

Security That Helps Attackers

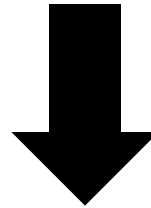
FORWARD Workshop, 2009

David Brumley

Carnegie Mellon University

`dbrumley@cmu.edu`

V
Vulnerable Program



P
Patched New Program

“Patches Help Security”
– Common Security Wisdom

Patches Can Help Attackers

– *Evil David*



Attacker



Attacker

Delayed Patch Attack

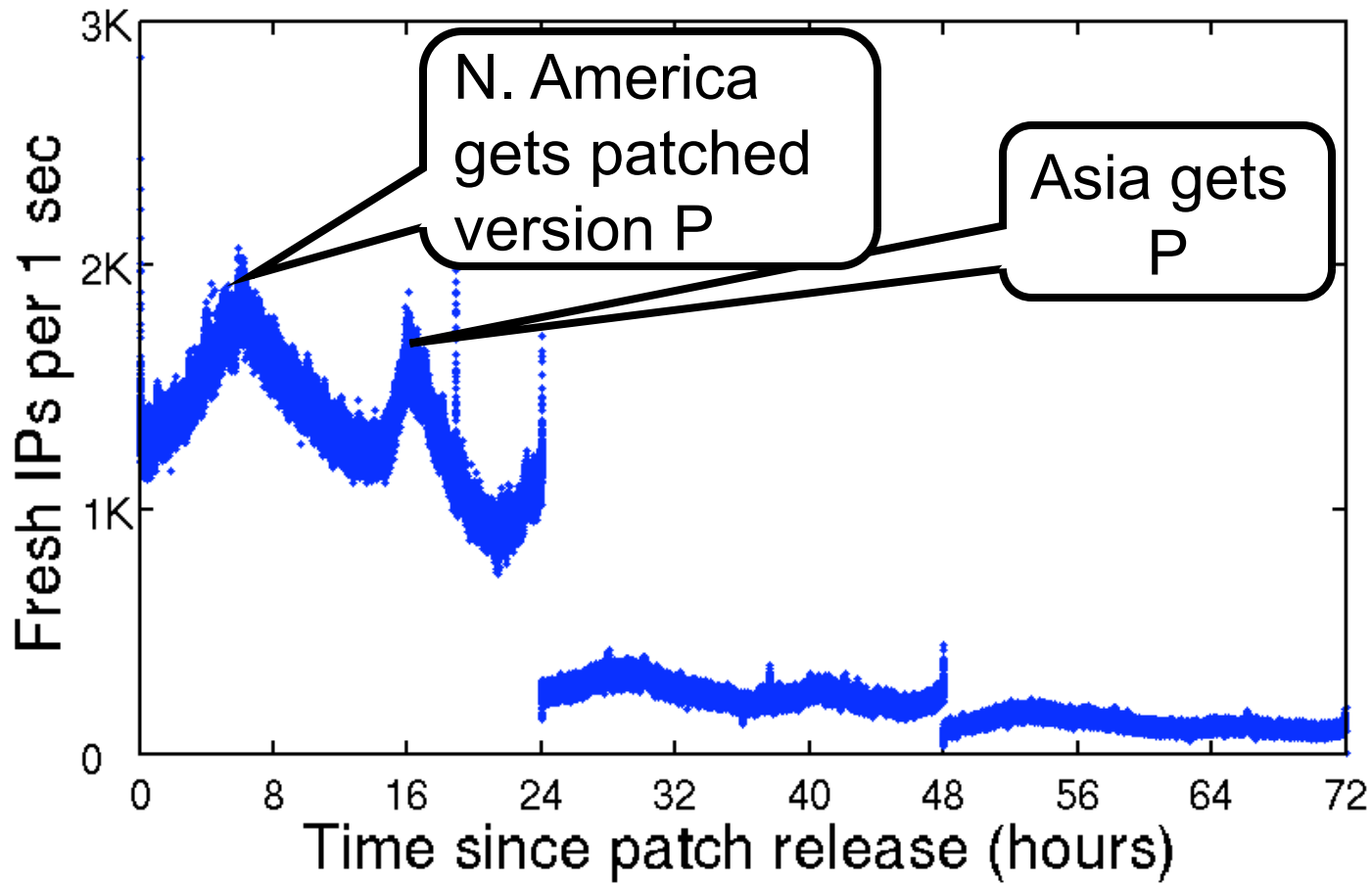


Patch Delay



Automatic Updates

ON



[Gkantsidis et al 06]

We can reverse engineer patch and generate *exploit* in *minutes* for input validation bugs



