

Visualizing Type Qualifier Inference with Eclipse

David Greenfieldboyce Jeffrey S. Foster
University of Maryland

Motivation

- Software has bugs
- Would like to allow programmers to specify and check additional program properties in a way that...
 - Programmers will accept
 - Lightweight
 - Scales to large programs
 - Solves many different problems

2

Type Qualifiers

- Create new analyses by extending standard type systems – C, Java, ML...
 - Programmers already use types
 - Programmers understand types
 - Just get programmers to write down a little more...

<code>const int</code>	ANSI C
<code>ptr(tainted char)</code>	Security vulnerabilities
<code>int → ptr(open FILE)</code>	File operations

3

Example: Format String Vulnerabilities

- I/O functions in C use format strings

```
printf("Hello!");           Hello!
printf("Hello, %s!", name); Hello, name !
```
- Instead of

```
printf("%s", name);
```

Why not
`printf(name);` ?

4

Format String Attacks

- Adversary-controlled format specifier

```
name := <data-from-network>
printf(name); /* Oops */
```

 - Attacker sets name = "%s%s%s" to crash program
 - Attacker sets name = "...%n..." to write to memory
- Particular bug not too common any more
 - People know to look out for them
- Similar issues still exist
 - User/kernel pointers in Linux kernel
 - But more complicated to explain...

5

Using Tainted and Untainted

- Add qualifier annotations to library functions

```
int printf(untainted char *fmt, ...)
tainted char *getenv(const char *)
```

tainted = may be controlled by adversary
untainted = must not be controlled by adversary
- Use subtyping to express relationships between qualifiers

6

Subtyping

```
void f(tainted int);
untainted int a;
f(a);
```

OK

f accepts **tainted** or **untainted** data

untainted ≤ **tainted**

untainted < **tainted**

```
void g(untainted int);
tainted int b;
f(b);
```

Error

g accepts only **untainted** data

tainted ≠ **untainted**

7

Demo Eclipse interface...

- show `taint1.c` file
- run `cqual` on it
- click along path and explain type inference

8

Type Qualifier Inference

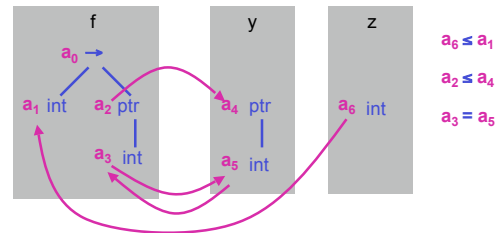
- Two kinds of qualifiers
 - Explicit qualifiers: **tainted**, **untainted**, ...
 - Unknown qualifiers: a_0, a_1, \dots
- Program yields constraints on qualifiers
 - $\text{tainted} \leq a_0$ $a_0 \leq \text{untainted}$
- Solve constraints for unknown qualifiers
 - Error if no solution

9

Constraint Generation

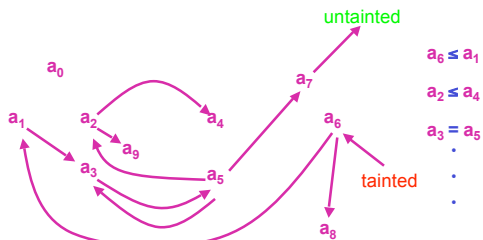
`ptr(int) f(x : int) = { ... }`

`y := f(z)`



10

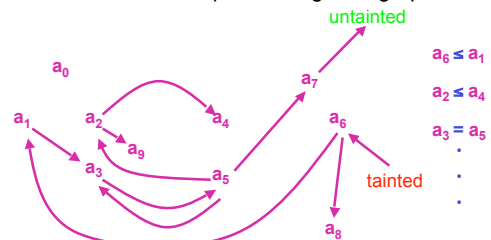
Constraints as Graphs



11

Satisfiability via Graph Reachability

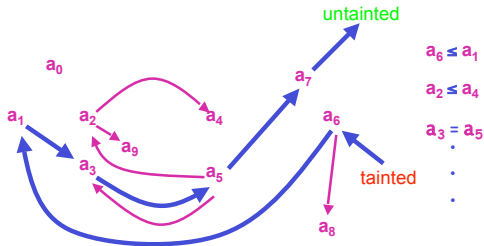
Is there an inconsistent path through the graph?



12

Satisfiability via Graph Reachability

Is there an inconsistent path through the graph?



13

Satisfiability in Linear Time

- Initial program of size n
 - Fixed set of qualifiers **tainted**, **untainted**, ...
- Constraint generation yields $O(n)$ constraints
 - Recursive abstract syntax tree walk
- Constraint solution takes $O(n)$ time
 - Works for semi-lattices, discrete p.o., products

14

About the Interface

- CQual run as a background process
 - Communicates with Eclipse via stdin/stdout
- Information sent in blocks as s-expressions
 - Could also use XML
 - Information only sent as needed by interface
- Eclipse controls display
 - More tightly integrated than previous version

15

Analyses Using CQual

- Const inference
 - A Theory of Type Qualifiers (PLDI'99)
- Format-string vulnerabilities
 - Detecting Format-String Vulnerabilities with Type Qualifiers (USENIX Sec'01)
- User/kernel pointer bugs in the Linux kernel
 - Johnson and Wagner, Finding User/Kernel Pointer Bugs with Type Inference (USENIX Sec'04)
- Others
 - Locking in linux kernel (PLDI'03), file operations, Y2K, initialization in kernel, ...

16

Future Work

- Integrate into Eclipse compilation process
- Improve error path heuristics
 - Where do we report an error?
 - Which path do we show the user?
- JQual – type qualifier analysis for Java
 - Basic version available real soon now

17

For More Information

<http://cqual.sourceforge.net>

(Contact us directly for Eclipse plugin)

18