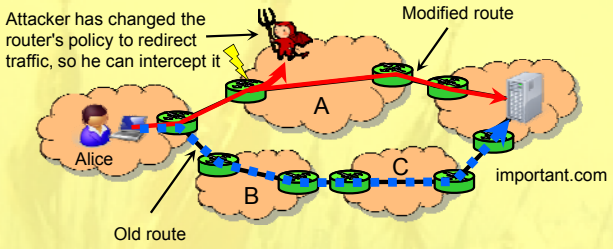
Secure Network Provenance

Andreas Haeberlen* Boon Thau Loo* Micah Sherr# Zachary G. Ives*
Wenchao Zhou# Mingchen Zhao* Arjun Narayan* Alexander Gurney* W. Brad Moore# Qiong Fei*
*University of Pennsylvania #Georgetown University

1 Problem: Secure forensics

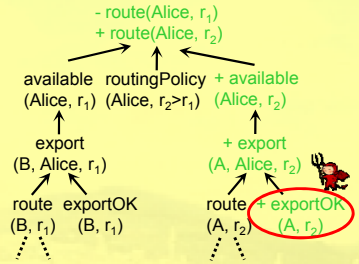


Scenario: Attacker has secretly **compromised** some unknown part of a distributed system

- Affected nodes may now run different software
- Data may be corrupted or destroyed
- Nodes can "tell lies" to confuse the administrators

Goal: Enable the administrators to **detect** and **correctly diagnose** the problem

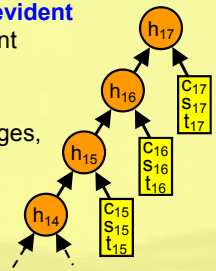
2 Approach: Data provenance



Idea: System should be able to "explain" its own state to the administrator

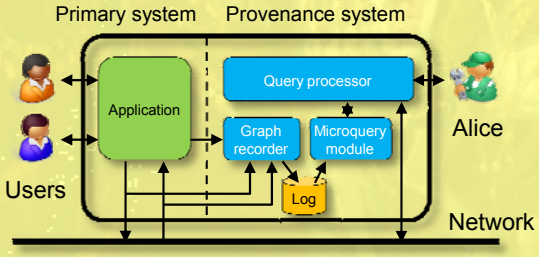
- Explanation contains the **provenance** of the state (based on concept from databases)
- Provenance should be **tamper-evident**: If the adversary tells lies, we can reliably detect this
- Effect: Misbehaving nodes must give the correct explanation (→discovery) or tell a lie (→discovery)

3 Key ideas



- Each node maintains a **tamper-evident log** of all the messages it has sent and received.
- If a compromised node modifies, forges, omits, or reorders messages, this can be detected
- Forensic investigator can **audit** a node's log and **replay** it to reconstruct its execution
- To **extract provenance**, the system can be instrumented; in some cases (declarative languages, 'maybe' rules), extraction can be automated
- Detection can be guaranteed for **observable messages** - that is, messages that directly or indirectly affect at least one correct node
- The investigator must trust his local machine, but otherwise **no trusted components are needed**

4 The SNooPy system

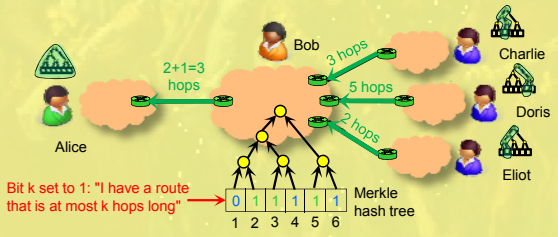


First practical implementation of SNP

- **Widely applicable** - evaluated with BGP inter-domain routing, a DHT, and Hadoop MapReduce
- Detection guarantees **formally proven**
- Reasonable overhead
- Code available from http://snp.cis.upenn.edu/

[TaPP'11, SIGMOD'11 demo, SOSP'11]

5 Protecting privacy with PVR

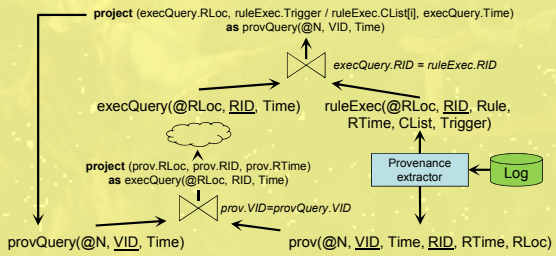


Problem: Provenance can reveal private data

- Solution: Special "distributed" **zero-knowledge proof** that the provenance is valid
- **Highly efficient**; single machine is enough to handle an entire ISP's proof+verification load
- Provable detection and privacy guarantees

[HotNets'11, SIGCOMM'12]

6 Storing provenance with DTaP



Problem: Store & query provenance efficiently

- Builds a model of the system's workload and automatically chooses **most efficient data structure** to store the provenance
- Can **partially reconstruct** the provenance graph (only the parts that are needed to answer the query)

[TaPP'12, VLDB'13]