Covert Channels

Implementation and evaluation of a covert channel built with with FLUSH+FLUSH

Was sind verdeckte Kanäle?

Verdeckte Kanäle beschreiben die unautorisierte Kommunikation zwischen zwei Kommunikationspartnern.

Was sind Side-Channel Attacks?

Die Observation von Systemen und Ausnutzung von öffentlichen Daten um Schlüsse über die eigentlich privaten Informationen zu erhalten.

Cachebasierte verdeckte Kanäle

- Nutzen aus, dass der Cache von unabhängigen Prozessen geteilt wird
- Techniken geben Auskunft darüber, ob der Sender bestimmte Speicherbereiche benutzt
- Unterscheidung zwischen shared-memory und Prime+Probe
    - shared-memory nutzt z.B. Unterschiede in Zugriffszeiten auf den Cache aus
    - Prime+Probe nutzt aus, dass verschiedene Speicherstellen auf eine Cachezeile mappen

Shared-Memory-Attacken

- Verwendung von CLFLUSH, um Daten aus dem Cache zu verdrängen
- Flush+Reload:
  - Messung von Zugriffszeiten
- Flush+Flush
  - Messung von CLFLUSH-Execution-Time

Flush+Reload

1. CLFLUSH um Daten zu verdrängen
2. Sender lädt ggf. Daten erneut
3. Erneutes laden der Daten durch Empfänger

- Falls kurze Zugriffszeit -> Sender hat Daten verwendet -> 1
- Falls lange Zugriffszeit -> Sender hat Daten NICHT verwendet ->0

Flush+Flush

1. CLFLUSH um Daten zu verdrängen
2. Sender lädt ggf. Daten erneut
3. Erneutes CLFLUSH durch Empfänger
   - Falls **lange** Execution-Time -> Sender hat Daten verwendet -> 1
   - Falls **kurze** Execution-Time -> Sender hat Daten NICHT verwendet ->0

CLFLUSH kann bei vorzeitig abbrechen, wenn die Daten NICHT im Cache liegen

Anschließend wiederholte Schritte 2,3

Vorteile von Flush+Flush

- Kein direkter Zugriff -> keine Cache Misses -> Stealthy gegenüber Hardwarecountern
- Prefetcher wird nicht ausgelöst
- Mehrere Cachelines werden abgehört, erhöhte Bandbreite

Einordnung in den Kontext - Timeline

1996 - erste Beschreibung von Seitenkanälen in der Wissenschaft. (Kocher)

2006 - Evict+Time und Prime+Probe werden vorgestellt. (Osvik)

2014 - Flush+Reload wird vorgestellt. (Yarom)

2016 - Flush+Flush wird vorgestellt. (Gruß)

2017 - Automatisierung von Seitenkanalangriffen (Gruß)

2018 - Die Meltdown Sicherheitslücke bei Intel-CPUs wird offengelegt. (Lipp)

2019 - Die Spectre Verwundbarkeit wird zusätzlich veröffentlicht. (Kocher)

