The Kerf Toolkit for Intrusion Analysis

by Javed Aslam, Sergey Bratus, David Kotz, Ron Peterson, and Daniela Rus

Network-based intrusions have become a significant security concern for system administrators everywhere. Existing Intrusion Detection Systems (IDSs), whether based on signatures or statistical learning of normal behavior, give too many false positives, miss intrusion incidents, and are difficult to keep current with all known attacks. Although recent high-level correlation tools have improved the quality of alerts to system administrators [1], [2], IDSs have a limited success rate, tend to detect only known attack types, and ultimately result in only an alert message to a human administrator. (*In this paper, we will not discuss the relatively recent development of Intrusion Prevention Systems, that offer active response to an intrusion without human intervention*). Thus human experts are still required to analyze the alert (and related data) to determine the attack's exact nature. Human experts are also the key tool for identifying, tracking, and disabling new attack forms. This work often involves experts from several organizations working together to share their observations, hypotheses, and attack signatures. Unfortunately, few tools help these experts in the process of analyzing log data.

To alleviate this situation, we developed the Kerf toolkit (so named for a kerf, which is the slit made by a saw as it cuts through a log). Its goal is to provide an integrated set of tools that aid system administrators in analyzing the nature and extent of an attack and then communicating the results to other administrators or law-enforcement agencies. Kerf contains semi-automated tools that help system administrators identify attack characteristics based on data from network and host-based sensors, develop a hypothesis about an attack's nature and origin, express and share that hypothesis with security managers from other sites (without sharing actual log data, which may be sensitive for their organization), test the hypothesis at other sites, and coordinate the testing results.

Kerf and intrusion analysis

Picture the typical System Administrator, responsible for a collection of hosts on one or several organizational subnets. Each host logs its activity using the Unix syslog facility or the Windows Event Logging service. An IDS monitors some or all hosts—possibly the entire network—and generates and logs alerts about potential attacks. Once a system administrator discovers an attack, he or she must put on an analyst hat and further investigate (see Figure 1).

Kerf is intended to assist in this investigation, commonly referred to as intrusion analysis, after an attack is detected. We assume that correct and complete host and network logs are available, up to a point. To ensure this, Kerf includes agents installed in monitored machines that forward encrypted log records to a secure, off-host logging server (see Figure 2). The analyst goal, then, is to reconstruct evidence of an attack from individual event records in the available logs.

