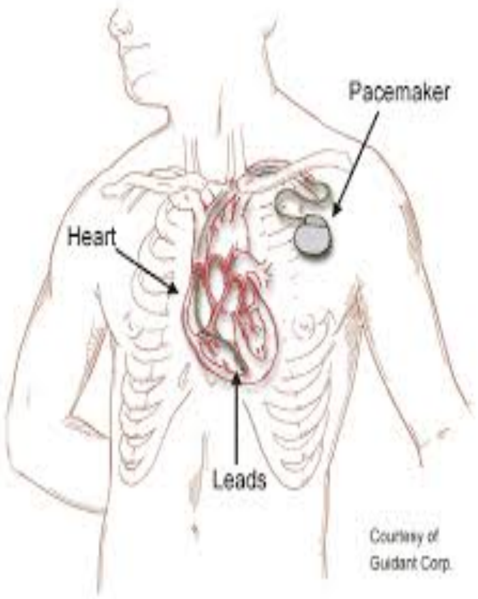
Formal Security Analysis of Smart Embedded Systems

Farid Molazem Tabrizi
Karthik Pattabiraman

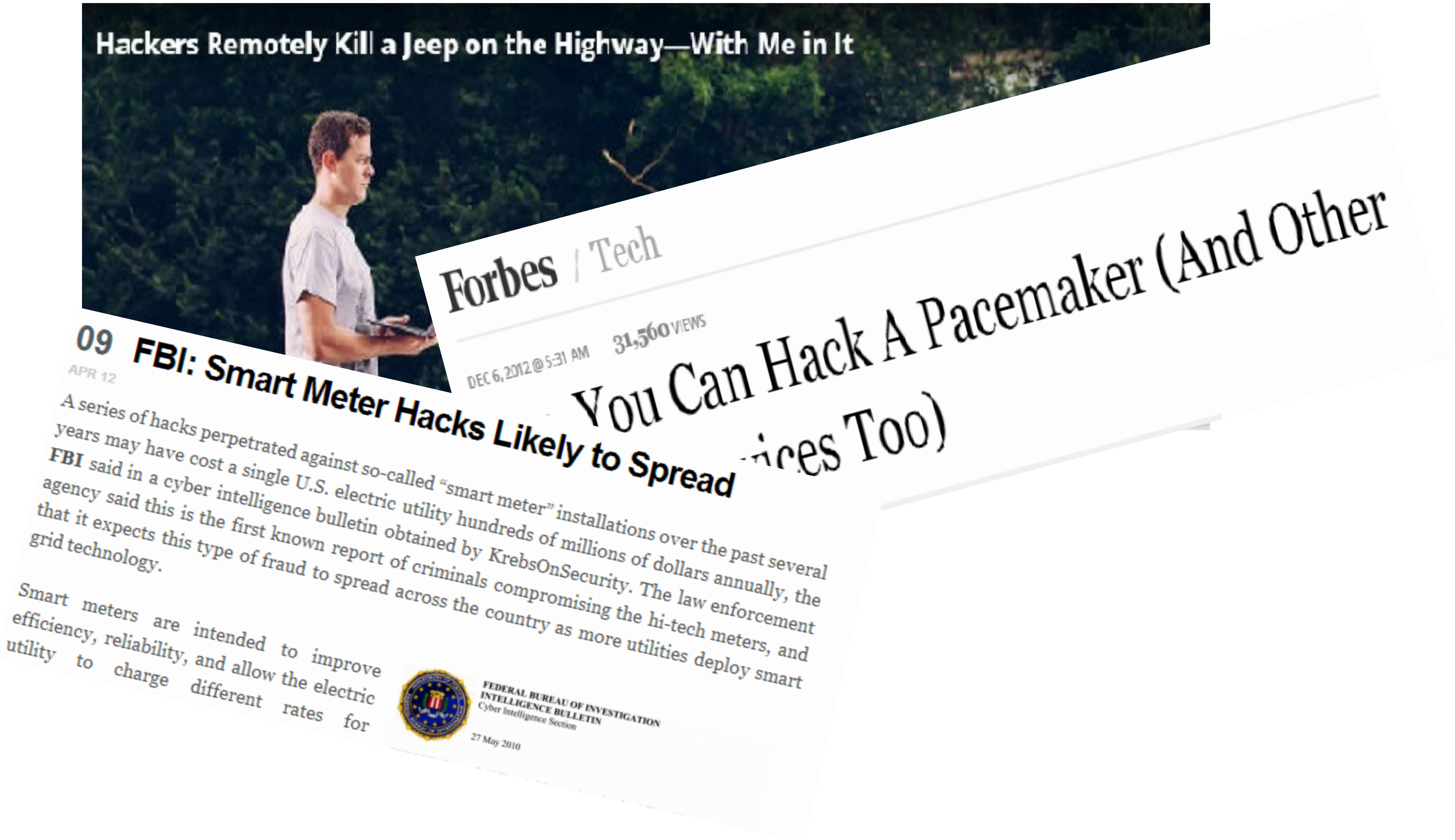
http://blogs.ubc.ca/karthik/

IoT Systems



Courtesy of Guidant Corp.

Security Attacks against IoT



Hackers Remotely Kill a Jeep on the Highway—With Me in It

Forbes / Tech

DEC 6, 2012 @ 5:31 AM 31,560 VIEWS

You Can Hack A Pacemaker (And Other ices Too)

09 APR 12

FBI: Smart Meter Hacks Likely to Spread

A series of hacks perpetrated against so-called "smart meter" installations over the past several years may have cost a single U.S. electric utility hundreds of millions of dollars annually, the **FBI** said in a cyber intelligence bulletin obtained by KrebsOnSecurity. The law enforcement agency said this is the first known report of criminals compromising the hi-tech meters, and that it expects this type of fraud to spread across the country as more utilities deploy smart grid technology.

