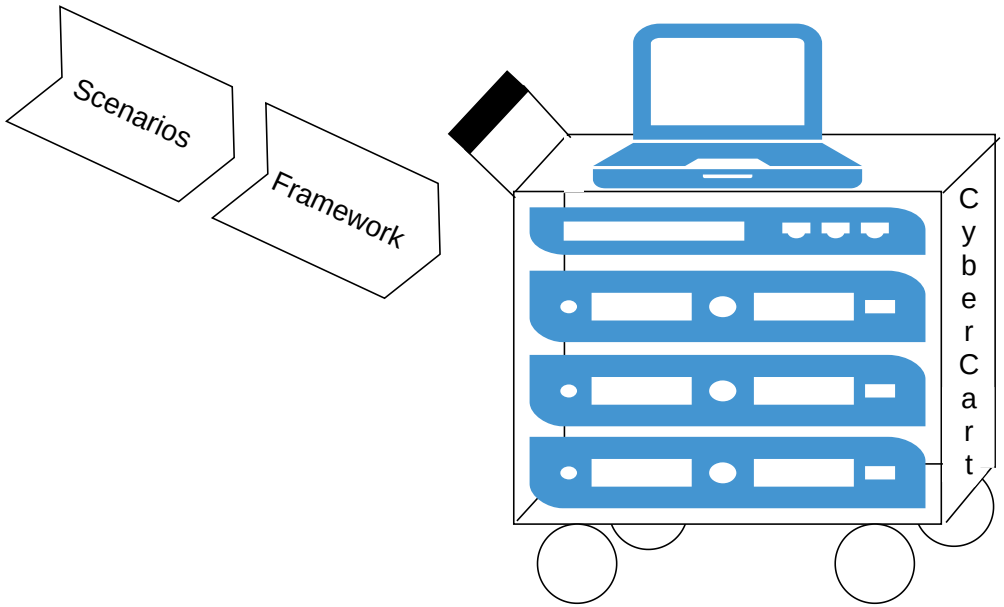
Some dedicated hardware racked into it

Some insights on cyberranges



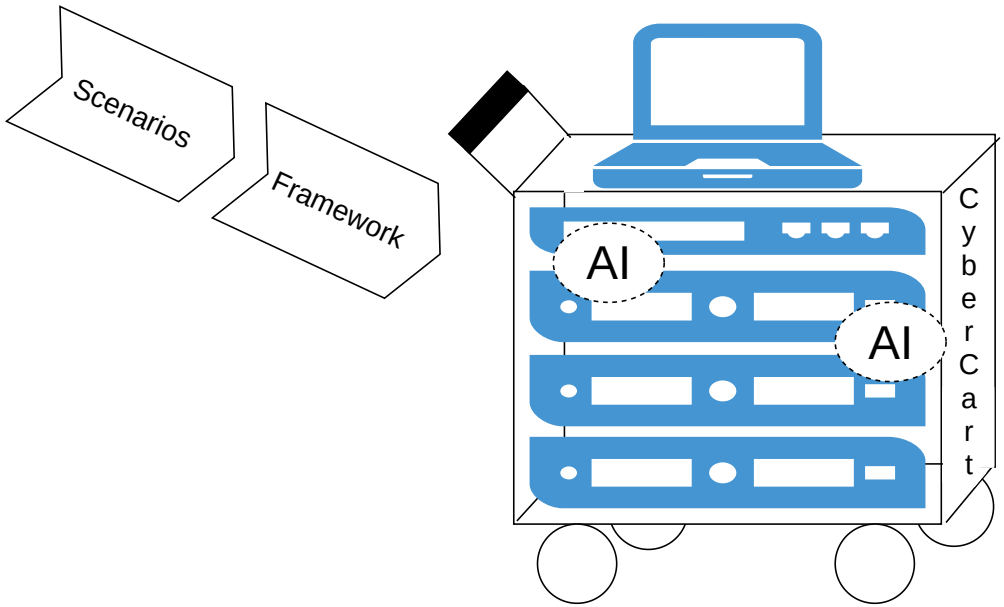
A framework to populate VMs

