Program Analysis
Information Flow Analysis
(Part 1)

Prof. Dr. Michael Pradel
Software Lab, University of Stuttgart
Winter 2020/2021
Outline

1. Introduction
2. Information Flow Policy
3. Analyzing Information Flows

Mostly based on these papers:

Secure Computing Systems

■ Overall goal: Secure the data manipulated by a computing system

■ Enforce a security policy
  □ Confidentiality: Secret data does not leak to non-secret places
  □ Integrity: High-integrity data is not influenced by low-integrity data
Information Flow

■ Goal of information flow analysis:
Check whether information from one "place" propagates to another "place"
  □ For program analysis, "place" means, e.g., code location or variable

■ Complements techniques that impose limits on releasing information
  □ Access control lists
  □ Cryptography
"Places" in program that hold data

Secret information → Possible? → Untrusted place

Trusted information ← Possible? ← Confidentiality

Integrity
Example: Confidentiality

Credit card number should not leak to visible

```javascript
var creditCardNb = 1234;
var x = creditCardNb;
var visible = false;
if (x > 1000) {
    visible = true;
}
```
Example: Confidentiality

Credit card number should not leak to visible

```javascript
var creditCardNb = 1234;
var x = creditCardNb;
var visible = false;
if (x > 1000) {
  visible = true;
}
```

Secret information propagates to `x`
Secret information (partly) propagates to `visible`
Example: Integrity

userInput should not influence who becomes president

```javascript
var designatedPresident = "Michael";
var x = userInput();
var designatedPresident = x;
```
Example: Integrity

userInput should not influence who becomes president

```javascript
var designatedPresident = "Michael";
var x = userInput();
var designatedPresident = x;
```

Low-integrity information propagates to high-integrity variable
Example: Integrity

userInput should not influence who becomes president

```javascript
var designatedPresident = "Michael";
var x = userInput();
if (x.length === 5) {
  var designatedPresident = "Paul";
}
```
Example: Integrity

userInput should not influence who becomes president

```javascript
var designatedPresident = "Michael";
var x = userInput();
if (x.length === 5) {
    var designatedPresident = "Paul";
}
```

Low-integrity information propagates to high-integrity variable
Confidentiality vs. Integrity

Confidentiality and integrity are dual problems for information flow analysis

(Focus of this lecture: Confidentiality)
Tracking Security Labels

How to analyze the flow of information?

- Assign to each value some meta information that tracks the secrecy of the value
- Propagate meta information on program operations
Example

```javascript
var creditCardNb = 1234;
var x = creditCardNb;
var visible = false;
if (x > 1000) {
    visible = true;
}
```

---. contains a secret value
Non-Interference

Property that information flow analysis aims to ensure:

Confidential data does not interfere with public data

- Variation of confidential input does not cause a variation of public output
- Attacker cannot observe any difference between two executions that differ only in their confidential input