Smart meters are intended to improve efficiency, reliability, and allow the electric utility to charge different rates for

FEDERAL BUREAU OF INVESTIGATION
INTELLIGENCE BULLETIN
Cyber Intelligence Section

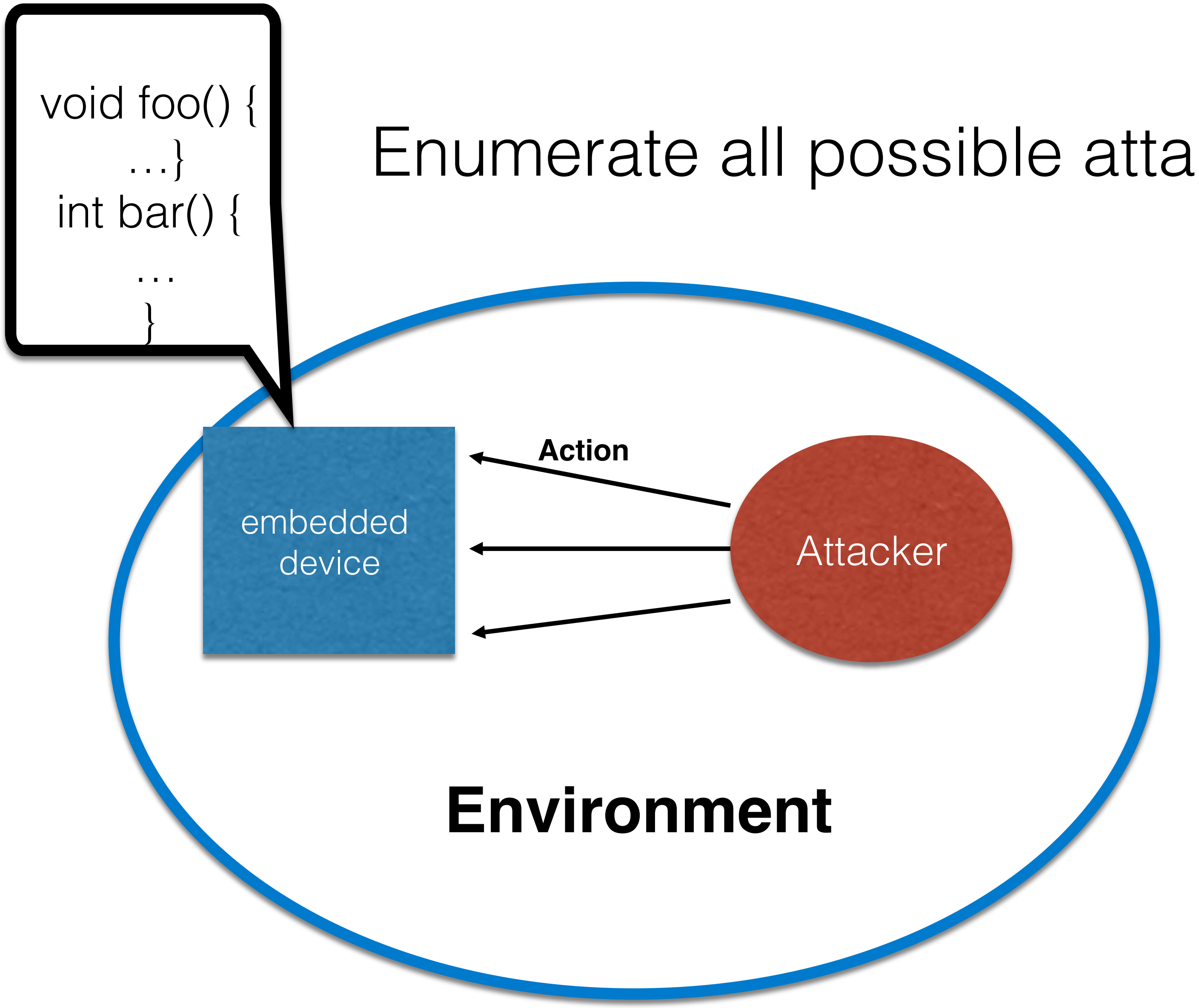
27 May 2010

Challenge

- No systematic technique to automatically find security vulnerabilities in IoT devices
  - Large attack surface
  - Attacker often has physical access
  - Devices are often resource constrained