Einordnung in den Kontext - Arbeiten

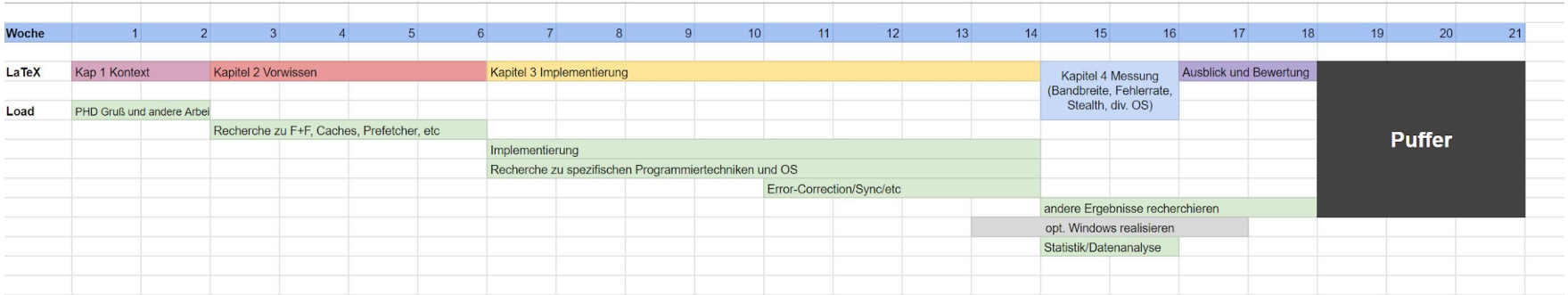
- Paul C. Kocher. “Timing Attacks on Implementations of Diffe-Hellman, RSA, DSS, and Other Systems”. In: Crypto. 1996.
- Daniel Gruss. “Software-based Microarchitectural Attacks”. Diss. Graz University of Technology, 2017. URL: https://gruss.cc/files/phd_thesis.pdf.
- Paul Kocher u. a. “Spectre Attacks: Exploiting Speculative Execution”. In: S&P. 2019.
- Moritz Lipp u. a. “Meltdown: Reading Kernel Memory from User Space”. In: USENIX Security Symposium. 2018.
- Yuval Yarom und Katrina Falkner. “Flush+Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack”. In: USENIX Security Symposium. 2014.
- Daniel Gruss u. a. “Flush+Flush: A Fast and Stealthy Cache Attack”. In: DIMVA. 2016.
- Dag Arne Osvik, Adi Shamir und Eran Tromer. “Cache Attacks and Countermeasures: the Case of AES”. In: CT-RSA. 2006.
- Maurice, Clémentine, et al. "Hello from the Other Side: SSH over Robust Cache Covert Channels in the Cloud." *NDSS*. Vol. 17. 2017.

Weitere Recherche

- Prime+Probe
  - Set Agreement
  - RS Code, Berger Code, Hadamard Code
- Gulash et al: Attack on Scheduling
- Pessl et al: Attack on DRAM
- Rowhammer
- RDTSC
- Networking: CSMA, Syncronisation, Self Clocked Signal
- Cache Slice, Complex Mapping Function, Cache Replacement Policy, Pages
- Noise by Full Duplex?
- Noise by increased word size?
- Content based page duplication
- Prefetcher
- CLFLUSH timing to derive which core is used
- Flush+Flush attack on AES
- MFENCE/LFENCE
- Table look aside buffer
- Daniel Gruß Dr-Arbeit
- Gantt Chart
- Einordnung in den Kontext, welche anderen dinge gibts
  Welche implementierungen gibt es bereits
- Kuniyasu Suzaki u. a. “Memory Deduplication as a Threat to the Guest OS”. In: EuroSec. 2011.: Seitenkanalangriffe auf RAM
- Prefetch Seitenkanalangriffe (Gruß)

Struktur

Ganttchart


Woche
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
LaTeX
Kap 1 Kontext
Kapitel 2 Vorwissen
Kapitel 3 Implementierung
Kapitel 4 Messung (Bandbreite, Fehlerrate, Stealth, div. OS)
Ausblick und Bewertung
Load
PHD Gruß und andere Arbei
Recherche zu F+F, Caches, Prefetcher, etc
Implementierung
Recherche zu spezifischen Programmiertechniken und OS
Error-Correction/Sync/etc
andere Ergebnisse recherchieren
opt. Windows realisieren
Statistik/Datenanalyse
Puffer

Forschungsfrage?

- Performance-, also Genauigkeit/Bandbreite/Stealth, Vergleich zwischen:
  - OS
  - Architekturen
  - Einfluss von anderen Variablen (Clock Speeds, #Cores, laufende Anwendungen, ...)
- Vergleich mit State-of-the-art Implementierungen
- Diskussion von Implementierungsdetails