Sistema de comunicaciones seguras con segmentación virtual de dominios

Avance de proyecto

Alberto Daniel Lange

Dirección: Juan Ignacio Vaccarezza
Codirección: Santiago Pérez Ghiglia

Ingeniería en Telecomunicaciones
Instituto Balseiro

26 de febrero de 2025

Contexto

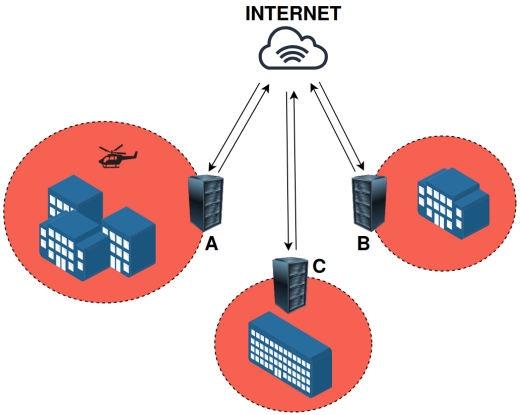

Figura 1: Esquema simplificado de operación del sistema.

Progreso alcanzado anteriormente

- Definición y validación de arquitectura lógica en laboratorio virtual.

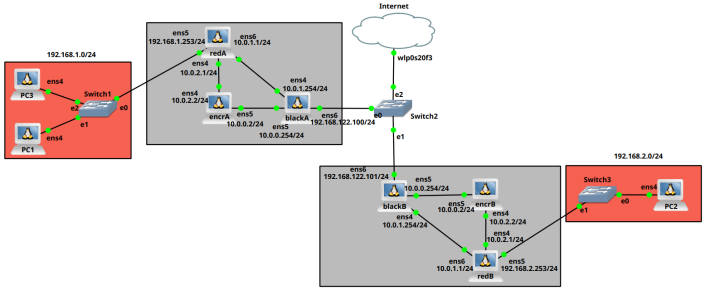

Figura 2: Implementación de la propuesta de solución en GNS3.

Pendientes

- Validación del encriptador en ambiente virtualizado.
- Implementación sobre hardware.

seL4

- Microkernel de código abierto.
- Formalmente probado.
- Hipervisor tipo 1.
- Aislamiento garantizado entre componentes.

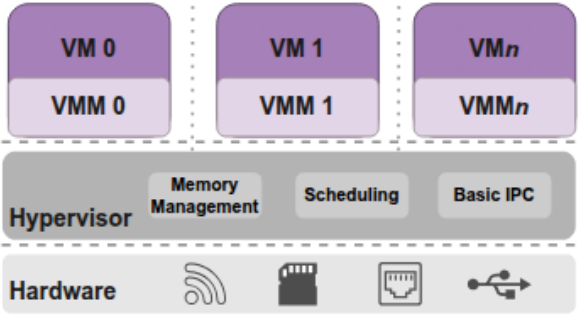

Figura 3: Arquitectura de seL4. (algo así, otra imagen)

CAmkES

- **C**omponent **A**rchitecture for **m**icro**k**ernel-based **E**mbedded **S**ystems.
- Framework de desarrollo para seL4.
- Arquitectura basada en componentes.
- Comunicación mediante interfaces bien definidas.

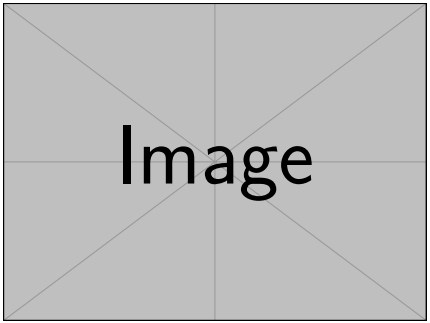

Figura 4: Arquitectura de componentes CAmkES.

Modelo minimal_64

- Ejemplo básico de CAmkES.
- Implementación mínima de una VM para arquitectura de 64 bits.
- Punto de partida para el desarrollo.

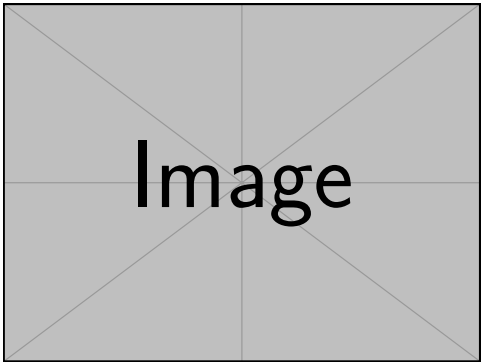

Figura 5: Arquitectura minimal_64.

