Universidad Zaragoza
1542

Trabajo de Fin de Grado

# MITIGACIÓN DE ATAQUES DE CANAL LATERAL BASADOS EN CARACTERIZACIÓN TÉRMICA Y ELÉCTRICA


AUTOR

Javier Corbalán Colino

DIRECTORES

Darío Suárez Gracia
Alejandro Valero Bresó



ESCUELA DE INGENIERÍA Y ARQUITECTURA

2019

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

# DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

D./Dª. Javier Corbalán Colino ,en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza, Declaro que el presente Trabajo de Fin de (Grado/Máster) Mitigación de ataques de canal lateral basados en monitorización térmica y eléctrica (Título del Trabajo)

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, a 28 de agosto de 2019

Fdo: Javier Corbalán Colino

# RESUMEN

Los ataques de canal lateral se han convertido en una vulnerabilidad de creciente importancia en las plataformas multinúcleo y, especialmente, en entornos cloud. Estos ataques aprovechan vulnerabilidades de la implementación física de un sistema en lugar de actuar sobre los algoritmos que ejecutan.

Este trabajo se centra en la monitorización térmica, energética y, sobre todo, en la transmisión a través de canales laterales térmicos. La transmisión térmica se basa en que un núcleo puede inducir cambios de temperatura en los de su alrededor, lo cual puede ser utilizado para transmitir información sigilosamente entre contenedores aislados y desplegados en núcleos físicos diferentes.

Con el objetivo de una mayor comprensión de estos ataques, así como para diseñar medidas de mitigación efectivas, se lleva a cabo una caracterización experimental y exhaustiva del ataque sobre un servidor multinúcleo real como los empleados por cualquier proveedor cloud. De esta manera, se identifica y se analiza cuantitativamente el impacto de los factores principales que afectan al éxito del ataque, como es el caso de la elección de los núcleos emisores y receptores en la creación del canal, la tasa de transmisión de datos o el nivel de carga de trabajo en el sistema. El ataque se ha completado con éxito total en máquinas reales, con virtualización por contenedores y parcialmente en máquinas con virtualización de sistema.

Partiendo de la caracterización anterior, se implementa una prueba de concepto para demostrar su impacto y su efectividad como método de extracción de información sensible. Finalmente, se desarrollan y evalúan módulos del kernel de Linux que mitigan por completo el ataque.

# ABSTRACT

Side-channel attacks have become a vulnerability of increasing relavance in multicore platforms, particularly in cloud environments. These attacks exploit vulnerabilities of the physical implementation of a system instead of focusing on the executed algorithms.

This essay focuses on thermal, power monitoring and, mostly, transmission through thermal side-channels. A thermal transmission is based on the fact that a core can induce temperature changes on those around it, which can be used as a way to covertly transmit information between isolated containers deployed in different physical cores.

In order to better understand these attacks, as well as to design effective mitigation techniques, an experimental and exhaustive characterization of such an attack is carried out on a real multicore server such as those used by any cloud provider. This way, the impact of the main factors that affect the success of the attack is identified and quantitatively analyzed, so is the case for the choice of the source and sink cores on the creation of the channel, the data transfer rate or the workload level in the system.

Based on the previous characterization, a proof of concept is implemented to demonstrate its impact and effectiveness as a sensitive data exfiltration technique. Finally, Linux kernel modules that completely mitigate the attack are developed and evaluated.

*Nadie acaba descifrando de qué trata la vida,*
*y no importa. Explora el mundo.*
*Casi todo es verdaderamente interesante*
*si ahondas en ello lo suficiente.*

**— Richard P. Feynman**

## AGRADECIMIENTOS

A mis directores: a Darío por tantas preguntas, a Alejandro por tantas respuestas y sobre todo a Diego por su silencio, que pronto será palabras. A mi familia por dármelo todo y estar ahí incondicionalmente. Cada logro mío será siempre suyo. A Marta, por su inestimable ayuda y por no dejarme parar de crecer como persona desde que la conocí. A Luis por la perspectiva y la templanza del que ha recorrido el camino antes que yo. A Zaragoza por el frío y a mis amigos por el calor.

A Paqui por marcharse habiéndolo dicho todo y a Ramón por irse sin decir nada.

Para todos ellos, no tengo más que gratitud.

# ÍNDICE GENERAL

# ÍNDICE DE FIGURAS

## ÍNDICE DE TABLAS



## ACRÓNIMOS

| | |
|---|---|
| TDP | Thermal Design Power |
| MV | Máquina Virtual |
| SO | Sistema Operativo |
| SPA | Simple Power Analysis |
| DES | Data Encryption Standard |
| DPA | Differential Power Analysis |
| RSA | Rivest, Shamir y Adleman |
| CCN | Centro Criptológico Nacional |
| BMC | Board Management Controller |
| GITSE | Grupo de Ingeniería Térmica y Sistemas Energéticos |
| PID | Proporcional, Integral y Derivativo |

RAPL Running Average Power Limit

SHA1 Secure Hashing Algorithm 1

SIMD Single Instruction, Multiple Data

HCC High Core Count

SOC Security Operations Center

IMC Integrated Memory Controller

ASHRAE American Society of Heating, Refrigerating and Air-Conditioning Engineers

vCPU Virtualized CPU

# 1 INTRODUCCIÓN

*Este capítulo introduce la motivación, objetivos, alcance y tareas principales que componen el presente trabajo, así como una breve descripción de la estructura general de la memoria.*

## 1.1 MOTIVACIÓN

La seguridad informática es una disciplina con creciente importancia. La información y los sistemas informáticos son activos que cada vez toman un porcentaje mayor del total de los haberes de las empresas. Otra tendencia creciente es que las empresas deleguen su infraestructura en grandes proveedores de cloud públicos, que a su vez se apoyan en tecnologías como la virtualización para proveer alta disponibilidad y elasticidad a un coste asequible.

*El departamento de defensa (DoD) estadounidense ha sacado a concurso público la adjudicación de un contrato por valor de 10 billones de dólares para una infraestructura cloud de defensa: JEDI*

Sin embargo, la sensibilidad de la información almacenada en la nube crece, lo que la convierte en un objetivo más lucrativo para ciberdelincuentes. Es por esta razón que los proveedores de cloud deben garantizar en todo momento el aislamiento entre los servicios de diferentes clientes, incluso en instancias virtualizadas que coexisten en la misma máquina física. Por otra parte, la complejidad creciente de los sistemas en los que se ejecutan estas máquinas virtuales, la mejora en su monitorización y la colocalidad espacial de la información sensible son caldo de cultivo para ataques de canal lateral. Estos ataques consisten en transmitir información a través de canales de comunicación accidentales como consecuencia de la implementación física de un sistema informático.

En los últimos años, tanto el interés como el número de ataques de canal lateral ha aumentado significativamente. Como prueba de ello, la Figura 1.1 ilustra el número creciente de artículos publicados acerca de este tipo de ataques buscando publicaciones en Google Scholar con la consulta *"side channel attack"*.

Sin embargo, entre los diversos tipos de canales laterales, el denominado como térmico ha sido poco explorado. Definido inicialmente para plataformas multinúcleo en [1], las publicaciones posteriores al respecto se centran en mejorar [2] o analizar desde un punto de vista teórico la capacidad del canal [3]. Pese a esto, no se han encontrado trabajos que en un entorno experimental caractericen con detalle los factores principales que afectan al éxito del ataque, sin lo cual resulta altemente complejo desarrollar propuestas de mitigación efectivas y focalizadas.

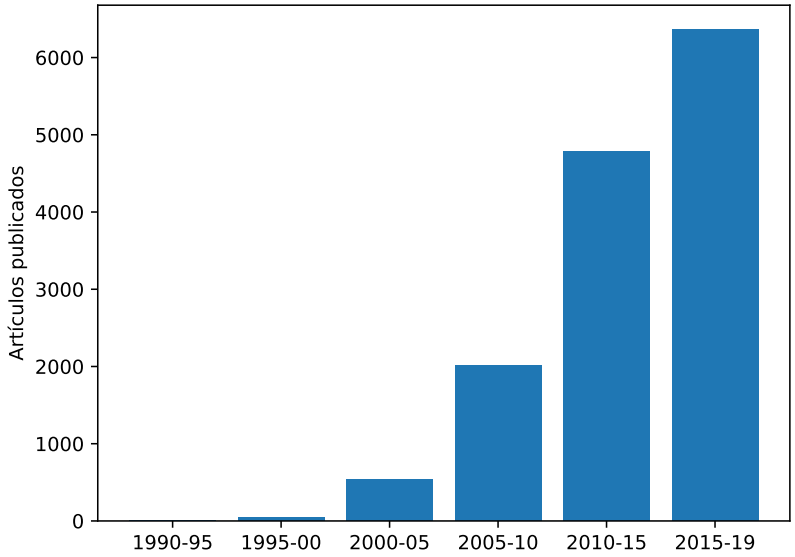



Figura 1.1: Artículos científicos encontrados en Google Scholar con la consulta *"side channel attack"* a fecha de 20 de agosto de 2019.

## 1.2 OBJETIVOS Y ALCANCE

El objetivo principal de este trabajo es caracterizar detalladamente ataques de canal lateral basados en análisis térmico y energético en servidores cloud para posteriormente desarrollar medidas de mitigación lo más efectivas posible. Para lograrlo, primero se debe investigar el estado del arte, replicar los ataques encontrados y portarlos a servidores con multiprocesadores si no son la plataforma nativa de los experimentos. Después, se deben caracterizar los factores principales que afectan a su probabilidad de éxito para, finalmente, diseñar medidas que actúen directamente sobre estos factores y mitiguen el alcance del ataque. En resumen, las tareas a realizar son las siguientes:

1. Buscar bibliografía acerca de ataques de canal lateral basados en monitorización energética y térmica, así como propuestas de mitigación.
2. Seleccionar ataques prometedores y realizables en plataformas multinúcleo para replicar los experimentos.
3. Preparar y caracterizar la plataforma en la que se van a realizar los experimentos.
4. Implementar y caracterizar los ataques seleccionados.
5. Con los resultados obtenidos, diseñar medidas de mitigación.
6. Implementar y evaluar las medidas de mitigación desarrolladas.

Se ha logrado la consecución de todas las tareas anteriores, caracterizando experimental y exhaustivamente el ataque lateral térmico para, a continuación, desarrollar y evaluar medidas de mitigación sencillas y efectivas.

## 1.3 ESTRUCTURA DEL DOCUMENTO

El resto de la memoria se organiza como sigue: el Capítulo 2 introduce al lector algunos conceptos fundamentales para comprender el trabajo; el Capítulo 3 explora el trabajo previo sobre ataques de canal lateral de tipo térmico y la facilidad de realización de los mismos; el Capítulo 4 describe la instalación, el equipamiento y el software con el que se llevan a cabo los experimentos; el Capítulo 5 caracteriza el ataque a partir de los experimentos realizados; partiendo de la caracterización anterior, el Capítulo 6 plantea un escenario para mostrar lo dañino que puede ser este ataque; en el Capítulo 7 se desarrollan y evalúan diferentes propuestas de mitigación. Finalmente, el Capítulo 8 resume las conclusiones principales y el trabajo futuro. En el Anexo A se puede consultar un informe de dedicación que desglosa las horas de trabajo empleadas en cada tarea y un diagrama de Gantt para ver cómo se distribuyen a lo largo de la duración del proyecto.

# 2 FUNDAMENTOS

*Este capítulo expone varios conceptos y definiciones clave para la comprensión del trabajo como los ataques de canal lateral, la disipación térmica en procesadores y las diferencias entre técnicas de virtualización.*

## 2.1 CANALES LATERALES Y ATAQUES

### 2.1.1 *Canales Laterales*

En el ámbito de la computación, un canal lateral es un canal de comunicación que se origina inesperadamente como producto del establecimento de un canal de comunicación deseado, debido a la implementación física del sistema que lo alberga. Algunos ejemplos de canal lateral son los siguientes:

- Mostrar una imagen en un monitor LCD genera a su alrededor interferencias en forma de variaciones en el campo electromagnético como efecto de las oscilaciones de alta frecuencia en el hardware del controlador de la pantalla. Medir las variaciones en dicho campo con suficiente precisión permite reconstruir el mensaje original [4].
- Los condensadores de los sistemas informáticos producen emanaciones acústicas de alta frecuencia, comúnmente denominadas como *coil whine*, que están relacionadas con la información que procesan internamente. Analizando los patrones de dichas emanaciones se pueden extraer claves criptográficas [5].
- En plataformas multinúcleo, un núcleo es capaz de calentar a los de su alrededor. Modulando la información en forma de calentamiento o reposo, se puede establecer un canal de comunicación térmico. Esto se conoce como transmisión por canal lateral térmico [1].

