Understanding and Countering Insider Threats In Software Development

Michael Franz
University of California, Irvine

Presented by Ivan Hristov
Department of Computer Science
Dresden University of Technology

Winter Semester 2008

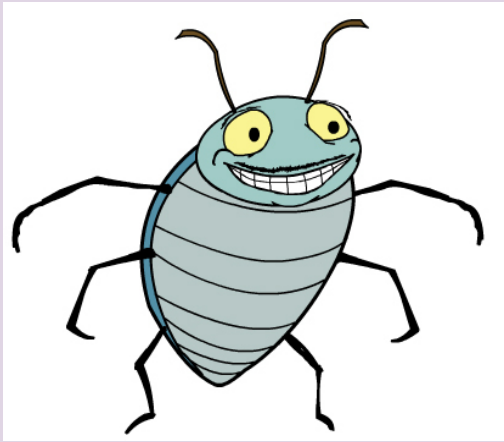
iv.hristov@yahoo.com

Part I

Presentation

"Bug or feature?"

Bugs - bad mistakes or good profit



Bugs can be power!

"Ispa Scientia Potestas Est - Knowledge is power."

Sir Francis Bacon

The Problem

We live in a chaos!

There are bad guys that want bugs!

Aim(s)

What for?

- “zombie farms”
- phishing
- governmental back doors
- other purposes

Conspiracy theory

Trojan horse

- $50 billion dollars industry
- espionage, “moles”
- “protection”

Sources of software bugs

Important aspects to consider

- "doors behind the back doors"
- stocks always matter
- outsourcing
- how well your company treats you
- the good old friend Buddy

