Hybrid information flow analysis against web tracking

Thomas Jensen, Inria

Keynote, CRISIS'2017
Overview

1. Web tracking.
2. Knowledge-based information flow analysis.
3. Browser randomisation.
Thanks to EU ePrivacy Directive
Web tracking

Bigger browsing profiles

= increased value for trackers
= reduced privacy for users

(Hypothetical tracking relationships only.)
Browser extensions to the rescue?

AdBlockPlus: blocks scripts/requests only from known advertisement companies

Ghostery: blocks scripts/requests only from known tracking companies

• No protection from tracking by other companies
• No protection from tracking by the main (first-party) web site
Tracking by storing identity

**Cookies** are used to **track repeated visits** to a site.
Tracking by finger-printing

Browser and operating system properties are used to track repeated visits to a site.
Panopticlick: an early study on fingerprinting

- A study from 2010 showed that 83.6% of browser fingerprints were unique among the 500,000 browsers tested.

- **Fingerprints**: HTTP headers, browser and OS features, language, plug-ins, fonts, …
Protection mechanisms

Stateful tracking: well-known and being addressed at different levels:

- Third-party cookie blocking
- EU e-Privacy directive
- Non-interference analysis for JavaScript

Stateless tracking: less clear

- IP address tracking
- Web browser finger-printing
2. Knowledge-based information flow analysis
Use information flow to spot web tracking

Scripts can access information about configurations

**Goal**: Estimate the information that a script learns about configurations.

But some information leakage is reasonable!

Scripts may need information about browser and execution environment for optimal user experience.
# Information flow control

<table>
<thead>
<tr>
<th>Non-interference</th>
<th>Attacker knowledge</th>
<th>Quantitative information flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secret input does not flow into public output</td>
<td>What information about the secret flows to output in an execution.</td>
<td>How much information about secrets is leaked?</td>
</tr>
<tr>
<td>Yes/no ?</td>
<td>Secret &gt; 42</td>
<td>Leaks 17 bits</td>
</tr>
</tbody>
</table>
Attacker knowledge

```javascript
var x = 0;
if (name == "Firefox") {
  x = 1;
}
else {
  if (fonts == fontsSet1) {
    x = 2;
  }
}
output x;
```

Depending on the browser, different executions leak different amount of information.
Non-interference

A program is termination-insensitive non-interferent if it produces the same output for all low-equivalent input.

Example with two binary secrets $h_1, h_2 \in \{0,1\}$
Secure and insecure executions

The following program is not secure - but it has secure executions.

```
if h1 = 1 then x = 1
else skip;
if h2 = 1 then low = 1
else low = x;
output low
```

Initial values:

- [low = 1, x = 1]
- [low = 1, x = 0]

Secure
Leaks information about secret
Monitoring information flow

Dynamic information flow control

- Label values with security labels.
- Propagate labels during execution.
- Halt execution before outputting secret values.

Non-interference needs "no-sensitive-upgrade" principle:
no assignment to public variables under secret control

\[
\text{[secret = 1]}
\]

\[
\text{if secret = 1 then}
\]

\[
\text{output = 1}
\]

\[
\text{else output = public;}
\]

\[
\text{return output}
\]
Hybrid information flow analysis

Combine dynamic IF control with a static analysis

- statically analyse the non-executed branches
- admits more executions but limited by the precision of the static analysis

```
if secret = 1 then
    output = 1
else output = public;
return output
```

[secret = 1, public = 1]

Static analysis says public : [1]

[Le Guernic’07, Russo & Sabelfeld’10,…]
Hybrid monitor of attacker knowledge

Define a hybrid monitor that tracks the knowledge of an attacker.

Use knowledge to

• enforce non-interference,
• accept more secure executions.

Solution:

• A domain for representing attacker knowledge.
• Monitor formalised as an operational semantics.
Attacker knowledge

Knowledge: What does an attacker learn about the input state by seeing a particular output.

Real attacker knowledge

Safe approximation of knowledge
Domain of knowledge

\[ \kappa: \text{Var} \rightarrow \mathbf{K} \]

\[ \mathbf{K} \triangleq \text{Env} \rightarrow \text{Value} \cup \{\top, \bot\} \]

The knowledge in a variable \( \kappa(x) \):

which initial environments leads to a particular value of \( x \)?

\( \top \) (nothing known about \( x \) for these envs)

0

1
The hybrid monitor

Defined as an operational reduction semantics:

\[(P, \rho, K) \Downarrow (\rho', K')\]

- \(\rho, \rho' : \text{Env} \cup \{\cdot\}\),
- \(K, K' : \text{Var} \to K\) labeling of variables with knowledge,
- \(\text{Env} : \text{Var} \to \text{Val}\) for dynamic analysis,
- \(\cdot\) "empty" environment for static analysis.
Semantic rules

Rules come in pairs: one dynamic and one static

Dynamic analysis computes real values and knowledge

\[
\text{IF}_\text{DYN} \quad \frac{C[e]_\rho = \alpha}{(\text{if } e \text{ then } c_{tt} \text{ else } c_{ff}, (\rho, \kappa))} \Downarrow (\rho', \kappa) \Downarrow (\cdot, \kappa_{\alpha})
\]

