P4BID: INFORMATION FLOW CONTROL IN P4

PLDI’22

Karuna Grewal, Loris D’Antoni, Justin Hsu
SHIFTING TRENDS

Fixed Function Switch
SHIFTING TRENDS

Fixed Function Switch → Shift → Programmable Switch
SHIFTING TRENDS

Fixed Function Switch → Shift → Programmable Switch

P4 language
SHIFTING TRENDS

Fixed Function Switch → Shift → Programmable Switch

P4 language

Key-Value Store
Load Balancer
CUSTOMIZING A SWITCH

Parser

Processing Pipeline

Deparser

101010110101

X

Y

1010 + 011 + 110

X’

Y

Z’
A NEW CHALLENGE...

Programming Errors → Information Leak
A NEW CHALLENGE...

Programming Errors → Information Leak

Incoming Packet Header
A NEW CHALLENGE...

Programming Errors $\rightarrow$ Information Leak

\[
\text{public\_field} := \text{private\_field}
\]
A NEW CHALLENGE...

Programming Errors → Information Leak

public_field := private_field

Incoming Packet Header

Outgoing Packet Header
if (appID == foo) {
    priority = fast;
}

if (appID == bar) {
    priority = slow;
}
if (appID == foo) {
    priority = fast;
}

App ID: foo
Priority: fast

App ID: bar
Priority: slow

Host: 1.2.3.4
Host: 6.7.8.9

App ID: foo
Priority: fast

App ID: bar
Priority: slow

Host: 1.2.3.4
Host: 6.7.8.9
if (appID == foo) {
    priority = fast;
}
if (appID == foo) {
    priority = fast;
}

Having appID: foo can increase my priority
Having appID: foo can increase my priority

```java
if (appID == foo) {
    priority = fast;
}
```

App ID: foo
Priority: fast

App ID: bar
Priority: slow

Host: 1.2.3.4
Host: 6.7.8.9
if (appID == foo) {
    priority = fast;
}

App ID: foo
Priority: fast

Host: 1.2.3.4

App ID: bar
Priority: slow

Host: 6.7.8.9
if (appID == "foo") {
    priority = "fast";
}

App ID: foo
Priority: fast

App ID: bar
Priority: slow

Host: 1.2.3.4
Host: 6.7.8.9
if (appID == foo) {
    priority = fast;
}

Host: 1.2.3.4
App ID: foo
Priority: fast

Host: 6.7.8.9
App ID: bar
Priority: slow

Public Network

switch

Dest IP

Priority

AppID

6.7.8.9

?
if (appID == foo) {
    priority = fast;
}

Public Network

if (appID == foo) {
    priority = fast;
}
if (appID == foo) {
    priority = fast;
}

if (appID == bar) {
    priority = slow;
}
CONTRIBUTIONS
P4BID

Information flow control type system for P4

Implement P4BID on top of P4’s reference compiler

Encode networking properties as IFC properties
A QUICK REVIEW OF INFORMATION FLOW CONTROL
INFORMATION FLOW CONTROL REFRESHER

Security lattice

HIGH

↓

LOW
INFORMATION FLOW CONTROL REFRESHER

Security lattice

Label the variables

AppID  \[\rightarrow\] HIGH \(\equiv\) UNTRUSTED

Dest IP \[\rightarrow\] LOW \(\equiv\) TRUSTED

Priority
Security lattice

Label the variables

Security types

INFORMATION FLOW CONTROL REFRESHER
if ( HIGH == 1 ) {
    HIGH := LOW;
}
LOW := LOW + 1;
if ( HIGH == 1 ) {
    HIGH := LOW;
}
LOW := LOW + 1;
if ( HIGH == 1 ) {
    HIGH := LOW;
}
LOW := LOW + 1;
```plaintext
if ( HIGH == 1 ) {
    HIGH := LOW;
}
LOW := LOW + 1;
```
TYPE SYSTEM GUARANTEES

if ( HIGH == 1 ) {
    HIGH := LOW;
}
LOW := LOW + 1;

HIGH: 1
LOW: 5

HIGH: 2
LOW: 5

HIGH: 5
LOW: 6
if (HIGH == 1) {
   HIGH := LOW;
}
LOW := LOW + 1;
if ( HIGH == 1 ) {
    HIGH := LOW;
} else {
    LOW := LOW + 1;
}
INFORMATION FLOW CONTROL

CHALLENGES IN P4
P4 LANGUAGE: EXAMPLE APP2PRIORITY
P4 LANGUAGE: EXAMPLE APP2PRIORITY

Action Declaration

```
action set_priority(int new_priority) {
    hdr.priority = new_priority;
}
```
**P4 LANGUAGE: EXAMPLE APP2PRIORITY**

### Action Declaration

```c
action set_priority(int new_priority) {
    hdr.priority = new_priority;
}
```

### Table Declaration

(Installed at Runtime)

```c
table app2priority {
    key = {
        hdr.appID;
    }
    actions = {
        set_priority;
    }
}
```
P4 LANGUAGE: EXAMPLE APP2PRIORITY

Action Declaration

```p4
class hdr {
    appID: int;
    priority: int;
}

action set_priority(int new_priority) {
    hdr.priority = new_priority;
}
```

Table Declaration (Installed at Runtime)

```p4
table app2priority {
    key = {
        hdr.appID;
    }
    actions = {
        set_priority;
    }
}
```

Statement

```p4
apply {
    app2priority.apply();
}
```
LEAKS IN TABLES
LEAKS IN TABLES

table match_action {
    key = { high_key; }
    actions = { modify_low_field; }
}