Some insights on cyberranges



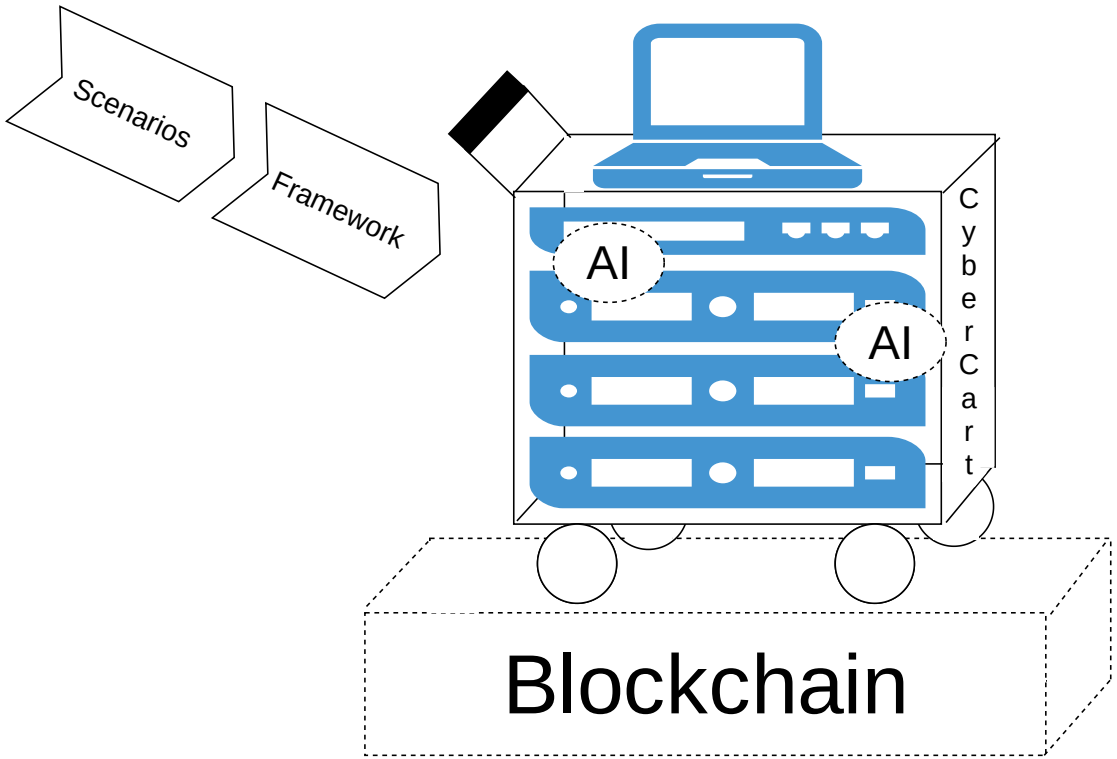
Some scenarios to play

Some insights on cyberranges



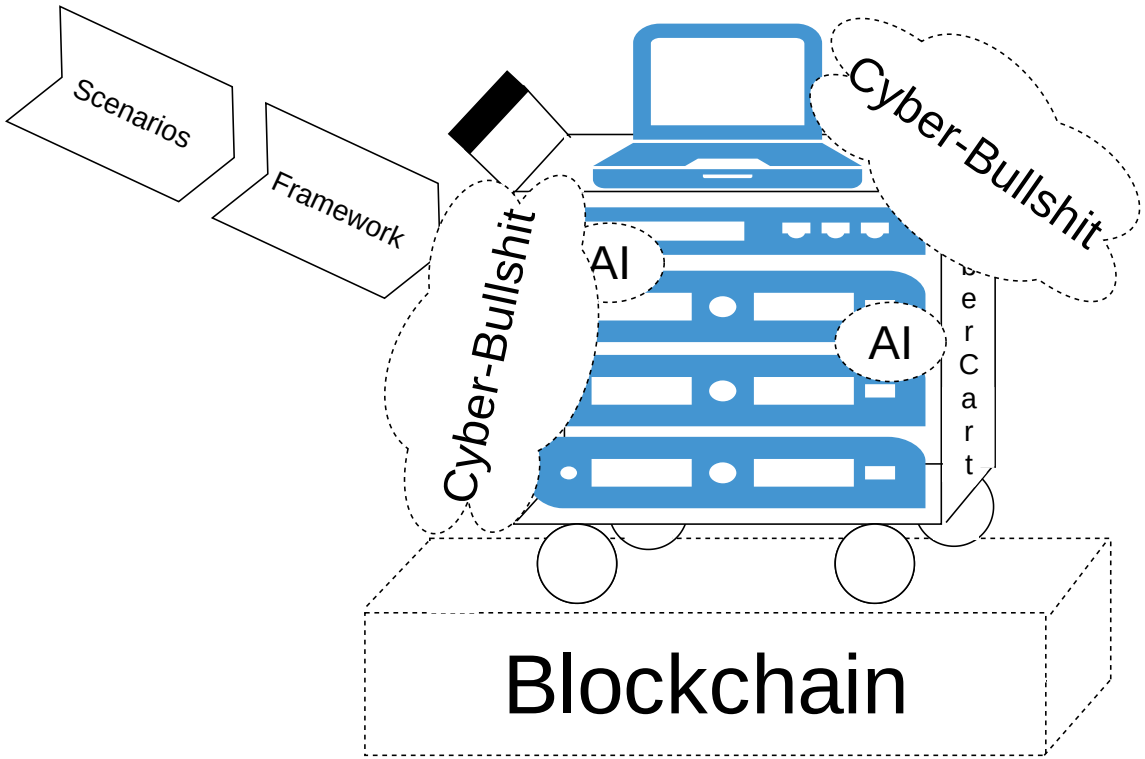
