CERIAS

The Center for Education and Research in Information Assurance and Security

A Framework to Find Vulnerabilities Using State Characteristics in Transport Protocol Implementations

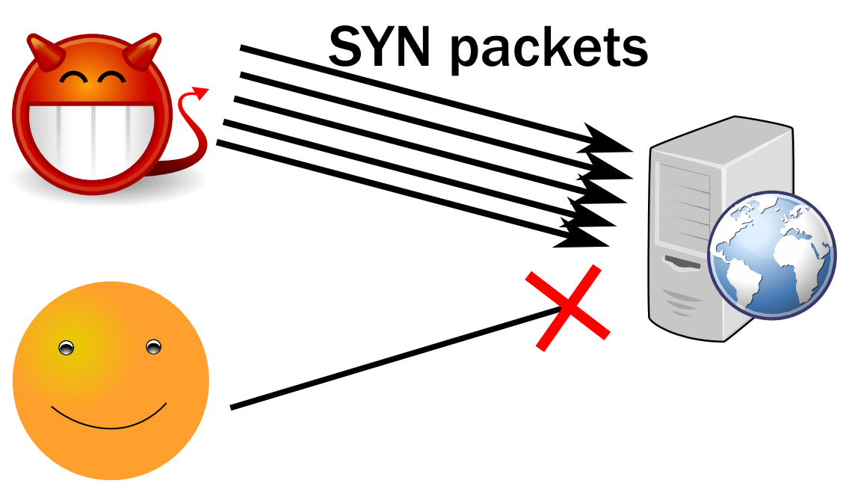
Samuel Jero, Hyojeong Lee, and Cristina Nita-Rotaru

Department of Computer Science, Purdue University

Motivation

- Transport protocols
  - Responsible for end-to-end communication
  - e.g. TCP, provides reliability, ordering, and fairness
  - STCP, QUIC, etc.
  - Many versions and implementations of each protocol
- Testing Models
  - Ignores implementation details
  - Misses implementation bugs
- Testing Implementations
  - Ad-hoc, manual, incomplete testing
- Numerous bugs and vulnerabilities remain

TCP SYN Flood Attack



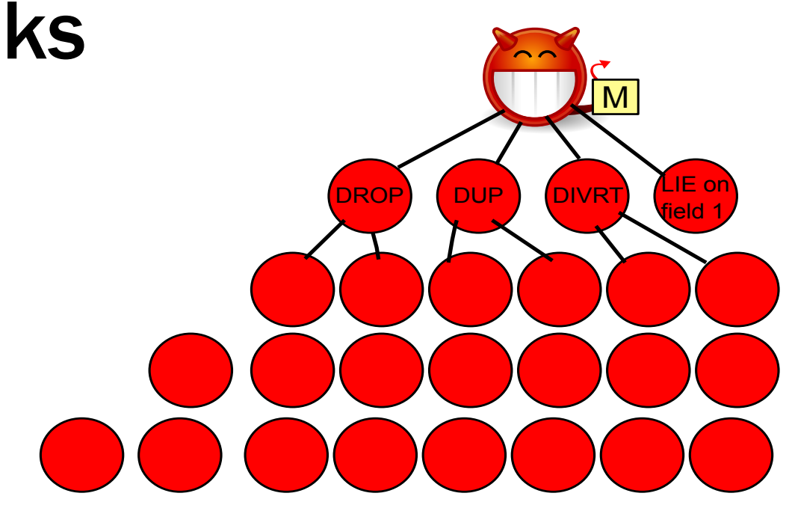
Examples of Attacks found in TCP Implementations:

- Reset attack (Watson 2004)
- SYN Flood (Eddy 2007)
- Ack Storm (Adramov 2011)
- Optimistic Ack (Savage 1999)
- Ack Division (Savage 1999)
- DupAck Spoofing (Savage 1999)
- Shrew (Kuzmanovic 2006)
- Induced-Shrew (Kumar 2009)
- ISN Prediction (Morris 1985)
- Linux Data without Ack flag bug (1999)
- Windows 95 OOB data crash (1997)
- Windows Sockstress attack (CVE-2009-4609)
- Sequence Number Recovery (Gilad 2012)

Need to systematically test protocol implementations in malicious senarios

Design Approach

- Capturing realism: test unmodified implementations
- Malicious / abnormal behaviors
  - Collected from previous studies regarding attacks
  - Conducted by modifying or injecting messages
- Mitigating state-space explosion problem
- A general framework
  - Not limited to a specific target environment / implementation / protocol