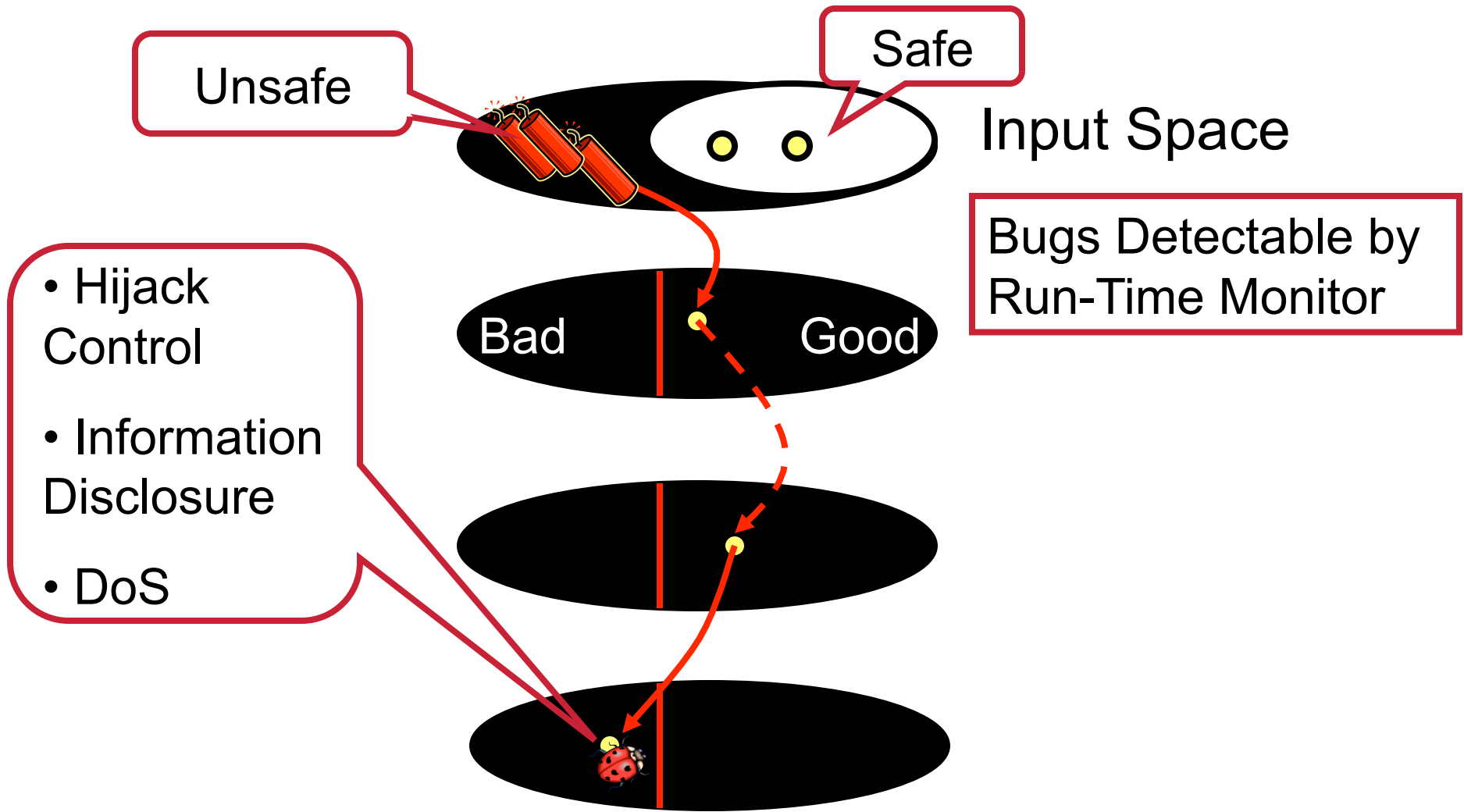
Minutes



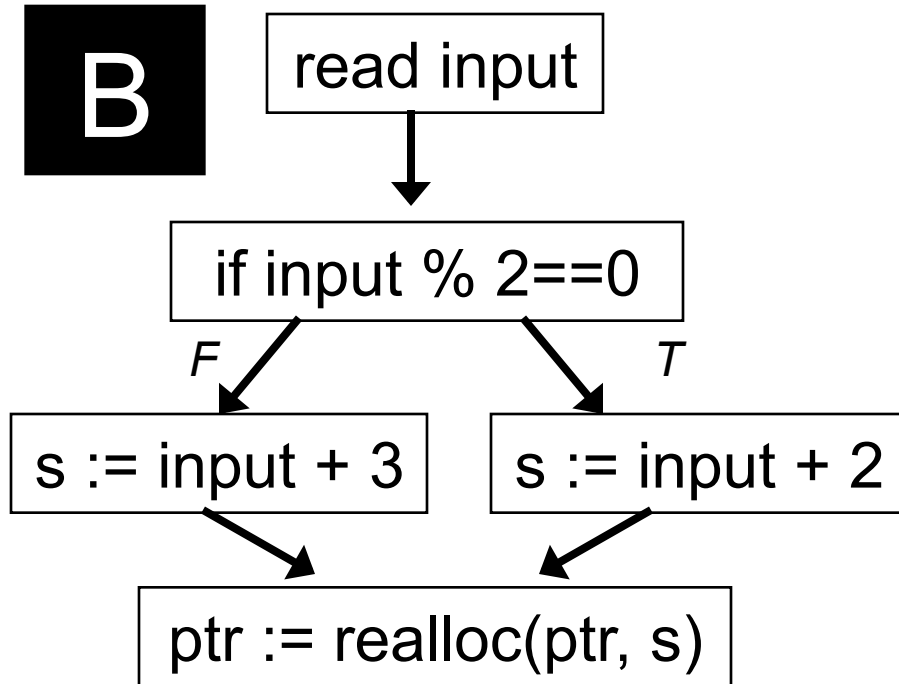
Evil David's Timeline

Input Validation Bugs

(Buffer Overflow, Format String, Integer Overflow, etc)