Of course you need AI to be taken seriously...

Some insights on cyberranges



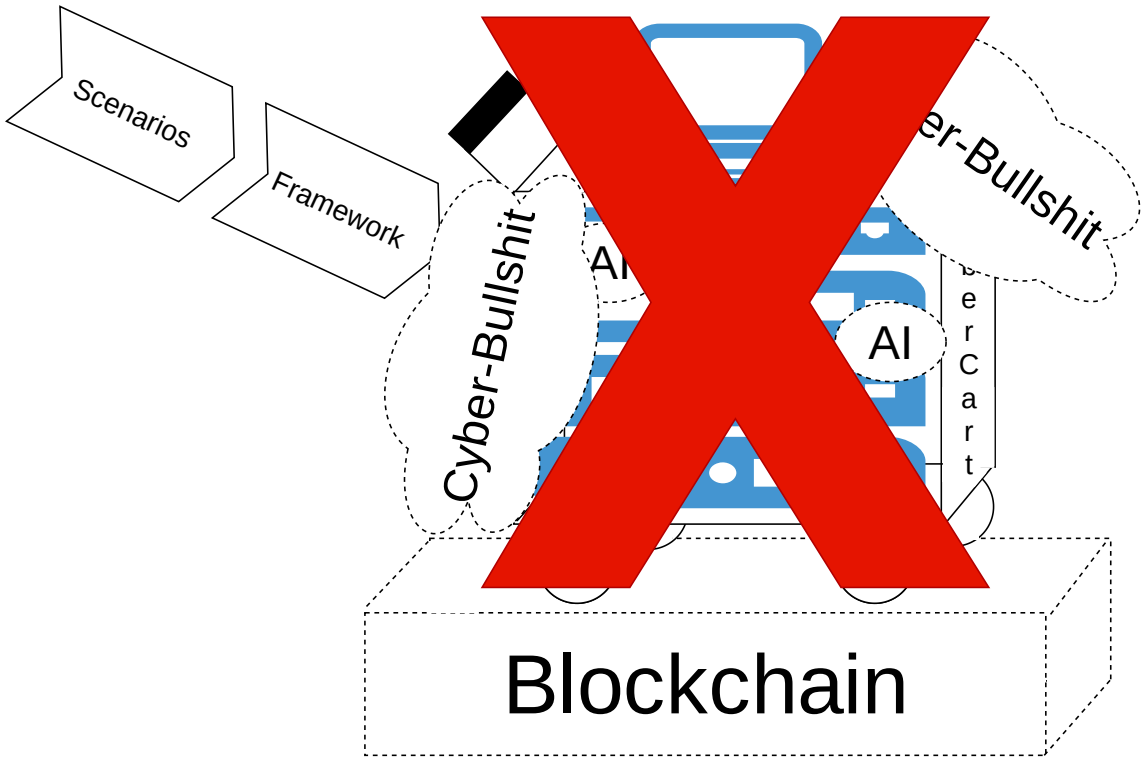
. . . and it is backed by some blockchain for securitay !

