Measurements:
Mapping Manipulation in the Internet

Brian Trammell, ETH Zürich, WP1 Lead

3 March 2016

mami

measurement and architecture for a middleboxed internet

measurement | architecture | experimentation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688421.The opinions expressed and arguments employed reflect only the authors' view. The European Commission is not responsible for any use that may be made of that information.

Supported by the Swiss State Secretariat for Education, Research and Innovation under contract number 15.0268. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government.

What We're Doing

- Methodology and Measurement Tool Design [1.1, from now]
  - Tracebox, Copycat2, Pathspider, Atlas for UDP, CGN detection
- Initial Measurements and Middlebox Detection [1.2, from April]
  - Atlas, Copycat/Tracebox on Ark etc. for IMC underway
- Data Model and Middlebox Observatory [1.4, ~~from June~~ in progress]
  - details on following slides

What Gets Done When (Y1)?

we are here

Data Management Plan *30.4.* **D4.1**

2nd Plenary Berlin

1st Review Barcelona?

Initial Report *31.12.* **D1.1**

jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec

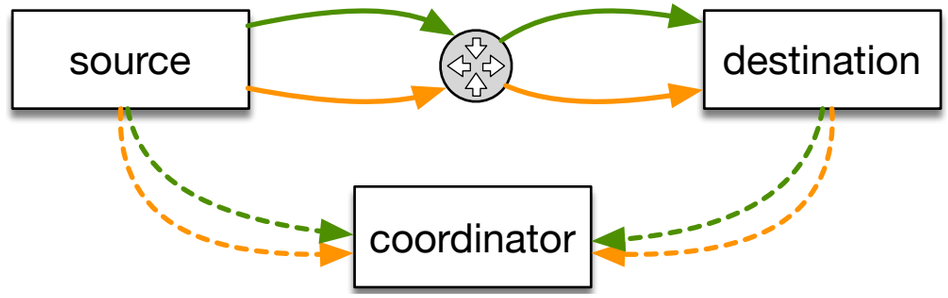
5.5./12.5. IMC deadline (UDP/TCP campaign)

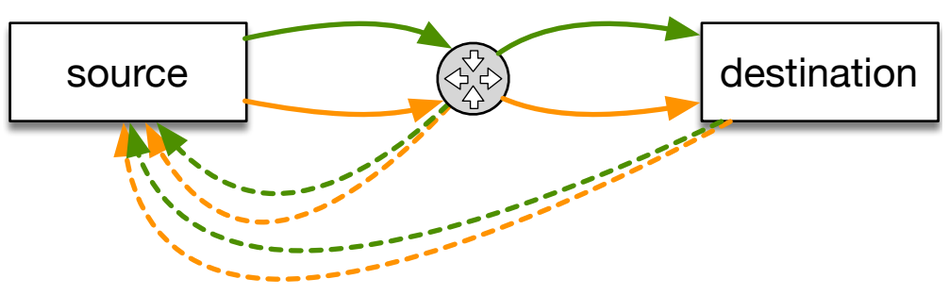
MS2 *31.10.* Internal observatory release

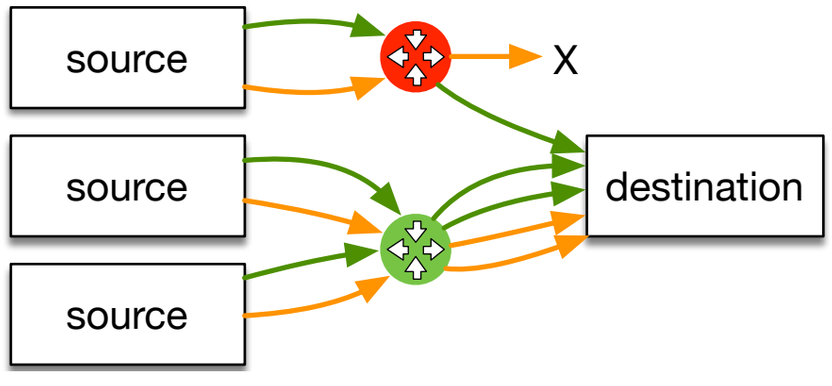
MS4 *31.12.* Observatory release

Active Measurement of Path Transparency

- Basic methodology: throw a bunch of packets with certain properties at the Internet, and see what happens.
  - Ideal: two-ended A/B testing
  - More scalable: one-ended A/B testing
  - Comparison with topology to isolate on-path vs near-endpoint impairments
- Observations from platform- and application-level logs of failed attempts to use protocol features also useful.
- Integrate heterogeneous observations from many campaigns for better insight.
  - Build an observatory for this integration







Observatory Requirements

- Accept data from a wide variety of sources, e.g.:
  - Raw output from tools we maintain.
  - Raw packet traces of active measurements.
  - “Here’s data from a measurement study, and references to commits in GitHub for the tools and configuration used to generate it.”
  - “Option foo breaks on these paths but not these paths; we’re not going to tell you where they are, but this set of them belong to a major mobile carrier.”
- Support path pseudonymization and aggregation for privacy.
- Support condition definition with enough precision to allow active measurements to reproduce observations on other paths.
- Integrate with existing tools, without restrictions on implementation.

Observation Data Model Proposal

- An observatory is a collection of single ***observations*** {***t, P, c***} where
  - ***t:*** time at which the observation was taken (and assumed valid)
  - ***P:*** designator of the path on which the observation was taken
    - sequence of node/network/multi-network identifiers or pseudonyms
  - ***c:*** variable-definition expression of the condition observed on that path
    - including **v**: vector of condition-specific variables
    - Initial plan: reference to external condition identifier linked to how it was generated (stable code, configuration, raw-data reference)
- "Raw" data within the observatory used to back these observations.
- Queries always operate on this intermediate observation representation.