

# Lezione 2: Introduzione a IMUNES

Claudio Ardagna, Patrizio Tufarolo – Università degli Studi di Milano

Insegnamento di Laboratorio di Reti di Calcolatori



# Riepilogo dell'ultima lezione

- ▶ Abbiamo visto:
  - ▶ Concetti di base del Software Defined Networking
  - ▶ Emulazione, simulazione, virtualizzazione
  - ▶ OpenVSwitch
  - ▶ Commandistica Linux

# Registratevi per accedere al laboratorio!

- ▶ Primo step:
  - ▶ Registrare le utenze per le postazioni di laboratorio!
    - ▶ Recatevi in Laboratorio Est, accedendo al chiosco con le vostre credenziali di ateneo (indirizzo e-mail e password).
    - ▶ Per accedere alle macchine Linux dovete usare:
      - ☐ Username: mail di ateneo **SENZA il suffisso @studenti.unimi.it**
      - ☐ Password: password di ateneo
- ▶ NB: Recatevi al chiosco anche se avete effettuato la registrazione in passato, in modo che vengano generate anche le credenziali per l'utilizzo di Linux
- ▶ Ad ogni utente sono assegnate risorse di storage limitate per la propria home directory. Non è possibile accedere allo storage da macchine non appartenenti al laboratorio didattico. Si raccomanda l'utilizzo di un dispositivo di storage di vostra proprietà



# Terminologia

- ▶ Topologia di rete - modello che rappresenta
  - ▶ La disposizione geometrica degli elementi di una rete di calcolatori (nodi del grafo di rete)
  - ▶ Le interconnessioni, fisiche o logiche, tra i nodi (archi del grafo di rete)
- ▶ Livello datalink
  - ▶ Livello 2 dello stack ISO/OSI
- ▶ Livello di rete
  - ▶ Livello 3 dello stack ISO/OSI
- ▶ MAC Address
  - ▶ Media Access Control address, indirizzo univoco a 48 bit assegnato dal produttore ad ogni scheda di rete compliant con il protocollo Ethernet, wireless o wired che sia. È usato come identificativo per la comunicazione a livello 2
- ▶ IP Address
  - ▶ Internet Protocol address, indirizzo a 32 bit assegnato dal network administrator ad un dispositivo di rete, che consente la comunicazione a livello 3



# Introduzione a IMUNES

## ▶ IMUNES

- ▶ Cos'è IMUNES
- ▶ Come avviare IMUNES in laboratorio
- ▶ Come installare (ed avviare) IMUNES sul proprio PC
- ▶ Funzionalità di IMUNES
- ▶ Esempio di una rete basilare basata su IMUNES
- ▶ Come salvare e ripristinare una topologia di rete



# IMUNES - 1

- ▶ IMUNES è un framework di emulazione/simulazione di topologie di rete che partiziona lo stack di rete del sistema operativo al fine di realizzare dei nodi virtuali, i quali possono essere interconnessi tra di loro con collegamenti simulati dal kernel, permettendo di comporre topologie di rete complesse
- ▶ Utilizza funzionalità native del kernel
- ▶ È compatibile con FreeBSD e Linux

## IMUNES - 2

- ▶ Dispone di un'interfaccia grafica per disegnare il grafo di rete e gestire singolarmente sia i nodi dell'infrastruttura che le proprietà dei link
- ▶ Implementa la possibilità di introdurre servizi di uso comune all'interno della rete su tutti i livelli dello stack
- ▶ Per funzionare, su Linux, usa i namespaces, OpenvSwitch e il software Docker

# Come avviare IMUNES in laboratorio

- ▶ Il comando per avviare IMUNES nel laboratorio didattico è

**\$ sudo reti**

- ▶ Sulle macchine di laboratorio non abbiamo privilegi di root
- ▶ Possiamo usufruire, tramite il comando «sudo», di un livello di accesso privilegiato solo per alcuni comandi utili al corso



# Come installare IMUNES sul proprio PC - 1

## ► Requisiti

- Sistema operativo Linux-based (se non lo avete già installato, è consigliato l'utilizzo di una macchina virtuale)
- Git
- Docker
- Open vSwitch
- Wireshark
- Librerie: TCL/TK > 8.6, ImageMagick
- Strumenti: util-linux (nsenter), xterm, make

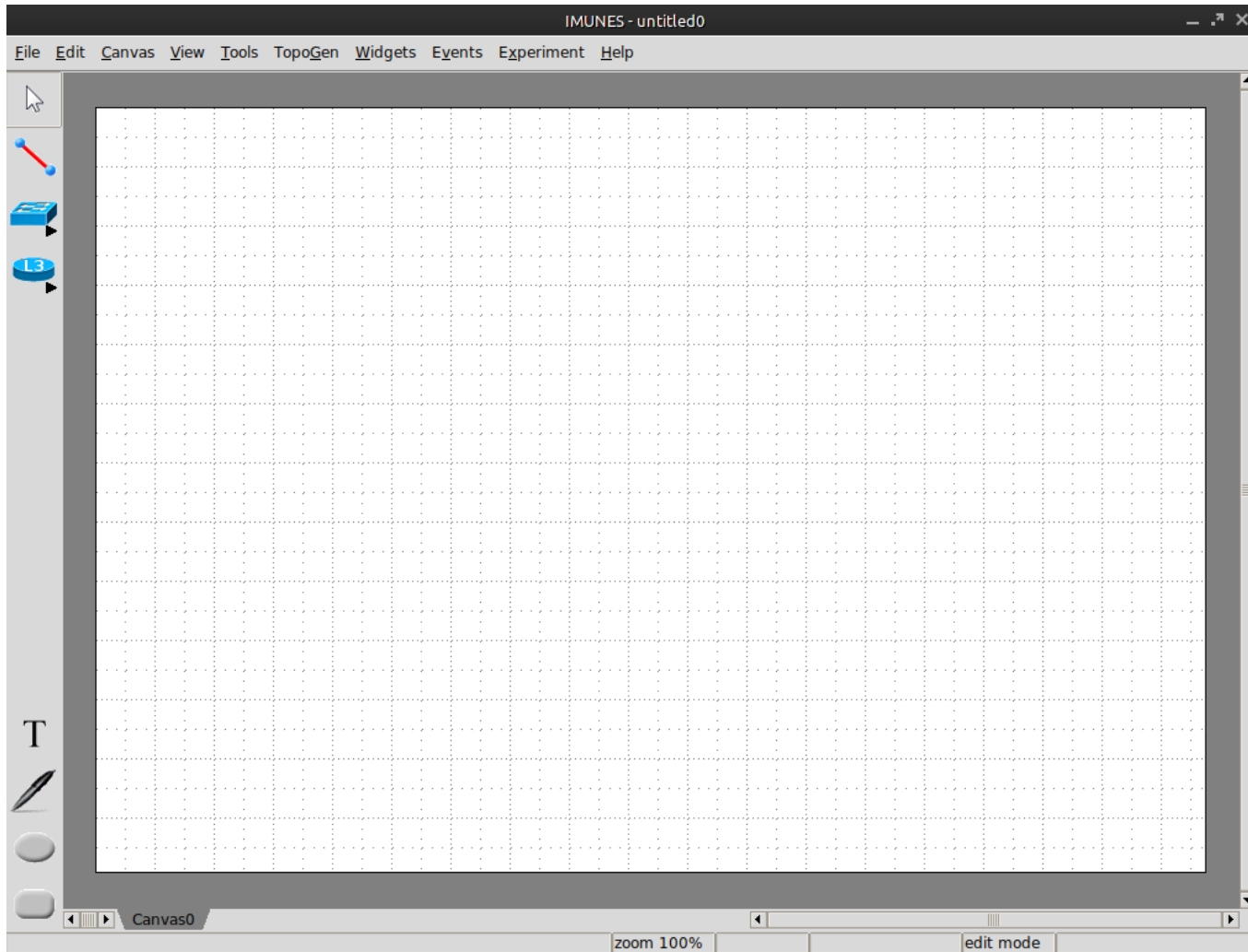
# Come installare IMUNES sul proprio PC - 2

