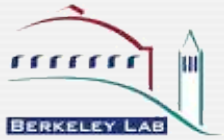
Distributed Cooperative Security Monitoring

Robin Sommer
Lawrence Berkeley National Laboratory

rsommer@lbl.gov
http://www.icir.org/robin



Cooperative Security Monitoring

- Internet sites monitor their network for breaches
  - E.g., with host-based and network-based IDSs
- Monitoring considers only local resources
  - Host activity, network traffic, firewall logs, etc.
- Many attacks however are of global nature
  - DDOS floods, bots, scanners, phishing, etc.
  - Cooperation between sites promises a large gain in detection quality
- Goal of the proposal:

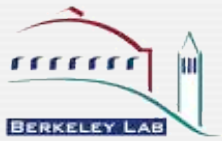
  Building a platform for cooperative monitoring

Real-World Constraints ...

- In practice, data sharing is *really* hard ...
  - Privacy and trust concerns severely limits what can be shared
  - Security policies differ across sites
- Sites must retain control over their operation
- Existing efforts fall short of being deployed
  - Operational systems are often of limited use, e.g.,
    - DShield (data quality unclear)
    - Internet Motion Sensor (restricted to darknet space)
  - Research systems often ignore operational deployment issues
- Primary objective for us:

  A system which addresses operational constraints

Viable Inter-Site Security Analysis

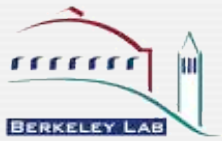


Restricting the Scope ...

- Primary setting: *Data sharing for incident analysis*
- Addressing operational constraints by limiting scope
  - Administrators would like to share data
  - But are reluctant to share *(i)* with everybody, and *(ii)* in an automated fashion
- Restricting the scope promises more utility
  - Exchange information across a small set of "similar" sites (e.g., DOE labs, UCs)
  - Keep the human in the loop for all policy questions
- Gives us a beneficial trust model
    - Assume that sites *usually* behave in a responsible manner
    - Yet failure of trust does not compromise security (but may require time)

When Security is Breached Today

- Lots of questions when host has been compromised
  - How did the attackers get in?
  - Where did they come from?
  - How can we detect/prevent such an attack in the future?
  - Was it just us?
- Analysts draw on many resources to answer & react
  - Local (Searching logs for evidence; extending monitoring/enforcement)
  - Remote (Contacting other sites for information; distributing results)
- Manual nature of this process makes it expensive
  - Time and resources are limited; only high-priority incidents can be examined
- Idea: Automate the steps requiring lots of effort but keep the human in the loop for important decisions



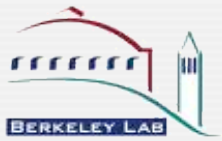
Informal Activity Description

"Attackers compromised SSH credentials, perhaps grabbed at one of **these sites**. They then fetched their botware from two different places, with filenames ending in **.jpg** even though the files are executables. They opened up backdoors on the **following ports**. These other sites connected to **those backdoors**. One signature of the activity is outbound email from a compromised host with **this subject**, or—maybe—Web activity fetching **these URLs**, but unfortunately we're not sure. If you find evidence of infection and can tell us about other URLs or attacking sites, we would much appreciate that so we can check them against our logs."

(Semi-)Automating the Process

- Once an incident is understood, a site codifies the relevant activity into a high-level *script*
  - Describes how to locate activity in *past* logs and in *future* activity
  - Describes what kind of information site would like to know from its partners
- Script is send to partners
- Receiving analyst manually decides whether
  - Is of interest
  - Confirms to local policy (e.g., privacy constraints)
  - Sufficient resources are available to run the script
- If so, the receiver executes the script to
  - Find past/future attacks on the local network
  - Answer the queries of sender

