VM間RowHammer防止技術のgem5を用いた検証システム

川崎 秀昌[1]　穐山 空道[2,a)]

概要：クラウド環境で VM 間 RowHammer による攻撃が脅威になっており、VM 間 RowHammer を防ぐ Hypervisor が求められている。しかし、VM 間 RowHammer を防ぐためには DRAM 上で隣接したメモリ領域を知る必要がある。DRAM 上で隣接したメモリ領域を知るためには、CPU 内部の MMU と DRAM のアドレス変換関数が必要である。gem5 は CPU や DRAM の動作を模倣するシミュレータであり、gem5 上で動作するソフトウェアは DRAM 上でのメモリ配置を把握できる。本研究では VM 間 RowHammer を防止技術を備えた Hypervisor を実現に向け、gem5 を用いた VM 間 RowHammer 防止技術を検証するシステムの実現を目指す。提案システムでは gem5 上に Bao を用いて、2 つの Linux の起動に成功した。

1. はじめに

ある VM から他の VM に対してビット反転を起こす VM 間 RowHammer という攻撃がある。RowHammer は DRAM 上で隣接したメモリ領域をアクセスせずにビット反転を起こすサイドチャネル攻撃である。ビット反転によってユーザが意図しない VM の停止や機密データの改ざんが起こりうる [1]。

クラウドの信頼性を保つために VM 間 RowHammer を防ぐ Hypervisor が求められている。クラウド環境では単一 Hypervisor 上に複数の企業が使用する VM が動作しており、データ改ざんが行われることでクラウドベンダーとデータ改ざんされた企業に対して経済的な損失をもたらす。

VM 間 RowHammer を防ぐには DRAM 上で隣接したメモリ領域を知る必要があり、CPU と DRAM のリバースエンジニアリングが必要である。CPU には物理アドレスから DRAM アドレスに変換するメモリコントローラ (MC) がある。DRAM にはソフトウェアレベルで確認できない内部動作がある。MC のアドレス変換関数と DRAM の内部動作を明らかにする必要がある。しかし、リバースエンジニアリングはエンジニアリングコストが高く、詳細な解析は困難である。

本研究では VM 間 RowHammer を防止する Hypervisor を実現に向け、マイクロアーキテクチャシミュレータの gem5 を用いた VM 間 RowHammer 防止技術を検証するシステムの実現を目指す。gem5 を用いることで CPU や DRAM の内部動作を確認でき、リバースエンジニアリングの必要はない。

[1] 立命館大学 情報理工学研究科
[2] 立命館大学 情報理工学部
[a)] s-akym@fc.ritsumei.ac.jp

2. gem5 上での Hypervisor 動作手法

2.1 gem5

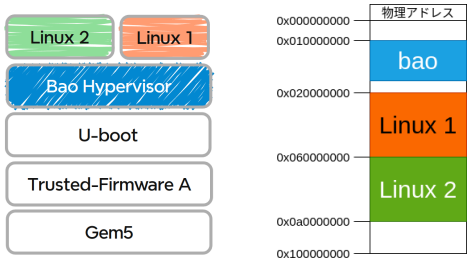
gem5 はオープンソースのマイクロアーキテクチャシミュレータである。gem5 はソフトウェアによるシミュレータであるため、CPU やメモリの内部処理を変更できる。gem5 によってリバースエンジニアリングを必要とせずに VM 間 RowHammer を起こせる。Arm アーキテクチャシミュレーションのフルシステムモードは仮想化支援機構が実装されており、Hypervisor の動作が可能である。gem5 は Fixed Virtual Platform (FVP) 互換の環境を提供しているため、FVP で動作する Trusted Firmware-A (TFA) を変更なしで動かせる [2]。FVP は Arm 社が提供しているシミュレーション環境であり、Arm アーキテクチャ限定である [3]。

2.2 実装

本稿では、gem5 上で Arm アーキテクチャにおける Hypervisor を動作させる。gem5 上で動作させる Hypervisor として FVP をサポートしている Bao を用いる [4]。gem5 の FVP は Ethernet を提供しないため、Linux のデバイスツリーから Ethernet を消去する。Linux の起動時に X509 証明書の読み込み処理が進行しなかったため、カーネル設定で無効化した。Bao の起動は gem5 上に TFA を起動させ、TFA により U-Boot、U-Boot より Bao を起動する。gem5 の semihosting を用いてホストコンピュータから Bao

表 1: 実装環境のシステム設定

gem5	version 24.0.0.1
	CPU: 4 cores
	Workflow : ArmTrustedFirmware
	Machine Type: VExpress_gem5_Foundation
Software	Trusted Firmware-A : v2.9
	U-Boot : v2022.10
	Bao Hypervisor : demo
	Linux : v6.1