Some insights on cyberranges



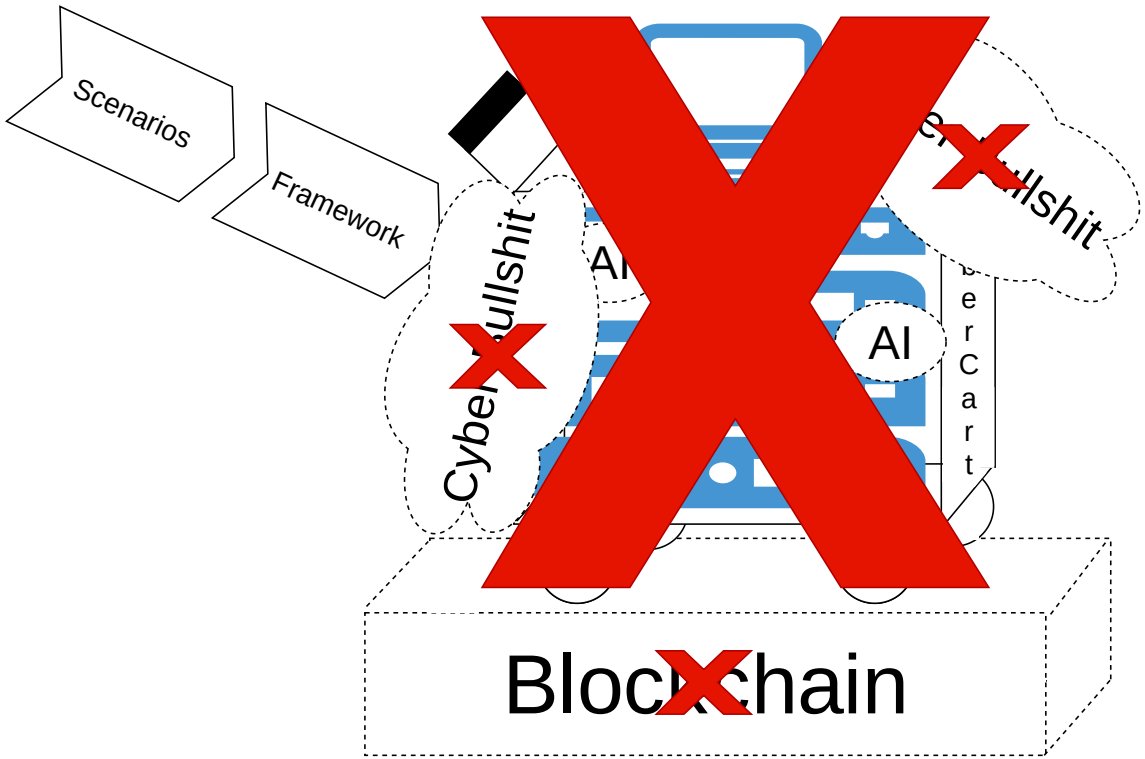
And surrounded (well, sold) by some cyber-bullshit

