HoliSec

Holistic Approach to Improve Data Security

A Study of the Interplay Between Safety and Security Using Model-Implemented Fault Injection

(published at the 14th European Dependable Computing Conference, Iaşi, Romania)

Behrooz Sangchoolie, Peter Folkesson, Jonny Vinter

FFI conference: DEx

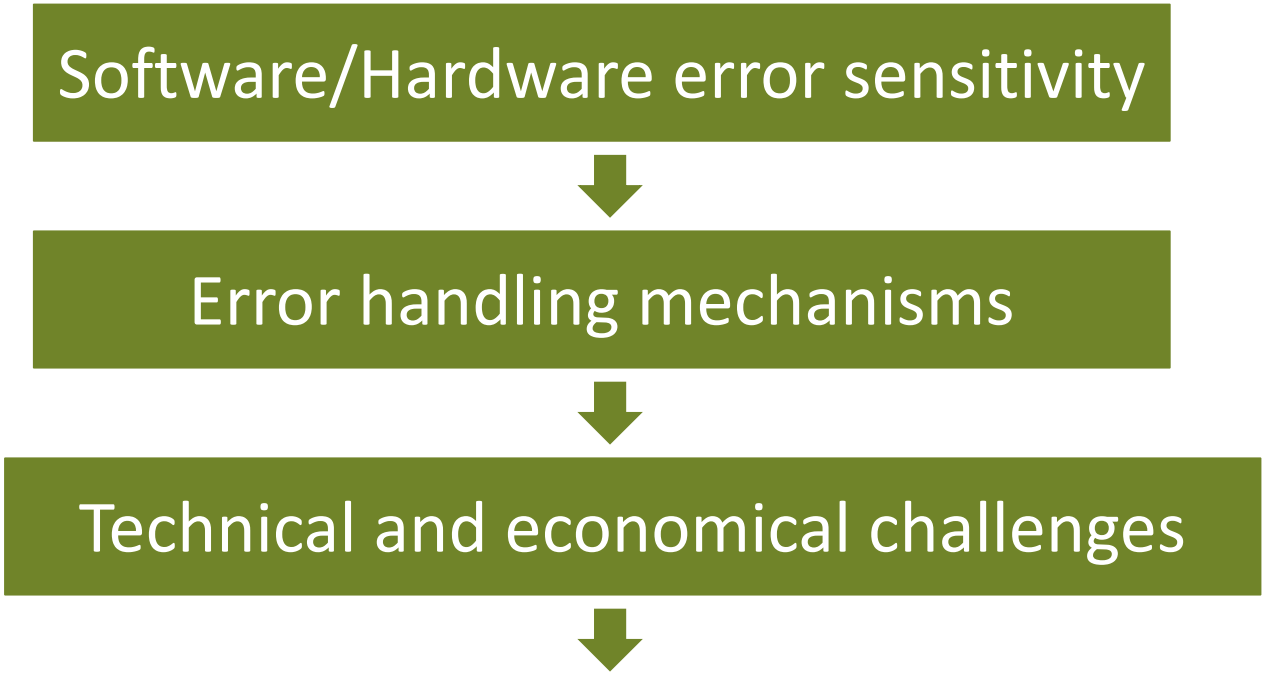
September 18, 2018

Background

Software/Hardware error sensitivity

- N. Seifert, D. Moyer, N. Leland and R. Hokinson, "**Historical trend in alpha-particle induced soft error rates of the Alpha™ microprocessor**," *2001 IEEE International Reliability Physics Symposium Proceedings. 39th Annual (Cat. No.00CH37167)*, Orlando, FL, 2001, pp. 259-265.
- T. Karnik and P. Hazucha, "**Characterization of soft errors caused by single event upsets in CMOS processes**," in *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 2, pp. 128-143, April-June 2004.
- S. Borkar, "**Designing reliable systems from unreliable components: the challenges of transistor variability and degradation**," in *IEEE Micro,* 25(6):10–16, 2005.
- V. Chandra and R. Aitken, "**Impact of Technology and Voltage Scaling on the Soft Error Susceptibility in Nanoscale CMOS**," *2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems*, Boston, MA, 2008, pp. 114-122.
- International technology roadmap for semiconductors (ITRS), 2015.
- …..

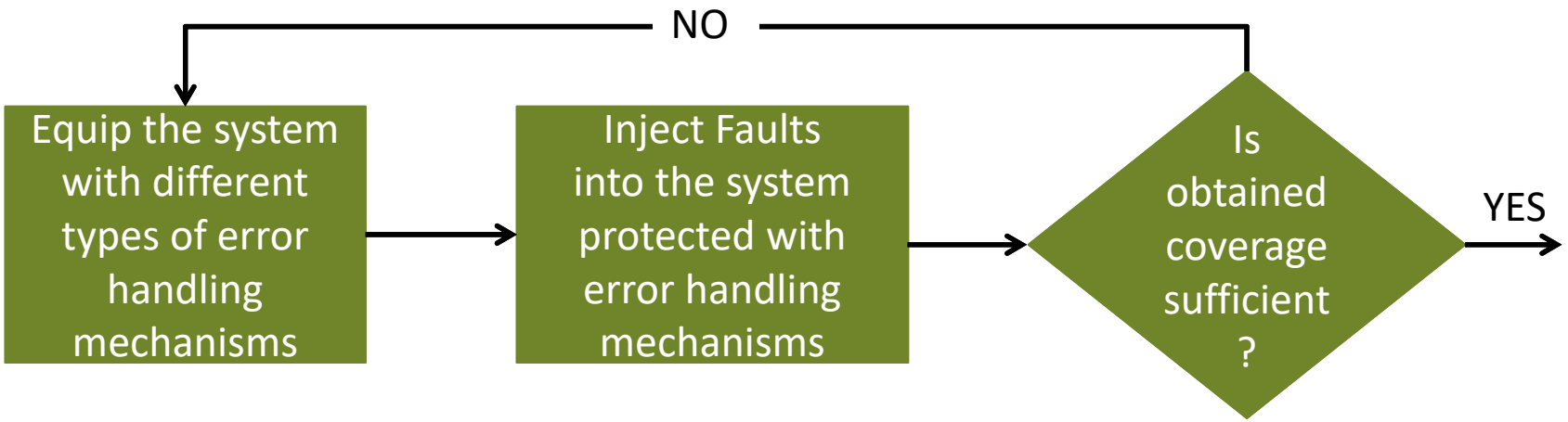
Background



The system's ability to detect and correct errors needs to be measured.

