High-Fidelity Secure Network Provenance

Tyler Nichols, Adam Bates, Dave Tian, and Kevin Butler
Computer and Information Science Department, University of Oregon

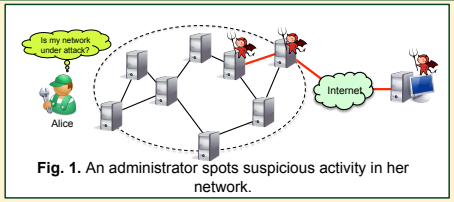

Fig. 1. An administrator spots suspicious activity in her network.

When an administrator, Alice, is running a data center and observes suspicious behavior, she must investigate whether the behavior is legitimate or malicious, as seen in the network infiltration attempt in Figure 1. To do so, a *network forensic* system is required in order to answer questions about the possible system breach. Past work has formalized this problem as a matter of *data provenance*, creating a *network provenance* system that gives Alice explanations as to the origins of network messages. *We present a novel design for a high-fidelity network provenance system by merging network activity with host-layer context.*

Network Provenance

Network provenance systems provide investigators with detailed records explaining why network traffic was generated at a given time to help identify faulty or malicious nodes. Existing network provenance schemes treat individual nodes in the network as black boxes by recording network traffic and by assuming that a message is generated as a direct result of other messages. This approach omits significant context from the internal state of the machine. Network traffic may be generated due to other systems events, such as inter-process communication. Without this host context, it is impossible to fully explain the cause of these network events.

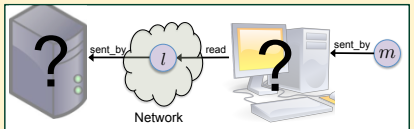

Fig. 2. Past network provenance systems have ignored the internal state of machines.

Host-Layer Provenance

Host-layer provenance produces records that detail every event that occurs with respect to system objects inside an individual machine. This provenance scheme tracks every object from the time it comes into existence up until its current state. Provenance data generation and verification occurs within the kernel, preventing rogue applications from tampering with the trusted computing base of the provenance monitor.

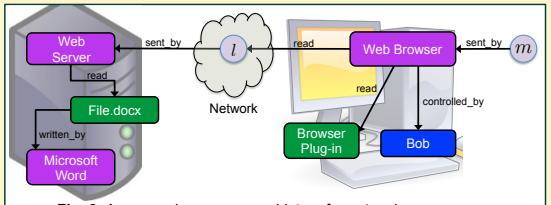

Fig. 3. An example provenance history for network message m.

Our approach is to utilize host-layer provenance collection in order to explain network events. Figure 3 shows a provenance graph that explains the network message m, which was derived from a combination of network events as well as internal events from two different hosts.

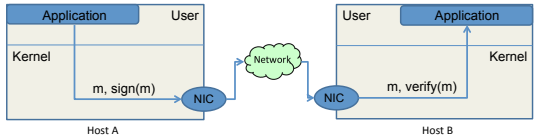

Fig. 4. Host A signs message m prior to exiting the system; Host B verifies the presence of A's signature prior to accepting message m

In our architecture, provenance monitors communicate via a cryptographic message commitment protocol. By implementing this functionality in the kernel, we remove the need to instrument individual applications, and ensure that provenance is collected even when an adversary has taken control of an application that is running in user space.

Software-Defined Network Provenance

In heterogeneous networks, such as those that run both Linux and Windows hosts, it is not currently possible to install a host-level provenance monitor in every machine. *We propose that the network itself can be used as a point of observation when this is the case.* Using Software-Defined Networking, a new paradigm in which network switches can be programmatically controlled, we can deploy a system of Provenance Verification Points (PVPs) to securely collect provenance for network events. This even enables provenance collection from hosts under attacker control. In future work, we intend to reconcile this network-layer provenance with the host-based approach, creating provenance of exquisite detail.

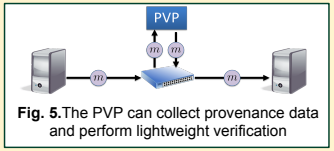

Fig. 5. The PVP can collect provenance data and perform lightweight verification

Conclusion

By combining host- and network-layer provenance mechanisms, we are able to generate records with extreme levels of detail, providing administrators with an all-encompassing view of their system. Furthermore, with the aid of Software-Defined Networking, network-layer provenance removes the need for individual machines to comply with the host-layer provenance protocol, which allows for greater coverage of heterogeneous systems.

OSRIS Oregon Systems Infrastructure Research & Information Security Laboratory

This research is supported by NSF Grant CNS-1254198.

For further information, contact
Tyler Nichols (*tnichols@cs.uoregon.edu*).