

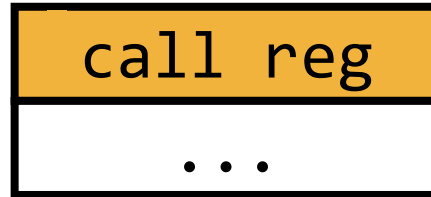
TONTOU

On the Exploitability of Time-of-Neutralization to Time-of-Use Windows

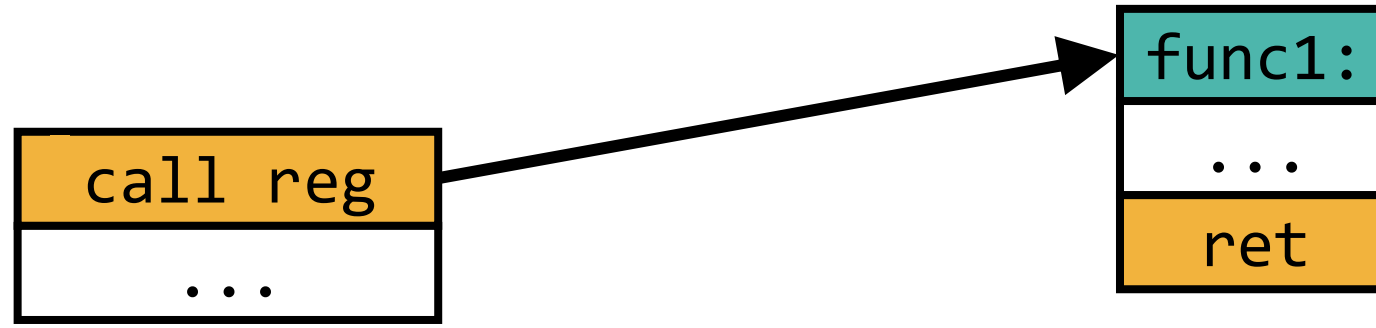
Daniël Trujillo, Mengjia Yan



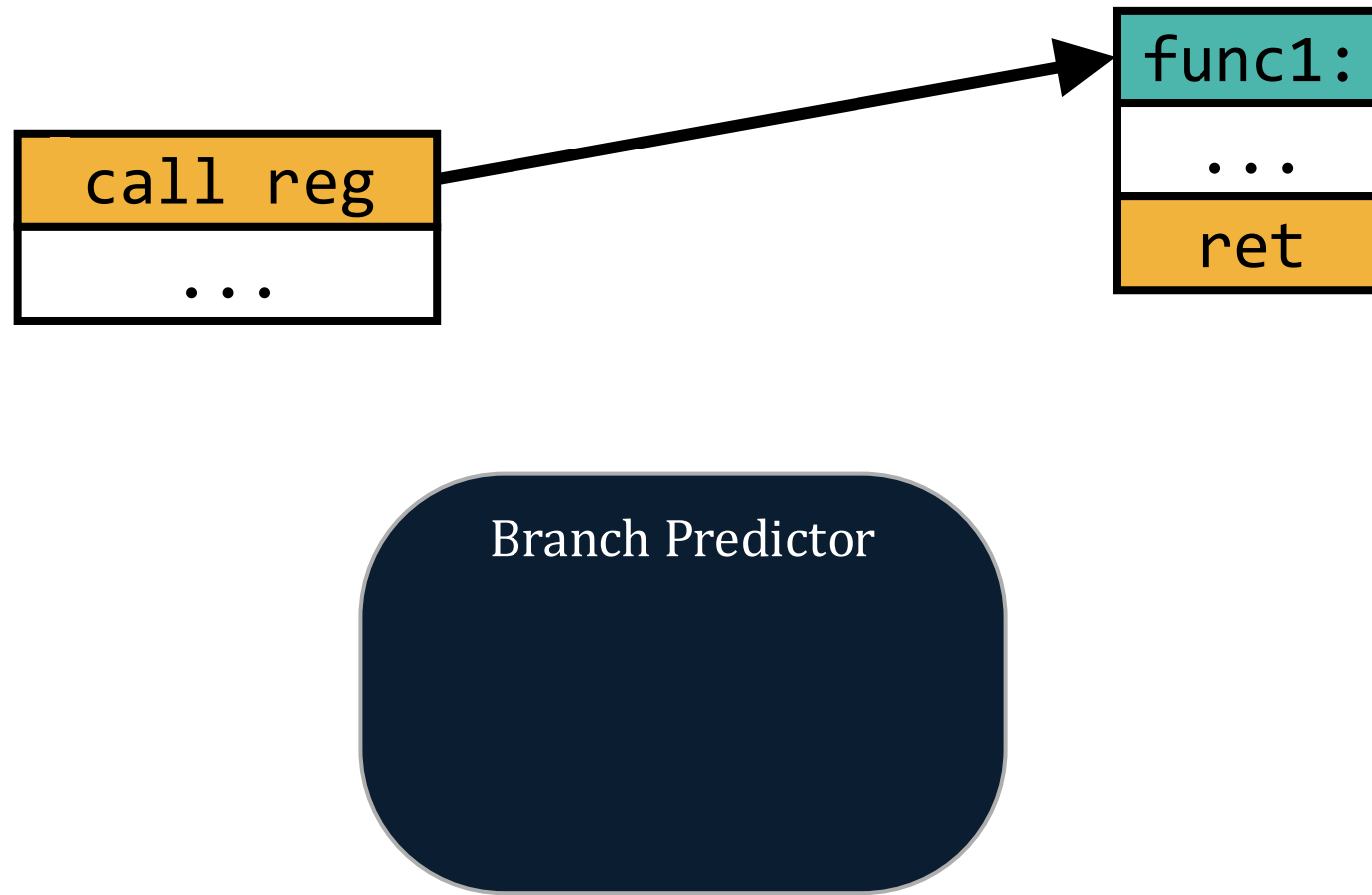
Speculative Execution



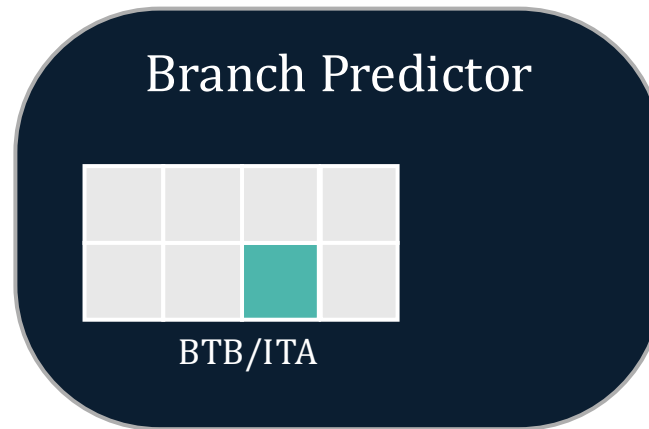
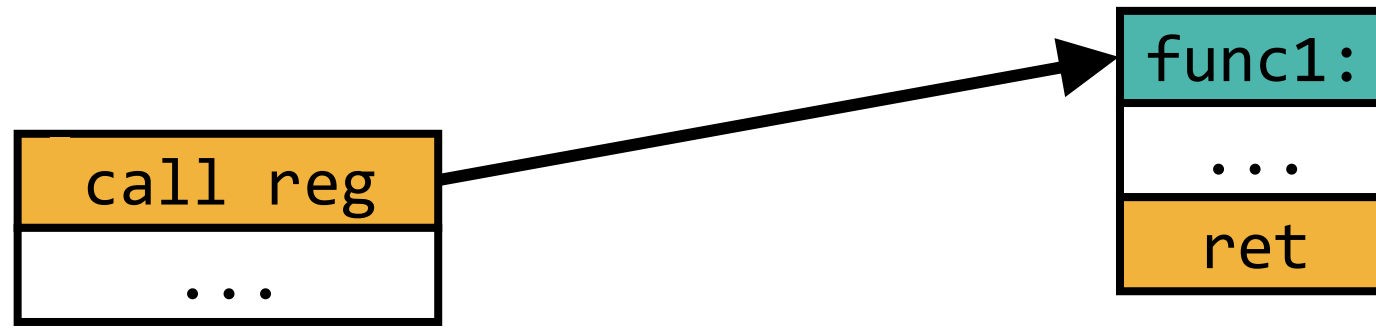
Speculative Execution



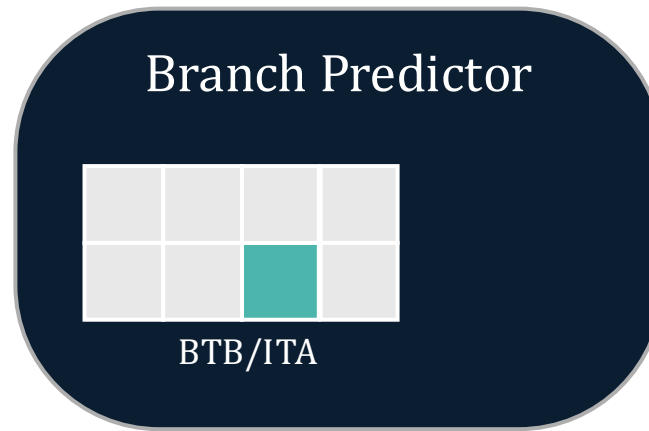
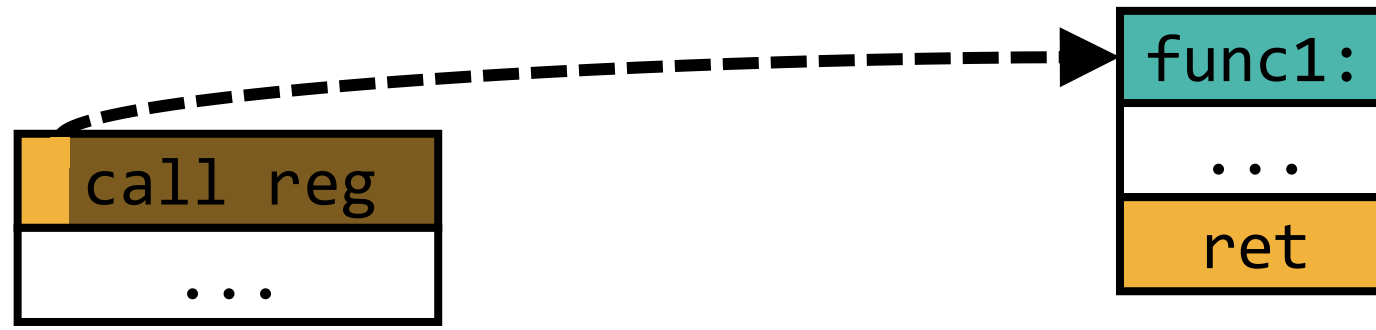
Speculative Execution



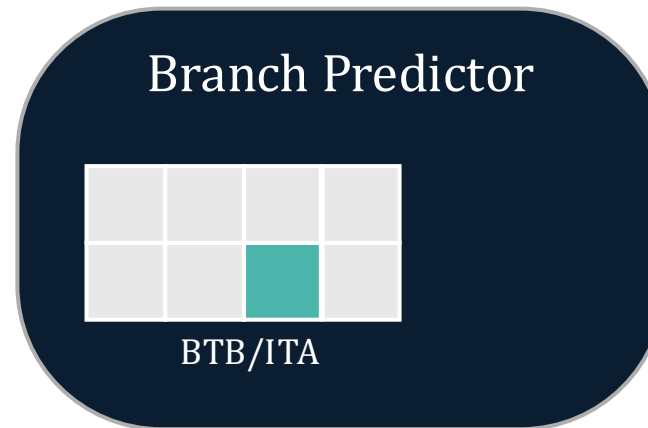
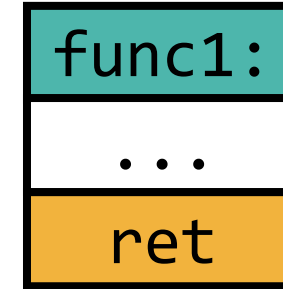
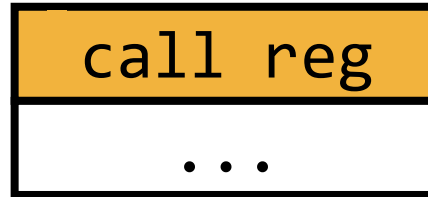
Speculative Execution



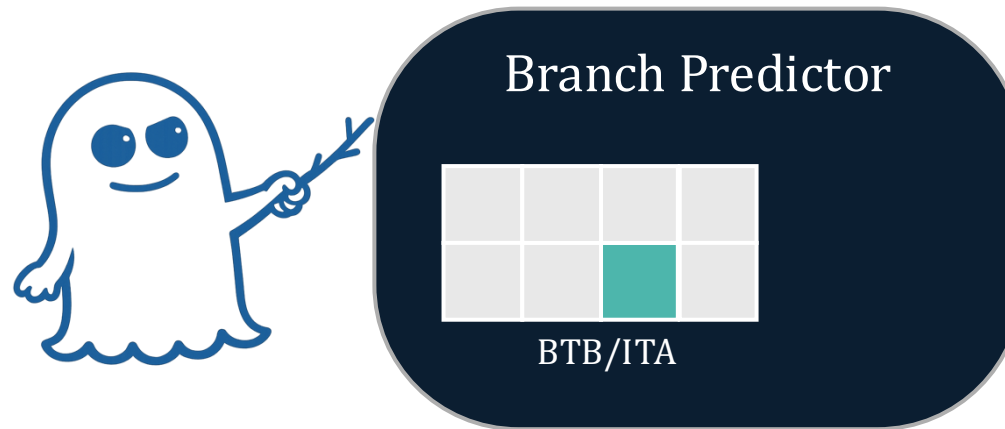
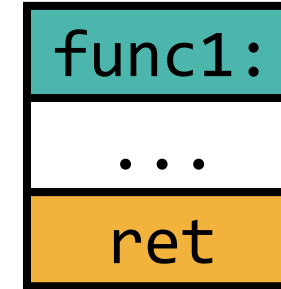
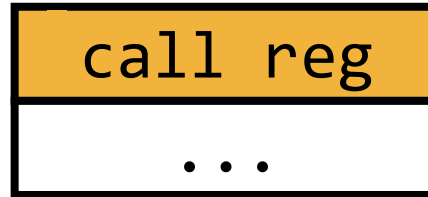
Speculative Execution



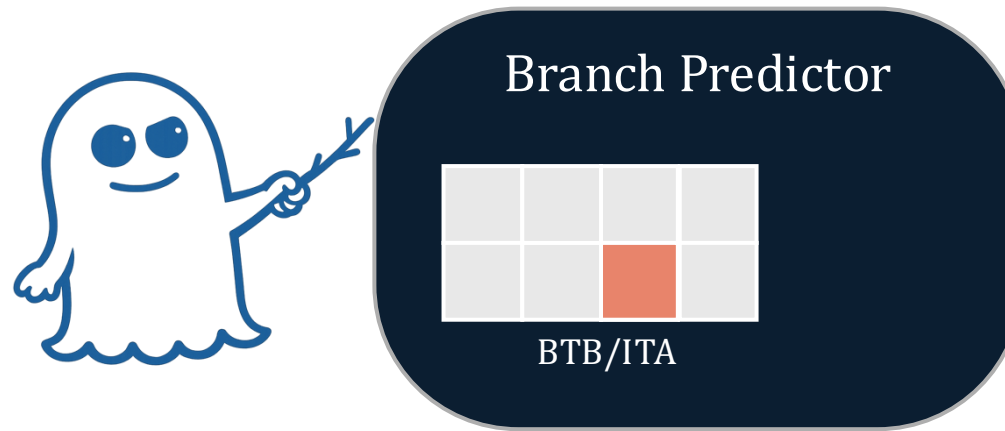
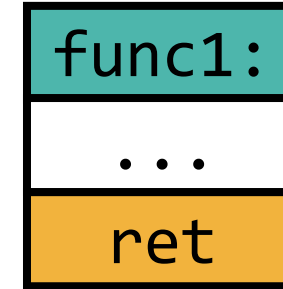
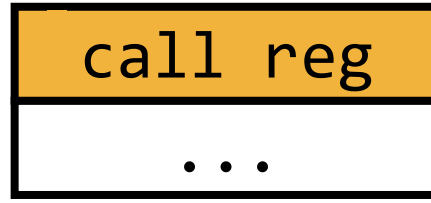
Spectre v2



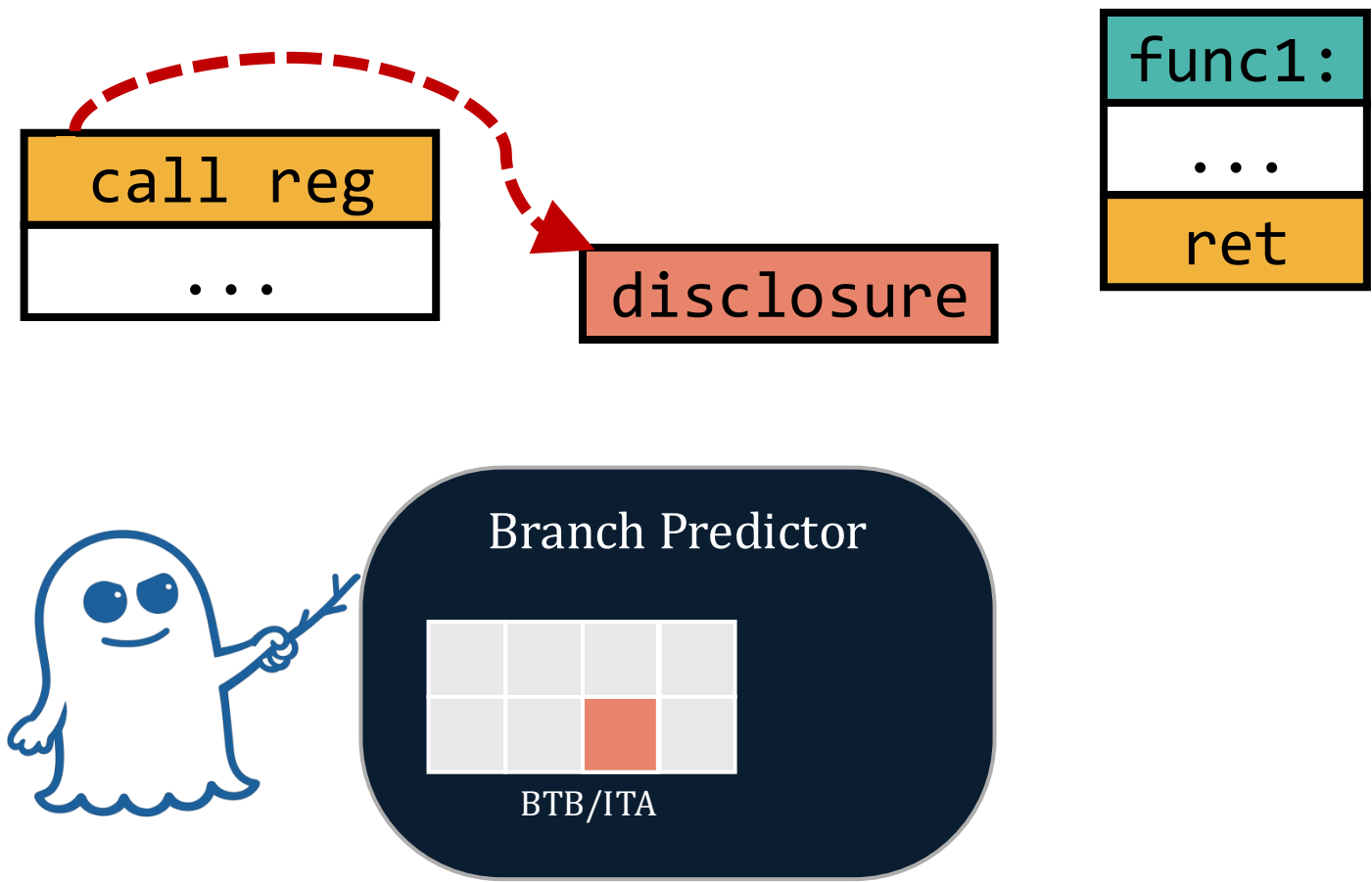
Spectre v2



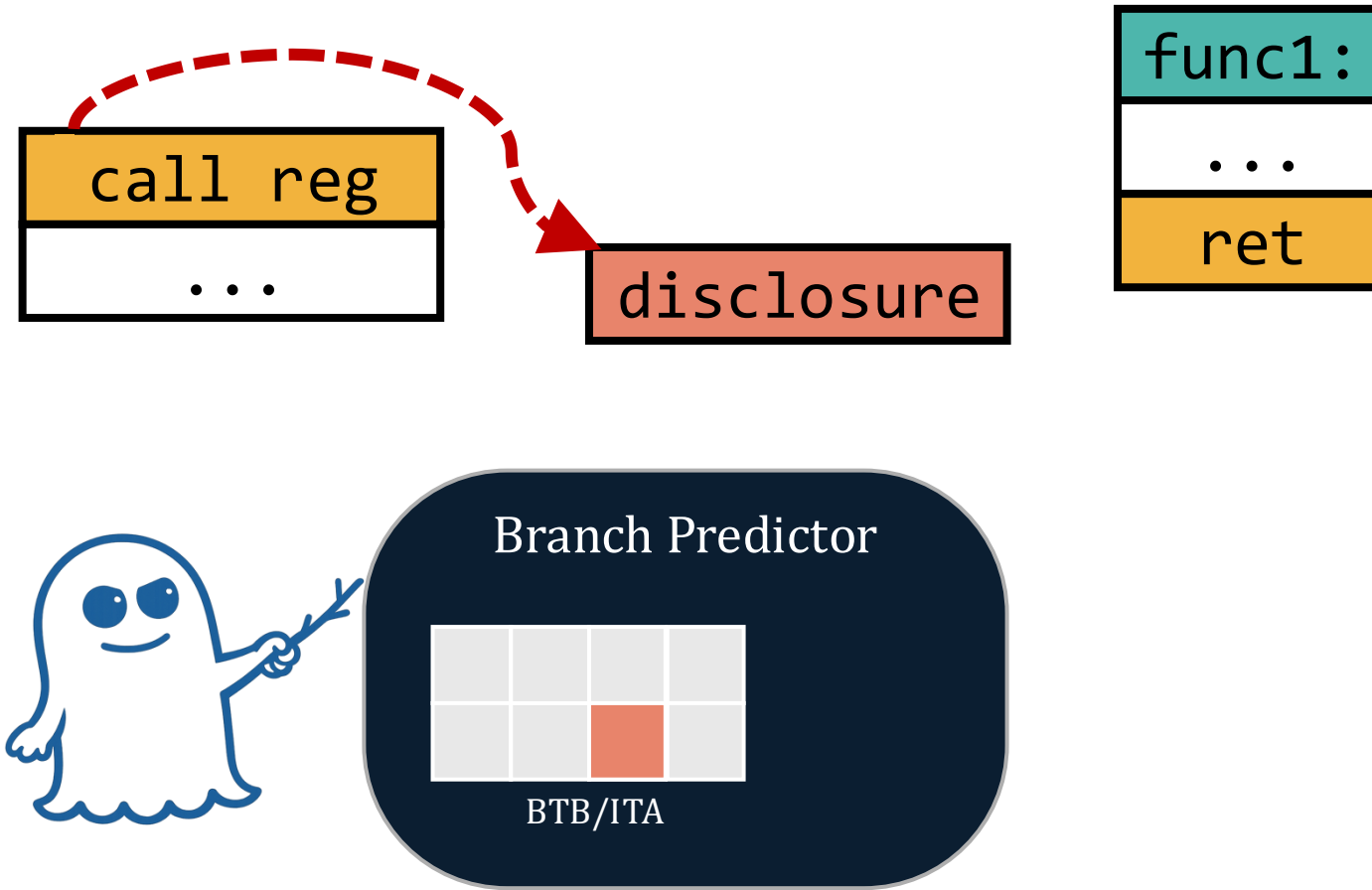
Spectre v2



Spectre v2



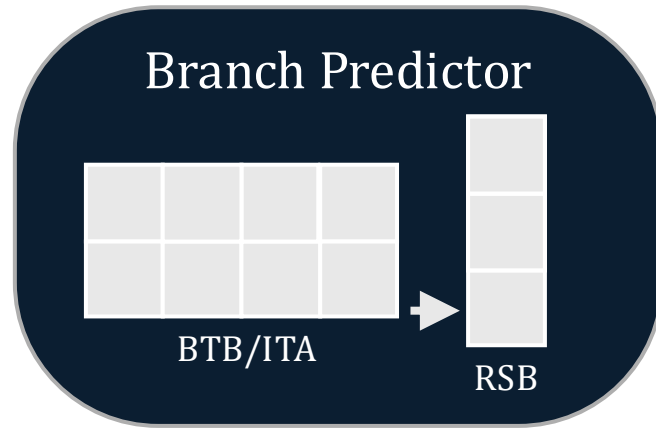
Spectre v2



Spectre v2 attacks leak through branch mispredictions

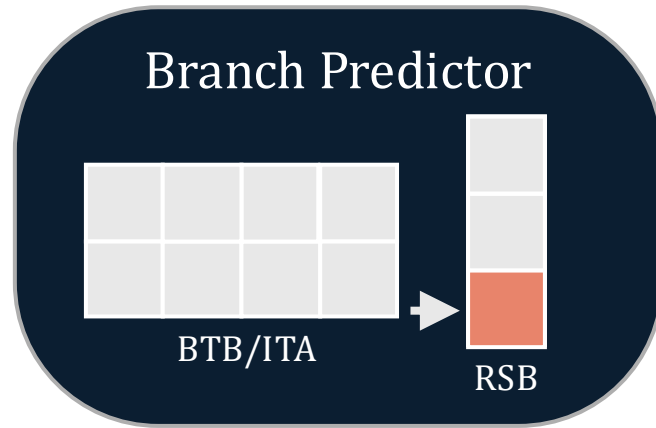
Return mispredictions

Return mispredictions



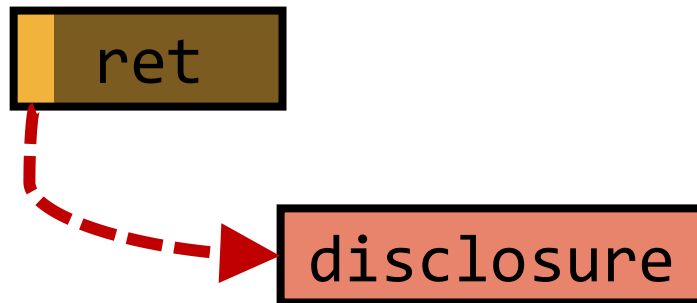
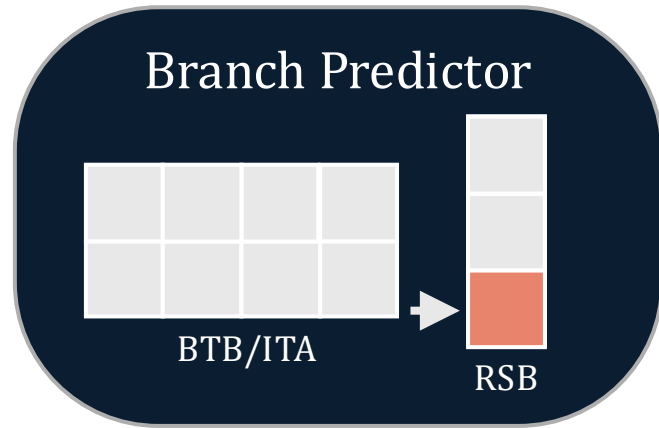
ret