- ▶ Preparazione dell'ambiente su Ubuntu 14.04 LTS
  - ▶ `$ sudo apt-get install git openvswitch-switch xterm wireshark \ make ImageMagick tk tcllib user-mode-linux util-linux`
  - ▶ `$ wget -qO- https://get.docker.com/ | sudo sh`
  - ▶ `$ service docker start`
  - ▶ `$ sudo docker run -v /usr/local/bin:/target jpetazzo/nsenter`
- ▶ Preparazione dell'ambiente su Ubuntu 15.04 e successivi
  - ▶ `$ sudo apt-get install git openvswitch-switch docker.io xterm \ wireshark make ImageMagick tk tcllib user-mode-linux util-linux`
- ▶ Preparazione dell'ambiente su altre distribuzioni Linux
  - ▶ Fare riferimento a <https://github.com/imagenes/imagenes>

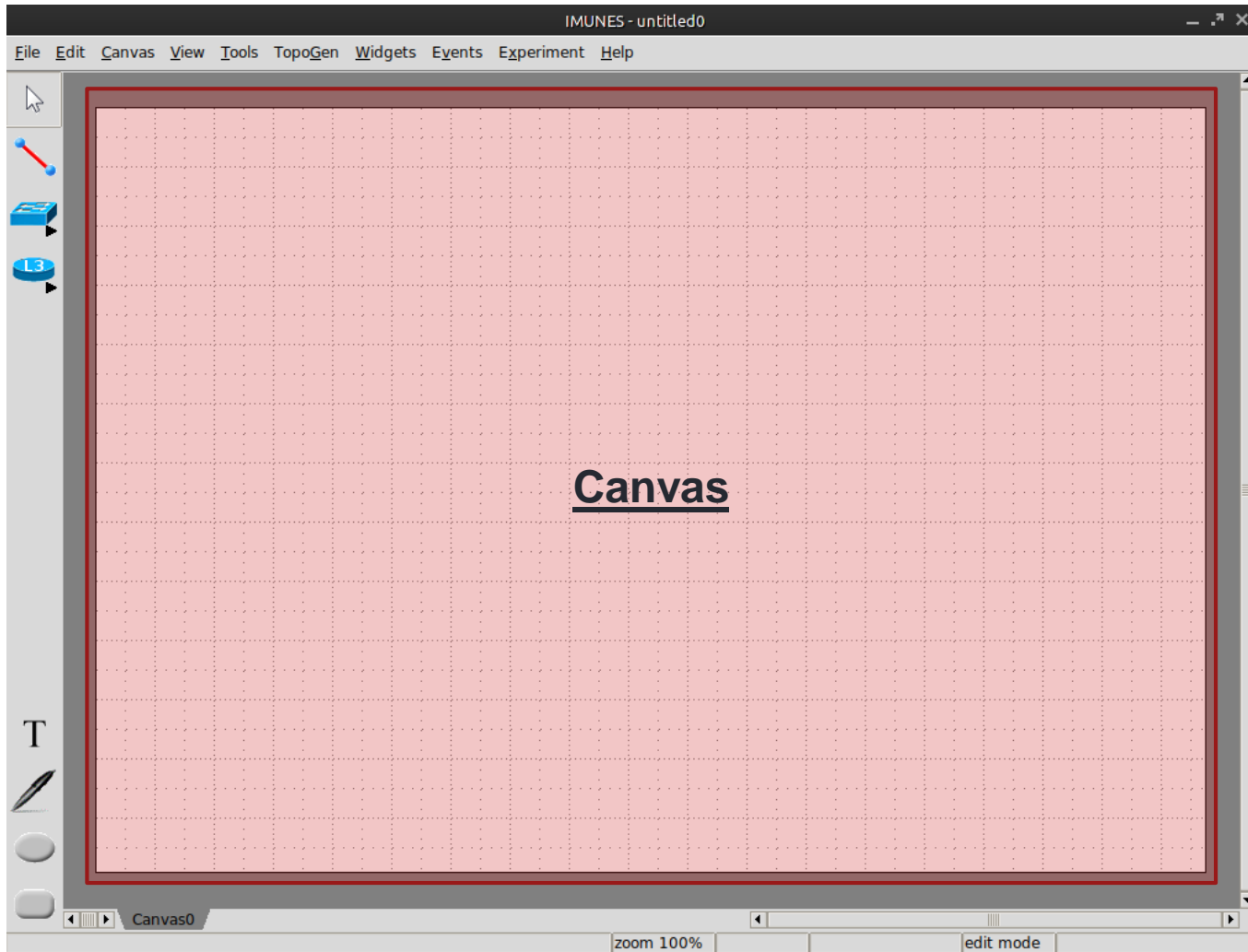
# Come installare IMUNES sul proprio PC - 3

