Shadow: Running Tor in a Box for Accurate and Efficient Experimentation

Rob Jansen and Nick Hopper
University of Minnesota
U.S. Naval Research Laboratory
rob.g.jansen@nrl.navy.mil
Time in seconds to complete 50 KiB request

Measured times on all sources per day

- Median
- 1st to 3rd quartile

The Tor Project - https://metrics.torproject.org/
Time in seconds to complete 50 KiB request

Measured times on all sources per day

- Median
- 1st to 3rd quartile

The Tor Project - https://metrics.torproject.org/
Time in seconds to complete 50 KiB request

Measured times on all sources per day

- Median
- 1st to 3rd quartile

The Tor Project - https://metrics.torproject.org/
Tor in a Box with Shadow

- Discrete event network simulator
  - Natively executes real applications
  - Simulates time, network, crypto, CPU
  - Model latency and bandwidth
- Efficient, accurate, controlled
- Single Linux-box without root
Shadow's Design I

- Simulation blueprint
- Discrete time events
Shadow's Design II

- Node management
- Function interposition

Shadow Memory Space

Tor Memory Space

Context Switch
Scallion – A Plug-in Running Tor

- Integrates Tor into Shadow

- Scalability
  - 1250 nodes in 10 GB RAM, 5x* - 10x** slowdown
  - 5750 nodes in 60 GB RAM, 40x** slowdown

* 3.3 GHz AMD Phenom II X6 1100T  ** 2.2 GHz AMD Opteron 6174
Accuracy Shadowing Tor

- Cumulative Fraction vs. 50 KiB Download Time (s)
- Cumulative Fraction vs. 1 MiB Download Time (s)
- Cumulative Fraction vs. 5 MiB Download Time (s)
Demonstrating Shadow's Utility

Tang & Goldberg [CCS 10]
Lightly Loaded Tor

Heavily Loaded Tor
Conclusion

➔ Efficient, accurate, controllable, repeatable

➔ Tor experiments on one machine
  • Larger scale than previously possible
  • New results from new capabilities

➔ Able to run many applications

➔ Freely available and usable software
Questions?

rob.g.jansen@nrl.navy.mil
cs.umn.edu/~jansen

shadow.cs.umn.edu
github.com/shadow
How Tor Works

Client

Relays

Server
Testing Tor Improvements

- Most popular anonymous communication system
  - 500K – 1M users

- New algorithms/protocols need testing

- No standard experimentation approach
## Recent Tor Experimentation*

<table>
<thead>
<tr>
<th>Category</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Live Tor and PlanetLab</strong></td>
<td>Bauer et al. [WPES 07], Hopper et al. [CCS 07], Tang and Goldberg [WPES 07], McCoy et al. [PETS 08], Snader and Borisov [NDSS 08], McLachlan and Hopper [WPES 09], McLachlan et al. [CCS 09], Chaabane et al. [NSS 10], Mulazzani et al. [CMS 10], Tang and Goldberg [CCS 10], Luo et al. [ACSAC 11]</td>
</tr>
<tr>
<td><strong>Emulation</strong></td>
<td>Chakravarty et al. [ESORICS 10], AlSabah et al. [PETS 11], Moore et al. [ACSAC 11]</td>
</tr>
<tr>
<td><strong>Simulation and Modeling</strong></td>
<td>Borisov et al. [CCS 07], O'Gorman and Blott [ASIAD 2007], Murdoch and Watson [PETS 08], Ngan et al. [FC 10], Jansen et al. [CCS 10]</td>
</tr>
</tbody>
</table>

* Not a comprehensive list
## Network Experimentation

<table>
<thead>
<tr>
<th>Approach</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>Not generalizable, inaccurate</td>
</tr>
<tr>
<td>Emulation</td>
<td>Large overhead, kernel complexities</td>
</tr>
<tr>
<td>PlanetLab</td>
<td>Hard to manage, bad at modeling</td>
</tr>
</tbody>
</table>
Tor in a Box with Shadow

→ Discrete event network simulator

→ Runs real application without modification

→ Accurate, efficient, scalable

→ Runs on Linux without root privileges
Shadow Architecture
Function Interposition

- Intercept, redirect function calls
- $ readelf -s shadow
  - 0 FUNC GLOBAL UND socket@GLIBC_2.2.5
  - 210 FUNC GLOBAL 13 vssocket_socket
- $ ldd shadow
  - libm.so.6 => /lib64/libm.so.6
  - libdl.so.2 => /lib64/libdl.so.2
  - libc.so.6 => /lib64/libc.so.6
Function Interposition

→ LD_PRELOAD=/home/rob/libpreload.so

→ Search my library first
Tor Circuit Scheduling

Circuit Input

Relay

Output
Tor Circuit Scheduling

Circuit Input

Relay

Output
Tor Circuit Scheduling

Circuit Input

Output

Relay
Tor Circuit Scheduling

Circuit Input

Relay

Output
Tor Circuit Scheduling

Round Robin

Circuit Input

Output

Relay
Tor Circuit Scheduling

EWMA [Tang and Goldberg CCS 2010]
Tor Circuit Scheduling

EWMA [Tang and Goldberg CCS 2010]
EWMA: Bottleneck

1 MiB/s

10 MiB/s

10 MiB/s

10 MiB/s

1 MiB/s
EWMA: Bottleneck

Cumulative Fraction vs. Web Download Time (s)

Cumulative Fraction vs. Bulk Download Time (s)
Summary

➔ Simulate time, network stack, crypto ciphers

➔ Model network latency and node bandwidth from real measurements

➔ Natively executes real application code