CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Your CPU Is the New Software

Exploiting Architectural Bugs at Hardware Speed

Michael Schwarz | THCon | May 2026

Who am I

Dr. Michael Schwarz

Faculty @ CISPA

Research on CPU security and side channels

michael.schwarz@cispa.de

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

ÆPIC Leak

Architectural leaks pioneer

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

ÆPIC Leak

Architectural leaks pioneer

PLATYPUS

Software power side channels

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

ÆPIC Leak

Architectural leaks pioneer

PLATYPUS

Software power side channels

ZombieLoad, **Fallout**, **Medusa**, **CacheWarp**, **GhostWrite**, and more

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

ÆPIC Leak

Architectural leaks pioneer

PLATYPUS

Software power side channels

ZombieLoad, **Fallout**, **Medusa**, **CacheWarp**, **GhostWrite**, and more

KPTI, **Cloudflare**

Defenses in production

My Past 10 Years

Spectre, **Meltdown**

Transient execution attacks

ÆPIC Leak

Architectural leaks pioneer

PLATYPUS

Software power side channels

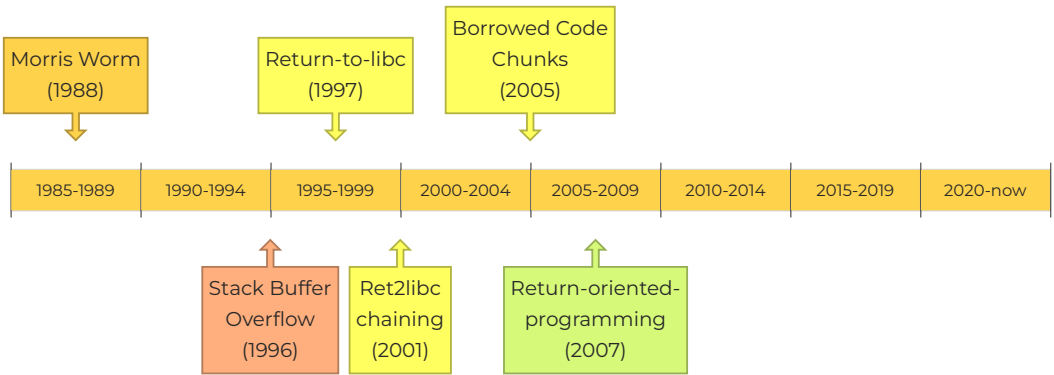
ZombieLoad, **Fallout**, **Medusa**, **CacheWarp**, **GhostWrite**, and more

KPTI, **Cloudflare**

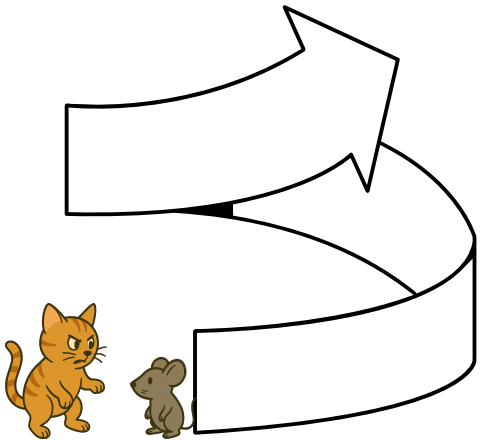
Defenses in production

"Yes, I probably broke your CPU."

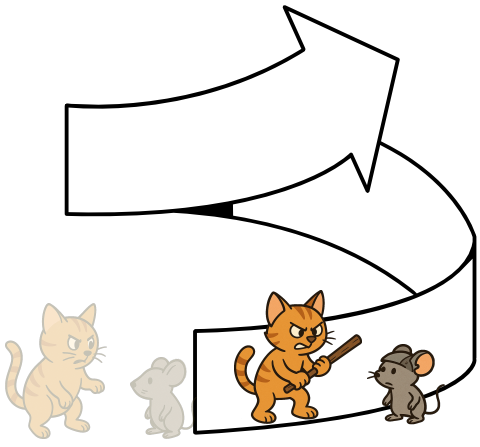
Short History of Software Exploitation



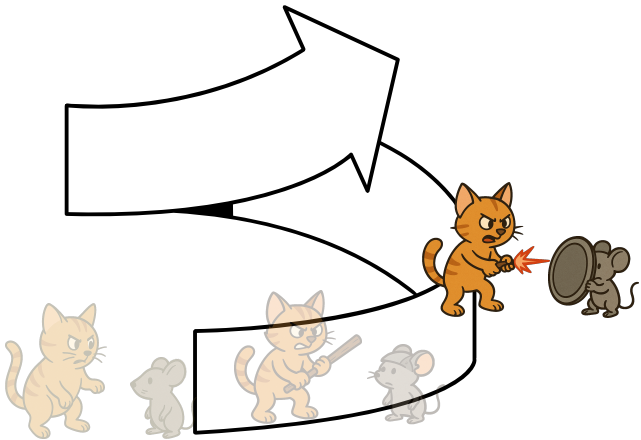
The Security Spiral (Not a Cycle!)



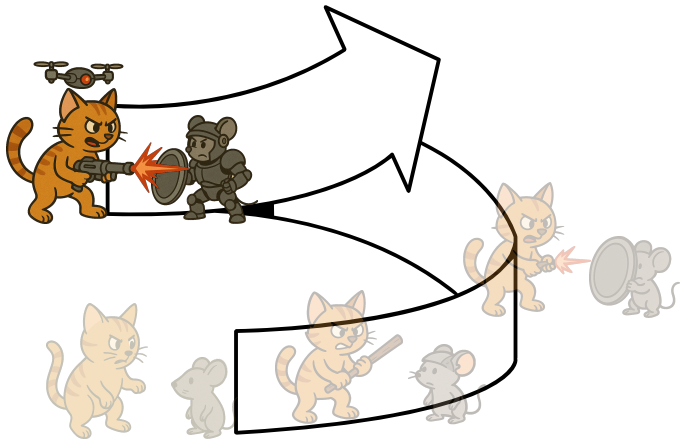
The Security Spiral (Not a Cycle!)



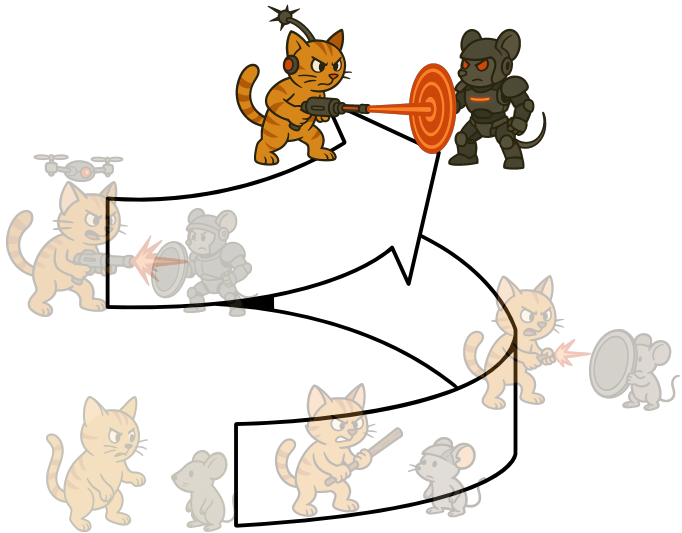
The Security Spiral (Not a Cycle!)