Insights

- Automatically inject malicious/abnormal behaviors and observe the result without altering the target code or environment
- Reduce the search space and find effective attacks

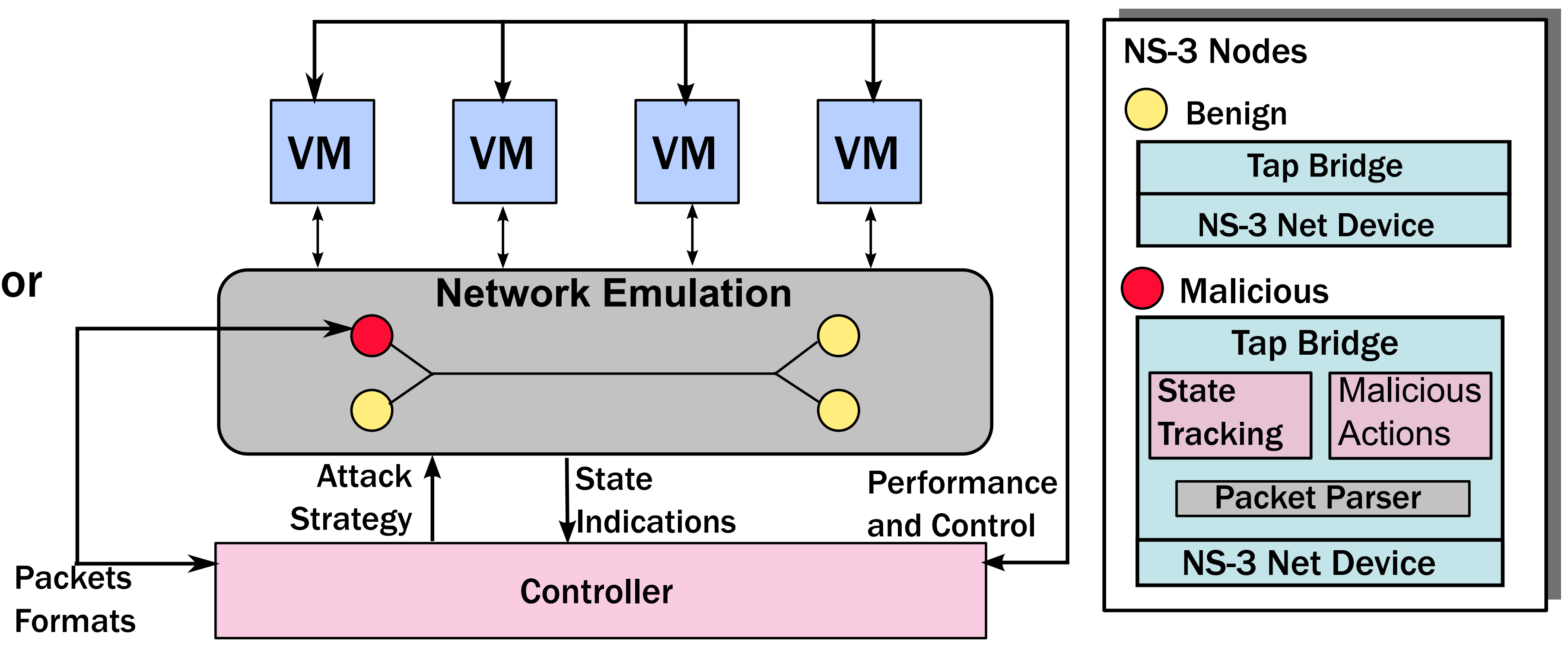
Hypothesis 1: There is a correlation between state characteristics and effective attack strategies