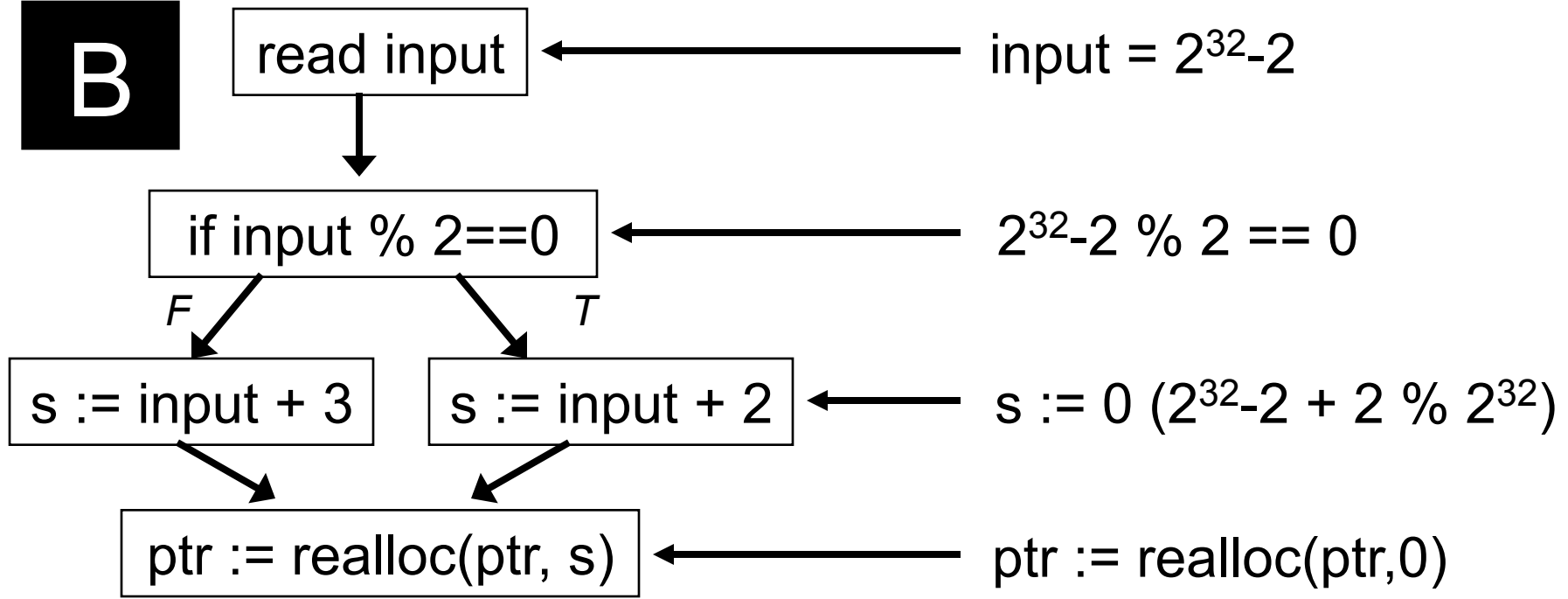
Running Example



- All integers unsigned 32-bits
- All arithmetic mod 2^{32}
- B is binary code

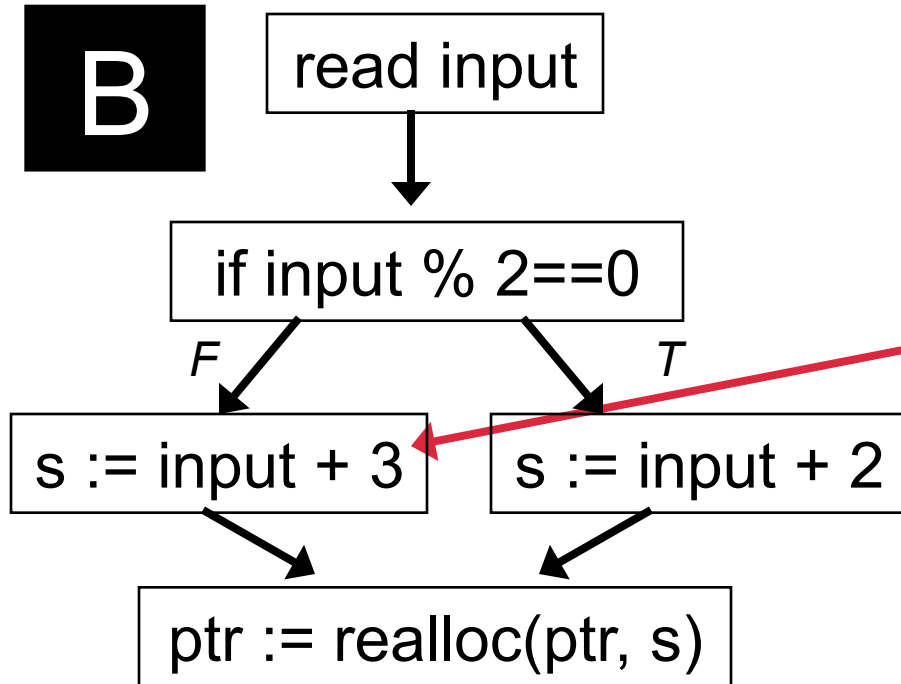