- ▶ Installazione di IMUNES vera e propria
  - ▶ `$ git clone https://github.com/imunes/imunes.git`
  - ▶ `$ cd imunes`
  - ▶ `$ sudo make install`
- ▶ Ottimizzazione necessaria
  - ▶ Ubuntu:
    - ▶ `$ echo 'DOCKER_OPTS="-s overlay"' | sudo tee /etc/default/docker`
    - ▶ `$ sudo service docker restart`
- ▶ Bisogna scaricare inoltre le immagini dei container Docker
  - ▶ `$ sudo imunes -p`
- ▶ Per avviare Docker
  - ▶ `$ sudo imunes`

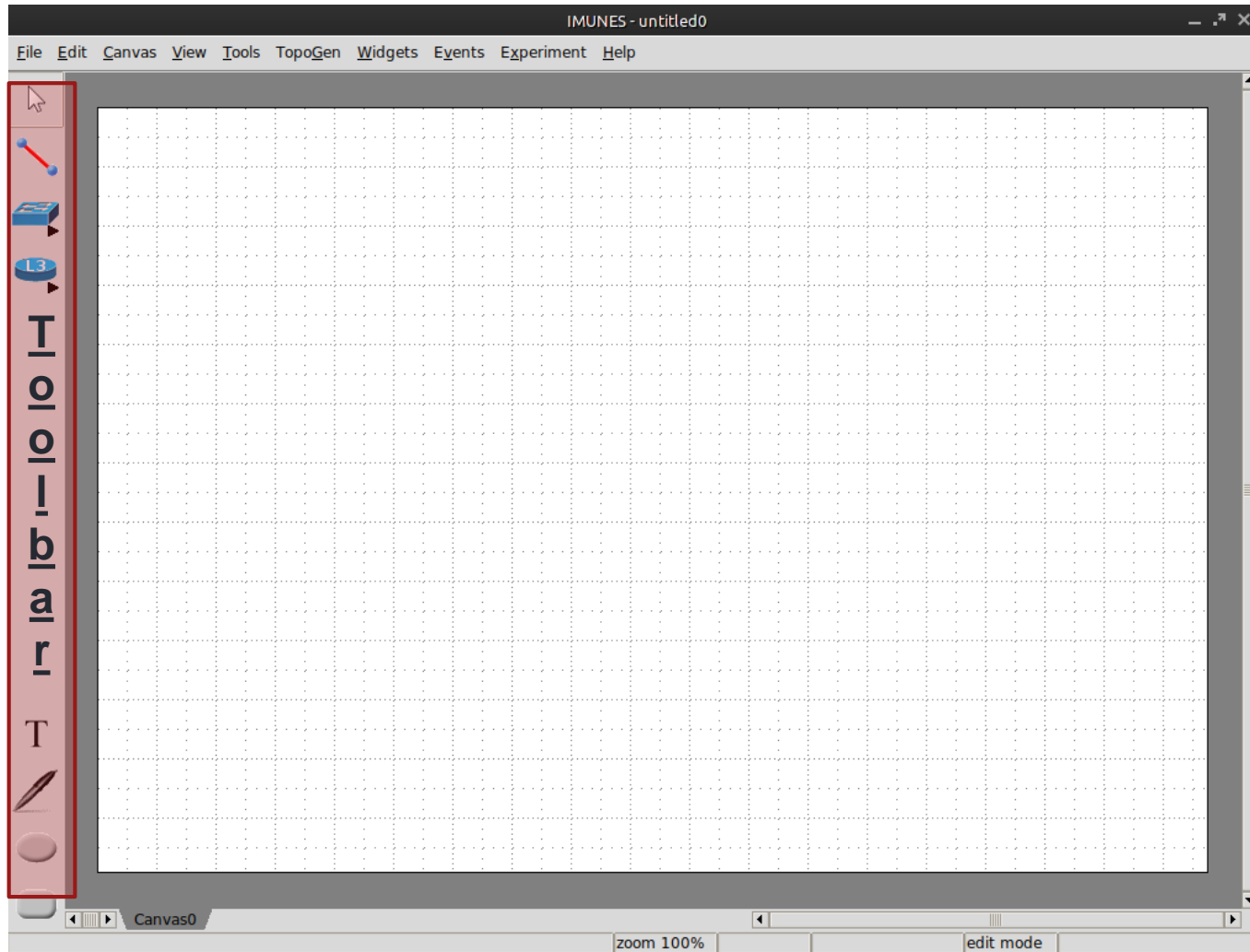
# Interfaccia di IMUNES



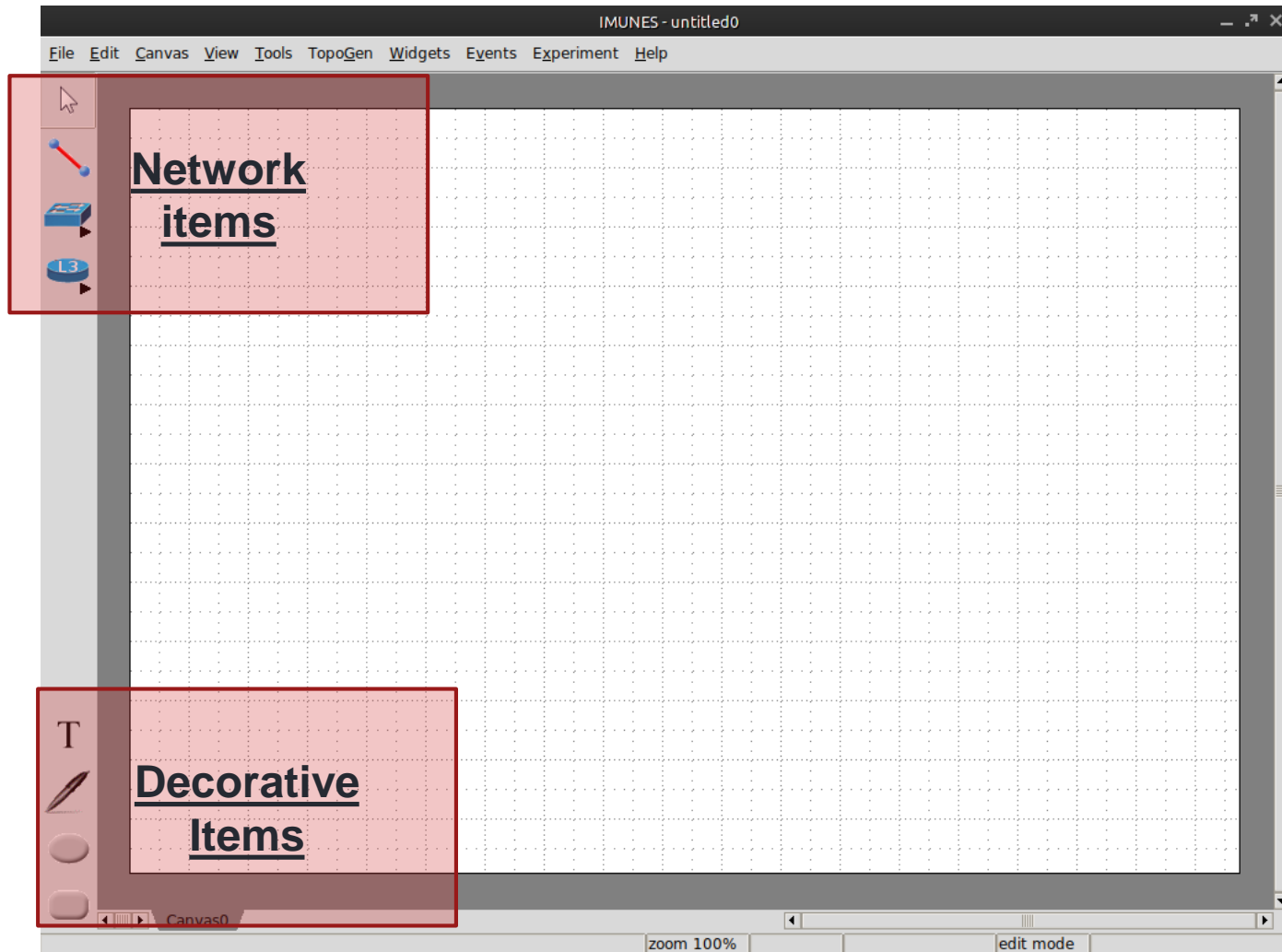
# Interfaccia di IMUNES



# Interfaccia di IMUNES

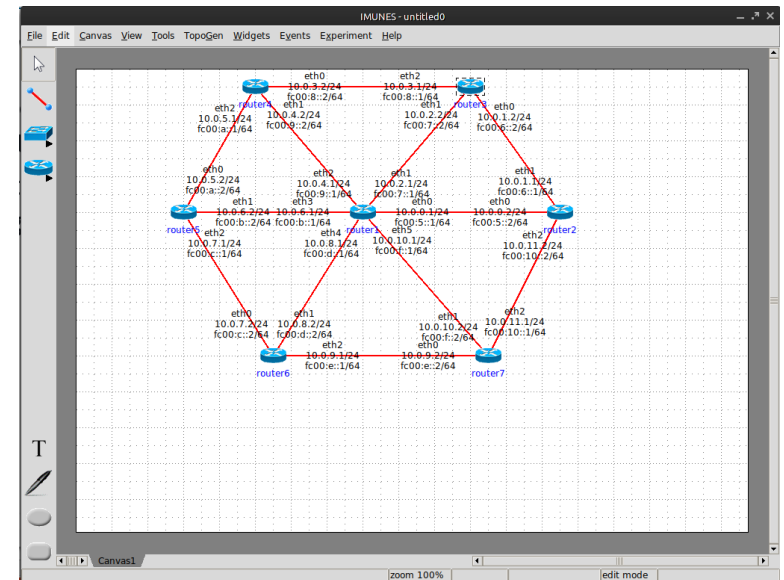


# Interfaccia di IMUNES



# IMUNES: Topology generator

- ▶ IMUNES è in grado inoltre di generare una topologia di rete geometrica composta da elementi omogenei con un click, tramite il menù «TopoGen»
- ▶ Questo consente di simulare in tempo reale scenari basati su configurazioni note
  - ▶ Ad esempio, generare una rete di router organizzati con topologia a «ruota» (wheel)





# IMUNES: Gestore degli eventi - 1

- ▶ IMUNES consente inoltre di gestire degli eventi monitorando il link tra due nodi
  - ▶ Questo consente di effettuare simulazioni automatizzando dei cambiamenti sulle proprietà dei link
- ▶ Gli eventi sono gestiti tramite l'Event Editor
- ▶ Ogni evento è associato a un link ed è espresso nel formato  
`{deadline} {target} {function} {parameter}`

