GULP: A Unified Logging Architecture for Authentication Data

Matt Selsky and Daniel Medina – Columbia University

ABSTRACT

We have implemented the Grand Unified Logging Project, GULP, a flexible aggregation system for authentication log data. The system merges disparate logs stored across various servers into a single format according to an XML schema. This single format is logged to a database and queried via a web interface. The strength of this system lies in the ability to correlate information across multiple logging sources and display relevant information through a simple interface.

Introduction

At Columbia University, each person is given a unique username or "UNI" (University Network ID) by the Academic Information Systems (AcIS) group. A process is in place for UNIs to be activated, creating a password that allows access to various services.

These services run on many different hosts and have disparate logging facilities. For example, when a student logs into CourseWorks (web-based course management), a successful authentication record is stored in the CourseWorks database. Other logins, to CubMail (web-based email), CUNIX and PINEX (shell servers), and elsewhere, are similarly logged, but to other locations on local disk [2].

Logging data is stored in a variety of formats and is typically stored locally on the host which provides the service. Some example services and the logs formats they use:

UNIX wtmpx records (remote login to a server), unpacked from the binary format [5]:

```
dnm17 pts/42 mutie.cc.columbia.edu
       Fri Oct 29 09:21 - 10:08 (00:47)
```

Secure web servers run Apache and log in the common Apache text format [6]; see Display 1.

WIND (Web Identification Network Daemon) [3], which provides sign-on to various web applications, has a custom text format:

```
2004-10-29 09:21:00,000 Login - success
for dnm17:switchmgr (128.59.31.101)
[pass:...:...] r
```

CubMail is a webmail client based on the Horde's IMP project [4]. Logs are text-based; see Display 2.

We could send the logs via syslog to a remote host, but we currently have it configured to only log locally. Processing these logs is not as intensive as is seen in commercial enterprises, but our logs are still quite sizeable [15]. During the summer months alone, there are about 10 GB/week of logs from the main webservers, and another 9 GB/week from the mail servers. Early in the school year we have observed 15.2 GB/week and 12.5 GB/week for web and mail, respectively.

Table 1 on the next page lists the main services we provide, along with the methods they use for logging authentication events.

Problem

Web and other logs are already harvested for usage statistics. Typically, there is an operational need to determine the number of users of a service, the applications being used to access services (browser client, email client, etc.), and so on. These statistics are collected periodically, usually a few times each semester.

The authentication information contained in these logs is valuable for a variety of other purposes. We generate usage reports for budget requests and capacity planning, demographic reports to show which university divisions are using our services (and therefore which schools should purchase account upgrades for all their students), and client software reports to ascertain the software people are using to access our services. This helps us to determine utilization of site-licensed software and also to plan software to support in the future.

```
mutie.cc.columbia.edu - dnm17 [29/Oct/2004:09:21:00 -0500]
"(GET /sec/acis/networks/index.html HTTP/1.1)"
200 202573 "(ref -)" " (client Mozilla/5.0 (Macintosh; U; PPC
Mac OS X Mach-O; rv:1.7.3) Gecko/20040911 Firefox/0.10)"
```

Display 1: Common Apache text format log entry.

```
Oct 29 09:21:00 HORDE [notice] [imp] Login success for
dnm17@columbia.edu [128.59.31.101] to {localhost:143}
[on line 92 of "/etc/httpd/htdocs/horde/imp/redirect.php"]
```

Display 2: A custom text-based log entry from CubMail.

Authentication information (UNI and password) gives access to an individual's personal information, including payroll, financial aid data, grades and course registration, email, and personal contact information.

We use Kerberos [1] to provide a centralized authentication mechanism, but Kerberos logs lack interesting features (since the transactions are brokered between the Kerberos server and the service or application server). We wish to preserve as much client information relevant to the authentication as possible. Several of our services do not natively support Kerberos so they request a ticket on behalf of the user using the plaintext password. This type of event shows up in the Kerberos logs without any information about the end-host or any indication that the authentication attempt was successful; the log only shows that an initial ticket request was made from the service host for the user.

The AcIS security group regularly receives requests regarding personal login information from persons who believe their accounts are compromised. A decade ago, before the explosion in web-based applications, it would suffice to direct users to the onall [7] command, which in conjunction with the last command, would show the most recent logins on all of the UNIX timeshare hosts. last logs were never centralized as in [17], but the onall utility made searching for these standard logs easier.

The security group also receives requests to determine the owner of a particular host (usually in the form of an Internet Protocol (IP) address). Due to the model of local network access at Columbia (so-called "free love"), users are not required to log into the network to use it [8]. IP addresses may be linked to users via a multi-stage network logging procedure (mapping IP addresses to hardware addresses to switch ports to room information to room registration), or via authentication information. The latter is preferable when available.

