

## Why aren't we better off than we were 13 years ago?

- Ignorance
- · C is difficult to use securely
  - Unsafe functions
  - Confusing APIs
- Even security aware programmers make mistakes.
- Security Knowledge has not been codified into the development process

16 August 2001 David Larochelle

### Automated Tools

нонононононо

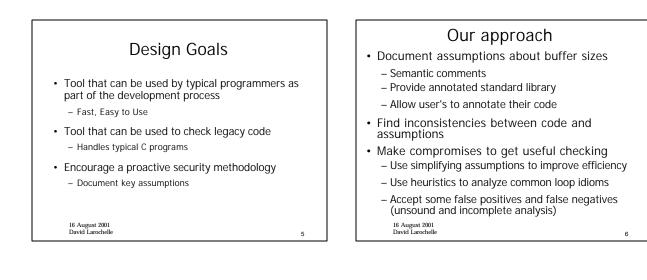
PBR.4211. R1.21

2

4

- Run-time solutions
  - StackGuard[USENIX 7], gcc bounds-checking, libsafe[USENIX 2000]
  - Performance penalty
  - Turns buffer overflow into a DoS attack
- Compile-time solutions static analysis
  - No run-time performance penalty
  - Checks properties of all possible executions

16 August 2001 David Larochell



### Implementation

- Extended LCLint
  - Open source checking tool [FSE '94] [PLDI '96]
  - Uses annotations
  - Detects null dereferences, memory leaks, etc.
- Integrated to take advantage of existing checking and annotations (e.g., modifies)
- Added new annotations and checking for buffer sizes

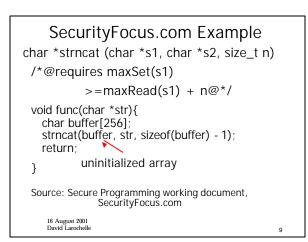
16 August 2001 David Larochelle

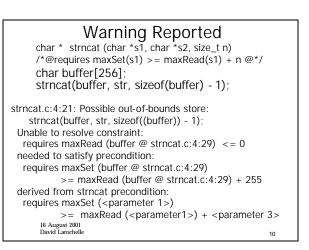
### Annotations

- · requires, ensures
- maxSet
  - highest index that can be safely written to
- maxRead

   highest index that can be safely read
- char buffer[100];
  - ensures maxSet(buffer) == 99

16 August 2001 David Larochelle

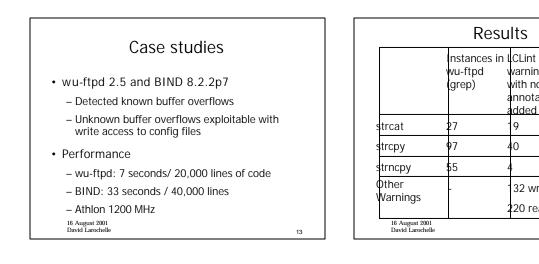


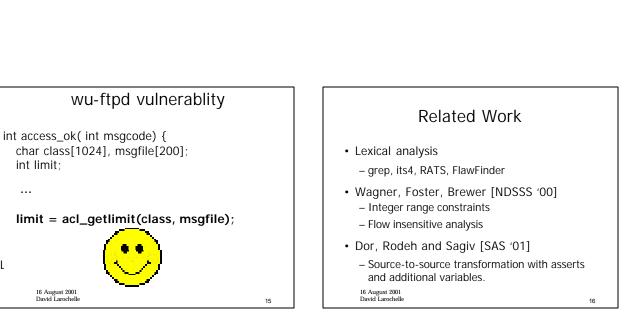


### Overview of checking Loop Heuristics Intraprocedural - But use annotations on called procedures and · Recognize common loop idioms global variables to check calls, entry, exit points Expressions generate constraints • Use heuristics to guess number of iterations - C semantics, annotations · Analyze first and last iterations • Axiomatic semantics propagates constraints Example: for (init; \*buf; buf++) · Simplifying rules (e.g. maxRead(str+i) ==> maxRead(str) - i) - Assume maxRead(buf) iterations - Model first and last iterations · Produce warnings for unresolved constraints 16 August 2001 David Larochell

11

16 August 2001 David Larochelle





...

LCLint

vith annotations

2

1

95 writes

66 reads

14

varning

varnings

nnotations

32 writes

20 reads

vith no

idded

9

# Impediments to wide spread adoption

- People are lazy
- Programmers are especially lazy
- Adding annotations is too much work (except for security weenies)
- Working on techniques for automating the annotation process

16 August 2001 David Larochelle

17

# Conclusion • 2014:??? • Will buffer overflows still be common? • Codify security knowledge in tools real programmers can use Beta version now available: <a href="http://lclint.cs.virginia.edu">http://lclint.cs.virginia.edu</a> David Larochelle David Evans larochelle@cs.virginia.edu Beta version available: <a href="http://lclint.cs.virginia.edu">http://lclint.cs.virginia.edu</a> David Larochelle David Evans larochelle@cs.virginia.edu