

A Platform for Finding Attacks in Unmodified Implementations of Intrusion Tolerant Systems

Hyojeong Lee*, Jeff Seibert+, Endadul Hoque*, Charles Killian* and Cristina Nita-Rotaru*

Department of Computer Science, Purdue University*, MIT Lincoln Labs+

Why Turret?

Intrusion Tolerant Systems

- Ensures correct operation and can make progress even when a fraction of nodes are compromised
- Previous work found attacks that can degrade performance severely so that the system is no longer practically usable
- Finding such attacks is extremely difficult

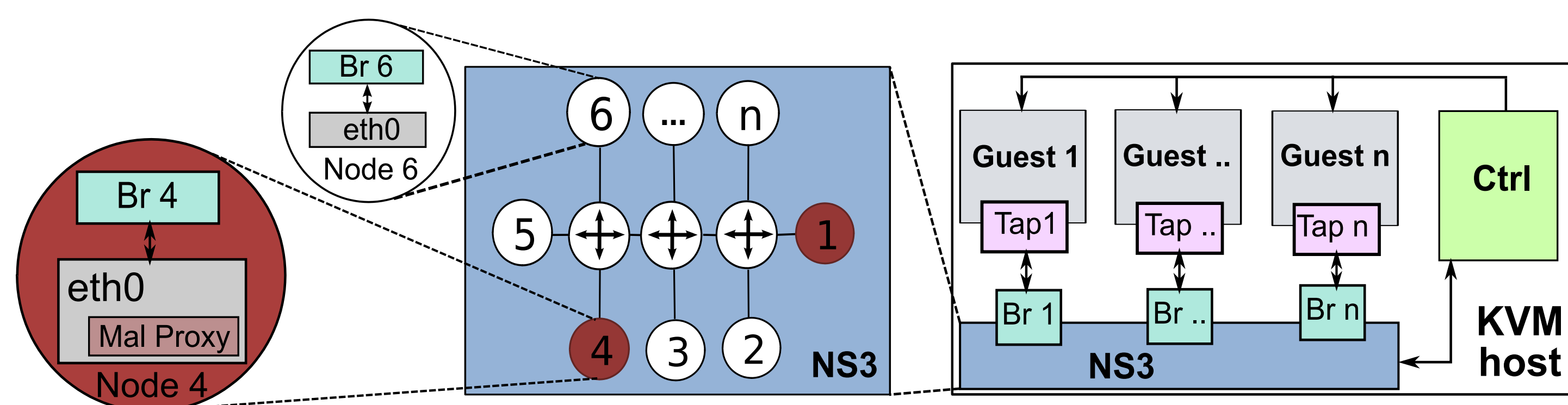
Challenges when Testing Implementations

- Hard to ensure correctness
- No modification
- Interaction with the environment is important
 - : should be tested under the same environment
- Minimal user effort

Turret

- A platform for automatically finding performance attacks in **unmodified** implementations of intrusion tolerant systems
- Minimal user effort and no limitation in implementation
 - Requires binary and an operating system
 - User needs to provide message format

Turret Design and Implementation



Malicious Actions

- Malicious actions: malicious proxies intercept all messages and inject malicious actions: delivery actions and lying action
- Message parser: allow inject sophisticated attacks based on message types and message fields

Entire System Snapshot

- Network emulator snapshot
- Controller can take snapshots of all VMs and the network together