# IMUNES: Gestore degli eventi – 2

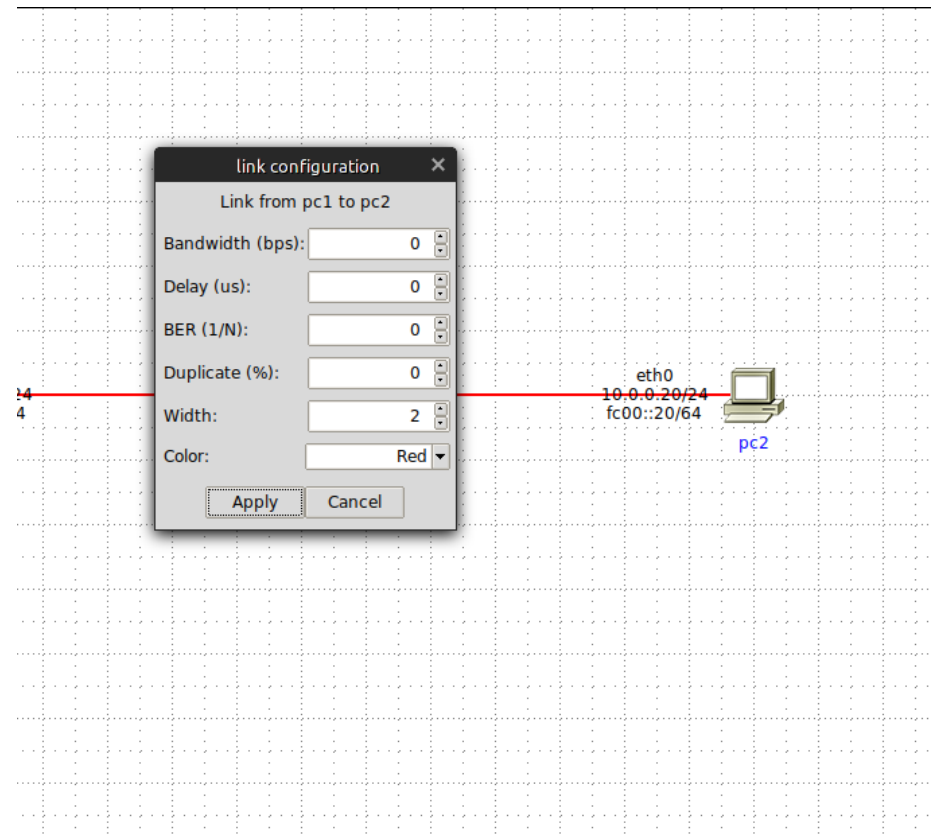
- ▶ Ogni evento è associato a un link ed è espresso nel formato  
    {deadline} {target} {function} {parameters}
- ▶ «Deadline» rappresenta il tempo  $t$  nel quale l'evento viene eseguito. È espresso in secondi, e specificato rispetto al tempo di lancio dell'esperimento
- ▶ «Target» è il target che viene modificato dall'evento. Può essere «bandwidth», «delay» «ber», «duplicate», «width», «color»
- ▶ «Function» è una funzione che descrive il modo in cui l'evento viene modificato. Può essere «const», «ramp», «rand», «square»
- ▶ «Parameters» sono i parametri da passare in input alla funzione
- ▶ Maggiori informazioni possono essere trovate nella documentazione di IMUNES all'indirizzo <http://imunes.net/dl/guide>



# IMUNES: Modifica delle proprietà dei link

Possiamo personalizzare

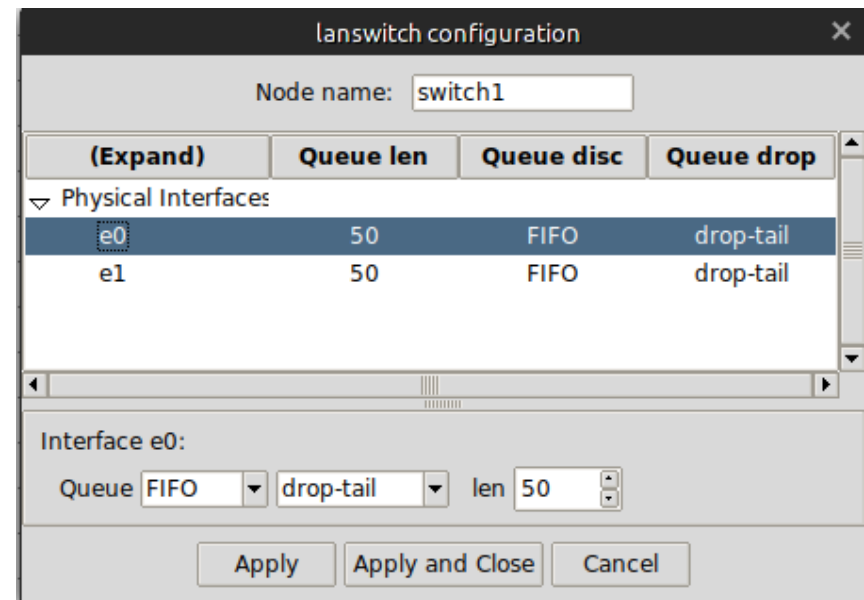
- ▶ Bandwidth (larghezza di banda, quindi velocità del canale di comunicazione)
- ▶ Delay (ritardo nella trasmissione della comunicazione)
- ▶ BER (Bit Error Ratio, tasso di errore, calcolato come rapporto di bit non ricevuti correttamente su bit trasmessi)
- ▶ Duplicate (Percentuale di pacchetti duplicati)
- ▶ Personalizzazioni stilistiche
  - ▶ Spessore della linea
  - ▶ Colore della linea



# IMUNES: Modifica delle proprietà dei nodi L2 (switch)

Possiamo personalizzare

- ▶ Nome (Nome simbolico associato allo switch, non ha alcuna valenza tecnica)
- ▶ Proprietà delle interfacce (generate dinamicamente) dello switch
  - ▶ Queue length
    - ▶ Lunghezza della coda di trasmissione
  - ▶ Queue disc
    - ▶ Scheduler per la selezione dei pacchetti da trasmettere
      - ☐ First In First Out
      - ☐ Weighted Fair Queueing
      - ☐ Deficit-Weighted Round Robin
  - ▶ Queue drop
    - ▶ Algoritmo di gestione della coda
      - ☐ Drop-tail
      - ☐ Drop-head
- ▶ Non è possibile assegnare VLAN tag ☹
  - ▶ Ma niente paura! Faremo ugualmente esercizi sulle VLAN!



# IMUNES: Modifica delle proprietà dei nodi L3 (host e pc)

Host e PC sono la stessa cosa, hanno solo diverso valore semantico

Possiamo personalizzare

- ▶ Nome del nodo
- ▶ Utilizzo delle funzionalità di rete di Docker
- ▶ Servizi
  - ▶ SSH, TcpDump, Ftp, Telnet
- ▶ Rotte statiche
- ▶ Script di configurazione da eseguire all'avvio

pc configuration

Node name: pc1

Configuration Interfaces

Attach external docker interface: ☐ enabled

Services: ☐ ssh ☐ tcpdump ☐ ftp ☐ telnet

Static routes:

Custom startup config: ☐ Enabled Editor

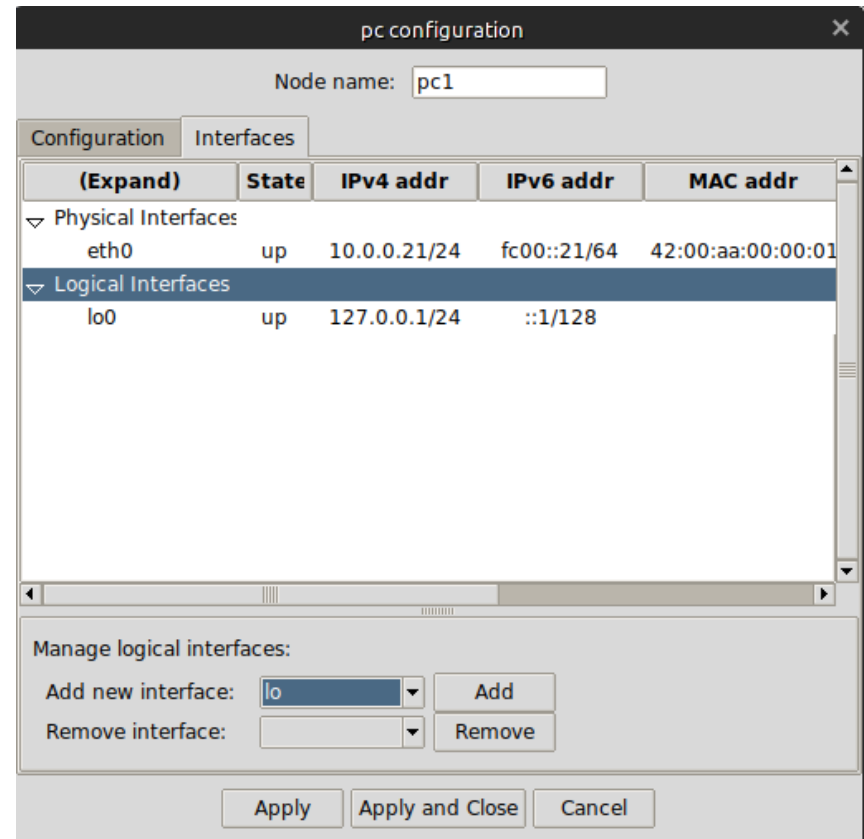
Selected custom config:

Apply Apply and Close Cancel

# IMUNES: Modifica delle proprietà dei nodi L3 (host e pc) - 2

## Interfacce

- ▶ Possiamo preconfigurare le interfacce da qui
  - ▶ In un primo momento effettueremo esercizi manuali, tralasciando questa parte. Andando avanti nel corso ci dedicheremo ai livelli più alti dello stack ISO/OSI, dando per scontata la configurazione di rete e utilizzando questi strumenti
- ▶ Esistono due tipi di interfacce
  - ▶ Fisiche
  - ▶ Logiche
    - ▶ Interfaccia di loopback
    - ▶ Eventuali VLAN a livello host



# IMUNES: Modifica delle proprietà dei nodi L3 (router)

Un router è un host «particolare» che può effettuare l'instradamento dei pacchetti

Possiamo personalizzare

- ▶ Nome (nome simbolico associato al router)
- ▶ Modello di router
  - ▶ Quagga
    - ▶ RIP / RIP-NG / OSPF 2 / OSPF 3
  - ▶ Static
- ▶ Servizi attivi
  - ▶ SSH / TCPdump / FTP / Telnet
- ▶ Rotte statiche
- ▶ Configurazioni personalizzate all'avvio del router
  - ▶ Le vedremo in dettaglio più avanti

router configuration

Node name: router1

Configuration Interfaces IPsec

Model: ☒ quagga ☐ xorp ☐ static

Protocols: ☒ rip ☒ ripng ☐ ospfv2 ☐ ospfv3

Attach external docker interface: ☐ enabled

Services: ☐ ssh ☐ tcpdump ☐ ftp ☐ telnet

Static routes:

Custom startup config: ☐ Enabled

Selected custom config:

Editor

Apply Apply and Close Cancel

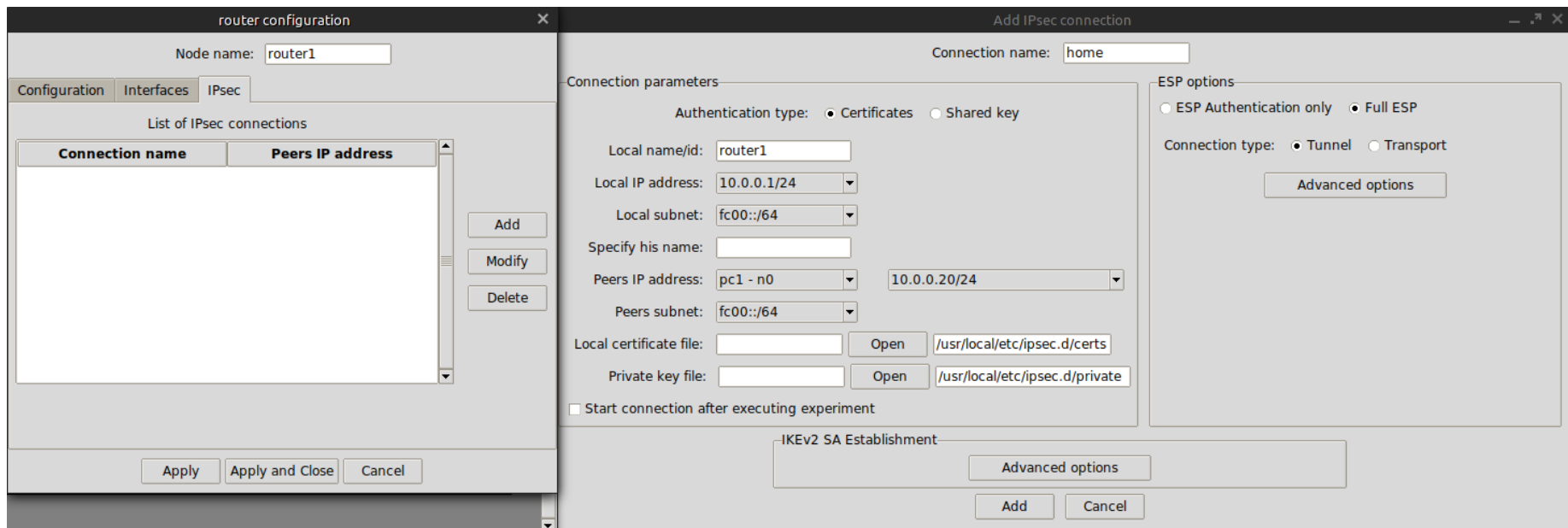
# Router: piccola parentesi!