Problem

Enumerate all possible attacks



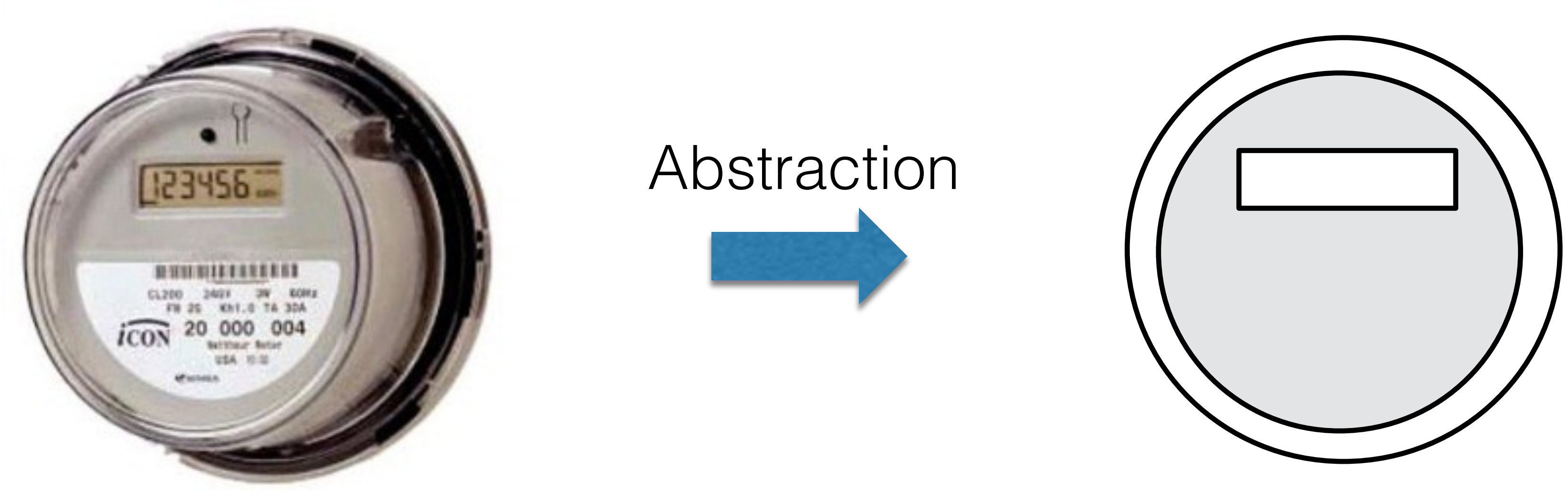
```
void foo() {
   …}
int bar() {
   …
}
```

Security Analysis

- **Attack trees [Byres 04, Morais 09]**
  - Predefined attack goals
  - Manual search
- **Attack graphs [Jha 02, Sheyner 02]**
  - Need vulnerabilities of the hosts
- **Formal analysis [Delaune 10, Miculan 11]**
  - Targets well-defined protocols

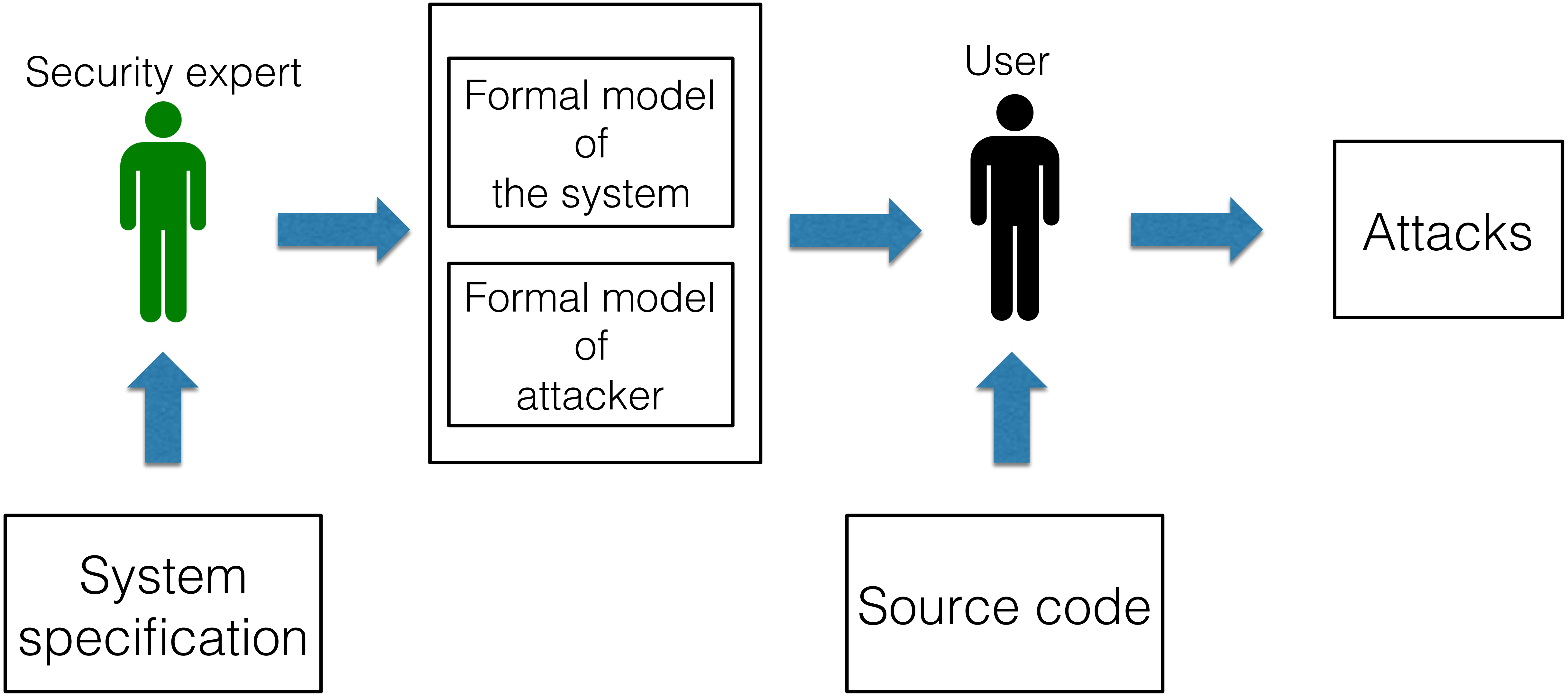
Our Approach: Idea

- IoT devices perform *specific* tasks
  - Define the right abstraction
    - Not too low level, not too high level


