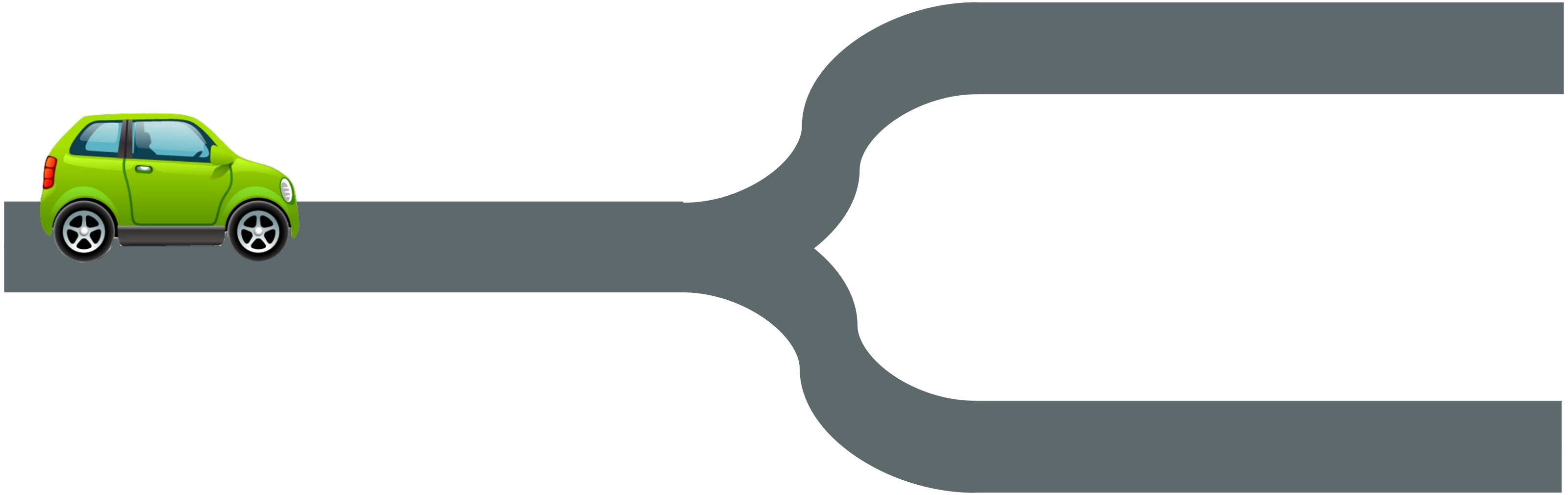
Ghosting ASLR

A Spectre Extension

by Bryant Curto

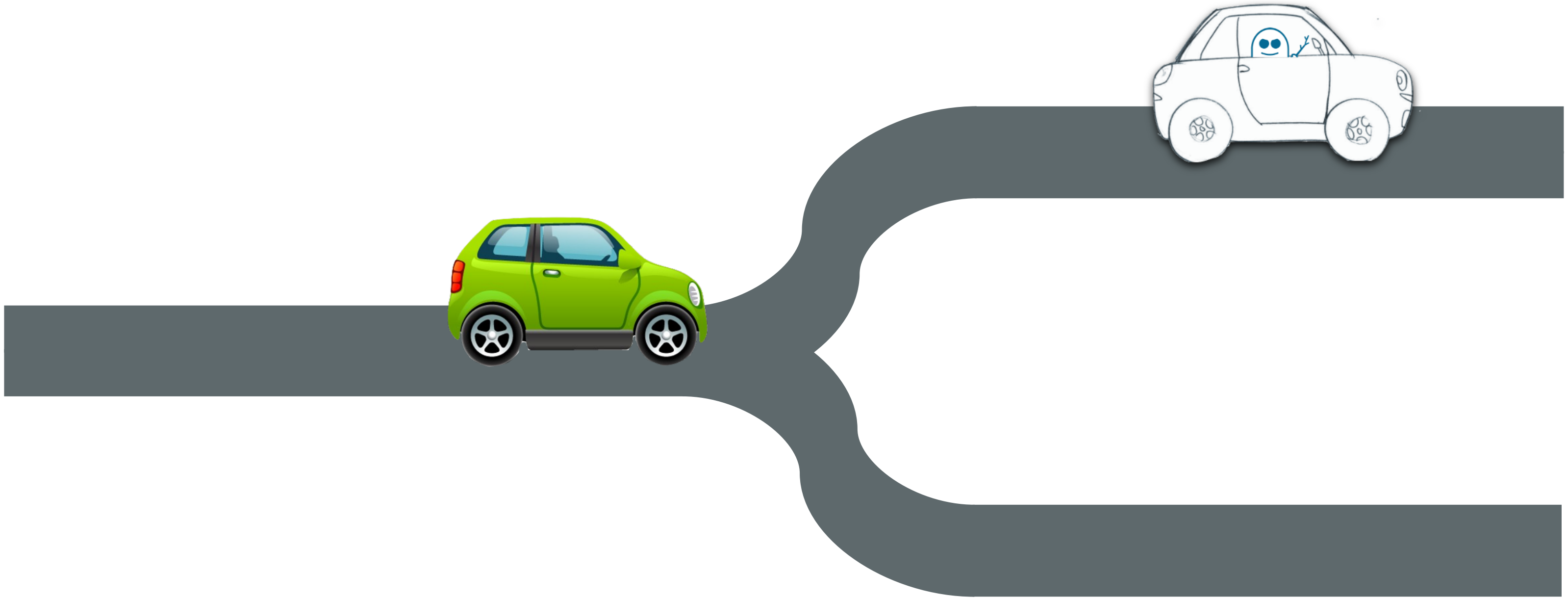
Background Information

Speculative Execution: Analogy

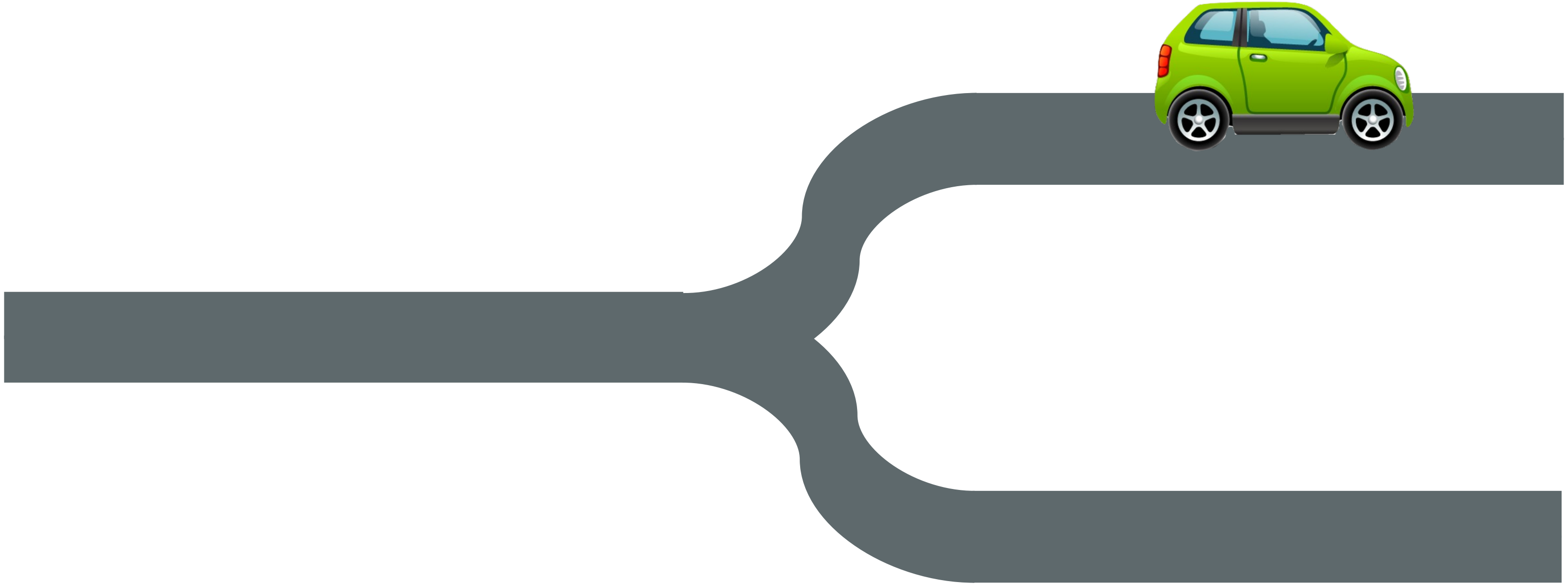


Speculative Execution: Analogy

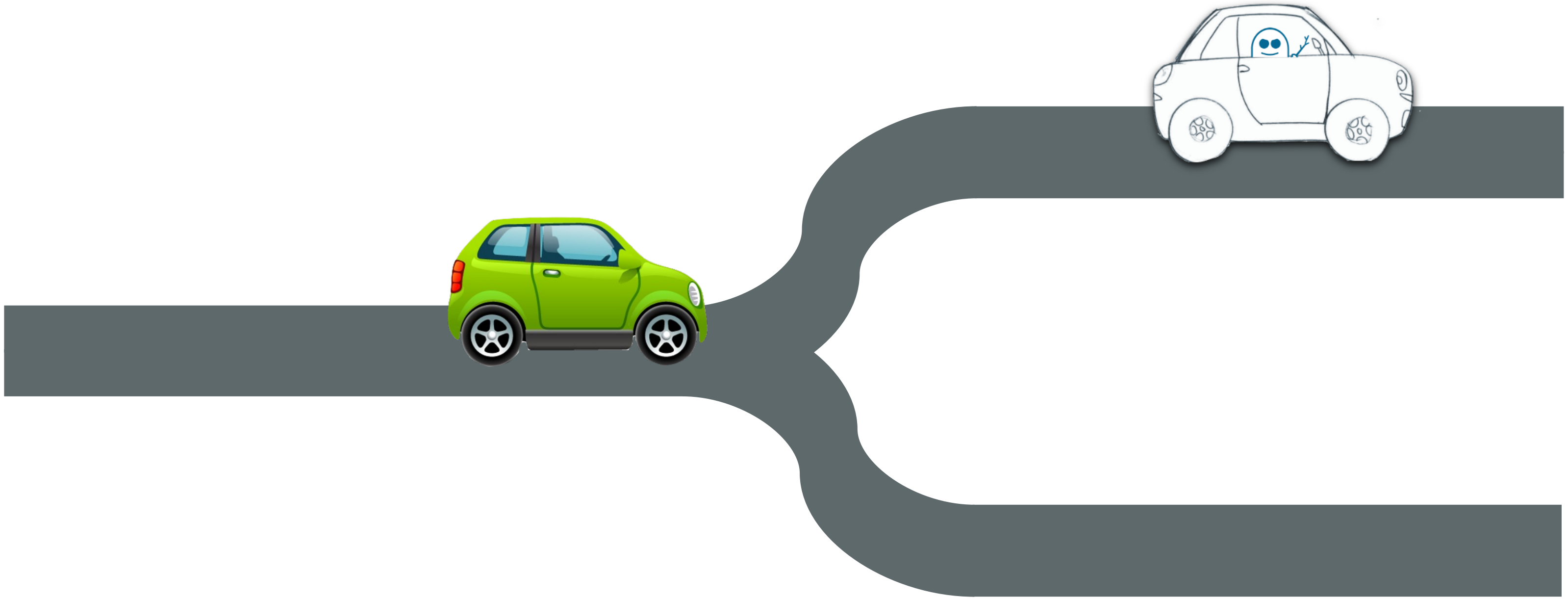
Speculative Execution: Analogy



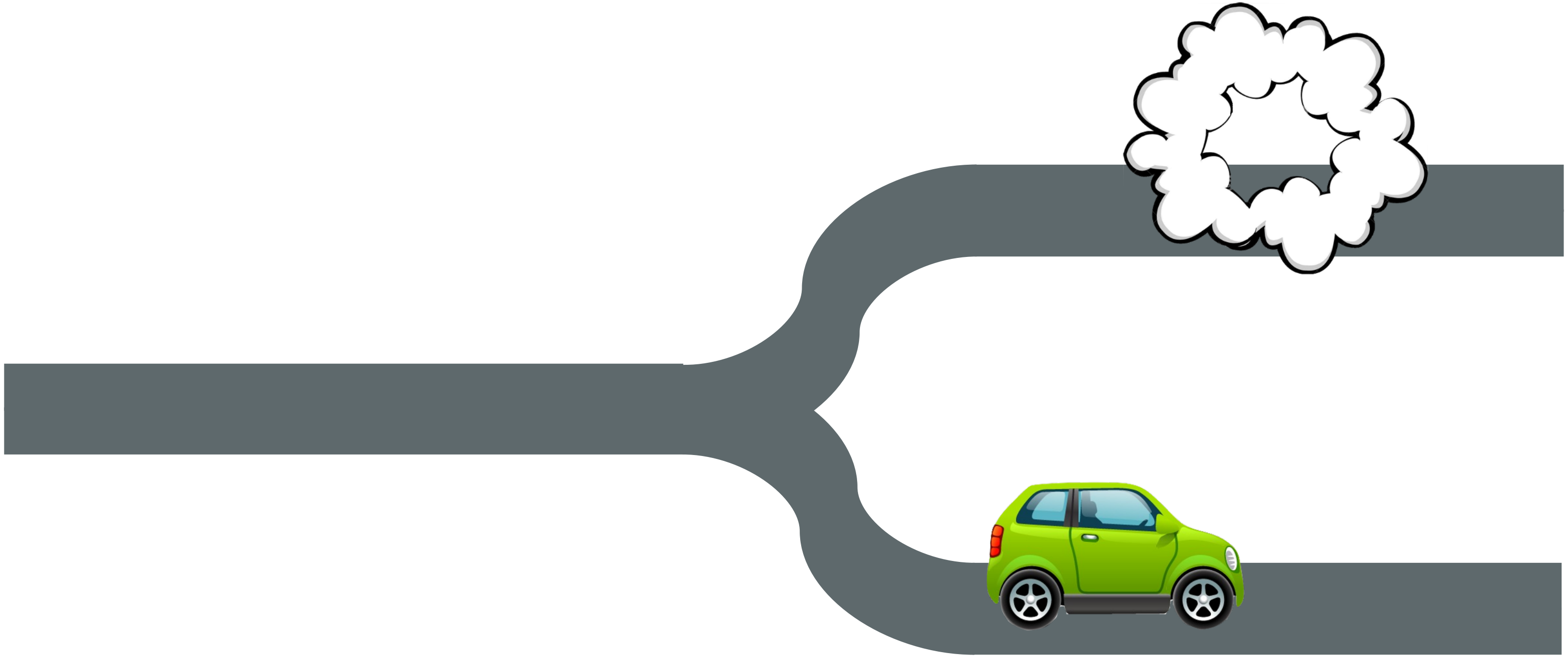
Speculative Execution: Analogy



Speculative Execution: Analogy



Speculative Execution: Analogy



Speculative Execution: Technical

- Modern processors executes multiple instructions at once.