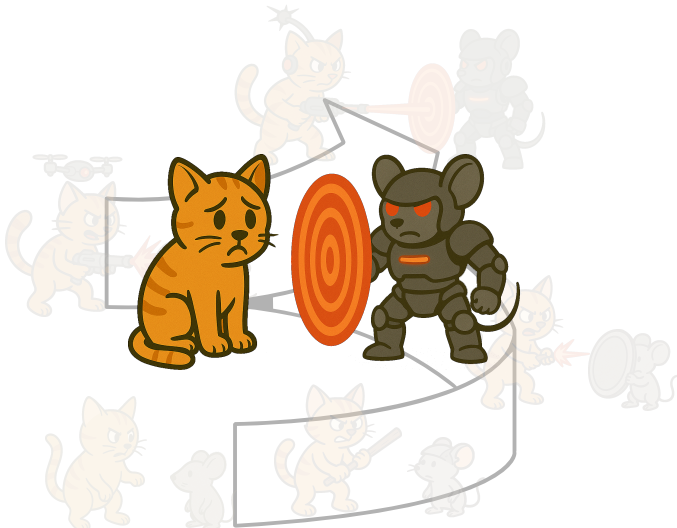
The Security Spiral (Not a Cycle!)



The Security Spiral (Not a Cycle!)

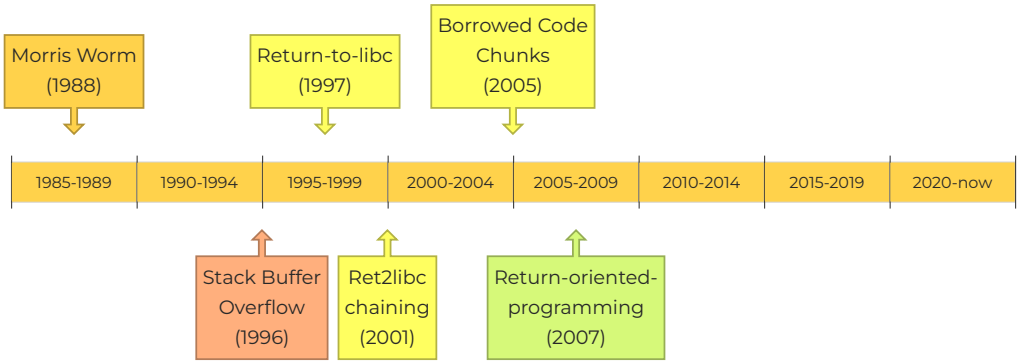


The Security Spiral (Not a Cycle!)

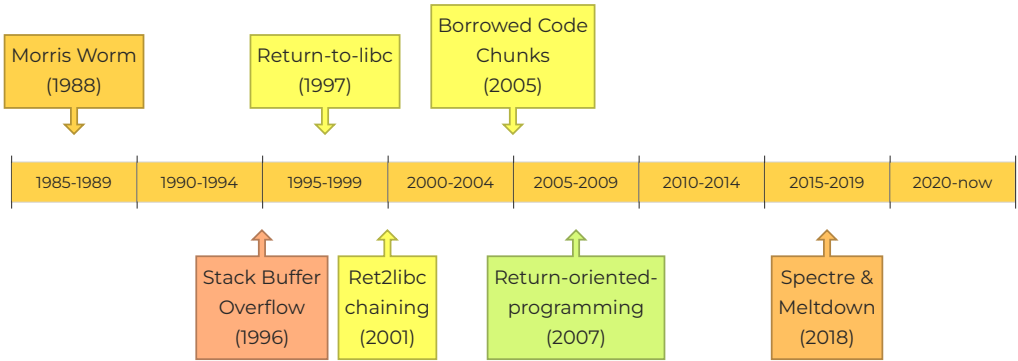


The Security Spiral (Not a Cycle!)

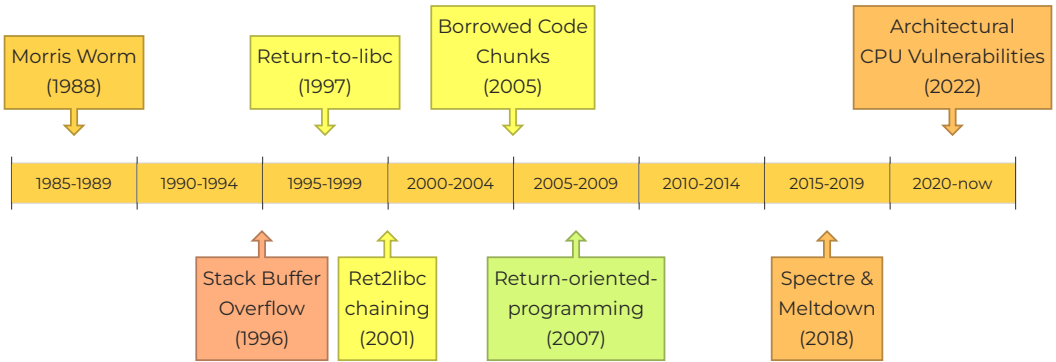
History to Now



History to Now



History to Now



CPUs Have Vulnerabilities Too

Meltdown and Spectre

⇓

Since Meltdown and Spectre, we know:
CPUs can have security vulnerabilities too

What About Numbers?

Intel Core i7-6700T ('15):
⚠ 29 CVEs

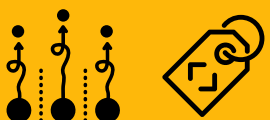
Spectre: PHT, BTB, STL, RSBU, RSBA, SWAPGS, SCSB
Meltdown/MDS: Meltdown, Meltdown 3a, Foreshadow, L1DES, VRS, VRSA, Fall-out, RIDL, ZombieLoad, TAA, Crosstalk, Downfall
LVI: LVI, LVI-NULL, FPVI
Other: ÆPICLeak, PLATYPUS, UMH, MCEA, DRPW, SBDS, Cache policy

Intel Core Ultra 9 285K ('24):
✓ 12 CVEs

Spectre: PHT, BTB, STL, RSBU, RRSBA, SWAPGS, SCSB, BHI, Intra-mode BTI, Post barrier, BPI
LVI: FPVI
Other: ÆPICLeak

Three Stages of CPU Attacks

Side-Channel Attacks



Indirect + Metadata

Three Stages of CPU Attacks

Side-Channel Attacks

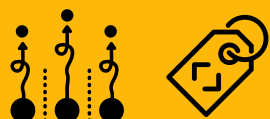
Indirect + Metadata

Transient-Execution Attacks

Indirect + Data

Three Stages of CPU Attacks

Side-Channel Attacks

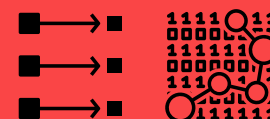


Indirect + Metadata

Transient-Execution Attacks

Indirect + Data

Architectural CPU Vulnerabilities



Direct + Data

Intel SGX

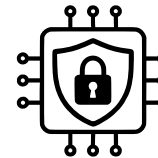
Intel SGX was a key enabler for research

Threat Models Change

Traditional

Attackers are
unprivileged

TEEs



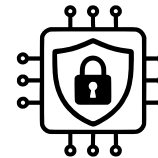
Attackers are
privileged

Threat Models Change

Traditional

Attackers are
unprivileged

TEEs



Attackers are
privileged

New attacks: page faults, performance counter, undervolting, ...

Intel SGX: Fallen into Disgrace

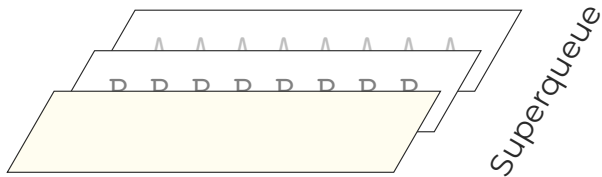
Designed before microarchitectural attacks

Leaking from the Superqueue

APIC

EOI	???
ISR	???
IRR	???