Static analysis called on-the-fly and propagates only knowledge

\[
\text{IF}_\text{STAT} \quad \frac{(c_{tt}, (\cdot, \kappa)) \Downarrow (\cdot, \kappa_{tt}) \quad (c_{ff}, (\cdot, \kappa)) \Downarrow (\cdot, \kappa_{ff})}{(\text{if } e \text{ then } c_{tt} \text{ else } c_{ff}, (\cdot, \kappa))} \Downarrow (\cdot, \Pi^F(e, \kappa_{tt}, \kappa_{ff}))
\]

Operator combining knowledge obtained in branches with knowledge from the condition
Enforcing non-interference

Before any output, the monitor checks non-interference:

- if non-interference then output is not modified.

\[
\text{OutNI} \quad \frac{(c, \text{init}(\rho)) \Downarrow (\rho', \kappa) \quad \rho'(x) = v \quad NI(\rho, \kappa(x), v)}{\langle c; \text{output } x, \rho \rangle \Downarrow v}
\]

The non-interference check is defined as:

\[
NI(\rho, K, v) \triangleq [\rho]_L \subseteq K^{-1}(v) \cup K^{-1}(\bot).
\]

set of low-equivalent env's
Example analysis

\[ h_1 = 0, h_2 = 1 \]

if \( h_1 = 1 \) then \( x = 1 \)
else skip;

if \( h_2 = 1 \) then \( \text{low} = 1 \)
else \( \text{low} = x \);

output \( \text{low} \)

Knowledge

\[ [\text{low} = 0, x = 0] \]

Initial

\( K(x) : \)

Static analysis

Static analysis of env's with \( h_2 = 0 \)

Static analysis of env's with \( h_1 = 1 \)

Use knowledge in \( x \)
Stock-taking

Knowledge-based information flow analysis

- computes a correct approximation of attacker knowledge,
- enforces non-interference,
- more permissive than existing hybrid monitors,
- can be combined with other monitors to enhance precision.
3. Browser randomisation against web tracking
How to enforce privacy?

The hybrid monitor evaluates how much a tracker learns about an individual user.

How to react when a script tracks too much?

- With 10 users, our Opera user will be uniquely identified.
- Halting the program or suppress output might still make you identifiable.
- Need collaboration between users.
A solution

• Users switch between configurations

• How many configurations are needed to guarantee privacy?

• Usability issue: users want to switch as little as possible

**Challenge**: find a distribution on configurations for all users such that
• Privacy is guaranteed
• Usability is maximised
Privacy: probability of guessing

If (name == “Opera”) x = A;
else x = B;
output x;

A priori

<table>
<thead>
<tr>
<th>Browser</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox</td>
<td>0.45</td>
</tr>
<tr>
<td>Chrome</td>
<td>0.45</td>
</tr>
<tr>
<td>Opera</td>
<td>0.10</td>
</tr>
</tbody>
</table>

A posteriori

<table>
<thead>
<tr>
<th>Browser</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox</td>
<td>0</td>
</tr>
<tr>
<td>Chrome</td>
<td>0</td>
</tr>
<tr>
<td>Opera</td>
<td>1</td>
</tr>
</tbody>
</table>

Probability $Pr^G$: guessing the secret in case of worst observation.

Stronger than definition based on Bayesian risk:

$$p(A) \max_i p(i|A) + p(B) \max_i p(i|B) = 0.55$$
Achieving privacy

Find a randomisation $R$ of user state such that when composed with tracking script $P$

\[ \Pr^G(R; P) < \text{privacy threshold}. \]

<table>
<thead>
<tr>
<th></th>
<th>&quot;Firefox&quot;</th>
<th>&quot;Opera&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>$id_1$</td>
<td>$x$</td>
<td>$1-x$</td>
</tr>
<tr>
<td>$id_2$</td>
<td>$1-y$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$o_1$</th>
<th>$o_2$</th>
<th>$o_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Firefox&quot;</td>
<td>$1/2$</td>
<td>$1/3$</td>
<td>$1/6$</td>
</tr>
<tr>
<td>&quot;Opera&quot;</td>
<td>$1/6$</td>
<td>$1/2$</td>
<td>$1/3$</td>
</tr>
</tbody>
</table>

$z = \Pr^G(R; P) = Ax + By + C$

$t = 0.6$
Adding usability

Users don't want to switch configuration:

- choose randomisation so that they get their original configuration as much as possible
- linear constraint on the randomisation matrix.
- problem reduces to linear programming.

<table>
<thead>
<tr>
<th>R</th>
<th>“Firefox”</th>
<th>“Opera”</th>
</tr>
</thead>
<tbody>
<tr>
<td>id₁</td>
<td>x</td>
<td>1-x</td>
</tr>
<tr>
<td>id₂</td>
<td>1-y</td>
<td>y</td>
</tr>
</tbody>
</table>

\[
\text{max } (x + y) \text{ s.t. } \\
0 \leq x \leq 1 \\
0 \leq y \leq 1 \\
Ax + By + C \leq t
\]
Wrapping up

Web tracking is done by different techniques
  • cookies, other browser storage, fingerprinting

Information flow analysis against fingerprinting
  • Hybrid information flow monitoring with static analysis
  • Computes tracker knowledge.

Enforcing browser anonymity
  • Randomisation of configurations
  • Combining privacy and usability
Based on a true story...