- What happens when processor must complete instruction it's processing to determine the next instruction to start process (conditional jump)?
- It guesses the next instruction.
- None of the results of the guessed instructions are saved (written back) until processor knows that the guess was correct.

```
if (some_condition) {
    // do something
} else {
    // do something else
}
...
```

```
cmp 0, some_condition
jne  IF_COND           # jump if cond is true
jmp ELSE_COND          # jump unconditionally

IF_COND:
        # do something
        jmp END
ELSE_COND:
        # do something else
END:
        ...
```

Spectre: Analogy

- Driving down the road, you are prevented from driving on certain streets by fences, and entering homes by locked doors.
- While in your ghost form, no such obstacles can stop you, allowing you to drive on any street and enter any home.
- You can enter any home and see (but not touch) anything that you want.

Spectre: Analogy

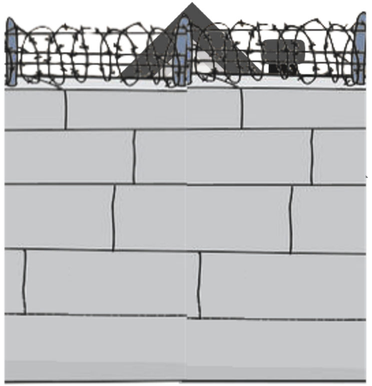
- Driving down the road, you are prevented from driving on certain streets by fences, and entering homes by locked doors.
- While in your ghost form, no such obstacles can stop you, allowing you to drive on any street and enter any home.
- You can enter any home and see (but not touch) anything that you want.

