CPUの脆弱性
MeltdownおよびSpectreの攻撃シナリオ

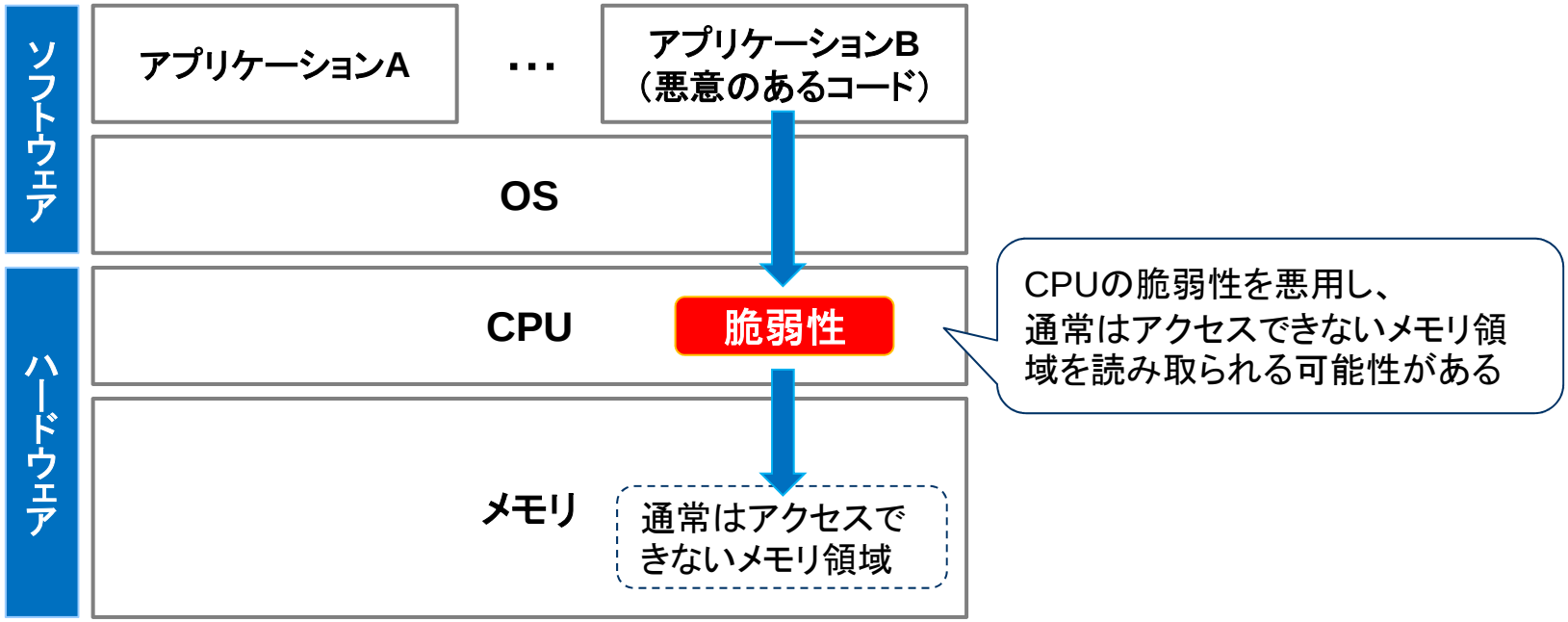
2018年1月31日

セクタンラボ

脆弱性の概要

- MeltdownおよびSpectreは、CPUのハードウェアレベルの脆弱性であり、悪意のあるコードから、カーネルメモリや他プロセスのメモリ領域を読み取られる可能性がある。
- 本脆弱性は、CPUの処理速度を最適化する機能「投機的実行（Speculative Execution）」の仕組みに起因しており、OSなどソフトウェア側で対策を実施すると、処理速度の低下を伴う可能性があることから、根本的な問題解決には、CPU側での対処が必要となる。

◆脆弱性のイメージ図



攻撃シナリオ （2018年1月29日現在）

- **本脆弱性の特徴を踏まえた攻撃シナリオを、下表に記載する。**
  （注意）今後、新たな攻撃コードの発表により状況が変わる可能性がある。

脆弱性の名称	CVE番号（通称）	攻撃に必要な権限	読み取れるメモリ	想定攻撃経路	攻撃シナリオ
スペクター Spectre	CVE-2017-5753 (Variant 1 Bounds check bypass)	一般ユーザー	自プロセス	ウェブ	利用者が、悪意あるJavaScriptが設置されたウェブサイトにアクセスすると、他のウェブサイトで入力したパスワードなど、ブラウザのプロセスメモリに記録されている機密情報が窃取される。
				ローカル	攻撃者が、クラウドのLinuxサーバ（eBPF機能を有するもの※1）など、仮想マシン上で悪意のあるコードを実行し、他の仮想マシンに記録されている機密情報を窃取する。
	CVE-2017-5715 (Variant 2 Branch target injection)	特権ユーザー（root権限）	他プロセス	ローカル	攻撃成功の制約条件が厳しく、現時点では現実的な攻撃シナリオなし
メルトダウン Meltdown	CVE-2017-5754 (Variant 3 Rouge cache data load)	一般ユーザー	カーネル	ローカル	攻撃者が、クラウドのサーバなど、仮想マシン上で悪意のあるコードを実行し、他の仮想マシンのメモリに記録されている機密情報を窃取する。

（※1）eBPF機能を有するLinuxでは、一般ユーザー権限でカーネルのインタープリタに任意のバイトコードを送信することができる。
つまり、カーネルの権限で任意のコードを実行できるため、カーネルメモリを読み取ることが可能となる。