A recent problem for the security group has been various applications (sometimes called "SpyWare") that proxy user web traffic. Some of these applications proxy not just normal web traffic, but also SSL-protected (HTTPS) traffic.[1] This is equivalent to the user sending their UNI and password information to a third party (along with the data contained in the pages she is visiting).

Since the Kerberos logs are not suitable for harvesting the authentication data we want, we must look to the logs stored on the individual servers, as described above. However these logs are not easily searchable through any standard interface.

Too often, the AcIS security group finds itself in a reactive position, responding to incidents that have become operational threats (such as mass account compromise, break-ins, and malware epidemics). During these incidents, there is typically a slow gathering of available information that may take hours and involve numerous staff members. Once logging information is centralized, we can gather this information much more rapidly. The application of data mining techniques may even enable proactive operations in the face of emerging threats.

Solution

We have implemented a flexible aggregation system that allows for easy querying of the relevant authentication data from disparate log sources. We gave the solution the moniker "GULP" (Grand Unified Logging Project). This was preferable to the half-serious "TIA" (Tracking ID Access/Total Information Awareness) and "ECHELON" (Experimentally Centralizing Host's Every Log-On Name). We examined several mechanisms for extracting useful features from this data.

Centralized Logging

To centralize the authentication information we want, we transform the log files to an XML file that is described by an XML schema. This document includes only the "interesting" information that we have defined, reducing the total amount of data retained.

Centralized Searching

This single format is then searched for useful information via a web interface. The advantage of this system lies in the ability to correlate information across multiple logging sources easily. Advanced searches can be defined and saved for future re-evaluation.

Data Mining

With the limited features we have from our authentication data, we can extract information regarding abnormal behavior. For example, evidence of spyware or a proxy server would be many different users connecting from a single source or network (off the campus network). A login from a source not seen before may indicate unauthorized access to an account. The ultimate goal, however, would be to provide a system to allow a

[1] By installing a trusted root certificate [9].

Service	*Function*	*Logging*
CUNIX	Shell servers (General purpose)	local, wtmpx
PINEX	Shell-based E-mail	local, wtmpx
CubMail	Web-based E-mail	local, custom
CourseWorks	Course-related materials	remote DB
Secure Web	SSL-protected pages on www1	local, Apache
WIND	Web-app sign-on platform	local, custom
RADIUS	VPN and dialup authentication	local, RADIUS detail

Table 1: Main services provided.

member of the security group to create rules as needed, rather than using hard-coded signatures.

Implementation

Centralized Logging

The advantage to using XML is that we can publish our schema and leave the responsibility for extracting the relevant data from the logs to the maintainer of the application generating the logs. The maintainer can then validate the generated document before contributing it to the central repository.[2] While we did not interact with any outside parties for the purposes of this project, it is easy to perform the required validation. The schema used is included in the Appendix.

The XML representations of the log files may then be transferred and stored in a relational database (MySQL, in our case) [10].

For our project, we used logs from UNIX timeshare hosts (CUNIX and PINEX), web application log-in servers (WIND), secure web servers (WWWS), and webmail (CubMail), all of which we have described above. We chose the timeshare hosts because ‘‘last’’ logs are traditionally important sources of remote login information (although fewer users now log in directly). We chose WIND logs because it controls access to some of the most important web-based applications at the university, including payroll information. We chose secure web logs because the format is very common and numerous proxies can be found and accessed via the secure web servers. We chose CubMail because it is a popular application, used by approximately 60% of our users.

We created simple parsers for each of the logs we intended to use. In the case of the ‘‘last’’, Apache, and Horde log parsers, we hope to have decent reuse potential, while WIND logs will probably be unique to our site.

Writing the parsers is fairly simple, with most parsers being less than 150 lines of code (and much of the code just setting up the connection to the database). The difficult parts of the parser are writing the regular expressions to extract revelant data from each log entry in the log file, dealing with disparate date and time formats, and reading binary log data. The more complex parsers also need to attempt to re-create session information using login and logout records that may not match up. We do not currently attempt to canonicalize usernames to the UNI since we do this in the web interface, by searching for the UNI and any other usernames associated with a person.

Centralized Searching

We created a simple web form, protected by an `.htaccess` file restricting access to our Security group, that allows searching via a username or remote IP address. The information returned includes the remote hostname, the service used, the local server that exported the log, the start and end time of the session (only the start time where the session concept does not apply), and a note if applicable (such as the TTY used, or the web page or service accessed). A link to an external WHOIS site is included for more information about the remote host [11].