Open source utopia

Some problems

- Lack of resources
- “Untraceability”
- Open source

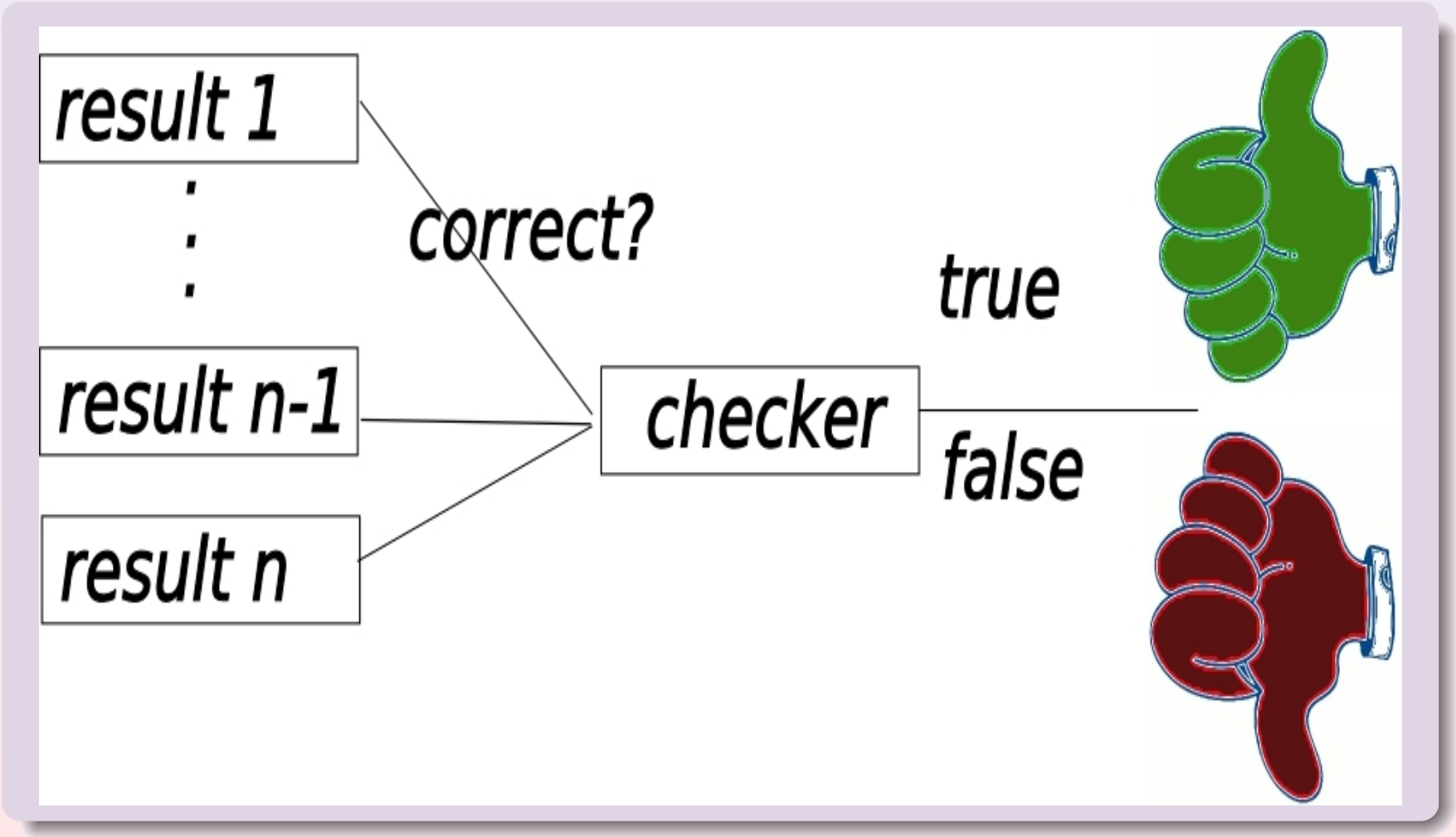
Author's Solution

The idea

Fault tolerance mechanism through

- Versioning
- Parallelism
- Consistency check

Author's Approach



Problem subset

What's treated?

1^{st} arbitrary code execution

2^{nd} specific input

What's NOT treated?

- covert channels
- "time bombs"

Use case

Scenario

- buffer overflows
- specific input
- "out-of-specification" behavior
- knowledge determinism