Return mispredictions

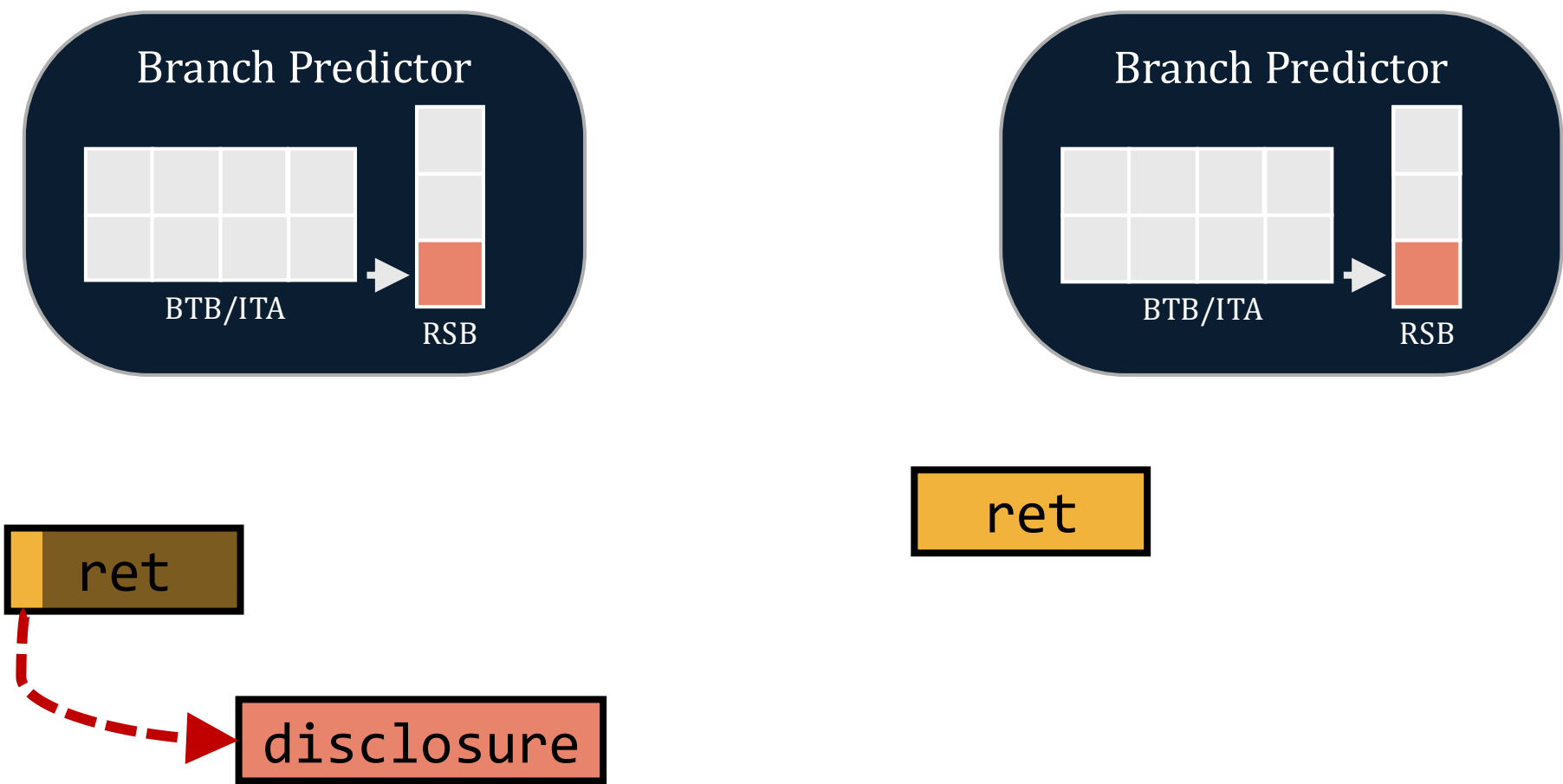


ret

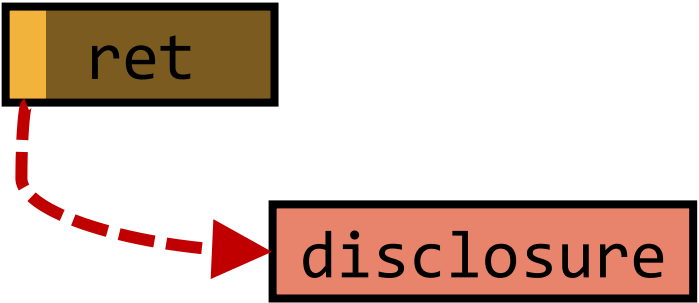
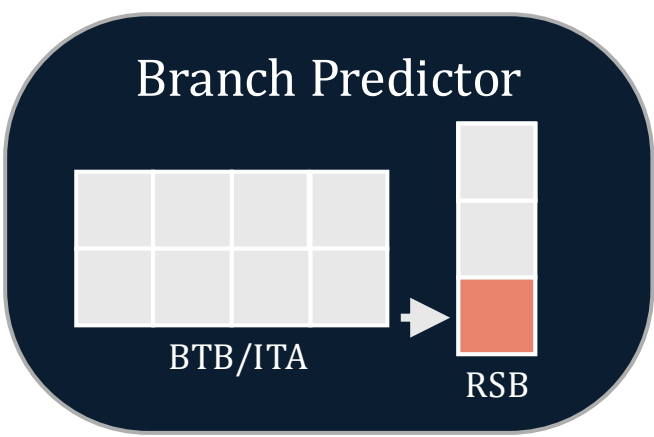
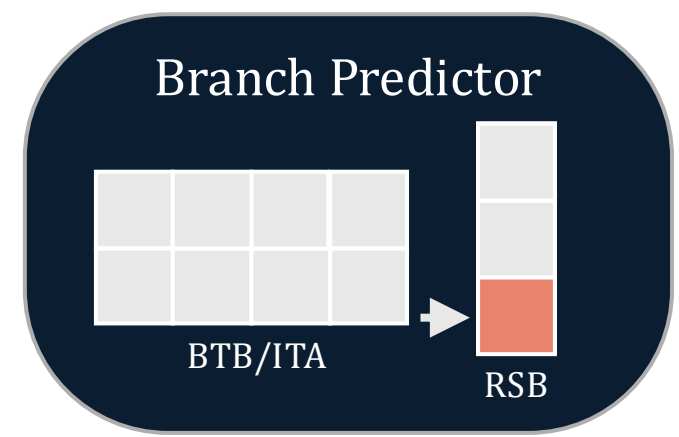
Return mispredictions



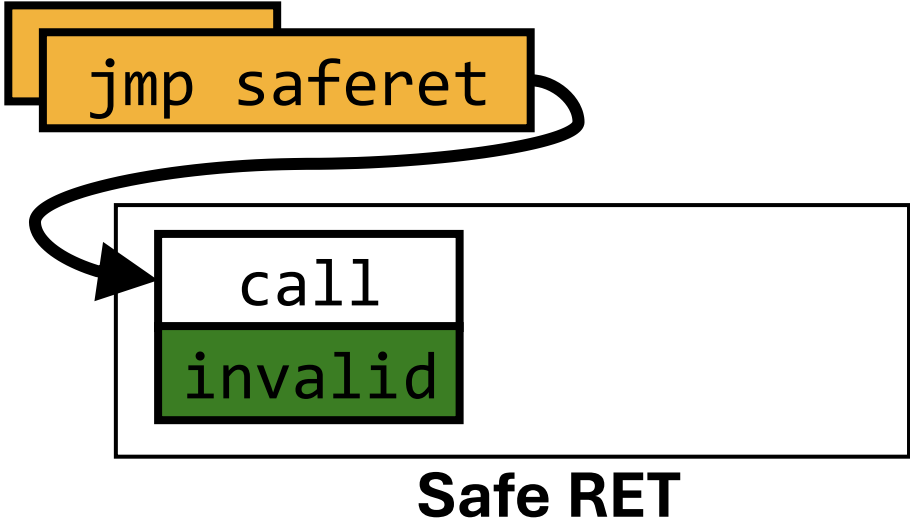
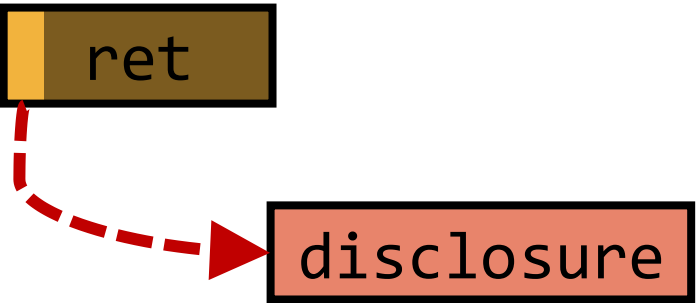
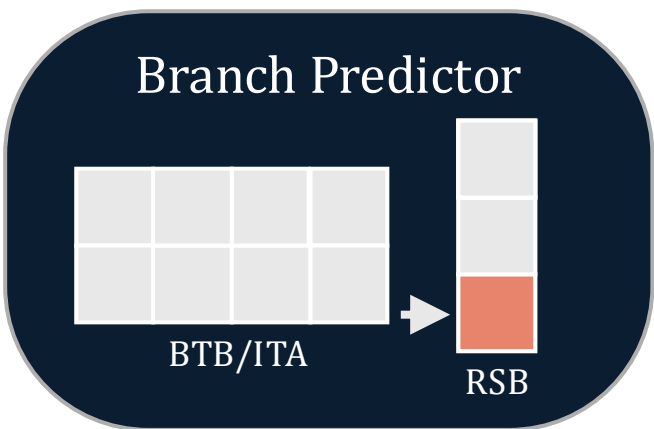
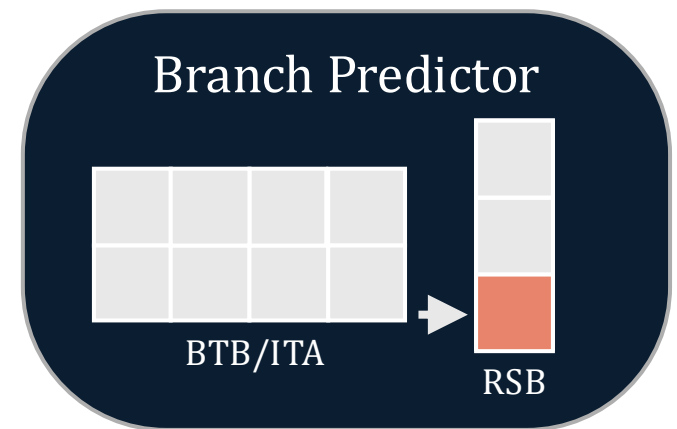
Return mispredictions



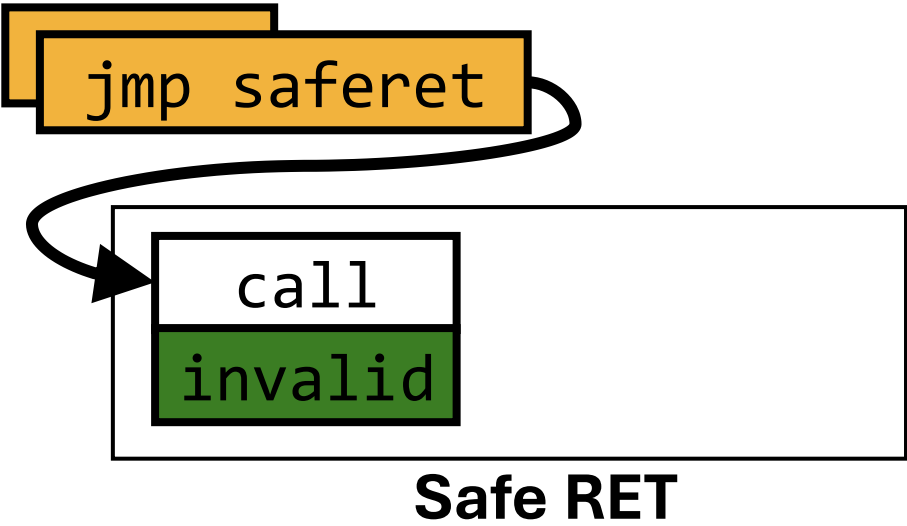
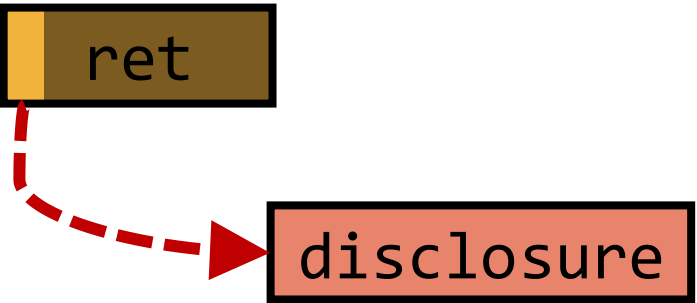
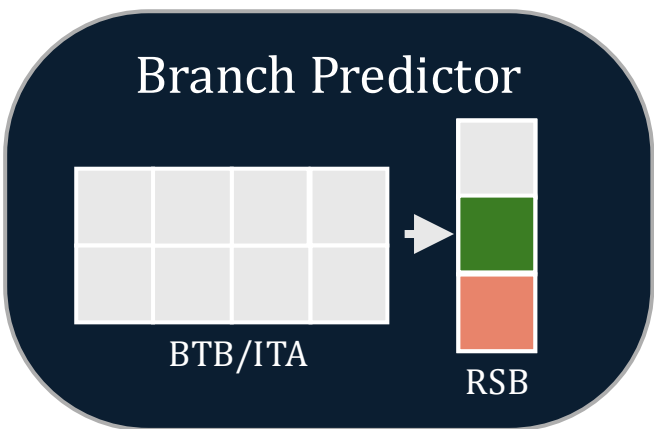
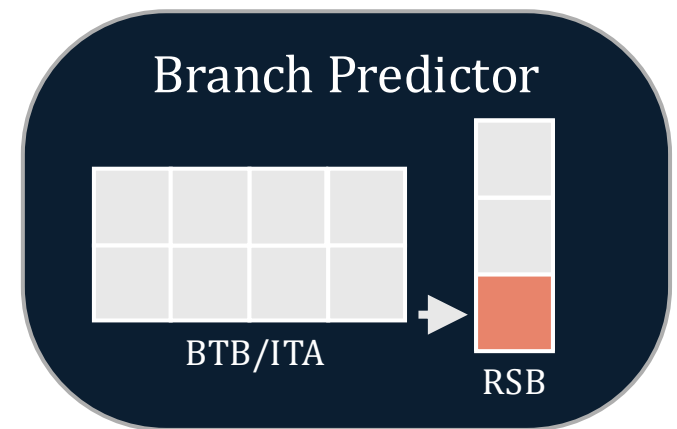
Return mispredictions



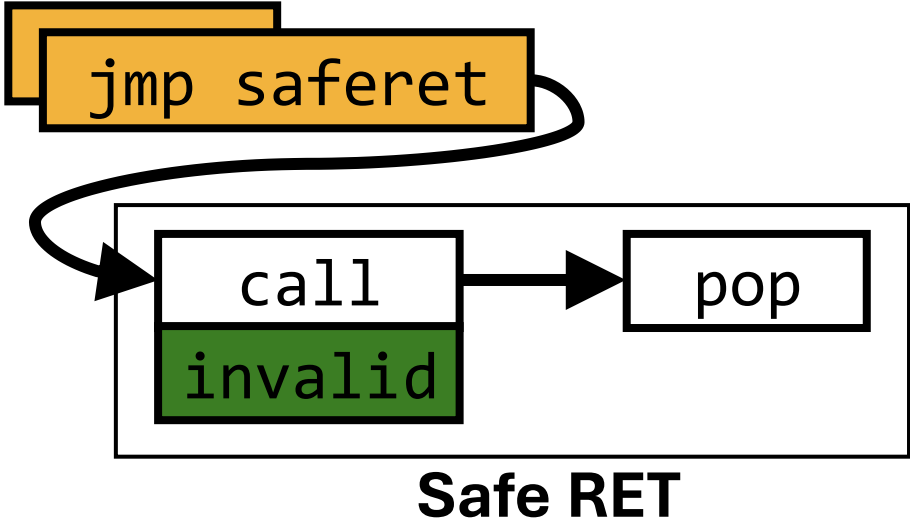
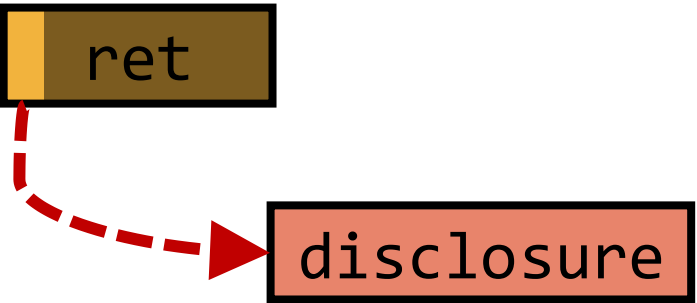
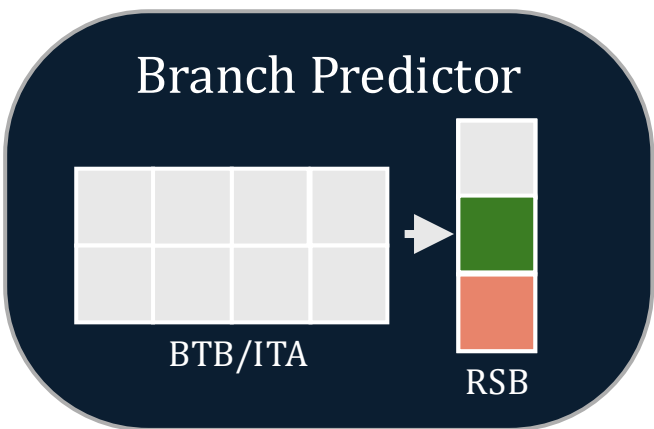
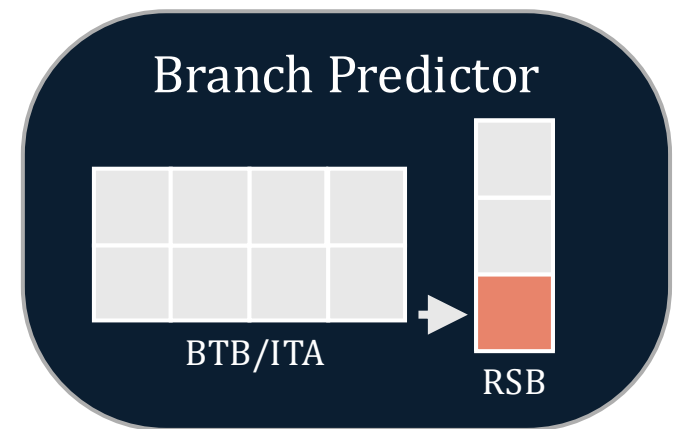
Return mispredictions



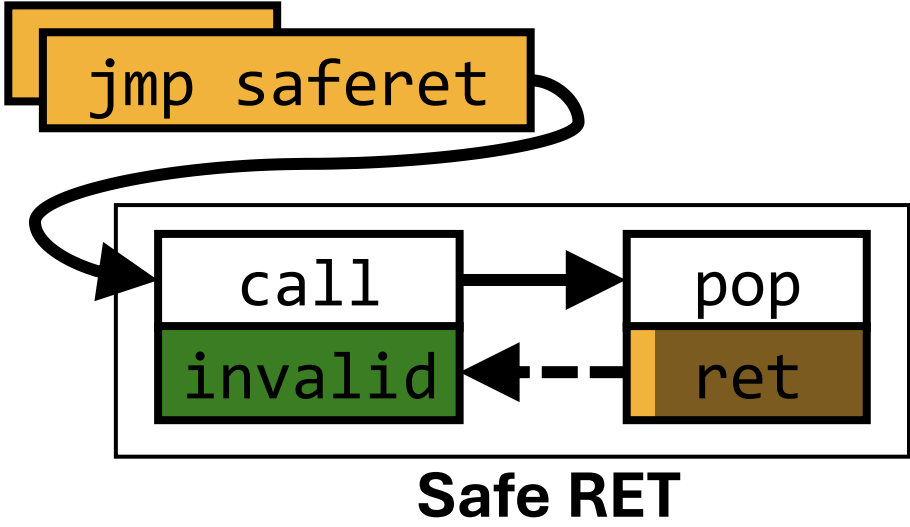
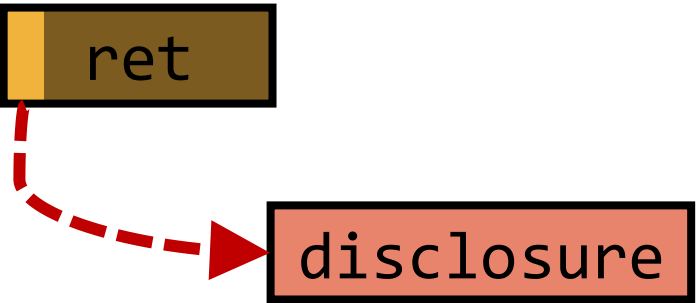
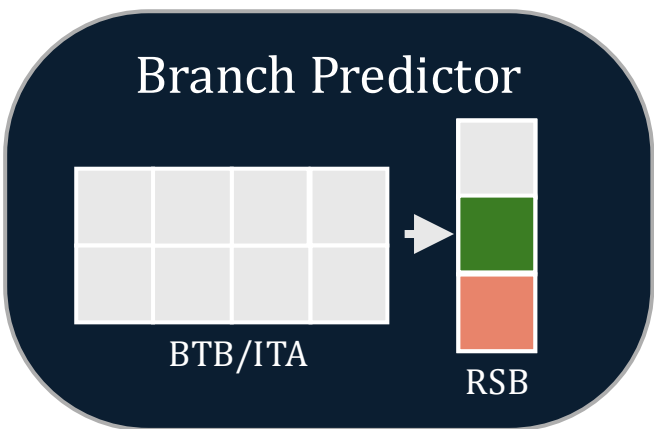
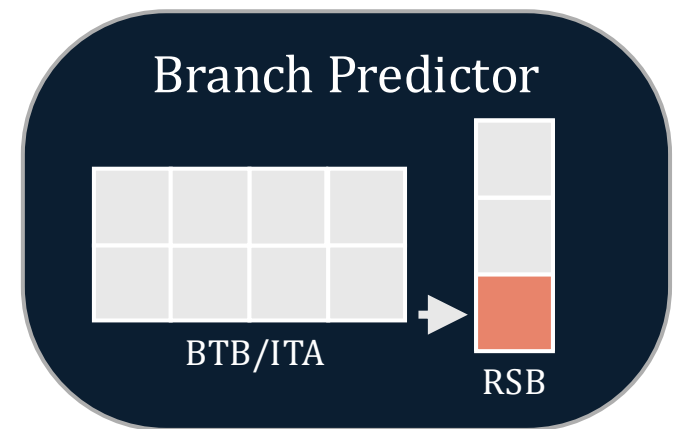
Return mispredictions



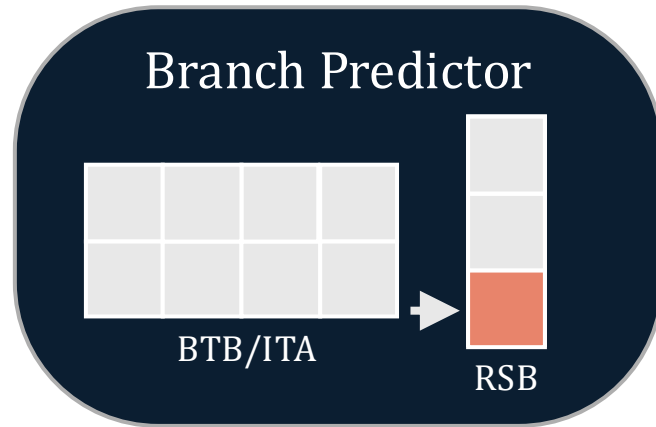
Return mispredictions



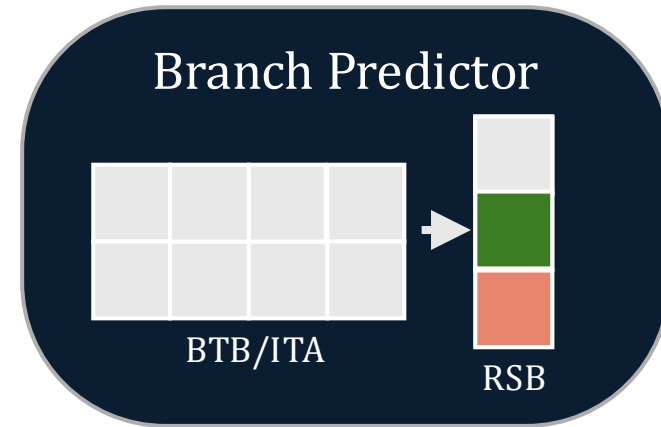
Return mispredictions



Protecting returns with SafeRET



Unprotected return



SafeRET

**SafeRET *neutralizes* the top of the RSB
right before a return**

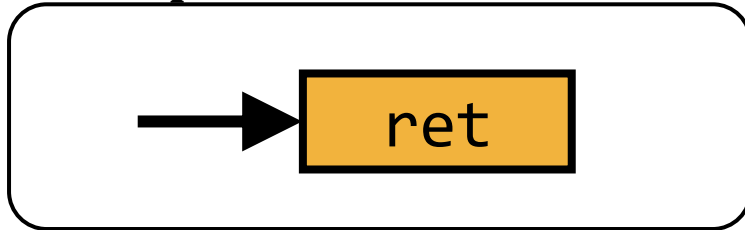
Neutralization

Removing the influence of the existing predictor state

Neutralization

Removing the influence of the existing predictor state

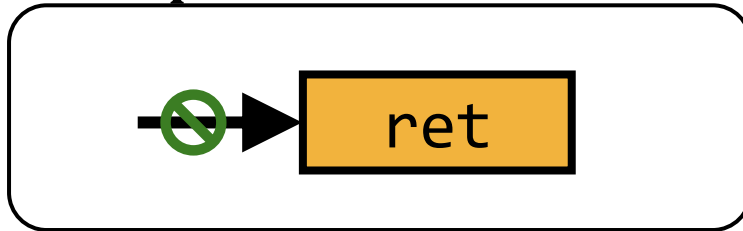
In-place neutralization



Neutralization

Removing the influence of the existing predictor state

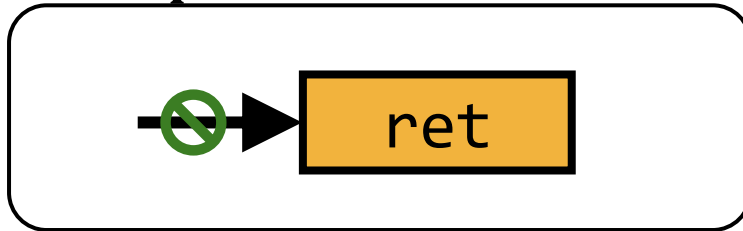
In-place neutralization



Neutralization

Removing the influence of the existing predictor state

In-place neutralization

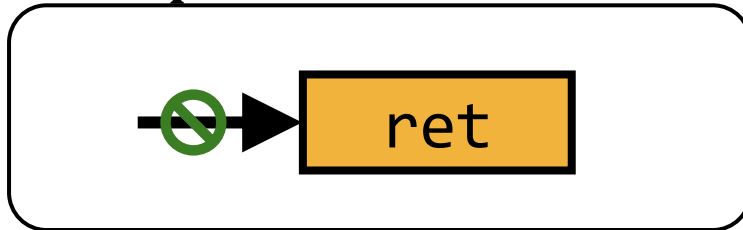


e.g., Safe RET, retpoline

Neutralization

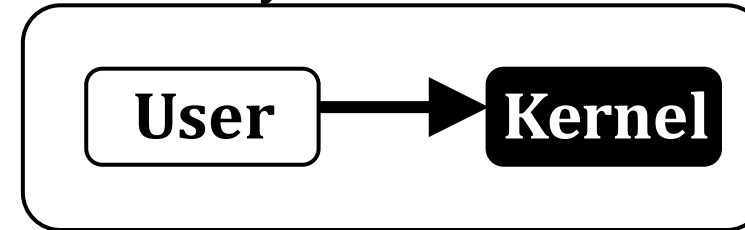
Removing the influence of the existing predictor state

In-place neutralization



e.g., Safe RET, retpoline

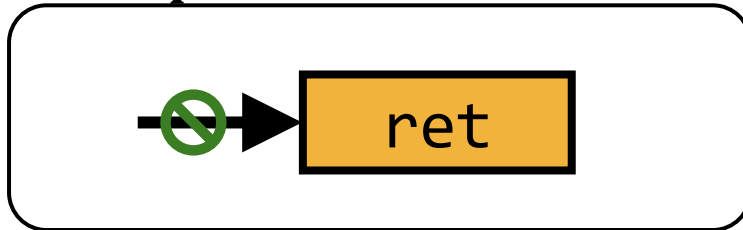
Entry neutralization



Neutralization

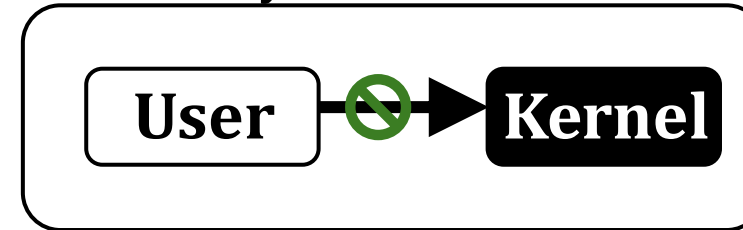
Removing the influence of the existing predictor state

