Funktionsweise und Evaluation von modernen Spectre-Angriffen

Abschlussvortrag zur Bachelorarbeit

Jan-Niklas Sohn, Betreuer: Dr. Felix Jonathan Boes

24. Juni 2021

Rheinische Friedrich-Wilhelms-Universität Bonn

Gliederung

- Thema und Ziel
- Grundlagen
- Spectre-Angriffe allgemein und konkret
- Evaluation: Methodik und Ergebnisse

Thema

- Spectre und Meltdown (2018)
- Forschungsfeld: Sicherheit spekulativer Ausführung
- Mittlerweile viele Varianten
- Weitreichende Konsequenzen für bestimmte Sicherheitsmodelle
  - Cloud Computing
  - JavaScript im Browser
  - Betriebssystem-Kerne

Ziel

- Funktionsweise konkreter Spectre-Angriffe
    - RIDL, ZombieLoad, Write Transient Forwarding, Store-to-Leak
- Implementierung in verschiedenen Varianten
- Evaluation hinsichtlich einheitlicher Metriken

Ziel

- Funktionsweise konkreter Spectre-Angriffe
  - RIDL, ZombieLoad, Write Transient Forwarding, Store-to-Leak
- Implementierung in verschiedenen Varianten
- Evaluation hinsichtlich einheitlicher Metriken
- Ergebnisse:
  - RIDL, ZombieLoad und Store-to-Leak reproduziert
  - Write Transient Forwarding nicht

Grundlagen

Speicher und Caches

Physischer und virtueller Speicher

- Isolation von Prozessen durch virtuelle Adressräume
- Zuordnung zwischen virtuellen und physischen Adressen auf *Page*-Ebene
- Fester Teil des Adressraums für Nutzerprozess und Betriebssystem-Kern

Caches

- Caches verringern Latenz eines Speicherzugriffs
- Kleiner und schneller als der Hauptspeicher
- Speicherzugriff erfolgt erst auf den Cache
- Verwaltet in *Cachezeilen*

Branch Prediction

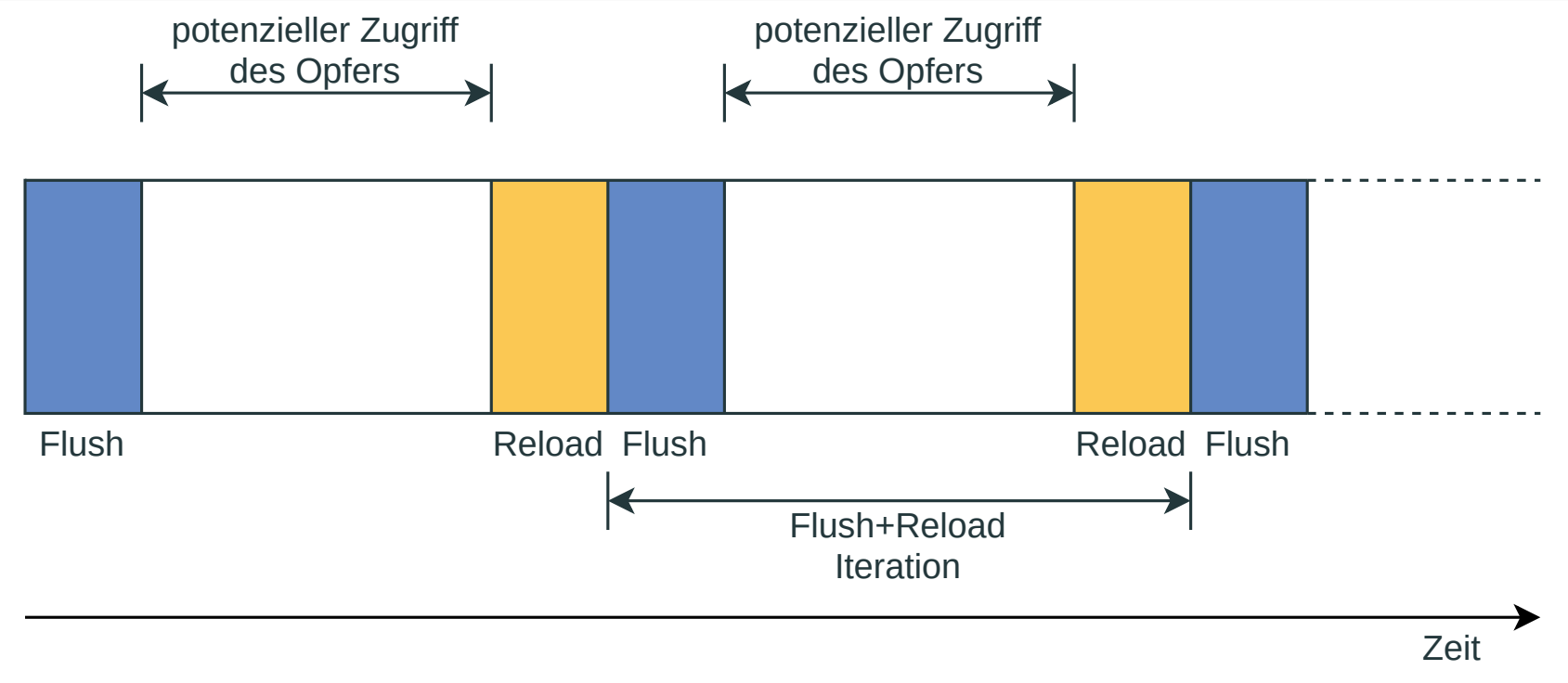
- Bedingung eines Sprunges nicht bekannt: *Branch Predictor* sagt Kontrollfluss voraus
- Folgende Instruktionen werden bereits ausgeführt
    - *Speculative Execution*
- Vorhersage korrekt: Ergebnisse werden übernommen
- Vorhersage falsch: Ergebnisse werden verworfen und der korrekte Pfad wird ausgeführt