Modelo zmq_samples

- Comunicación entre máquinas virtuales usando ZeroMQ.
- Base para la implementación del encriptador.

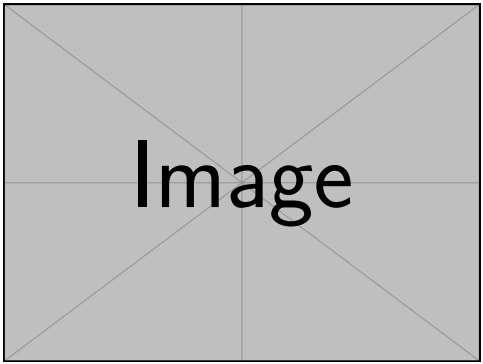

Figura 6: Arquitectura zmq_samples.

Estrategia de modelos en entornos virtualizados

- **¿Qué?**: Realizar modelos que validen progresivamente los componentes desarrollados.
- **¿Para qué?**:
  - Ligar problemas concretos a cada modelo y resolverlos de forma independiente.
  - Implementar un encriptador funcional en un entorno virtualizado como paso previo a su despliegue en hardware.

Modelo I: Introduciendo la arquitectura lógica

- Validar la arquitectura lógica de tres VMs.
- Funcionalidad *split-tunneling*.

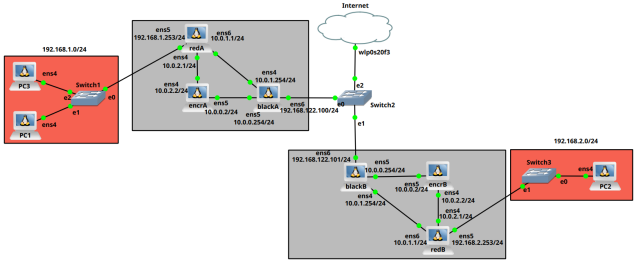

Figura 7: Arquitectura lógica en GNS3.

Modelo II: comunicando sitios con WireGuard

- Validación del kernel Linux 4.9 con soporte para WireGuard.
- Obtener parámetros PCI necesarios para el passthrough de la interfaz de red.

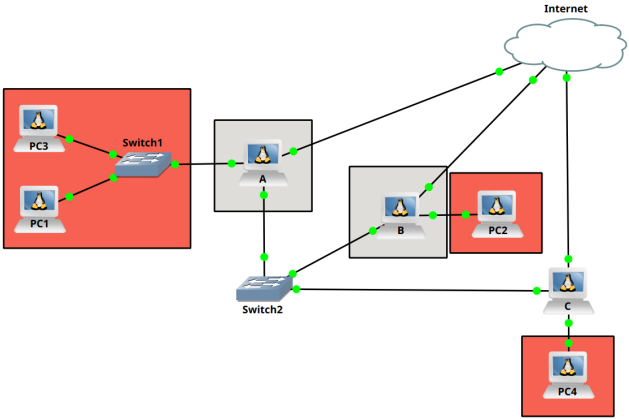

Figura 8: Arquitectura lógica en GNS3.

Modelo II: comunicando sitios con WireGuard - Sistema operativo VMs

- Buildroot como sistema de archivos.
- Kernel Linux 4.9.337.
- Parche de compatibilidad WireGuard con Linux ≤ 5.6.

Modelo III: utilizando seL4 como hipervisor

- Validación del passthrough de hardware y la compatibilidad del kernel Linux 4.9 con seL4.

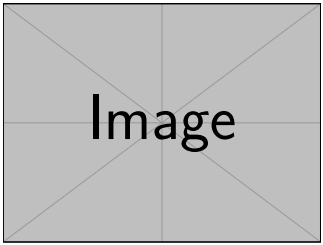

Figura 9: Algo

Modelo III: utilizando seL4 como hipervisor - Passthrough de hardware

CAmkES actúa de interfaz para la configuración del passthrough de dispositivos PCI a una VM a través de los Base Address Registers (BARs) e interrupciones.

Ejemplo de configuración en CAmkES:

```
vm0.vm_ioports = [start, end];
vm0.pci_devices = [bus, device, function, memory];
vm0.vm_irqs = [source, dest];
```

Modelo III: utilizando seL4 como hipervisor - Gestión de memoria VMs

`simple_untypedN_pool`: prealocación de memoria para las VMs.
`heap_size`: tamaño del heap del VMM.
`guest_ram_mb`: tamaño de la RAM asignada a cada VM.