In-place neutralization



e.g., Safe RET, retpoline

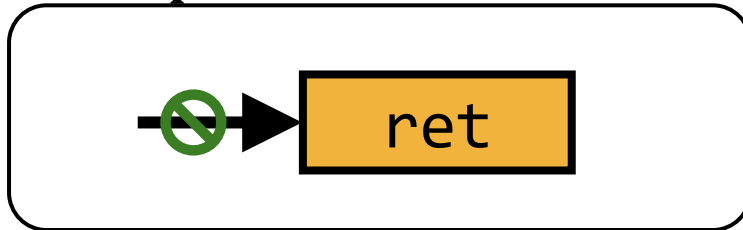
Entry neutralization



Neutralization

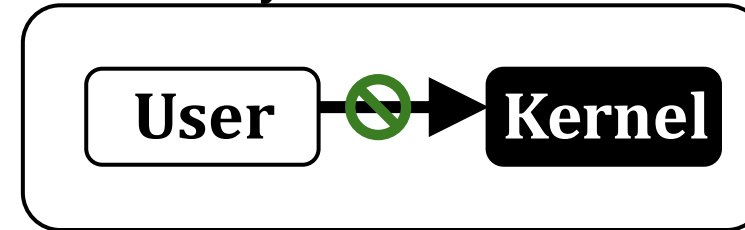
Removing the influence of the existing predictor state

In-place neutralization



e.g., Safe RET, retpoline

Entry neutralization

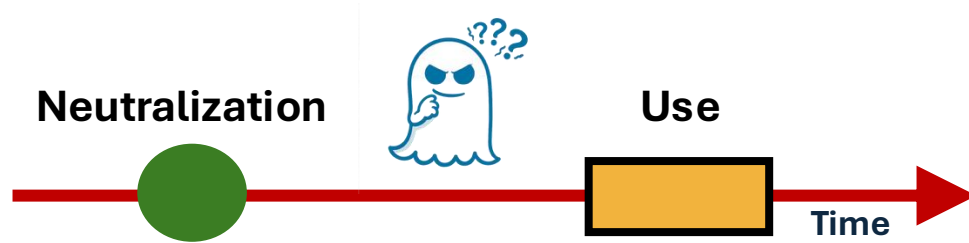


e.g., SW loop, BHI_DIS_S

This work

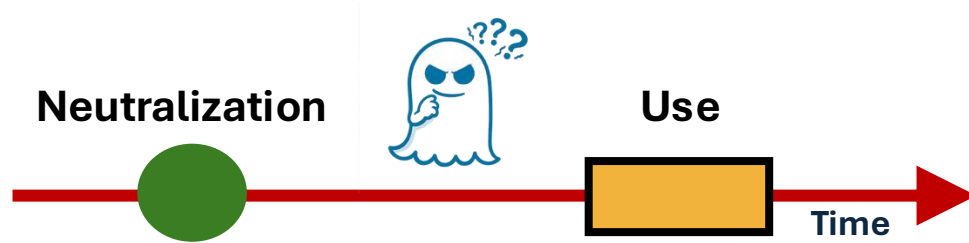
This work

1 Identification of TONTOU

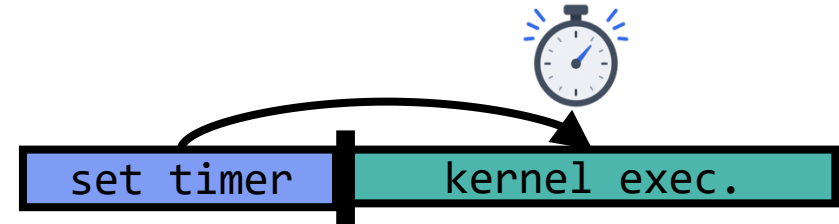


This work

1 Identification of TONTOU

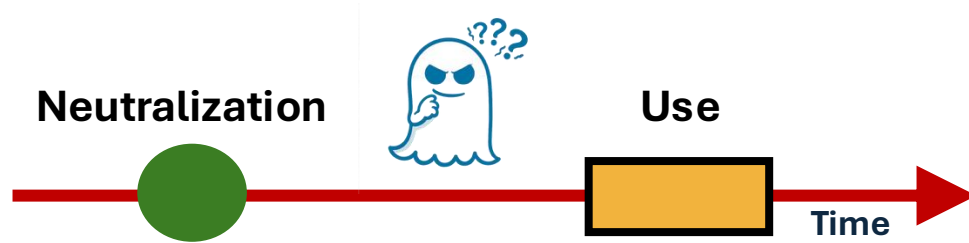


2 INTERRUPT INJECTION

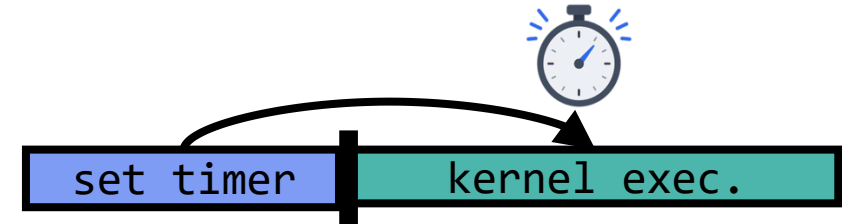


This work

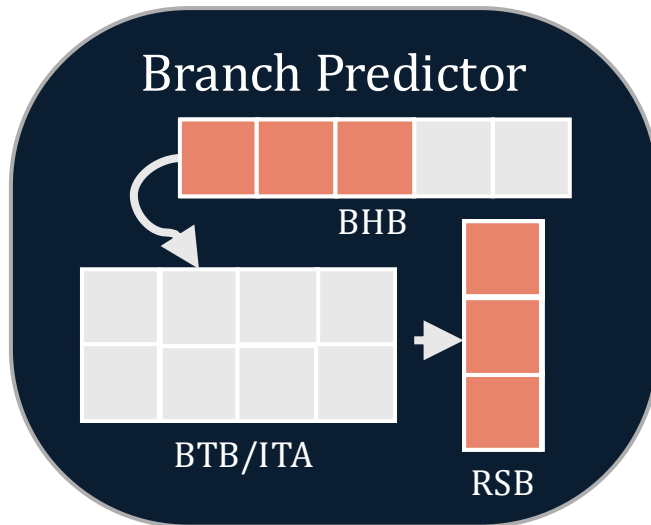
1 Identification of TONTOU



2 INTERRUPT INJECTION

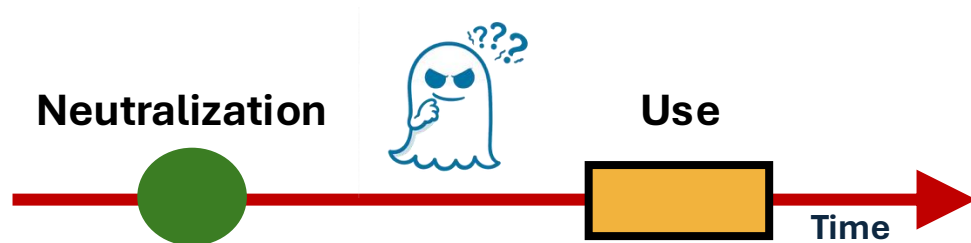


3 Branch predictor poisoning

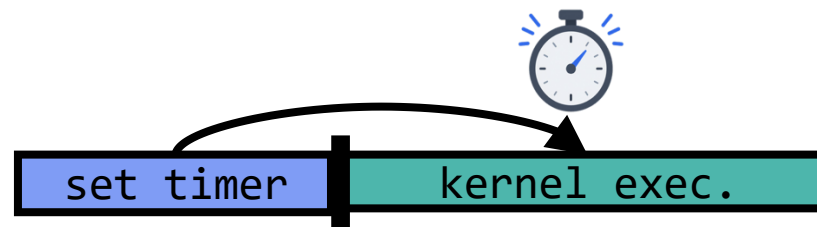


This work

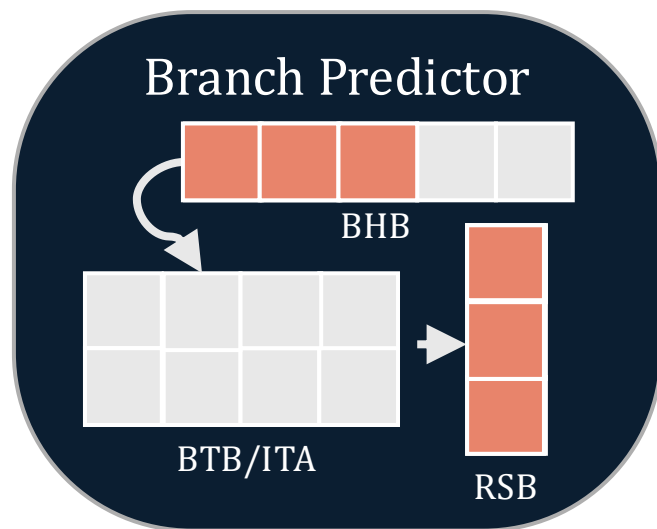
1 Identification of TONTOU



2 INTERRUPT INJECTION



3 Branch predictor poisoning



4 End-to-end attack

```
danieltrujillo — daniel@AMD-Zen-2: ~/interrupt_injection_demo — ssh - ssh zen2 — 180x50
root:$y$j9T$7jKpI/9QPCZwINqScIHKu1$VbnyVr7DwP2EkGUB1DAWXXcys5Kuc1TERFM.4ZCzoX9:206
daemon*:20134:0:99999:7:::
bin*:20134:0:99999:7:::
sys*:20134:0:99999:7:::
sync*:20134:0:99999:7:::
games*:20134:0:99999:7:::
man*:2013 :0:99999:7:::
lp*:20134:0:99999:7:::
mail*:20134:0:99999:7:::
news*:20134:0:99999:7:::
uucp*:20134:0:99999:7:::
proxy*:20134:0:99999:7:::
www-data*:20134:0:99999:7:::
```

Time-of-Neutralization to Time-of-Use (TONTOU)

Time-of-Neutralization to Time-of-Use (TONTOU)

Neutralization



Time-of-Neutralization to Time-of-Use (TONTOU)

Neutralization



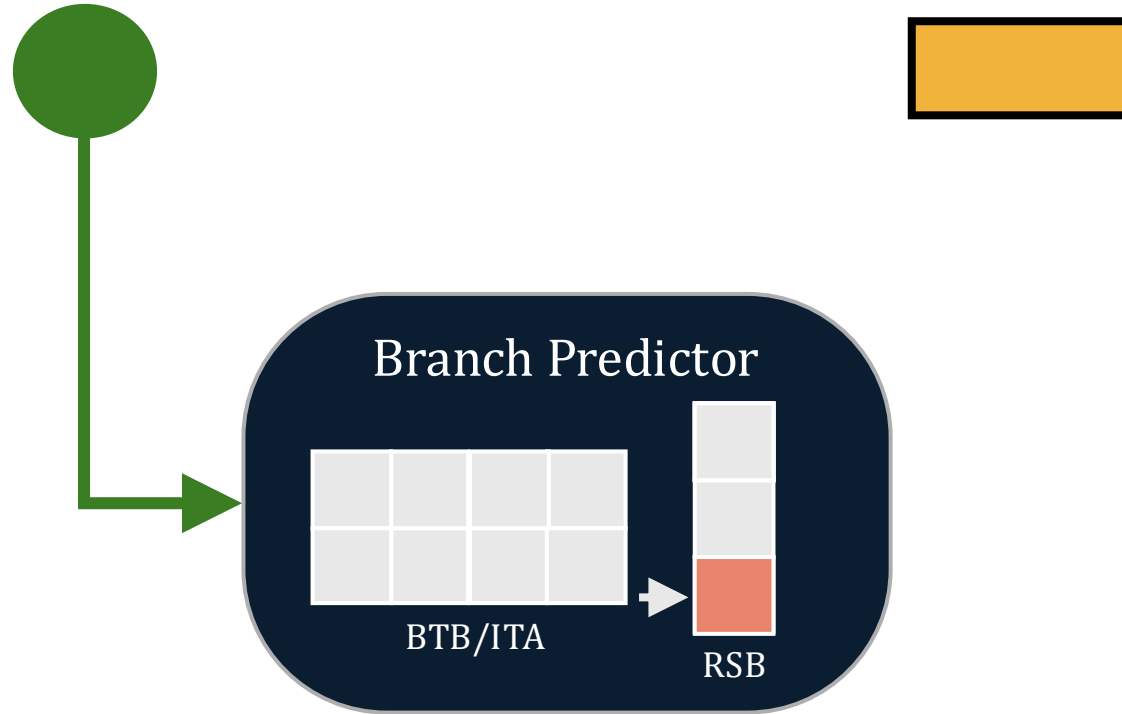
Use



Time-of-Neutralization to Time-of-Use (TONTOU)

Neutralization

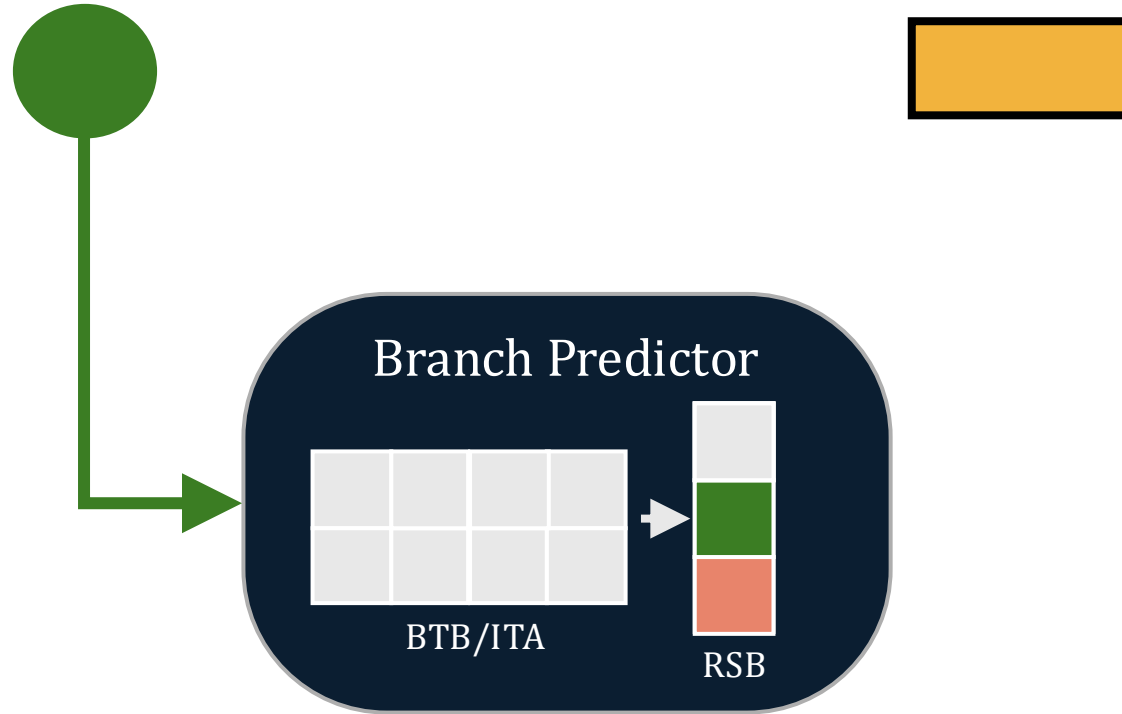
Use



Time-of-Neutralization to Time-of-Use (TONTOU)

Neutralization

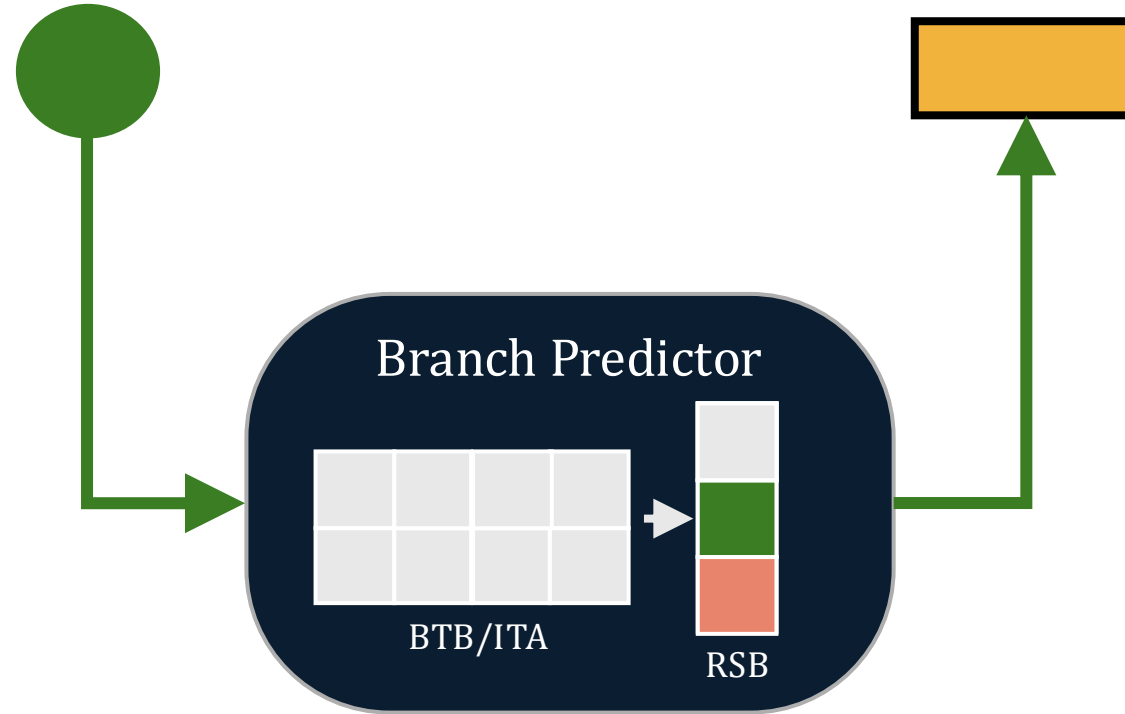
Use



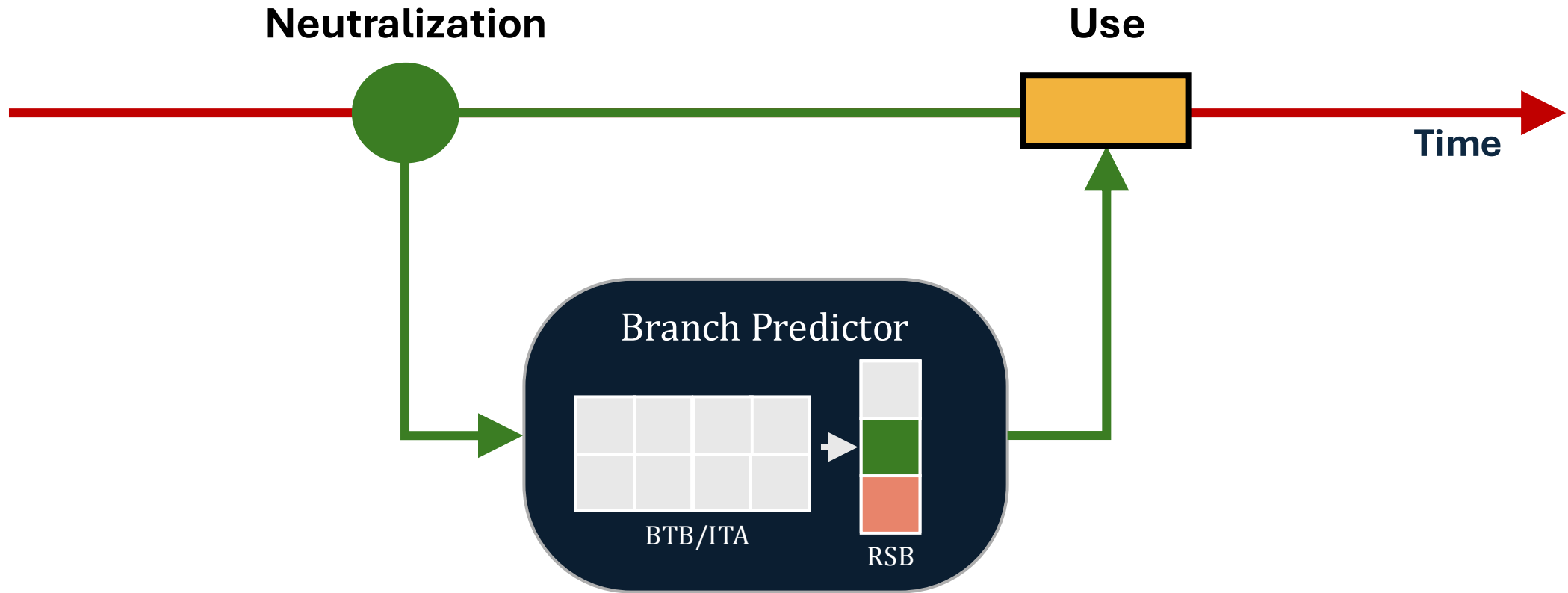
Time-of-Neutralization to Time-of-Use (TONTOU)

Neutralization

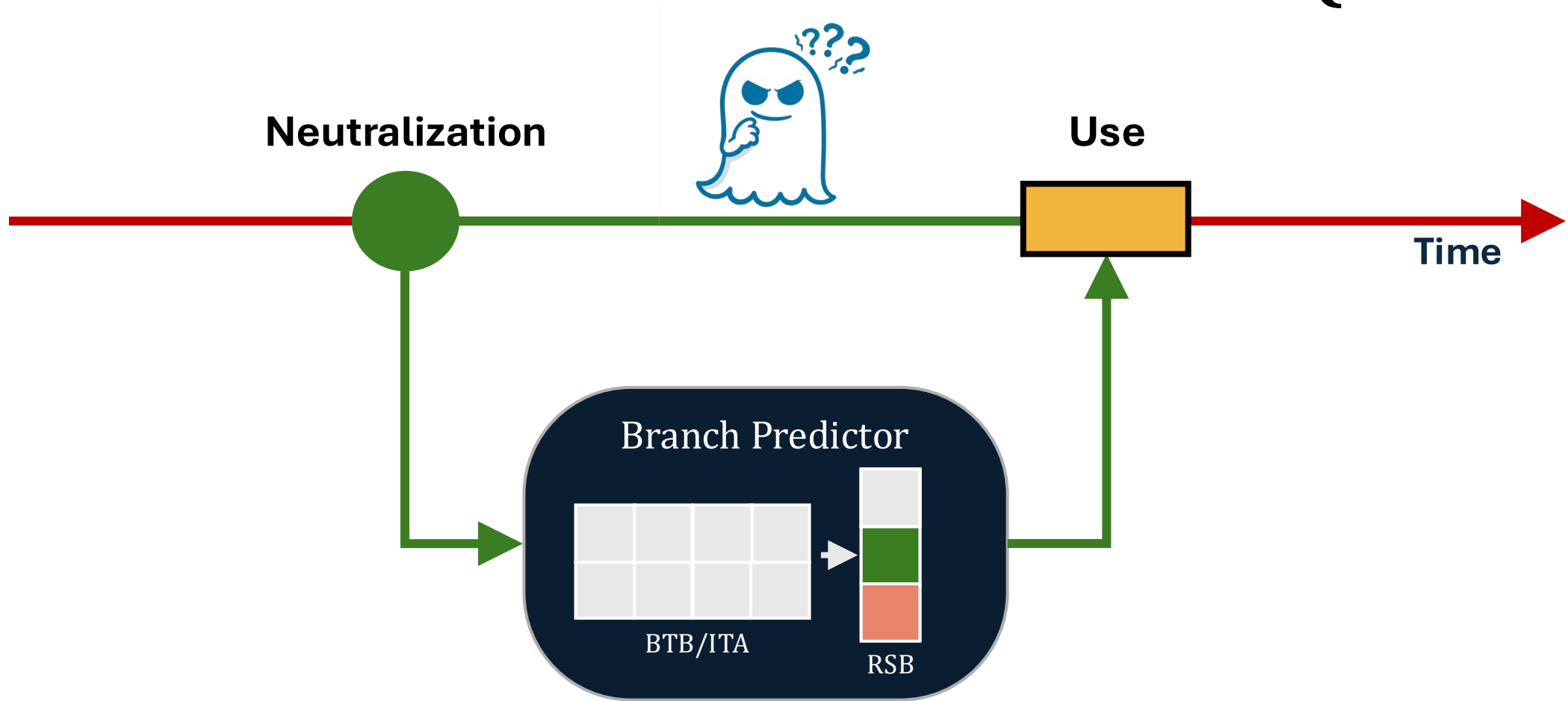
Use



Time-of-Neutralization to Time-of-Use (TONTOU)

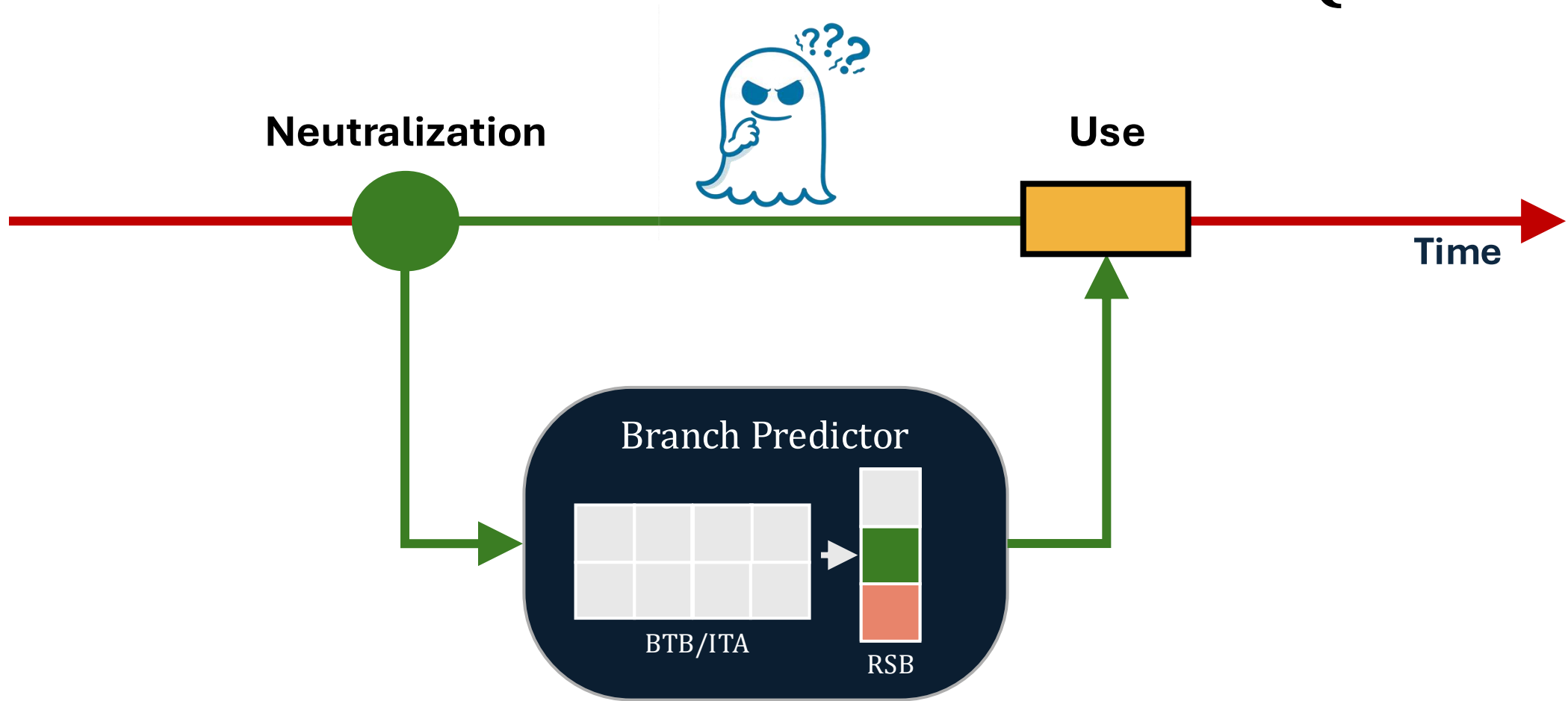


Time-of-Neutralization to Time-of-Use (TONTOU)