Background

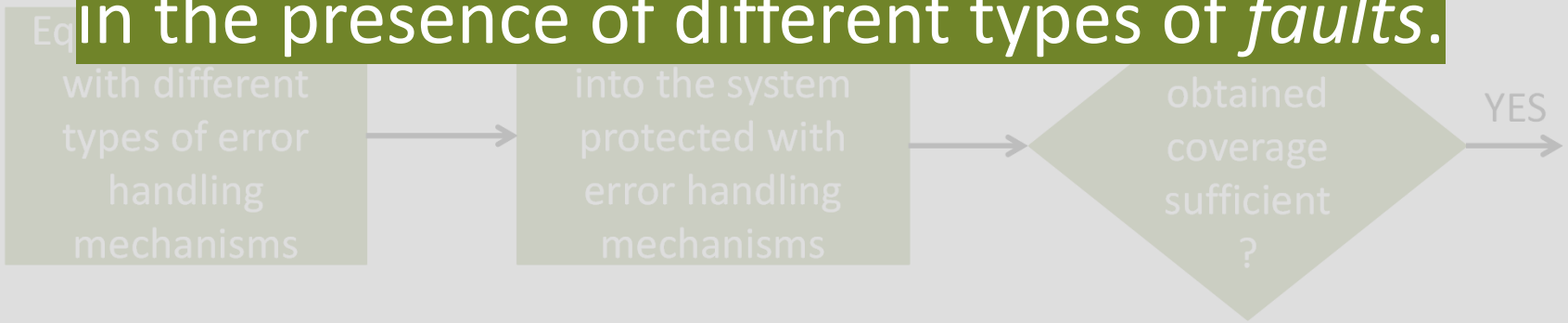
- Fault injection
  - To obtain coverage estimates for applications
  - Iteratively improve coverage based on the errors missed by *error handling mechanisms*



Background

- Fault injection
  - To obtain coverage estimates for applications
  - Iteratively improve coverage based on the errors

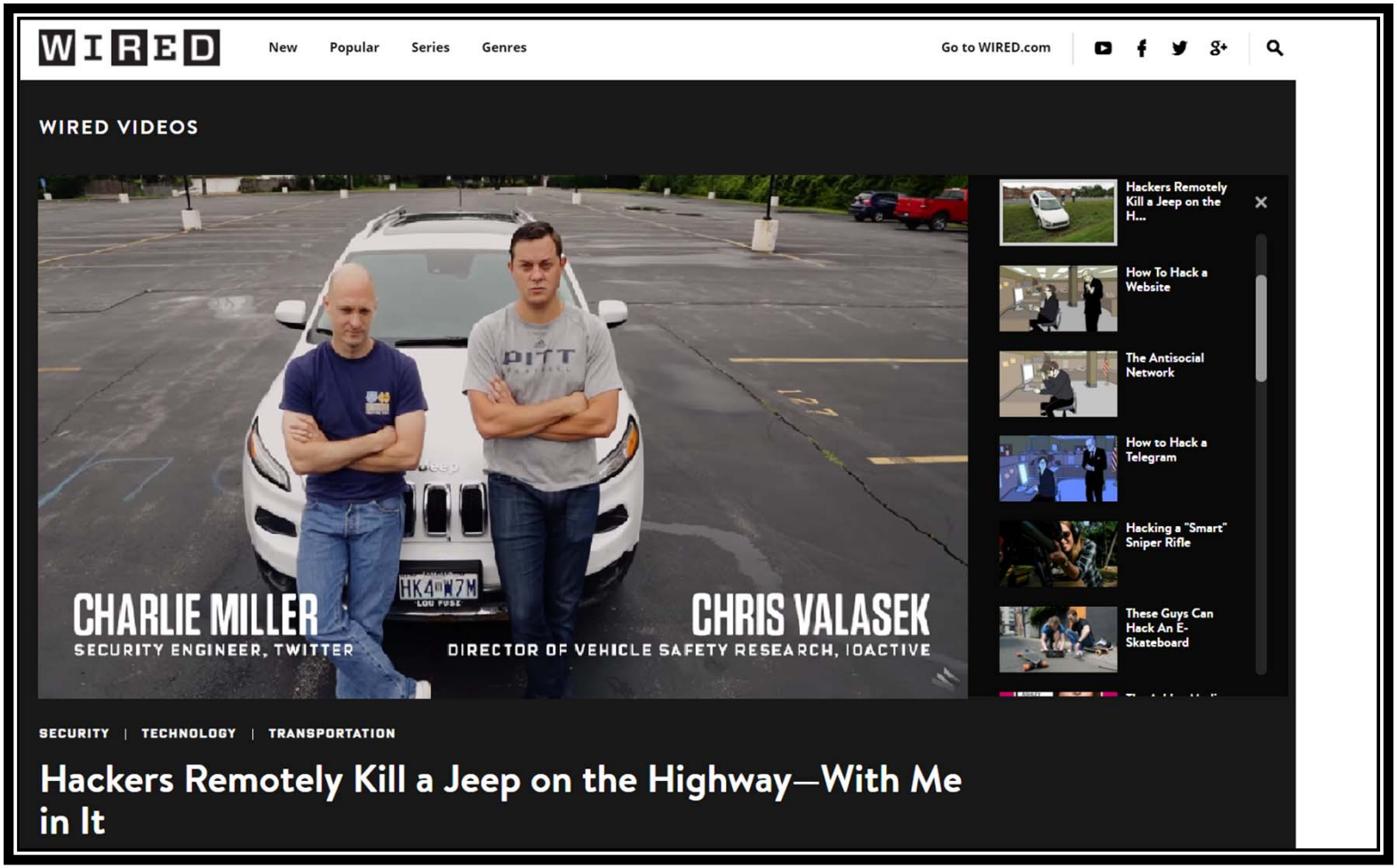
Fault injection has been effectively used to evaluate the effectiveness of *safety mechanisms* in the presence of different types of *faults*.



What About Security Attacks?

- Automotive domain
  - Cars were not created with Internet connectivity in mind, and adding this connectivity as an afterthought raises many *security concerns*.

What About Security Attacks?



What About Security Attacks?

Team of hackers take remote control of Tesla Model S from 12 miles away

Chinese researchers were able to interfere with the car's brakes, door locks and other electronic features, demonstrating an attack that could cause havoc

▲ Now that cars such as Tesla's are increasingly high-tech and connected to the internet, cybersecurity has become as big an issue as traditional safety features. Photograph: Jim Dyson/Getty Images

What About Security Attacks?