Spectre: Technical

- A vulnerability found in many modern processors employing speculative execution that allows an attacker to read private data
- It is caused by the fact that speculative execution causes measurable side effects

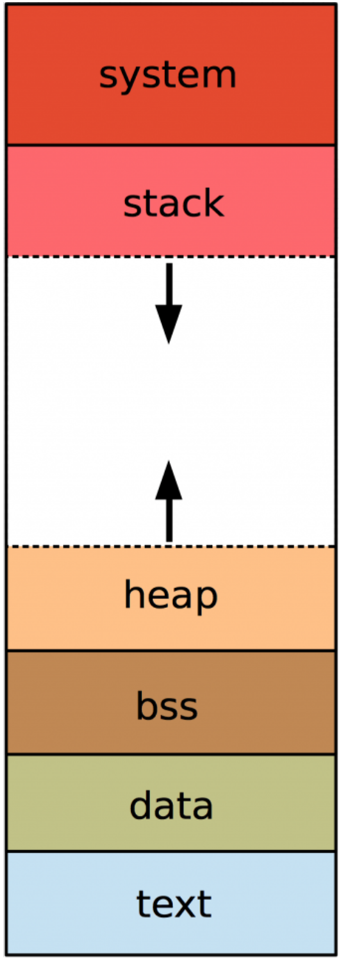
ASLR: Analogy

- There are houses whose locations must be kept secret.
- Streets leading to these buildings are blocked off, and, periodically, they are moved to a new secret location.
- Simply learning of these buildings' locations can enable one to access them.
- One can, not only look within the buildings, but also take and leave things behind.