- ▶ Un router non è altro che un dispositivo che è in grado di effettuare il forwarding dei pacchetti, ovvero di «instradare» un pacchetto verso un host appartenente ad una rete diversa (anche eterogenea) rispetto a quella in cui la comunicazione sta avvenendo
- ▶ Ogni host, computer, server può avere il ruolo di Router
- ▶ Sulla maggior parte dei router domestici c'è Linux, ed essi **hanno esattamente lo stesso funzionamento** di un router IMUNES!
- ▶ Per «trasformare» qualunque host Linux in un router, basta abilitare il flag di IP forwarding sul kernel tramite il comando `sysctl`
  - ▶ `# sysctl net.ipv4.ip_forward=1`
  - ▶ `# sysctl net.ipv6.conf.all.forwarding=1`



# IMUNES: Modifica delle proprietà dei nodi L3 (router) - 3

- Sui router è inoltre possibile dichiarare dei tunnel IPsec per la realizzazione di VPN all'interno della topologia



# IMUNES: Simulazione di rete - 1

- ▶ La simulazione del funzionamento di una rete può essere eseguito tramite il menù Experiment
- ▶ Lo stesso menù consente anche di terminare l'esperimento. **Attenzione: tutti i cambiamenti effettuati durante l'esecuzione dell'esperimento verranno persi alla terminazione dello stesso**
- ▶ Nel caso IMUNES venga chiuso durante l'esecuzione dell'esperimento è possibile riprendere il controllo dello stesso tramite il menù «Experiment»/ « Attach to experiment»

# IMUNES: Simulazione di rete - 2

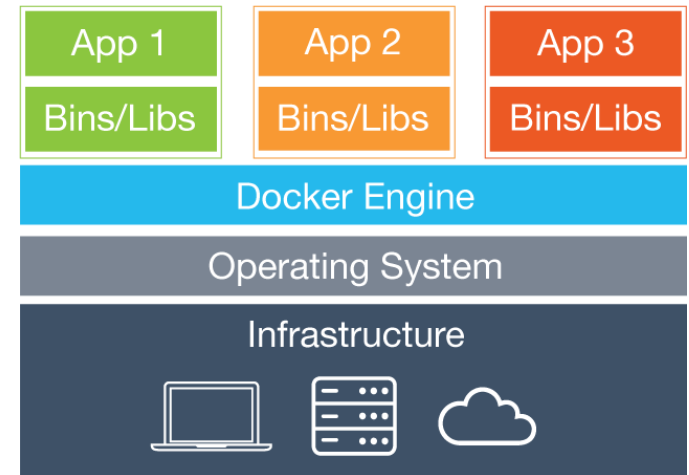
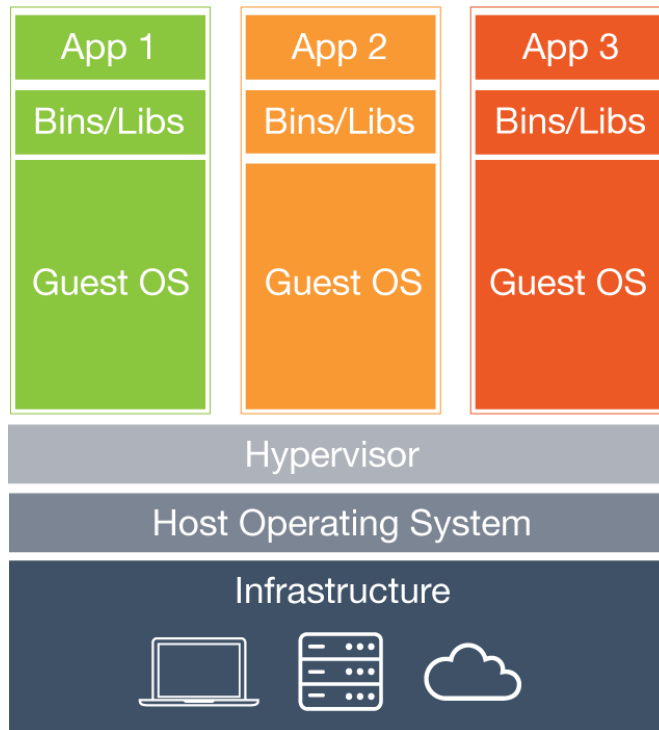
- ▶ In fase di esecuzione dell'esperimento la nostra topologia prende vita
- ▶ Facendo doppio click sui link, si possono modificare le proprietà del canale di comunicazione
- ▶ Facendo doppio click sui nodi di livello 3, si può accedere al terminale relativo al nodo
- ▶ Ogni nodo è quindi separato dall'altro, come se i dispositivi simulati fossero effettivamente delle macchine indipendenti (**docker**)

# IMUNES: Come funziona?

- ▶ IMUNES è un orchestratore
- ▶ La sua attività è basata sulla gestione di due componenti
  - ▶ OpenVSwitch (switch implementato)
  - ▶ Docker (software che automatizza il deployment delle applicazioni, chiudendole in un «container»)
    - ▶ CONCETTO DI CONTAINER
      - Il container è un «contenitore» minimale di applicazioni
      - Il contenimento dell'applicazione è basato su due funzionalità del kernel:
        - Segregazione del filesystem
        - Namespace di rete