Research Questions

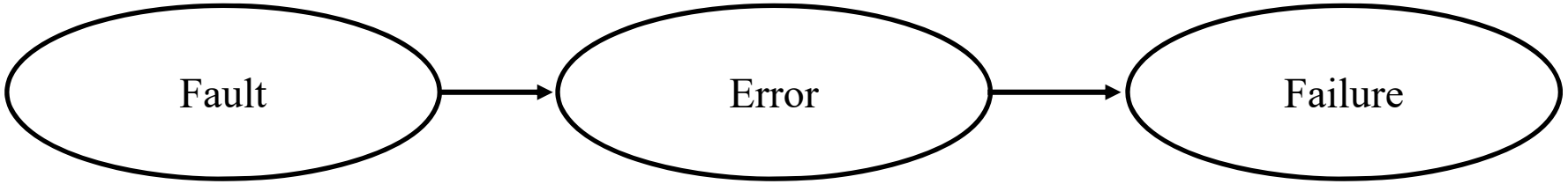
How effective is it to use fault injection to inject *security attacks*?

To what extent can *security attacks* influence system safety?

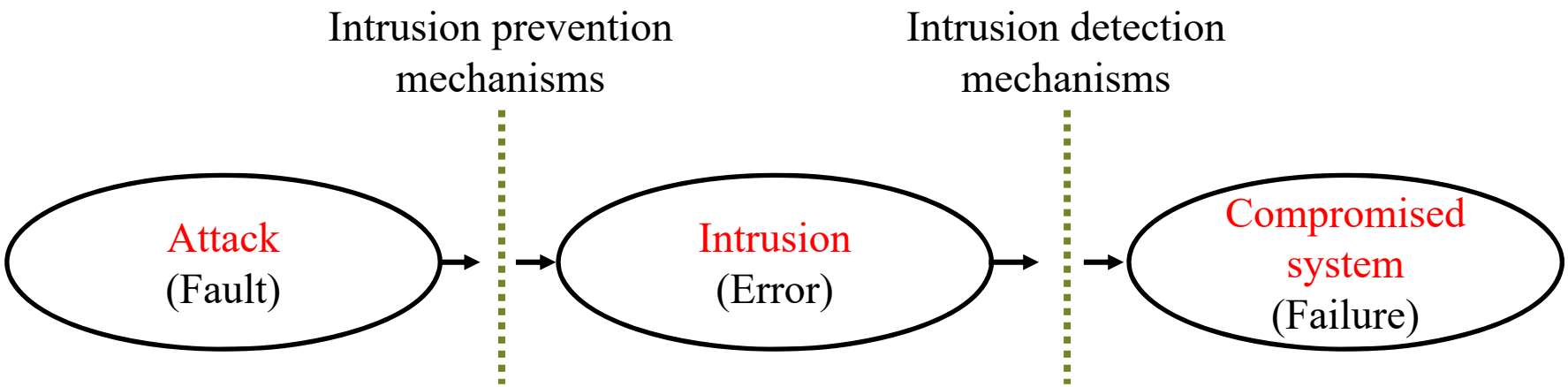
Why Using Fault Injection to Simulate Security Attacks?

- Fault injection has been successfully used to evaluate system safety
- Fault injection maps well into Security Testing frameworks
  - Fuzz testing
  - Vulnerability testing
  - Penetration testing

