Alias-aware propagation of pattern-based properties in PHP applications

François Gauthier and Ettore Merlo
francois.gauthier@polymtl.ca

Previous work

- Propagation of security properties through traversal of **static** security patterns.
- Inter-procedural, flow and *security*-sensitive, linear time complexity.

Intra-procedural

```
if(user_can(‘read’)){
  // Execute privileged
  // code
}
else{
  // Print error and
  // exit
}
```

Inter-procedural

```
if(user_can(‘read’))
  read();

function read(){
  return sql_query(...);
}
```

Extension

- Propagation of security properties through traversal of **dynamic** security patterns.
- E.g. aliasing of security patterns to **variables** and **parameters**.

Variable aliasing

```
$canRd = user_can(‘read’);

if($canRd)
  read();

function read(){
  return sql_query(...);
}
```

Parameter aliasing

```
write(user_can(‘write’));

function write($perm){
  if($perm){
    sql_query(...);
  }
  return;
}
```

Analysis implementation

Manual implementation

Pros:

- Efficient
- Fine-tuning

Cons:

- Difficult to implement
- Hard to modify

Datalog + BDDs

Pros:

- Incremental implementation
- Fast prototyping

Cons:

- BDD optimization is NP-Hard.
- Memory-consuming

A simple Datalog program

Pat2Var(v, f_n, f_i, p) :- PatAssign(v, f_n, f_i, p).

Pat2Var(v, f_n, f_i, p) :- VarAssign(v, f_n, f_i, w),
Pat2Var(w, f_n, f_i, p).

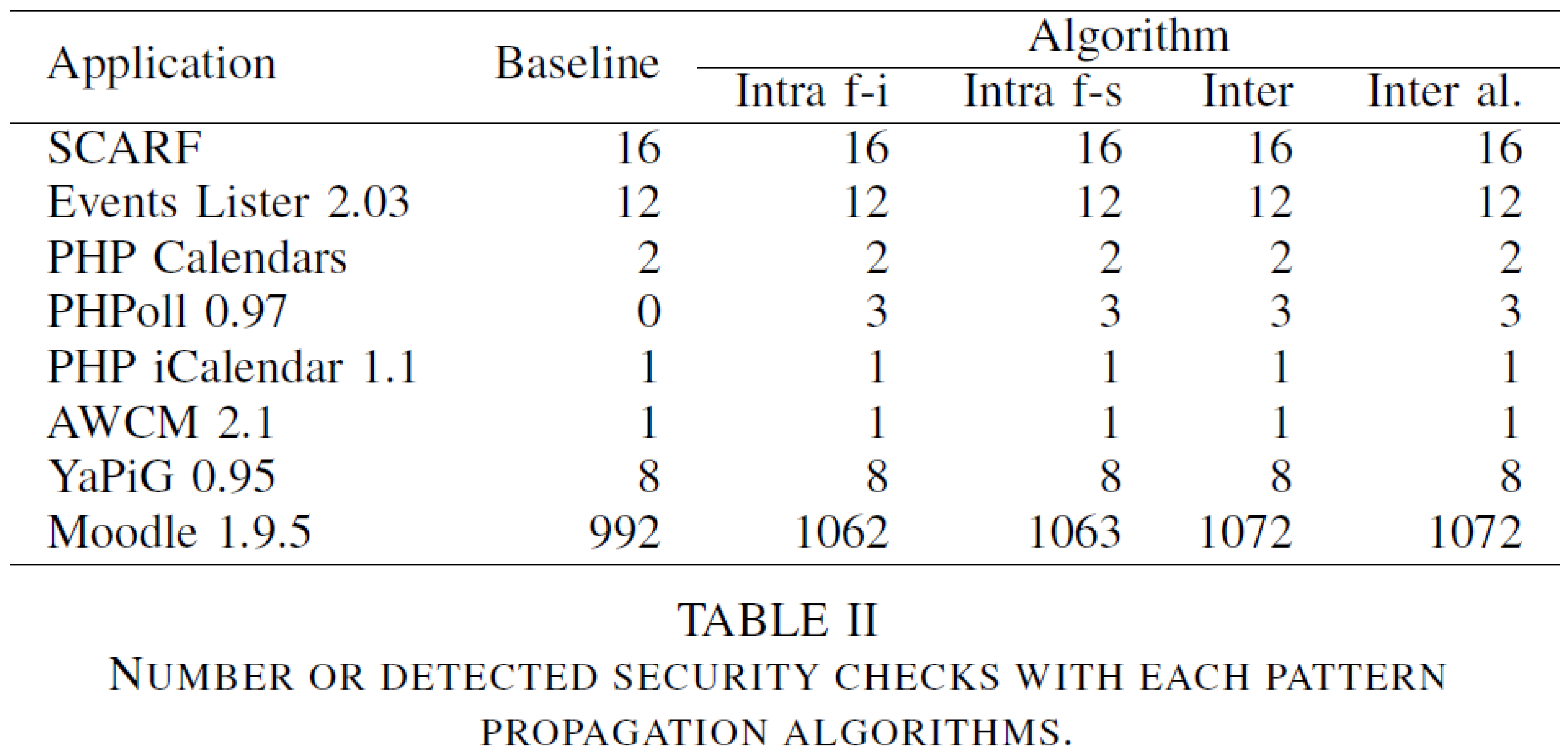
```
$canRd=user_can('read');
$canWt=user_can('write');

if($op == "write")
  $perm = $canWt;
else
  $perm = $canRd;

foo($perm);
```