ASLR: Technical

- Address Space Layout Randomization
- Prevent an attacker from knowing memory layout, and using that knowledge for an attack



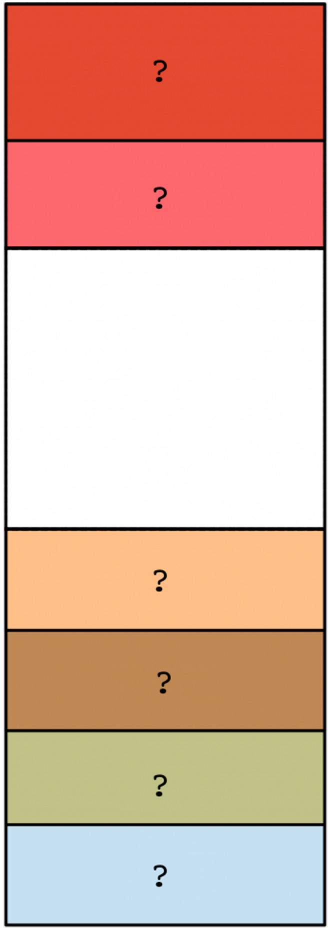
My Research

Analogy

- Become ghost to bypass ASLR’s obstacles and find the locations of the buildings it is trying to hide
- Can then launch another attack to get full access to these buildings (ie. look inside, take things, and leave things behind)

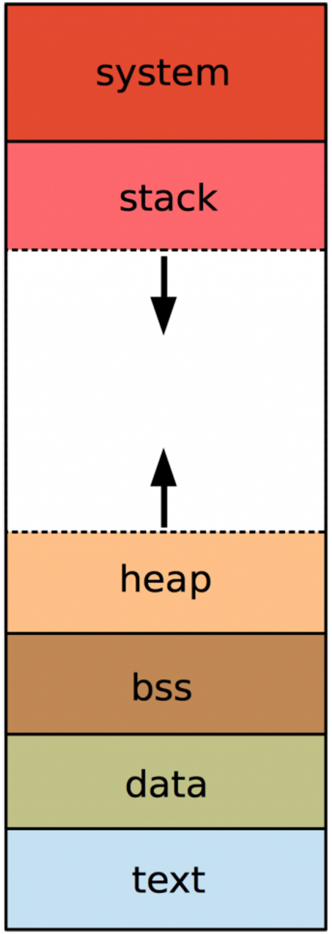
Technical

- Use Spectre techniques to find shuffled data (not read it)
    - Then derive the name and job of each piece of data
    - ASLR is therefore defeated
- We can now launch another attack to execute arbitrary code, read/write memory, and more.



Technical

- Use Spectre techniques to find shuffled data (not read it)
    - Then derive the name and job of each piece of data
    - ASLR is therefore defeated
- We can now launch another attack to execute arbitrary code, read/write memory, and more.



Conclusion

- Demonstrated that Spectre can be used to defeat ASLR.

Future Work

- Turn proof-of-concept into robust demo
- Create mathematical model to predict reliability of the detection algorithm

Acknowledgements

I would like to acknowledge **Prof. Christopher Mitchell** and **Thomas Dickerson** for mentoring me as I performed this research, and for being awesome people.

References

- Evtyushkin, Dmitry, Dmitry Ponomarev, and Nael Abu-Ghazaleh. "Jump over ASLR: Attacking branch predictors to bypass ASLR." Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on. IEEE, 2016.

- Hennessy, John L., and David A. Patterson. Computer Architecture: A Quantitative Approach. 5th ed., Morgan Kaufmann Publ., 2012.

- Horn, Jann, and Project Zero. “Reading Privileged Memory with a Side-Channel.” Project Zero, 3 Jan. 2018, googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html.

- Hund, Ralf, Carsten Willems, and Thorsten Holz. "Practical timing side channel attacks against kernel space ASLR." Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.

- Kocher, Paul, et al. "Spectre Attacks: Exploiting Speculative Execution." arXiv preprint arXiv:1801.01203 (2018).

Spectre: Bonus

- To exploit this vulnerability, an attacker must:
    - Force the processor to speculatively execute a piece of code,
    - Be able to influence that code's execution so that the processor reads memory dependent on user input, and
    - Be able to detect the side effects caused by this speculative execution.