### 2.1.2 *Ataques de Canal Lateral*

Un ataque de canal lateral utiliza estos canales accidentales para filtrar información sensible del sistema o bien para transmitir información a través de ellos sin permiso. Sin embargo, cabe destacar que los humanos que disponen de acceso legítimo al sistema no se consideran una parte de la implementación física y los ataques que se basan en manipularlos (ingeniería social), coaccionarlos o torturarlos

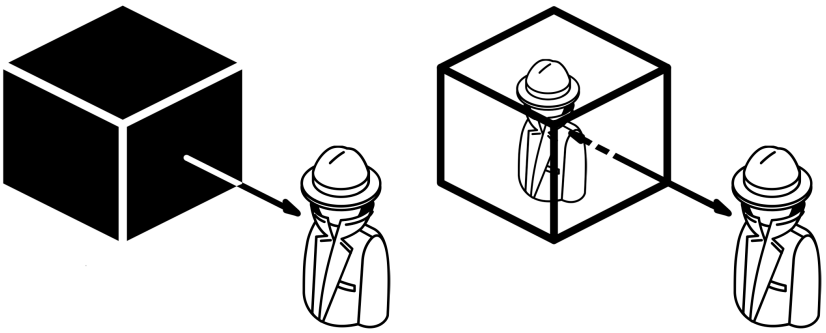
*El "criptoanálisis de manguera de goma" es un eufemismo que designa tortura.*

(criptoanálisis de manguera de goma) no se consideran ataques de canal lateral.

### 2.1.3 *Clasificación de Ataques de Canal Lateral*

El concepto de ataque de canal lateral es muy amplio, lo que hace necesario establecer una clasificación. En [6] se diferencian dos grandes familias de ataques:

- Ataques pasivos o de "caja negra": con estos ataques se puede extraer información sensible del sistema simplemente observándolo desde el exterior. La reconstrucción de imágenes a partir de emanaciones electromagnéticas de LCDs y la extracción de claves criptográficas a partir de emanaciones acústicas, citados en la Sección 2.1.1, se consideran como ataques pasivos.
- Ataques activos o de "caja blanca": en este caso, hace falta algún tipo de intervención activa con el sistema para, o bien extraer la información sensible, o bien hacer uso del canal lateral. Un ejemplo de ataque activo es la transmisión a través de canal lateral térmico (véase Sección 3.3).



(1) Ataque pasivo o de "caja negra". (2) Ataque activo o de "caja blanca".

Figura 2.1: Esquematización de ataques pasivos y activos.

## 2.2 DISIPACIÓN TÉRMICA EN PROCESADORES

Los canales laterales térmicos se fundamentan en la generación de calor por parte de los núcleos de un procesador. En estos sistemas, toda la energía eléctrica consumida se disipa en forma de calor. El consumo en una CPU se calcula como la suma de energía dinámica ($E_{dyn}$) y energía estática ($E_{static}$) de acuerdo con la Ecuación 2.1.

$$E_{cpu} = E_{dyn} + E_{static} \tag{2.1}$$

$E_{dyn}$ se refiere a la energía consumida por el cambio de estado o conmutación de los transistores que implementan el procesador. Por tanto, este consumo depende tanto del número total de transistores como del factor de actividad en el sistema. Otros factores que intervienen en el consumo dinámico son la tensión de alimentación del circuito así como la frecuencia de operación del procesador. Por su parte, $E_{static}$ hace referencia al producto entre la tensión de alimentación y la suma de corrientes de fuga, las cuales aumentan con el número de transistores y la temperatura del procesador. Se trata por tanto de un consumo que está presente aunque los transistores no muestren actividad. Cabe destacar que los ataques de canal lateral térmico suelen emplear mayoritariamente variaciones en $E_{dyn}$.

Para que el procesador siga funcionando correctamente, el calor generado, ya sea por consumo dinámico o estático, debe de ser evacuado con un sistema de refrigeración adecuado. Normalmente, en equipos de sobremesa se suele optar por disipadores de cobre con refrigeración activa en el propio disipador, mientras que en los servidores, se colocan disipadores pasivos sobre la CPU, así como ventiladores en la parte posterior y anterior de la caja, garantizando un flujo de aire correcto a todos los componentes. Para garantizar que el sistema de refrigeración elegido es el adecuado, los fabricantes de CPUs facilitan el Thermal Design Power (TDP), el cual indica qué potencia debe ser capaz de disipar de manera continuada el sistema de refrigeración.

## 2.3 MÁQUINAS VIRTUALES

Se denomina Máquina Virtual (MV) a una réplica software eficiente y aislada del hardware de una máquina real [7]. En este contexto, se introduce el concepto de hipervisor, un software que se ejecuta entre el hardware y la máquina virtual para proveer a las aplicaciones de un entorno idéntico a la máquina original, con pequeñas pérdidas en velocidad y que controla por completo todos los recursos del sistema. Por su parte, un entorno cloud es, básicamente, un centro de datos que ofrece MVs abaratando el coste de la redundancia de servicios y ofreciendo un mayor aprovechamiento del hardware gracias a la elasticidad que permite la virtualización.

Cualquier solución de virtualización debe respetar los siguientes principios [7]:

- Seguridad: el hipervisor debe mantener en todo momento el control completo de los recursos virtualizados.
- Fidelidad: el comportamiento de una aplicación en una MV debe de ser idéntico al de la misma aplicación ejecutándose en una máquina real.
- Eficiencia: la pérdida de rendimiento de la máquina virtual debe de ser aceptable.

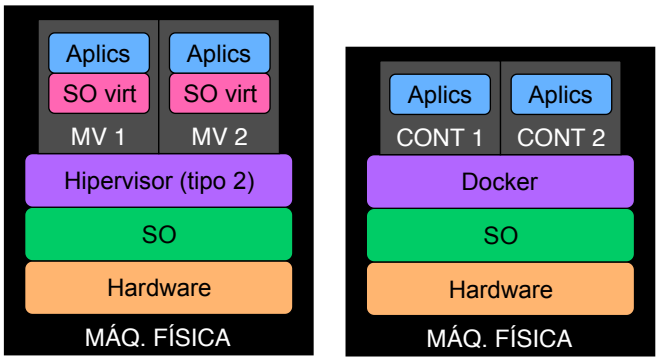
Actualmente, las opciones de virtualización son muy variadas pero en este trabajo se va a hacer hincapié en dos tipos concretos de MV: las de sistema y las de proceso o contenedor.

### 2.3.1 *Máquina Virtual de Sistema*

En este tipo de implementación, el sistema virtualizado consta de un Sistema Operativo (SO) completo ejecutándose sobre un hipervisor de tipo 2, quien le presenta la capa de hardware. Los hipervisores de tipo 2 se diferencian de los tipo 1 en que se ejecutan sobre un SO entre ellos y el hardware, mientras que los otros se ejecutan directamente sobre el hardware. Cada vez que el sistema virtualizado necesita ejecutar código privilegiado, el hipervisor interviene la instrucción, realiza los cambios necesarios y devuelve el control al sistema virtualizado de forma transparente. Este proceso se denomina *trap&emulate*.

### 2.3.2 *Máquina Virtual de Proceso o Contenedor*

Otro tipo de virtualización muy utilizado actualmente son los contenedores. Los contenedores se basan en mecanismos del SO [8] para aislar (*chroot jails* y *namespaces*) y asignar recursos (*cgroups*) y así evitar la pérdida de rendimiento generada por otro SO adicional entre las aplicaciones y el hardware. La Figura 2.2 muestra una comparativa entre ambos tipos de virtualización.




(1) Virtualización de sistema tipo 2.

(2) Virtualización de procesos (Docker).

Figura 2.2: Comparación de los tipos de virtualización presentados.

# 3 ESTADO DEL ARTE

*Este capítulo presenta el estado del arte del trabajo, diferenciando entre ataques basados en análisis energético y térmico sobre un dispositivo electrónico, y aquellos ataques que establecen un canal lateral térmico de comunicación. El capítulo concluye presentando brevemente algunas técnicas para mitigar estos ataques.*

## 3.1 ANÁLISIS ENERGÉTICO

El Simple Power Analysis (SPA) consiste en analizar las fluctuaciones en consumo energético de un dispositivo electrónico para extraer información sensible de este, ya sean las instrucciones que se están ejecutando o los datos de entrada. Este ataque es comúnmente realizado sobre dispositivos empotrados relativamente sencillos. Las primeras referencias a este ataque en el ámbito académico datan de 1999, momento en el cual se publica un artículo donde se obtienen claves de cifrado y descifrado del algoritmo Data Encryption Standard (DES) empleando SPA [9].

*Entre los documentos filtrados por Edward Snowden se menciona la presentación en una conferencia de la CIA de un ataque DPA para extraer claves criptográficas de chips TPM de cinco fabricantes.*

En el mismo artículo también se introduce el concepto de Differential Power Analysis (DPA), el cual consiste en analizar una gran cantidad de trazas energéticas del chip en vez de una sola para, a continuación, realizar operaciones estadísticas y obtener información sensible a partir de las trazas.

En otro artículo posterior de los mismos autores, se puede observar como, tras 12 años, muchos dispositivos y diferentes algoritmos criptográficos siguen siendo vulnerables a DPA [10]. Cabe destacar que este ataque se utiliza a menudo en seguridad ofensiva de dispositivos empotrados y que existen soluciones comerciales que lo automatizan como el ChipWhisperer® de NewAE Technology mostrado en la Figura 3.1 actualmente a la venta por 250 dólares en su versión Lite.

## 3.2 ANÁLISIS TÉRMICO

Aunque un ataque mediante análisis energético puede resultar muy efectivo, no siempre se tiene la posibilidad de monitorizar el consumo energético de un dispositivo. Sin embargo, la Primera Ley de Joule postula que los conductores eléctricos disipan un calor proporcional a la resistencia del conductor y la corriente que lo atraviesa. Debido a esta ley, los dispositivos electrónicos siempre disipan calor. Concretamente en las CPUs, compuestas de miles de millones de transistores cambiando su estado continuamente, el calor instantáneo disipado

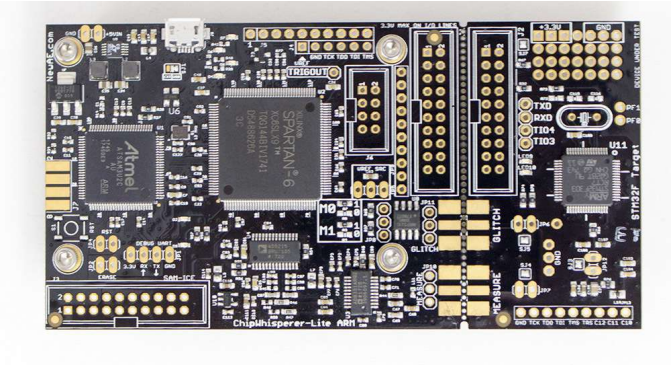



Figura 3.1: ChipWhisperer®-Lite (CW-1173).

está muy relacionado con la carga de trabajo del sistema. De esta manera, aunque no se pueda obtener una traza del consumo energético, se puede inferir a partir de una traza térmica. Solamente con análisis térmico se puede atacar una implementación de Rivest, Shamir y Adleman (RSA) en controladores AVR [11]. El inconveniente principal de estos ataques es que a menudo requieren acceso físico al dispositivo y hardware adicional, así como la necesidad de desencapsular[1] el microcontrolador.

## 3.3 TRANSMISIÓN A TRAVÉS DE CANALES LATERALES TÉRMICOS

A diferencia de los ataques anteriores, que se clasifican como pasivos debido a que el atacante se limita exclusivamente a la escucha o monitorización de un sistema, la transmisión a través de canales laterales es de tipo activo, puesto que requiere de un emisor y un receptor. Esta sección se refiere en concreto al canal lateral de tipo térmico.

La idea detrás de este ataque se presenta en [1], donde se plantea la posibilidad de utilizar la pequeña distancia física entre los núcleos de una CPU y su capacidad para calentarse o enfriarse rápidamente para establecer un canal lateral. Para ello, basta con elegir un núcleo emisor así como otro receptor y contar con un proceso con permisos de usuario en cada uno de los núcleos. Entre ambos núcleos se debe acordar un lapso de tiempo $T_b$, definido como el tiempo que dura la transmisión de un bit. Para transmitir un '1' lógico, el proceso emisor ejecuta una carga pesada en el núcleo emisor, mientras que para transmitir un '0' lógico lo deja reposar. El proceso receptor monitoriza la sonda térmica del núcleo en el que se ejecuta y, debido a la pequeña distancia física entre los núcleos, se pueden detectar los cambios de

1 Se denomina desencapsulado a la retirada de la capa de protección plástica o cerámica de circuitos integrados para revelar la oblea litografiada del interior. Normalmente el proceso se realiza atacando químicamente el encapsulado, con una cortadora láser o una fresadora **CNC!**.

temperatura del emisor. La Figura 3.2 muestra un esquema con la creación de un canal térmico entre dos núcleos emisores y receptores.

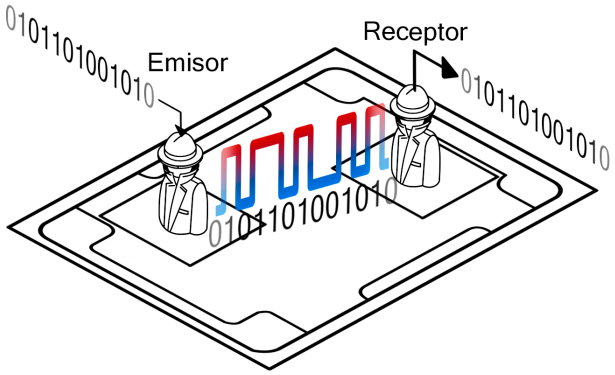



Figura 3.2: Esquematización del canal lateral térmico.