Victim (SGX)

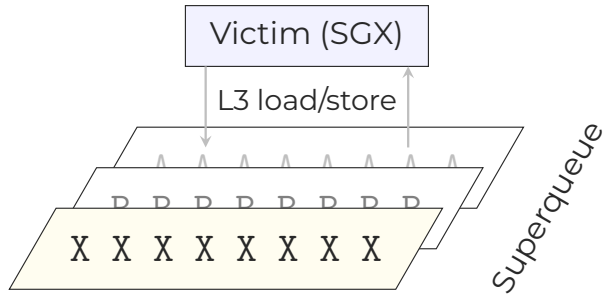


Attacker

Leaking from the Superqueue

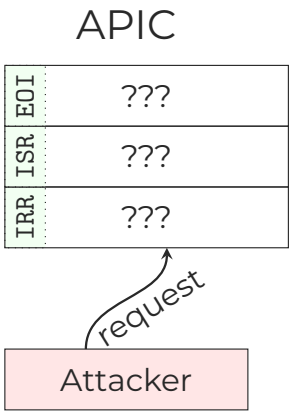
APIC

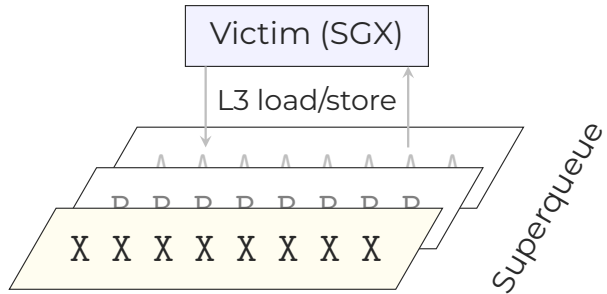
EOI	???
ISR	???
IRR	???



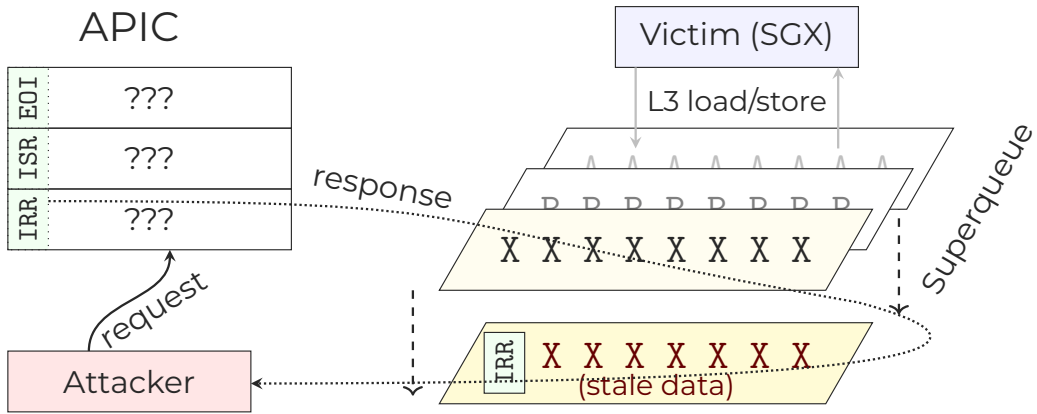
Attacker

Leaking from the Superqueue



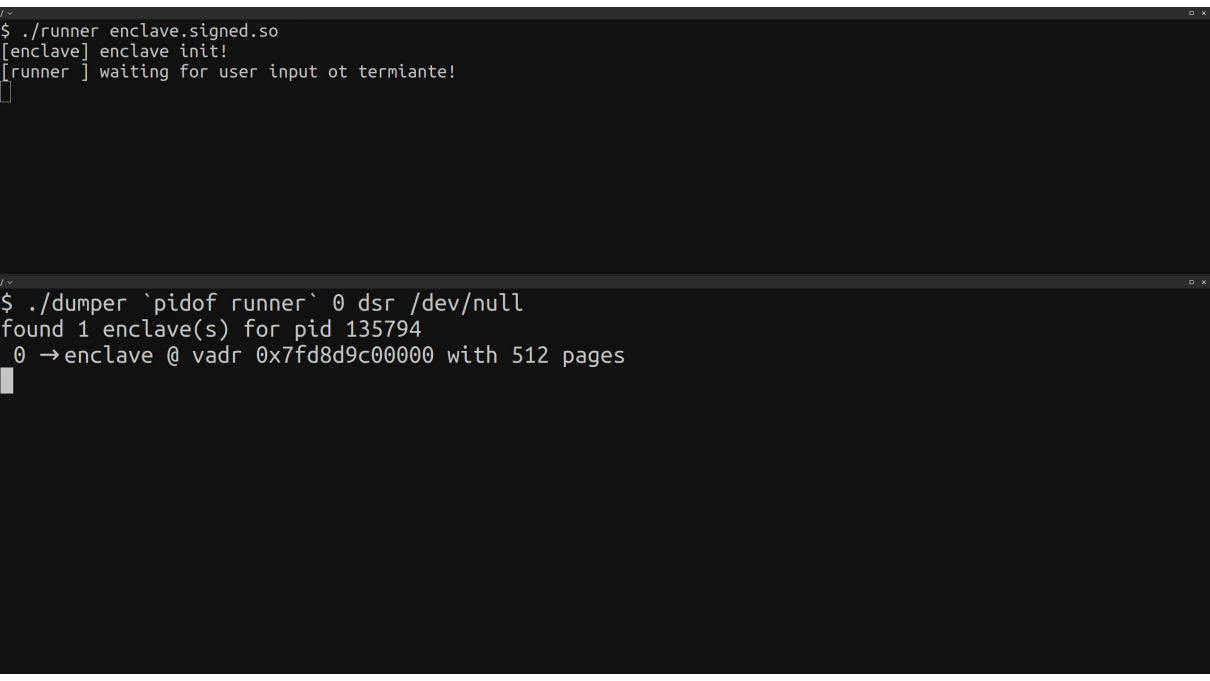


Leaking from the Superqueue



```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!
```

```
$ ./dumper `pidof runner` 0 dsr /dev/null
```