Using Fault Injection to Simulate Security Attacks

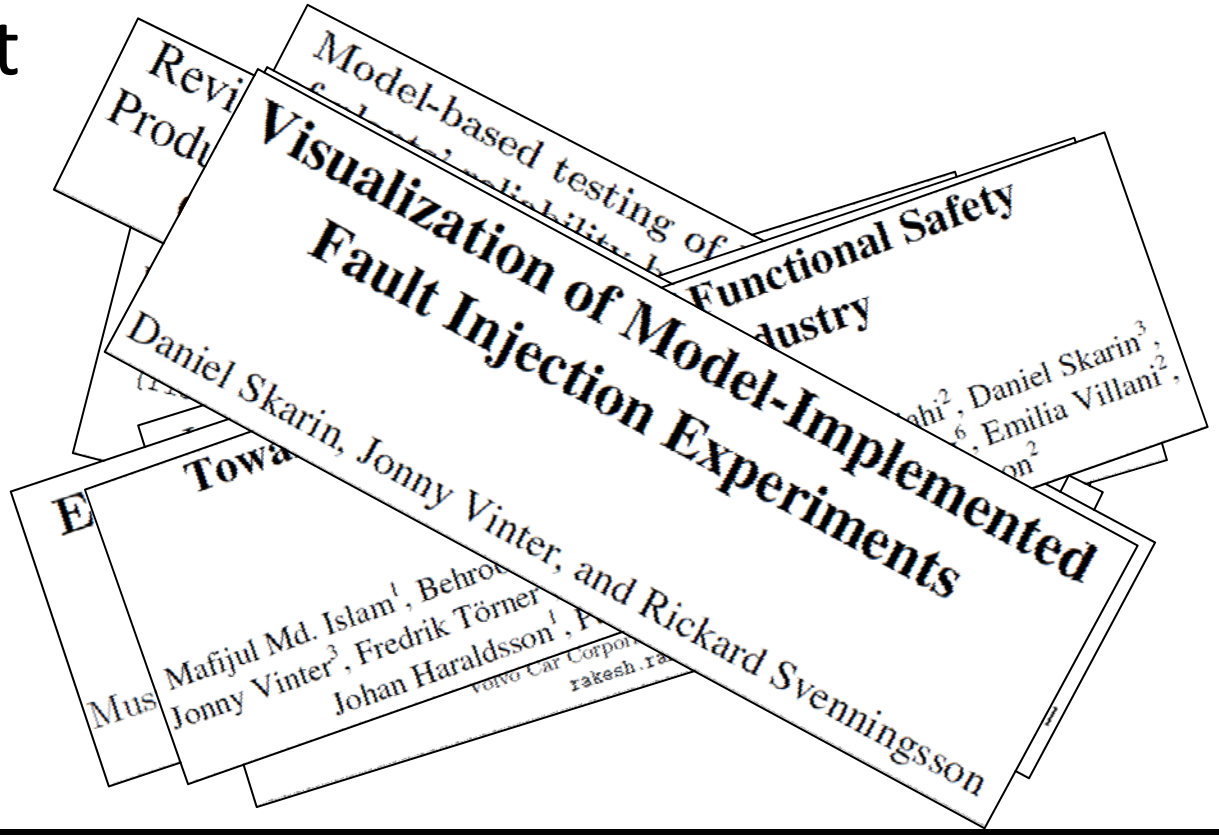


Using Fault Injection to Simulate Security Attacks



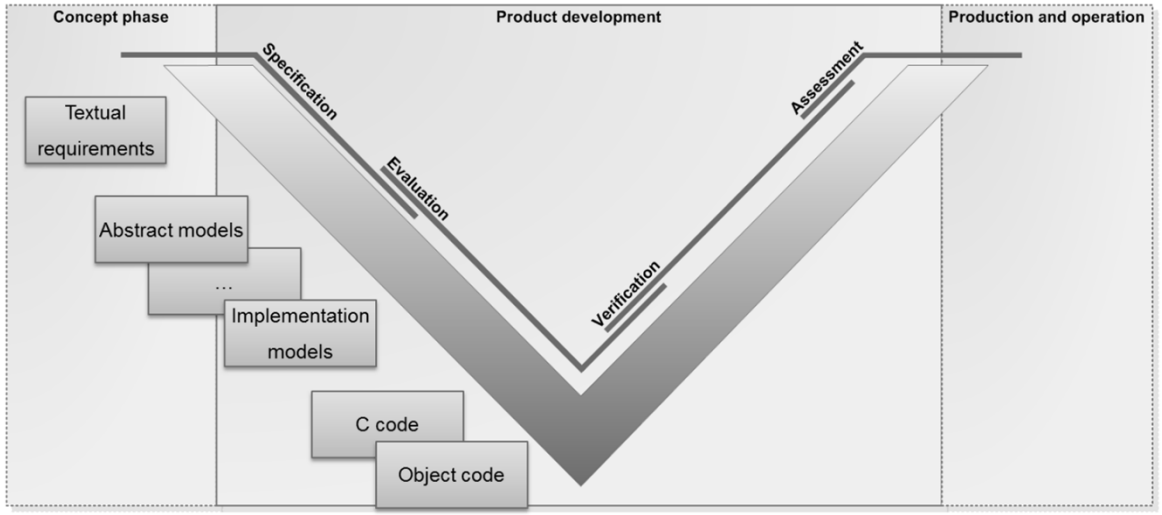
MODel-Implemented Fault Injection (MODIFI)

- Integrating Fault Injection into Model-based Development



MODel-Implemented Fault Injection (MODIFI)

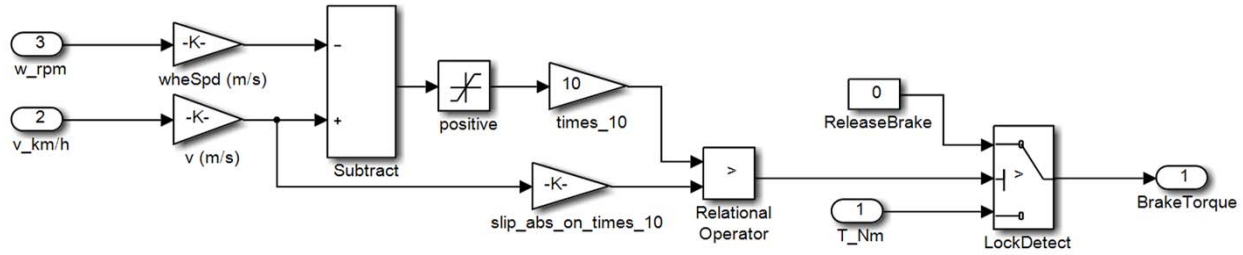
- Useful for early dependability evaluation of software developed as models



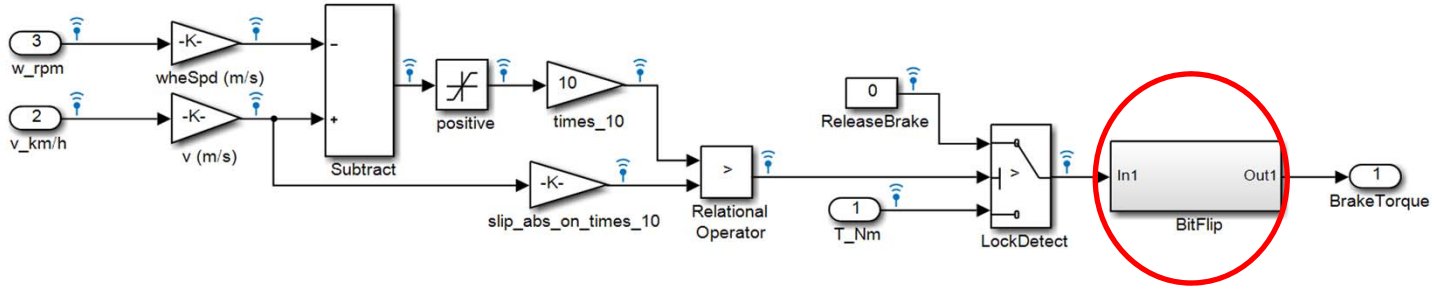
MODel-Implemented Fault Injection (MODIFI)

- Injecting Faults in Simulink Models

Original model



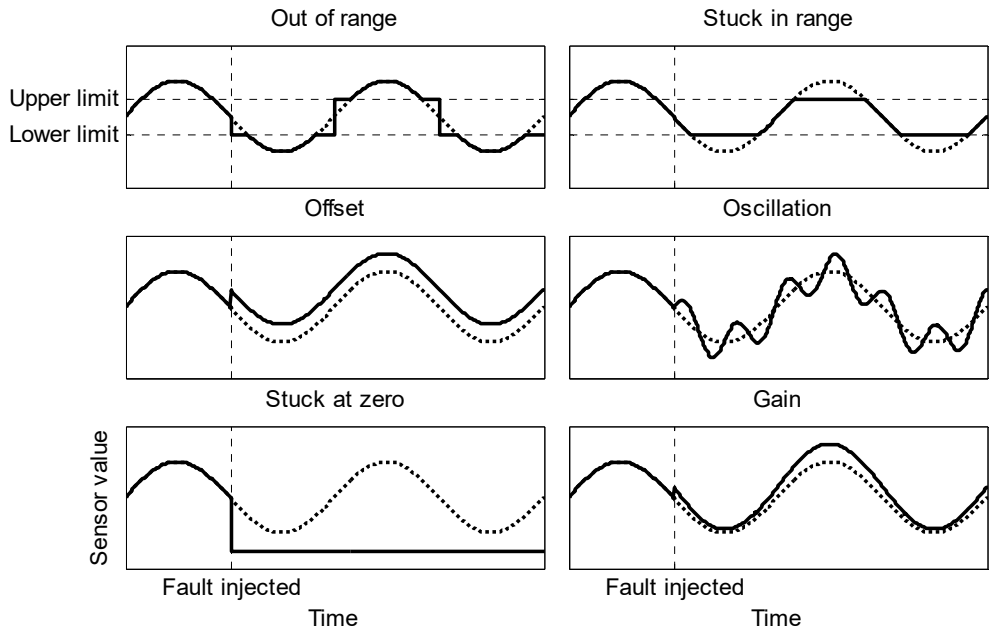
Model with fault injection support



MODel-Implemented Fault Injection (MODIFI)

