Proceedings of the 2003 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY June 2003

The Kerf toolkit for intrusion analysis

Javed Aslam, Sergey Bratus, David Kotz, Ron Peterson, Daniela Rus, and Brett Tofel
Department of Computer Science, and the Institute for Security Technology Studies, Dartmouth College
{jaa, sergey, dfk, rapjr, rus, btofel}@cs.dartmouth.edu
Poster to be presented by David Kotz

I. OVERVIEW

We consider the problem of *intrusion analysis* and present the *Kerf Toolkit*, whose purpose is to provide an efficient and flexible infrastructure for the analysis of attacks. The Kerf Toolkit includes a mechanism for securely recording host and network logging information for a network of workstations, a domain-specific language for querying this stored data, and an interface for viewing the results of such a query, providing feedback on these results, and generating new queries in an iterative fashion. We describe the architecture of Kerf in detail, present examples to demonstrate the power of our query language, and discuss the performance of our implementation of this system.

II. Goals and Assumptions

The goal of our *Kerf* project is to provide tools that aid system administrators in analyzing the nature and extent of an attack, and communicating the results with other administrators or law-enforcement agencies. A central tenet of the project is the recognition that human experts do, and will, play a critical role in the process of identifying, tracking and disabling computer attacks [ACKR01]. We also recognize that an important part of the discovery, analysis, and defense against new distributed attacks is the cooperation that occurs between experts across different organizations. Furthermore, many installations do not have the expertise necessary to develop full attack analyses. Thus, we are building semi-automated tools that help system administrators to (1) identify the characteristics of an attack given data from network and host-based sensors, (2) develop a hypothesis about the nature and origin of the attack, (3) share that hypothesis with security managers from other sites, (4) test that hypothesis at those other sites and coordinate the results of testing, and (5) archive the data necessary for use as evidence [Goa99].

III. Architecture Components

Database. Many intrusions involve multiple hosts, and the evidence for many intrusions may be seen in multiple types of logs. To support fast retrieval of relevant records, the logging host stores incoming log records in a database, indexing on important fields including host, facility, any IP address mentioned in the record, and any user name mentioned in the record. This approach also serves to isolate the log collection mechanism from the analysis mechanism, and to limit the amount of parsing, indexing, and searching that must be done within our analysis tool. The current implementation uses MySQL.

Domain-specific query language. Given the database of log records, the analyst could use SQL queries to search for relevant records. SawQL (pronounced SAW-quill) is our extension to SQL designed specifically to express a sysadmin's hypothesis about an attack. This extension provides four critical features:

- SawQL includes keywords to describe common features of log records, such as hostnames, IP addresses, and user names. In our implementation, the logging host parses each incoming log record to extract these fields for the database record, so later queries can quickly extract matching records.
- SawQL can express, and retrieve, *sequences* of log records. A normal SQL query describes, and retrieves, individual log records. Most attacks, however, involve a sequence of actions that are visible as a sequence of records in one or more log files. The goal of intrusion analysis is to piece together this sequence of actions.
- SawQL can express connections between records in a sequence, using variable names. The query execution engine correlates log records into sequences with consistent variable bindings. For example, service 'adduser' AND USER %newuser and service 'login' AND USER %newuser match two records that refer to the same user.
- SawQL can also express the temporal relationship between records in a sequence; for example, RELTIME +/- 5 minutes. In some attacks, the temporal proximity of two events is the critical feature in identifying the attack. This feature also serves to constrain the search. Our current SawQL implementation supports all of these features. The latter two features correlate records based on time or other features, and are a significant aid to the analyst. Using traditional tools such as grep, an analyst must take the results of one search, extract interesting elements (such as time, user name, or an

ISBN 0-7803-7808-3/03/$17.00

IP address), and run new searches on each one of them. Manual use of the command line, or writing ad-hoc scripts, can be error prone, time consuming, and difficult to manage. By building these features into the language of Kerf we hope to avoid those problems and speed the discovery of interesting links in the data.

Visualization. The centerpiece of the Kerf toolset is the *Landing* application, which provides a graphical interface to the sysadmin. Landing allows the user to enter SawQL queries, displays the results of the queries, and allows the user to provide feedback to the hypothesis engine. Our current implementation is complete, written in Java, although we intend to further explore and adapt the nature of the interface as we gain more experience with the intrusion analysis process and as we develop more advanced visualization techniques. Given the amount of log data collected from an organization's hosts, many queries will retrieve a large number of matching sequences. It is critical to help the sysadmin to organize and visualize these sequences. Our current implementation presents the set of sequences as a set of trees, and uses semantic compression to reduce the matching sequences to a set of patterns that describe those sequences. We intend to explore other approaches.

Hypothesis engine. Given a SawQL query from the sysadmin, Kerf extracts and displays the matching sequences. The GUI allows the sysadmin to mark each sequence "suspicious" or "innocuous", and to indicate the interesting elements of each suspicious sequence. Using any feedback provided (not all sequences need be marked), the engine uses algorithms drawn from the machine-learning community to suggest new queries that better fit the suspicious data, aiding with hypothesis re-finement. This component is under development. When complete, the hypothesis engine will also support extrapolation and generalization.

Secure logging One of the first actions taken by a hacker on a compromised system is to remove traces of their intrusion from the system's logs. Because we want Kerf to be of practical use, we crafted a simple system to securely log and store log information. We implemented a secure logging host that can receive, decode and store logging information from multiple sources. The logging host sniffs symmetric key Blowfish encrypted UDP datagrams from its networked clients, while the logging host itself does not have an IP address. Thus, the logging host is relatively secure from conventional Internet attacks. The current implementation accepts Unix syslogs and Windows Event Logs. We expect to add support for httpd logs, IDS events in IDMEF format, and network logs [MV02].

IV. Contributions

Kerf makes three major contributions: the domain-specific query language SawQL, the query engine that supports correlation of records by time and by feature (using variables), and the compressed visualization approach. Our poster will describe the architecture of the Kerf system, present some examples of the use of Kerf for intrusion analysis, discuss the performance of our initial prototype, and compare our work with related research.

REFERENCES

[ACKR01] Jay Aslam, Marco Cremonini, David Kotz, and Daniela Rus. Using mobile agents for analyzing intrusion in computer networks. In *Proceedings of the Workshop on Mobile Object Systems at ECOOP 2001*, July 2001.

[Goa99] Terrance Goan. A cop on the beat: Collecting and appraising intrusion evidence. *Communications of the ACM*, 42(7):46–52, July 1999.

[MV02] Andrew Mitchell and Giovanni Vigna. Mnemosyne: Designing and implementing network short-term memory. In *Proceedings of ICECCS '02*, Greenbelt, MD, December 2002. IEEE Computer Society Press.

ISBN 0-7803-7808-3/03/$17.00