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

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

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

State-based Malicious Action Injection
- Execution
- State
- Action Suggestions
- History of effective actions/state

Protocol State Tracking
- Messages
- State Information
- State Machine Description

Examples of Attacks based on TCP packet manipulation
- SYN flood attack (K evil 2019)
- SYN attack (Wiener 2015)
- Ack Storm Attack (2012)
- Ack division (Savage 1999)
- Ack flooding attack (2017)
- New connection (Shih 2013)
- New connection (Marie 1998)
- Window DoS (Morris 1985)
- Excessive Out of Band (Lee 2018)
- Buffer overflow (Shih 2006)
- Sequence Number Recovery (2018)

SYN packets

TCP SYN Flood Attack

Need to systematically test protocol implementations in malicious senarios