Abstraction

- Allows us to systematically find vulnerabilities

High-level picture



Security expert

Formal model of the system

Formal model of attacker

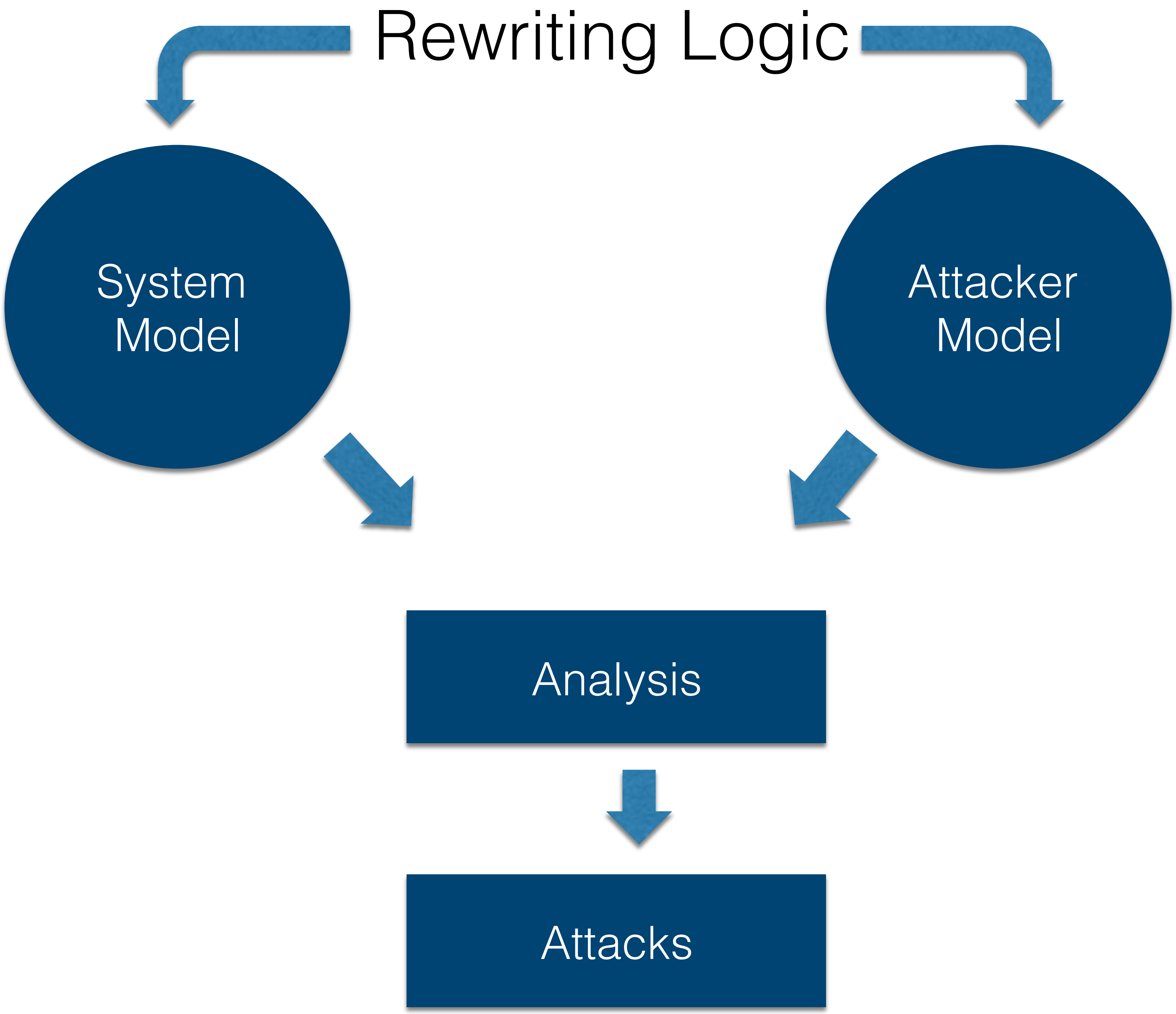
User

Attacks

System specification

Source code

Abstraction



Rewriting Logic

System Model

Attacker Model

Analysis

Attacks

Abstraction: System Model

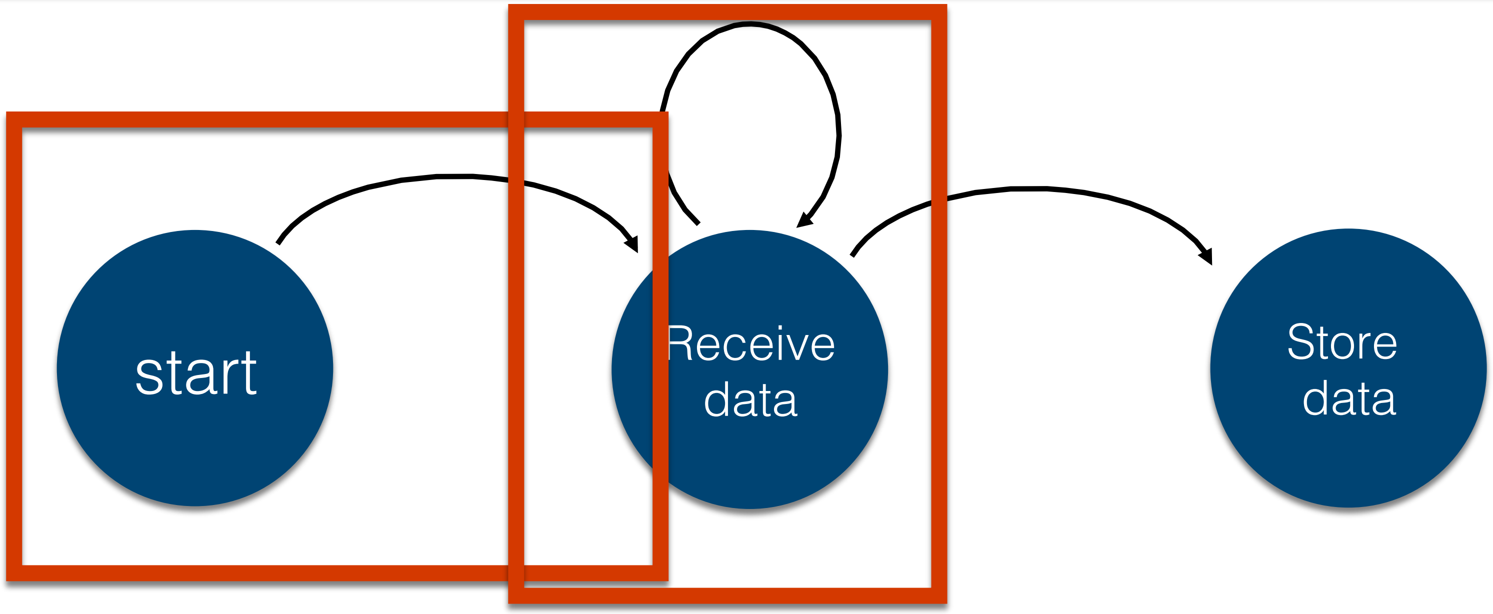