Some insights on cyberranges



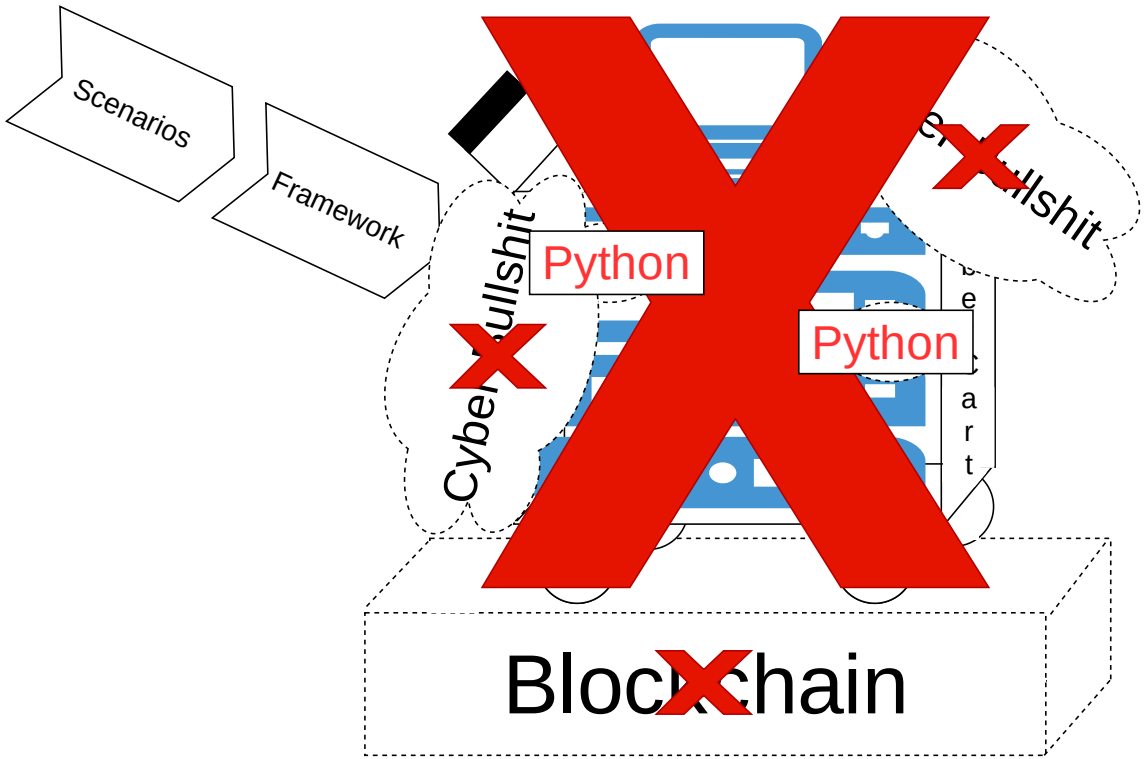
We can run without dedicated hardware. . .

Some insights on cyberranges



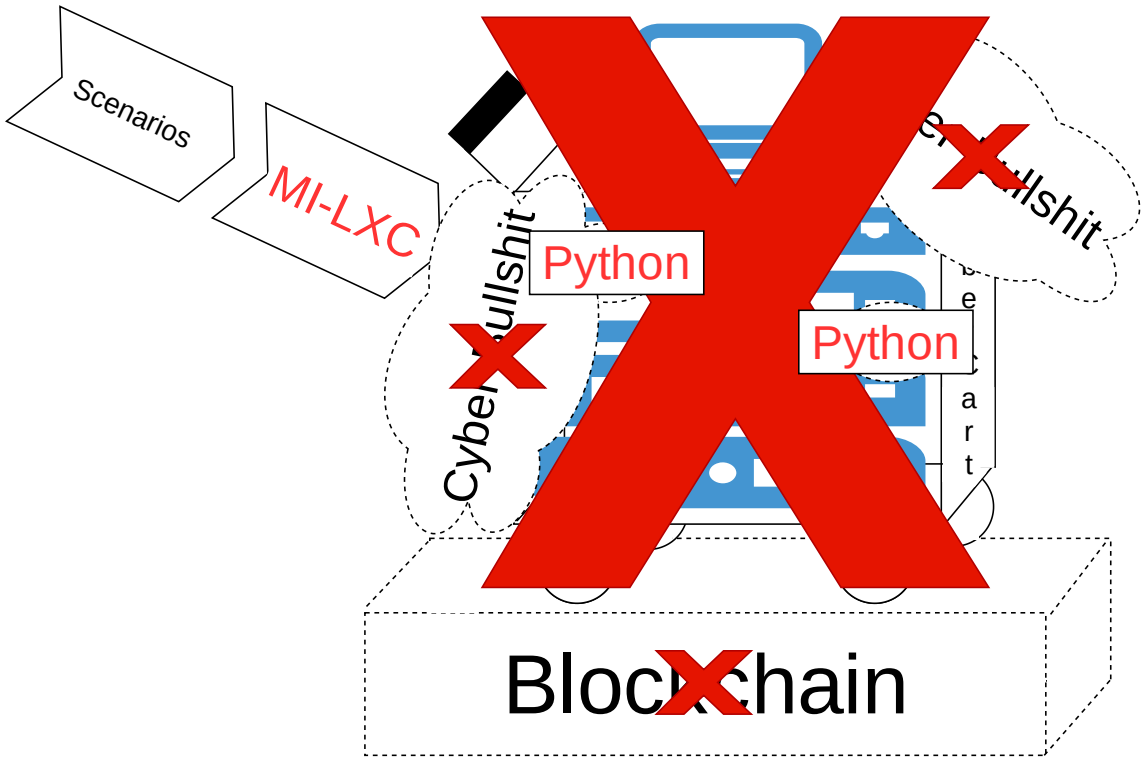
. . . and we don't really need any bullshit