参考文献

- **Meltdown/Spectre特設サイト**
  - https://meltdownattack.com/
- **Meltdown 論文**
  - https://meltdownattack.com/meltdown.pdf
- **Spectre 論文**
  - https://spectreattack.com/spectre.pdf
- **Google Project Zero**
  - https://googleprojectzero.blogspot.jp/2018/01/reading-privileged-memory-with-side.html
- **GitHub（攻撃実証コード）**
  - https://github.com/topics/meltdown

（参考）脆弱性の理解につながる技術情報

投機的実行（Speculative Execution）

- CPUの処理速度と比較すると、メモリの読み書き速度は100倍以上も遅く、ボトルネックになるため、CPU内部には、高速に読み書きできる「キャッシュメモリ」が搭載されており、今後実行するコードや頻繁に利用するデータなどをキャッシュメモリに格納し、処理の高速化を図っている。

- また、CPUは、原則としてプロセスのコードを順番に実行していくが、例えば条件分岐（IF文）などメモリの読み込み処理で時間がかかる場合、CPUは条件分岐の判定処理を待たずに、分岐先を予測しコードの実行を進めてキャッシュする。
- 条件分岐の判定処理が確定し、予測が外れていた場合は、予測したコードの処理結果を取り消したうえで、処理をやり直しとなるが、予測が当たっていた場合は、メモリ読み込みの待ち時間を無駄にせずにコードの実行を進めることができるため、処理の高速化につながる。CPUのこのような仕組みを「投機的実行」という。

Meltdown/Spectreの基本概念

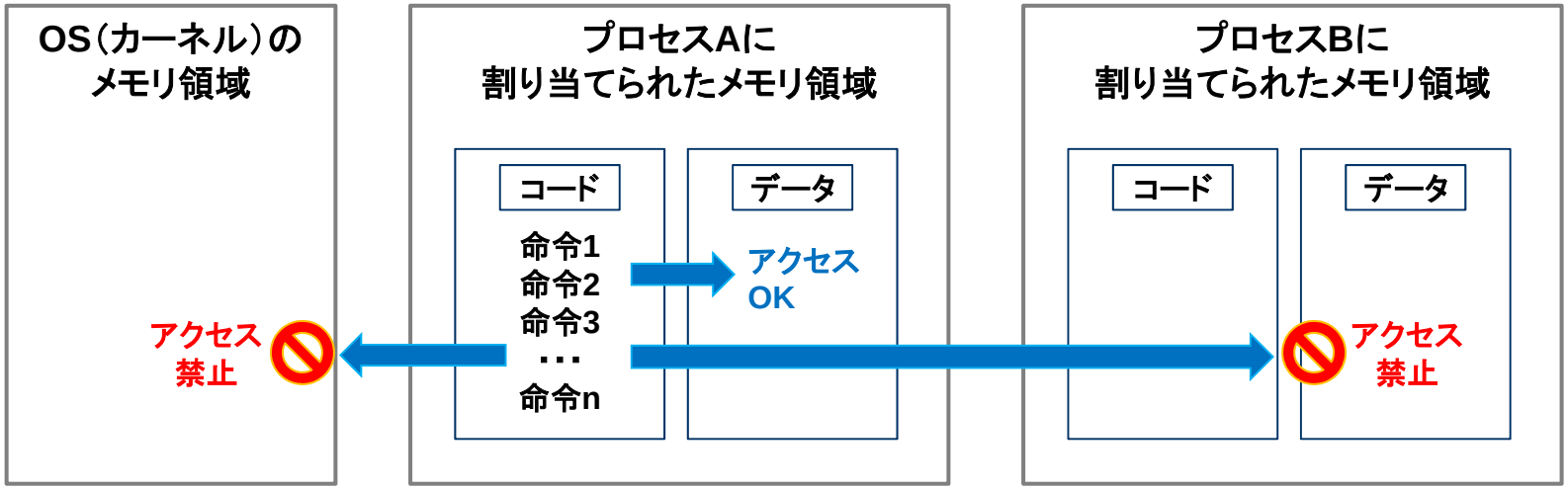
- 一般的なプログラムでは、メモリの読み書きの際に、アクセスして良い領域なのか判断し、条件分岐する処理を行う場合があるが、投機的実行の分岐予測を誤った方向に誘導することで、投機的実行として、本来であればアクセスできないメモリ領域にアクセスすることができる。
- CPUが、分岐予測が外れていたことを認識した時点で処理は取り消しされるが、メモリアクセスの痕跡がキャッシュメモリに残るため、メモリアクセス速度を計測することで、投機的実行により不正に読み取ったメモリの値を推測することができることが判明した。

- Meltdown、Spectreは、いずれも投機的実行の仕組みを悪用し、不正にメモリを読み取る攻撃手法の名称であり、具体的な悪用方法の違いにより、異なる名称がつけられているものである。

メモリ管理の仕組み

- OSは、起動したプログラム（プロセス）ごとに、独立したメモリ領域を割り当てし、コードとデータを配置する。各プロセスは、原則として自プロセスに割り当てられたメモリ領域に対してのみアクセスが許可されており、OS（カーネル）や他プロセスのメモリ領域にはアクセスできない。
- しかし、処理の効率化のため、各プロセスのメモリ領域のアドレス空間のなかに、カーネルのメモリ領域のアドレスもマッピングされているため、Meltdown/Spectreにより、カーネルのメモリ空間にアクセスできてしまう。
  - 逆にいえば、カーネルのページテーブルを分離すれば、各プロセスからカーネルメモリへのアクセスはできなくなる。

◆メモリ管理のイメージ図



※実際には、カーネルのメモリ領域のアドレスが、各プロセスのアドレス空間の中にマッピングされている。