LEAKS IN TABLES

table match_action {
  key = { high_key; }
  actions = { modify_low_field; }
}

if (high_key == foo) {
  modify_low_field();
}
else if (high_key == bar) {
  skip;
}
LEAKS IN TABLES

```c
table match_action {
    key = { high_key; }
    actions = { modify_low_field; }
}

if (high_key == foo) {
    modify_low_field();
}
else if (high_key == bar) {
    skip;
}
```
LEAKS IN TABLES

```plaintext
table match_action {
    key = { high_key; }
    actions = { modify_low_field; }
}

if (high_key == foo) {
    modify_low_field();
}
else if (high_key == bar) {
    skip;
}
```

- Branch on a HIGH variable
- Action writes to a LOW variable
REVISITING RUNNING EXAMPLE

Public Network

- App ID: foo
  - Priority: fast
  - Host: 1.2.3.4

- App ID: bar
  - Priority: slow
  - Host: 6.7.8.9
SECURITY TYPES

Packet Header

- **ApplD**: UNTRUSTED
- **Dest IP**: TRUSTED
- **Priority**
SECURITY TYPES

Packet Header

<table>
<thead>
<tr>
<th>AppID</th>
<th>UNTRUSTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest IP</td>
<td>TRUSTED</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
</tr>
</tbody>
</table>

AppID: \(<\text{AppID}_t, \text{UNTRUSTED}\>\),
DestIP: \(<\text{DestIP}_t, \text{TRUSTED}\>\),
Priority: \(<\text{Priority}_t, \text{TRUSTED}\>\)
SECURITY TYPES

Packet Header

<table>
<thead>
<tr>
<th>AppID</th>
<th>UNTRUSTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest IP</td>
<td>TRUSTED</td>
</tr>
<tr>
<td>Priority</td>
<td></td>
</tr>
</tbody>
</table>

```c
hdr {
  AppID:  <AppID_t, UNTRUSTED>,
  DestIP: <DestIP_t, TRUSTED>,
  Priority: <Priority_t, TRUSTED>
}
```
Table Declaration

table app2priority {
    key = { hdr.appID; }
    actions = { set_priority; }
}

BUGGY TABLE
Table Declaration

```c
#define foo
#define bar

table app2priority {
    key = { hdr.appID; }
    actions = { set_priority; }
}

if (hdr.appID == foo) {
    set_priority(prio1);
}
else if (hdr.appID == bar) {
    set_priority(prio2);
}
...```

---

16
Table Declaration

```plaintext
table app2priority {
    key = { hdr.appID; }
    actions  = { set_priority; }
}
```

```
if (hdr.appID == foo) {
    set_priority(prio1);
}
else if (hdr.appID == bar) {
    set_priority(prio2);
}
...
```

Branch on an UNTRUSTED variable
Table Declaration

```plaintext
table app2priority {
    key = { hdr.appID; }  
    actions = { set_priority; }  
}
```

if (hdr.appID == foo) {
    set_priority(prio1);
}
else if (hdr.appID == bar) {
    set_priority(prio2);
}
...

Branch on an UNTRUSTED variable

Action writes to a TRUSTED variable
DETECTING LEAKS IN P4BID
TYPING JUDGEMENT

$$\Gamma \vdash_{pc} stmt \rightarrow \Gamma'$$
Initial Typing Context

\[ \Gamma 
\vdash_{pc} stmt \rightarrow \Gamma' \]
TYPING JUDGEMENT

Initial Typing Context

\[ \Gamma \vdash_{pc} stmt \vdash \Gamma' \]

Final Typing Context

appID: <int, UNTRUSTED>
destIP: <int, TRUSTED>
appID: <int, UNTRUSTED>
TYPING JUDGEMENT

Initial Typing Context

\[
\Gamma \vdash_{pc} stmt \vdash \Gamma'
\]

no writes to variables below PC

Final Typing Context

appID: <int, UNTRUSTED>

destIP: <int, TRUSTED>
Suppose

$$\Gamma \vdash_{pc} stmt \vdash \Gamma'$$
NON-INTERFERENCE THEOREM

Suppose

$$\Gamma \vdash_{pc} stmt \vdash \Gamma'$$

then

stmt is non-interfering, i.e, no High to Low information flow
LEAKY TABLE

table app2priority {
    key = { hdr.appID; }
    actions = { set_priority; }
}

\[ \Gamma \vdash_{LOW} \text{app2priority.apply()} \rightarrow \Gamma' \]

NOT provable!!
LEAKY TABLE

table app2priority {
  key = { hdr.appID; }
  actions = { set_priority; }
}

\[ \Gamma \vdash_{LOW} \text{app2priority.apply()} \rightarrow \Gamma' \]

NOT provable!!
LEAKY TABLE

```
{table app2priority {
  key = {hdr.appID; } hdr.destIP
  actions = { set_priority; }
}
}
```

\[ \Gamma \vdash_{LOW} \text{app2priority.apply()} \rightarrow \Gamma' \]

NOT provable!!
LEAKY TABLE

table app2priority {
    key = { hdr.appID; } hdr.destIP
    actions = { set_priority; }
}

\[ \Gamma \vdash_{LOW} \text{app2priority.apply()} \rightarrow \Gamma' \]
SEE THE PAPER FOR MORE ON...

Full type system
Non-interference theorem and proof
Several network scenarios using IFC, including isolation

OPEN QUESTIONS ...

Recirculation
Inter-Packet Leaks