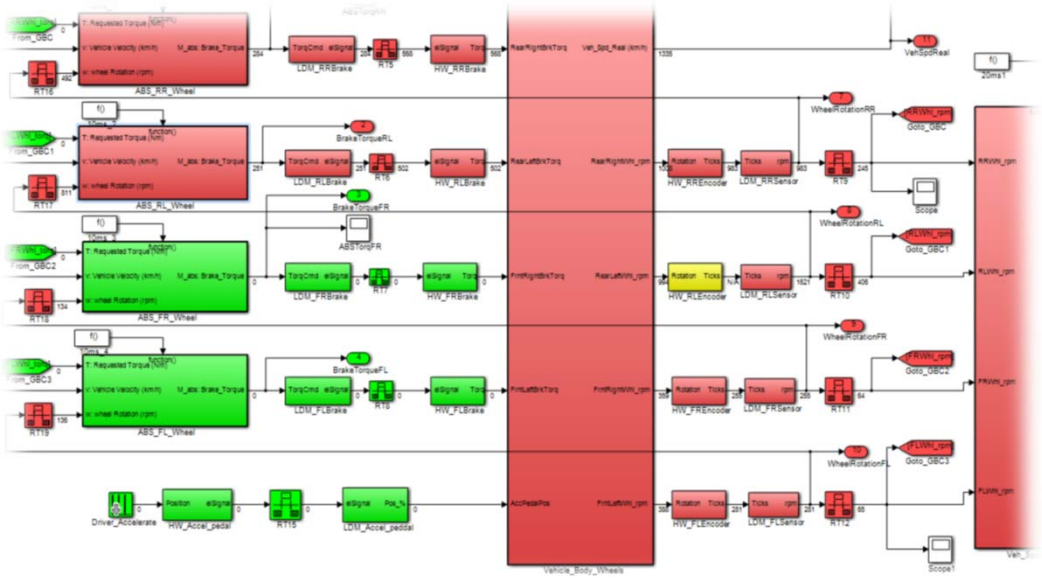
- Provides a large number of fault models

Sample fault models
Single/multiple bit-flip
Oscillations
Offset
Stuck-at-zero
Stuck-at-one
Stuck-at-value
Stuck-in-range
Stuck-OutOf-range



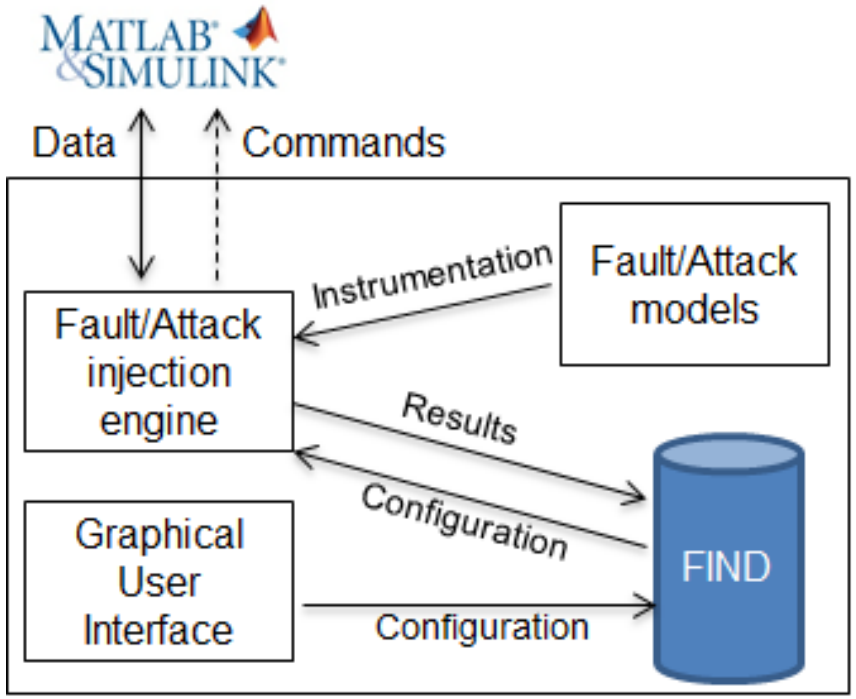
MODel-Implemented Fault Injection (MODIFI)