Some insights on cyberranges



AI is just python scripts, right ?

Some insights on cyberranges



Finally, we need some framework to bootstrap scenarios

MI-LXC: A Framework to build virtual infrastructures

A Mini-Internet

What ?

- An environment as close as possible to the real internet
- Information systems (with open services SMTP/HTTP, centralized authentication, file servers, backup, VPN, . . .)
- Interconnection (AS BGP)
- Common services (DNS root, IANA numbering)

How ?

- Versionable, versatile ⇒ Program the infrastructure
- SLOC-scalable ⇒ Mutualize lines
- Rapid to execute, easy to use. . .

Existing frameworks

- Networking frameworks but with no facilities for creating various hosts (Marionnet, Internet Simulator)
- Docker-based tools without *init* and thus no complete systems (Dockernet, Kathara)
- Labtainers, based on Docker, uses a deprecated image with *systemd* + high code complexity
- SecGen geared towards creating vulnerable VMs rather than large systems (Virtualization)

And so...

Let's create a new one ;)

Related tools

"Virtualization"

- VM ? Too resource-expensive
- Containers ! LXC (no init in docker)

Bootstrapping

- Vagrant is more VM-ish (LXC plugin unmaintened)
- LXC Python binding allows to create containers

Provisionning

- Puppet/Ansible deal with mass/run problems we don't have
- Bash scripts

MI-LXC: the generation part

A Python script

- Creates LXC containers
- Topology specified in a JSON file
- Customized provisionning for each container
- *Templates* (mail server, mail client, BGP router, . . .)
- 410 SLOC in `mi-lxc.py`

MI-LXC: the current infrastructure 1/2

At the global level

- A IANA-like authority, attributing ASN, IP space and TLDs
- An alternative DNS root, augmenting the real root with a .milxc
- Several AS (transit, ISP, organization), BGP routing
- An Open DNS resolver

At some local levels

- DNS zones for target.milxc and isp-a.milxc
- SMTP servers for @target.milxc and @isp-a.milxc
- Graphical mail clients (configured)
- HTTP with a dokuwiki on www.target.milxc
- Suricata, OSSEC, Prelude, NSD, BIRD, Postfix, Dovecot, ...

MI-LXC: the current infrastructure 2/2

Initial mini-internet

- 20 containers, 8 internal bridges, 4GB HDD, 800MB RAM
- 698 lines in all provisionning scripts, 165 lines in the topology JSON

And so

- Versionnable
- SLOC-scalable
- Quite small memory/HDD/CPU footprint

What we can do ?