Rewriting logic:

- Rewrite rules
  - Equations

Start ➔ sensorData(0, 0)

sensorData(r, n) ➔ sensorData(r, n) sensorData(r+1, 0)

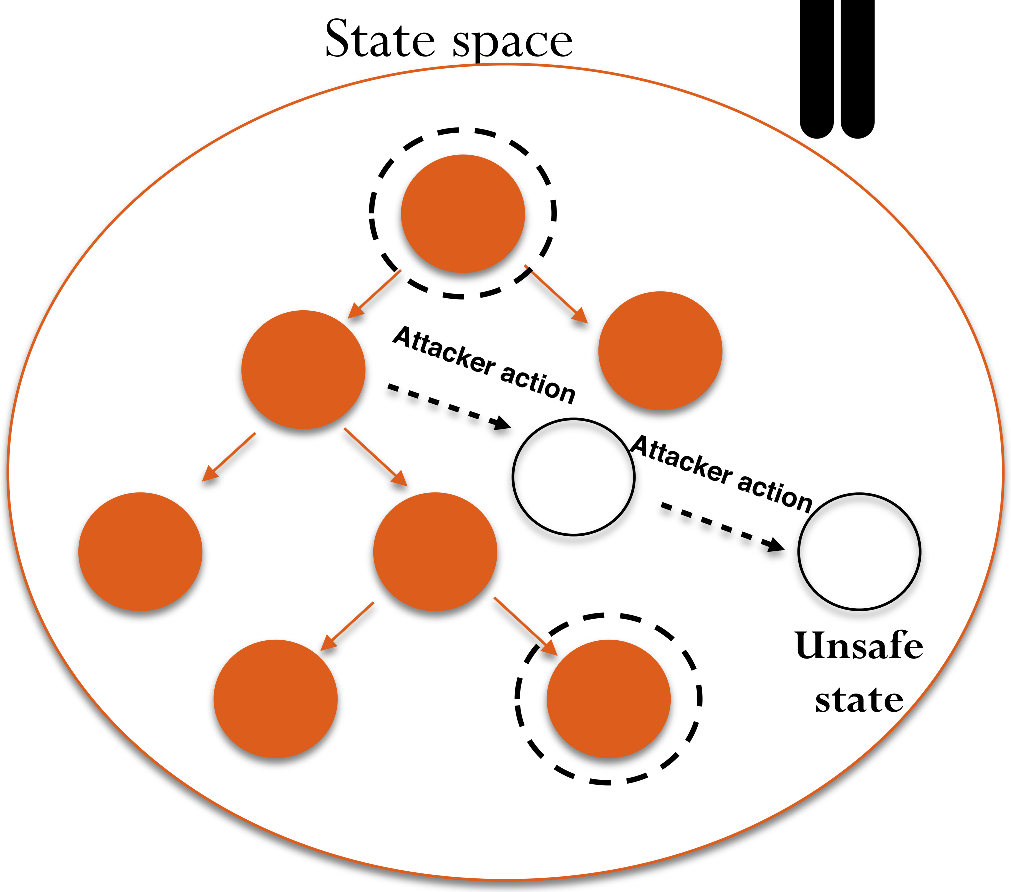
sensorData(r, n) ➔ sensorData(r, n+1)



Abstraction: Attacker Model

Attacker action:
e.g. access to the i*th* sensor channel

sensorData(c1, v1) sensorData(c2, v2) sensorData(c3, v3)➔
sensorData(c1, v1) sensorData(c3, v3) *if c2 = i*