- Includes support for analyzing and visualizing fault injection results

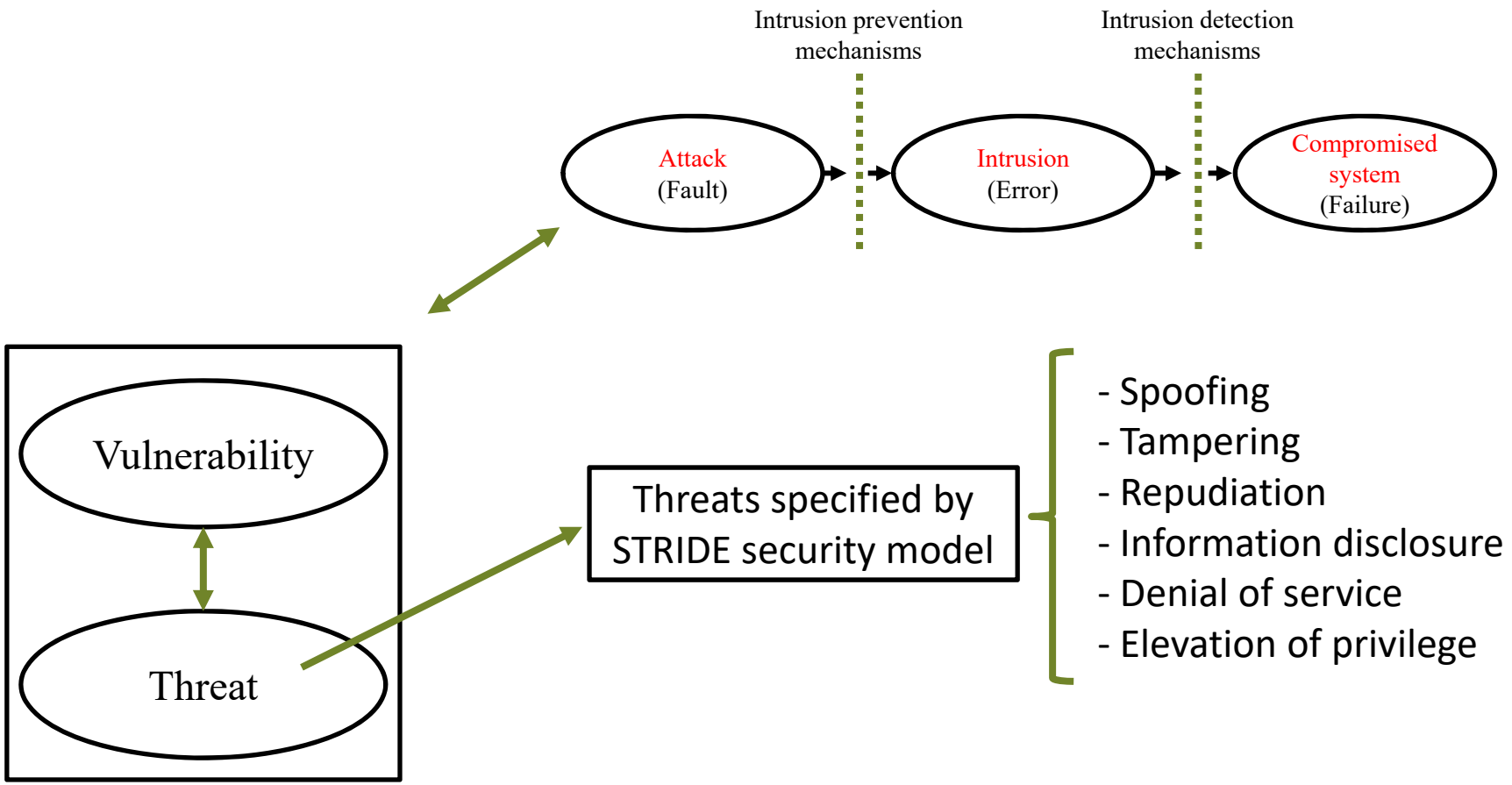


Model-Implemented Attack Injection

- Overview of MODIFI



Model-Implemented Attack Injection

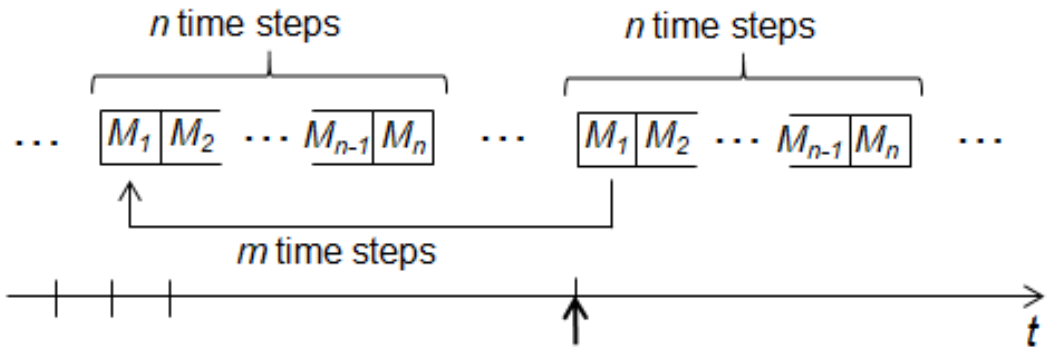


Model-Implemented Attack Injection

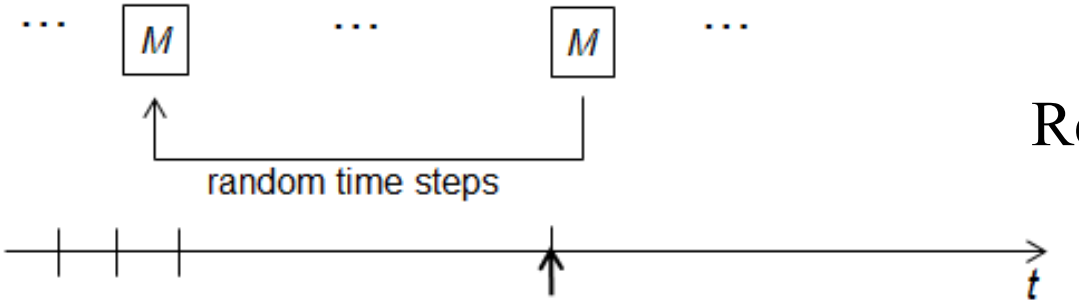
- Cybersecurity attacks

Threat	Attack	*G*eneric or *S*ystem-*S*pecific	*A*ctive or *P*assive
Spoofing	Fake messages	S-S	A
	Spoofing	S-S	A
Tampering	Corrupt data or code	G	A
	Deliver malware	S-S	A
	Alter	S-S	A
	Inject	S-S	A
	Corrupt messages	G	A
	Manipulate	S-S	A
	Configuration change	S-S	A
Repudiation	Replay	G	A
	Repudiation of message transmission and receipt	S-S	A
Information disclosure	Listen	G	P
	Intercept	G	A
	Eavesdropping	S-S	P
	Illegal acquisition	S-S	P
	Covert channel	S-S	P
Denial of service	Disable	S-S	A
	Denial of service	S-S	A
	Jamming	G	A
Elevation of privilege	Illegal acquisition	S-S	P
	Gain root access	S-S	P

Cybersecurity Attack Models

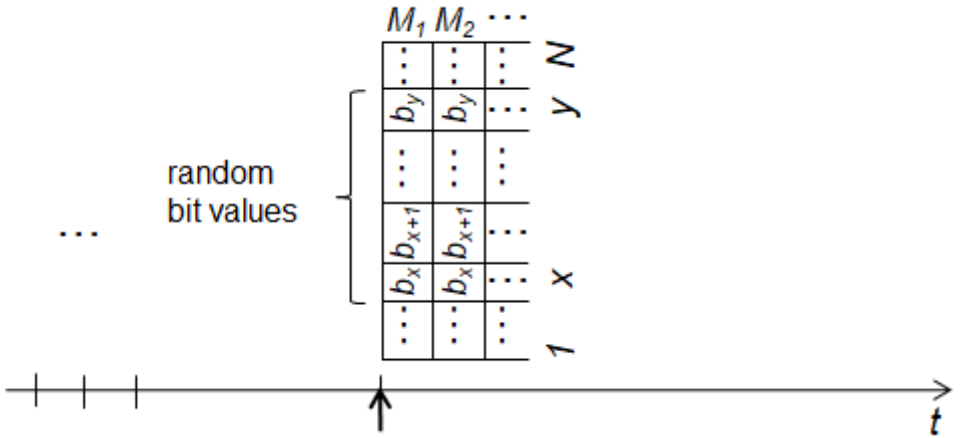


Replay(n, m)