The username and remote address are also linked back to the CGI to allow easy inversion of the search on either term. This type of inversion is quite typical (when, say, trying to determine what other logins have come from a strange address). In this manner, we have slightly more features than a simple log-grepper allows.

We have also created an additional web form to show users their own authentication history, after logging in, but we have not deployed it as of yet.

Data Mining

We had several pre-conceptions about what would constitute an anomalous login. We conjectured that there would be two kinds of (global) abnormalities: many connections coming from a single address for many different users; and a single user logging in from many different locations.

We collected frequency statistics for both of the above abnormalities and quickly discovered that our assumptions were not refined enough. Of 40,000 user accounts observed over three weeks, over 10,000 were seen from more than six remote locations. Almost 6,000 were seen logging in from over 10 locations. (10 users from a German ISP logged in from more than 60 distinct locations within that ISP’s address space).

We attempted to use BGP [12] information to retrieve the Autonomous System (AS) number associated with each remote IP address. In this case, a user’s 60 remote locations would be represented by a single AS number (belonging to the ISP). According to these new metrics, we find that most users typically log in from fewer than three ASs (work, home, and the Columbia network). Nevertheless, with 40,000 users, we will still experience many false alerts (and miss many legitimate violations).

In the other direction, we found almost 900 addresses (of over 120,000) from which more than 12 users logged into our systems. Of these, we know that some are classified by our security group as malicious proxies (as described above). Many of the remaining addresses belong to benign web proxies, NAT-ing routers, and corporate gateways.

Applications

In the end, the search tool is more useful for revealing anomalous behavior than a global set of rules. Allowing limited access and providing a useful interface, we can deploy the search tool to the larger

[2] Experience has shown that publishing a required schema that cannot be easily validated by the source and repository parties is pointless.

user community. Colorizing logins from various sources by network, a user may easily audit her own login history [13, 16].

A user is more likely to be aware of what qualifies as an anomalous login than a system administrator responsible for thousands of users (especially given our diverse user population and the limit of the log records we are processing). Making this record available to the user is no different than a phone bill or credit card bill, an itemized list that the user can use to check for fraudulent activity and transactions. We show an example of this report in Figure 1 below.

We used this user search tool to investigate twelve recent security incidents reported by end-users. One such incident involved a student travelling overseas; he had used a computer terminal ‘‘administered by a guy who admitted to me in a moment of intoxication that he’s a criminal hacker.’’ Needless to say, the student was concerned about the security of his account.

In six of the twelve cases, the tool confirmed the suspicions of the user that someone else was using their account. In the other cases, we did not observe any anomalous patterns, possibly due to either gaps in our data or gaps in our coverage (we are not yet collecting data from all available log sources). In one instance we identified a supposedly secure web application on a departmental server that was in fact using plaintext ftp for file uploads to CUNIX.

In another incident, the identity of the miscreant was discovered. A student suspected that someone was reading her email because she often found her message flags altered. Using the search tool, the security group found a number of abnormal logins from a public campus terminal. Inverting the search on the public terminal, they found that the same individual had logged in to the terminal before the complaining student. Apparently, the miscreant would check his mail first, then hers.

The security group can also create custom searches as required. The search shown in Figure 2 below quickly identifies all malicious MarketScore proxies (as defined above). When these proxies were first identified, it took two days to formulate the entire list of proxy sources and countless staff time since different staff members were familiar with the log locations and contents for different services. Currently, logs are collected of users of the proxies once a week. With this tool, a current list can be created instantly.

Future Work

Numerous areas for development are open to us now that we have a viable central logging system. We also see a number of improvements that can be made to the applications we have already created.

We will take steps to properly normalize the username (certain logs do not record the UNI of an individual and instead log a username, which in the case of staff members, may not be equivalent). We have currently handled this in the web form.

We will further improve the idea of a ‘‘session’’ by correlating login and logout messages from some of the sources that did not clearly identify records as belonging to a particular session (as has been done with CubMail logins).

We will expand the logs that we feed into the system, including POP, IMAP, authenticated SMTP, RADIUS, and CourseWorks logs.

medina	160.39.246.251	dyn-wireless-246-251.dyn.columbia.edu	cunix	walnut	2004-11-29 17:11:25	2004-11-29 17:22:19
medina	160.39.246.251	dyn-wireless-246-251.dyn.columbia.edu	cunix	banana	2004-11-29 18:09:58	2004-11-30 01:24:30
medina	70.19.109.194	pool-70-19-109-194.ny325.east.verizon.net	cunix	papaya	2004-11-29 23:50:55	2004-11-30 00:28:36
medina	70.19.109.194	pool-70-19-109-194.ny325.east.verizon.net	cunix	mango	2004-11-30 07:54:22	2004-11-30 09:09:55
medina	128.59.25.155	dynamic-25-155.dyn.columbia.edu	cunix	mango	2004-11-30 10:46:48	2004-11-30 12:57:39
medina	128.59.31.101	mutie.cc.columbia.edu	cunix	hazelnut	2004-11-30 13:04:35	2004-11-30 17:12:46
medina	128.59.59.215	manheru.cc.columbia.edu	pinex	persimmon	2004-11-30 22:11:45	2004-11-30 22:16:02