Hypothesis 2: Some characteristics have observable metrics

Use observable metrics to find more effective attack strategies

Turret-T Architecture

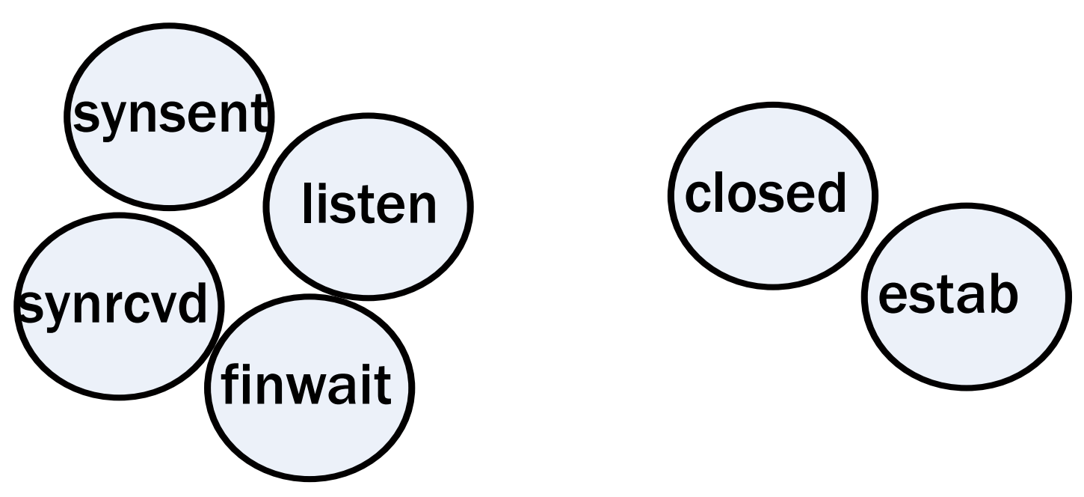
- Based on Turret, a platform to find attacks in distributed systems
- Runs unmodified target system in virtual machines
- Virtual machines connected with network emulator
- Malicious proxy intercepts packets and inject actions in network emulator
- Controller guides search
- Leverage state information