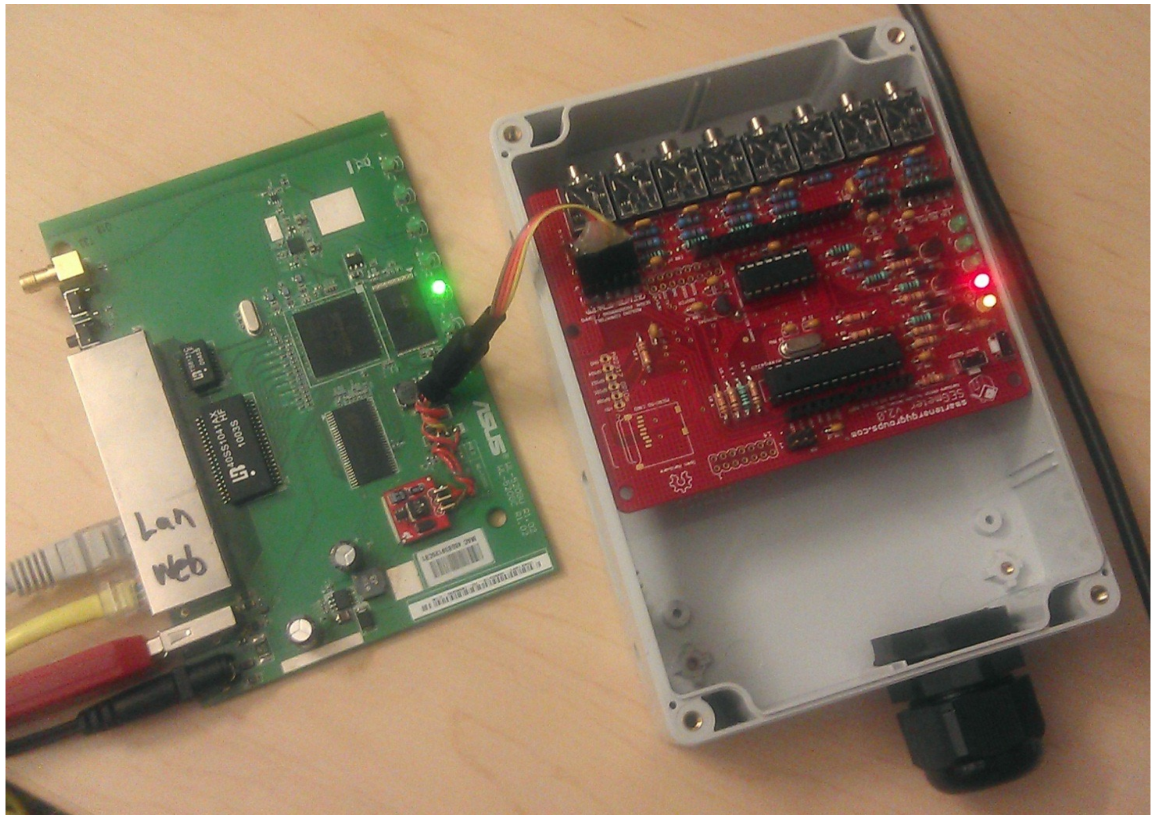


Explicit model checking:

Start ➔ receive(c1, v1) *where v1 < 0*

Case study

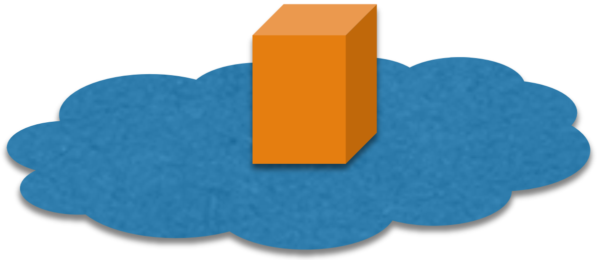
- SEGMeter: an open source smart meter
- Sensor board: Receive raw data
- Communication board: talk to server
- Code base: Lua and C (~ 3000 LOC)



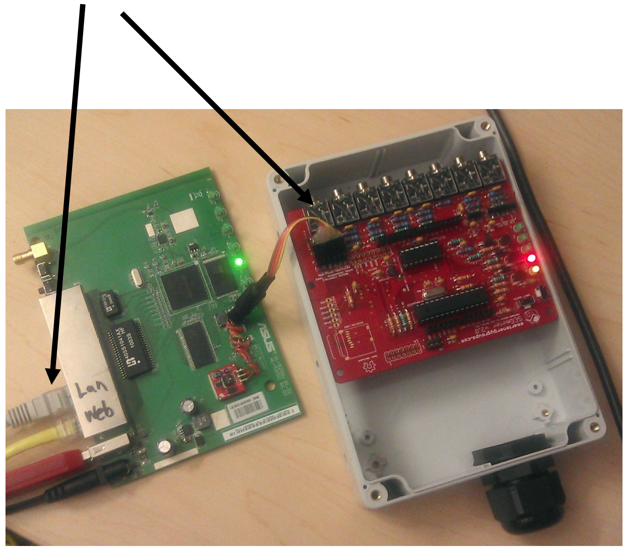
Threat model

- Access

Root access to a node in grid network [Mo et al. 2012]



Read/Write access to communication interfaces[McLaughlin et al. 2010]