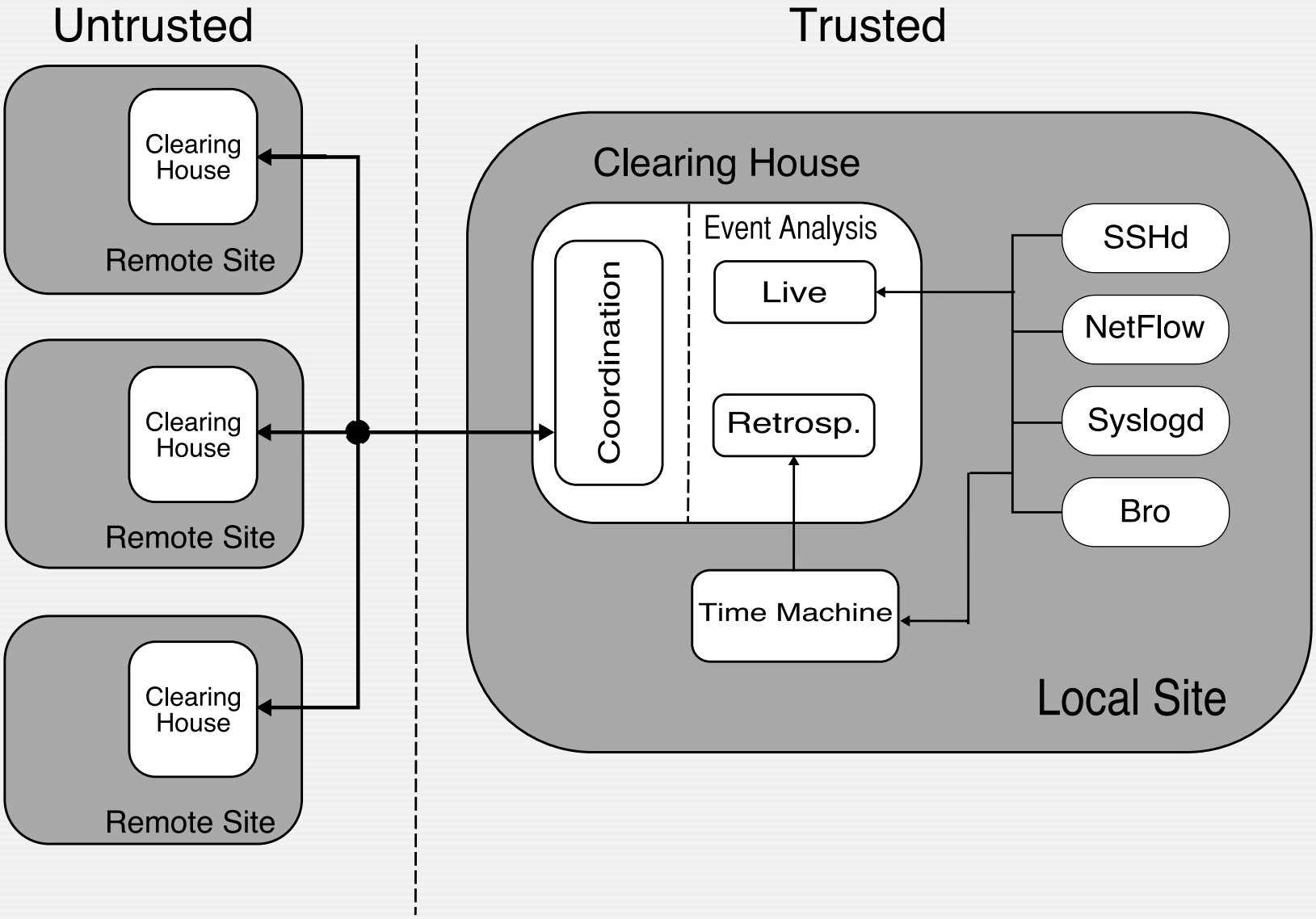
Building A Framework For Semi-Automated Information Sharing



Building a Framework

- Goal: Building a framework supporting our model
  - Automating the routine steps
  - Relying on the human operator for key decisions
- Three main ingredients
  - Unified data model to facilitate automatic data collection & retrieval
  - Data archiver supporting queries for past activity—a "*Time Machine*"
  - Communication platform to facilitate easy & secure data exchange