$ ./runner enclave.signed.so
[enclave] enclave init!
[runner] waiting for user input ot termiante!
$ ./dumper `pidof runner` 0 dsr /dev/null
found 1 enclave(s) for pid 135794
0 →enclave @ vadr 0x7fd8d9c00000 with 512 pages

```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!
```

```
22 →....|SPIC|UOUS|.AMO|....|HE.G|REEN|.LEA|....|THAN|.WOU|LD.O|....|WISE|.BE.|THE.|
24 →....|ANY.|LIGH|T.UP|....|HE.R|IDDL|E_.W|....|WOUL|D.YO|U.DO|....|YOU.|WERE|.A.B|
26 →....|____|ED.B|_H_}|....|.AND|E...|_H_E|....|ED.B|Y.DA|Y_.S|....|.THE|SE.C|ATER|
28 →....|__..|_...|___E|....|_U..|@___|_U..|....|.AMO|NG.T|HE.T|....|.HER|BAGE|.AND|
30 →....|R.SU|CH.C|IRCU|....|NCES|.THE|.BRO|....|OLOR|.REA|LLY.|....|MES.|A.PR|OTEC|
32 →....|.BRO|WN_.|CONC|....|D.TH|EMSE|LVES|....|THE.|GROU|ND_.|....|THAT|.WE.|WERE|
34 →....|____|p_.,|__{_|....|$_ J|3...|P*d_|....|_U_i|_<~_|___?|....|_+-4|0_.,|_s__|
36 →....|AT.T|HEY.|ARE.|....|N_.S|OME.|OF.T|....|STIL|L.RE|TAIN|....|THE.|GREE|N.CO|
38 →....|.DAY|.CAM|E.FI|....|.AND|.THA|T.TH|....|OWN.|COLO|R.IS|....|ATER|.ADA|PTAT|
40 →....|____|____|____|....|....|_7g_|B___|....|__..|p_U_|__W_|....|TSTD|C_VE|RSIO|

done[attacker] thread finished!
[main] finished
$
```

```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!
```

```
10 →....|TS.A|__U_|T___|....|EAM.|LOOK|S.AL|....|.LIK|E.A.|FLAS|....|.SUN|'...|T_DE|
12 →....|CE_.|THE.|LION|....|E.AN|TELO|PE_.|....|THE.|WILD|.DON|....|ARE.|ALL.|SAND|
14 →....|.NEI|THER|.TRE|....|BRUS|HWOO|D_.N|....|]7$_|G___|__V_|....|_D__|_X__|j___|
16 →....|OR.A|SSIM|ILAT|....|O.TH|AT.O|F.TH|....|RROU|NDIN|G.CO|....|Y.IS|.ABS|OLUT|
18 →....|BIRD|.__(|D.AL|....|HE.F|UR.O|F.AL|....|E.SM|ALLE|R.MA|....|S.AN|D.TH|E.SK|
20 →....|E.NE|XT.P|OINT|....|THE.|COLO|R.OF|....|.MAT|URE.|CATE|....|LARS|_.SO|ME.O|
22 →....|SPIC|UOUS|.AMO|....|HE.G|REEN|.LEA|....|THAN|.WOU|LD.O|....|WISE|.BE.|THE.|
24 →....|ANY.|LIGH|T.UP|....|HE.R|IDDL|E_.W|....|WOUL|D.YO|U.DO|....|YOU.|WERE|.A.B|
26 →....|____|ED.B|_H_}|....|.AND|E...|_H_E|....|ED.B|Y.DA|Y_.S|....|.THE|SE.C|ATER|
28 →....|__..|_...|___E|....|_U..|@___|_U..|....|.AMO|NG.T|HE.T|....|.HER|BAGE|.AND|
30 →....|R.SU|CH.C|IRCU|....|NCES|.THE|.BRO|....|OLOR|.REA|LLY.|....|MES.|A.PR|OTEC|
32 →....|.BRO|WN_.|CONC|....|D.TH|EMSE|LVES|....|THE.|GROU|ND_.|....|THAT|.WE.|WERE|
34 →....|____|p_.,|__{_|....|$_ J|3...|P*d_|....|_U_i|_<~_|___?|....|_+-4|0_.,|_s__|
36 →....|AT.T|HEY.|ARE.|....|N_.S|OME.|OF.T|....|STIL|L.RE|TAIN|....|THE.|GREE|N.CO|
38 →....|.DAY|.CAM|E.FI|....|.AND|.THA|T.TH|....|OWN.|COLO|R.IS|....|ATER|.ADA|PTAT|
```

```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!
```

```
$ ./dumper `pidof runner` 0 dsr /dev/null
found 1 enclave(s) for pid 135794
 0 →enclave @ vadr 0x7fd8d9c00000 with 512 pages
leaking page   30/ 512
 0 →....|TED.|FROM|.THE|....|ORS.|OF.A|NIMA|....|Y.SI|R.JO|HN.L|....|CK.I|N.A.|BOOK|
 2 →....|LS.I|S.BY|.NO.|....|S.A.|MATT|ER.O|....|ANCE|_.IT|.DEP|....|.ON.|MANY|.CON|
 4 →....|L.FR|__j_|[___|....|[___|_GN_|U___|....|SS.C|ONSP|ICUO|....|____|.___|U___|
 6 →....|BUT.|FEW.|BRIG|....|.COL|ORED|.ANI|....|_.TH|ERE.|ARE_|....|EVER|_.NO|T.A.|
 8 →....|ISHE|R.IT|SELF|....|OUGH|.SO.|BRIG|....|.COL|ORED|_.IS|....|NO.M|EANS|.EAS|
10 →....|TS.A|__U_|T___|....|EAM.|LOOK|S.AL|....|.LIK|E.A.|FLAS|....|.SUN|'...|T_DE|
12 →....|CE_.|THE.|LION|....|E.AN|TELO|PE_.|....|THE.|WILD|.DON|....|ARE.|ALL.|SAND|
14 →....|.NEI|THER|.TRE|....|BRUS|HWOO|D_.N|....|]7$_|G___|__V_|....|_D__|_X__|j___|
16 →....|OR.A|SSIM|ILAT|....|O.TH|AT.O|F.TH|....|RROU|NDIN|G.CO|....|Y.IS|.ABS|OLUT|
18 →....|BIRD|.__(|D.AL|....|HE.F|UR.O|F.AL|....|E.SM|ALLE|R.MA|....|S.AN|D.TH|E.SK|
20 →....|F.NE|XT.P|OINT|....|THE.|COLO|R.OF|....|.MAT|URE.|CATE|....|LARS|_.SO|ME.O|
```