Existing defense strategies

Basic idea

Ruin the attacker's knowledge determinism

Drawback

Randomization is difficult

Proposed defense strategy

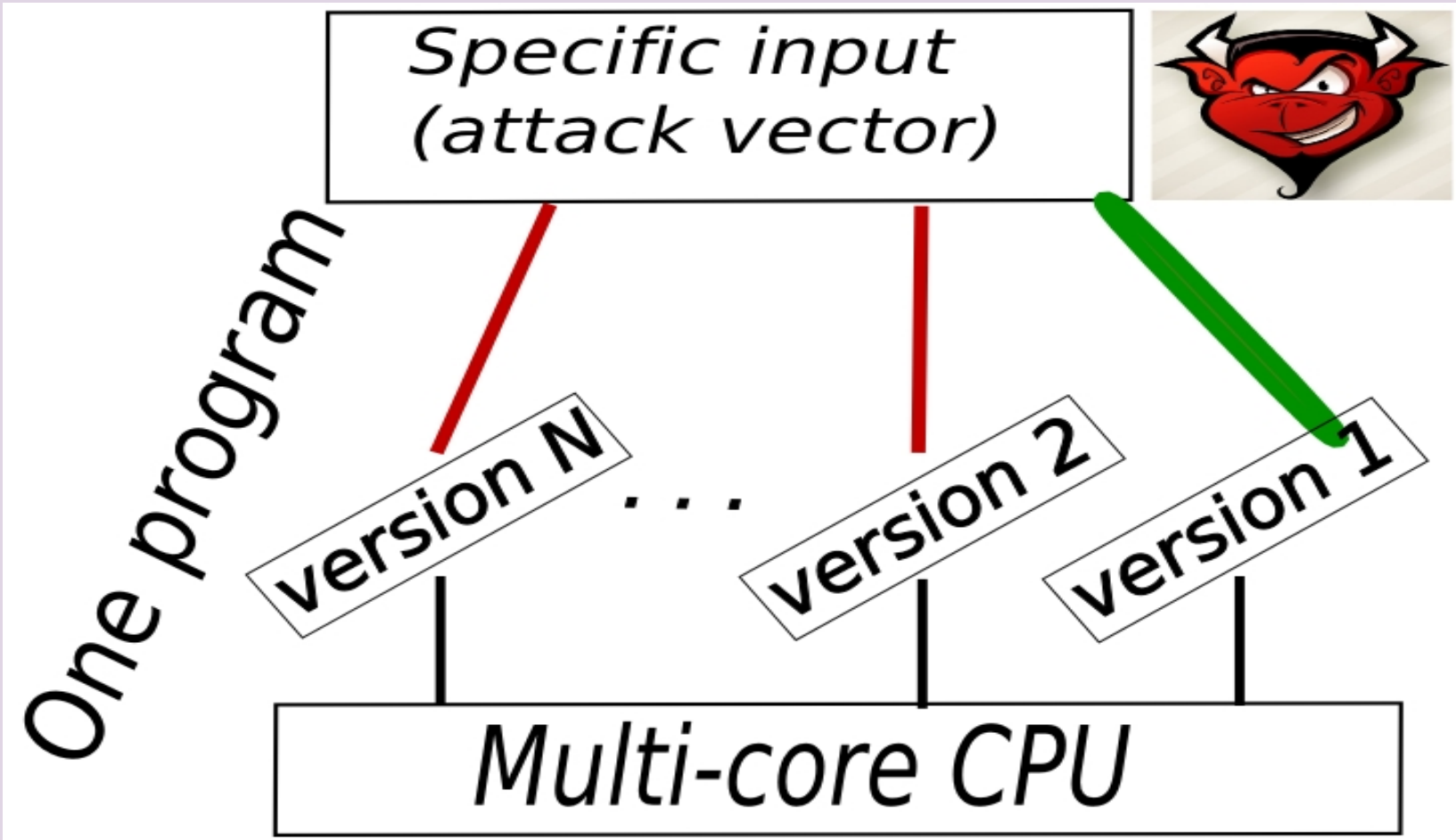
Improvement

- slightly different versions
- parallelism
- monitoring
- optionally - randomization