Clearing House Architecture



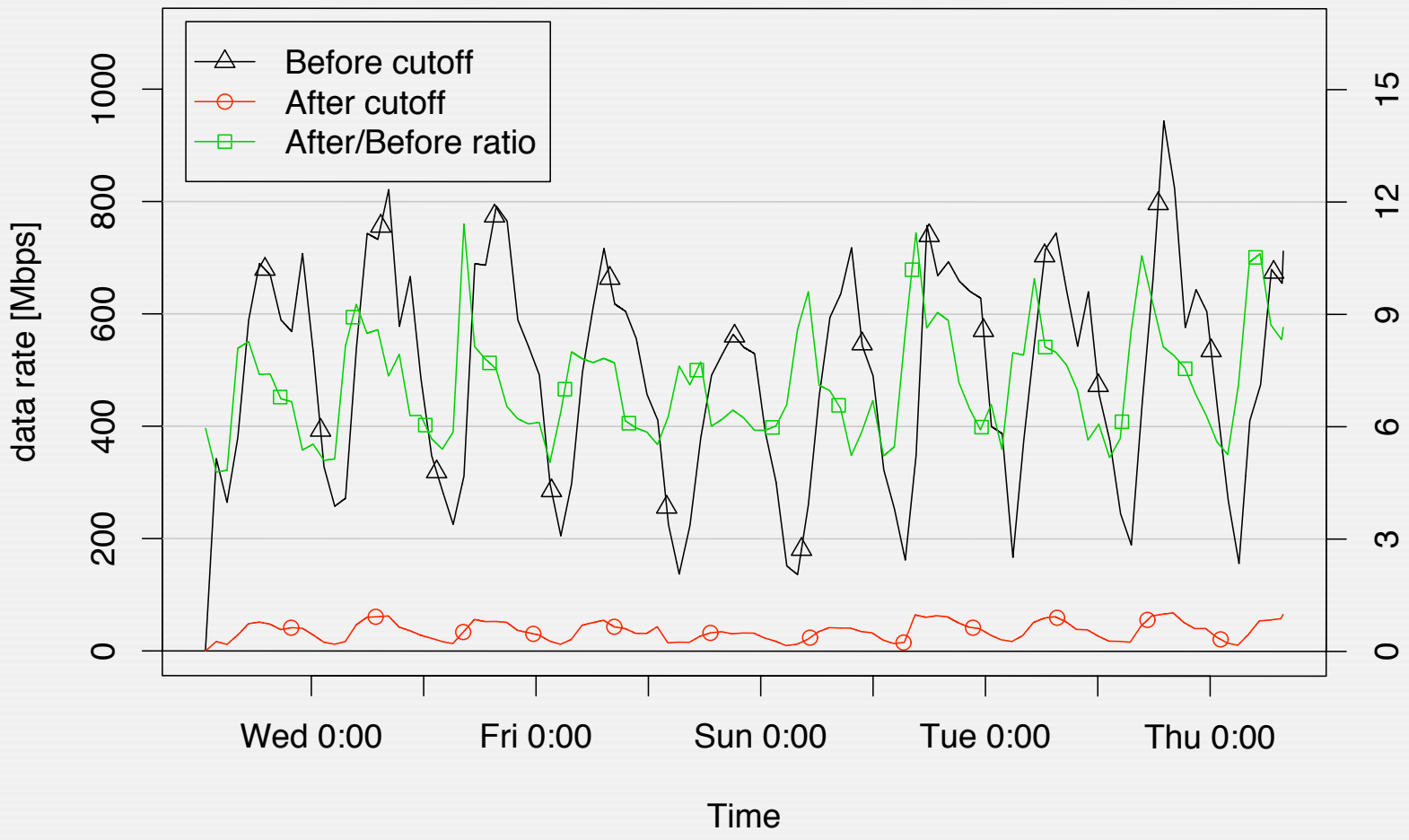
Data Model of Network Activity

- Event-based data collection & dissemination
- Abstract network activity into high-level *events*
  - E.g., `connection_attempt`, `http_request`, `ssh_login`
  - Events are policy-neutral
  - The "Bro approach" to security monitoring
- Instrumenting network components to provide events
  - Already done for Bro-style events via Broccoli (e.g, Apache, syslog)
  - In the process of extending this further (e.g., sshd)