Inicialmente, el receptor se encuentra a la espera de un preámbulo por parte del emisor. Una vez detectado el preámbulo, comienza la transmisión y detección de la secuencia de bits del mensaje mediante flancos ascendentes y descendentes del perfil térmico. De esta manera, el receptor es capaz de reconstruir el mensaje original. Cabe destacar que para detectar secuencias de bits con un valor lógico constante, como por ejemplo 1-1-1-1-1, el receptor inicia un *timeout* desde el momento en que se detecta el último flanco. Asumiendo por ejemplo un *timeout* de 20 s (segundos) y un $T_b$ de 1 s, si durante los siguientes 20 segundos no se detectase ningún flanco, esta situación se interpretaría como la recepción de 20 bits seguidos con valor lógico '1'. Una vez vencido el *timeout*, el receptor da como finalizado el envío del mensaje.

Con un $T_b$ menor de un segundo se obtiene una tasa de transmisión de 1,33 bps (bits por segundo) sin corrección de errores o 0,33 bps aplicando Hamming(7,4) como código de corrección de errores.

Uno de los principales problemas de este método es que la tasa de transmisión es baja, inconveniente que se intenta paliar en [2], donde se emplea un mecanismo de codificación basado en ciclos de trabajo a una frecuencia fija. Además, los autores establecen diferentes canales a diferentes frecuencias para así incrementar la tasa de transmisión hasta en un 370 %. A diferencia del trabajo anterior, donde el canal lateral se establece en hardware real, en este artículo se utiliza un simulador desarrollado en C++, con DSENT [12] como modelo energético y Hotspot [13] como modelo térmico. Finalmente, también se menciona que la transmisión básica descrita en [1] es muy sensible al ruido de fondo.

## 3.4 OTROS ATAQUES RELACIONADOS

Existen otros ataques basados en modular información como incrementos o decrementos de carga de trabajo en el procesador. En este

sentido, un trabajo interesante es [14], en el cual el emisor modula la información de esta manera, pero el receptor es un proceso que únicamente mide el tiempo en el que tarda en incrementar un contador hasta llegar a un valor determinado. Aprovechando que el governor *ondemand* varía la frecuencia de la CPU en función de la carga de trabajo, el receptor debería tardar más o menos en incrementar el contador dependiendo de si el emisor emite un '1' o un '0', lo que permite tasas de transmisión de hasta 20 bps.

Por otra parte, el ataque descrito en [15] se aprovecha de que normalmente los sistemas solo activan los ventiladores cuando se encuentran bajo carga. De esta manera, el receptor puede detectar los cambios acústicos generados por la activación o parada del ventilador. La tasa de transmisión obtenida es de hasta 0,25 bps.

## 3.5 MITIGACIÓN

La mitigación de este tipo de ataques no resulta sencilla puesto que se deben a la propia implementación física del sistema. En la mayoría de ocasiones, si los ataques no se tienen en cuenta en la fase de diseño, resulta imposible mitigarlos al 100 % una vez el producto ya se encuentra en el mercado. Para comprender la importancia de este tipo de vulnerabilidades y sobre todo en entornos de Inteligencia, hay que destacar la existencia de estándares como TEMPEST [16], los cuales se encargan de garantizar que los sistemas informáticos no generen ningún tipo de emanaciones a partir de las que se pueda reconstruir información sensible. En España, esta certificación la otorga el Centro Criptológico Nacional (CCN) y se debe aplicar a toda información clasificada Nacional Confidencial o superior, afectando a equipos, sistemas e instalaciones.

Generalmente, las propuestas de mitigación siguen dos enfoques diferentes. El primer enfoque permite actuar sobre el elemento que filtra la información sensible del sistema. Algunos ejemplos:

- Añadir ruido en el sistema que evite que se pueda distinguir la información filtrada [17].
- Diseñar software y hardware que no genere variaciones térmicas ni energéticas suficientemente significativas [9, 18].
- Añadir aislamiento que evite las emanaciones térmicas [16].
- Usar una metodología de diseño tolerante a ciertas fugas de información sensible [9].

Por otra parte, si el receptor utilizado para decodificar la información del canal lateral se encuentra dentro del sistema, se puede actuar sobre él de la siguiente manera:

- Limitando el acceso a los sensores utilizados para decodificar la información [1].
- Reduciendo la resolución de dichos sensores.

# 4 ENTORNO EXPERIMENTAL

*Este capítulo se organiza en dos partes diferenciadas: por un lado, se detallan las características hardware del servidor sobre el cual se pretenden caracterizar ataques, así como la instalación de control de temperatura para que las condiciones ambientales externas al servidor sean las adecuadas; por otro lado, se presenta el software básico del servidor y el software específico diseñado para la creación de ataques de canal lateral térmico.*

## 4.1 HARDWARE E INSTALACIÓN DE CONTROL DE TEMPERATURA

Los experimentos han sido realizados en un servidor típico de un centro de datos como es el SuperMicro con chasis SuperChassis 815TQC-R706WB2, con factor de forma de 1U[1] y los siguientes componentes principales, anotados en la Figura 4.1:

1. Placa base SuperMicro X11DDW-L con dos sockets Intel LGA-3647.
2. 2 CPUs Intel Xeon Gold 6130 a 2,10 GHz, con 16 núcleos cada una.
3. 92 GB de memoria RAM DDR4 a 2666 MHz.

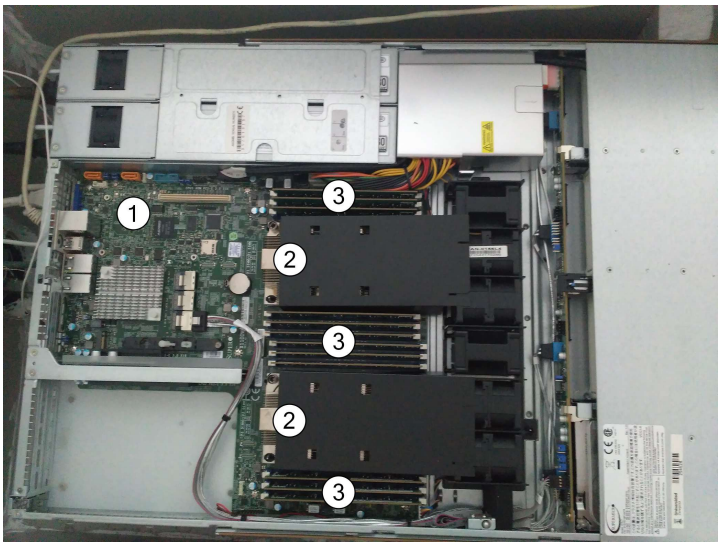



Figura 4.1: Servidor utilizado en los experimentos.

1 Los diseñadores de armarios o *racks* de servidores utilizan *unidades de rack* para establacer la altura de los servidores. 1U se refiere a 44.45 mm o 1.75 pulgadas. Un rack consta normalmente de 42 Us.

El hecho de que el servidor cuente con dos CPUs diferentes permite que la mayoría de tareas del sistema se ejecuten en una CPU y los experimentos en la otra, de manera que las mediciones de temperatura de los núcleos no se vean afectadas.

Dado que el servidor se encuentra en el laboratorio del Grupo de Ingeniería Térmica y Sistemas Energéticos (GITSE), para poder encender y apagar el servidor remotamente sin molestar, se ha implementado un sistema que se apoya en el protocolo IPMI que provee el BMC del servidor. Sin embargo, exponer el BMC a Internet no es una práctica segura, por lo que se utiliza una Raspberry Pi bastionada con una interfaz de red USB adicional para actuar de intermediario. De esta manera, solo se expone a Internet el puerto 22/tcp de la Raspberry Pi, que no debería suponer un riesgo significativo si ha sido bastionada correctamente. La configuración de red se puede observar en la Figura 4.2.

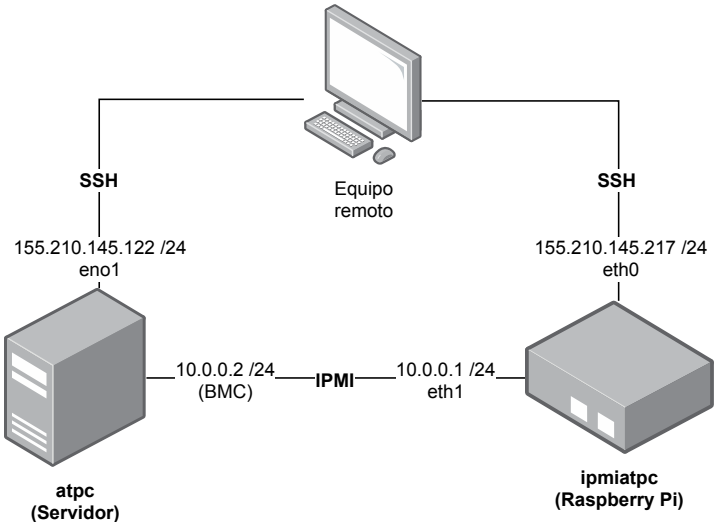



Figura 4.2: Configuración de red para encendido y apagado remoto seguro.

Para caracterizar los ataques de transmisión a través de canales térmicos minimizando las condiciones ambientales externas al servidor, resulta conveniente mantener una temperatura y caudal de aire constantes en el frontal del servidor. Para ello, se ha tenido en cuenta la instalación de control de temperatura del laboratorio del GITSE de la Universidad de Zaragoza, la cual se ha utilizado en trabajos previos [19, 20]. Esta instalación permite dotar de una mayor profundidad a la caracterización, ya que en el estado del arte [1-3] no se provee un entorno experimental con temperatura y caudal controlados.

La Figura 4.3 muestra un esquema de esta instalación de control de temperatura, donde se puede identificar, a lo largo de un circuito cerrado de aire, una máquina frigorífica, una resistencia, un ventilador y el servidor contenido en un compartimento. Por otra parte, la Figura 4.4 muestra fotografías de cada uno de los componentes de la

instalación de control de la temperatura. A continuación se detallan las características más relevantes de cada uno de estos componentes:

1. Máquina frigorífica que enfría agua hasta 11 ºC y extrae el calor del aire del circuito cerrado que atraviesa un intercambiador de calor.

2. Resistencia conectada a controlador Proporcional, Integral y Derivativo (PID) con sonda térmica que permite amortiguar los cambios de temperatura cuando se enciende y se apaga el compresor de la máquina frigorífica.

3. Ventilador regulable con caudalímetro, que permite establecer un caudal de aire constante en el frontal del servidor.

4. Cerramiento de poliestireno aislante para el servidor con dos cámaras aisladas que simulan el pasillo frío y el pasillo caliente de un centro de datos.

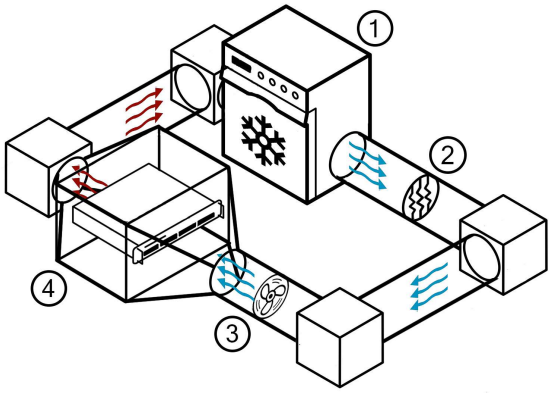



Figura 4.3: Esquema de la instalación para el control de temperatura.

## 4.2 SOFTWARE

Tanto la configuración como el software básico del servidor son similares al de otros servidores del Departamento de Informática e Ingeniería de Sistemas. En concreto, el servidor cuenta con un sistema operativo CentOS 7.5.1804 con kernel Linux 3.10.0-957.27.2.el7.x86_64 y una configuración y despliegue inicial llevada a cabo por el administrador, José Antonio Gutierrez.

### 4.2.1 *Monitorización Energética*

En cuanto a la monitorización energética, el primer problema que se encuentra es de dónde obtener datos en directo del consumo instantáneo de la CPU. Se puede obtener con el driver Intel Running Average


(1) Máquina frigorífica.

(2) Resistencia con control PID.


(3) Ventilador con caudalímetro.

(4) Servidor en el cerramiento.

Figura 4.4: Componentes de la instalación para el control de la temperatura.

Power Limit (RAPL), que hay que instalar ya que no es un software que se encuentre comúnmente instalado en los SOs UNIX.

Añadido al problema de la poca disponibilidad de drivers que obtengan el consumo instantáneo, RAPL solo provee datos de consumo de la CPU completa, en lugar de núcleo por núcleo como sería deseable.

Teniendo en cuenta todos los inconvenientes anteriores, se ha monitorizado una operación criptográfica que se ejecuta en un núcleo para obtener su perfil energético y de ahí se ha intentado extraer información sensible. Sin embargo, como el consumo energético obtenido hace referencia a la CPU completa, el experimento no ha resultado provechoso, puesto que se añade demasiado ruido como para poder observar detalles relevantes.