Running Example

B



Using **ptr** is a problem

Running Example

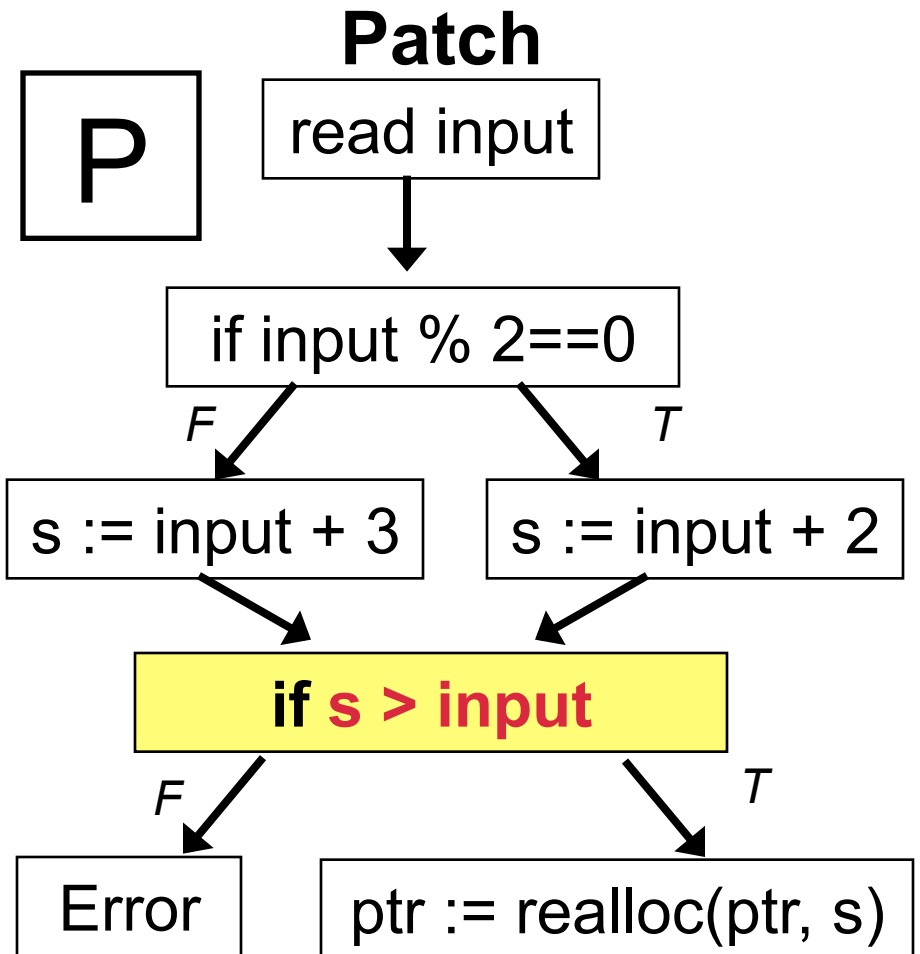
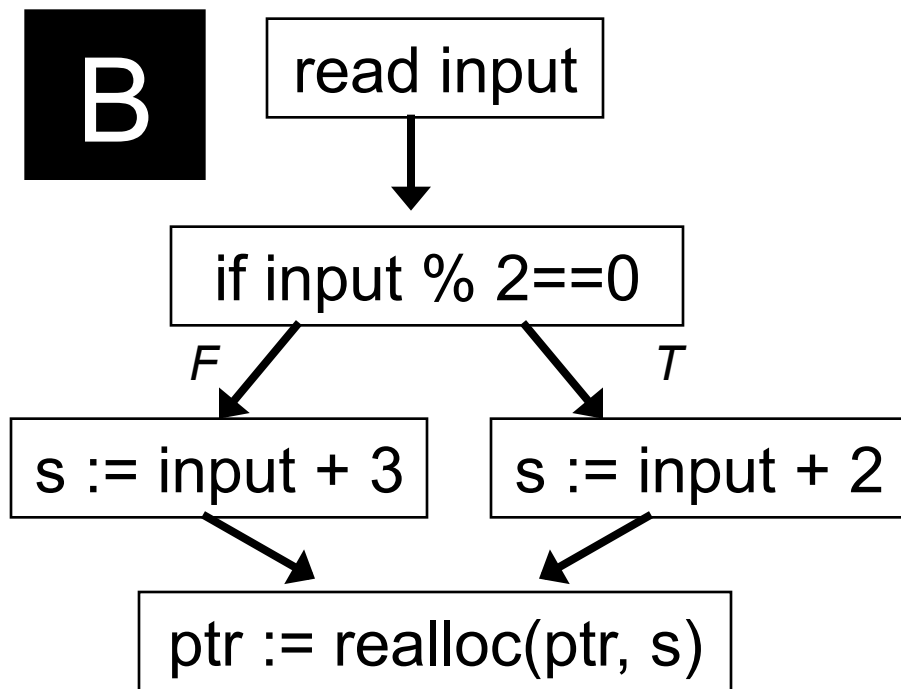