(a) 実装システムの概要　(b) 物理メモリ配置

図 1: 実装したシステム

の実行ファイルを読み込み、実行する。

3. VM 間 RowHammer の検証システム

本章では gem5 を用いて VM 間 RowHammer の防御手法を容易に検証可能なシステムを提案する。検証は VM 間 RowHammer の防御手法を本システムで動作させ、VM 間 RowHammer が起きないと示すことである。

VM 間 RowHammer の脅威モデルは以下である。

- 攻撃者はクラウド上に VM を構築できる。
- 攻撃者は VM 上の OS の特権を持つ。
- 攻撃者は自身の VM に隣接する他 VM は認識できない。

攻撃者は VM 上の OS の特権を持つためカーネルモジュールや eBPF などを用いて、VM に割り当てられた物理アドレスに直接アクセスするページを対応付けられる。

構築したシミュレーション環境のシステム設定を表 1 に示す。図 1a に実装システムの全体像、図 1b に物理メモリアドレス上の VM 配置を示す。

本研究では意図的に VM 間 RowHammer を起こすために 2 つの VM を作成し、2 つの VM のメモリ領域が DRAM 上で隣接するように配置する。本環境では VM1 と VM2 を物理アドレス上で隣接するように配置した。

本環境では物理アドレスから DRAM アドレスに変換する関数をリバースエンジニアリングなしに取得できる。gem5 が提供している物理アドレスから DRAM アドレスに変換する関数の一例を図 2 に示す。DRAM 上で隣接するメモリは Rank と Bank がそれぞれ同一であり、Row の値が連続値である。例として本環境の DRAM 上に配置されたメモリ領域の一部を図 3 に示す。VM1 のアドレス領域内の 0x05ff80000 は Row 3071 であり、VM2 のアドレス領域内の 0x060000000 は Row 3072 である。以上で示したアドレスは Rank 0、Bank 0 で連続である。

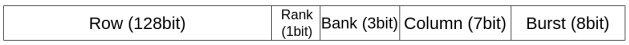


図 2: 物理アドレスから DRAM アドレスへの変換

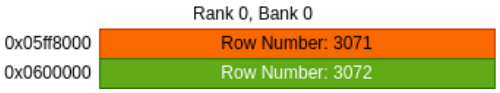


図 3: DRAM 上のメモリ配置

4. 関連研究と今後の課題

関連研究として、gem5 の RISC-V アーキテクチャのフルシステムモードを用いて Hypervisor を動作させる研究がある [5]。RISC-V の M モードを使用している Hypervisor の動作を実現している。本研究は gem5 上で Arm アーキテクチャの Hypervisor を起動できる点が異なる。

その他に OP-TEE を gem5 上で動作させる TEE-Time がある [6]。TEE-Time は OP-TEE のキャッシュタイミング攻撃を調査するシステムである。本研究は Hypervisor を gem5 上に構築することでメモリサイドチャネル攻撃やキャッシュサイドチャネルに対して Hypervisor が堅牢かを確認できる点が異なる。

今後の課題は、既存の Hypervisor 上で VM 間 RowHammer を起こせるかを確認し、VM 間 RowHammer の防御手法を容易に評価できるシステムを構築することである。

謝辞　本研究は JST，さきがけ，JPMJPR22P1 の支援を受けたものである．

参考文献

[1] Loughlin, K., Rosenblum, J., Saroiu, S., Wolman, A., Skarlatos, D. and Kasikci, B.: Siloz: Leveraging DRAM Isolation Domains to Prevent Inter-VM Rowhammer, *SOSP*, p. 417–433 (2023).

[2] Adrian, H.: Running Trusted Firmware-A on gem5, `https://community.arm.com/arm-research/b/articles/posts/running-trusted-firmware-a-on-gem5` (2020).

[3] Arm: Fixed Virtual Platform (FVP), `https://www.arm.com/ja/products/development-tools/simulation/fixed-virtual-platforms`.

[4] Martins, J., Tavares, A., Solieri, M., Bertogna, M. and Pinto, S.: Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems, *NG-RES*, pp. 3:1–3:14 (2020).

[5] Peter, Y. H. H., Xiongfei, L., Jin, C., Andrea, M., Thannirmalai, M. S. and Naxin, Z.: Supporting RISC-V Full System Simulation in gem5, *CARRV* (2021).

[6] Forcioli, Q., Chaudhuri, S. and Danger, J.-L.: TEE-Time: A Dynamic Cache Timing Analysis Tool for Trusted Execution Environments, *ISQED*, pp. 1–8 (2024).