Search Strategy

- Brute force, Greedy

Controller

At (1)
Launch the system and
stay in listening mode

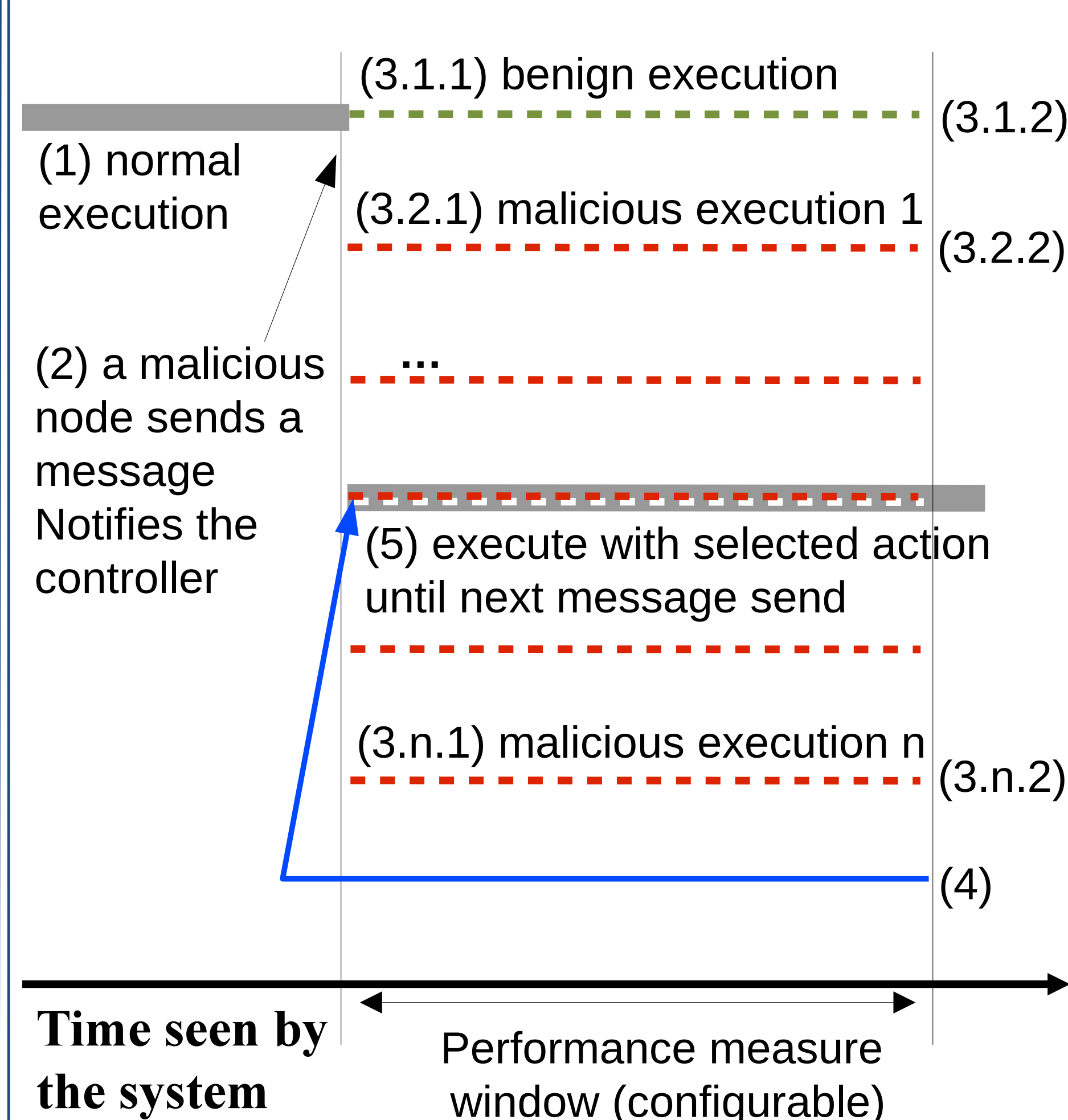
At (2)
Create a snapshot

For each (3.i)
Before (3.i.1)
 Command malicious proxy
At (3.i.2)
 Log the performance and rollback
(Repeat)

After (3.n.2): (4)
Select the worst performance and rollback

At (5)
Choose the selected
action and stay in
listening mode

Target System



Results

- Applied on 5 different intrusion tolerant systems

- PBFT: OSDI 99
- Steward: DSN 06
- Zyzyva: SOSP 07
- Prime: DSN 08
- Aardvark: NSDI 09
- Found 29 attacks (23 new)

