Elicitor: Usage-Frequency Independent Detection of Resource-Release Omission Faults

Suman Saha, Julia Lawall and Gilles Muller
Regal-LIP6/INRIA Paris, France

Introduction

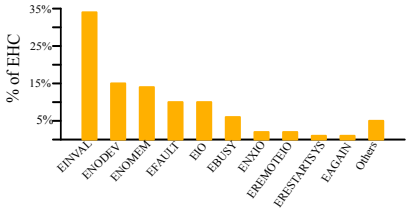
Potential sources of errors

- Inappropriate user requests.
- Defective or nonconforming devices.
- Inconsistent configuration.
- Bugs in kernel-level code.

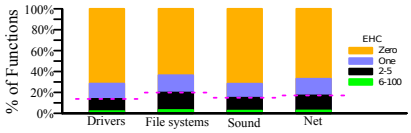
Handling errors is key to the continuing viability of the system.

- User expects the system to remain functional
- Use error-handling code (EHC).

Reasons for errors in drivers (Linux-2.6.34)



EHC in Linux-2.6.34



Motivation

EHC may contain faults

- Missing resource-releasing operations.

Consequences

- System Crash
- Deadlock
- Run out of resources
- Inconsistent state

Faults in EHC may be exploited by malicious users.

Elicitor, a Tool to Detect Faults

Method

- Use local information (in the same function).
- Use model of correct EHC to detect defective EHC in the same function.

```
hw = wl1251_alloc_hw();
...
ret = spi_setup(spi);
if(ret < 0) {
  ...
  goto out_free;
}
...
w1->set_power =
            pdata->set_power;
if(!w1->set_power) {
  ...
  return -ENODEV;
}
...
out_free:
   ieee808211_free_hw(hw);
   return ret;
```

Preliminary Results

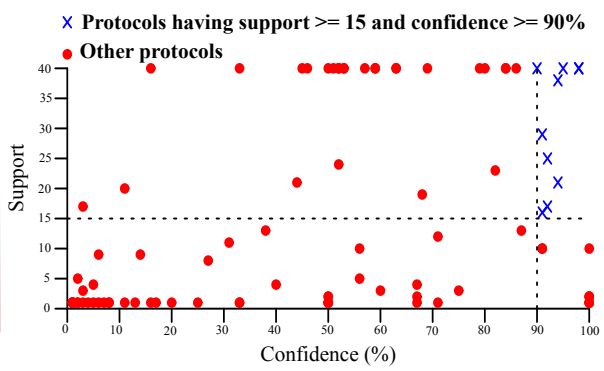
Linux-2.6.34	Reports	Bugs	FP	TODO
Drivers	277	216	50	11
File systems	47	20	27	0
Sound	30	18	12	0
Net	35	8	19	8
Total	**389**	**262**	**108**	**19**

Existing Approaches

Well-known approaches

- Find protocols using Data-mining.
- Statistics to detect potential protocols.
- Identify violations of these protocols
- Problem: depend on thresholds. Therefore, miss important protocols that do not satisfy threshold values

Elicitor vs Data-Mining



88 protocols associated to 230 faults can not be detected by Data-mining using these thresholds (support and confidence) values.

Examples: PR-Miner (FSE'05), Bugs as Deviant Behavior (SOSP'01)