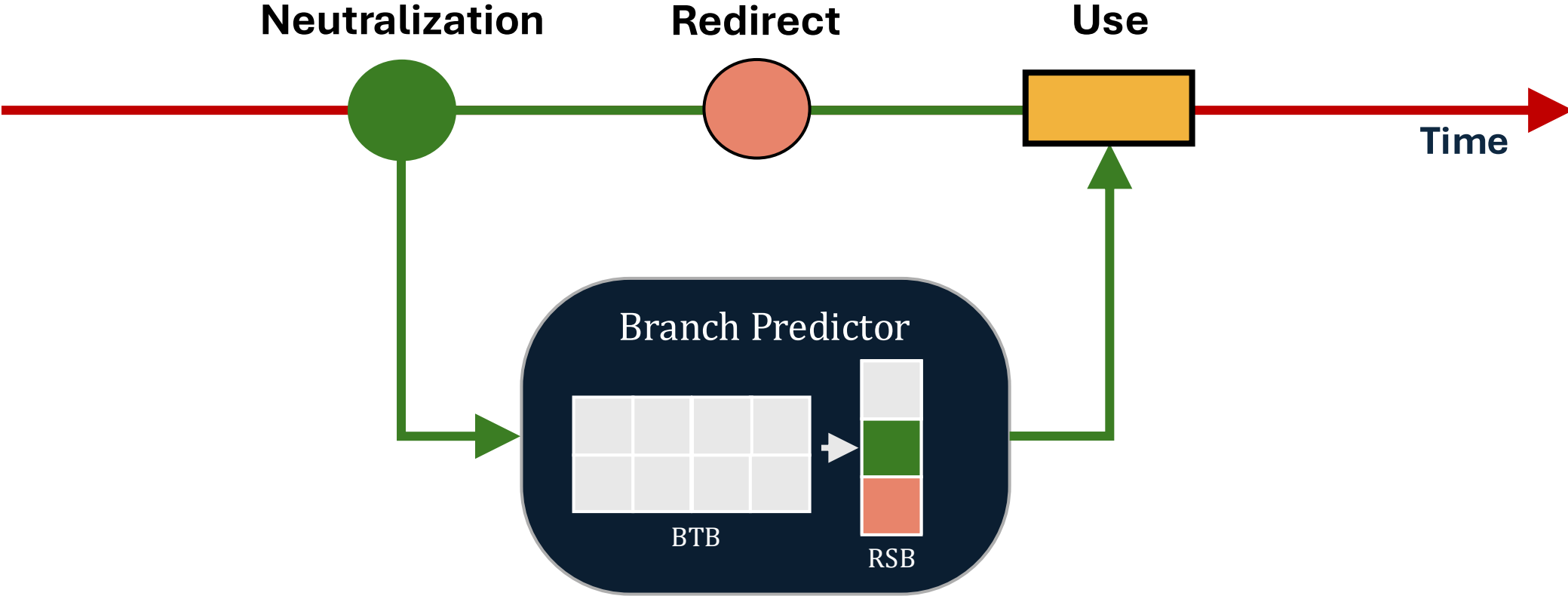
Security assumption: no attacker influence in this time window

Time-of-Neutralization to Time-of-Use (TONTOU)

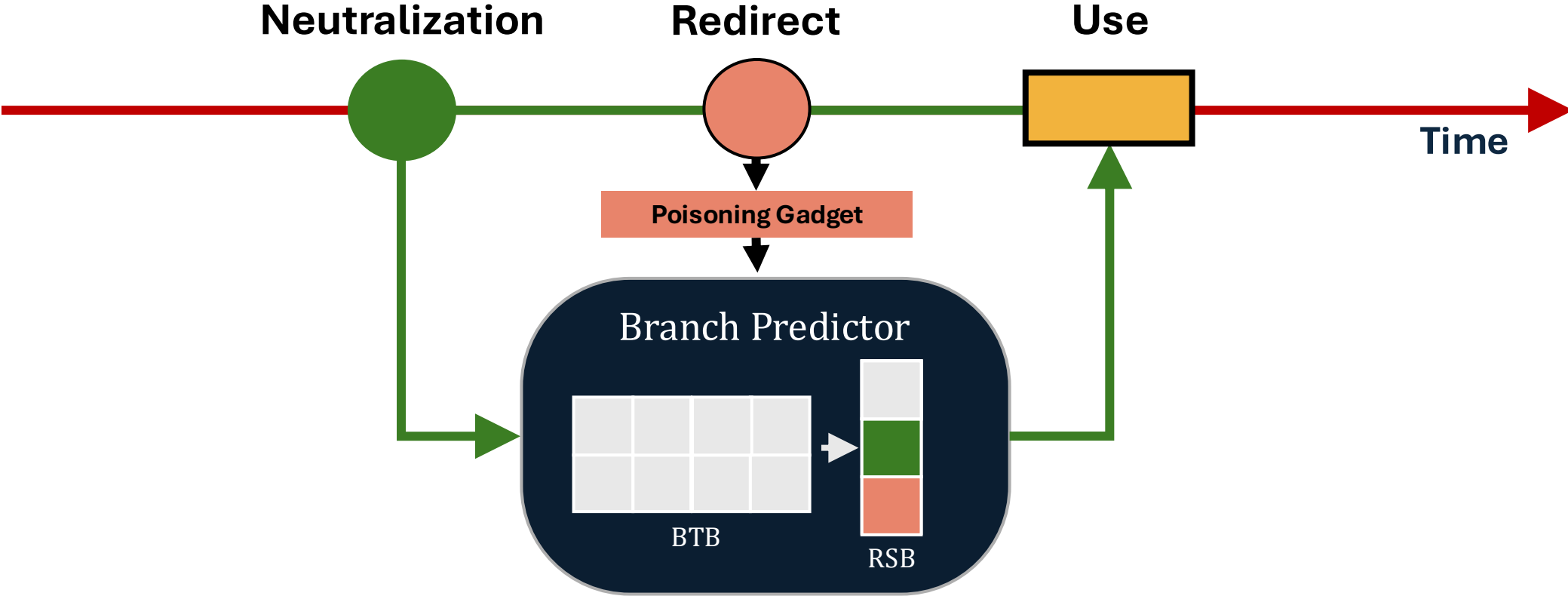


Security assumption: no attacker influence in this time window

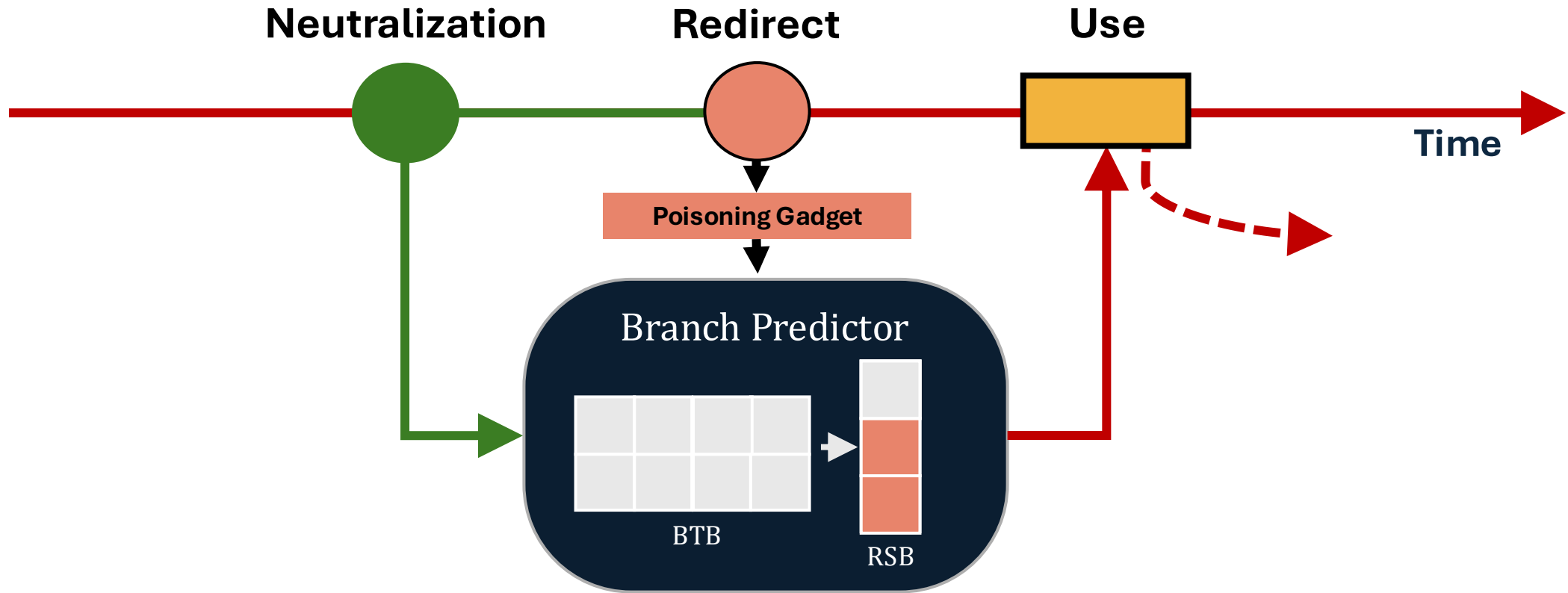
Time-of-Neutralization to Time-of-Use (TONTOU)



Time-of-Neutralization to Time-of-Use (TONTOU)

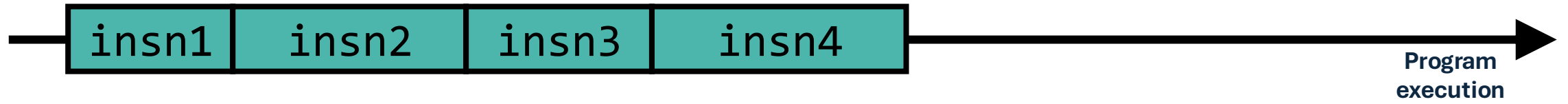


Time-of-Neutralization to Time-of-Use (TONTOU)

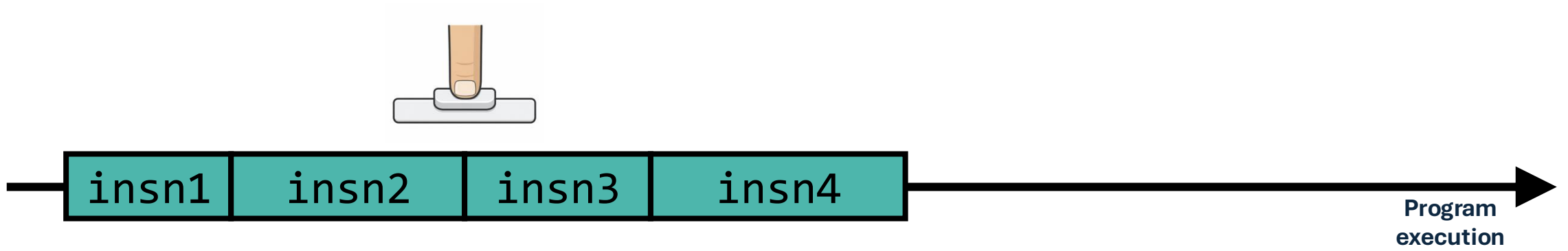


TONTOU attacks break neutralization-based Spectre defenses

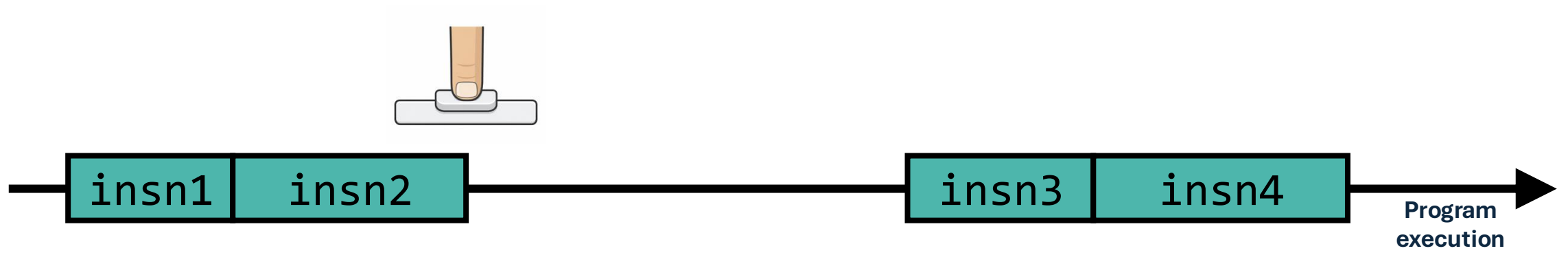
Redirecting the CPU



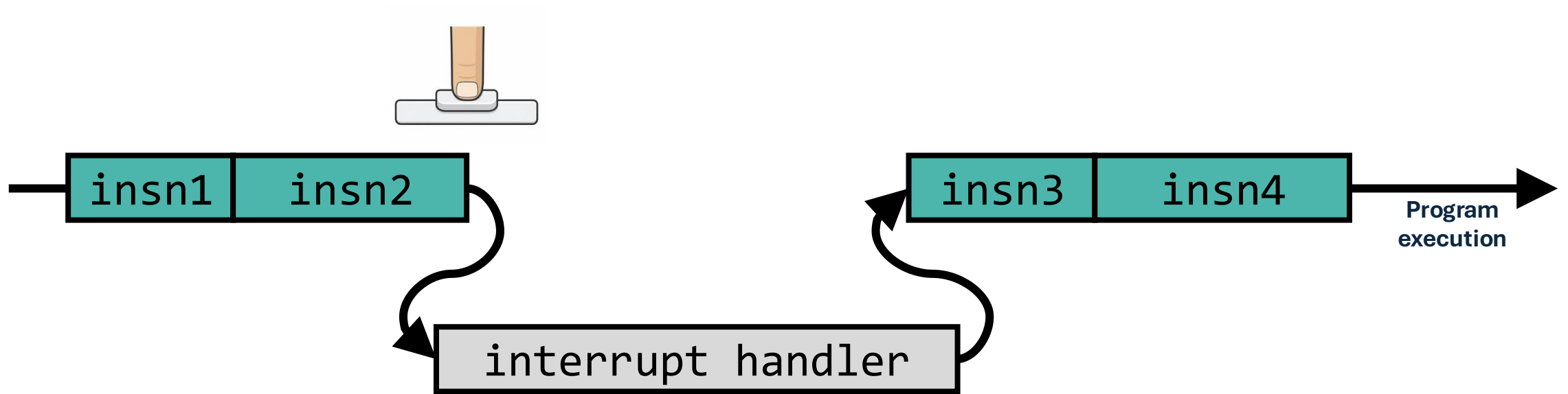
Redirecting the CPU



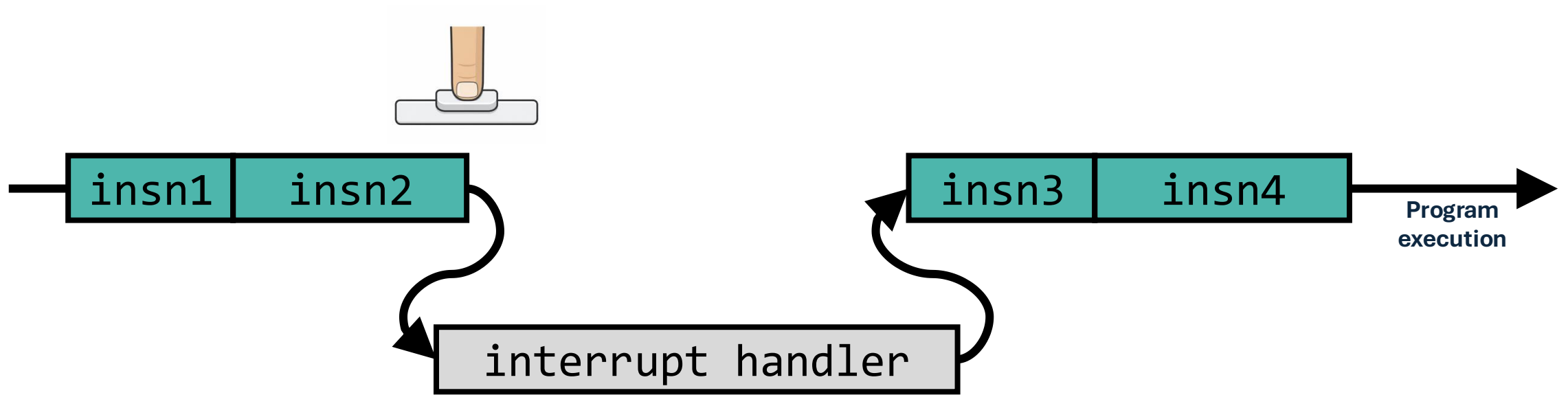
Redirecting the CPU



Redirecting the CPU

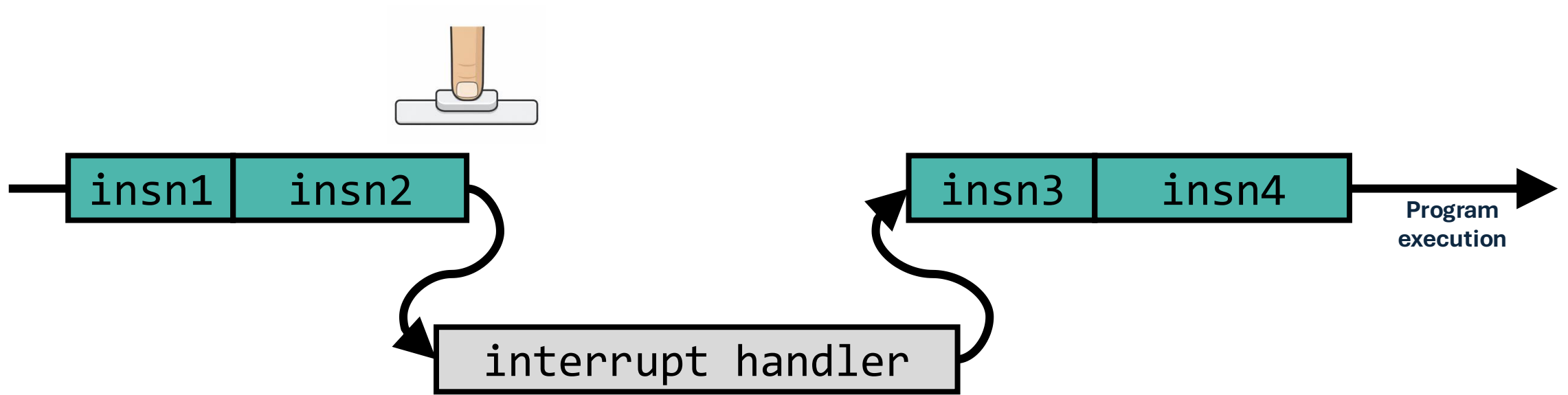


Redirecting the CPU

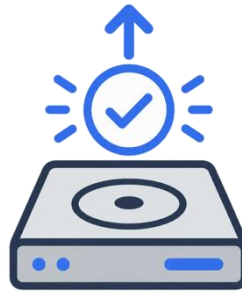


Packet arrival

Redirecting the CPU

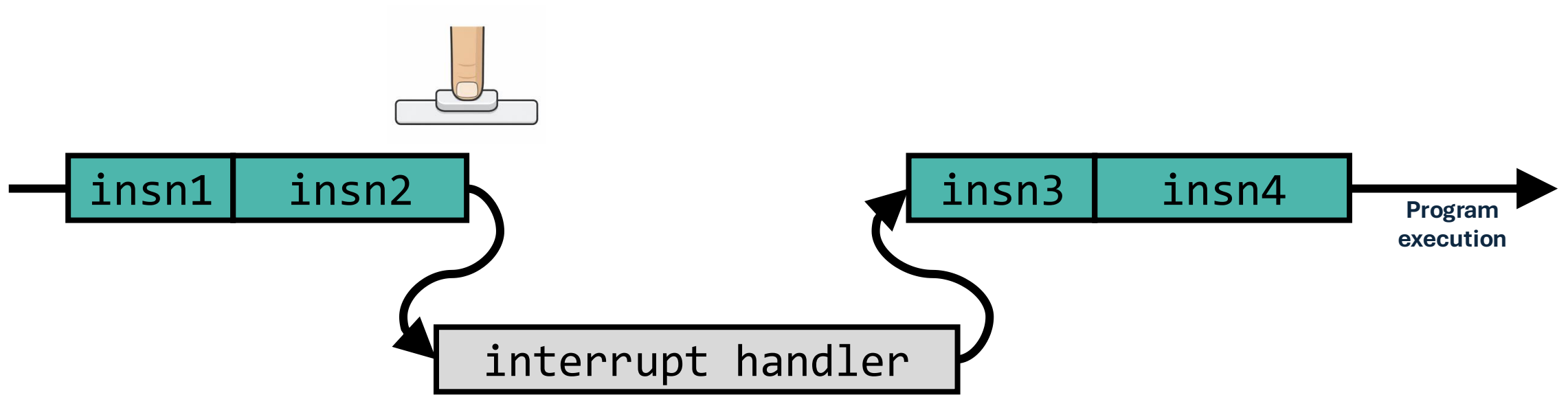


Packet arrival

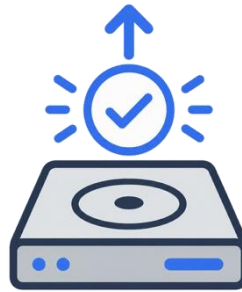


I/O completion

Redirecting the CPU



Packet arrival



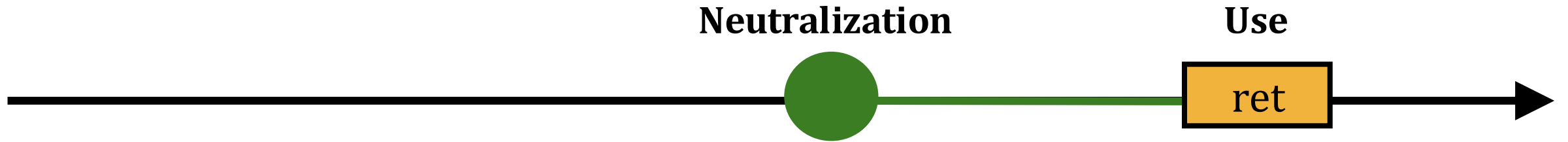
I/O completion



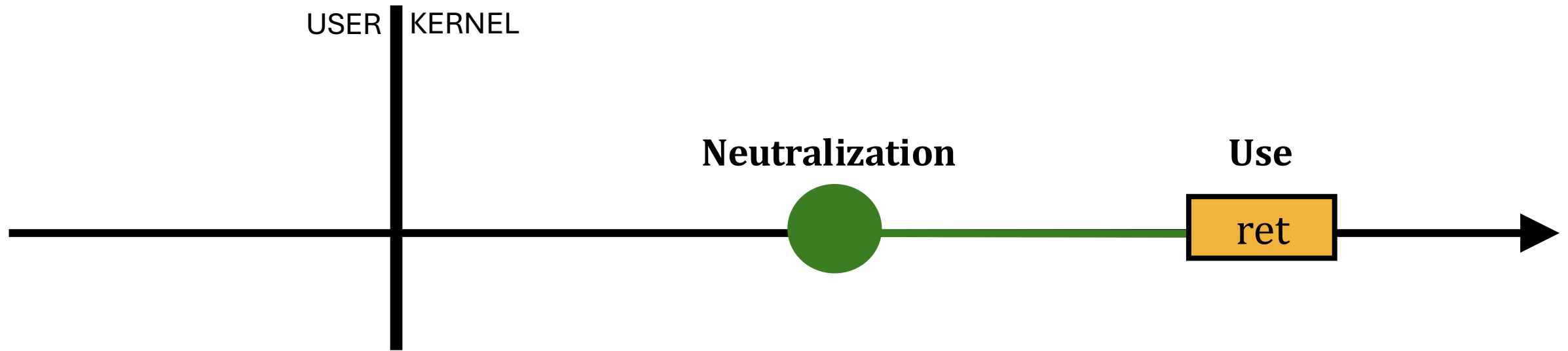
Timer expiry

INTERRUPT INJECTION

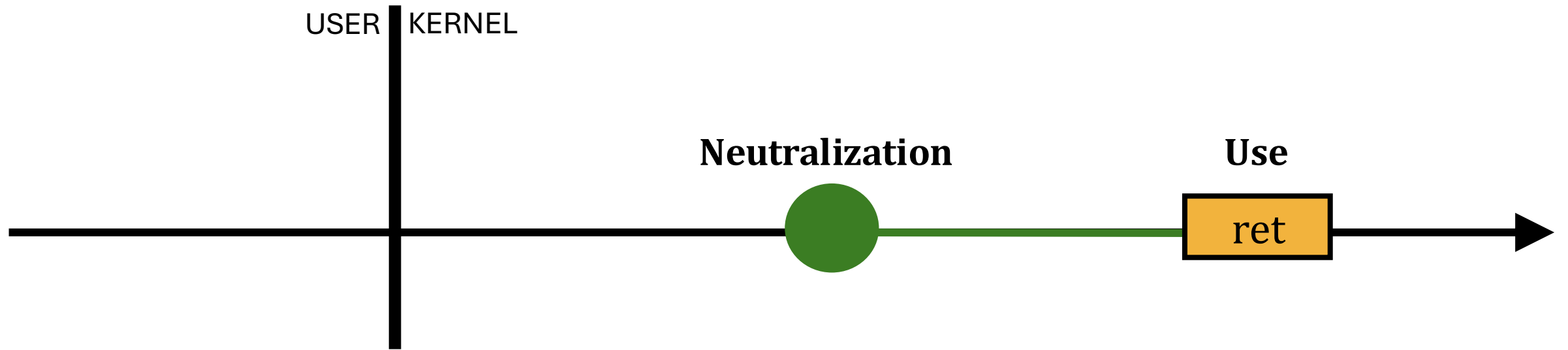
INTERRUPT INJECTION



INTERRUPT INJECTION

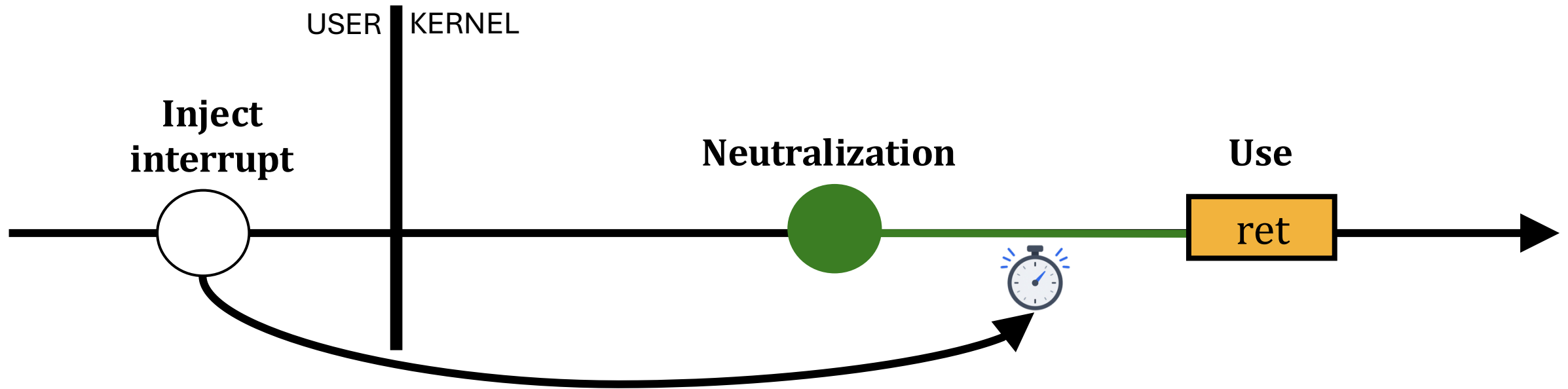


INTERRUPT INJECTION



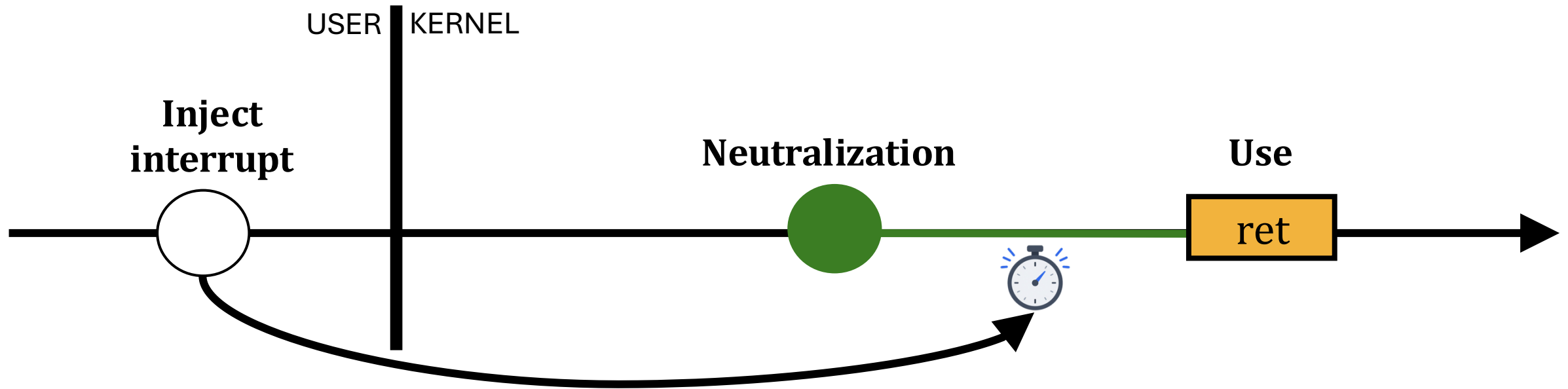
How to inject an interrupt in the kernel?