The Time Machine

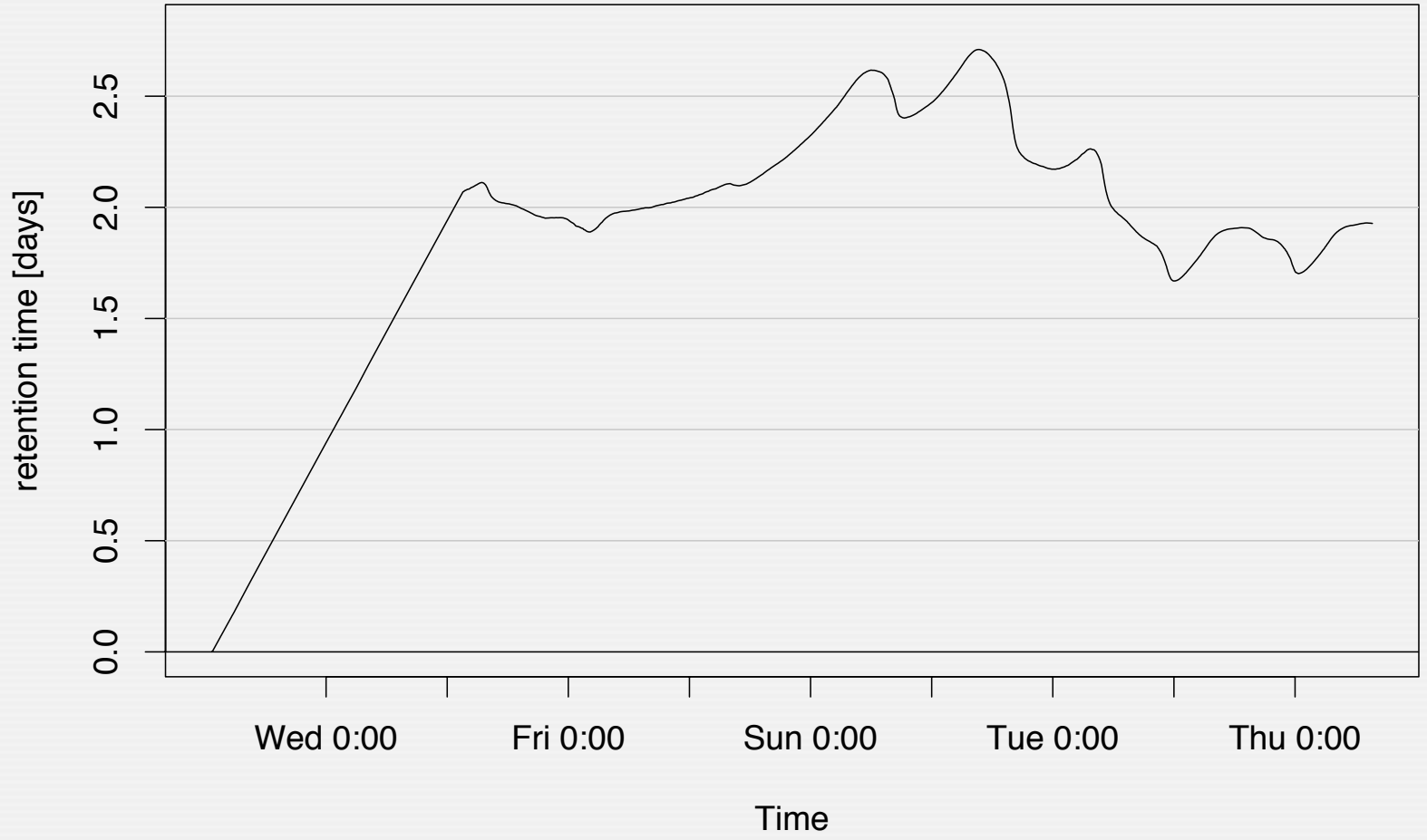
- High-performance activity recorder
  - Records large volumes of events to disk
  - Provides query interface to retrieve stored data
  - Expires old data
- We have built a Time Machine for *network packets*
  - Efficient packet bulk-recorder for high-volume network streams
  - Heuristic for volume reduction: store only the first few KB per connections
  - Oldest packets are discarded once available buffer space is filled