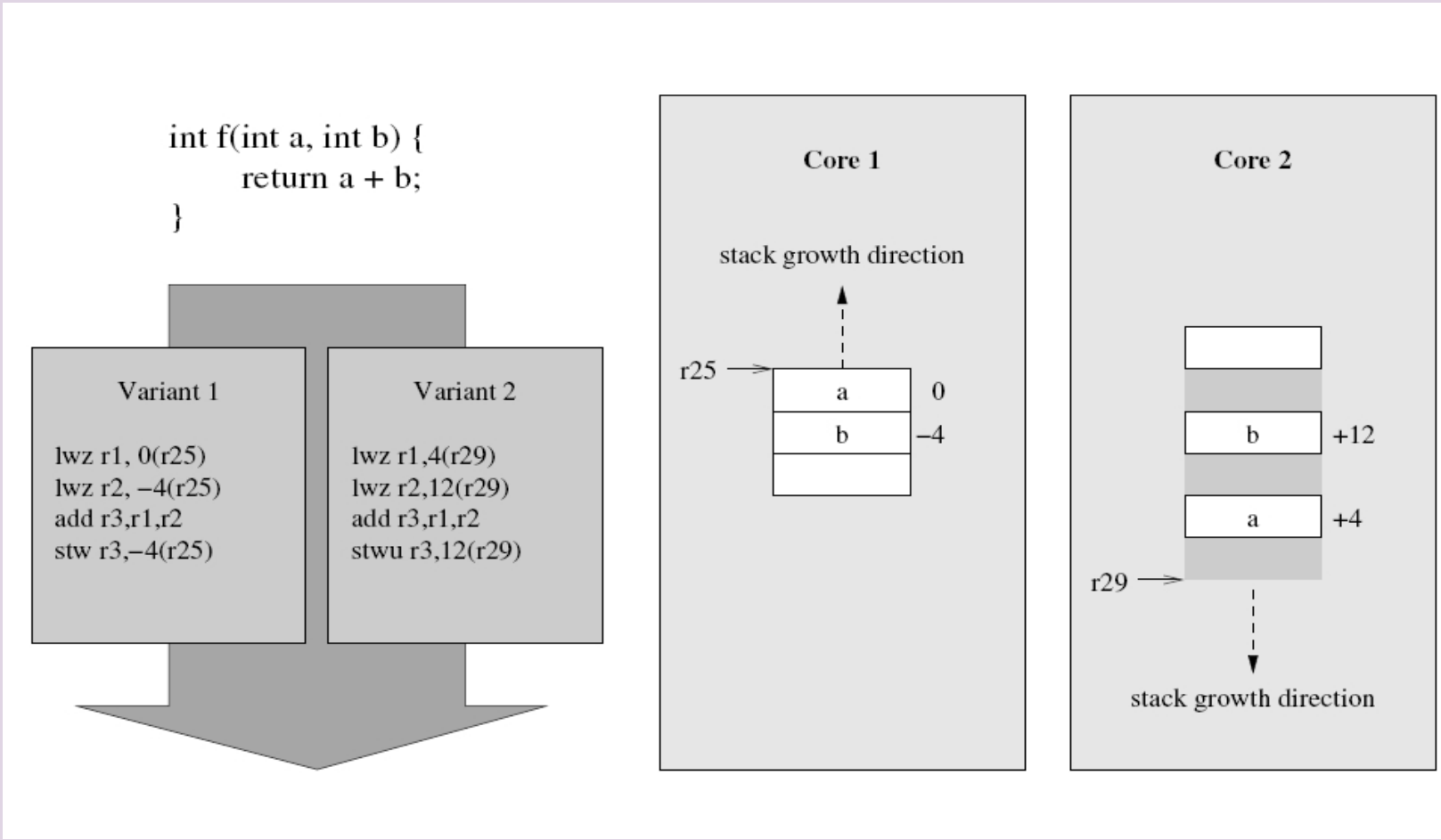
Basic idea

One specific input is meaningful to only *one* program version

Basic Idea


Specific input
(attack vector)
One program
version N
. . .
version 2
version 1
Multi-core CPU

Basic Idea



Two variants of the same program.[Fra08]

Additional variation

Where?

- register reallocation
- heap randomization
- code relocation
- OS Entry Point Randomization

Checkpointing - take the shortcut

Overall process

1st identical inputs

2nd behavior synchronization

3rd internal states monitoring

How far do you trust your OS?

OS calls as synch points

Checkpointing - take the shortcut

Overall process

1^{st} identical inputs

2^{nd} behavior synchronization

3^{rd} internal states monitoring

How far do you trust your OS?

OS calls as synch points

Checkpointing - stay on the safe side

Trusted Computing

- 1^{st} trusted hypervisor
- 2^{nd} hardware component
- 3^{nd} additional registers

Cost?

0.001% of the total CPU transistor amount

Checkpointing - stay on the safe side

Trusted Computing

1^{st} trusted hypervisor

2^{nd} hardware component

3^{nd} additional registers

Cost?