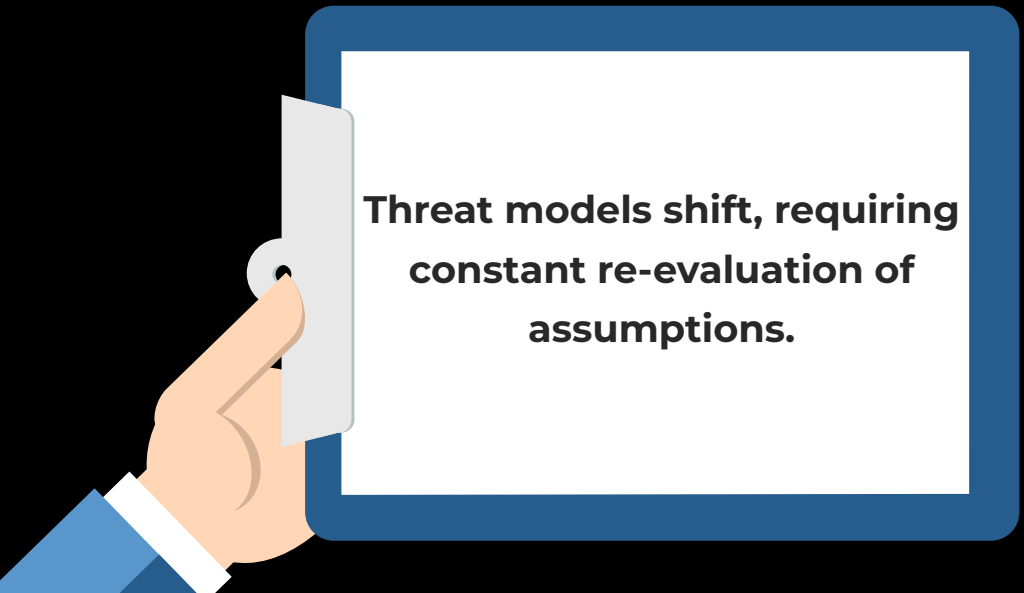
```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!

$ cat enclave.c
unsigned char PALIGN text_lower[] =
    "adapted from the colors of animals by sir john lubbock in a book of natural history (1902, ed. david starr "
    "jordan)the color of animals is by no means a matter of chance; it depends on many considerations, but in the "
    "majority of cases tends to protect the animal from danger by rendering it less conspicuous. perhaps it may be "
    "said that if coloring is mainly protective, there ought to be but few brightly colored animals. there are, "
    "however, not a few cases in which vivid colors are themselves protective. the kingfisher itself, though so "
    "brightly colored, is by no means easy to see. the blue harmonizes with the water, and the bird as it darts along "
    "the stream looks almost like a flash of sunlight. desert animals are generally the color of the desert, thus, for "

$ ./dumper `pidof runner` 0 dsr /dev/null
found 1 enclave(s) for pid 135794
 0 →enclave @ vadr 0x7fd8d9c00000 with 512 pages
leaking page   30/ 512
 0 →....|TED.|FROM|.THE|....|ORS.|OF.A|NIMA|....|Y.SI|R.JO|HN.L|....|CK.I|N.A.|BOOK|
 2 →....|LS.I|S.BY|.NO.|....|S.A.|MATT|ER.O|....|ANCE|_.IT|.DEP|....|.ON.|MANY|.CON|
 4 →....|L.FR|__j_|[___|....|[___|_GN_|U___|....|SS.C|ONSP|ICUO|....|_...|.___|U___|
 6 →....|BUT.|FEW.|BRIG|....|.COL|ORED|.ANI|....|_.TH|ERE.|ARE_|....|EVER|_.NO|T.A.|
 8 →....|ISHE|R.IT|SELF|....|OUGH|.SO.|BRIG|....|.COL|ORED|_.IS|....|NO.M|EANS|.EAS|
10 →....|TS.A|__U_|T___|....|EAM.|LOOK|S.AL|....|.LIK|E.A.|FLAS|....|.SUN|'...|T_DE|
12 →....|CE_.|THE.|LION|....|E.AN|TELO|PE_.|....|THE.|WILD|.DON|....|ARE.|ALL.|SAND|
14 →....|.NEI|THER|.TRE|....|BRUS|HWOO|D_.N|....|]7$_|G___|__V_|....|_D__|_X__|j___|
16 →....|OR.A|SSIM|ILAT|....|O.TH|AT.O|F.TH|....|RROU|NDIN|G.CO|....|Y.IS|.ABS|OLUT|
18 →....|BIRD|.__(|D.AL|....|HE.F|UR.O|F.AL|....|E.SM|ALLE|R.MA|....|S.AN|D.TH|E.SK|
20 →....|F.NE|XT.P|OINT|....|THE.|COLO|R.OF|....|.MAT|URE.|CATE|....|LARS|_.SO|ME.O|
```

```
$ ./runner enclave.signed.so
[enclave] enclave init!
[runner ] waiting for user input ot termiante!

$ cat enclave.c
unsigned char PALIGN text_lower[] =
    "adapted from the colors of animals by sir john lubbock in a book of natural history (1902, ed. david starr "
    "jordan)the color of animals is by no means a matter of chance; it depends on many considerations, but in the "
    "majority of cases tends to protect the animal from danger by rendering it less conspicuous. perhaps it may be "
    "said that if coloring is mainly protective, there ought to be but few brightly colored animals. there are, "
    "however, not a few cases in which vivid colors are themselves protective. the kingfisher itself, though so "
    "brightly colored, is by no means easy to see. the blue harmonizes with the water, and the bird as it darts along "
    "the stream looks almost like a flash of sunlight. desert animals are generally the color of the desert, thus, for "

$ ./dumper `pidof runner` 0 dsr /dev/null
found 1 enclave(s) for pid 135794
 0 →enclave @ vadr 0x7fd8d9c00000 with 512 pages
leaking page   30/ 512
 0 →....|TED.|FROM|.THE|....|ORS.|OF.A|NIMA|....|Y.SI|R.JO|HN.L|....|CK.I|N.A.|BOOK|
 2 →....|LS.I|S.BY|.NO.|....|S.A.|MATT|ER.O|....|ANCE|_.IT|.DEP|....|.ON.|MANY|.CON|
 4 →....|L.FR|__j_|[___|....|[___|_GN_|U___|....|SS.C|ONSP|ICUO|....|_...|.___|U___|
 6 →....|BUT.|FEW.|BRIG|....|.COL|ORED|.ANI|....|_.TH|ERE.|ARE_|....|EVER|_.NO|T.A.|
 8 →....|ISHE|R.IT|SELF|....|OUGH|.SO.|BRIG|....|.COL|ORED|_.IS|....|NO.M|EANS|.EAS|
10 →....|TS.A|__U_|T___|....|EAM.|LOOK|S.AL|....|.LIK|E.A.|FLAS|....|.SUN|'...|T_DE|
12 →....|CE_.|THE.|LION|....|E.AN|TELO|PE_.|....|THE.|WILD|.DON|....|ARE.|ALL.|SAND|
14 →....|.NEI|THER|.TRE|....|BRUS|HWOO|D_.N|....|]7$_|G___|__V_|....|_D__|_X__|j___|
16 →....|OR.A|SSIM|ILAT|....|O.TH|AT.O|F.TH|....|RROU|NDIN|G.CO|....|Y.IS|.ABS|OLUT|
18 →....|BIRD|.__(|D.AL|....|HE.F|UR.O|F.AL|....|E.SM|ALLE|R.MA|....|S.AN|D.TH|E.SK|
20 →....|F.NE|XT.P|OINT|....|THE.|COLO|R.OF|....|.MAT|URE.|CATE|....|LARS|_.SO|ME.O|
```

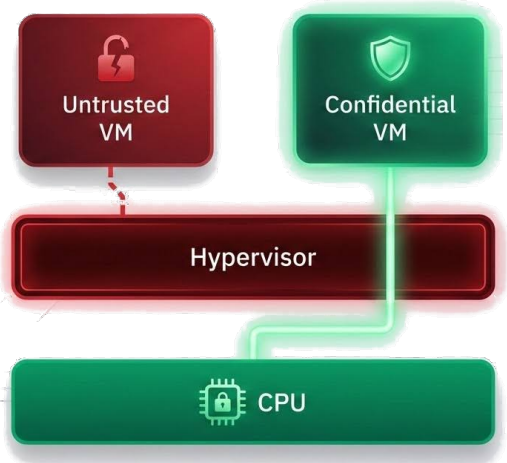
One Vendor was Confident

AMD (2019)

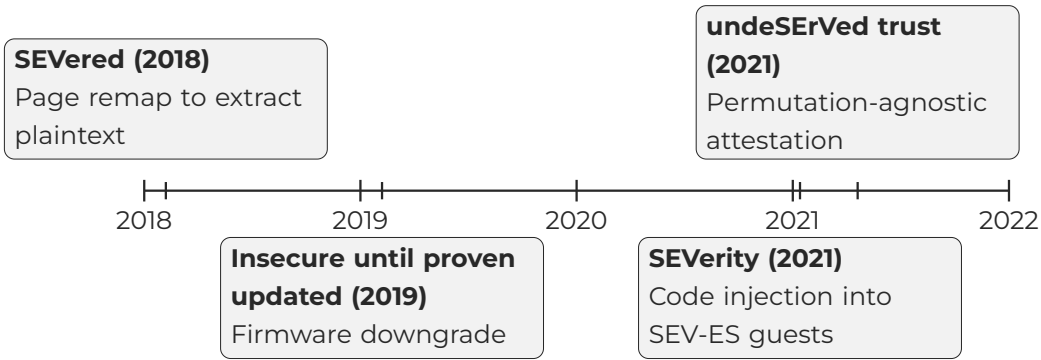
AMD believes that our hardware paging architecture and protection checks help AMD processors not be affected by many side-channel vulnerabilities, regardless of whether Simultaneous Multi-Threading (SMT) is enabled or disabled.


Threat models shift, requiring constant re-evaluation of assumptions.

Confidential VMs: New Generation TEEs



Breaking Confidentiality via Integrity



Integrity is Essential

AMD (AMD-SB-3005)

SEV and SEV-ES features are not designed to protect guest VM memory integrity.

Integrity seen as “bonus”

→ **No confidentiality without integrity**

The Evolution of SEV

SEV

SEV-ES

The Evolution of SEV

SEV

SEV-ES

SEV-SNP

AMD SEV-SNP: State-of-the-art Version

AMD SEV-SNP

SEV-SNP is designed to prevent software-based integrity attacks and reduce risk associated with compromised memory integrity. **The basic principle of SEV-SNP integrity is that if a VM is able to read a private (encrypted) page of memory, it must always read the value it last wrote.**

AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More

Legacy Instructions

INVD Instruction

- Flushes internal CPU caches

INVD Instruction

- Flushes internal CPU caches
- Does not write modifications to memory

INVD Instruction

- Flushes internal CPU caches
- Does not write modifications to memory

Intel

Use this instruction with care. Data cached internally and not written back to main memory will be lost. [...] software should instead use the WBINVD instruction.