Transient Execution

- Speculative Execution auf inkorrektem Pfad
- Alternativ: Tritt auch bei Prozessor-Exceptions auf
- Ergebnisse der Transient Execution werden verworfen
- Zustand des Caches wird nicht zurückgesetzt!

Flush+Reload

- Cache-basierter Seitenkanalangriff
  - Nutzt Unterschiede in der Zugriffszeit, um Informationen abzuleiten
- Angreifer kann Zugriff auf Cachezeile detektieren
- Ablauf:

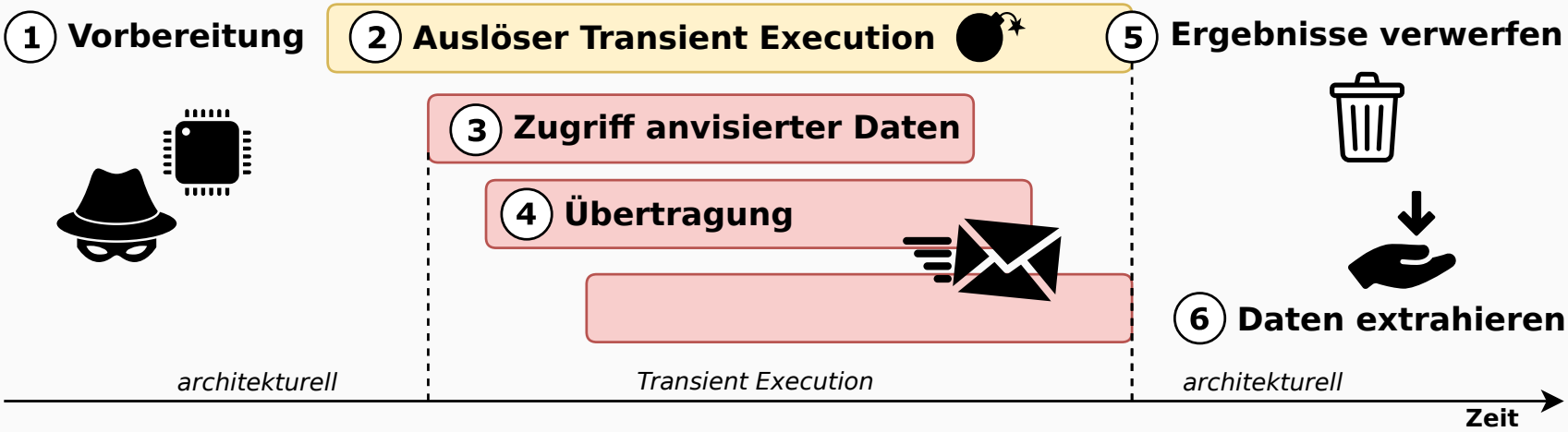


Flush+Reload

- Flush+Reload gleichzeitig für unterschiedliche Cachezeilen
- Byte aus Transient Execution übertragen:
  - Transient Execution lädt eine von 256 Cachezeilen
  - Anschließend eingeladene Cachezeile durch Flush+Reload bestimmen

Spectre-Angriffe

Spectre-Angriffe allgemein



- **Phase 1**: Flush der Cachezeilen
- **Phase 2**: Eintritt in Transient Execution
- **Phase 3**: Zugriff auf anvisierte Daten
- **Phase 4**: Einladen einer Cachezeile, abhängig von extrahierten Daten
- **Phase 5**: Ende der Transient Execution
- **Phase 6**: Bestimmung der eingeladenen Cachezeile durch Flush+Reload

Spectre-Type vs. Meltdown-Type

- Unterschieden nach Art der Transient Execution:

Spectre-Type

- Transient Execution durch falsche Branch Prediction

Meltdown-Type

- Transient Execution durch Prozessor-Exception
- Weiter unterschieden nach:
  - Art der Prozessor-Exception
  - Element des Prozessors, aus dem Daten extrahiert werden
- Ausgelöste Prozessor-Exception wird behandelt oder unterdrückt

Konkrete Spectre-Angriffe

Angriff	Prozessor-Exception	Quelle extrahierter Daten
RIDL	Page-Fault	Line-Fill Buffer
ZombieLoad	General Protection	Store Buffer