PatAssign (*canRd*, func, file, "read")
PatAssign (*canWt*, func, file, "write")

VarAssign(*perm*, func, file, *canWt*)

VarAssign(*perm*, func, file, *canRd*)

Results – Security patterns



Application	Baseline	Algorithm			
		Intra f-i	Intra f-s	Inter	Inter al.
SCARF	16	16	16	16	16
Events Lister 2.03	12	12	12	12	12
PHP Calendars	2	2	2	2	2
PHPoll 0.97	0	3	3	3	3
PHP iCalendar 1.1	1	1	1	1	1
AWCM 2.1	1	1	1	1	1
YaPiG 0.95	8	8	8	8	8
Moodle 1.9.5	992	1062	1063	1072	1072

TABLE II
NUMBER OR DETECTED SECURITY CHECKS WITH EACH PATTERN PROPAGATION ALGORITHMS.

Conclusion

- Datalog allows for fast and incremental development of data-flow algorithms.

- **Intra-procedural** (variable aliasing) analysis yields the most significant recall improvements.
- **Flow-sensitivity** was **not worth** the increased time complexity.
- **Inter-procedural** (parameter aliasing) analysis further improves the recall of our analysis.

- Overall, the presented extension identifies significantly more security checks with very few false positives.

What next?

- Automatic detection of vulnerabilities based on the reverse-engineered security model!
- See **ACMA**, the Access Control Model Analyzer at **WCRE 2012**.

This research work is supported by the NSERC under the Discovery Grant Program.