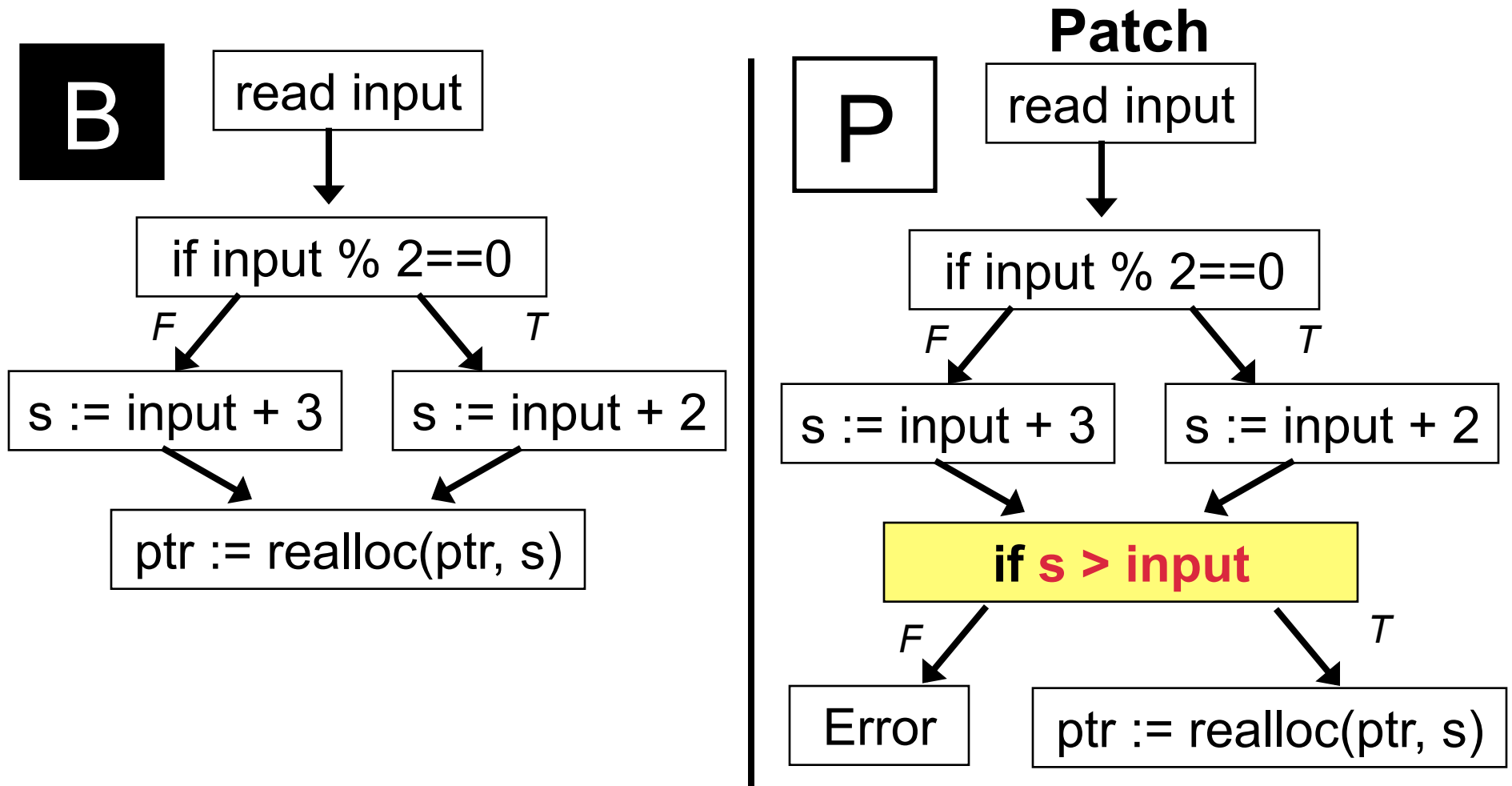
Wanted:

$s > input$

**Integer Overflow
when:**

$\neg(s > input)$

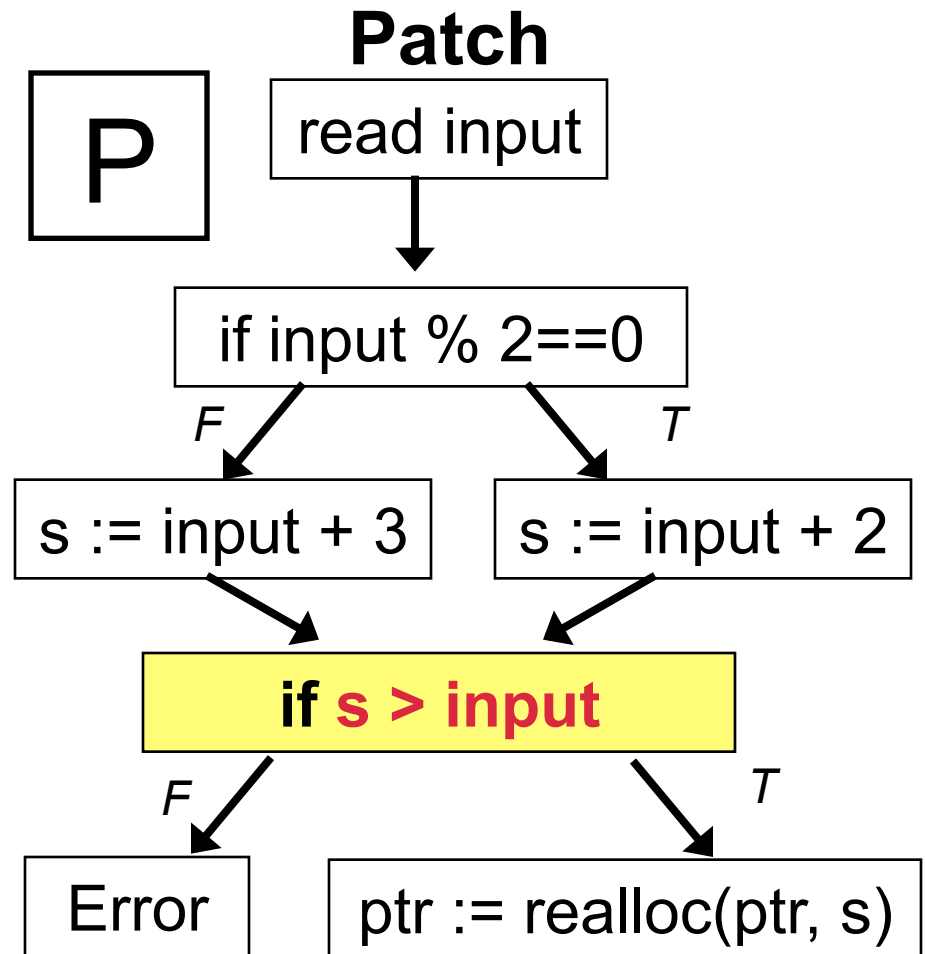




Exploits for B are inputs that fail
new safety condition check in P
 $(s > \text{input}) = \text{false}$