CacheWarp: Basic Primitive

Hypervisor

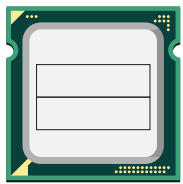
```
invd
```

Victim VM

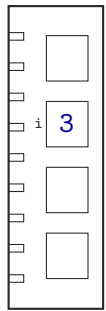
```
printf("%d", i);
i = 4;
printf("%d", i);
```

Output

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

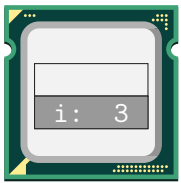
```
invd
```

Victim VM

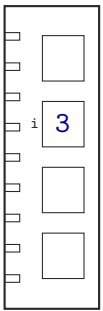
```
printf("%d", i);
i = 4;
printf("%d", i);
```

Output

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

invd

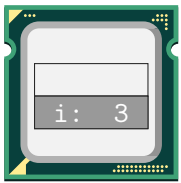
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

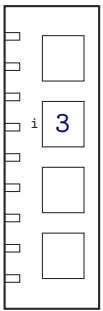
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

```
invd
```

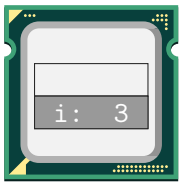
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

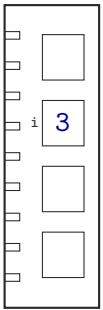
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

```
invd
```

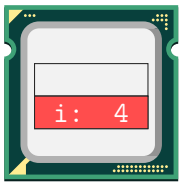
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

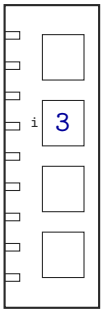
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

- `invd`

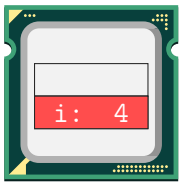
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

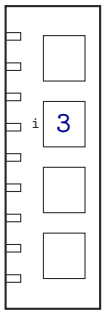
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

- `invd`

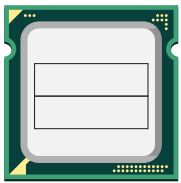
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

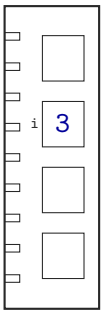
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

```
invd
```

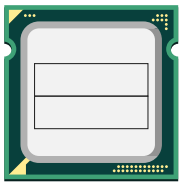
Victim VM

```
printf("%d", i);
i = 4;
▶ printf("%d", i);
```

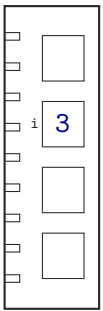
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

```
invd
```

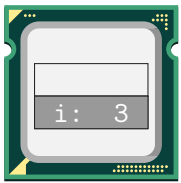
Victim VM

```
printf("%d", i);
i = 4;
▶ printf("%d", i);
```

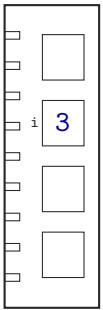
Output

```
3
```

CPU Cache



DRAM



CacheWarp: Basic Primitive

Hypervisor

```
invd
```

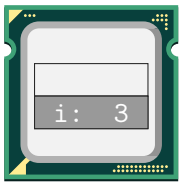
Victim VM

```
printf("%d", i);
i = 4;
printf("%d", i);
```

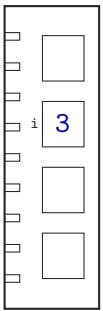
Output

```
3
3
```

CPU Cache



DRAM



sudo Exploit

```
int uid = 0;

uid = getuid();
if( uid == 0 ) {
   // continue as root
} else {
   // ask for password
}
```

sudo Exploit

```
int uid = 0;

uid = getuid(); <-- INVD
if( uid == 0 ) {
   // continue as root
} else {
   // ask for password
}
```

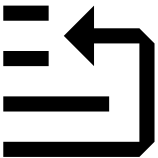
sudo Exploit

```
int uid = 0;


if( uid == 0 ) {
   // continue as root
} else {
   // ask for password
}
```

Timewarp: Attacking Return Addresses

Explicit Writes

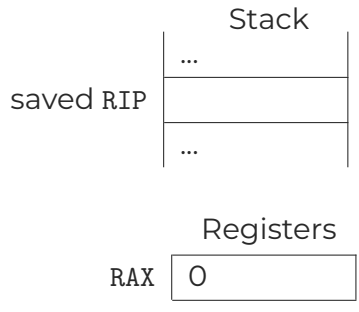
Compiler-induced Writes



Return Addresses

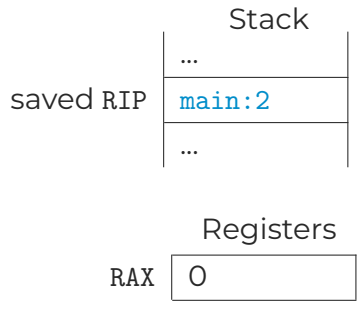
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



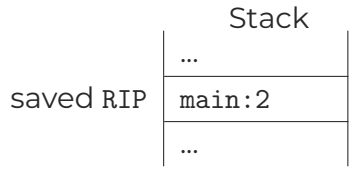
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



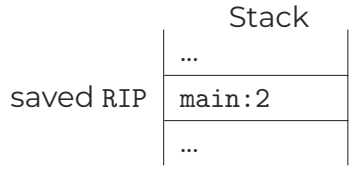
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



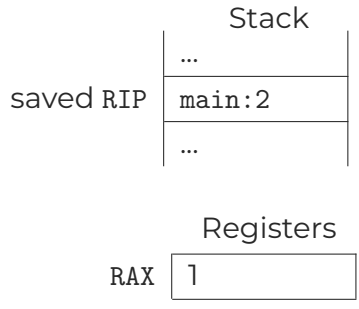
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



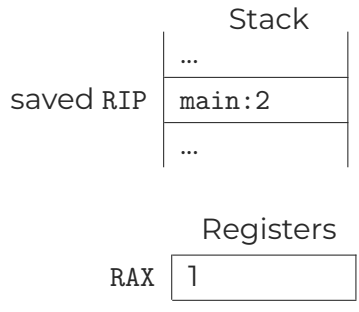
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```

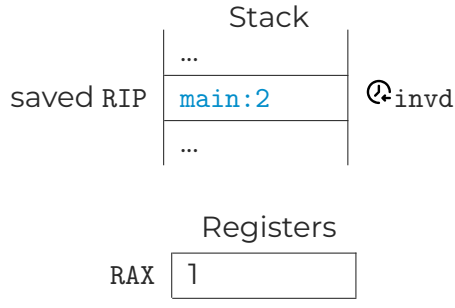


Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```

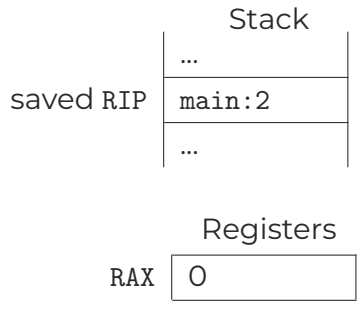
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



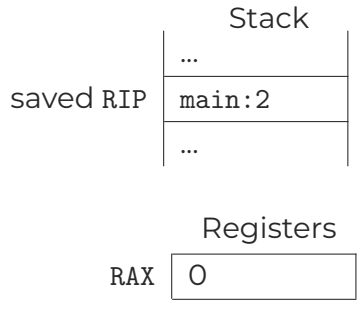
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```



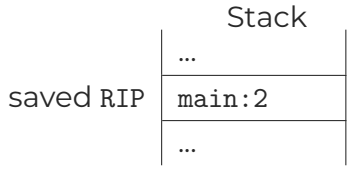
Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```

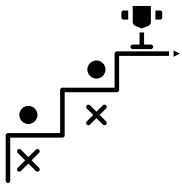


Timewarp

```
1  call    ret1
2  cmp     rax, 0
3  je      win
4  call    ret0
5  jmp     fail
6
7  ret0:
8    mov   rax, 0
9    ret
10
11 ret1:
12   mov   rax, 1
13   ret
```