INTERRUPT INJECTION



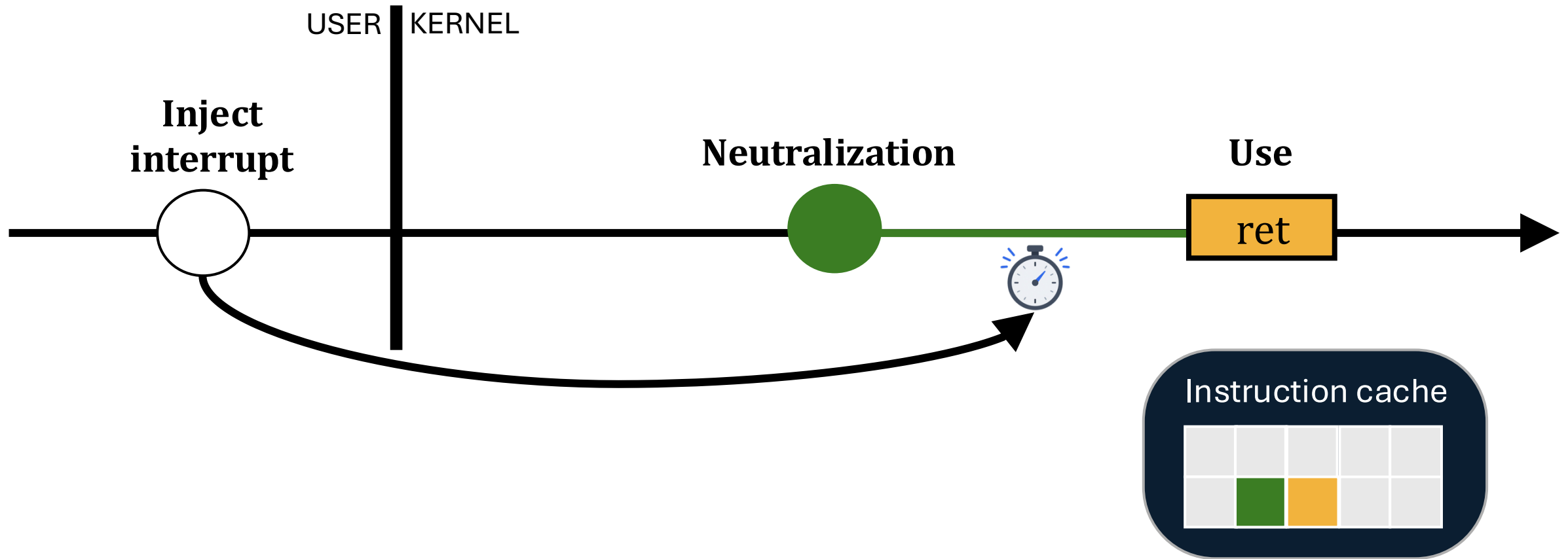
How to inject an interrupt in the kernel?
We use ordinary timers!

INTERRUPT INJECTION



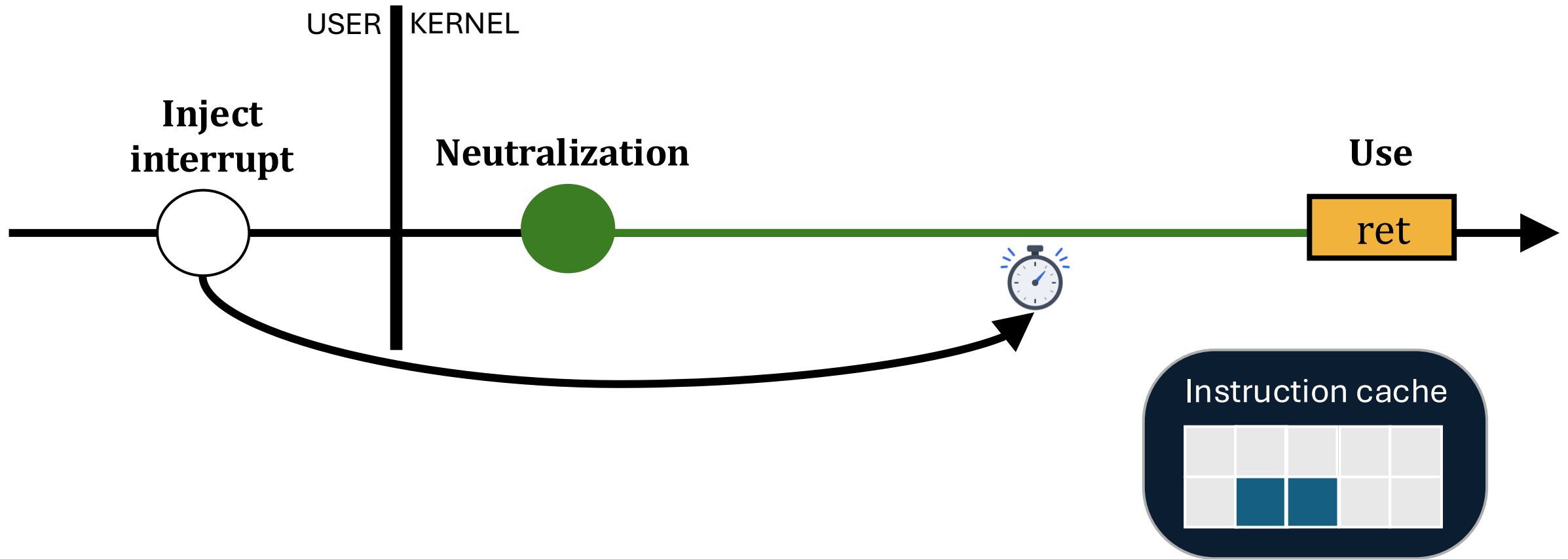
What if the TONTOU window is short?

INTERRUPT INJECTION



What if the TONTOU window is short?

INTERRUPT INJECTION

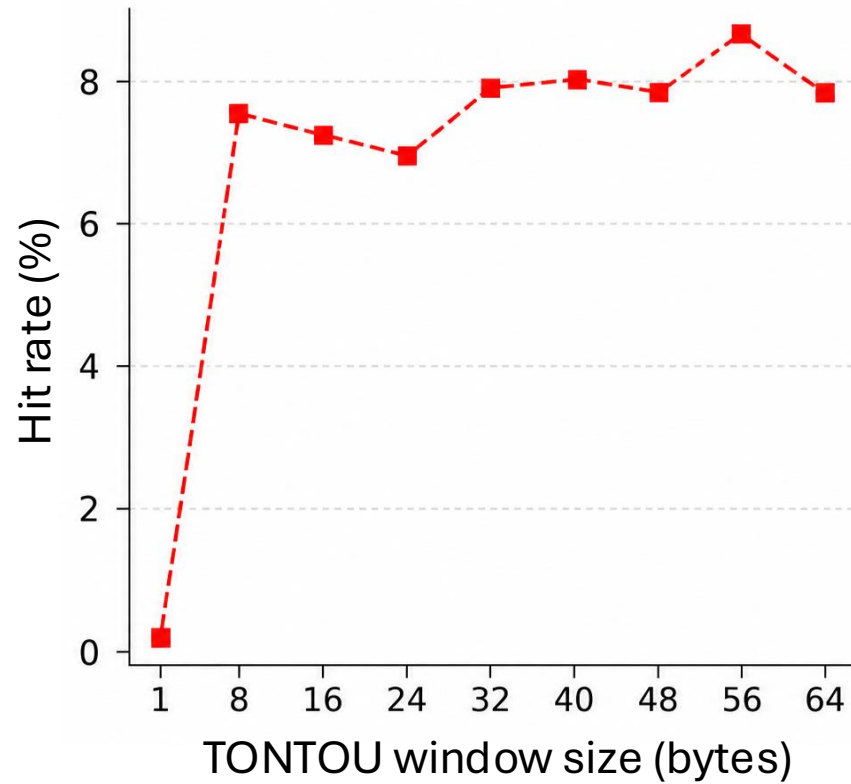


What if the TONTOU window is short?
We slow it down by evicting it from the cache!

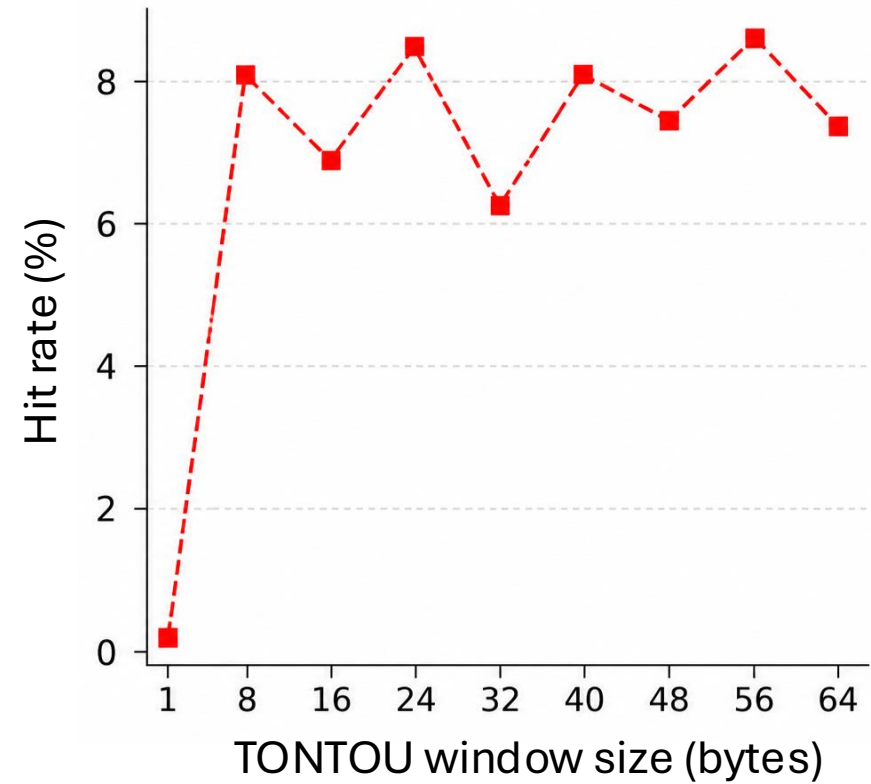
INTERRUPT INJECTION

INTERRUPT INJECTION

AMD Zen 2

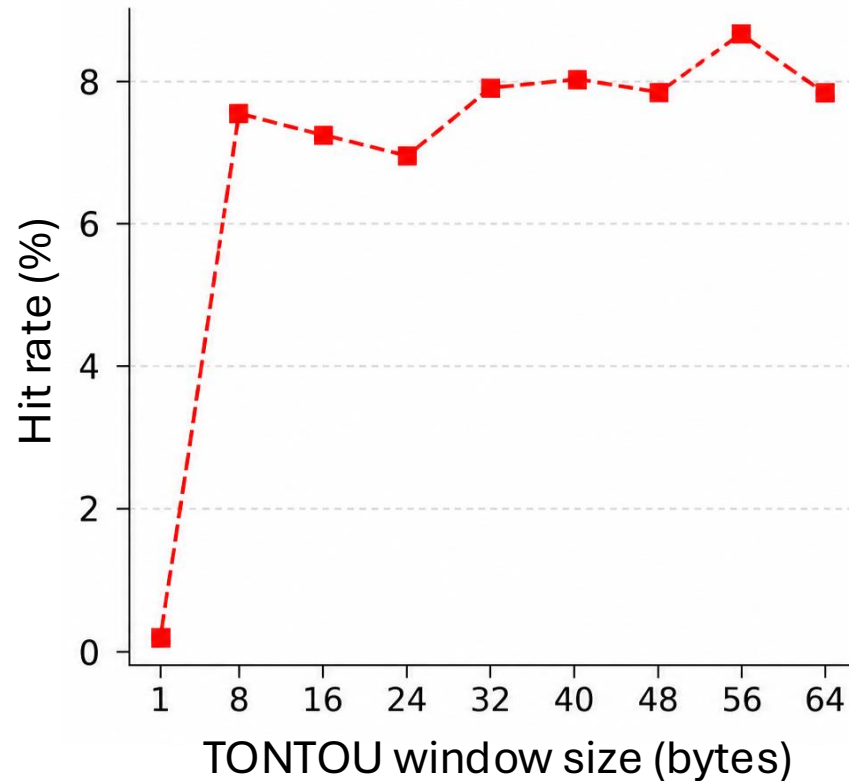


AMD Zen 4

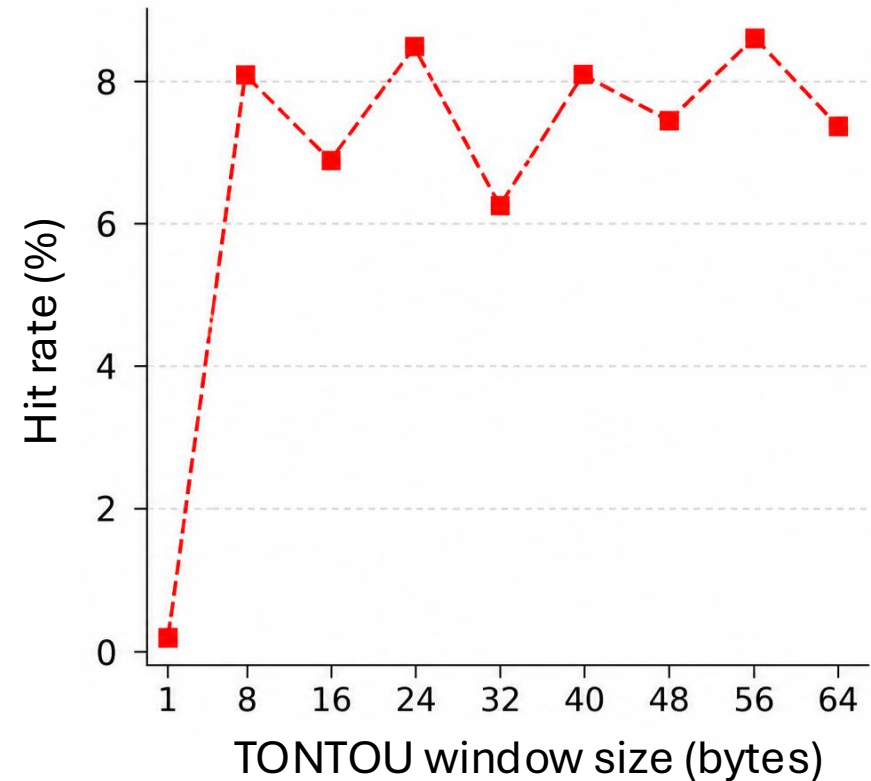


INTERRUPT INJECTION

AMD Zen 2



AMD Zen 4



We can hit a small TONTOU window with relatively high probability

Poisoning using existing techniques

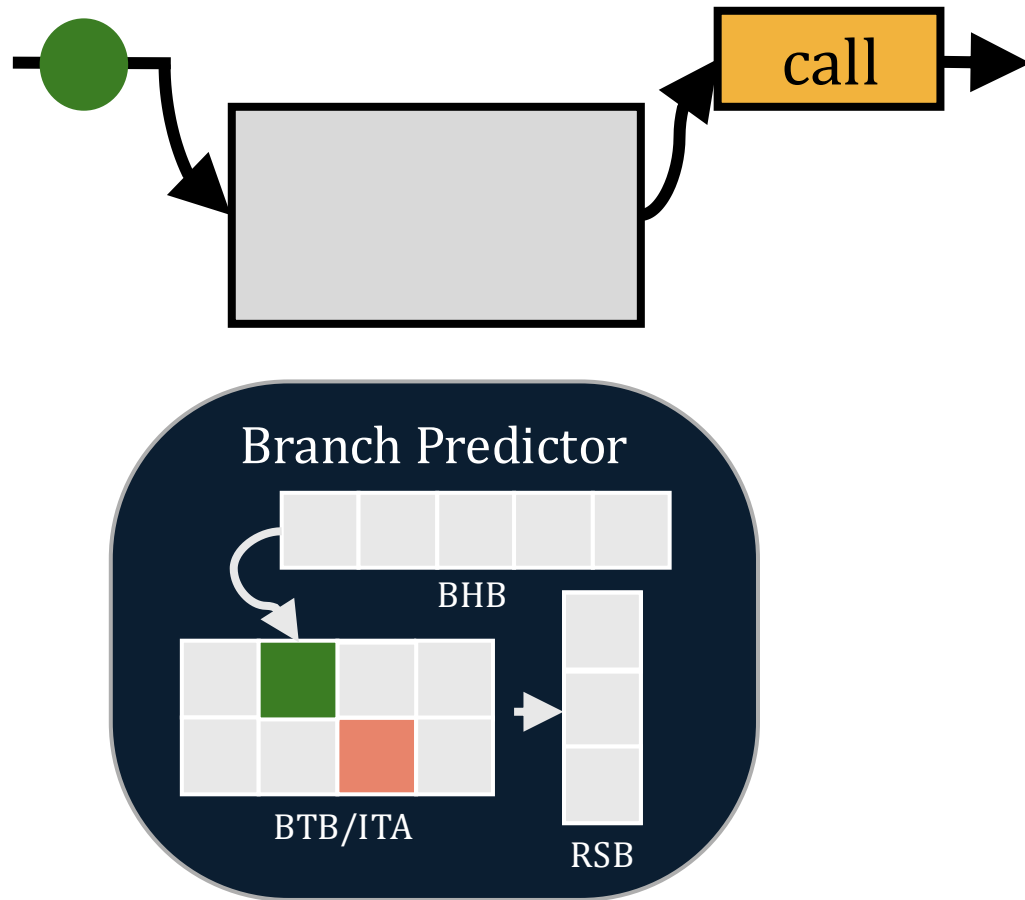


Poisoning using existing techniques



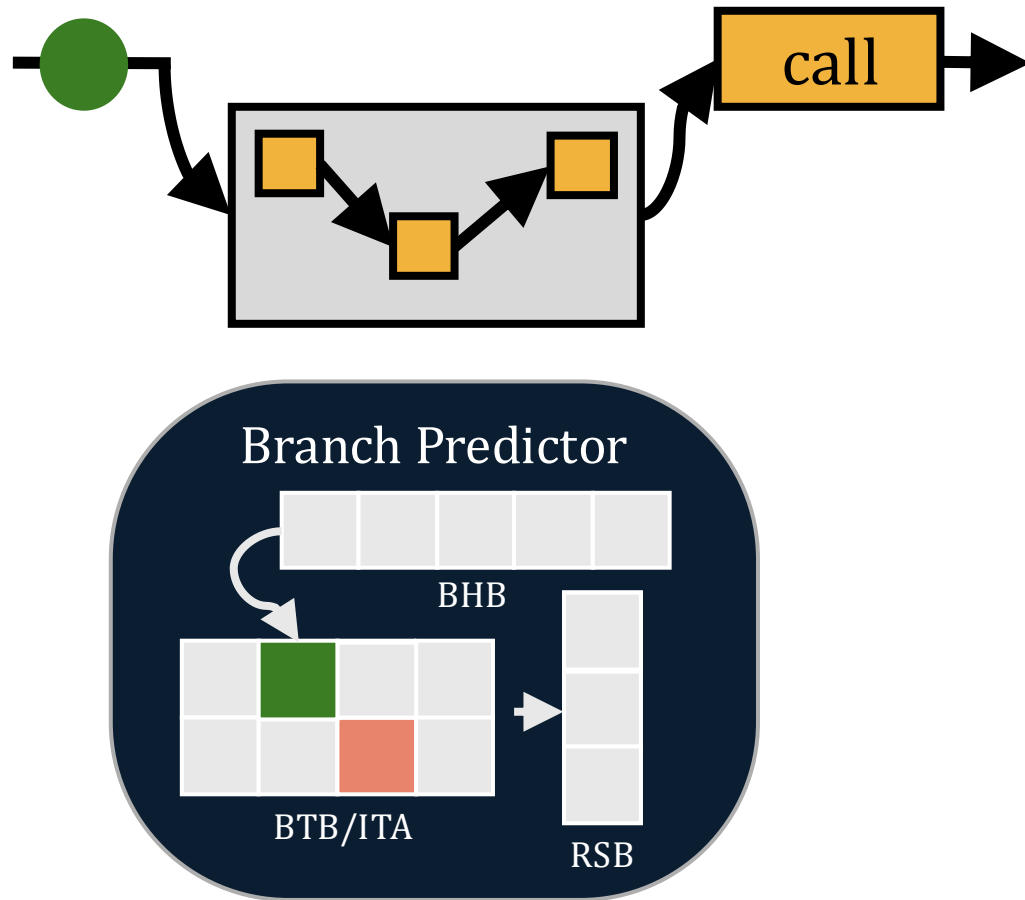
Interrupts mistrain jumps/calls
by poisoning the BHB

Poisoning using existing techniques



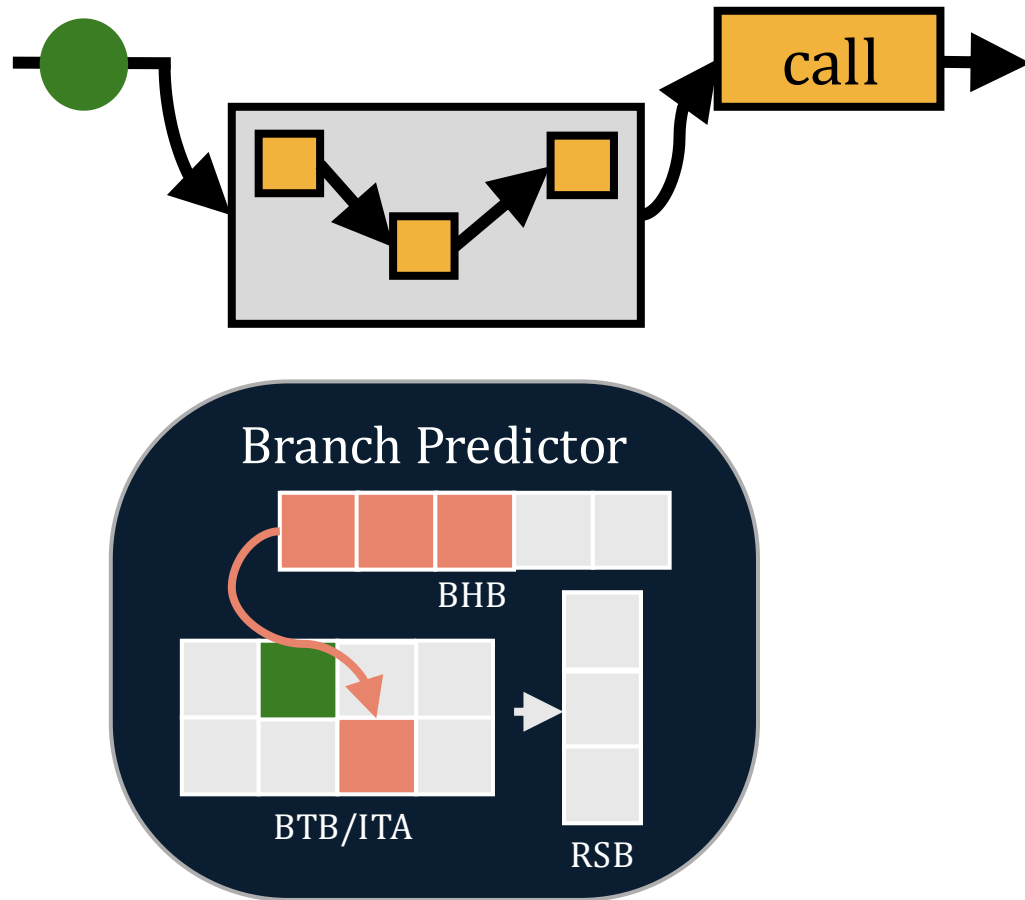
Interrupts mistrain jmps/calls
by poisoning the BHB

Poisoning using existing techniques



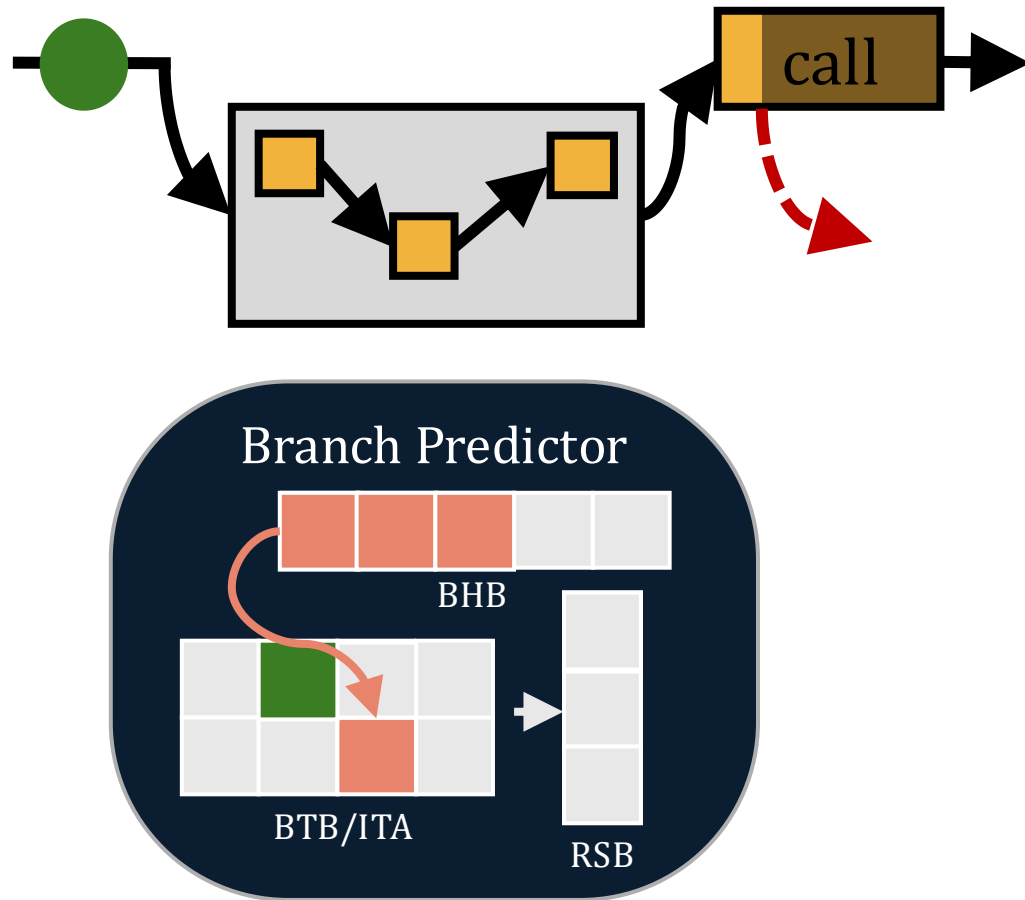
Interrupts mistrain jmps/calls
by poisoning the BHB