- Actions
  - Drop messages
  - Replay messages
  - Reboot meter

Evaluation

Performance

Using Maude [Clavel 15]:
http://maude.cs.illinois.edu/

Less than a second → up to 2 hours

3.4 GHz CPU, 16GB RAM

Evaluation

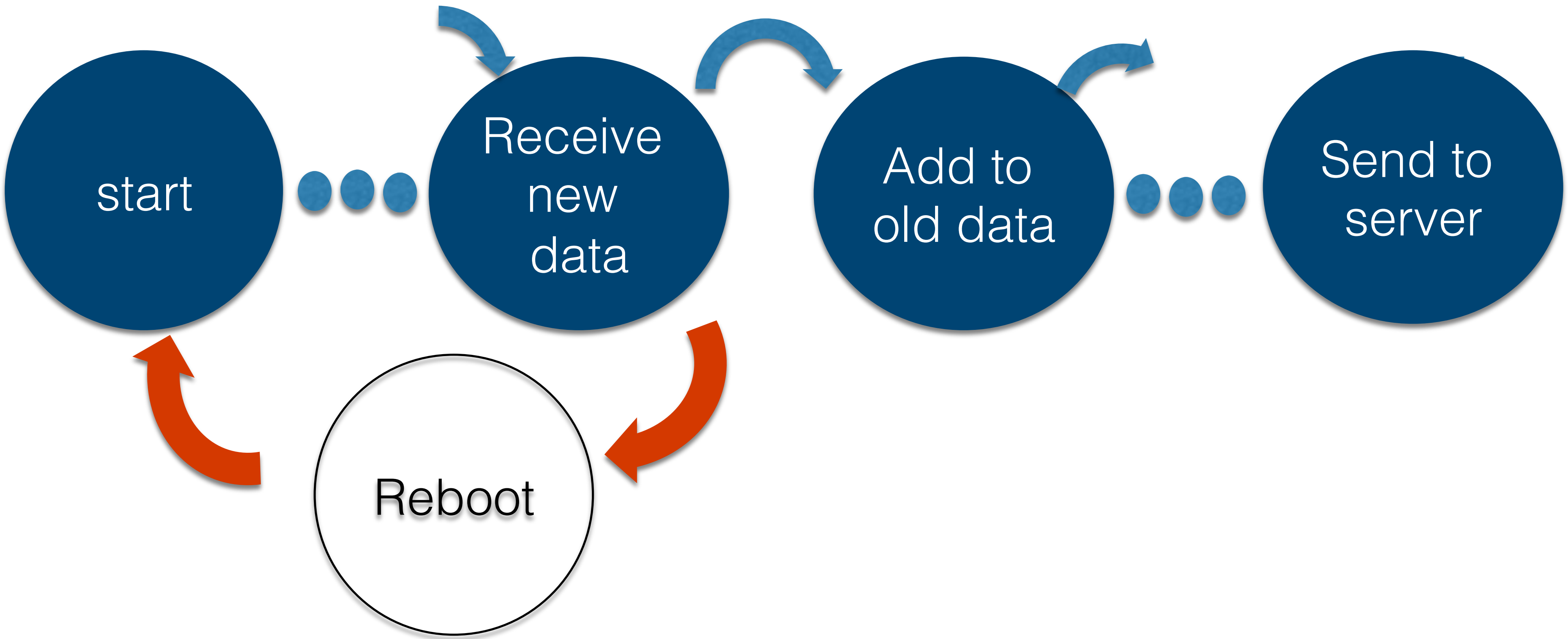
Practicality

- Query for paths to unsafe states

> *search sensor(N1, M1) sensor(N2, M2) sensor(N3, M3) ⇒ stored(N1, M1) stored(N2, M2)*