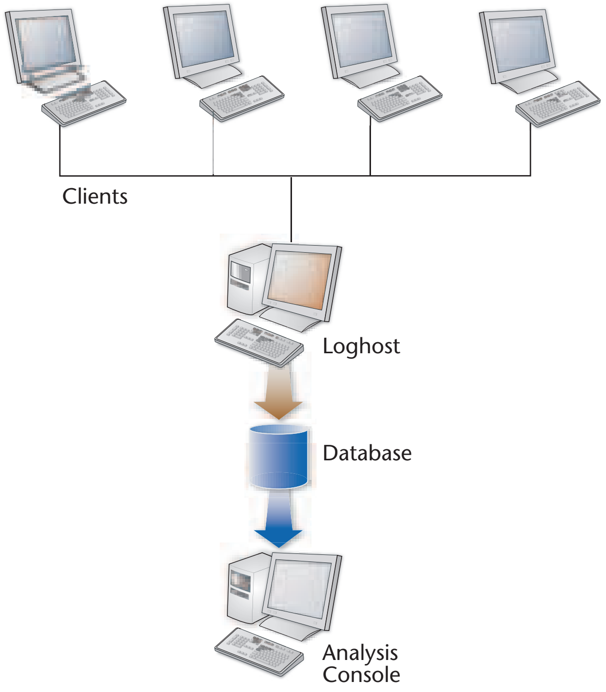

Figure 1: Overview of Kerf physical architecture

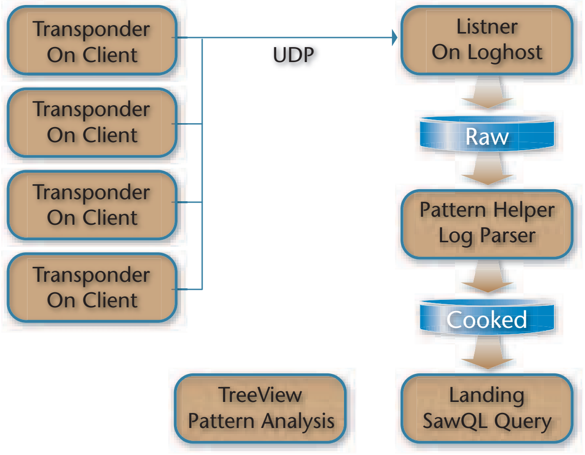

Figure 2: Overview of Kerf software architecture

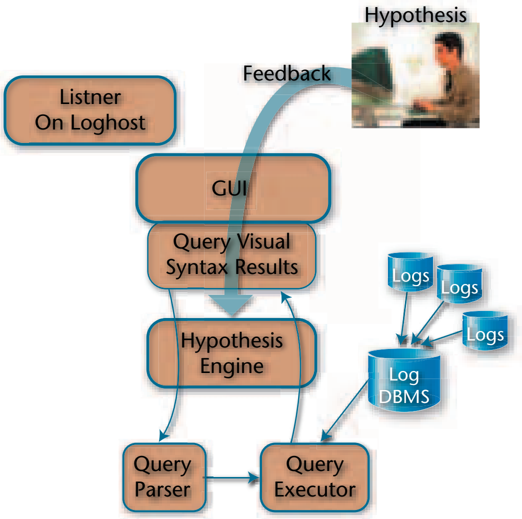

Figure 3: Hypothesis refinement: the Kerf approach

The analysis process is inherently interactive: an analyst begins with a vague mental hypothesis about what happened and then uses Kerf tools to test and revise that hypothesis (see Figure 3).

The process is also inherently iterative: each new piece of information permits the analyst to revise the hypothesis and explore further. The hypothesis is refined, as information that partially confirms it is discovered, and is expanded, as the analyst tries new approaches that broaden the investigation. The result is a specific hypothesis about an attack's source and nature and the concrete evidence to support the hypothesis.

Many tools for parsing text-based system logs currently available to system administrators [3], [4], [5] rely on extensions of *regular expressions*, which require syntactically complex constructions to search logs for relevant entries or to extract relevant parameters from them. This, in turn, often requires writing ad hoc scripts to correlate events from different logs or hosts. A number of tools that store parsed logs in relational databases, such as the Microsoft LogParser [6] (for which Burnett [7] is an excellent tutorial), permit users to express certain correlations in the form of Structured Query Language (SQL) queries with joins, but such expressions very quickly tend to grow intractable. In such conditions, any systematic recording of hypotheses, actions, and results for later study becomes very difficult. Because the analysis process is difficult and tedious, most system administrators can't fully explore and understand an attack or document it so that others can study it. Kerf aims to make intrusion analysis more efficient by providing the following:

- A secure mechanism for network and host logging to a dedicated log server, which keeps the logs' records in a relational database

- A correlation engine that accepts queries in SawQL, a domain-specific extension of SQL that is designed to concisely describe sequences of records correlated on their various parameters including their timestamps

- The PatternHelper tool to help a user write patterns for extracting parameters from free text-log formats, such as UNIX syslog

- A User Interface (UI) front end, Landing, that adaptively organizes large result sets from SawQL queries for more convenient viewing and analysis and permits a user to attach his or her own tags to records; in particular, to mark them as "suspicious" or "innocuous"

- The Hypothesis engine (under development) that aids a user with query generalization and refinement by learning from user feedback and adjusting the application's data organization algorithms or by suggesting new queries

Event correlation

It is natural to describe an intrusion as a sequence of events, some of which leave their traces in the form of records in various logs. These records are likely to be correlated on some parameters (e.g., the corresponding events may originate from or take place on the same host or may involve a logged common value associated with some protocol). Even more likely, they will be correlated on time (e.g., one event occurs before or after another, within a short period of time).

SawQL, the SQL-based Kerf query language, permits convenient expression of relative or absolute temporal and parameter correlations at the same time that it abstracts away the gory details of database joins. Thus queries in SawQL naturally represent sequences of correlated events and can be used to express and share hypotheses. Examples of SawQL expressions that describe actual intrusions can be found on the Kerf project Web site. [8]

Data organization and presentation

In the practice of intrusion analysis, there inevitably occurs a scenario in which a query returns many screenfuls of matching log records; each of which are full of diverse records; refining the query appears possible only after the majority of these records have been examined. In such situations, automated data organization algorithms that attempt to summarize and classify the data can save an analyst time and effort. Kerf uses entropy-based, recursive data organization algorithms to produce a tree-form representation of query results every time the results exceed a user-defined threshold size.