WTF	Microcode Assist	Line-Fill Buffer

- Store-to-Leak beobachtet Anwesenheit von Speicherzuordnungen im Adressbereich des Betriebssystem-Kerns

Evaluation

Methodik

Umgebung

- Intel Core i5-8250U, Linux 5.10
- KASLR und Maßnahmen gegen Spectre-Angriffe deaktiviert
- Minimale Systemlast

Angriffsszenario

- Opfer-Prozess liest oder schreibt wiederholt einen festen Wert
- Angreifender Prozess extrahiert diesen Wert

Methodik

Erfasste Metriken

- Erfolgsrate: Anteil der korrekt ermittelten Bytes
- Datenrate: Berechnet aus Dauer des Angriffs

Verschiedene Varianten

- Speicherzugriffe des Opfers: Lesend oder schreibend
- Prozessor-Exception: Behandelt oder unterdrückt

Ergebnisse

Variante	Erfolgsrate (%)	Datenrate (B/s)
RIDL Basis	97,96	443,9
RIDL Load	56,27	450,6
RIDL Signal	74,69	745,5
RIDL Transient	0,000	757,1
ZombieLoad Basis	94,31	747,1
ZombieLoad Load	93,25	768,2
ZombieLoad Transient	99,70	780,3
WTF Basis	0,000	752,1
WTF Transient	0,000	764,1

- Store-to-Leak: Ermittelt Basisadresse des Betriebssystem-Kerns in 0,5 ms

Zusammenfassung und Ausblick

Zusammenfassung

- Spectre-Angriffe, Spectre-Type vs. Meltdown-Type
- Evaluation:
  - RIDL, ZombieLoad, Store-to-Leak erfolgreich reproduziert
  - Write Transient Forwarding nicht reproduziert

Weitere Forschungsmöglichkeiten

- Spectre-Type Angriffe
- Auf weiteren Systemen evaluieren
- Andere Angriffsszenarien
- Mit aktivierten Gegenmaßnahmen

Quellen

- Abbildung auf Folie 10 modifiziert von Abbildung 3.1 in:
  - Gruss, Daniel: „Transient-Execution Attacks“, 2020, URL: https://gruss.cc/files/habil.pdf (besucht am 15.01.2021)