### 4.2.2 *Monitorización Térmica*

Por otra parte, se puede subsanar la baja disponibilidad de herramientas de monitorización energética utilizando las sondas térmicas de la CPU, dado que en los SOs UNIX se encuentran comúnmente disponibles mapeadas en ficheros a los cuales se puede acceder con permisos de usuario.

Aprovechando esta funcionalidad del SO, se obtiene el perfil térmico de la misma operación criptográfica que en el experimento anterior. Sin embargo, al igual que en la monitorización energética, los resultados no se muestran relevantes, puesto que las variaciones en temperatura

resultan menos precisas que las de consumo energético. Además, las sondas de Intel solo proveen una resolución de $\pm 1$ ºC.

### 4.2.3 *Transmisión Térmica*

El software específico diseñado y utilizado para transmitir a través del canal lateral térmico es un sistema relativamente sencillo. La Figura 4.5 muestra un esquema de las tres partes principales de este sistema:

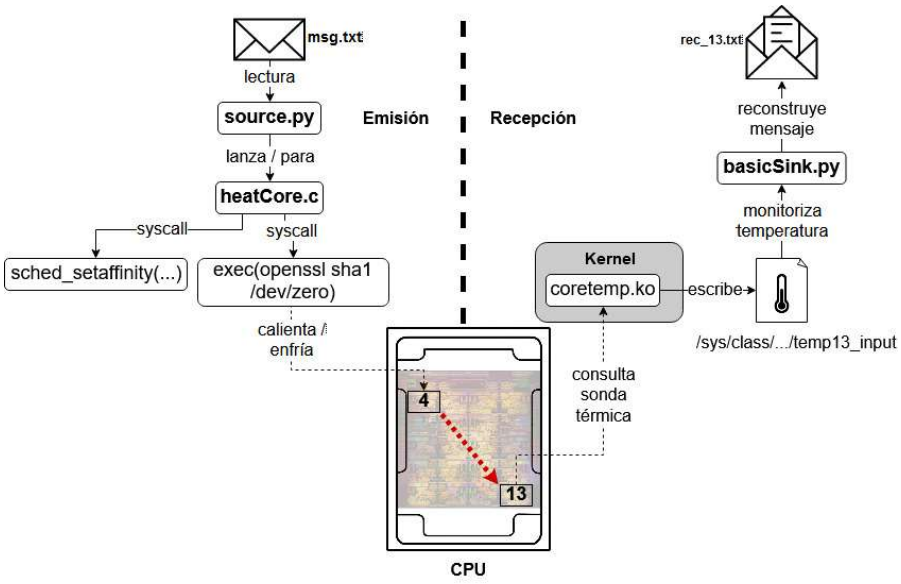



Figura 4.5: Esquema del software para transmitir un mensaje a través del canal térmico. En este ejemplo se establece un canal entre los núcleos 4 y 13 de una CPU.

- "Calentador": se trata de un script llamado heatCore.c y desarrollado en C que se encarga de calentar un núcleo concreto. Cuando se ejecuta en forma de proceso, en primer lugar utiliza la llamada al sistema *sched_setaffinity* para solicitar al planificador o *scheduler* que el proceso sea ejecutado en el núcleo desde el cual se pretende emitir (núcleo 4 en la figura). Esta solicitud no se garantiza al 100 %; sin embargo, el *scheduler* suele respetar la solicitud. En caso contrario, se mantiene al proceso en espera en lugar de planificarlo en un núcleo libre no especificado. Una vez el proceso se encuentra en el núcleo deseado, se ejecuta el script *openssl sha1 /dev/zero*. Este script calcula el hash Secure Hashing Algorithm 1 (SHA1) de una cadena interminable de ceros, lo cual establece una carga de trabajo del 100 % en el núcleo y lo calienta de forma efectiva. La versión inicial del programa utilizaba un bucle interminable. Sin embargo, siguiendo la metodología de [1], se observa que las operaciones criptográficas calientan el

núcleo más rápidamente. La versión de OpenSSL utilizada es la 1.0.2k-FIPS compilada con el repertorio de instrucciones SIMD SSE2.

- Emisor: se refiere al script source.py desarrollado en Python. Este script modula la temperatura del núcleo emisor lanzando y matando al proceso calentador en función de los bits del mensaje de entrada *msg.txt*.

- Receptor: se trata del script basicSink.py desarrollado en Python y ejecutado en el núcleo receptor (núcleo 13 en la figura). Este script reconstruye el mensaje (*rec_13.txt*) a partir de los cambios de temperatura monitorizando los ficheros */sys/class/hwmon/hwmon2/temp*_input*. En estos ficheros, el kernel de Linux escribe la temperatura reportada por las sondas térmicas de cada núcleo cada 2 ms y con una resolución de $\pm 1$ ºC.

# 5

# EXPERIMENTOS REALIZADOS Y CARACTERIZACIÓN PROPUESTA

*Este capítulo recoge los resultados experimentales más relevantes. Puesto que los ataques pasivos no se han mostrado efectivos en el servidor objeto de estudio, la experimentación se basa en cuantificar el impacto de los factores principales en la transmisión de información mediante un ataque activo basado en un canal lateral térmico. De acuerdo con la bibliografía existente, estos factores son la distancia física entre núcleos emisores y receptores, el periodo de tiempo requerido para el envío de un bit de información, la elección del núcleo emisor, el ruido de fondo en términos de calor generado por aquellos núcleos que no participan en la transmisión y la temperatura del aire de entrada al servidor.*

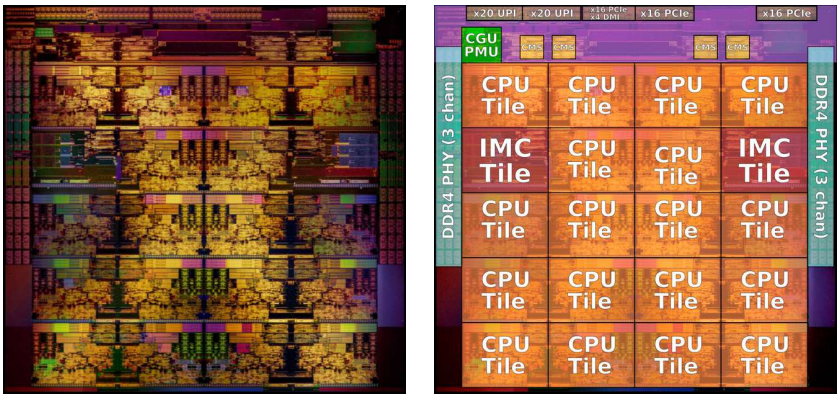
*Una de las métricas principales que cuantifican el impacto de los factores sobre el canal lateral es la tasa de error en el envío de un mensaje. Esta tasa se mide como el número de bits del mensaje recibido que difieren con el original, comparando la cadena de principio a fin. Si el mensaje recibido es de menor longitud, todos los bits de la diferencia cuentan como erróneos. De esta manera, cuando se indica que un receptor tiene una tasa de error del 100 %, significa que no ha sido capaz de detectar ningún flanco en el perfil térmico, es decir, no puede reconstruir ningún bit del mensaje.*

## 5.1 DISTANCIA ENTRE NÚCLEOS

Un factor importante a la hora de asegurar el éxito en la transmisión de información a través de un canal térmico es la elección de los núcleos emisores y receptores. La mayoría de la bibliografía [1-3] menciona ligeramente que esta elección afecta de forma severa a la tasa de errores en la transmisión. Esto tiene bastante sentido, puesto que físicamente hay núcleos más alejados y más cercanos dentro del empaquetado.

La Figura 5.1 muestra la distribución física de los núcleos de la CPU Intel Xeon Gold 6130, la cual se corresponde con la arquitectura *Skylake Server* HCC. Se aprecian un total de 18 núcleos, por lo que, a falta de una mayor descripción por parte del fabricante, se cree que, para la versión de 16 núcleos utilizada, se dejan dos *tiles* libres o simplemente desactivados.

Cuando un proceso se ejecuta en una CPU, el *scheduler* le asigna un núcleo con un identificador lógico. Sin embargo, no existe ninguna documentación acerca de la asignación o mapeo de identificadores lógicos a cada núcleo físico en una CPU. No obstante, se ha diseñado




(1) Litografía (2) Litografía anotada

Figura 5.1: Litografía de la arquitectura de Intel Skylake Server HCC [21].

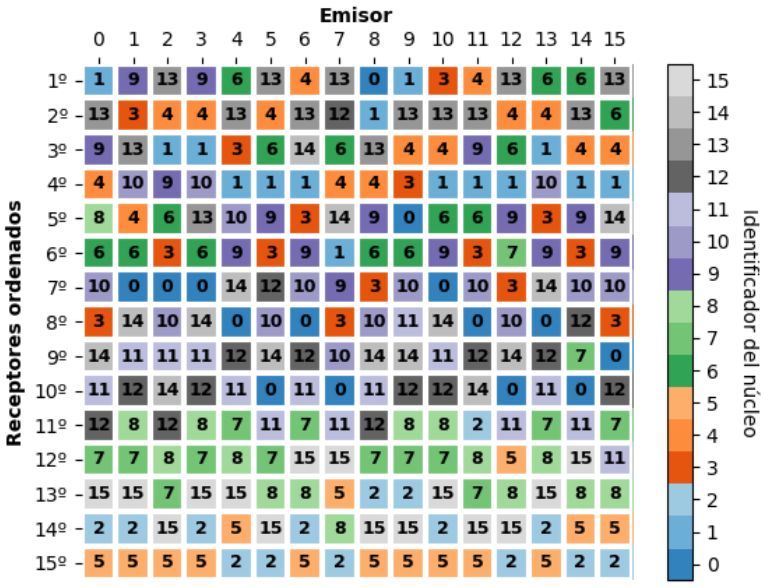
un primer experimento para tratar de inferir la mencionada asignación. Para ello se procede como sigue:

- En primer lugar, se calienta un núcleo con una carga de trabajo grande durante un intervalo suficiente de tiempo (200 segundos) como para calentar al resto de núcleos en reposo.
- Se mide el calor recibido por cada núcleo a lo largo de los 200 segundos y se ordenan los núcleos de mayor a menor calor recibido.
- Para obtener resultados más robustos, se realiza la mediana de 10 ejecuciones, siempre esperando a que la temperatura media de la CPU descienda hasta 40 ºC antes de volver a calentar un núcleo.

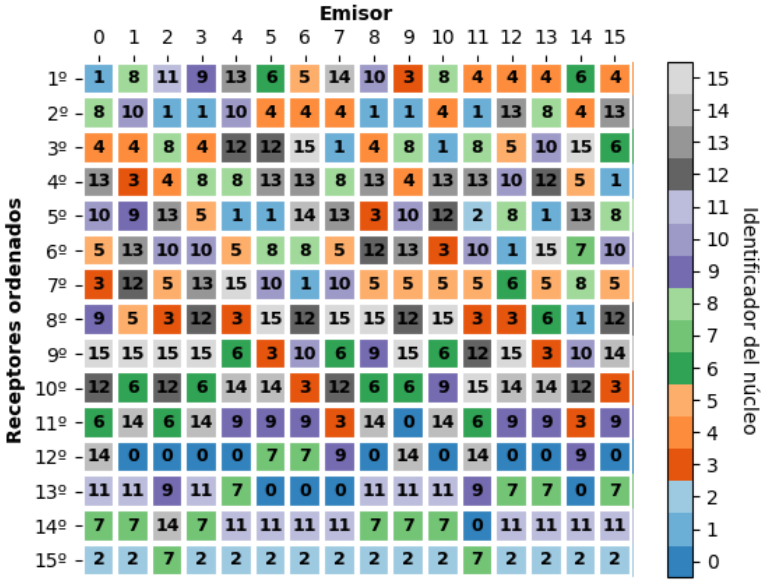
Intuitivamente, los núcleos más cercanos al emisor deberían recibir más calor. La Figura 5.2 muestra, por cada CPU del servidor, una matriz de calor donde cada núcleo emisor que se calienta (columnas) tiene asociada una lista de núcleos receptores ordenada de mayor a menor calor recibido (filas). Por ejemplo, cuando el emisor es el núcleo 7, el segundo receptor que más se calienta es el núcleo 12 para la CPU 1.

A partir de las listas ordenadas de núcleos y la fotografía de la distribución de los núcleos litografiados se deberían poder establecer restricciones de proximidad que permitiesen inferir la distribución física en su totalidad. Sin embargo, al realizar este experimento se obtienen resultados que no permiten inferir la estructura física con exactitud, aunque se realizan observaciones interesantes.