More precisely, the records are grouped by the unique values of their parameters. The order of grouping is chosen adaptively, based on the distributions of values of each parameter across the given result set. The resulting groups correspond to intermediate nodes of the tree, which are marked with the parameter values common to all records contained under a node. Thus the upper levels of the tree serve as a summarization of the result set.

An important side effect of this grouping method is that it will likely highlight "abnormal" events, which are of greatest interest in attack analysis. The data organization algorithm is tuned to produce trees of moderate depths and branching factors to aid the following typical tasks:

- Discovering the actual composition of result sets

- Understanding the distribution and ranges of selected parameter values and finding subsets of records with anomalous values

- Navigating to subsets of interest

- Extracting subsets of interest for use with another query

The snapshot in Figure 4 below shows a set of 1357 Snort portscan alerts, grouped first by destination port and then by source and destination IPs in Frame (A). Frame (B) summarizes the value ranges of other parameters in the selected group. Both Frames (A) and (B) can be used for user tagging of groups or individual records (not shown). Frame (C) accepts commands in an internal scripting language, and Frames (D) and (E) show status messages.

A user can add levels of grouping or define his or her own classification tree templates, bypassing the algorithm entirely or running it only on subtrees of a pre-defined classification. This method is useful in cases when the overall expected structure of the log data is well understood, whereas seeing where a new batch of records ends up in a pre-defined classification may provide a useful clue. All user operations on a dataset can be recorded and replayed on other comparable result sets. A user can also directly define his grouping and classification rules in an internal template language.

Kerf users will notice that the simplest operations on group nodes of a tree (i.e., subsets of the result set) are functionally similar to UNIX command chains—*"grep ... | sort | uniq -c | sort -n"* or *"select distinct ... group by ... order by ..."* statements of SQL environments—while providing much more flexibility in defining and connecting the filters and in keeping all records within a common and reusable classification framework.

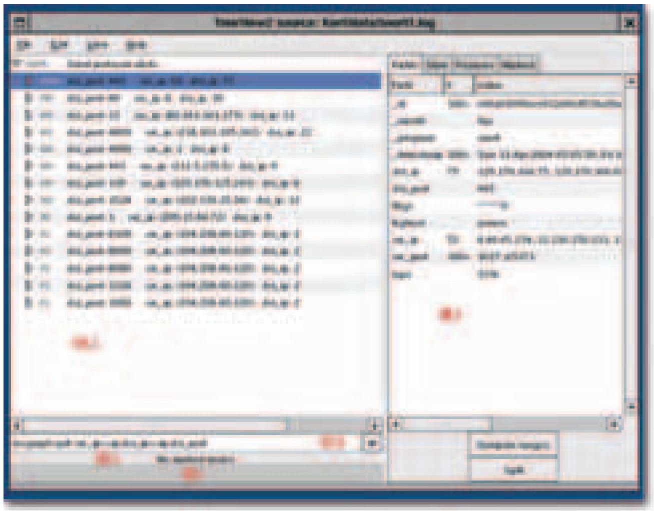

Figure 4: Application snapshot

An example of adaptive data organization

The following example shows how adaptive data organization can elucidate the structure of a moderately sized result set at a single glance. Here, a flat list of authentication records from an actual UNIX system log, selected by a simple query without correlation, is presented as an adaptively constructed tree. The user is a System Administrator who is concerned with logins from the network of a certain Internet Service Provider (ISP) and wants a brief summary of failed and successful logins. A query for login events from *.isp.net returns some 600 records.

Subsequently, it is determined that all logins originated from two legitimate users who happened to inhabit distinct dynamic IP ranges, one of whom was prone to typos. The feature pair (user, host) was found by the data-organization algorithm to produce the best tree form. The user was thus presented with a 12-line summarization of the 600-line result set. It also became clear that most logins came from one user and his login records were further grouped by month. (See Figure 5.)