Poisoning using existing techniques



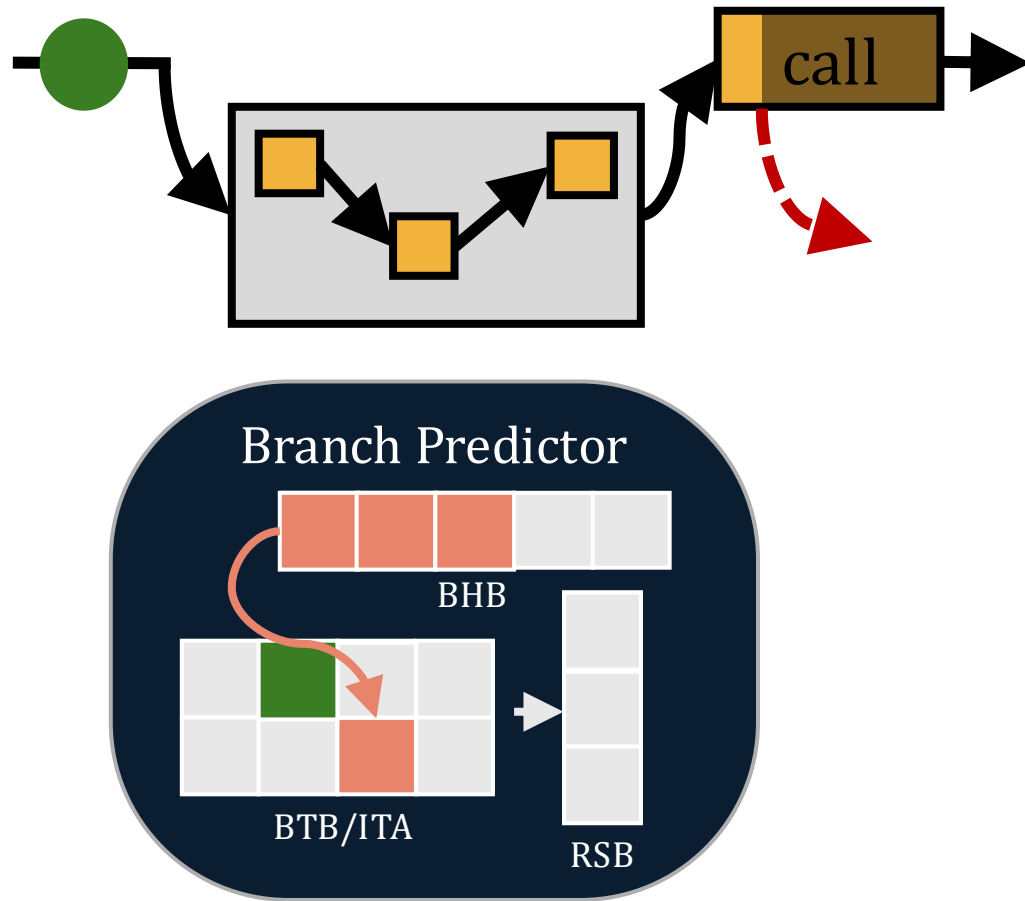
Interrupts mistrain jmps/calls
by poisoning the BHB

Poisoning using existing techniques



Interrupts mistrain jmps/calls
by poisoning the BHB

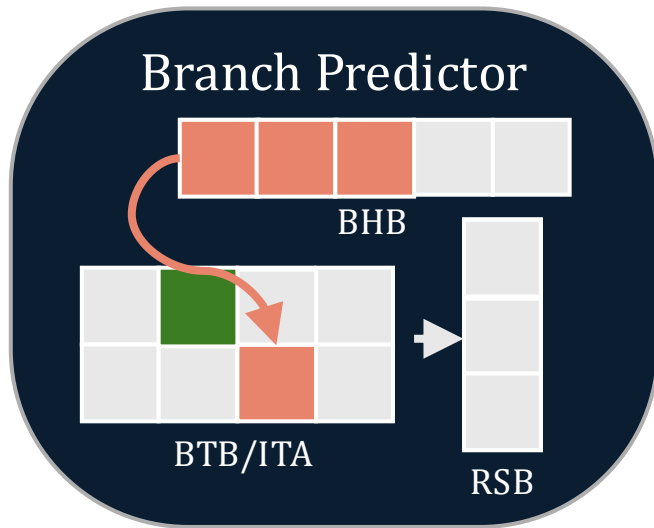
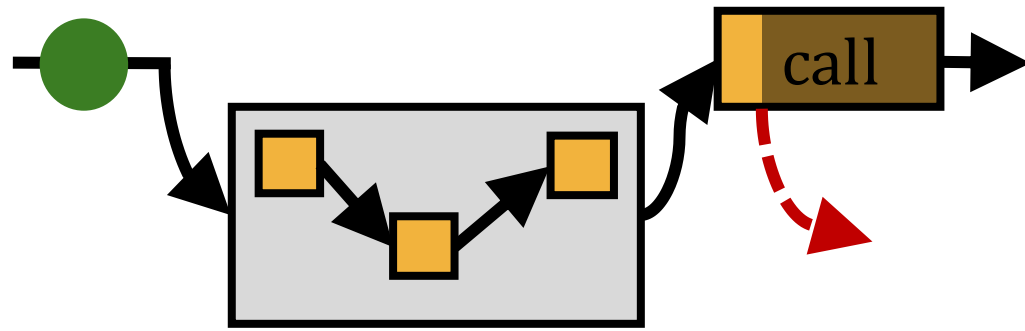
Poisoning using existing techniques



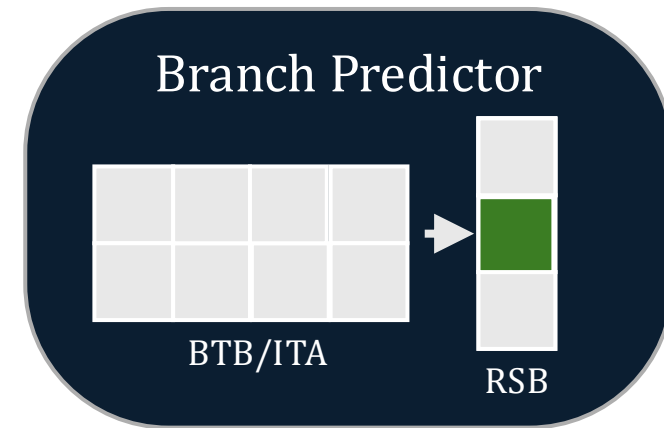
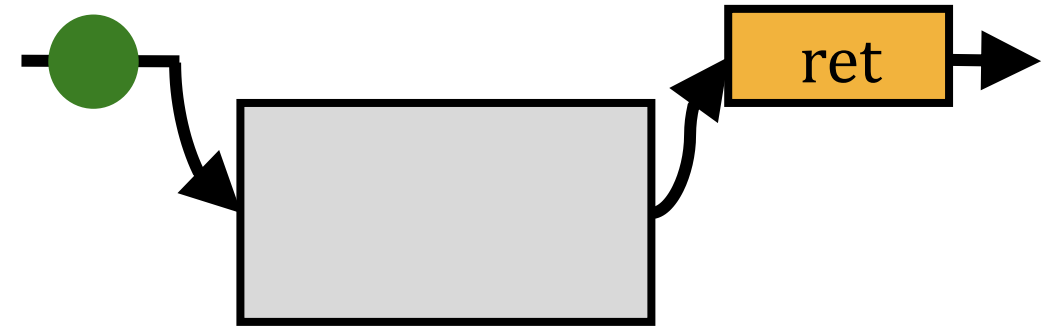
Interrupts mistrain jmps/calls
by poisoning the BHB

Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

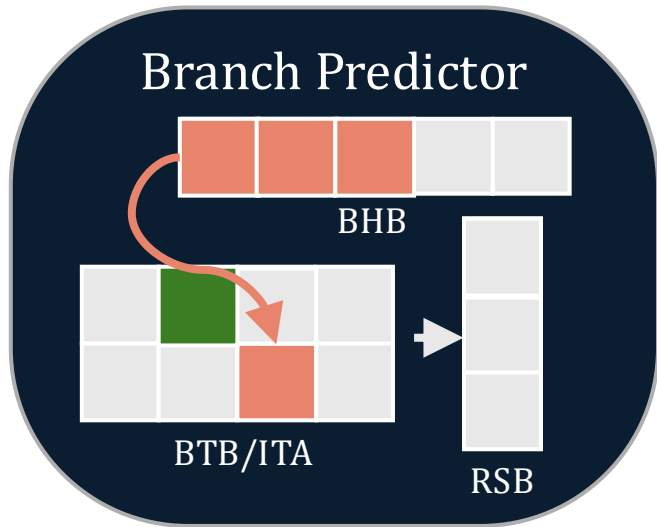
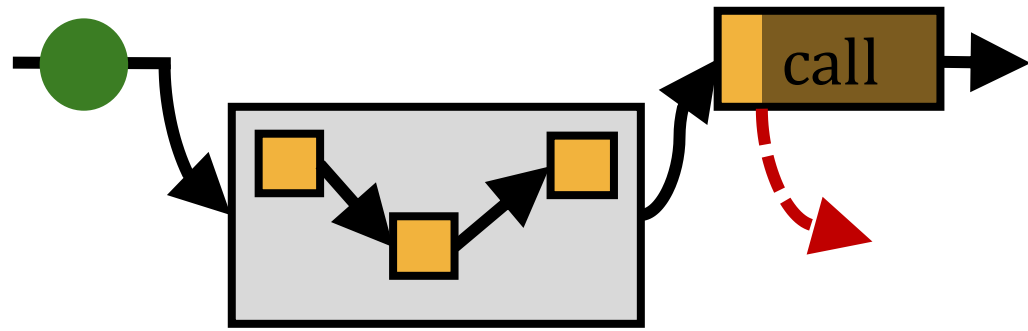


Interrupts mistrain jmps/calls
by poisoning the BHB

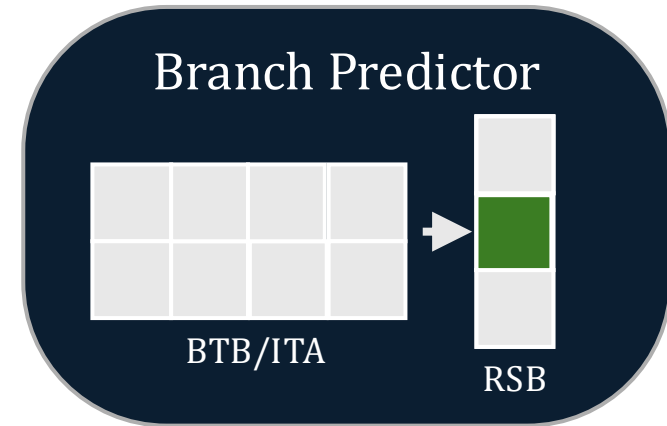
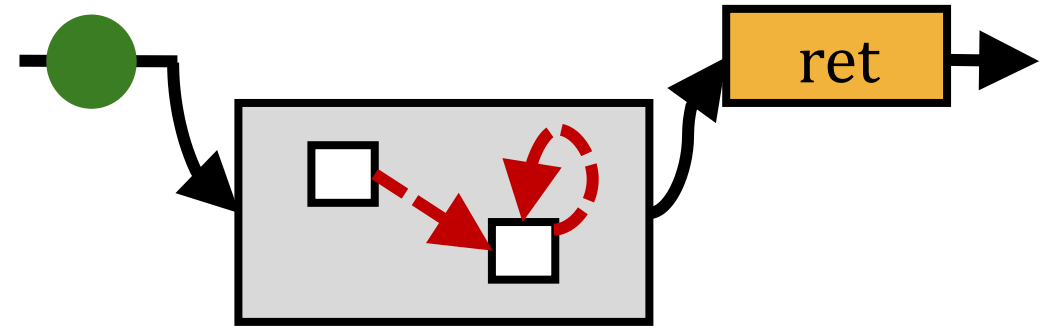


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

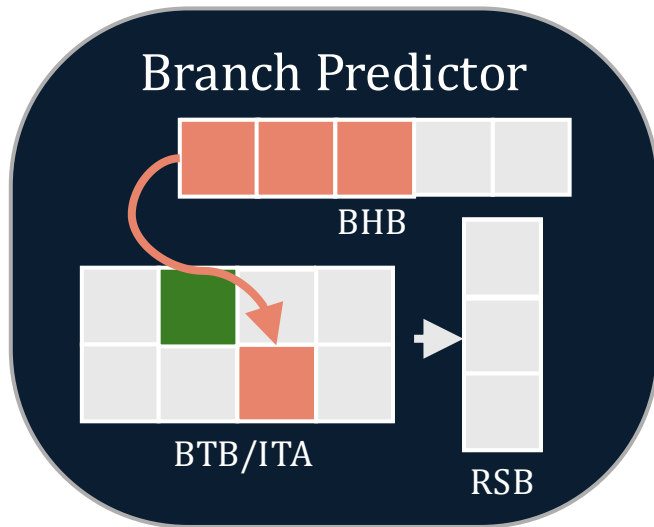
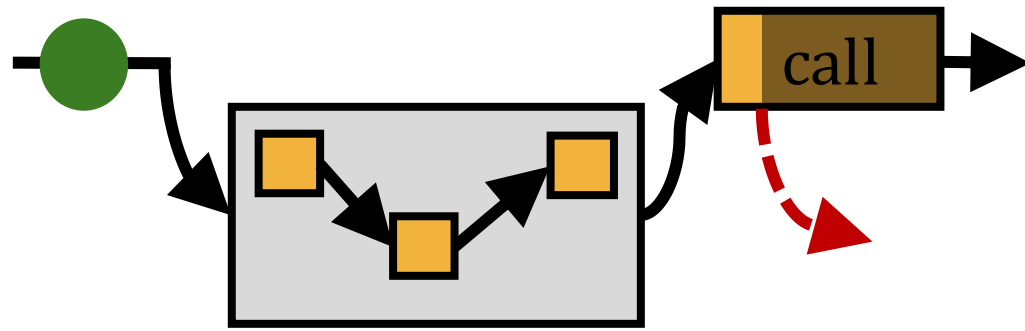


Interrupts mistrain jmps/calls
by poisoning the BHB

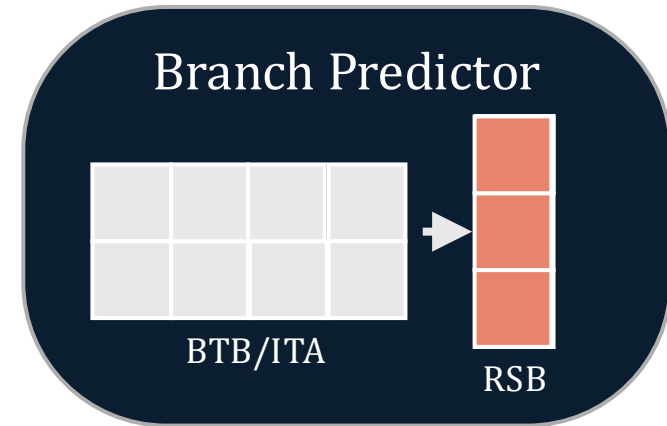
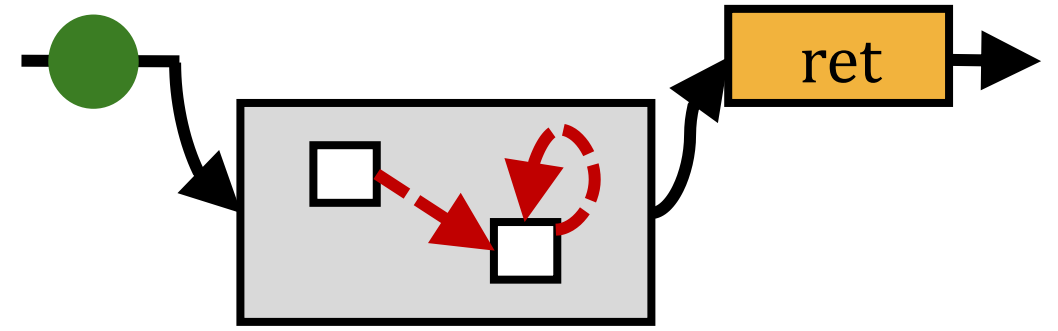


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

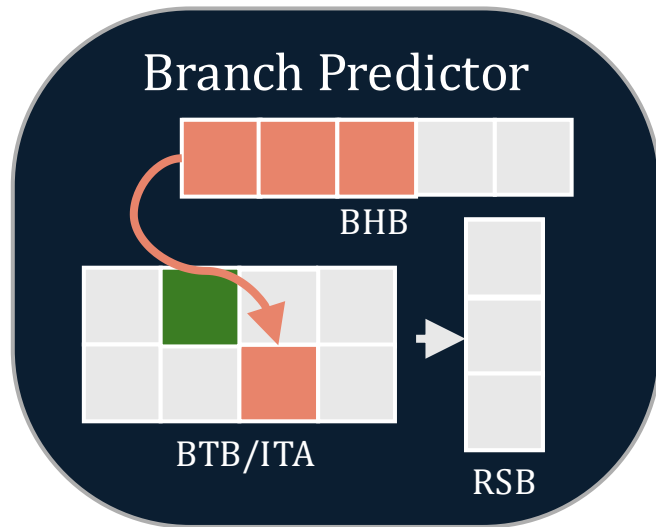
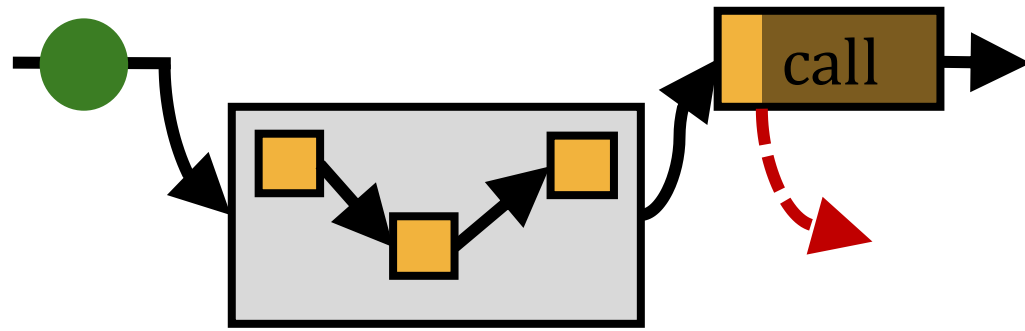


Interrupts mistrain jmps/calls
by poisoning the BHB

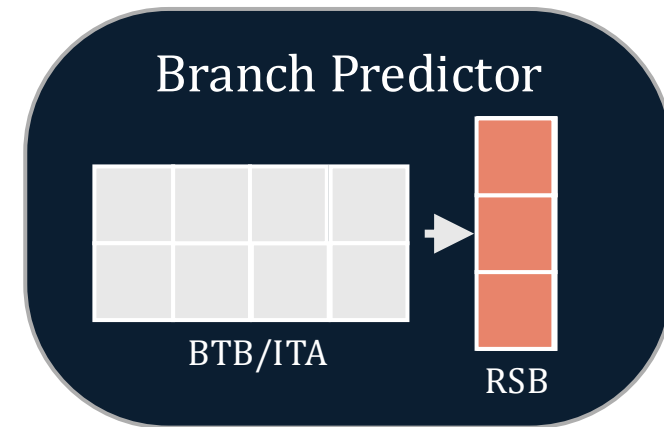
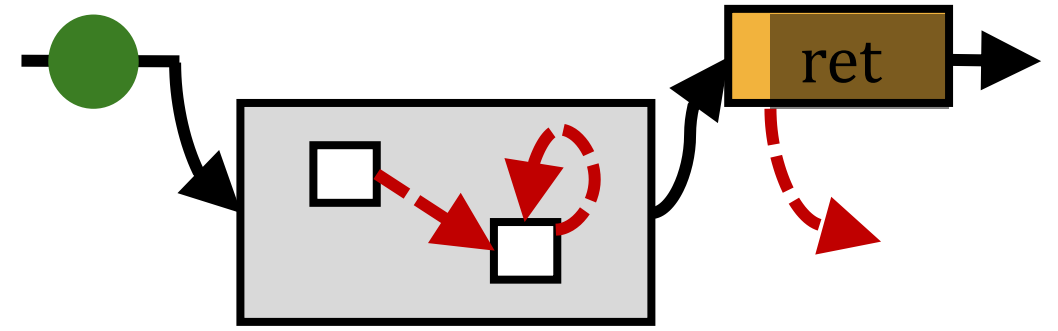


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

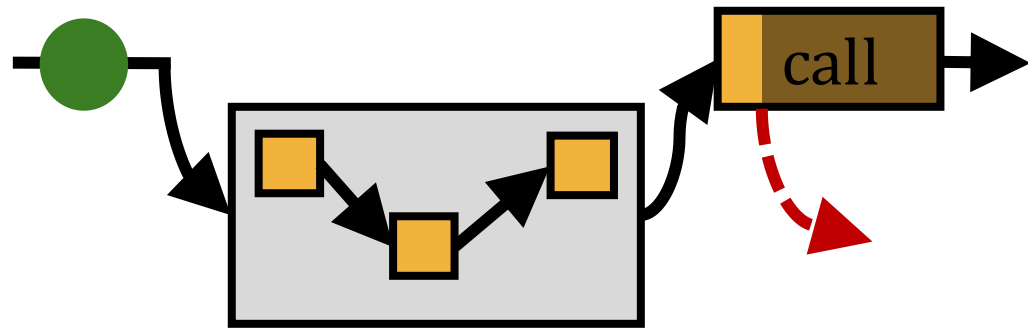


Interrupts mistrain jmps/calls
by poisoning the BHB

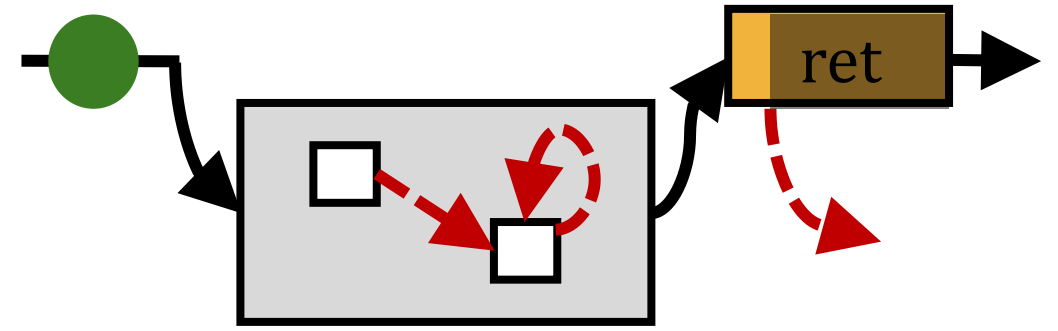


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

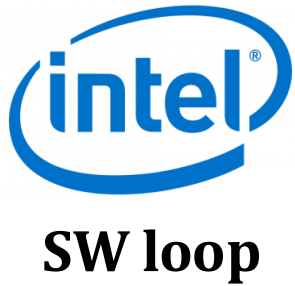
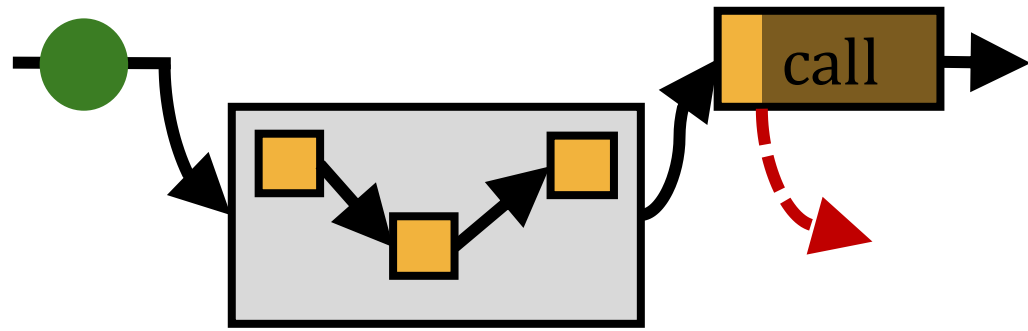


Interrupts mistrain jumps/calls
by poisoning the BHB

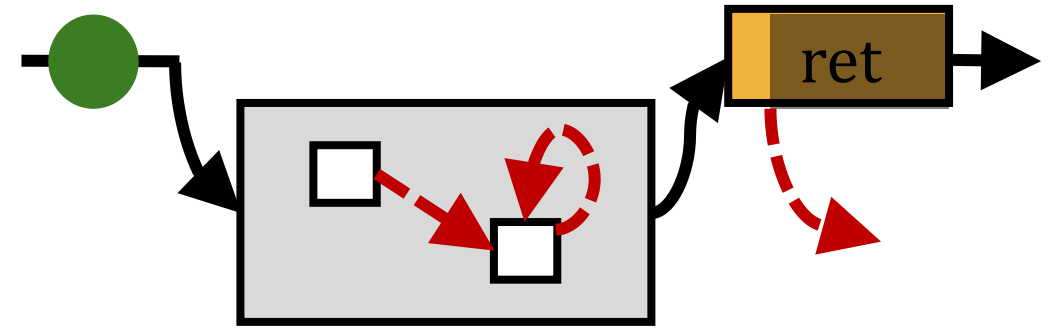


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

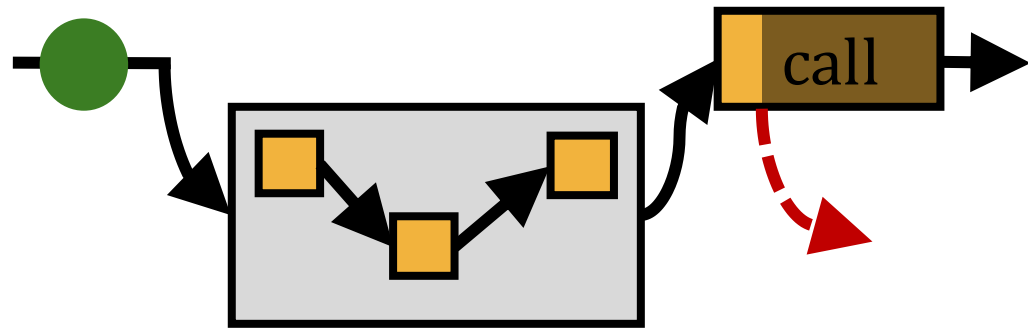


Interrupts mistrain jmps/calls
by poisoning the BHB

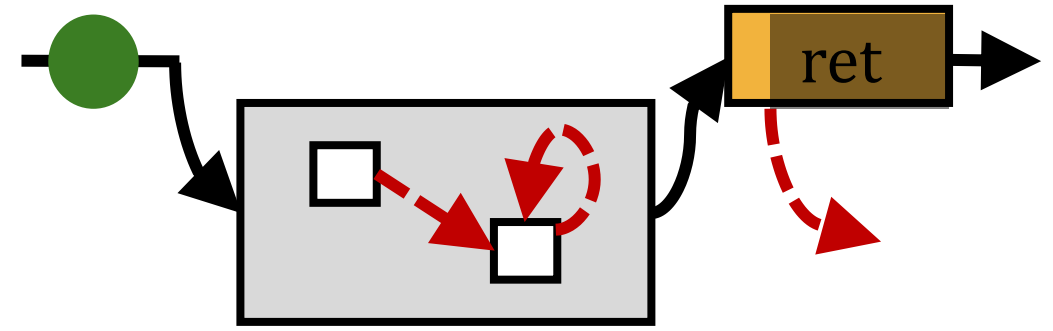


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

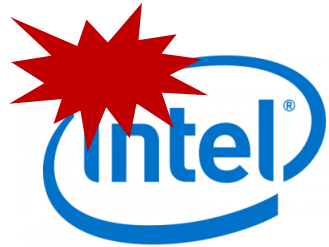
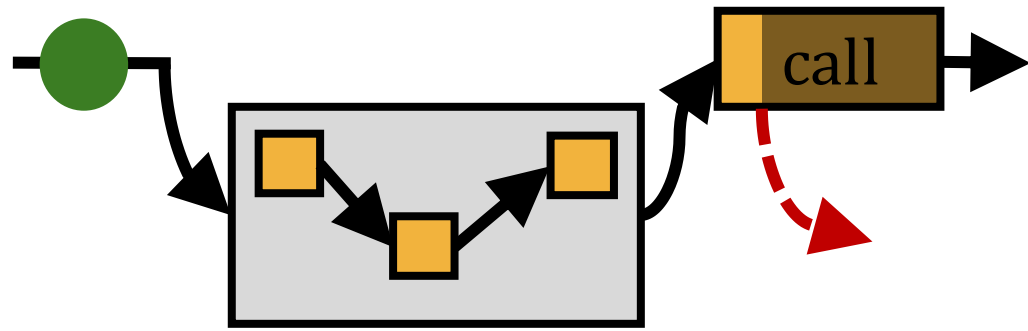