The Time Machine



10 Gbps campus link, 3-6 TB/day, 50,000 hosts
15KB cut-off, 800G buffer

The Time Machine



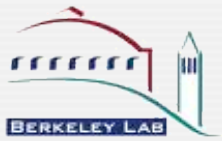
10 Gbps campus link, 3-6 TB/day, 50,000 hosts
15KB cut-off, 800G buffer

The Time Machine

- High-performance activity recorder
  - Records large volumes of events to disk
  - Provides query interface to retrieve stored data
  - Expires old data
- We have built a Time Machine for *network packets*
  - Efficient packet bulk-recorder for high-volume network streams
  - Heuristic for volume reduction: store only the first few KB per connections
  - Oldest packets are discarded once available buffer space is filled

The Time Machine

- High-performance activity recorder
  - Records large volumes of events to disk
  - Provides query interface to retrieve stored data
  - Expires old data
- We have built a Time Machine for *network packets*
  - Efficient packet bulk-recorder for high-volume network streams
  - Heuristic for volume reduction: store only the first few KB per connections
  - Oldest packets are discarded once available buffer space is filled
- Interfaced Time Machine with a NIDS
  - NIDS can control the time-machine
  - NIDS can query the time-machine for past traffic
    - Enables retrospective analysis when new insight is available

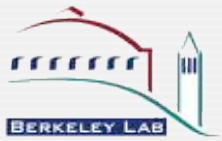


Generalized Time-Machine

- Generalizing the Time Machine
  - Input stream is events (rather than packets)
  - Volume reduction via event-specific filtering (rather than cut-off)
  - Aggregation as expiration strategy (rather than discard-oldest-data)
- Provides a comprehensive log of network activity
  - Stores low-level, heterogeneous, policy-neutral activity
  - Enables queries for past network activity
  - Allows policy-specific filtering
- Pretty useful as a stand-alone forensic tool
  - Examining compromises from external attackers
  - Understanding insider activity

A Generic Communication Platform

- Build a communication platform which
  - Supports our specific application-scenario
  - But is also sufficiently generic to later be extended to other approaches
- Basic setup
  - Set of sites which interconnect their security monitoring infrastructure
  - Communication primitives to exchange data
- Requirements for (generic) communication
  - Does not enforce a particular communication model (e.g., events, unicast/multicast)
  - Provides sites with direct, fine-grained control
  - Not restricted to a particular local monitoring setup
  - Does not impact the existing local monitoring
  - Is robust against failures and attacks



Working on the Communication...

- Designing a communication protocol
  - With support for publish/subscribe and point-to-point communication
  - With security in mind
- Building a C++ library which implements the protocol
  - Application-independent
  - Perl-prototype exists
- Simple first application: Hot-list sharing via Bro
  - As test of the library in a real setting

Summary & Outlook

- Semi-automated approach for inter-site cooperation
  - Automate the routine step but keep the human in the loop for decisions
- Designing and implementing of suitable framework
- Goal is to set this up for real
  - We have a couple of candidate sites (LBNL, NERSC, UCB, Univs. Munich)
- In the long run, framework has further potential
  - Generic design will allow other applications
  - Success of the semi-automated model might convince sites to try more
- Activity-logging infrastructure is useful by itself

Thanks ...

Robin Sommer

Lawrence Berkeley National Laboratory

rsommer@lbl.gov
http://www.icir.org/robin

This work is supported by the Office of Science and Technology at the Department of Homeland Security. Points of view in this document are those of the author(s) and do not necessarily represent the official position of the U.S. Department of Homeland Security or the Office of Science and Technology.