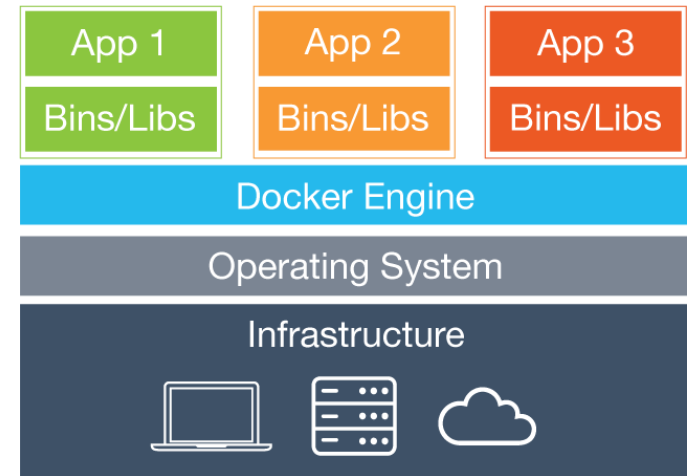
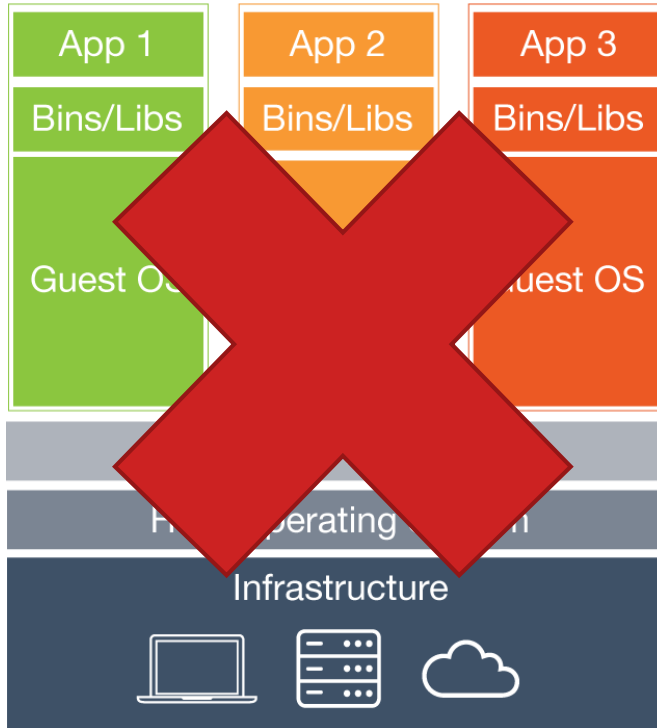


# IMUNES: Come funziona? - Docker



- Contiene lo stretto necessario per eseguire una singola applicazione o servizio
- IMUNES li usa in modo leggermente anomalo, trattandoli come delle piccole virtual machine, ma NON SONO delle virtual machine!

# IMUNES: Come funziona? - Docker



- Contiene lo stretto necessario per eseguire una singola applicazione o servizio
- IMUNES li usa in modo leggermente anomalo, trattandoli come delle piccole virtual machine, ma NON SONO delle virtual machine!

# Segregazione del filesystem: chroot

- ▶ La segregazione del filesystem è ottenuta tramite il comando chroot (change root)
- ▶ La «change root» è una funzionalità del kernel che serve a cambiare la directory di riferimento dei processi in esecuzione, e per i sottoprocessi da essi generati
- ▶ Nei sistemi Unix serve a isolare i limiti operativi di un'applicazione
- ▶ In molti sistemi è utilizzato come uno strumento di sicurezza ma NON nasce come uno strumento di sicurezza
- ▶ Questo ha portato dapprima alla definizione del concetto di «jail» (sistemi FreeBSD)

# Segregazione dello stack di rete: namespaces

- ▶ Analogamente, si può segregare lo stack di rete a livello kernel tramite l'utilizzo di «namespace»
- ▶ Un «namespace» è uno spazio dei nomi, il significato, cambiando ambito, è analogo ai namespace di C++ o XML, oppure al «package» di Java
- ▶ È, banalmente, un contesto nel quale il processo nasce ed evolve. Un processo chiuso all'interno di un namespace, può conoscere solo le risorse dichiarate all'interno di quel namespace.
- ▶ In questo caso le risorse sono risorse di rete (interfacce, tabella di routing, regole sul firewall)
- ▶ La funzionalità è gestita tramite il comando **ip netns** che abbiamo visto la volta scorsa





# IMUNES – Salvataggio e reload delle modifiche all'esperimento - 1

- ▶ IMUNES supporta il salvataggio delle topologie, tramite il menù «*File / Save as*» ma attenzione:

**Il salvataggio dell'esperimento (istanza di una topologia) non è supportato. Qualunque cambiamento effettuato all'interno degli host (creazione file, configurazioni etc.) verrà perso premendo su «Terminate experiment» o spegnendo la macchina.**

**Tuttavia...**

# IMUNES – Salvataggio e reload delle modifiche all'esperimento – 2

... potete usare il comando «hcp» per copiare file da/nei container e il comando «himage» per lanciare comandi nei container.

Inoltre abbiamo sviluppato due script:

uno per salvare i cambiamenti effettuati all'interno degli host (quindi file modificati, creati, ecc.) e un altro per ricaricare i cambiamenti eseguiti.

Questi script, e le istruzioni per utilizzarli, sono disponibili all'indirizzo:

<https://blog.tufarolo.eu/imunes-export-import/>

Gli script saranno utilizzabili in laboratorio dalla prossima lezione tramite i comandi:

```
$ sudo imunes-export
```

```
$ sudo imunes-import
```



# Esercizio 1

- ▶ Iniziare da una topologia vuota
- ▶ Creare due PC
- ▶ Collegare i due PC con un link
- ▶ Aprire il terminale del PC1 ed effettuare un ping verso il PC2
- ▶ Aprire il terminale del PC2 ed effettuare un ping verso il PC1

## Esercizio 2

- ▶ Iniziare da una topologia vuota
- ▶ Creare due PC
- ▶ Collegare i due PC con un link
- ▶ Disabilitare l'auto assegnamento degli indirizzi IP togliendo il tick ai menù Tools / IPv4 auto-assign address e Tools / IPv6 auto-assign address
- ▶ Utilizzare la commandistica di rete appresa nella scorsa lezione per assegnare gli indirizzi IP alle macchine affinché appartengano alla stessa sottorete
- ▶ Lanciare un Ping per vedere se le due macchine comunicano

# Commandistica di rete per Linux

- ▶ `ip link` – gestisce l'interfaccia a livello 2
  - ▶ `ip link show dev {device}`
    - ▶ Mostra le informazioni di livello 2 per un device
  - ▶ `ip link set {up|down} dev {device}`
    - ▶ Imposta lo stato del link
- ▶ `ip addr` – gestisce l'interfaccia a livello 3
  - ▶ `ip addr show dev {device}`
    - ▶ Mostra le informazioni di livello 3 per un device
  - ▶ `ip addr {add|del} {ip | ip/cidr} [netmask {netmask}] dev {device}`
    - ▶ Aggiunge/rimuove un indirizzo IP a un device

# Conclusioni

- ▶ Abbiamo esplorato le caratteristiche del software IMUNES
- ▶ Abbiamo imparato a creare una topologia di rete simulata e ad eseguire un esperimento
- ▶ Abbiamo imparato a mettere in comunicazione due host su IMUNES a livello IP utilizzando la commandistica Linux appresa nella lezione precedente