Interrupts mistrain jmps/calls
by poisoning the BHB

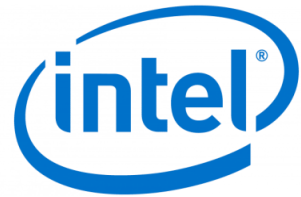


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

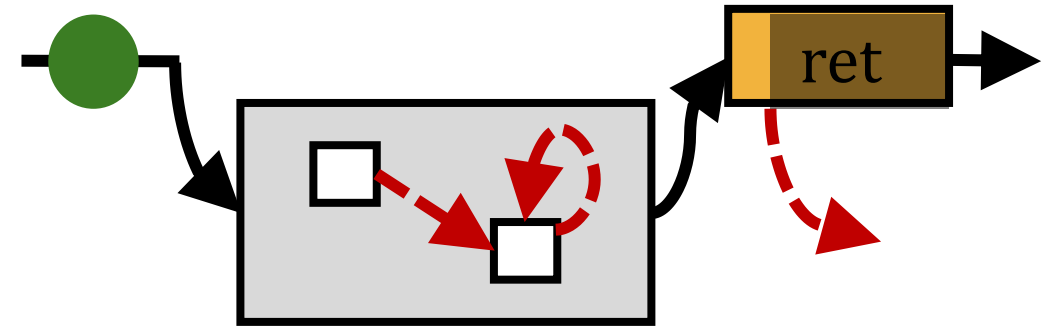


SW loop



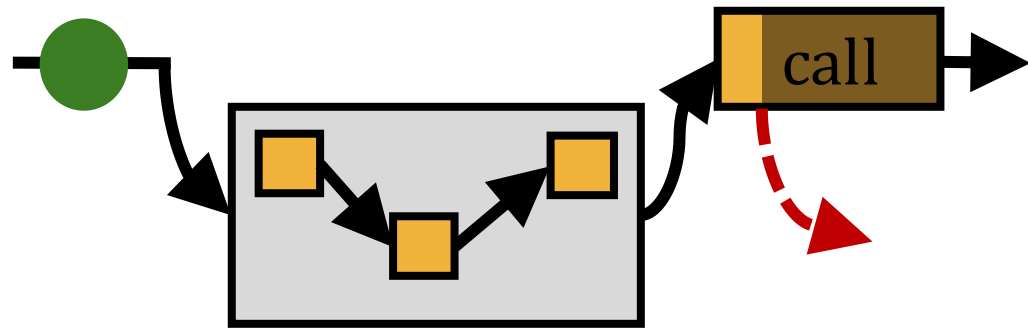
BHI_DIS_S

Interrupts mistrain jmps/calls
by poisoning the BHB

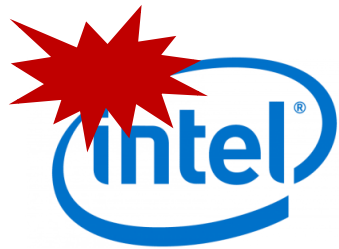


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

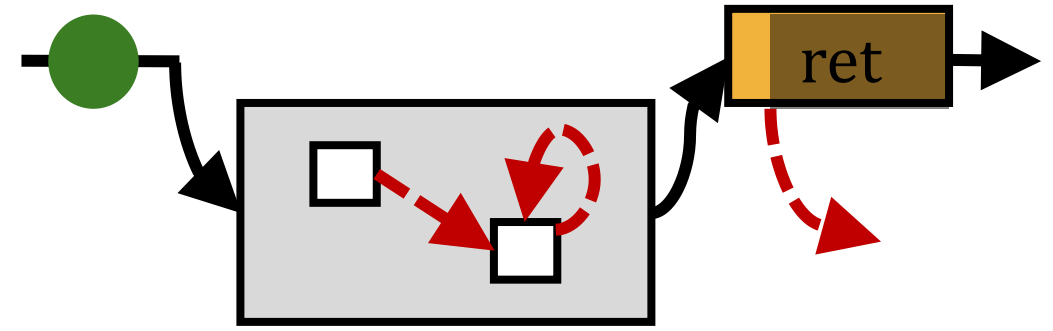


SW loop



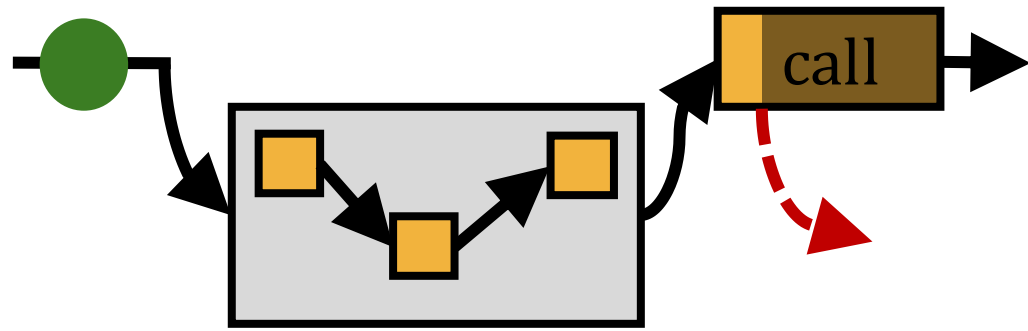
BHI_DIS_S

Interrupts mistrain jmps/calls
by poisoning the BHB

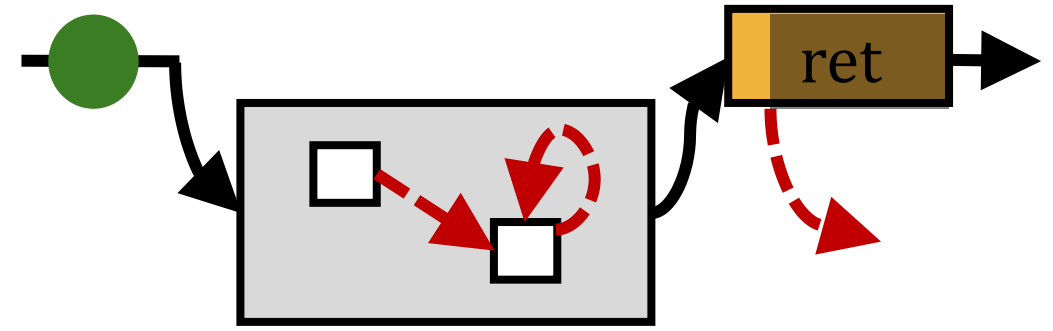


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

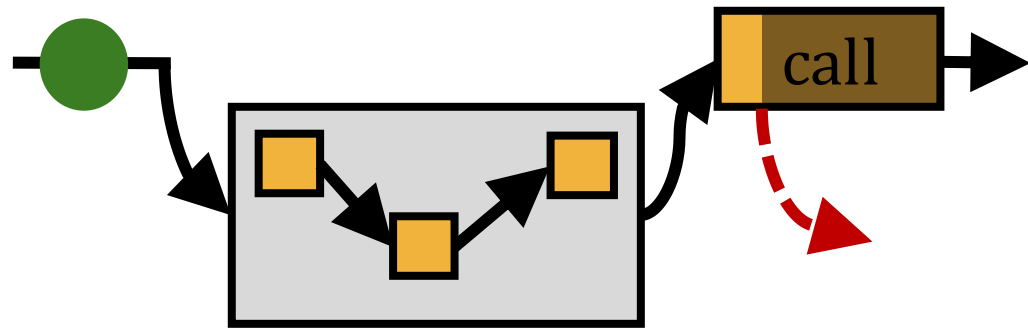


Interrupts mistrain jmps/calls
by poisoning the BHB

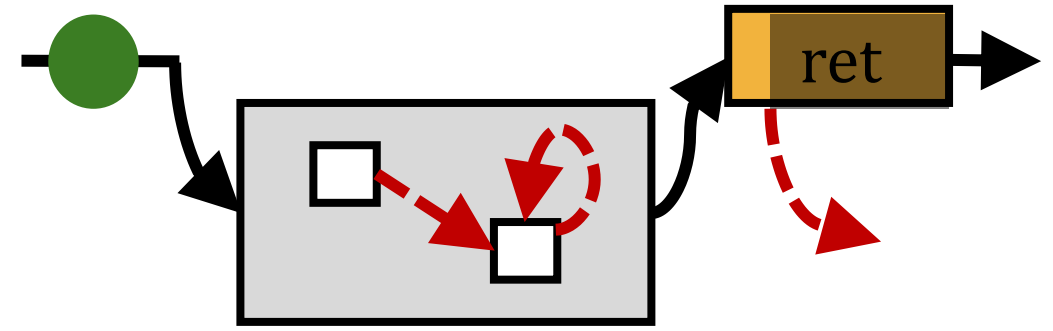


Interrupts mistrain returns
by poisoning the RSB transiently

Poisoning using existing techniques

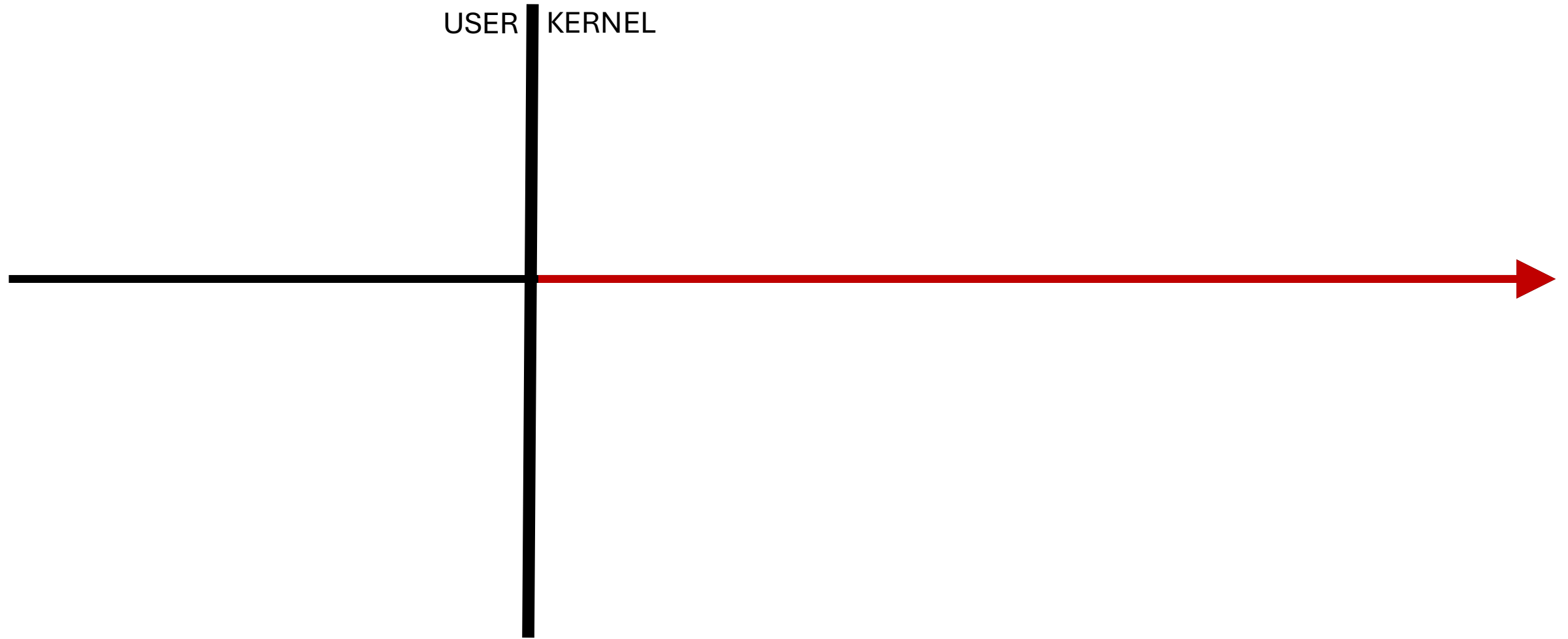


Interrupts mistrain jmps/calls
by poisoning the BHB

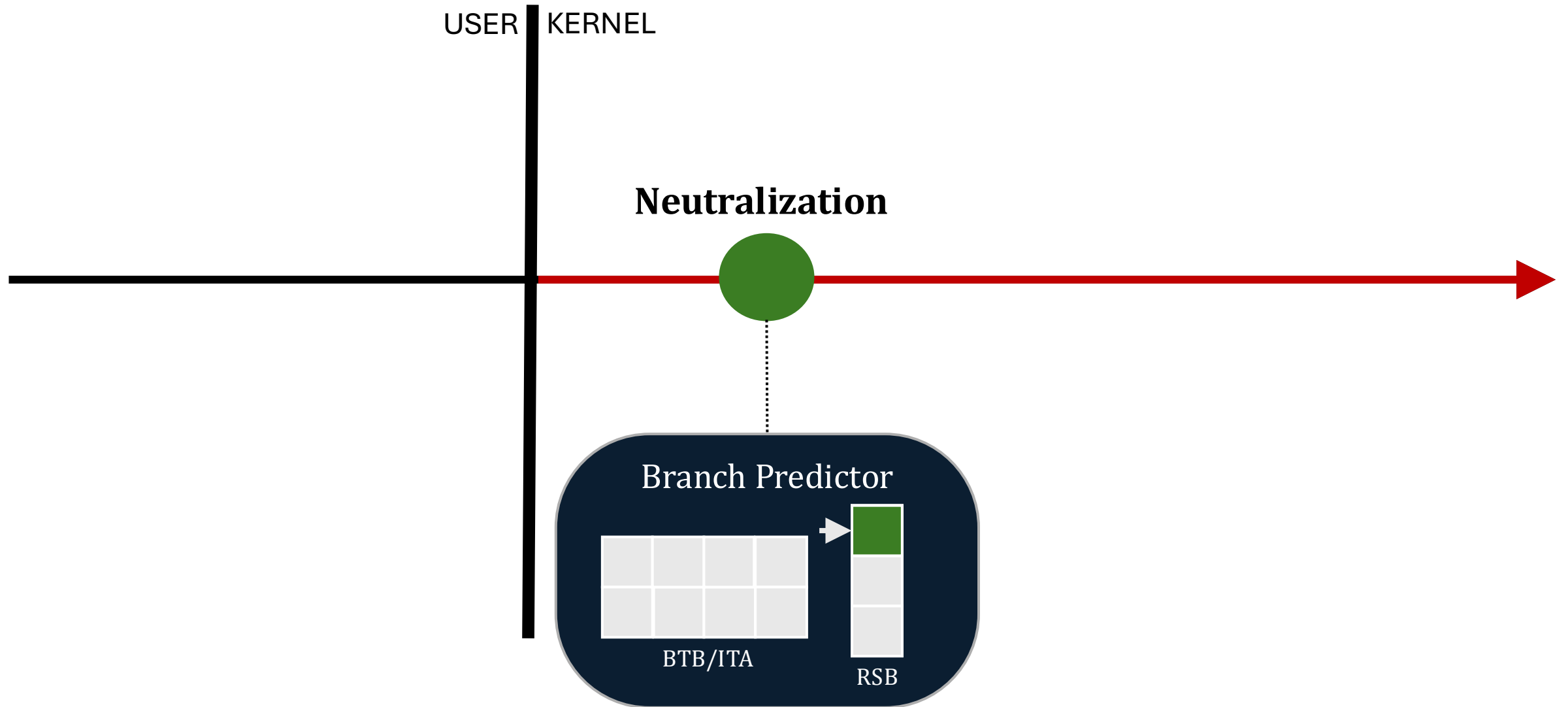


Interrupts mistrain returns
by poisoning the RSB transiently

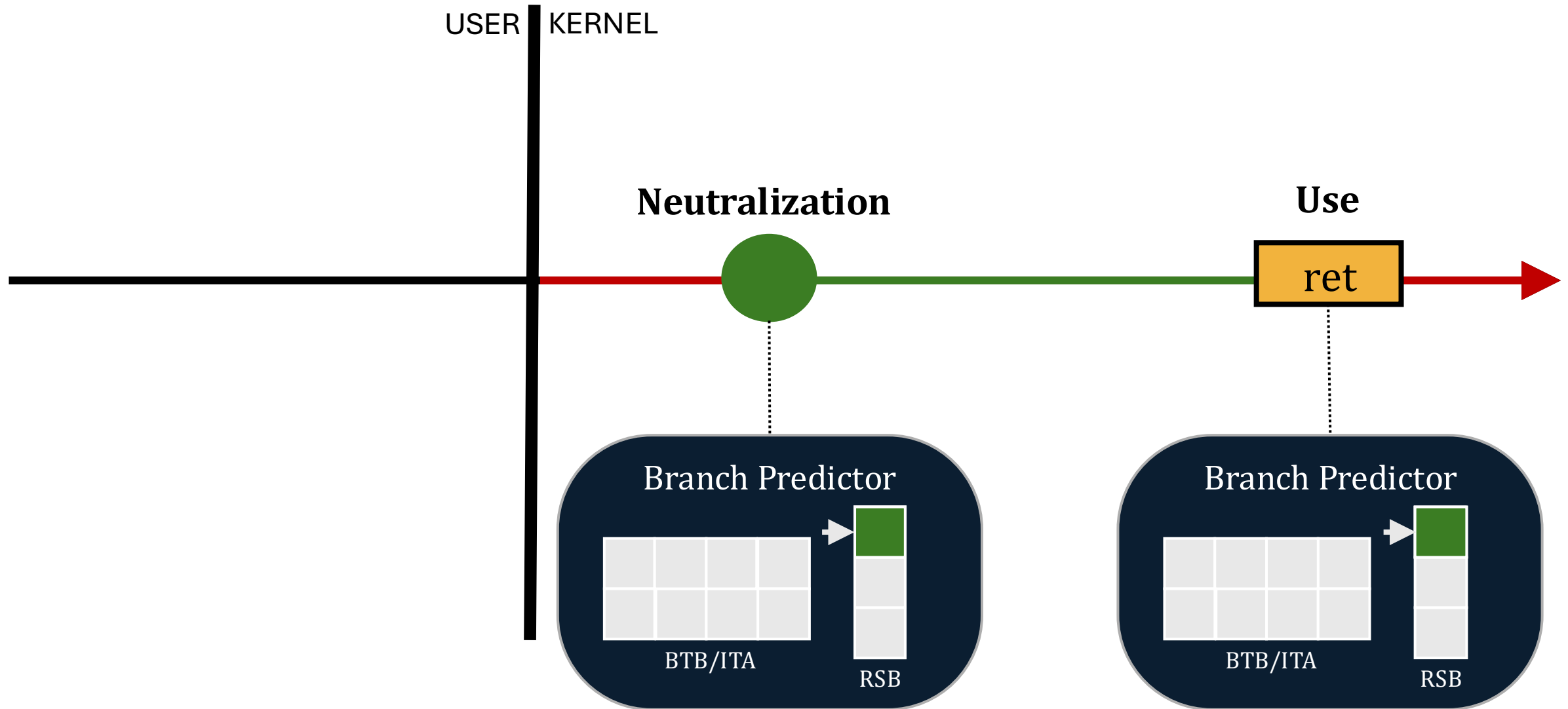
INTERRUPT INJECTION breaks SafeRET



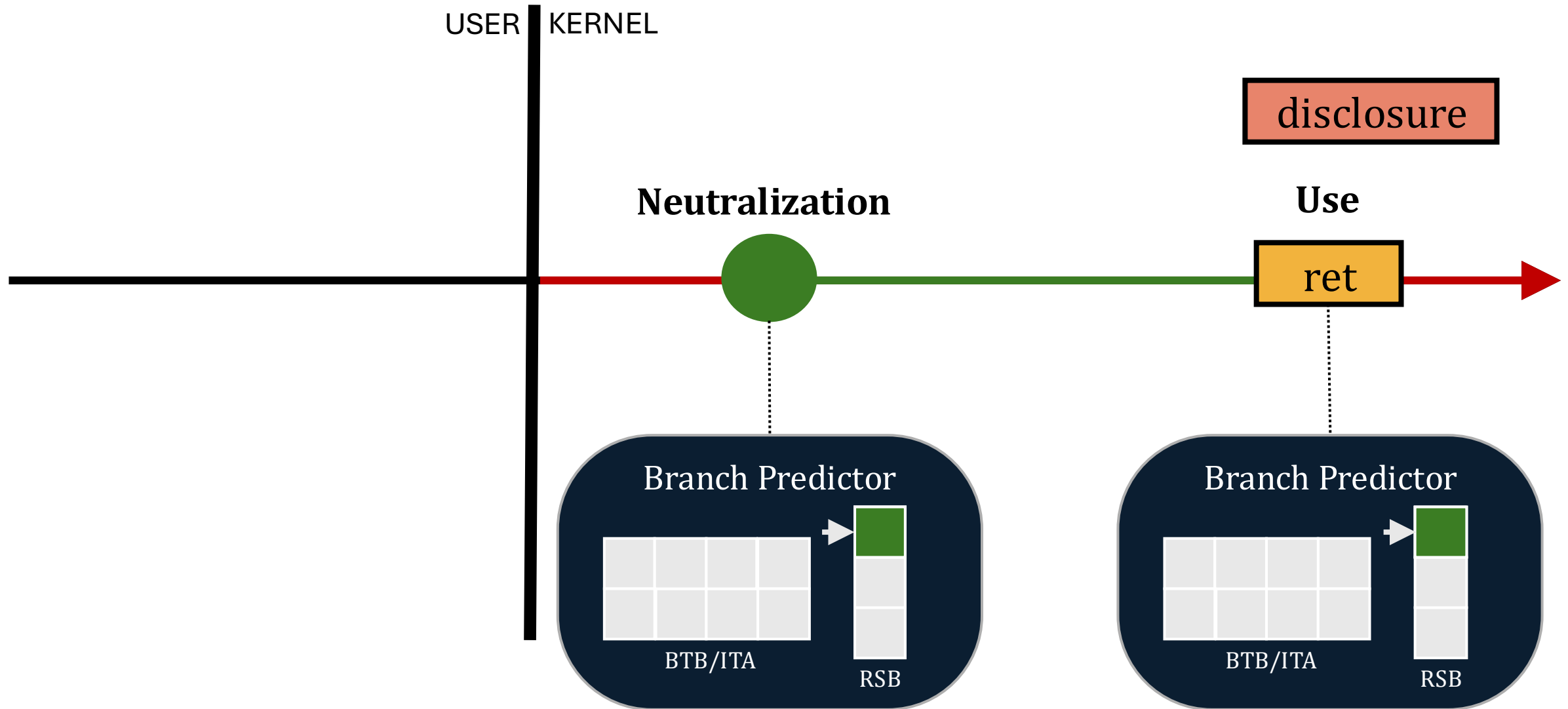
INTERRUPT INJECTION breaks SafeRET



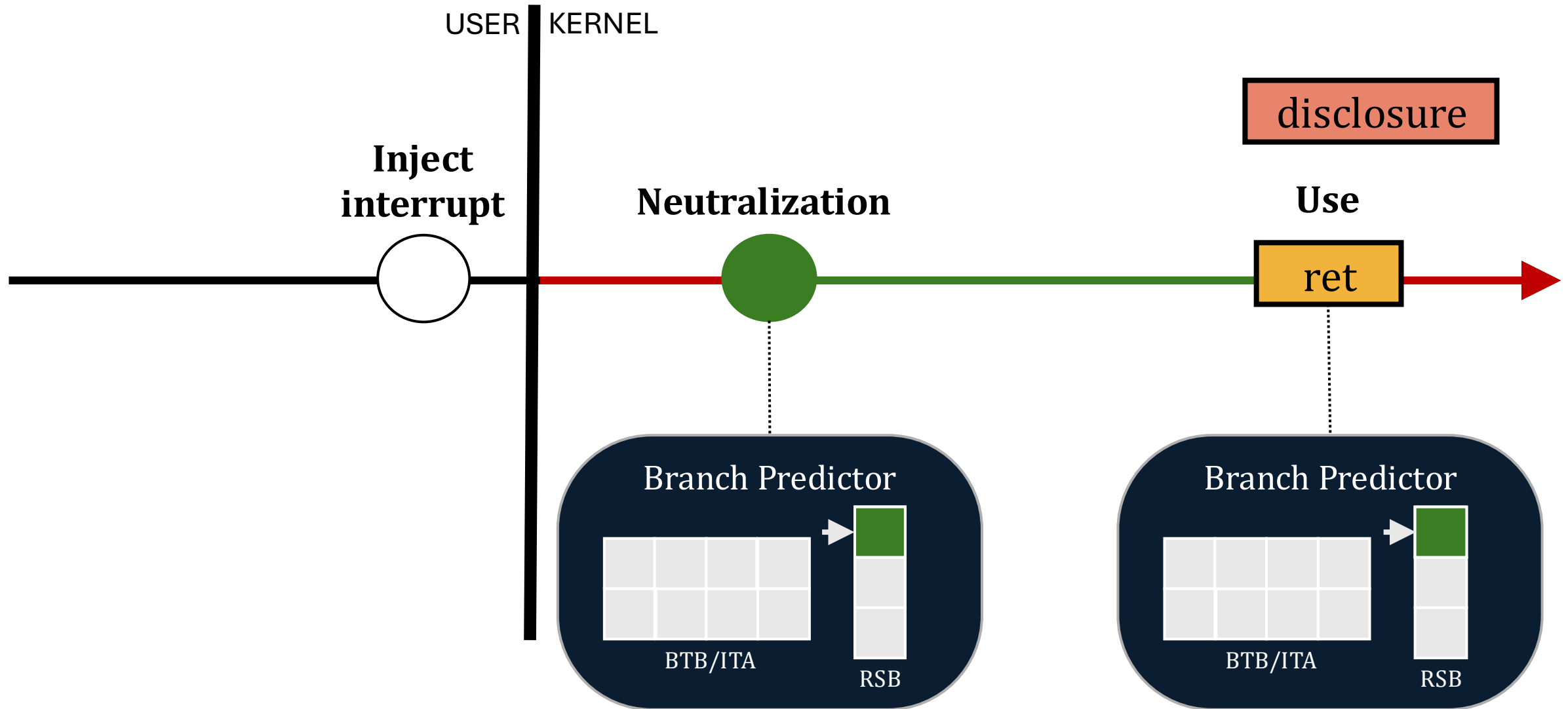
INTERRUPT INJECTION breaks SafeRET



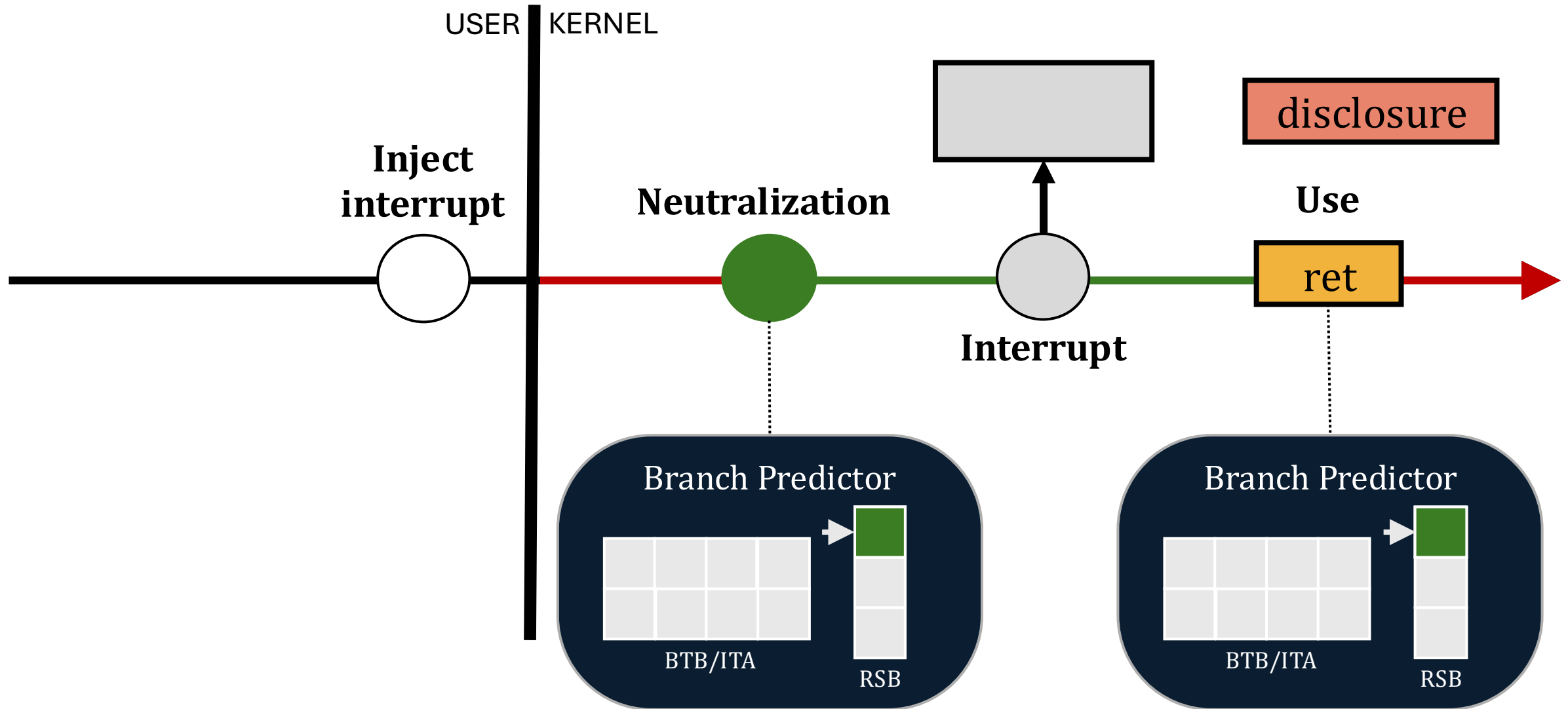
INTERRUPT INJECTION breaks SafeRET



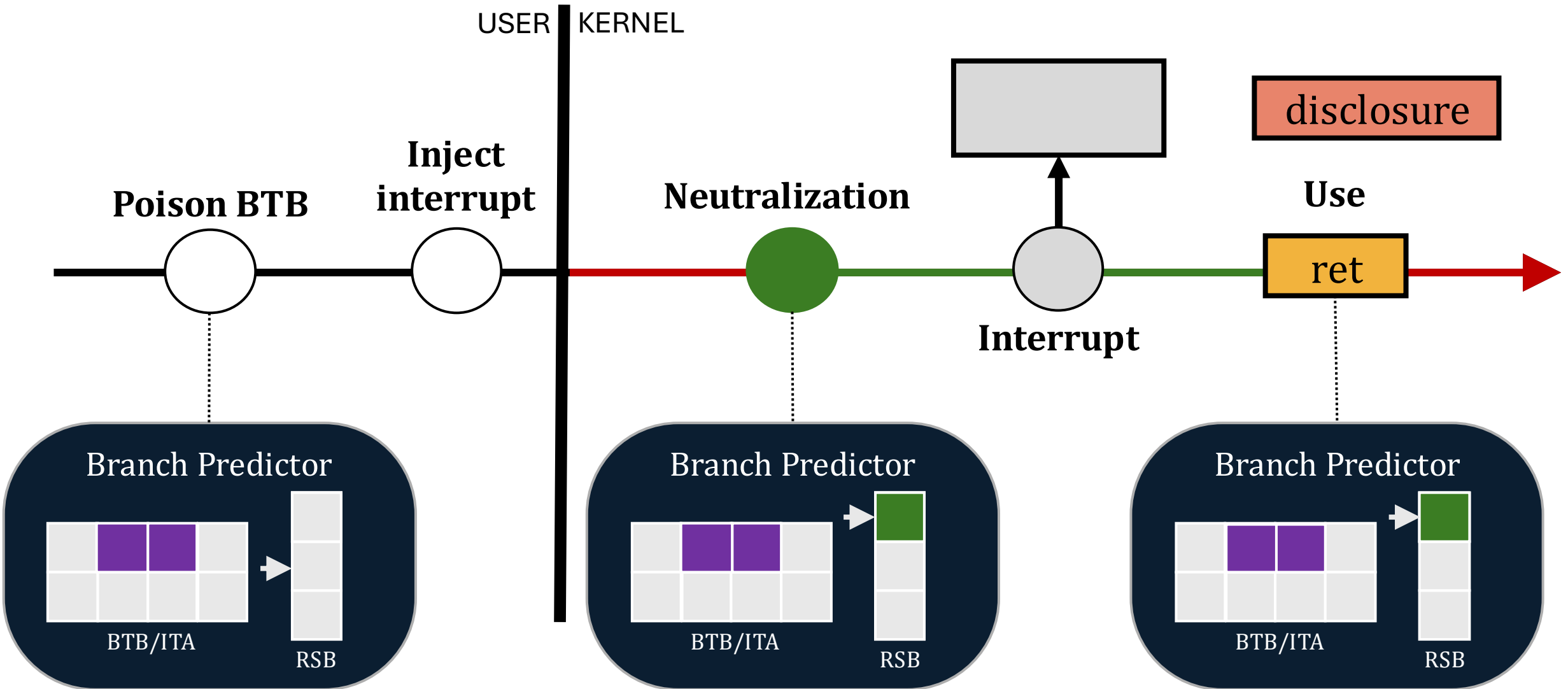
INTERRUPT INJECTION breaks SafeRET



INTERRUPT INJECTION breaks SafeRET

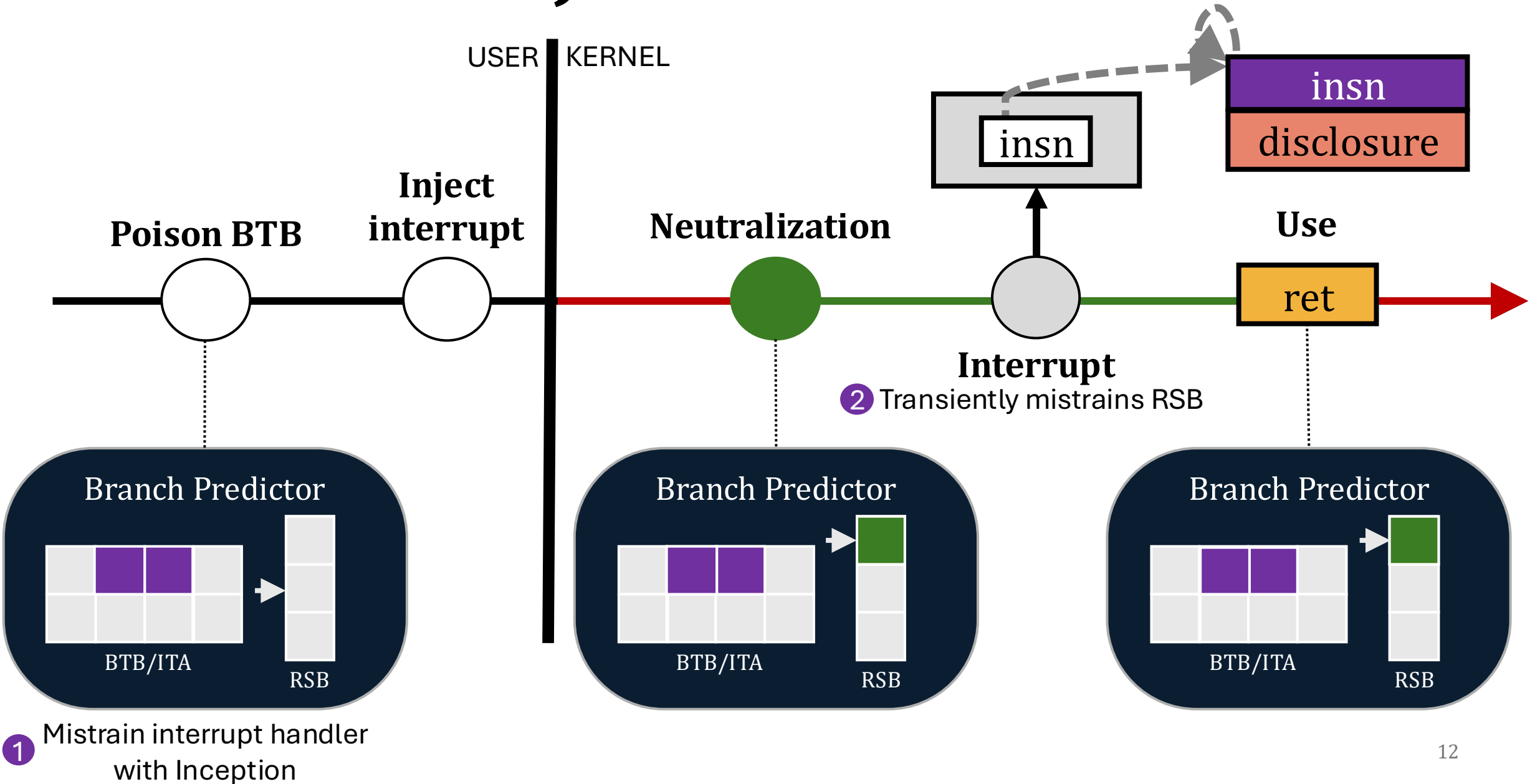


INTERRUPT INJECTION breaks SafeRET

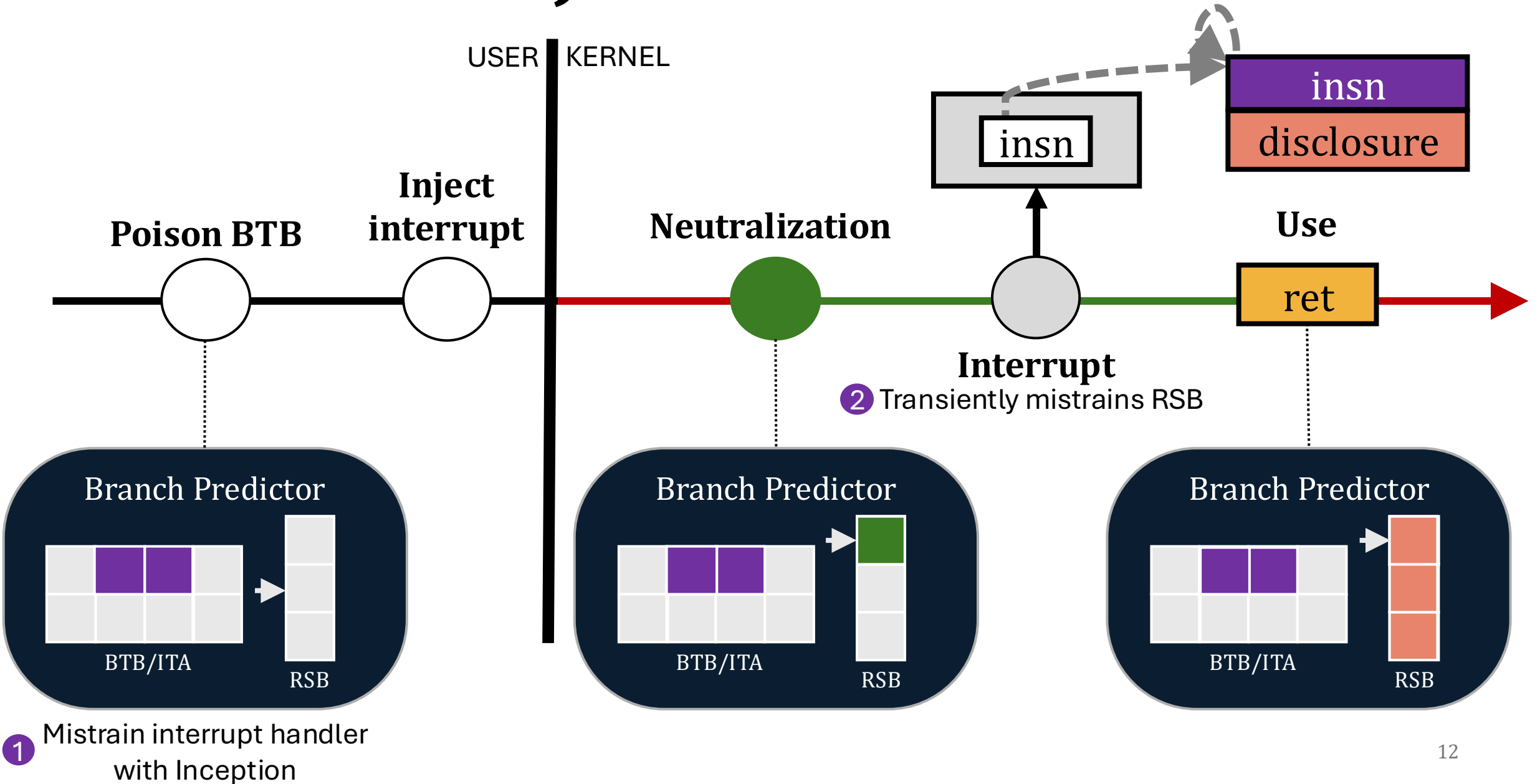


1 Mistrain interrupt handler with Inception

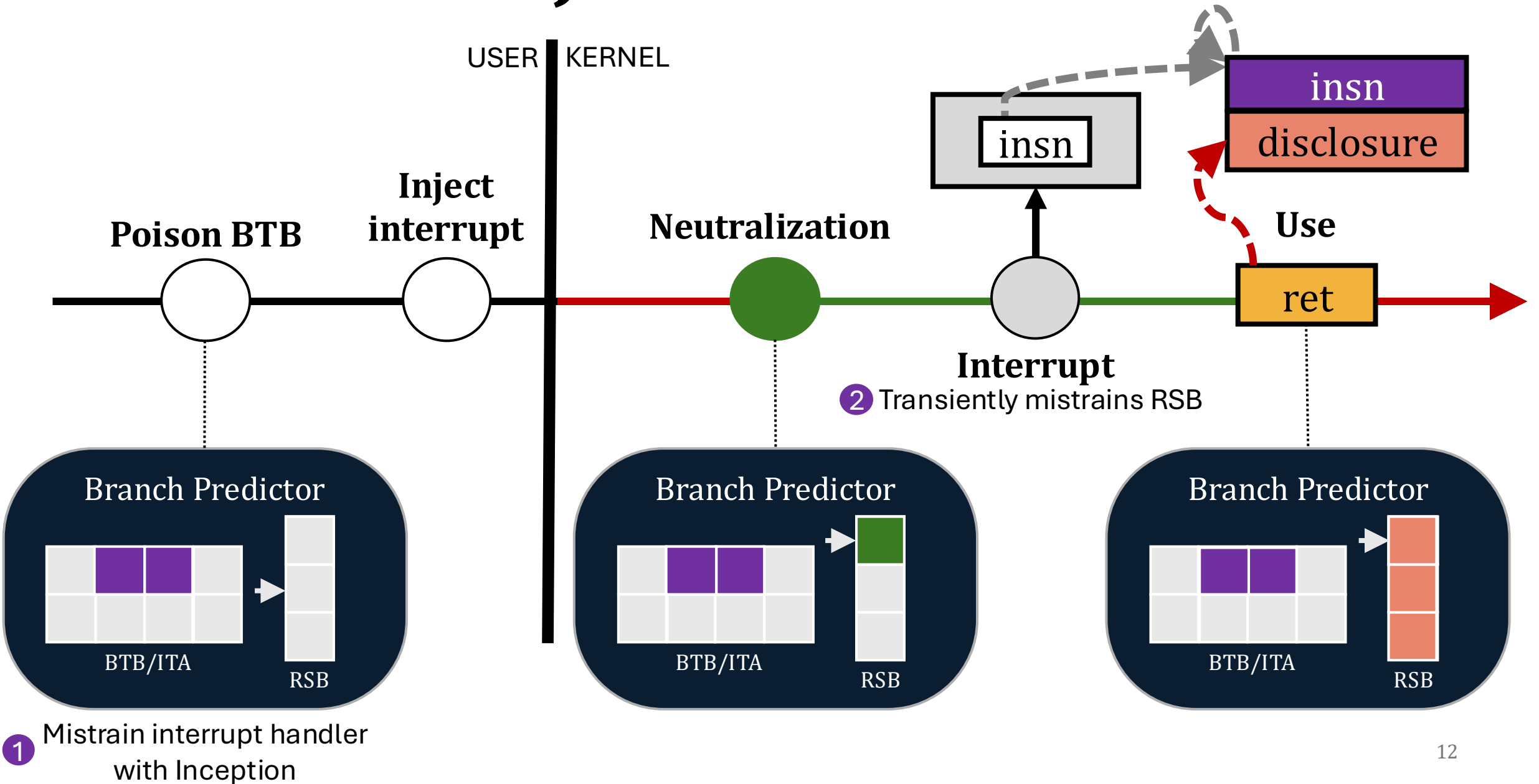
INTERRUPT INJECTION breaks SafeRET



INTERRUPT INJECTION breaks SafeRET



INTERRUPT INJECTION breaks SafeRET



End-to-end attack on AMD Zen 2



End-to-end attack on AMD Zen 2



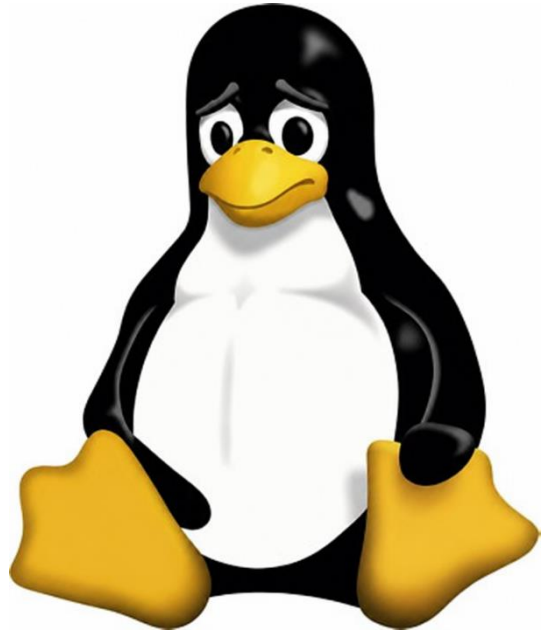
- Stock Linux (6.14.0-37-generic)

End-to-end attack on AMD Zen 2



- Stock Linux (6.14.0-37-generic)
- All default mitigations enabled (SafeRET, KASLR, etc.)

End-to-end attack on AMD Zen 2



- Stock Linux (6.14.0-37-generic)
- All default mitigations enabled (SafeRET, KASLR, etc.)
- No inserted modules, only unprivileged user execution

End-to-end attack on AMD Zen 2

End-to-end attack on AMD Zen 2



Break
KASLR

548s

End-to-end attack on AMD Zen 2

Break
KASLR

548s

Locate
/etc/shadow

18m

End-to-end attack on AMD Zen 2

Break
KASLR

548s

Locate
/etc/shadow

18m

Leak

5.47 bytes/s

End-to-end attack on AMD Zen 2

Break
KASLR

548s

Locate
/etc/shadow

18m

Leak

5.47 bytes/s

```
danieltrujillo — daniel@AMD-Zen-2: ~/interrupt_injection_demo — ssh - ssh zen2 — 180x50