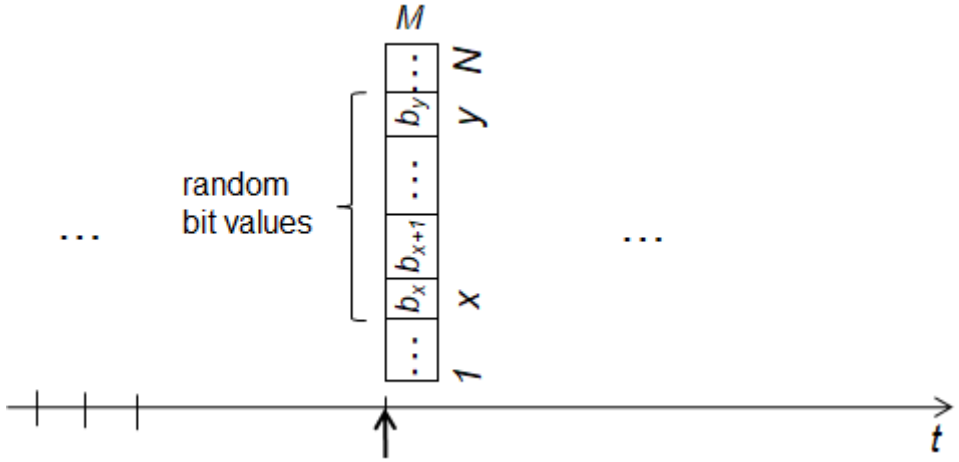


ReplayRandomMessage

Cybersecurity Attack Models

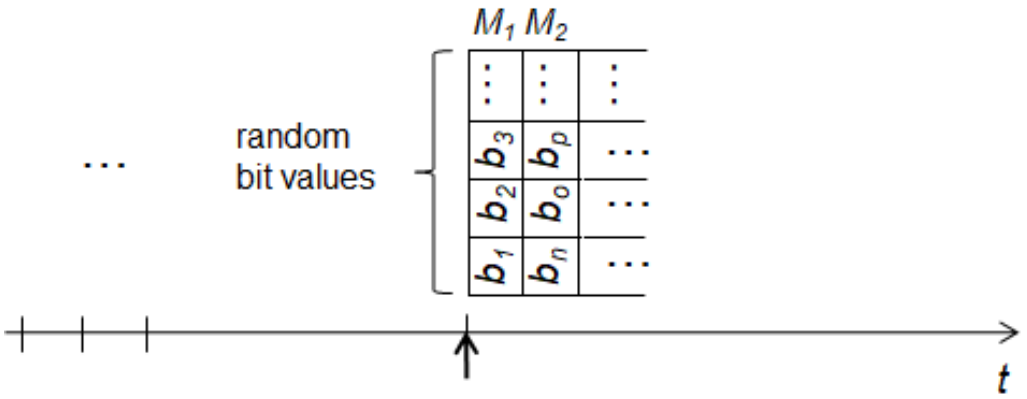


CorruptDataOrCode(x, y)

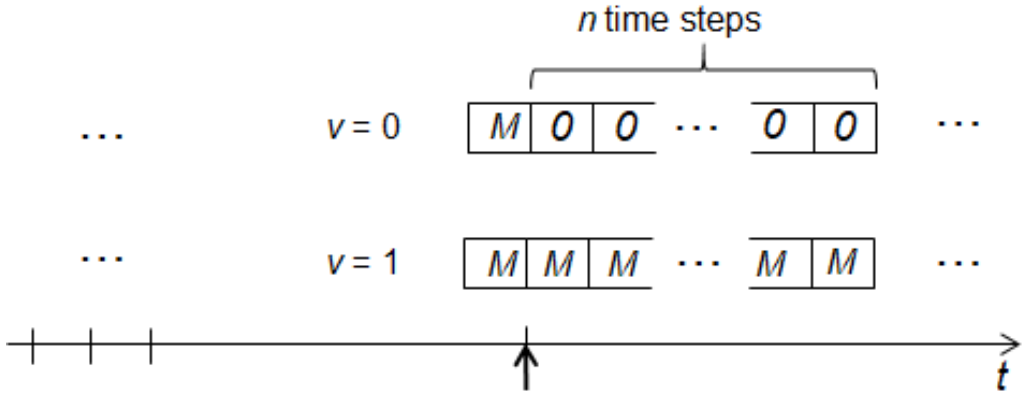


CorruptMessage(x, y)

Cybersecurity Attack Models



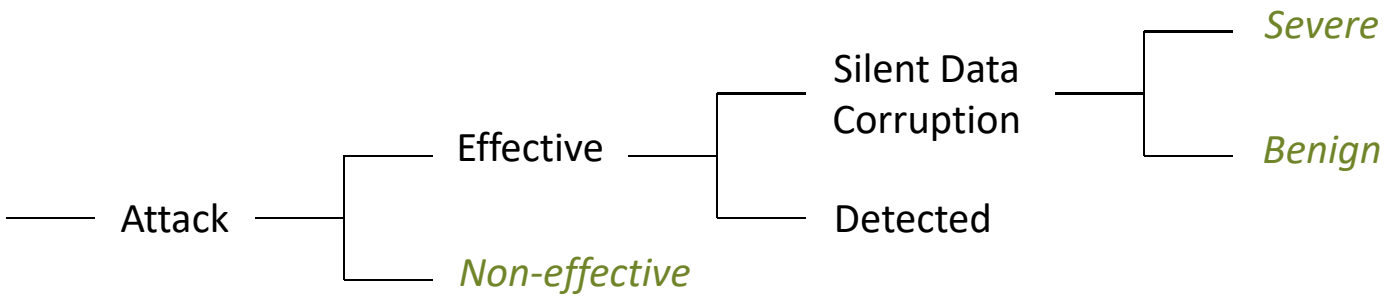
Jamming



Intercept(n, v)