Modelo IV: implementando el encriptador en seL4

- Solución completa como paso previo a la implementación en hardware.

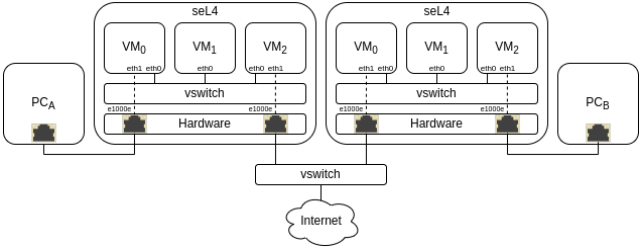

Figura 10: Algo.

Modelo IV: Encriptador en seL4 - Comunicación entre VMs

¿Cómo funciona?

Virtio-Net en VM

- El host implementa un dispositivo PCI virtual (virtio-net).
- La VM encuentra este dispositivo y carga el driver virtio-net.
- En la transmisión, la VM escribe los datos a enviar en un buffer asociado a una cola del hipervisor. El VMM notifica al host que hay datos listos en la cola (virtqueue).

Virtio-Net en host

- El host lee el paquete de la virtqueue y lo copia a un buffer dentro del espacio de memoria de la VM destino.
- El host inyecta una interrupción en la VM destino para notificarle que hay datos disponibles.

Modelo IV: Encriptador en seL4 - Integración

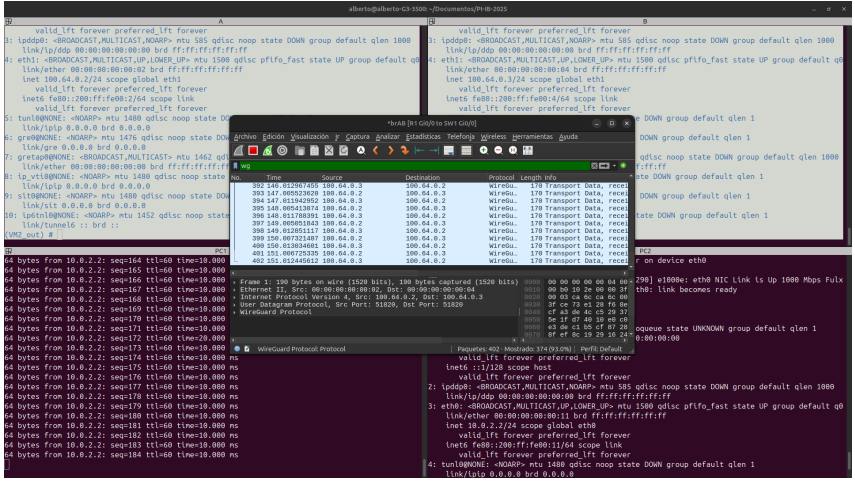

Figura 11: Ping entre PCs.

Implementación en hardware

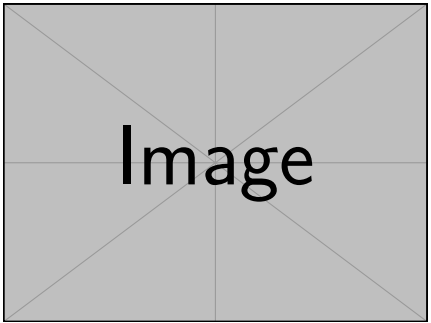

Figura 12: SuperMicro SYS-E300-9D.

Implementación en hardware

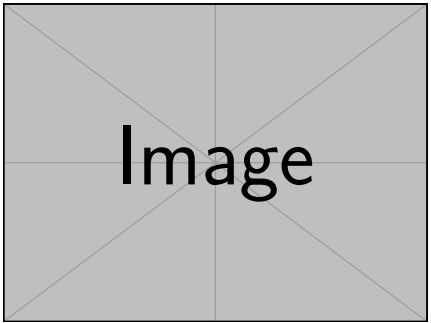

Figura 12: SuperMicro SYS-E300-9D.

Desafíos:

- ✓ Redirección de consola.
- × Passthrough de controlador Ethernet.
- × Throughput entre VMs.

Conclusiones

- Algo

¡Muchas gracias!
¿Preguntas?