0.001% of the total CPU transistor amount

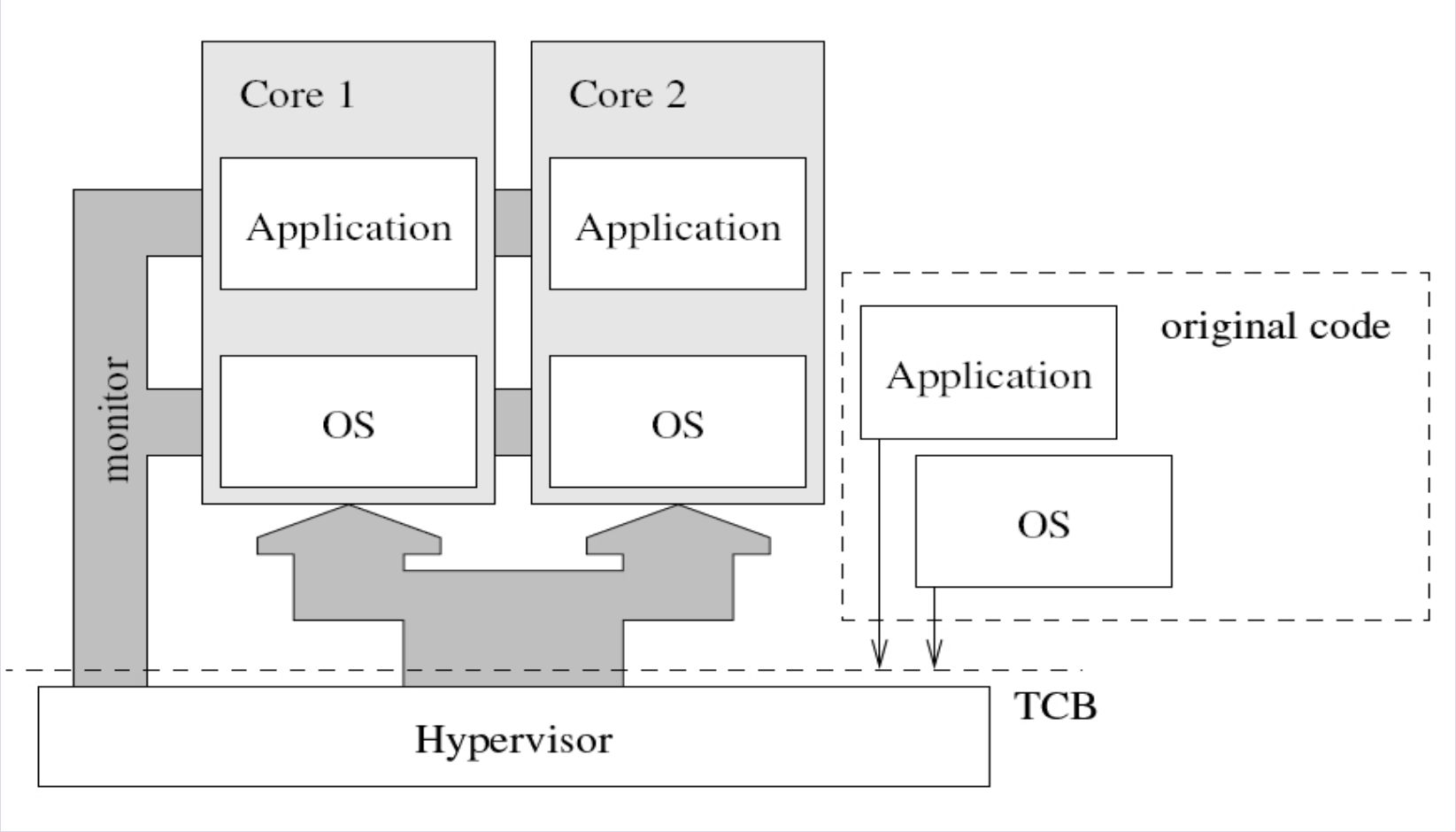
Slightly Different Versions

HOWTO create multiple versions?

- HW virtualization
- storage address remappings
- hypervisor on-demand code translation

Overall architecture

Trusted Code Base



TCB is a hypervisor.[Fra08]

Some discussion points

Does virtualization equate panacea?

What type of cost is the important one?

Checkpoint protocols scheduling?

Part II

References


Michael Franz.
Understanding and countering insider threats in software development.
International MCETECH Conference, pages 81–90, 2008.

Part III

Questions?