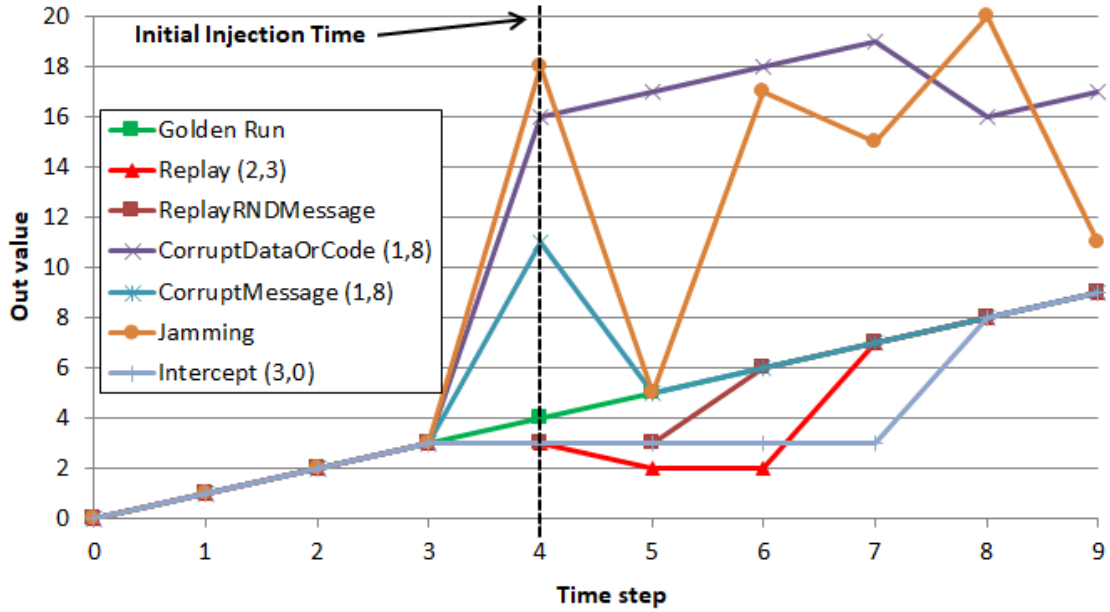
Interplay between Safety and Security

- Outcome Classification



Interplay between Safety and Security

- CAN bus



Interplay between Safety and Security

- Brake-by-wire controller

Attack model	Number of attacks	Non-effective	Benign	Severe
Replay(n, m)	1388	712	676	0
ReplayRandomMessage	1388	586	768	34
CorruptDataOrCode(x, y)	1388	457	629	302
CorruptMessage(x, y)	1388	784	537	67
Jamming	1388	97	294	997
Intercept(n, 0)	1388	540	797	51

Interplay between Safety and Security

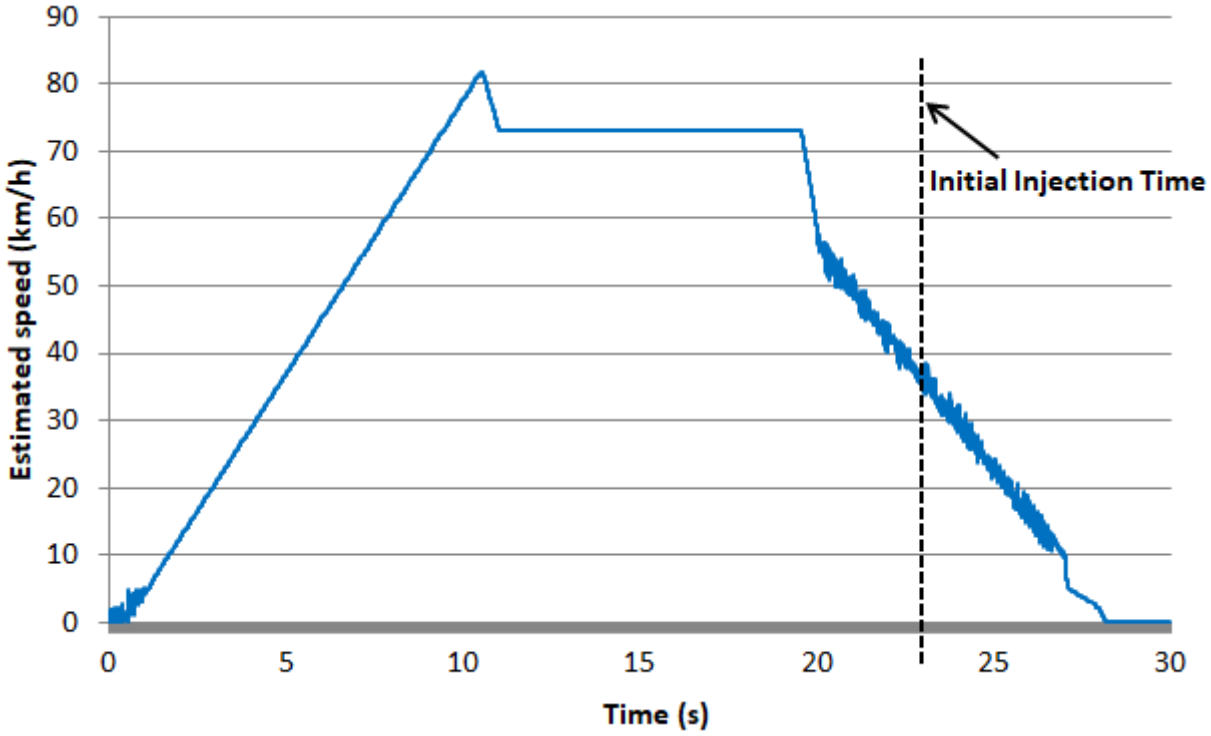
- Brake-by-wire controller

Attack model	Number of attacks	Non-effective	Benign	Severe
Replay(n, m)	1388	712	676	0
ReplayRandomMessage	1388	586	768	34
CorruptDataOrCode(x, y)	1388	457	629	302
CorruptMessage(x, y)	1388	784	537	67
Jamming	1388	97	294	997
Intercept(n, 0)	1388	540	797	51

n and *m* are randomly selected between 1 and 10.

Interplay between Safety and Security

- Brake-by-wire controller



Interplay between Safety and Security

- Brake-by-wire controller

Attack model	Number of attacks	Non-effective	Benign	Severe
Replay(n, m)	1388	712	676	0
Intercept(n, 0)	1388	540	797	51

n and *m* are randomly selected between 1 and 10.

Attack model	Number of attacks	Non-effective	Benign	Severe
Replay(n, m)	1388	398	557	433
Intercept(n, 0)	1388	249	292	847

n and *m* are randomly selected between 1 and 1000.

Summary

- Using fault injection to model cybersecurity attacks
  - Model implemented attack injection
- Interplay between safety and security
  - Evaluation of six cybersecurity attacks

Implication

- Mapping of Cybersecurity Attacks to Commonly Used Fault Models

Cybersecurity Attack Models	Traditional Fault Models
Replay(n, m)	-
ReplayRandomMessage	-
CorruptDataOrCode(x, y)	Multiple stuck-at
CorruptMessage(x, y)	Multiple bit-flips
Jamming	Oscillations
Intercept(n, v)	Stuck-at-zero or Stuck-at-value

Implication

Take-away:

Safety and security analysis of computer systems should be done by a mixed group of safety and security experts.

Future Work

- Modeling many more cybersecurity attacks
  - As well as combining different attacks
- Evaluating safety and security mechanisms
  - Analyzing the impact of security mechanism on system safety
  - Analyzing the impact of safety mechanism on system security

?

Contact:

Behrooz Sangchoolie
behrooz.sangchoolie@ri.se
+46 10 516 61 89

Peter Folkesson
peter.folkesson@ri.se
+46 10 516 54 16

RISE Research Institutes of Sweden
RISE Safety and Transport Electronics

RI. SE
Research Institutes of Sweden