Exploit Generation

1. Diff B and P to identify location of new safety check
2. Create input that fails new checks in P
 - Formal methods
 - Significant details omitted
3. Verify input is exploit
4. Exploit others

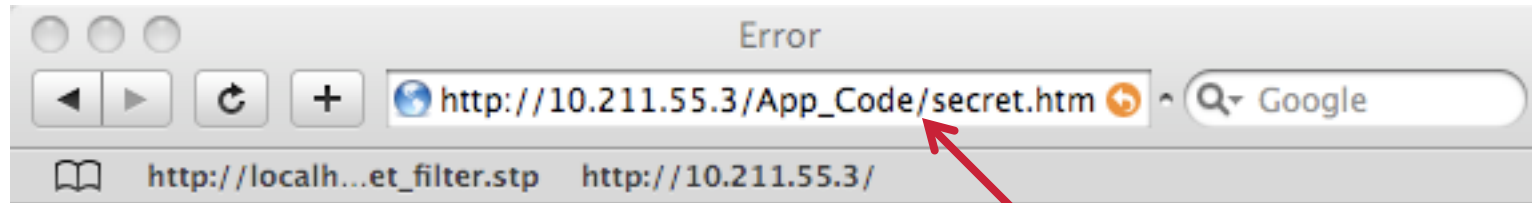


Result Overview

ASPNet_Filter	Information Disclosure	29 sec
GDI	Hijack Control	135 sec
PNG	Hijack Control	131 sec
IE COMCTL32 (B)	Hijack Control	456 sec
IGMP	Denial of Service	186 sec

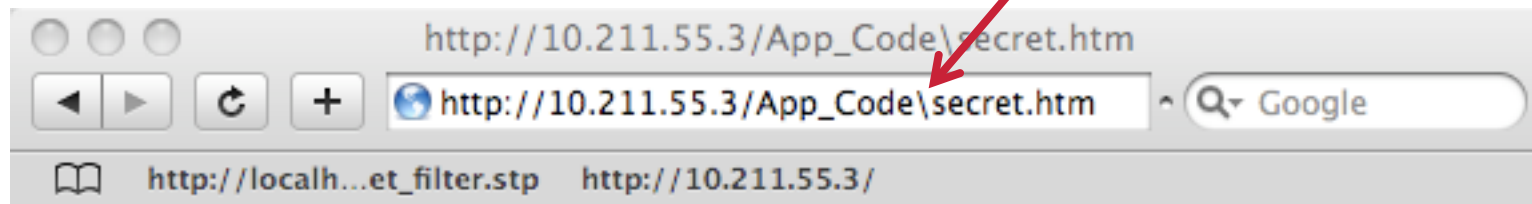
- **No public exploit for 3 out of 5**
- **Exploit unique for other 2**

Example/Demo



The system cannot find the file specified.