En primer lugar, se puede observar que existen núcleos que son consistentemente mejores receptores que otros y que algunos de éstos se repiten en ambas CPUs. Es el caso de los núcleos 1 y 4. De estas observaciones se puede deducir que probablemente, debido al diseño




(1) Resultados en la CPU 1




(2) Resultados en la CPU 2

Figura 5.2: Matrices con los núcleos receptores ordenados de mayor a menor calor recibido por cada núcleo emisor calentado.

de la arquitectura del procesador, estos núcleos se encuentren más cercanos a elementos que se calientan y añaden ruido, como puede ser el caso de los controladores de memoria, tiles Integrated Memory Controller (IMC), en la Figura 5.12. Otras posibles razones pueden ser

simplemente las variaciones en el proceso de fabricación del procesador (*process variation*) o la distribución de la pasta térmica, las cuales pueden afectar a la disipación de calor en el chip. Por el contrario, también se pueden observar núcleos que permanecen menos calientes que el resto, independientemente del núcleo emisor. Este es el caso de los núcleos 2, 5 y 15 en la CPU 1 y los núcleos 2, 7 y 11 en la CPU 2. Aparte de las razones ya expuestas de *process variation* o distribución de pasta térmica, también puede tratarse de núcleos aislados, como aquellos ubicados en las esquinas de la CPU o como vecinos de los núcleos libres o desactivados.

Por otro lado, también cabe destacar el hecho de en ambas CPUs se pueden observar parejas de núcleos que establecen relaciones bidireccionales de mejor emisor↔mejor receptor. Es el caso de las parejas 1↔9 y 4↔6 en la CPU 1 y las parejas 4↔13, 3↔9, 5↔6 y 8↔10 en la CPU 2. Estas parejas pueden deberse a que los núcleos implicados sean vecinos en la distribución de núcleos en el chip.

## 5.2 TRANSMISIÓN BÁSICA

Una vez caracterizada la distancia entre núcleos, se procede a establecer un canal básico de transmisión. De acuerdo con el experimento anterior, se elige una pareja emisor-receptor que favorezca a la transmisión, es decir, mejor emisor↔mejor receptor. Por ejemplo, este es el caso de los núcleos 4 y 13 como emisor y receptor, respectivamente, en la CPU 2. Se enviarán dos mensajes con longitudes diferenciadas: un primer mensaje consta de una longitud arbitraria de 14 bits obtenidos pseudo-aleatoriamente de */dev/urandom*; el segundo mensaje constituye una transmisión más realista con hasta 100 bits pseudo-aleatorios a enviar. El preámbulo se establece como el envío de un '1' durante 15 segundos seguido de un '0' con duración $T_b$=2000 ms. La principal finalidad de este experimento es comprobar el software desarrollado (véase Sección 4.2) y verificar si se puede replicar el experimento realizado en [1]. La Tabla 5.1 muestra los parámetros principales de la transmisión básica. Salvo para el último experimento, la temperatura no está controlada pero se ha comprobado que se mantiene alrededor de 35 ºC. Esto se hace para evitar la necesidad de desplazarse físicamente al laboratorio y arrancar la instalación de control de temperatura salvo en los experimentos en los que sea completamente necesario.

Con fines ilustrativos, la Figura 5.3 muestra la monitorización de la temperatura para la transmisión de 14 bits entre los núcleos implicados. El núcleo receptor es capaz de calentarse y enfriarse lo suficiente por efecto del emisor como para recibir correctamente el mensaje con una tasa de errores nula. En el caso del mensaje largo de 100 bits se obtienen idénticos resultados, confirmando la robustez del canal entre dos núcleos favorables a la transmisión. En la imagen se puede

| PARÁMETRO | VALOR |
|---|---|
| Emisor | 4 |
| Receptor | 13 |
| Longitud de los mensajes | 14 y 100 bits |
| $T_b$ | 2000 ms |
| Mediana de n repeticiones | 1 |
| Temperatura de pasillo frío | ~35 ºC |
| Caudal en el pasillo frío | 20 $m^3$/h |

Tabla 5.1: Parámetros de la transmisión básica.

observar tanto el preámbulo de 16 segundos, los bits del mensaje y el *trailing bit*, que es el último bit del mensaje pero invertido para poder generar un flanco más y así indicar que se ha terminado de transmitir.

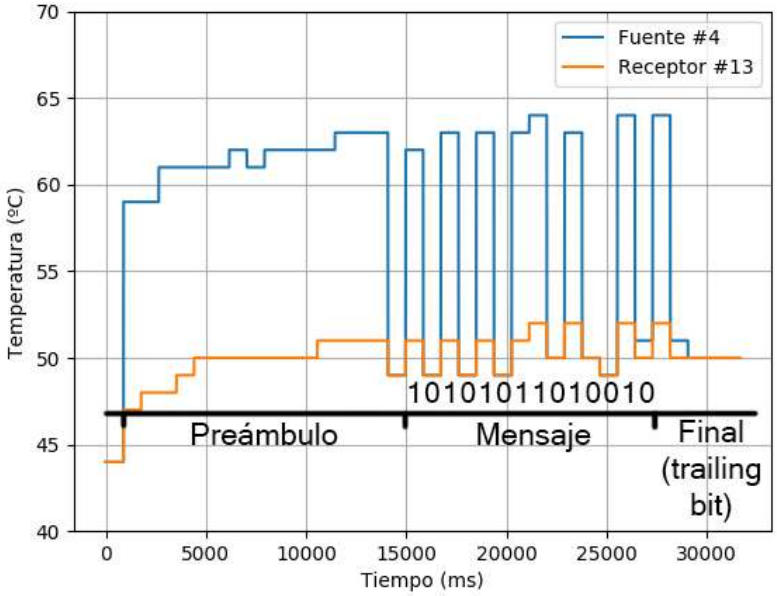



Figura 5.3: Perfil térmico de la transmisión básica de 14 bits.

## 5.3 ELECCIÓN DE $T_b$

Una vez establecido con éxito un canal lateral térmico en el servidor, resulta de interés obtener el valor óptimo de $T_b$[1] para esta plataforma, ya que la tasa de transmisión (en bps) depende de este parámetro.

Para establecer el mejor $T_b$ para esta plataforma, se obtiene la mediana de la tasa de errores de 10 transmisiones del núcleo 4 a los mejores núcleos receptores de calor, es decir, los núcleos 13 y 10 de

1 Recuérdese que $T_b$ es el tiempo de calentamiento para enviar un '1' o de reposo para enviar un '0' como parte de la secuencia de bits del mensaje.

acuerdo con el experimento de la Sección 5.1. El mensaje transmitido tiene una longitud de 100 bits, mientras que el caudal en el pasillo frío se mantiene en 20 $m^3$/h .

Como se puede apreciar en los resultados de la Figura 5.4, valores de $T_b$ demasiado pequeños (<1000 ms) obtienen una tasa de error bastante elevada debido a que los núcleos no disponen de tiempo suficiente como para calentarse. Nótese que el valor mínimo de $T_b$ que obtiene una tasa de error nula es aquel con una duración de 1000 ms para ambos núcleos. Este valor difiere de los 750 ms reportados en [1], por lo que se concluye que se trata de un valor dependiente de la plataforma, pero en cualquier caso, el resultado se encuentra dentro de un rango relativamente pequeño. En el resto de experimentos se utilizará un valor de $T_b$ de 1000 ms (es decir, una tasa de transmisión de 1 bps). Respecto a la forma del gráfico, el resultado intuitivo sería que la gráfica describiese una parábola. Sin embargo, se pueden observar repuntes en 1250 ms para ambos núcleos y en 2000 ms para el núcleo 13. La causa de estos picos es desconocida, aunque se pueden deber o bien a la implementación del software o al funcionamiento propio del procesador.

Finalmente, nótese también que la tasa de error es ligeramente superior en el núcleo 10 respecto al núcleo 13 en la mayoría de experimentos. Esto se debe a que los núcleos 10 y 13 son el segundo y primer mejor receptor, respectivamente.

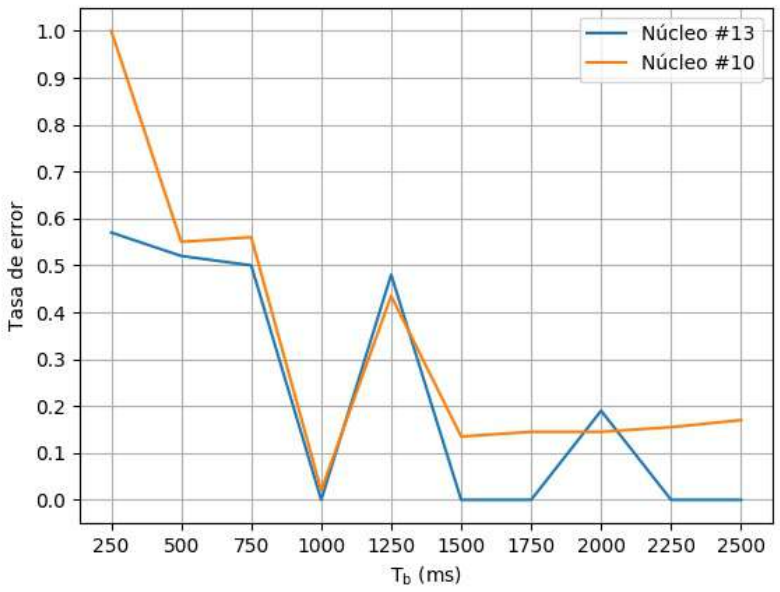



Figura 5.4: Tasa de error para cada $T_b$ en incrementos de 250 ms.

## 5.4 CORRELACIÓN ENTRE TASA DE ERROR Y DISTANCIA ENTRE NÚCLEOS

En el experimento anterior se intuye como, para un $T_b$ determinado, la distancia entre núcleos juega un papel importante en la tasa de error. Esta sección estudia la correlación entre la tasa de error y la distancia entre núcleos. Para ello, se toma el núcleo 4 como emisor y se realiza la transmisión del mensaje al resto de núcleos. La Tabla 5.2 muestra los parámetros principales del presente experimento.

| PARÁMETRO | VALOR |
|---|---|
| Emisor | 4 |
| Receptores | Resto |
| Longitud del mensaje | 100 bits |
| $T_b$ | 1000 ms |
| Mediana de n repeticiones | 10 |
| Temperatura de pasillo frío | $\sim$35 $^{o}$C |
| Caudal en el pasillo frío | 20 $m^3$/h |

Tabla 5.2: Parámetros para la comparación de núcleos receptores.

La Figura 5.5 muestra los resultados. En el eje X se puede observar una lista ordenada de núcleos de izquierda a derecha de acuerdo con la creciente de la tasa de error mostrada en el eje Y. Los números situados por encima de la línea se refieren al identificador de núcleo en cada punto.

Para los 7 mejores receptores, la tasa de error se mantiene por debajo de un 10 %. A partir de este punto, la tasa de error aumenta significativamente hasta situarse por encima de un 40 % para el receptor número 12. Más allá de este receptor, la pendiente aumenta dramáticamente y asciende hasta una tasa de error de 100 % en el peor receptor.

Como cabría esperar, existe similitud entre estos resultados y el orden de receptores establecido para el núcleo 4 en la matriz de calor (véase la Figura 5.22). Por ejemplo, 6 de los 7 mejores receptores en este experimento (núcleos 13, 10, 12, 8, 1 y 15) también se encuentran entre los 7 primeros en la matriz de calor. Por otro lado, los dos peores receptores (núcleos 11 y 2) coinciden en ambos experimentos.

## 5.5 PEOR RECEPTOR COMO EMISOR

El experimento anterior ha mostrado como la elección del núcleo receptor resulta determinante para el éxito del ataque. Por el contrario, esta sección estudia el impacto de la elección del núcleo emisor. Para ello, se tomará como emisor al peor receptor, es decir, al núcleo 2. Nótese que este núcleo no es solamente el peor receptor para el emisor

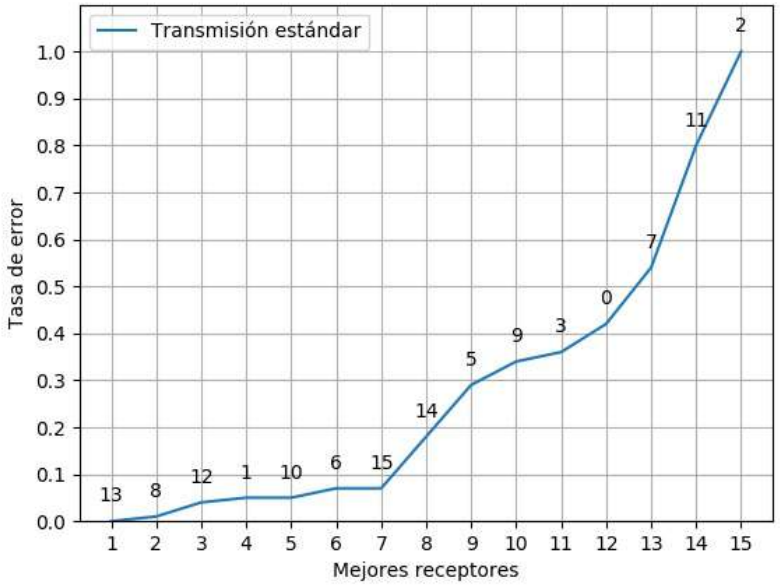



Figura 5.5: Tasa de error de mejor a peor núcleo receptor para el núcleo emisor 4. Los números sobre la línea identifican a cada núcleo receptor.