Legit

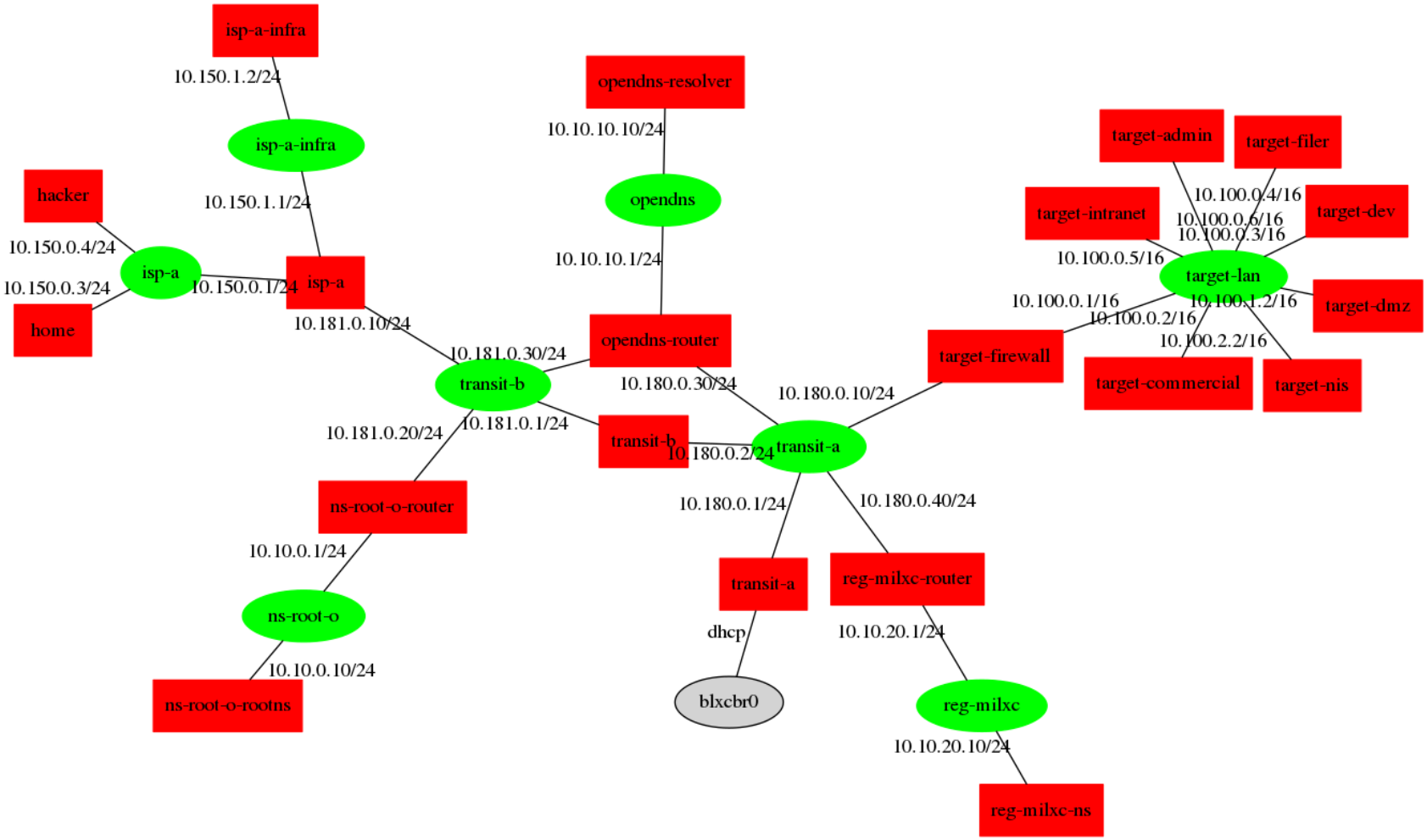
- Send mails
- DNS query inside MI-LXC and outside (the *real* internet)
- Access remote webpages hosted on a container
- Monitor/Filter traffic

Attacks

- DNS and BGP attacks
- Phishing
- Open (reverse-)shells
- Pivot inside a private network
- ...