**Flip '/' to '\'
to reveal hidden
files**

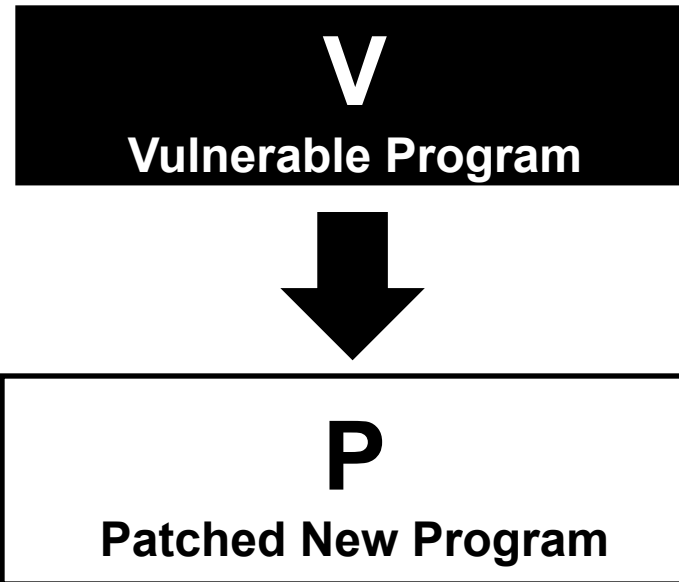


You shouldn't see me

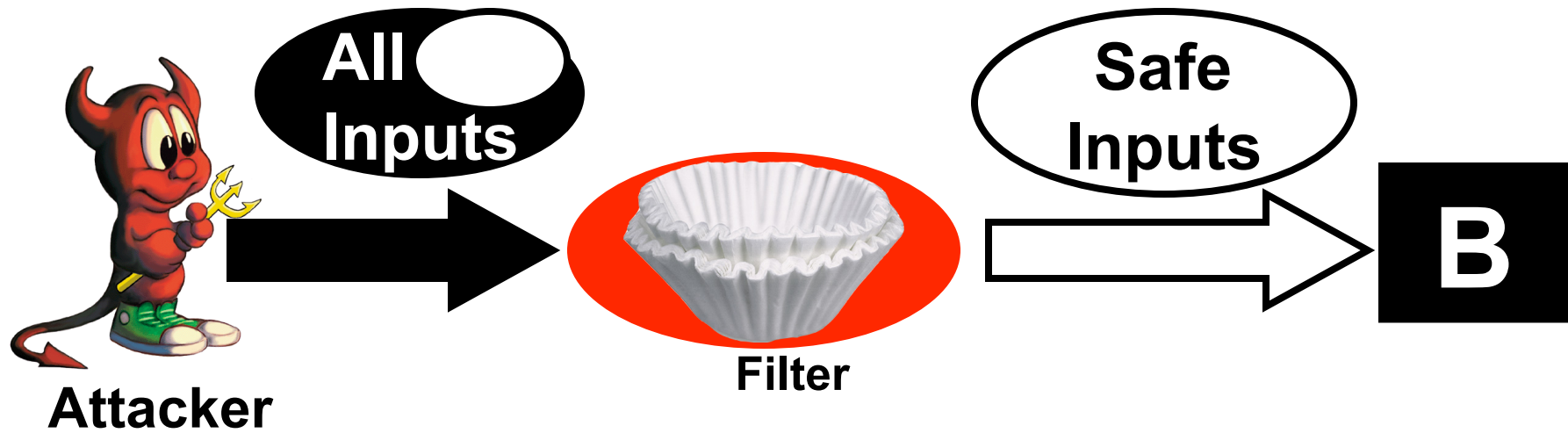
**I could have been a database file, program, password file,
contained top-secret launch codes, etc**

We need Delayed Patch Attack Defenses

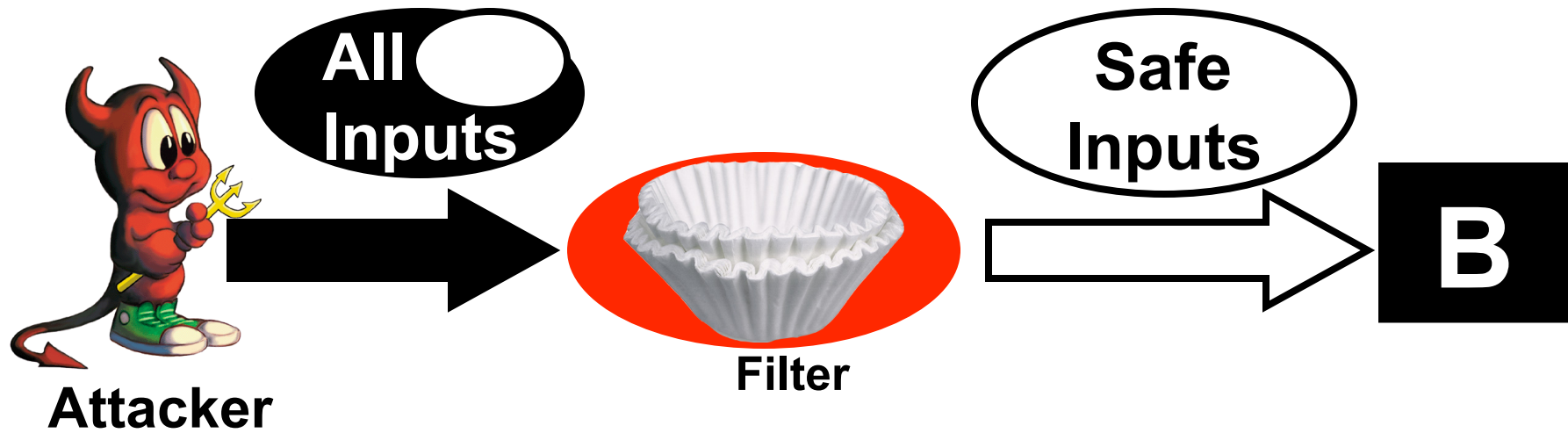
- **Improvements in verification improves results**
- **Code Analysis: Obfuscate patches**
 - Prevents easily finding differences
 - Con: May slow down program, may be insufficient
- **Crypto: Encrypt patch initially, broadcast decryption key**
 - Fair: Everyone applies patch simultaneously
 - Con: How do we test patches before deployment?



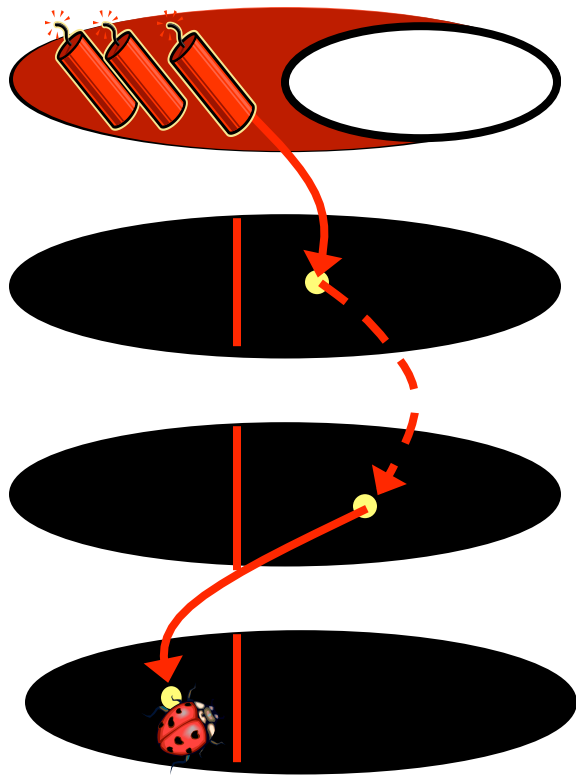
What about other defenses?