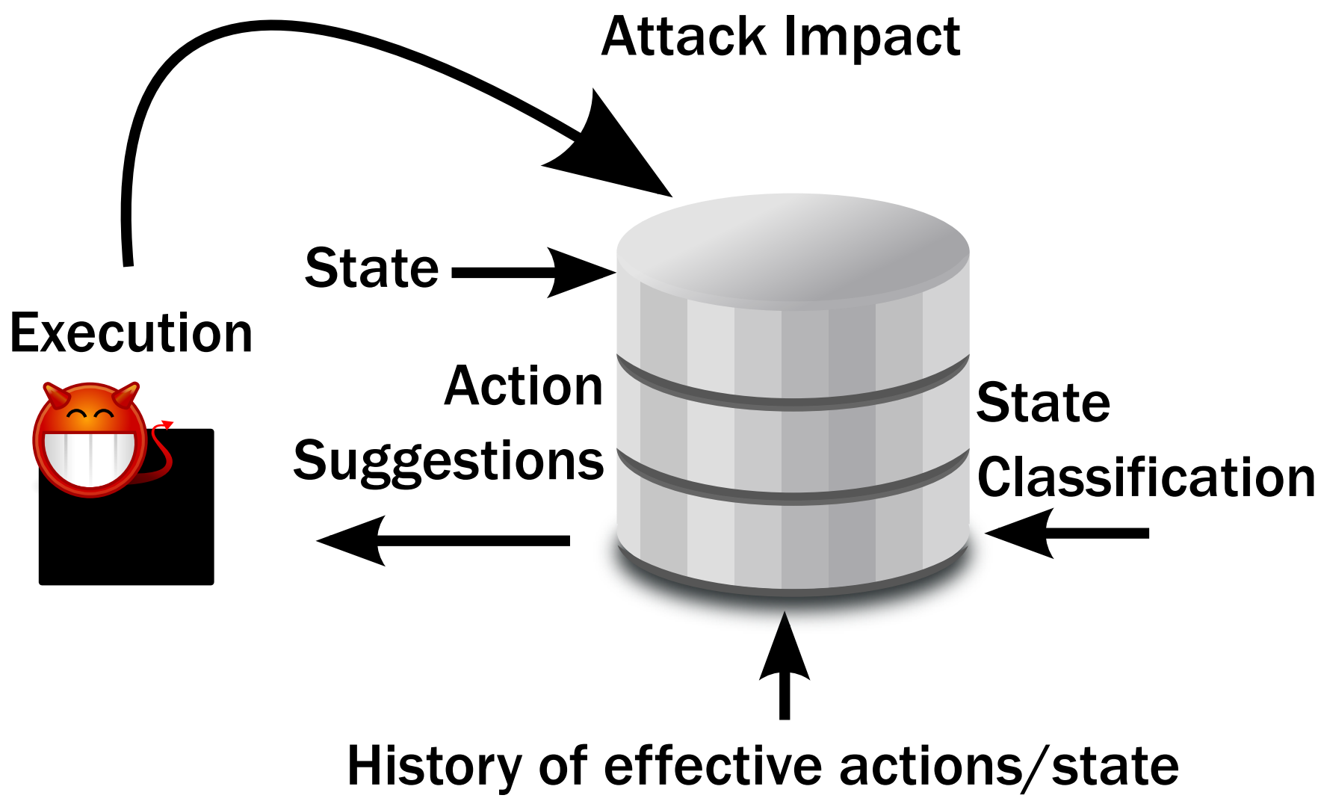
State Information Leverage

Automated State Classification

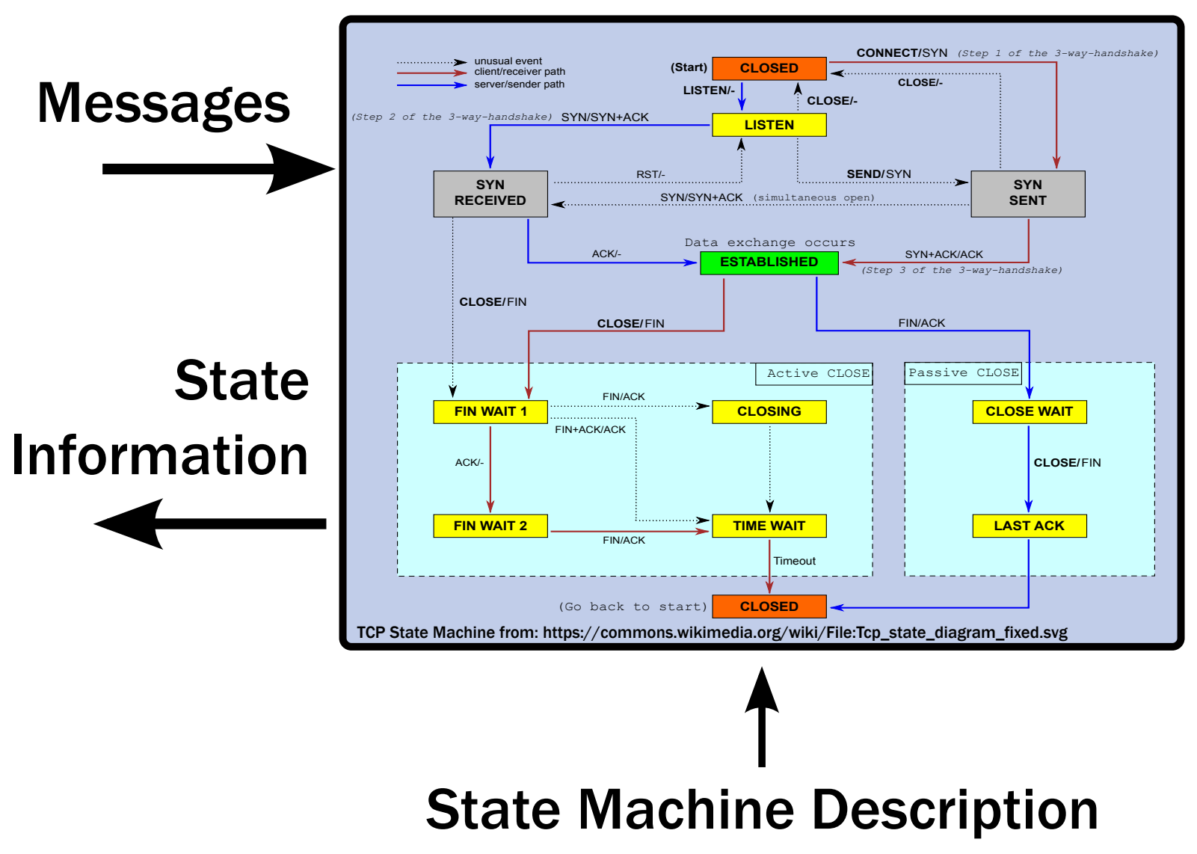
Classify states based on observable characteristics through learning phase e.g. time spent, throughput, etc.



State-based Malicious Action Injection



Protocol State Tracking



TCP State Machine from: https://commons.wikimedia.org/wiki/File:Tcp_state_diagram_fixed.svg

Malicious Actions:

- DROP
- DUPLICATE
- DIVERT
- DELAY
- BURST
- LIE (on field)
- INJECT
- WINDOW

CERIAS

PURDUE UNIVERSITY