

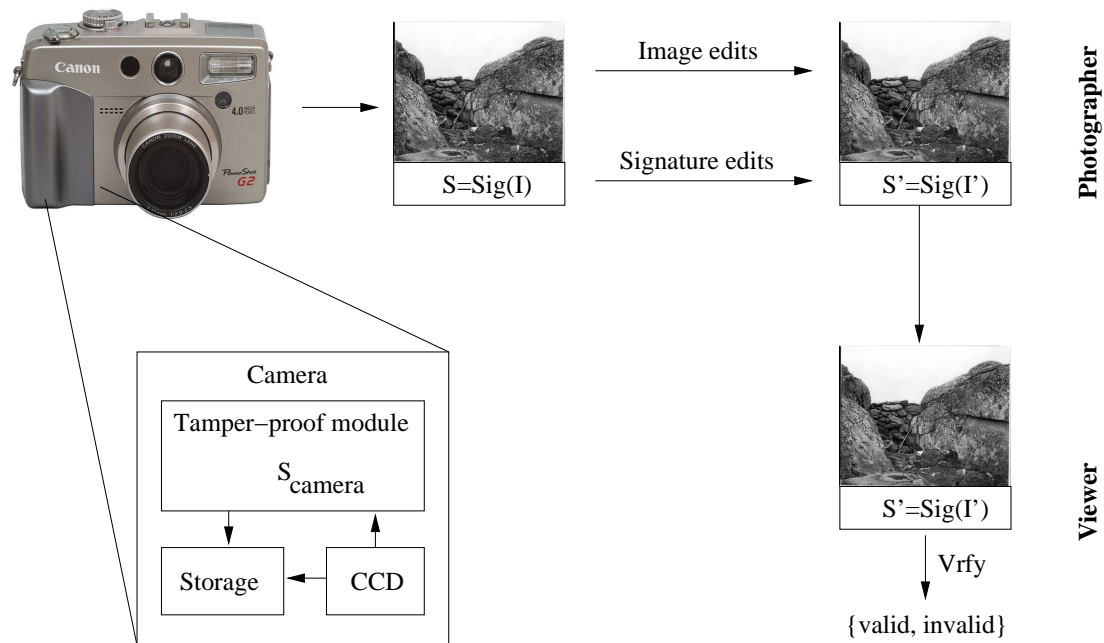
Finding Software Bugs Using Compile-Time Analysis

Rob Johnson

`rtjohnso@cs.sunysb.edu`

SUNY Stony Brook

Authenticating Reality



Trustworthy Cellphones

- “Networked credit cards”
- Need strong isolation
- Use Virtual Machines
- Support low-security apps, e.g. games
- Protect hi-security apps, e.g. digital wallet



+



Fraud

Software Security

- Most Internet attacks exploit software bugs
 - Slammer
 - Code Red
 - Sasser
 - Ramen
- Security bugs enable sophisticated attacks
 - Bring down Internet
 - DDoS attacks
 - Spam
 - Online fraud
- Fixing bugs \Rightarrow improved security

Example: Format String Bugs

```
int printf (char * fmt, ...);

int log_username (char * name)
{
    char *secret = "password";
    printf ("login from: ");
    printf (name);
}
```

- Problem: What if name is "%s %n"
- Then `printf` interprets un-initialized stack as pointers.
 - May crash program
 - May print out `secret`
 - May corrupt stack, giving attacker control

Finding Bugs at Compile-Time

- Auditing compiler
 - Knows about common coding mistakes
 - Searches source for unsecure code
 - Emits warnings to developer
- Catches bugs early – no patching
- Automatically audits code for bugs
- Enables non-specialists to write secure code
- Provides security guarantees

Previous Results: CQual/Oink

- Program auditing tool for C/C++
- Goal: analyze entire Debian Linux distribution
 - Eradicate format-string bugs
 - 1000s of packages
 - 250 MLOC
- Preliminary results (Berkeley):

Packages	400
Warnings	100
Real bugs	87

Real-world Impact

OMEGA: Next Gen Code Auditor

- Multiple languages: C, C++, Java, Fortran, Objective-C, Objective-C++, Ada
- Better analysis engines: type inference, model checking, type-state inference.
- Easier development of new bug-finding tools: simple language for specifying rules
- More flexibility: trade off soundness vs. completeness vs. speed
- Goal: 100s of bugs → 1000s of bugs
- Goal: widespread developer adoption

OMEGA: Strategy

- Leverage SUNY-SB GCC plugin [Callanan]
 - Support GCC languages “for free”
 - Support real-world code
 - Easy deployment path
- Make specification language look like target language
 - Easier for programmers to use
 - Easier to write correct specifications
 - Easier to support multiple languages

OMEGA: Targets

- Race conditions
- SQL injection bugs
- UNICODE bugs
- Buffer overflows
- Integer overflow bugs
- Input validation bugs
- Temporary file creation bugs
- User ID management errors
- Crypto usage mistakes
- Application-specific security bugs