root:$y$j9T$7j$kpI/9QPCZwINqScIHKu1$VbnyVr7DWp2EkGUB1DAWXXcys5Kuc1TERFM.4ZCzoX9:20646:0:99999:7:::
daemon*:20134:0:99999:7:::
bin*:20134:0:99999:7:::
sys*:20134:0:99999:7:::
sync*:20134:0:99999:7:::
games*:20134:0:99999:7:::
man*:2013 :0:99999:7:::
lp*:20134:0:99999:7:::
mail*:20134:0:99999:7:::
news*:20134:0:99999:7:::
uucp*:20134:0:99999:7:::
proxy*:20134:0:99999:7:::
www-data*:20134:0:99999:7:::
backup*:20134:0:99999:70:::
list@*:20134:0:99999:7:::
irc*:20134:0:99999:7:::
_apt*:20134:0:99999:7:::
nobody*:20134:0:99999:7:::
systemd-network:!:0:20134:0:99999:7:::
systemd-timesync:!:*:20134:0:99999:7:::
dxcpcd:!:20134:0:99999:7:::
messagebus:!:20134:0:99999:7:::
syslog:!:20134:0:99999:7:::
```

Mitigations

Mitigations



Mitigations



- Kernel patch: detects interrupts during SafeRET

Mitigations

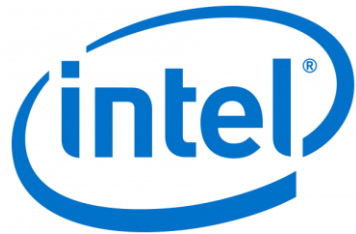


- Kernel patch: detects interrupts during SafeRET
- Emulates rest of SafeRET if interrupt is detected

Mitigations



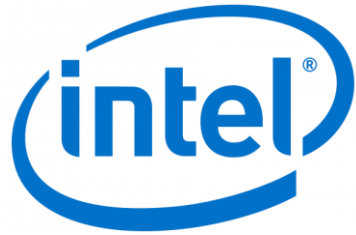
- Kernel patch: detects interrupts during SafeRET
- Emulates rest of SafeRET if interrupt is detected



Mitigations



- Kernel patch: detects interrupts during SafeRET
- Emulates rest of SafeRET if interrupt is detected

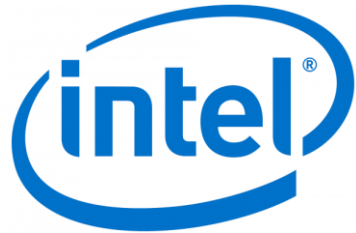


- Acknowledges underlying behavior

Mitigations



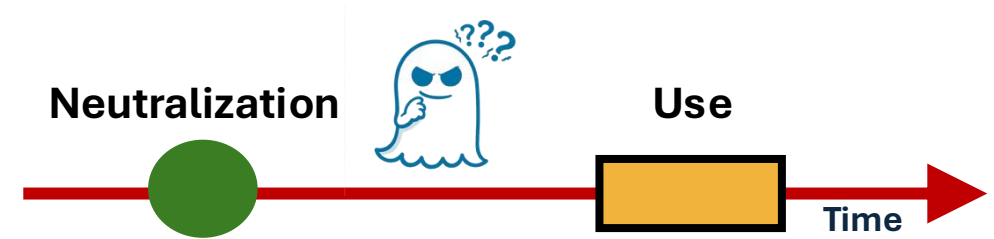
- Kernel patch: detects interrupts during SafeRET
- Emulates rest of SafeRET if interrupt is detected



- Acknowledges underlying behavior
- Exploitability depends on many factors

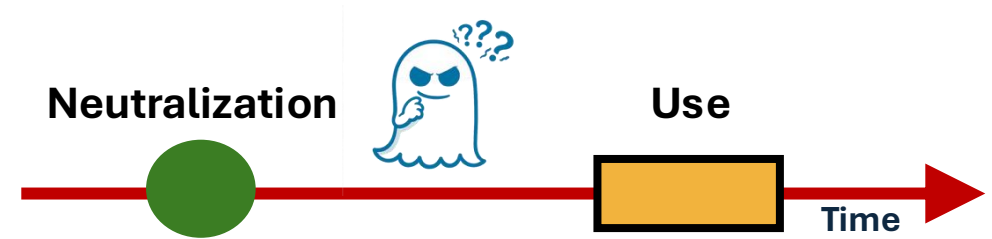
Conclusion

Conclusion

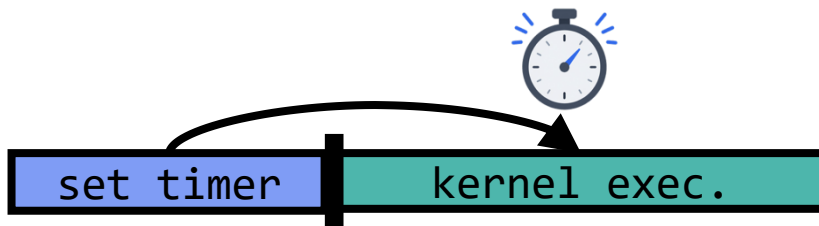


- Time-of-Neutralization to Time-of-Use (TONTOU) windows exist

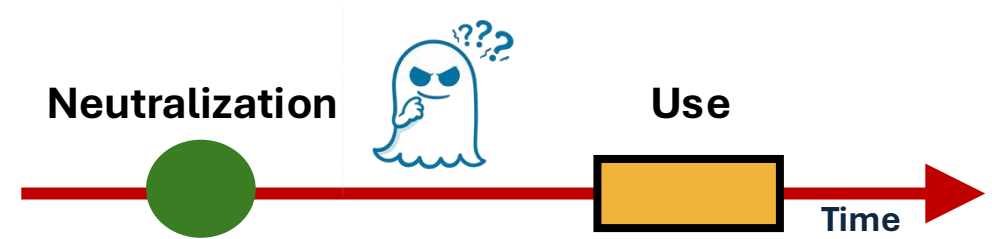
Conclusion



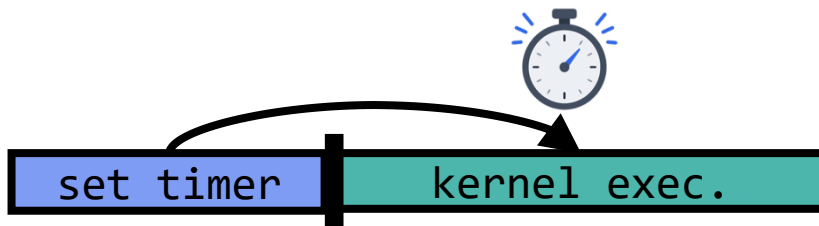
- Time-of-Neutralization to Time-of-Use (TONTOU) windows exist
- Attackers can redirect during the TONTOU window through interrupts



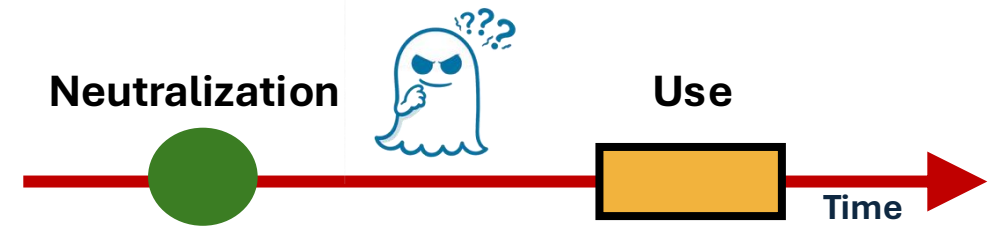
Conclusion



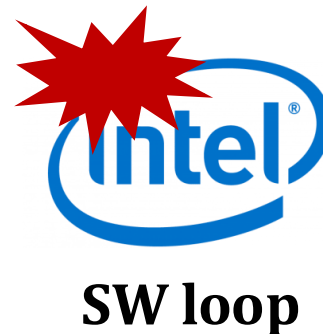
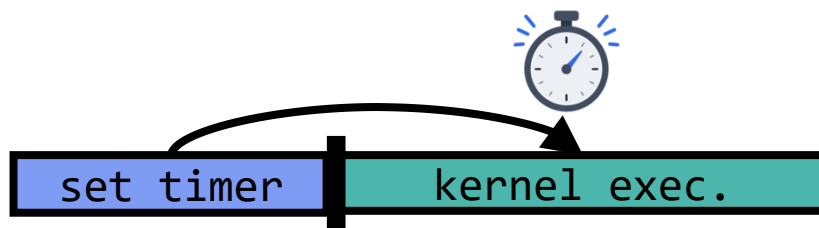
- Time-of-Neutralization to Time-of-Use (TONTOU) windows exist
- Attackers can redirect during the TONTOU window through interrupts
- Predictions for all indirect branches can be poisoned by interrupts



Conclusion



- Time-of-Neutralization to Time-of-Use (TONTOU) windows exist
- Attackers can redirect during the TONTOU window through interrupts
- Predictions for all indirect branches can be poisoned by interrupts
- INTERRUPT INJECTION breaks the latest Spectre v2 mitigations



Thank you! Questions?



danieltr@mit.edu



<https://github.com/MATCHA-MIT/TONTOU-Interrupt-Injection>