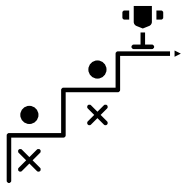


Synchronize + Exploit



Single step to store

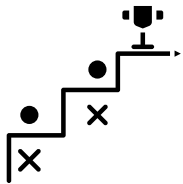
Synchronize + Exploit



Single step to store

Flush entire cache

Synchronize + Exploit

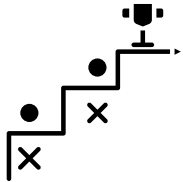


Single step to store

Flush entire cache

Step over store

Synchronize + Exploit

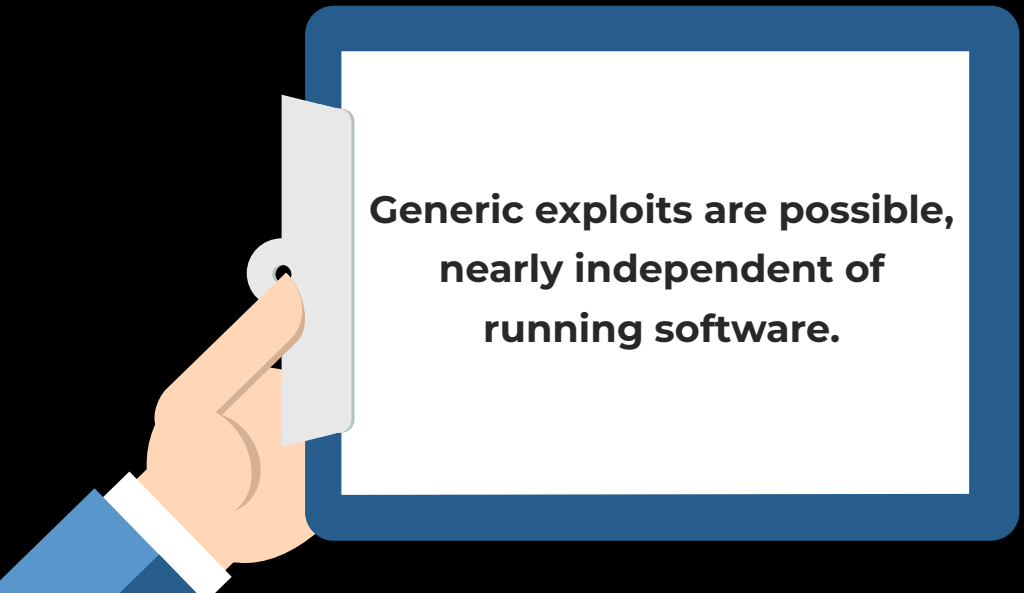


Single step to store

Flush entire cache

Step over store

`INVD` (=throw away) store


Generic exploits are possible,
nearly independent of
running software.

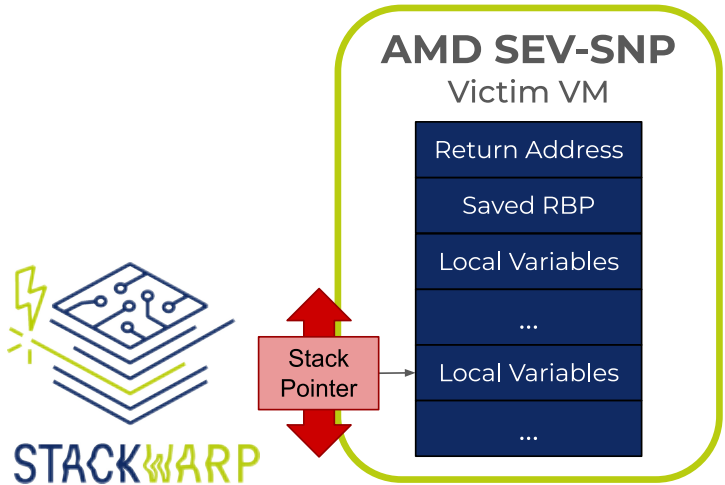
AMD: Not Great, Not Terrible

AMD Security Bulletin

AMD has provided a hot-loadable microcode patch and updated the firmware image [...] This issue has not been found to impact AMD 4th generation “Genoa” EPYC™ processors (“Zen 4” microarchitecture).

AMD INVD Instruction Security Vulnerability (AMD-SB-3005)

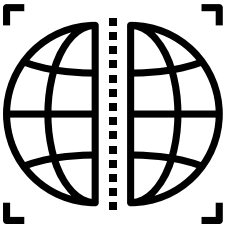
StackWarp - TL;DR



Our Goal

VM keeps running

Architectural state corrupted



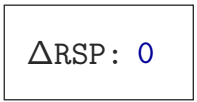
Triggerable from hyperthread

The Culprit: The Stack Engine

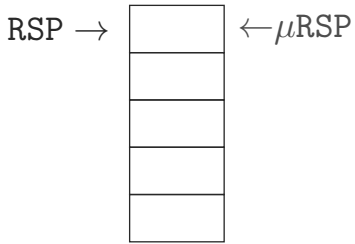
Program

```
push 1
push 2
push 3
mov %rax, %rsp
```

Stack Engine



Victim Stack

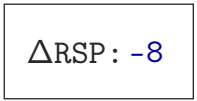


The Culprit: The Stack Engine

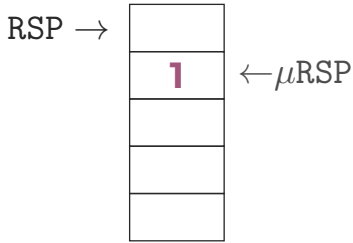
Program

```
push 1
push 2
push 3
mov %rax, %rsp
```

Stack Engine



Victim Stack



The Culprit: The Stack Engine

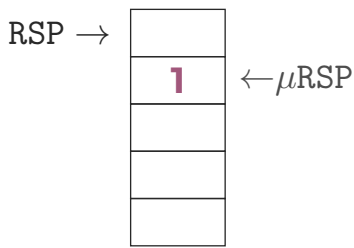
Program

```
  push 1
▶ push 2
  push 3
  mov %rax, %rsp
```

Stack Engine

ΔRSP: -8

Victim Stack

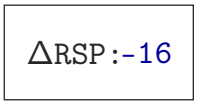


The Culprit: The Stack Engine

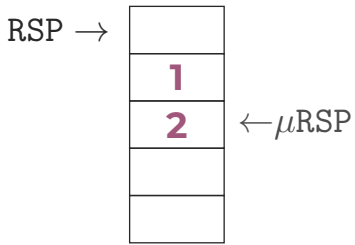
Program

```
push 1
▶ push 2
push 3
mov %rax, %rsp
```

Stack Engine



Victim Stack

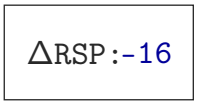


The Culprit: The Stack Engine

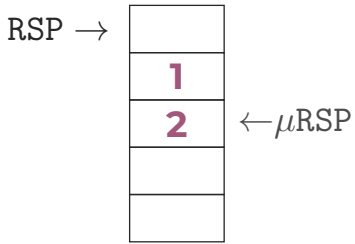
Program

```
push 1
push 2
▶ push 3
mov %rax, %rsp
```

Stack Engine



Victim Stack

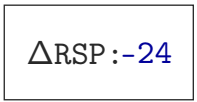


The Culprit: The Stack Engine

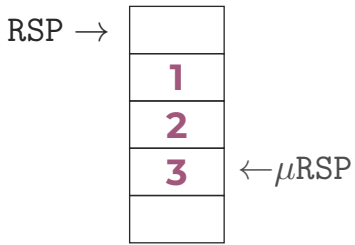
Program

```
push 1
push 2
▶ push 3
mov %rax, %rsp
```