4, sino que también lo es para 14 de los 15 posibles emisores de acuerdo con la matriz de calor de la Figura 5.22.

La Figura 5.6 muestra los resultados, manteniendo la línea de la transmisión base desde el emisor 4 a efectos de comparación. Para los tres primeros núcleos, la tasa de error es idéntica en ambos experimentos. Sin embargo, desde el cuarto receptor hasta el séptimo, la tasa de error con el emisor 2 es más del doble respecto a la tasa con el emisor 4, llegando hasta casi un 20 %. A partir de este punto y hasta el receptor 12, la tasa de error con el emisor 2 siempre queda por encima de la tasa con el emisor 4. Sin embargo, para el receptor 13 la tasa de error coincide, y sorprendentemente, para los dos peores receptores, la tasa de error con el emisor 2 se sitúa por debajo de la del emisor 4, y siempre por debajo de un 70 %.

Finalmente, al contrario que para el emisor 4, existen bastantes diferencias en el orden de receptores entre la matriz de calor y el experimento actual. En este caso, para los 7 primeros receptores, tan solo coinciden 3 de ellos (núcleos 11, 1 y 8) en la matriz de calor.

## 5.6 RUIDO DE FONDO

Los experimentos anteriores obtienen una buena tasa de error cuando se consideran los mejores receptores en la transmisión. Se trata de un sistema que no ejecuta ninguna carga aparte de los procesos necesarios para crear el canal térmico entre el núcleo emisor y recep-

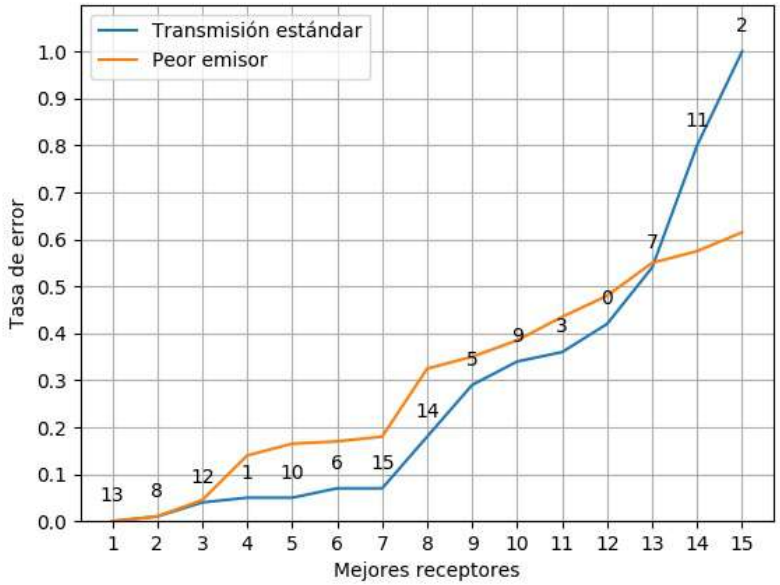



Figura 5.6: Tasa de error de mejor a peor núcleo receptor para los núcleos emisores 4 (transmisión base) y 2 (peor emisor).

tor. Sin embargo, en un escenario real es muy probable que algunos núcleos ejecuten otros procesos totalmente ajenos al canal, y el calor emitido por los mismos pueda afectar a la calidad de la transmisión del canal. Esta sección evalúa el ruido generado por la ejecución de otros procesos en la CPU.

En concreto, el siguiente experimento evalúa el peor caso posible en el que todos los núcleos se encuentran al 100 % de carga de trabajo en la CPU. Para ello, se utiliza el programa *stress-ng*, generando una carga de fondo en todos los núcleos mientras se realiza la transmisión. El comando utilizado ha sido *stress-ng –cpu 32 –timeout 115s*.

La Figura 5.7 muestra la tasa de errores añadiendo ruido de fondo. El núcleo emisor se corresponde con el número 4. A efectos de comparación, se mantienen en el gráfico los resultados de los experimentos anteriores. El impacto del ruido de fondo resulta determinante, puesto que el mejor receptor pasa de una tasa de error nula hasta una tasa de casi un 80 %. Para sucesivos receptores, se produce un incremento constante hasta alcanzar el 100 % en el peor receptor. De estos resultados se puede concluir que la carga de trabajo del sistema es determinante para el éxito del ataque.

También cabe destacar que la adición de ruido cambia la lista ordenada de mejores receptores respecto a la transmisión base; sin embargo, para los 7 mejores receptores, sin importar el orden específico, en ambos casos existen coincidencias hasta con 5 receptores (núcleos 12, 11, 1, 13 y 15).

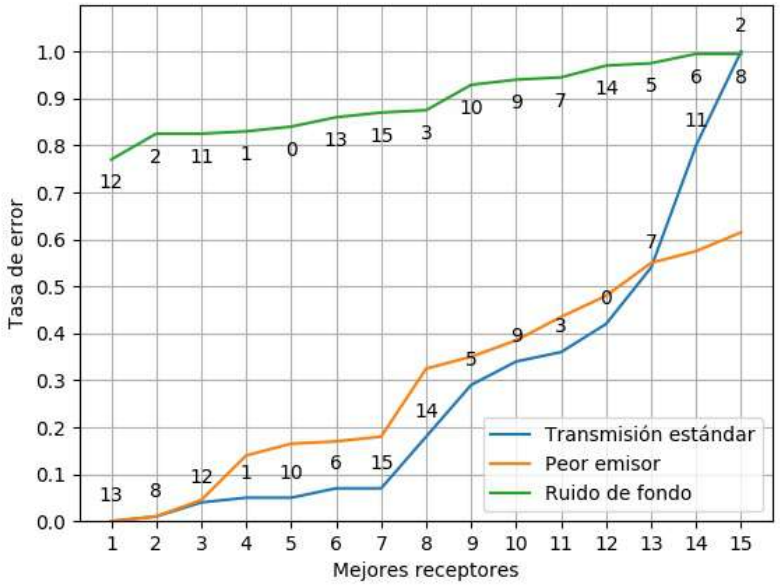



Figura 5.7: Tasa de error de mejor a peor núcleo receptor para los núcleos emisores 4 con ruido (ruido de fondo) y sin ruido (transmisión base), y 2 sin ruido (peor emisor).

## 5.7 INFLUENCIA DE LA TEMPERATURA

Para analizar cómo influye la temperatura del aire de entrada al servidor, se realiza un experimento considerando los valores de temperatura de 22, 27 y 35 °C, simulando la temperatura del pasillo frío de un centro de datos mediante la instalación de control de temperatura (véase la Sección 4.1). Nótese que los dos primeros valores de temperatura se encuentran dentro del rango de condición del aire definido por el estándar de ASHRAE (desde 18 hasta 27 °C) [22], mientras que el tercero es la temperatura máxima de operación del servidor utilizado. La Tabla 5.3 muestra los parámetros principales del experimento.

| PARÁMETRO | VALOR |
|---|---|
| Emisor | 4 |
| Receptores | {13,8,12,1,10,6,15,14,5} (9 mejores) |
| Longitud del mensaje | 100 bits |
| $T_b$ | 1000 ms |
| Mediana de n repeticiones | 5 |
| Temperatura de pasillo frío | {22,27,35} °C |
| Caudal en el pasillo frío | 20 $m^3$/h |

Tabla 5.3: Parámetros para el estudio de cambios en la temperatura.

La Figura 5.8 muestra la tasa de error para los diferentes puntos de temperatura y los 9 mejores núcleos receptores. Se puede apreciar que la relevancia de la temperatura es mínima para el rango seleccionado. Esto se debe a que, pese a que es cierto que la temperatura en reposo que detectan las sondas dentro de la CPU está relacionada con la temperatura del aire en el pasillo frío, los cambios de la temperatura en reposo no afectan a la detección del mensaje por parte del núcleo receptor. Esto se debe a que el receptor detecta flancos ascendentes o descendentes para interpretar bits '1' o '0' y el emisor genera incrementos de temperatura suficientemente amplios para ello.

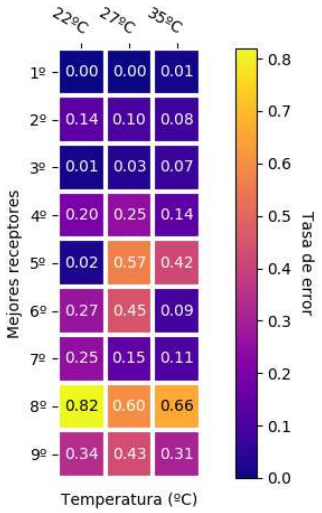



Figura 5.8: Tasa de error para los 9 mejores receptores con temperatura controlada.

## 5.8 CARACTERIZACIÓN PROPUESTA

De los experimentos anteriores se puede concluir que la transmisión de información a través de canales térmicos se considera viable en el servidor objeto de estudio. Además, la caracterización de los factores principales que afectan al canal lateral ha mostrado que:

- Se han identificado núcleos como mejores y peores receptores de calor, independientemente del núcleo emisor, de acuerdo con la obtención de matrices de calor. A partir de estas matrices no es posible inferir distribución física de los núcleos en el chip.
- La elección de $T_b$ es fundamental para el éxito del ataque. Valores de $T_b$ por debajo de 1000 ms aumentan significativamente la tasa de error. Por otro lado, la tasa de transmisión obtenida de 1 bps (sin aplicar corrección de errores) es relativamente baja.
- La elección del núcleo emisor no resulta determinante, sobretodo considerando tan solo a los primeros mejores núcleos receptores.

Por el contrario, una vez establecido el núcleo emisor, la elección del núcleo receptor sí que resulta determinante para el ataque.

- Para asegurar el éxito del ataque, el sistema no debe tener una carga alta de trabajo en aquellos núcleos que no participan en el canal.
- La temperatura del aire de entrada al servidor ha resultado irrelevante para el éxito del ataque, por lo que se descarta como medida de mitigación el hecho de elevar la temperatura del pasillo frío del centro de datos.

# 6

# PRUEBA DE CONCEPTO: EXTRACCIÓN DE INFORMACIÓN EN CLOUDS COMPARTIDOS

*En este capítulo se describe un escenario en el que puede ser útil la transmisión a través de canales térmicos aunque no aparente un riesgo real por sí sola. Como se menciona en el Capítulo 7, es importante destacar que este ataque no es aplicable a entornos cloud basados en virtualización de sistemas sin contenedores.*

Se presupone un escenario en el que la empresa víctima del ataque dispone de contenedores Docker en un cloud público de tamaño reducido. Un atacante explota una vulnerabilidad en uno de esos contenedores y consigue abrir una sesión en él con permisos de usuario. El objetivo del atacante es obtener y extraer una información secreta que se encuentra en ese contenedor. Una vez obtenida la información, quiere transmitirla fuera de la empresa de una manera poco obvia. Para ello, se decide emplear un canal térmico y así evitar canales habituales probablemente monitorizados por el Security Operations Center (SOC) de la empresa víctima.

Con sus permisos de usuario, el atacante carga el programa emisor que se encarga de transmitir la información. Además, se encarga de anotar las características del hardware, así como de obtener un identificador único de la máquina (por ejemplo de */etc/machine-id*), como se puede apreciar en la Figura 6.1.

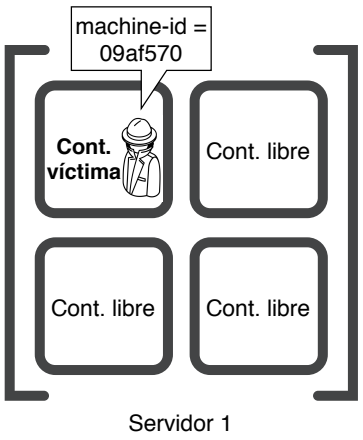



Figura 6.1: Contenedor víctima infectado e identificador obtenido.

Por otra parte, el atacante también se encarga de solicitar contenedores con el mismo tipo de hardware para aumentar las posibilidades de que el contenedor sea desplegado en la misma máquina física que el anterior. De esta manera, el atacante puede solicitar una gran cantidad

de contenedores o ir probando poco a poco en función de sus recursos o de lo sigiloso que desee ser.

En el momento en que un contenedor receptor desplegado por el atacante reporta el mismo identificador que el contenedor emisor, significa que ambos contenedores se encuentran en la misma máquina, y puede por tanto comenzar la transmisión a través del canal. La Figura 6.2 representa un diagrama de esta situación en la máquina referida como *Servidor 1*. Nótese que desplegar un contenedor en otra máquina, como es el caso del *Servidor 2*, aporta un identificador distinto. Asimismo, en la Figura 6.3 se muestra una traza de ejecución de esta transmisión mediante contenedores desplegados en el servidor objeto de estudio.