- Some map to the same execution path

Attack Example 1: Rebooting



S1 ➔ S2 where data(s1) *not sent* & cycle=start

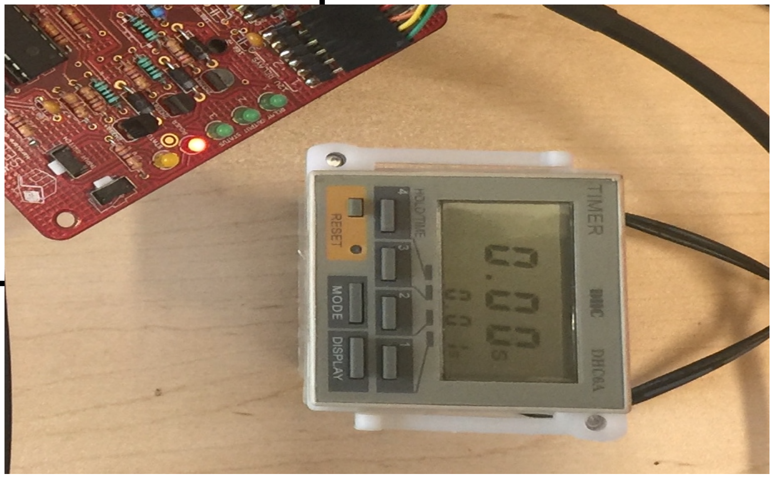
Attack Example 1: Rebooting

Will lose data if reboot

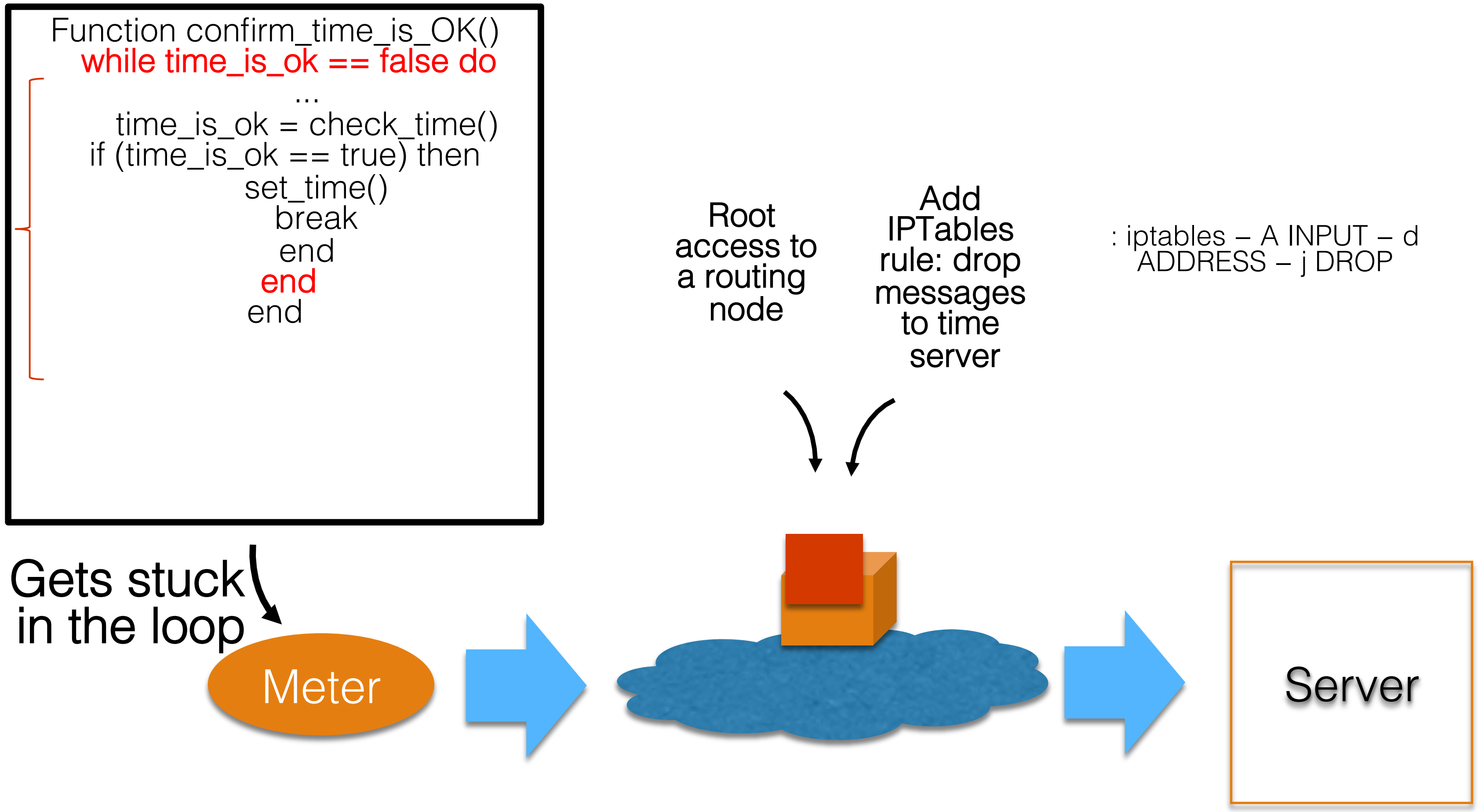
Vulnerability window

Open file in write mode

```
function update_node_list()
 all_data = get_node_list
 all_data = merge_table(current,all_data)
data_file = assert(io.open(dataFile, “w”))
 for key, value in pairs(node_list) do
       data_file::write(data)
 end
 assert(data_file::close())
end
```



Attack Example 2: Drop Messages



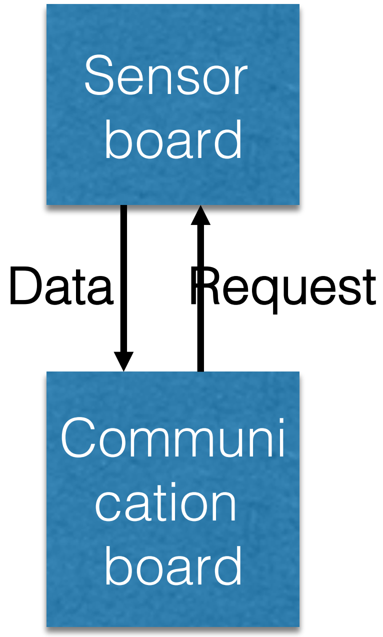
```
Function confirm_time_is_OK()
    while time_is_ok == false do
        ...
        time_is_ok = check_time()
        if (time_is_ok == true) then
            set_time()
            break
        end
    end
end
```

Root access to a routing node

Add IPTables rule: drop messages to time server

: iptables – A INPUT – d ADDRESS – j DROP

Attack Example 3: Spoofing



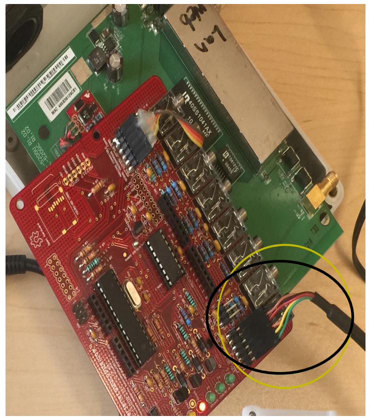
Sensor board

Data

Request

Communication board

Normal behavior



Find serial communication configuration (a handful common configs, a couple of hundreds total configs

One of the common configs worked in our case

Use USB to 6-pin serial connector from laptop to meter

Replay data request

Receive data on the laptop – data deleted from sensor board

Conclusion

- **IoT devices perform specific tasks**
  - Formalize their operations
  - Formalize the attacker
  - Perform automated analysis
  - Find real vulnerabilities

"Formal Security Analysis of Smart Embedded Systems",
Farid Molazem Tabrizi and Karthik Pattabiraman,
Annual Computer Security Applications Conference (ACSAC), 2016

Videos of attacks found by our technique:
http://www.ece.ubc.ca/~faridm/acsac.html