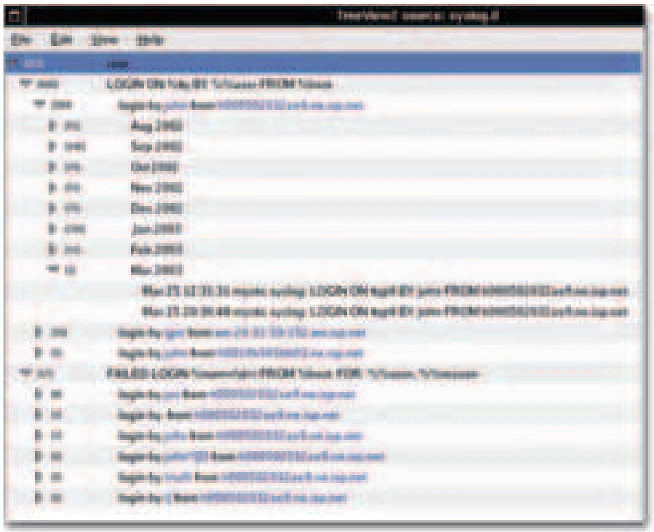

Figure 5. Treeview, some nodes expanded. The Kerf module for adaptive display of query results chose this summarization of the 600+ login events from *.isp.net.

Work in progress

In the near term, we plan to extend our system to handle other types of logs; in particular, IDS logs in the Intrusion Detection Message Exchange Format (IDMEF) and kernel audit logs, such as Sun Solaris BSM [9] and Linux Snare [10] and Syscalltrack. [11]

In the long term, a major goal of the Kerf project is to provide semi-automated tools to aid an analyst in hypothesis generation, refinement, archiving, generalization, and extrapolation. To this end, we are developing the following:

- A hypothesis engine, consisting of a hypothesis-generation module to assist a user in formulating the initial hypothesis
- A hypothesis-refinement module to assist in modifying the initial hypothesis to better target suspicious behavior
- A hypothesis-sharing module to assist in taking the final hypothesis and archiving it for later use, extrapolating it for other specific users and domains, and generalizing it for wider applicability

We expect our new algorithms and tools to be a unique contribution to the current state of intrusion analysis, by automating the existing best-of-breed analysis practices, and offering new powerful and flexible data organization techniques.

Acknowledgments

Marco Cremonini and Andrea Schiavoni designed and implemented an earlier prototype of a similar log-collection and analysis tool. We thank Giovanni Vigna and George Bakos for their helpful discussions and their assistance in locating host and network-log data. We also appreciate the input of other members of Dartmouth College's Institute for Security Technology Studies (ISTS), who commented on the current state of the art and the needs of IDS tools. Kerf is a project of the Institute for Security Technology Studies at Dartmouth College, supported under Award number 2000-DT-CX-K001 from the U.S. Department of Homeland Security, Science and Technology Directorate. 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 Science and Technology Directorate. ■

References

[1] Haines, J., Dorene Kewley Ryder, Laura Tinnel & Stephen Taylor. (2003, January/February). Validation of sensor alert correlators. IEEE Security & Privacy, 1(1), 46–56.

[2] Peng N, Yun Cui & Douglas S. Reeves. (2002). Analyzing intensive intrusion alerts via correlation. In Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection (RAID), Vol. 2516 of Lecture Notes in Computer Science, pp. 74–94. SpringerVerlag, October 2002. [Online] Available: http://link.springerny.com/link/service/series/0558/papers/2516/25160074.pdf

[3] Bird, T. & Ranum, M. Log analysis resources. http://www.loganalysis.org/

[4] Chuvakin, A. (August 2002). Advanced log processing. SecurityFocus.com [Online]. Available: http://online.securityfocus.com/infocus/1613

[5] Allison, Jared. (2002) Automated log processing.; login:, 27(6), 17–20.

[6] http://www.microsoft.com/windows2000/downloads/tools/logparser/

[7] Burnett, Mark (2003). Forensic log parsing with Microsoft's logparser. Loganlaysis.0rg [Online]. Available: http://www.securityfocus.com/infocus/1712, July 2003 7.

[8] http://kerf.cs.dartmouth.edu

[9] http://www.sun.com/software/security/audit/

[10] www.intersect alliance.com/projects/Snare/

[11] http://syscalltrack.sourceforge.net/

About the Authors

Javed Aslam

Javed Aslam is an associate professor in the College of Computer and Information Science at Northeastern University. His research interests include machine learning, information retrieval, computer security, and algorithm design and analysis. Aslam received a BS in electrical engineering and mathematics from the University of Notre Dame and a PhD in computer science from the Massachusetts Institute of Technology (MIT). He is a member of the Association for Computing Machinery (ACM) and the Institute of Electrical & Electronics Engineers (IEEE). He may be reached at jaa@ccs.neu.edu.