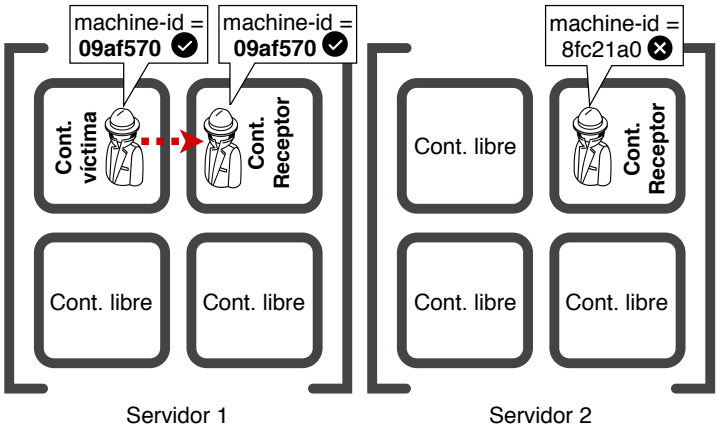



Figura 6.2: Coincidencia de identificador, comienzo de transmisión.

```
1: @victim:~
[www-data@victim ~]$ ls
heatCore  heatCore.c  msg.txt  source.py
[www-data@victim ~]$ cat /etc/machine-id
a393dc68bda043098190519783b83f5
[www-data@victim ~]$ python3.6 ./source.py msg.txt
Read message (ASCII): SUPERSECRETO

Read message (bin): 10101001101010101010000010001010101001001010011010001010100001101010010010001010101
1000100111100001010
Sending preamble for  15000 ms...
10Sending the message: 10101001101010101010000010001010101001001010011010001010100001101010010010001010101
0101000100111100001010
1Message sent
[www-data@victim ~]$
```

(1) Terminal del emisor tras la transmisión.

```
2: @attacker:~
[badguy@attacker ~]$ ls
basicSink.py
[badguy@attacker ~]$ python3.6 ./basicSink.py a393dc68bda043098190519783b83f5
Matching machine ID, starting...
Initial core temp: 63.0 C
Waiting for preamble...
Preamble received correctly
Receiving message: 1101010011010101010100000100010101010010010010011010001010100001101010010010001010101
1000100111100010101
Core 13 - Message received and timed out: SUPERSECRETO

[badguy@attacker ~]$
```

(2) Terminal del receptor tras la transmisión.

Figura 6.3: Capturas de pantalla tras la ejecución del ataque.

Para ofuscar la intención del programa emisor que es el que está sujeto a la vigilancia del SOC de la empresa víctima, se podría intentar camuflar como *malware* minador de criptomonedas, ya que para calentar el procesador ejecuta operaciones criptográficas. Además, al calentar solo un núcleo, si el SOC solo monitoriza el uso de la CPU completa en lugar de núcleos individuales, no va a reportar un porcentaje alto de uso como consecuencia del calentamiento exclusivo del núcleo emisor. Una vez realizada la transferencia de información a través del canal, se puede eliminar el ejecutable y redesplegar el contenedor receptor en otra máquina física para posteriormente enviar la información al exterior por cualquier canal habitual, ahora ya sin la monitorización del SOC de la empresa víctima.

# 7 PROPUESTAS DE MITIGACIÓN

*En el capítulo anterior se aprecia cómo un canal térmico puede ser explotado para extraer información sensible de un contenedor en un cloud público. Para evitarlo, en el presente capítulo se analizan y se desarrollan diferentes propuestas de mitigación tanto hardware como software.*

## 7.1 MITIGACIONES HARDWARE

Una opción que mitiga por completo el riesgo de transmisión entre máquinas virtuales es el uso de virtualización de sistema en lugar de virtualización de procesos. Los hipervisores exponen a los sistemas virtualizados Virtualized CPUs (vCPUs). Esto significa que asignan a cada sistema una porción de los recursos de la CPU física. Sin embargo, aunque se especifique que una máquina virtual solo puede usar un núcleo concreto, si el hipervisor es de tipo 2 (véase la Sección 2.3), éste tendrá que realizar la llamada al sistema *sched_setaffinity*, la cual no garantiza al 100 % que una vCPU se limite a un único núcleo físico. Este hecho se traduce en que las vCPUs no son asignaciones precisas uno a uno con núcleos concretos de la CPU real.

Con todo, exponer los sensores de la CPU a los sistemas virtualizados no resulta adecuado, ya que no siempre van a mostrar valores útiles ni tampoco precisos. La decisión de diseño del uso de hipervisores tipo 2, como los de VirtualBox o KVM, tiene el efecto colateral de ser una medida de seguridad completamente efectiva contra la transmisión térmica entre sistemas virtualizados, ya que el receptor no puede acceder a las sondas térmicas de la CPU. Sin embargo, sí que se ha conseguido la transmisión desde un sistema virtualizado a la máquina que lo alberga. No obstante, este caso es muy poco probable en entornos cloud, donde la seguridad de las máquinas físicas suele ser mucho mayor que la de las máquinas virtualizadas.

## 7.2 MITIGACIONES SOFTWARE

De acuerdo con lo expuesto en la sección anterior, la virtualización de sistemas completos es una medida efectiva contra los ataques laterales térmicos. Sin embargo, es poco realista que por este motivo un cloud de contenedores replantee su modelo de negocio, conviertiéndose en un cloud de virtualización de sistemas. Debido a esta razón, se propone una solución software más viable y barata.

La solución consiste en editar *coretemp.c*, el módulo de kernel que provee la temperatura de las sondas térmicas al resto del sistema.

Las propuestas desarrolladas y probadas han sido, por un lado, la adición de ruido en las lecturas de los sensores, y por otra, el bloqueo de aquellos procesos que solicitan insistentemente la lectura de los sensores.

### 7.2.1 *Adición de Ruido*

Esta propuesta consiste en añadir ruido a las lecturas de los sensores indiscriminadamente. La implementación es relativamente sencilla y se puede observar el pseudocódigo en el Algoritmo 1.

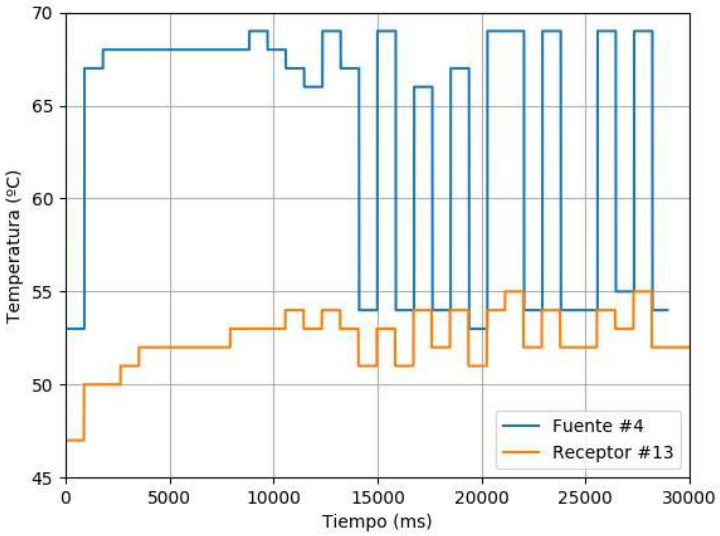
**Algoritmo 1:** coretemp_noisy.c

```
Resultado: Añade ruido a la temperatura leída con una
           probabilidad del 50 %.
temp ← LeerSensor();
ruido ← Prob5050();
si ruido entonces
    positivo ← Prob5050();
    si positivo entonces
        devolver temp + 3;
    fin
    en otro caso
        devolver temp − 3;
    fin
fin
en otro caso
    devolver temp;
fin
```

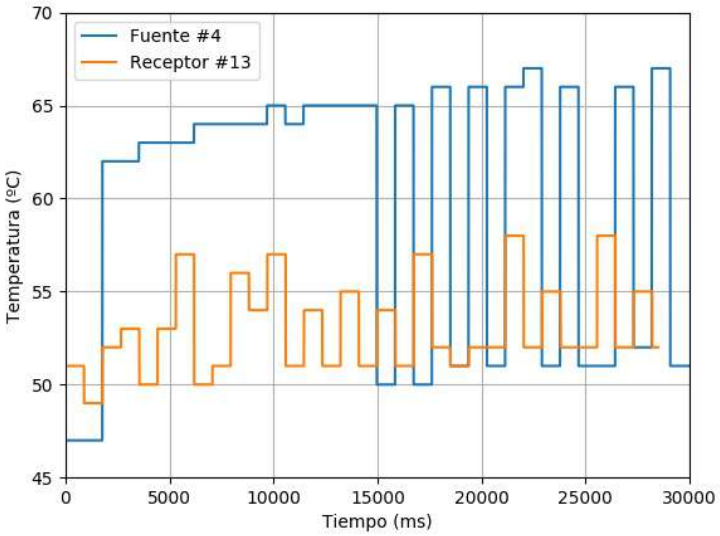
El inconveniente de esta propuesta es que es demasiado invasiva puesto que puede añadir ruido a procesos legítimos. Además, produce flancos artificiales para el receptor, aumentando o disminuyendo la temperatura leída en 3 °C, pero solo en un 50 % de las lecturas, lo que termina homogeneizando la tasa de error al 50 %. La Figura 7.2.1 muestra los efectos de esta propuesta sobre la recepción del mensaje por parte del receptor. Nótese que el preámbulo tiene lugar en los primeros 15 segundos. A efectos de comparación, también se añade la traza de la recepción original sin medidas de mitigación. En resumen, la adición de ruido no se puede considerar como una propuesta satisfactoria.

### 7.2.2 *Bloqueo de Procesos*

El bloqueo de procesos supone una solución más complicada pero por contra mucho más focalizada y menos invasiva. Esta propuesta se encarga de monitorizar qué procesos están solicitando la temperatura muchas veces en un periodo corto de tiempo. Si el proceso excede un




(1) Transmisión mediante el módulo *coretemp.ko* original del kernel.




(2) Transmisión con adición de ruido mediante el módulo *coretemp_noisy.ko*

Figura 7.1: Trazas de transmisión del módulo original y de la propuesta de adición de ruido.

número de peticiones en el periodo establecido, se añade a una lista de bloqueo. Esto significa que si un proceso de la lista de bloqueados solicita la temperatura, recibe siempre un valor constante, de forma que el receptor no puede inferir ningún flanco y, por tanto, no puede reconstruir el mensaje, resultando en una tasa de error del 100 % y mitigando por completo el ataque.

La Figura 7.2 muestra los efectos de esta propuesta en la traza de recepción del mismo mensaje considerado en la sección anterior.

Debido a que el periodo de bloqueo es menor que el preámbulo, se consigue bloquear al receptor antes siquiera de que el emisor comience la transmisión efectiva del mensaje, mitigando por completo el ataque. El Algoritmo 2 muestra un pseudocódigo de esta solución.

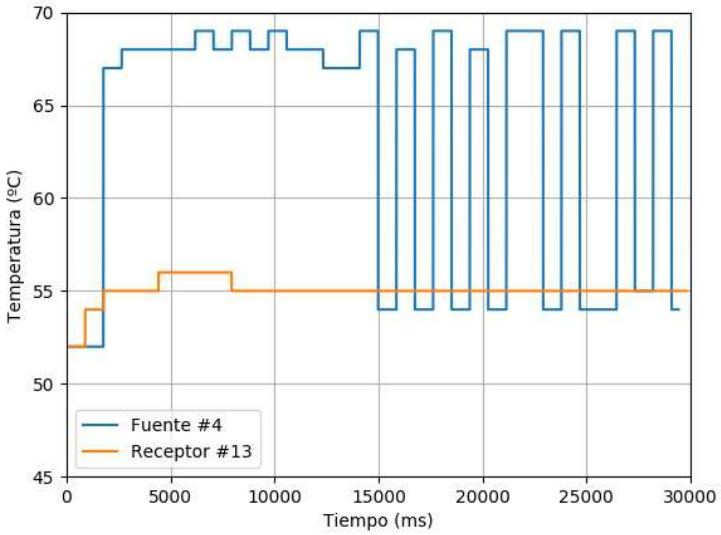



Figura 7.2: Traza de transmisión con la propuesta de bloqueo de procesos mediante el módulo *coretemp_pidban.ko*.

**Algoritmo 2:** coretemp_pidban.c

**Resultado:** Bloquea procesos que consultan la temperatura demasiado a menudo.

```
temp ← LeerSensor();
si estaBloqueado(pidActual) entonces
    temp ← tempCongelada;
fin
si no, si pidActual = ultimoPid entonces
    contadorBloq ← contadorBloq + 1;
    si contadorBloq ≥ LIMITECONT entonces
        bloquearProceso(pidActual);
    fin
    tempCongelada ← temp;
fin
en otro caso
    contadorBloq ← 0;
fin
ultimoPid ← pidActual;
devolver temp;
```

# 8 CONCLUSIONES Y TRABAJO FUTURO

*El presente capítulo expone las conclusiones extraídas de la realización del trabajo, así como posibles líneas de investigación futuras.*

## 8.1 CONCLUSIONES PRINCIPALES

En este trabajo se ha seguido un procedimiento prototípico para la gestión de una vulnerabilidad, en concreto la transmisión de información sensible a través de núcleos de un procesador mediante un ataque de canal lateral térmico [1]. Para ello, se ha analizado el alcance de este ataque buscando trabajos relacionados, exponiendo y explotando una prueba de concepto que muestra el peligro del ataque en una máquina real, caracterizando detalladamente los factores que ejercen un mayor impacto sobre el éxito del ataque y, finalmente, mitigándolo de la forma menos invasiva posible para el funcionamiento de los sistemas afectados. En otras palabras, no solo se han conseguido reproducir los resultados experimentales expuestos en [1] para el servidor cloud objeto de estudio, sino que también se han extendido los experimentos mediante la caracterización de los factores principales que afectan al ataque, proporcionando además una prueba de concepto funcional para posteriormente implementar medidas que mitigan el ataque por completo.