Stack Engine



Victim Stack

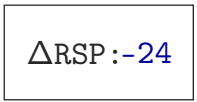


The Culprit: The Stack Engine

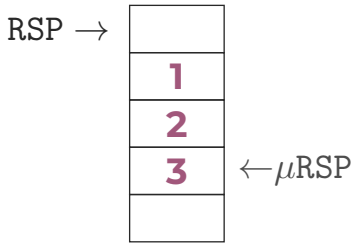
Program

```
push 1
push 2
push 3
▶ mov %rax, %rsp
```

Stack Engine



Victim Stack



The Culprit: The Stack Engine

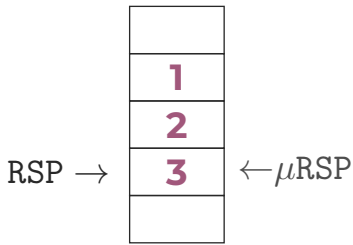
Program

```
push 1
push 2
push 3
▶ mov %rax, %rsp
```

Stack Engine

ΔRSP: 0

Victim Stack



StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
▶ `pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 0	RSP →
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
▶ `pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 8	RSP →
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
▶ [*SE off*]	`push 1`	ΔRSP: 8	RSP →
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
▶ [*SE off*]	`push 1`	ΔRSP: 8	RSP →
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	▶ `push 1`	ΔRSP: 8	RSP →
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	▶ `push 1`	ΔRSP: 8	RSP → 1
[*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 8	RSP → **1**
▶ [*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 8	RSP → 1
▶ [*SE on*]	`push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 0	RSP →
[*SE on*]	▶ `push 2`		

StackWarp: Basic Primitive

Attacker VM	Victim VM	Stack Engine	Victim Stack
`pop %rax`			
[*SE off*]	`push 1`	ΔRSP: 0	RSP → **2**
[*SE on*]	▶ `push 2`		

Attack Targets

Fault-attacks on Crypto

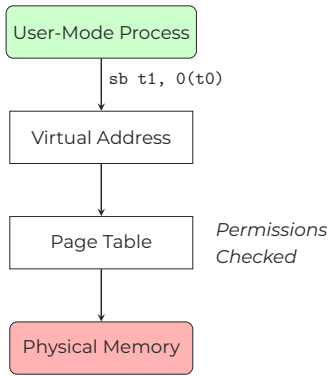
Privilege Escalation

Kernel Code Execution

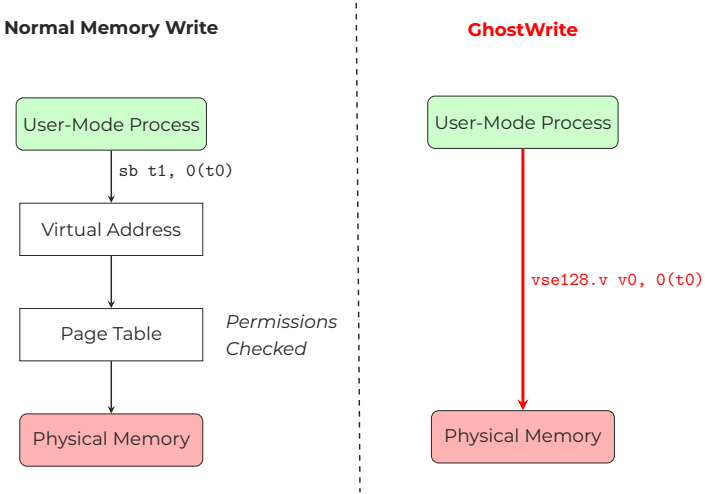
GhostWrite

GhostWrite: Circumventing Virtual Memory

Normal Memory Write



GhostWrite: Circumventing Virtual Memory



GhostWrite

An architectural bug in a RISC-V vector implementation allowed writes to **physical** memory.

Result: Write-anything-anywhere primitive

Architectural bugs are not an x86-only problem.


People find ways to reliably exploit even the smallest corruptions.

Pandora’s Box

Transient execution opened Pandora’s box

Pandora’s Box

Security vulnerabilities in CPUs are real

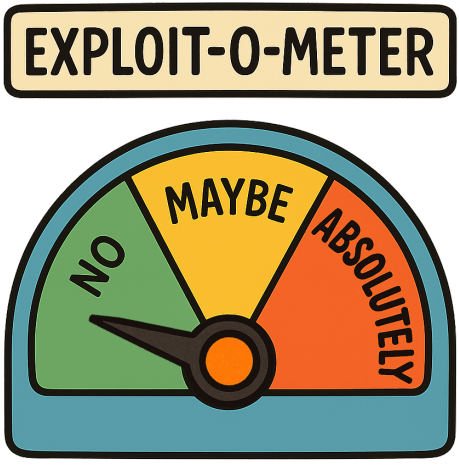
Pandora’s Box

Side channels often used to see them

Pandora’s Box

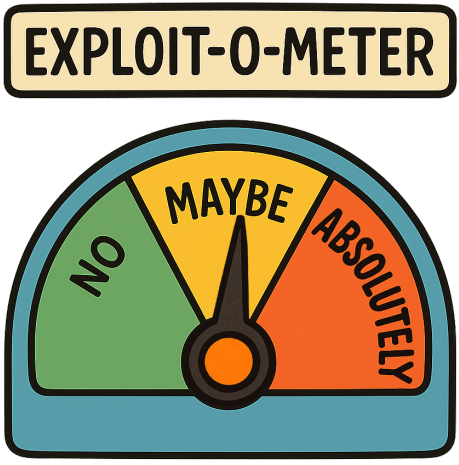
Architectural bugs keep showing up

Expectations for CPU Vulnerabilities



Exploits now?

Expectations for CPU Vulnerabilities



Exploits in a few years?

Expectations for CPU Vulnerabilities

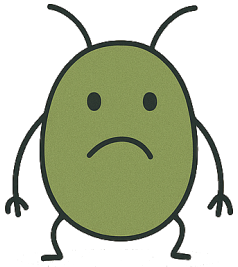
Exploits if somone builds automated tooling?

CPU Security Testing is in its Infancy

Technique	Software	CPU
Manual	✓	✓
Random Fuzzing	✓	✓
Guided Fuzzing	✓	❍
Static Analysis	✓	❍
Symbolic Execution	✓	❍
Sanitizer	✓	✗

Software Bugs vs CPU Bugs

Software Bug



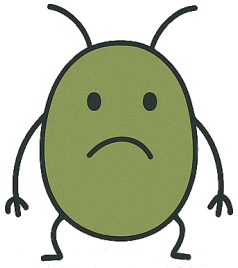
Often just crashes

Directly seen

Per program

Software Bugs vs CPU Bugs

Software Bug

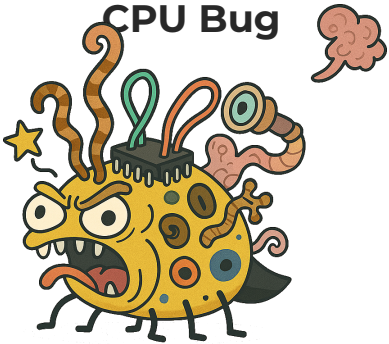


Often just crashes

Directly seen

Per program

CPU Bug



Refuses to crash

Indirectly seen

Cross-domain interaction

Software Testing vs CPU Testing

Feature	Software	CPU
Input	Data	Code
Can run in a debugger	✓	❍
Has feedback	✓	❍
Emulation	fast	slow
Crashes	✓	rarely
Execution	linear	parallel
Has laws of physics	✗	✓

Manual vs. Automated Testing

Manual

2

CacheWarp, EntrySign

Automated

5

ÆPICLeak, ZenBleed, Reptar, StackWarp, GhostWrite

Takeaways

CPUs must be treated like software

just faster, harder to debug, and deployed at massive scale

CPU bugs are exploitable

across VMs, enclaves, and privilege boundaries

Testing still in its infancy

needs observability, metrics, strategies

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Your CPU Is the New Software

Exploiting Architectural Bugs at Hardware Speed

Michael Schwarz | THCon | May 2026