Demo

Topology


isp-a-infra
10.150.1.2/24
isp-a-infra
hacker
10.150.1.1/24
10.150.0.4/24
isp-a
10.150.0.1/24
isp-a
10.150.0.3/24
home
10.181.0.10/24
opendns-resolver
10.10.10.10/24
opendns
10.10.10.1/24
opendns-router
10.181.0.30/24
transit-b
10.180.0.30/24
10.181.0.20/24
10.181.0.1/24
transit-b
10.180.0.2/24
ns-root-o-router
10.10.0.1/24
ns-root-o
10.10.0.10/24
ns-root-o-rootns
10.180.0.10/24
transit-a
10.180.0.1/24
10.180.0.40/24
transit-a
reg-milxc-router
dhcp
10.10.20.1/24
blxcbr0
reg-milxc
10.10.20.10/24
reg-milxc-ns
target-admin
target-filer
10.100.0.4/16
target-intranet
10.100.0.6/16
target-dev
10.100.0.3/16
10.100.0.5/16
target-lan
10.100.0.1/16
10.100.1.2/16
target-dmz
10.100.0.2/16
target-firewall
10.100.2.2/16
target-commercial
target-nis

How to use it ?

GNU/Linux (Debian, Ubuntu, Arch, Kali)

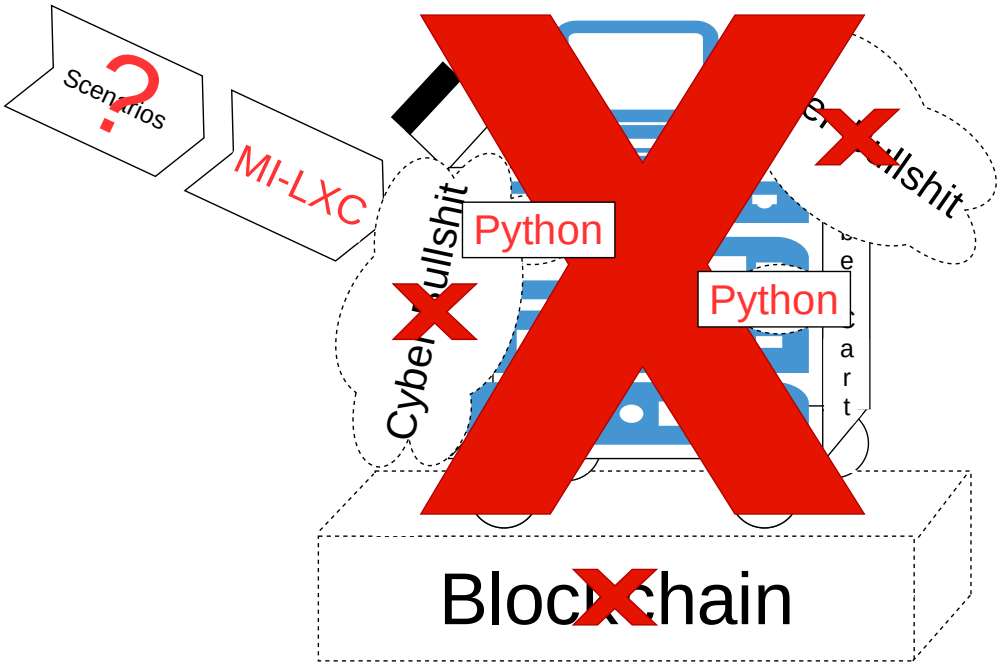
- `git clone https://github.com/flesueur/mi-lxc.git`
- `./mi-lxc create` (15-20 minutes)
- `./mi-lxc start`
- `./mi-lxc attach dmz` ; ./mi-lxc display hacker
- `./mi-lxc print`

Other systems

- `git clone https://github.com/flesueur/mi-lxc.git`
- `cd vagrant && vagrant up` (20-25 minutes)
- `./mi-lxc start` (inside the VM)
- `./mi-lxc attach dmz` ; ./mi-lxc display hacker
- `./mi-lxc print`

What's next ?

And now ?



- More scenarios
- Python activity inside the infrastructure
- Infrastructure / Security tools to support various situations

Mini-Internet using LXC (MI-LXC) : A first step towards a free CyberRange ?

François Lesueur
francois.lesueur@insa-lyon.fr
@FLesueur

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

Pass The SALT, July 2 2019

INSA Lyon, Département Télécommunications, Services et Usages,
CITI, DynaMid group