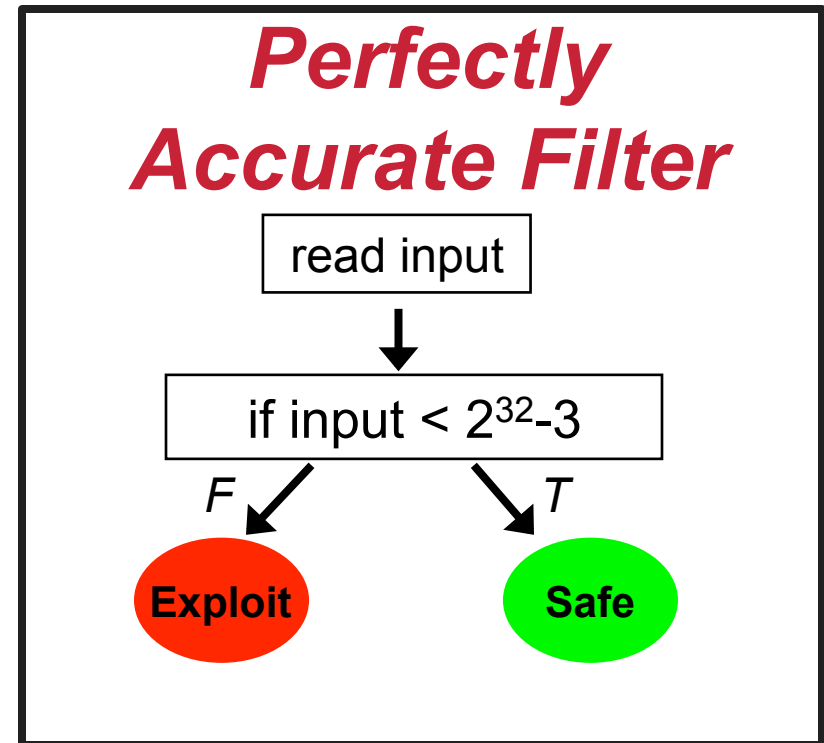
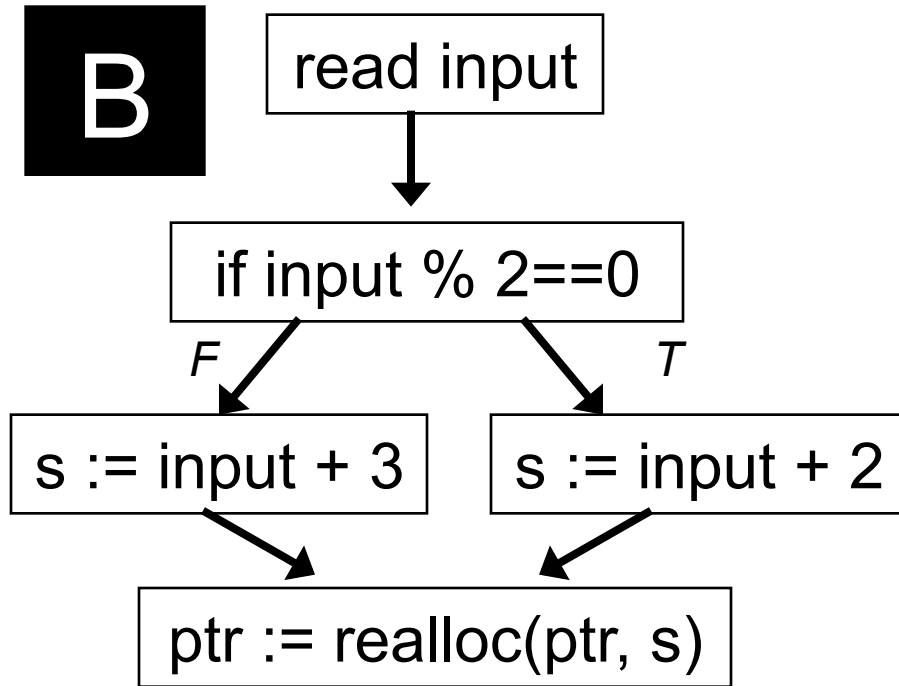
Filter-Based Defenses



Filter:
Recognizer
for all
unsafe
inputs



Given:
1. Exploit
2. B

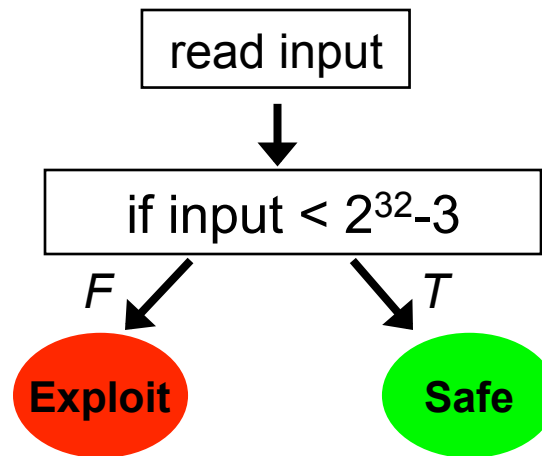


Filter =
Recognizes when $\neg(s < input)$

Accurate Filters Leak Information Too

“Solve”
filter for
Exploits:

$2^{32}-3$
 $2^{32}-2$
 $2^{32}-1$



Summary

Threat model:

Security can help attackers
by leaking weaknesses

Airplanes, tanks, trains,
and other physical systems
are also vulnerable

Thank You

David Brumley
Carnegie Mellon University
dbrumley@cmu.edu