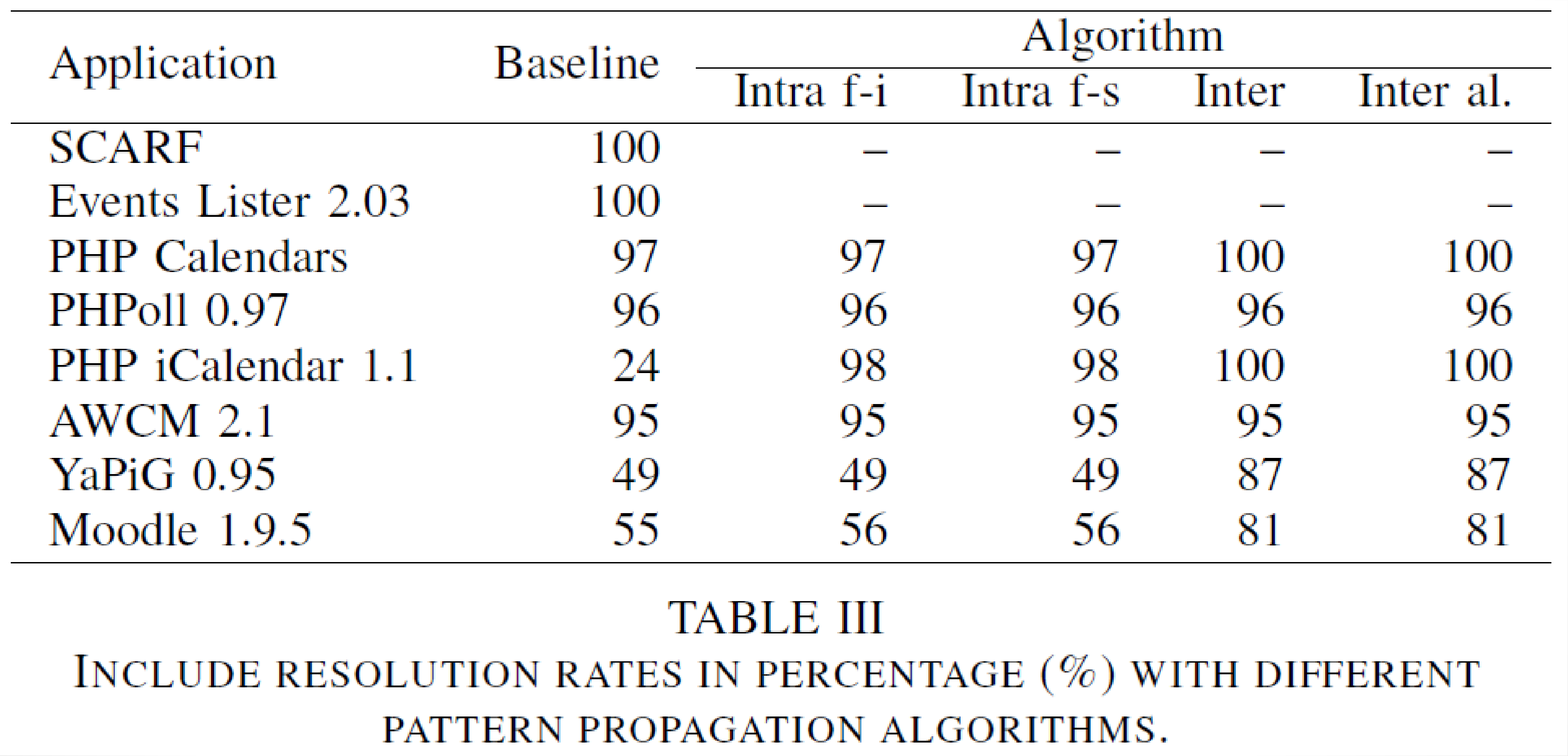
CRSNG
NSERC

F.G. acknowledges support from the FQRNT scholarship program.

Fonds de recherche
Nature et
technologies
Québec

Results – Include resolution

```
include(″__DIR__/lib″+ $lang +″/index.php″);
```



Application	Baseline	Algorithm			
		Intra f-i	Intra f-s	Inter	Inter al.
SCARF	100	–	–	–	–
Events Lister 2.03	100	–	–	–	–
PHP Calendars	97	97	97	100	100
PHPoll 0.97	96	96	96	96	96
PHP iCalendar 1.1	24	98	98	100	100
AWCM 2.1	95	95	95	95	95
YaPiG 0.95	49	49	49	87	87
Moodle 1.9.5	55	56	56	81	81

TABLE III

INCLUDE RESOLUTION RATES IN PERCENTAGE (%) WITH DIFFERENT PATTERN PROPAGATION ALGORITHMS.