Sergey Bratus

Sergey Bratus is a postdoctoral research associate in Dartmouth College's Computer Science Department. His research focuses on applying machine learning and Artificial Intelligence (AI) techniques to intrusion analysis. He received his undergraduate education at the Moscow Institute of Physics and Technology and his PhD from Northeastern University. He is a member of Usenix and an associate member of the Free Software Foundation. He may be reached at sergey@cs.dartmouth.edu.

David Kotz

David Kotz is a professor of computer science at Dartmouth College, director of the Center for Mobile Computing, and executive director of the ISTS. His research interests include context-aware mobile computing, pervasive computing, wireless networks, and intrusion detection. He received an AB in computer science and physics from Dartmouth and a PhD in computer science from Duke University. He is a member of the ACM, IEEE Computer Society, Usenix, and Computer Professionals for Social Responsibility (CPSR). He may be reached at dfk@cs.dartmouth.edu.

Ron Peterson

Ronald Peterson is a senior programmer in Dartmouth College's Computer Science Department and owner of Peterson Enterprises, which develops PC-based, Musical Instrument Digital Interface (MIDI) musical instruments and graphics software. His research interests include cybersecurity, wireless sensor systems, cattle herding, mobile agents, and machine-vision interfaces for novel musical instruments. He received a BA in physics from Lawrence University. He may be reached at rapjr@cs.dartmouth.edu.

Daniela Rus

Daniela Rus is an associate professor in the Electrical Engineering and Computer Science department at MIT. Her research interests include distributed robotics, mobile computing, and self-organization. She has a PhD in computer science from Cornell University. She was the recipient of an National Science Foundation (NSF) Career Award, is an Alfred P. Sloan Foundation Fellow, and a Class of 2002 MacArthur Fellow. She may be reached at rus@csail.mit.edu.

continued from page 11…

"IATAC Spotlight on Research—Dartmouth"

has been engaged in identifying and addressing critical research areas required in cyber security and critical-infrastructure protection. One result of its efforts is the I3P cyber security Research and Development (R&D) agenda, which identifies critical gaps in cyber security and provides a list of recommended research priorities. [7]

The I3P Consortium recently launched two major cyber-security research projects that involve half the I3P's member institutions. Over the next two years, research teams will focus on developing models, tools, and technologies to protect SCADA systems used in the oil and gas industry and to gain a better understanding of the economic factors influencing cyber-security decisions. The first project, launched in March 2005 and led by Sandia National Laboratories, is an $8.5M effort to identify SCADA vulnerabilities and the interdependencies between SCADA systems and other critical infrastructures. [8] Researchers will develop metrics and models for assessing and managing SCADA security and will create next-generation SCADA systems with built-in security. The second research initiative, led by the RAND Corporation and worth $3M over two years, will help quantify the costs of cyber attacks and measure the effectiveness of current security tools and policies. [9]

I3P also supports a fellowship program designed to increase the number of cyber-security experts and researchers to fill the gap areas it has identified. [10] This program provides up to $150,000 in financial support for successful applicants. Five fellows are appointed each year, and the fellows are required to conduct research at one of the I3P member organizations. To be eligible for the program, research candidates must have received their doctorate no more than three years ago and have strong backgrounds in fields related to the gap areas. While the 2005 fellows have already been determined, a call for proposals will be released later this year for the 2006 program. ■

References

[1] http://www.ists.dartmouth.edu/

[2] http://www.ists.dartmouth.edu/cstrc/mission.php

[3] http://www.pqsnet.net/projects.php

[4] http://www.dartmouth.edu/%7Epkilab/

[5] http://www.ists.dartmouth.edu/er3c/mission.php

[6] http://www.thei3p.org/

[7] http://www.thei3p.org/about/2003_Cyber_Security_RD_Agenda.pdf

[8] http://www.thei3p.org/about/news/20050302_scada.html

[9] http://www.thei3p.org/about/news/20050516_econ.html

[10] http://www.thei3p.org/fellowships/index.html