A partir de la caracterización propuesta, se puede concluir que la transmisión a través de un canal térmico es una técnica realizable y efectiva de extracción de información en clouds públicos basados en contenedores Docker. Los factores principales que afectan a la transmisión mediante un canal térmico son: la elección del núcleo receptor para un emisor determinado, la carga de trabajo en aquellos núcleos del procesador que no participan en el canal y la elección del periodo de tiempo ($T_b$) de envío de un bit de información. Para el servidor objeto de estudio, $T_b$ se ha fijado en 1 s, lo cual se traduce en una tasa de transmisión de 1 bps en el canal.

Por otro lado, tanto el uso de virtualización de sistemas como el módulo de kernel desarrollado para bloquear procesos mitigan por completo la transmisión entre contenedores. Aún con la ausencia de la mitigación propuesta, resulta de utilidad que los SOC de clouds públicos monitoricen activamente los accesos a las sondas térmicas. De esta manera, es relativamente sencillo detectar e interrumpir una transmisión a través de un canal térmico.

## 8.2 TRABAJO FUTURO

Respecto al trabajo futuro, sería interesante ahondar en la aplicación de ataques pasivos como DPA o monitorización térmica en entornos cloud. También sería de interés la reproducibilidad y caracterización de las versiones mejoradas [2] de transmisión a través de canales térmicos, las cuales no se han considerado por la ausencia de resultados experimentales detallados en trabajos previos. Finalmente, también se podría extender el módulo de kernel propuesto para la mitigación implementando funciones que incrementarían su usabilidad como las listas blancas de procesos.

# A

# ANEXOS

*Este Anexo incluye las horas de dedicación al trabajo por parte del proyectando, así como un diagrama de Gantt que permite al lector constatar la planificación de cada una de las tareas.*

## A.1 DEDICACIÓN

La duración completa del trabajo ha sido de 269 horas. El desglose en horas de trabajo, de acuerdo con las tareas definidas en la Sección 1.2, es el siguiente:

- Estudio de la bibliografía y selección de ataques: 13 horas
- Preparación de la plataforma experimental: 14 horas
- Implementación y caracterización de los ataques, junto con la ejecución de experimentos: 179 horas
- Diseño, desarrollo y evaluación de módulos de kernel para mitigar el ataque: 18 horas
- Redacción y revisión de la memoria y documentación: 45 horas

La implementación de los ataques y la experimentación son con diferencia la tarea que más tiempo suma al cómputo total de dedicación, constituyendo más de la mitad del tiempo dedicado al proyecto. En concreto, de las 179 horas dedicadas a esta tarea, 57 horas corresponden a la implementación de los ataques, mientras que las 122 horas restantes hacen referencia a la ejecución de experimentos.

## A.2 DIAGRAMA DE GANTT

La Figura A.1 muestra un diagrama de Gantt con las tareas asociadas al proyecto. Estas tareas a su vez se desglosan en subtareas para apreciar con mayor claridad la planificación del trabajo.

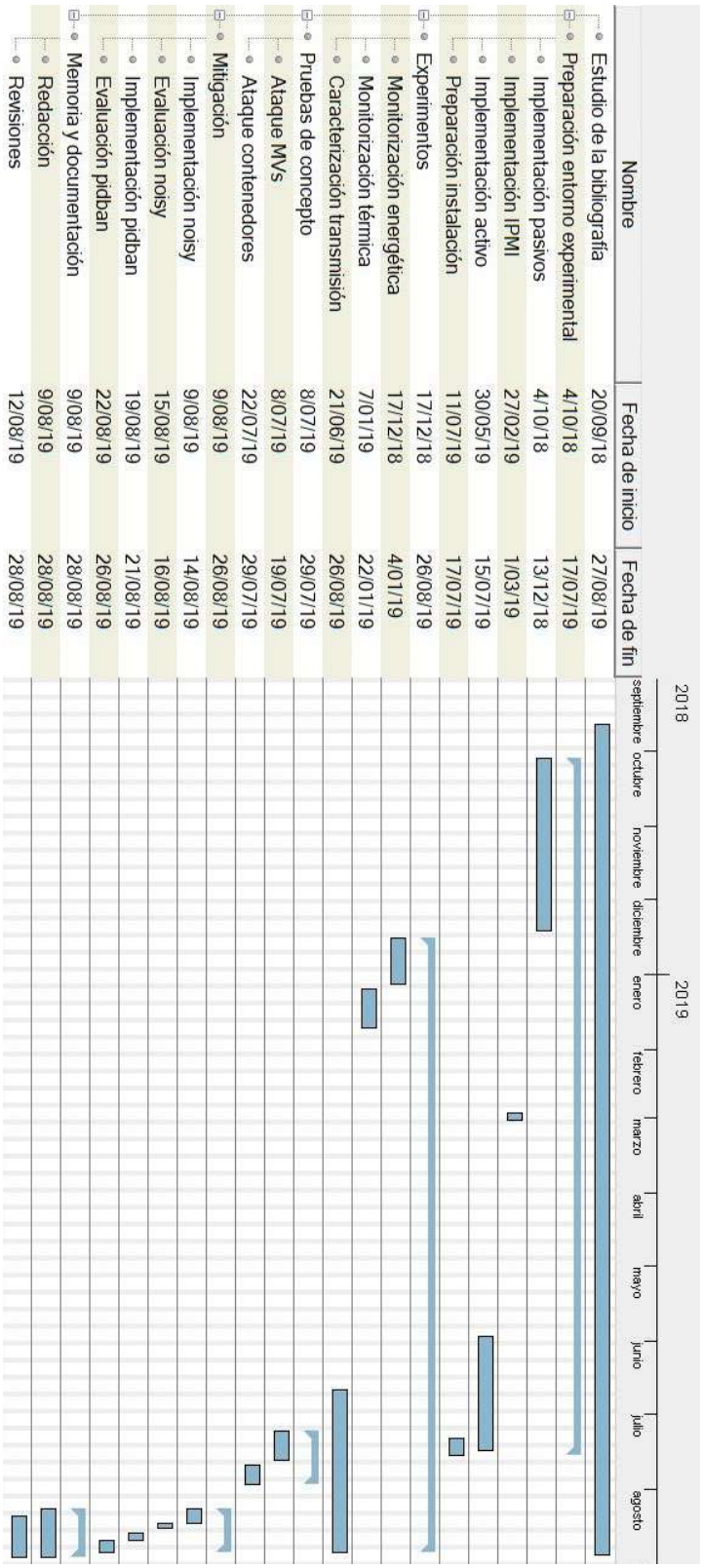



Figura A.1: Diagrama de Gantt.

# BIBLIOGRAFÍA


[1] Ramya Jayaram Masti, Devendra Rai, Aanjhan Ranganathan, Christian Müller, Lothar Thiele, Srdjan Čapkun y E T H Zürich. «Thermal Covert Channels on Multi-core Platforms This paper is included in the Proceedings of the». En: *24th USENIX Security Symposium (USENIX Security 15)* (2015).

[2] Zijun Long, Xiaohang Wang, Yingtao Jiang, Guofeng Cui, Li Zhang y Terrence Mak. «Improving the efficiency of thermal covert channels in multi-/many-core systems». En: *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*. Vol. 2018-Janua. Institute of Electrical y Electronics Engineers Inc., 2018, págs. 1459-1464. ISBN: 9783981926316. DOI: 10.23919/DATE.2018.8342241.

[3] Davide B. Bartolini, Philipp Miedl y Lothar Thiele. «On the capacity of thermal covert channels in multicores». En: *Proceedings of the Eleventh European Conference on Computer Systems - EuroSys '16* (2016), págs. 1-16. DOI: 10.1145/2901318.2901322. URL: http://dl.acm.org/citation.cfm?doid=2901318.2901322.

[4] Markus G. Kuhn. «Electromagnetic Eavesdropping Risks of Flat-Panel Displays». En: May (2010), págs. 88-107. DOI: 10.1007/11423409_7.

[5] Daniel Genkin, Adi Shamir y Eran Tromer. «RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis». En: *Journal of Cryptology* 3, no.3, J (2014), págs. 444-461. ISSN: 16113349. DOI: 10.1007/978-3-662-44371-2_25. URL: http://link.springer.com/10.1007/978-3-662-44371-2_25.

[6] Raphael Spreitzer, Veelasha Moonsamy, Thomas Korak y Stefan Mangard. «Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices». En: *IEEE Communications Surveys and Tutorials* 20.1 (2018), págs. 465-488. ISSN: 1553877X. DOI: 10.1109/COMST.2017.2779824. arXiv: arXiv:1611.03748v3.

[7] Gerald J Popek y Robert P Goldberg. «Formal requirements for virtualizable third generation architectures». En: *Communications of the ACM* 17.7 (1974), págs. 412-421. ISSN: 00010782. DOI: 10.1145/361011.361073. URL: http://portal.acm.org/citation.cfm?doid=361011.361073.

[8] Docker. *Docker Overview*. 2019. URL: https://docs.docker.com/engine/docker-overview/ (visitado 22-08-2019).

[9] Paul Kocher, Joshua Jaffe y Benjamin Jun. «Differential Power Analysis». En: *Encyclopedia of Cryptography and Security*. Vol. 1666. 1999, págs. 388-397. ISBN: 978-3-540-66347-8 978-3-540-48405-9. DOI: 10.1007/3-540-48405-1_25. URL: https://link.springer.com/content/pdf/10.1007/3-540-48405-1_25.pdf.

[10] Paul Kocher, Joshua Jaffe, Benjamin Jun y Pankaj Rohatgi. «Introduction to differential power analysis». En: *Journal of Cryptographic Engineering* 1.1 (2011), págs. 5-27. ISSN: 21908508. DOI: 10.1007/s13389-011-0006-y.

[11] Michael Hutter y Jörn Marc Schmidt. «The temperature side channel and heating fault attacks». En: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8419 LNCS (2014), págs. 219-235. ISSN: 16113349. DOI: 10.1007/978-3-319-08302-5_15.

[12] Chen Sun, Chia-hsin Owen Chen, George Kurian, Lan Wei, Jason Miller, Anant Agarwal, Li-shiuan Peh y Vladimir Stojanovic. «DSENT – A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic». En: *Nocs* (2012).

[13] Runjie Zhang, Mircea R Stan y Kevin Skadron. «HotSpot 6.0 : Validation , Acceleration and Extension». En: (2015), págs. 0-7.

[14] Murugappan Alagappan, Jeyavijayan Rajendran, Milos Doroslovacki y Guru Venkataramani. «DFS covert channels on multicore platforms». En: *IEEE/IFIP International Conference on VLSI and System-on-Chip, VLSI-SoC*. IEEE Computer Society, 2017. ISBN: 9781538628805. DOI: 10.1109/VLSI-SoC.2017.8203469.

[15] Mordechai Guri, Yosef Solewicz, Andrey Daidakulov y Yuval Elovici. «Fansmitter: Acoustic Data Exfiltration from (Speakerless) Air-Gapped Computers». En: (2016). arXiv: 1606.05915. URL: http://arxiv.org/abs/1606.05915.

[16] CCN. *Certificación TEMPEST*. URL: https://www.ccn.cni.es/index.php/es/menu-organismo-de-certificacion-es/certificacion-tempest-menu-es (visitado 25-08-2019).

[17] Raghavendra Pradyumna Pothukuchi, Sweta Yamini Pothukuchi, Petros Voulgaris y Josep Torrellas. «Maya: Falsifying Power Sidechannels with Operating System Support». En: (2019). arXiv: 1907.09440. URL: http://arxiv.org/abs/1907.09440.

[18] Shengshuo Lu, Zhengya Zhang y Marios Papaefthymiou. «A 1.25pJ/bit 0.048mm 2 AES core with DPA resistance for IoT devices». En: *2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. Vol. 2. 2142. IEEE, 2017, págs. 65-68. ISBN: 978-1-5386-3178-2. DOI: 10.1109/ASSCC.2017.8240217. URL: http://ieeexplore.ieee.org/document/8240217/.

[19] Belén Zalba Nonay. «Almacenamiento Térmico de Energía Mediante Cambio de Fase. Procedimiento Experimental». Tesis doct. Universidad de Zaragoza, 2002.

[20] Dariel Figueredo Piñero. *Análisis y optimización de prestaciones y eficiencia energética de un servidor para centros de datos*. 2018. URL: https://zaguan.unizar.es/record/78605/files/TAZ-TFG-2018-4549.pdf.

[21] Wikichip. *Arquitectura Skylake Server*. 2019. URL: https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server) (visitado 24-08-2019).

[22] ANSI/ASHRAE Standards Committee. *Energy Standard for Data Centers (ANSI Approved). ANSI/ASHRAE Standard 90.4-2016*. 2016.