Figure 1: Sample search on user medina.

user2111	216.148.246.70	proxys.sj3.marketscore.com	CubMail	jujube	2004-12-06 01:05:04
user2111	216.148.246.70	proxys.sj3.marketscore.com	CubMail	jujube	2004-12-06 01:11:11
user2131	66.119.34.39	proxys.ia2.marketscore.com	CubMail	durian	2004-12-06 01:16:47
user317	170.224.224.102	proxys.or3.marketscore.com	CubMail	passionfruit	2004-12-06 08:35:15
user2113	170.224.224.70	proxys.or2.marketscore.com	CubMail	jujube	2004-12-06 09:04:10
user2102	216.148.244.70	proxys.sj2.marketscore.com	CubMail	jujube	2004-12-06 09:18:59
user2113	170.224.244.70	proxys.or2.marketscore.com	CubMail	jujube	2004-12-06 09:19:18
user55	216.148.246.134	proxys.sj4.marketscore.com	CubMail	passionfruit	2004-12-06 09:33.42

Figure 2: Sample MarketScore logins (users obfuscated).

We will improve the user-facing application to query personal login information. Any user tools that decrease support staff time are a boon.

We will further evaluate possible machine learning algorithms with more satisfactory error rates, and possibly incorporate these algorithms into the user-facing tool.

We will look at using real-time log-processing frameworks, such as SHARP, to collect information as it is available [14].

We will further research more widely-used standards for sharing logging messages, such as Internet2's ccBAY and Conostix's IPFC [18, 19].

Availability

This paper and related code can be found online at http://www.columbia.edu/acis/networks/advanced/gulp/ .

Author Information

Matt Selsky earned his BS in Computer Science from Columbia University. He has been working at Columbia University since 1999, most recently as an engineer in the UNIX Systems Group. He works on e-mail-related services and is currently pursuing an MS in Computer Science from Columbia University. Reach him electronically at selsky@columbia.edu.

Daniel Medina completed his BS and MS in Computer Science at Columbia University. Since 2002, he's worked in the Network Systems Group at Columbia University. He can be reached at medina@columbia.edu.

Bibliography

[1] *Kerberos: The Network Authentication Protocol*, http://mit.edu/kerberos/ , Accessed 7 December, 2004.

[2] ''Columbia's Central UNIX Hosts,'' http://www.columbia.edu/acis/sy/cunix/ , Accessed 2 December, 2004.

[3] *Restricting Access: WIND*, http://www.columbia.edu/acis/webdev/wind.html , Accessed 2 December, 2004.

[4] *IMP Webmail Client*, http://www.horde.org/imp/ , Accessed 6 December, 2004.

[5] *wtmpx – utmpx and wtmpx database entry formats*, Sun Online Manual Pages, Accessed 2 December, 2004.

[6] *Apache HTTP Server Log Files*, http://httpd.apache.org/docs/logs.html , Accessed 6 December, 2004.

[7] *onall – Run a command on a group of hosts*, AcIS Online Manual Pages, Accessed 2 December, 2004.

[8] Kundakci, Vace, *'Free Love' and Secured Services*, EDUCAUSE Review, pp. 66-67, http://www.educause.edu/ir/library/pdf/ERM0266.pdf , Nov/Dec, 2002.

[9] *doxdesk.com: Parasite: MarketScore*, http://www.doxdesk.com/parasite/MarketScore.html , Accessed 6 December, 2004.

[10] *MySQL: The World's Most Popular Open Source Database*, http://www.mysql.com/ .

[11] *Whois Proxy*, http://grove.ufl.edu/bro/ , Accessed 6 December, 2004.

[12] Rekhter, Y. and T. Li, *RFC 1771: A Border Gateway Protocol 4 (BGP-4)*, March, 1995.

[13] Takada, T. and H. Koike, ''Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs,'' *Proceedings of the 6th International Conference on Information Visualization (IV '02)*, July, 2002.

[14] Bing, M. and C. Erickson, ''Extending UNIX System Logging with SHARP,'' *Proceedings of the 14th Systems Administration Conference (LISA 2000)*, December, 2000.

[15] Sah, A., ''A New Architecture for Managing Enterprise Log Data,'' *Proceedings of the 16th Systems Administration Conference (LISA '02)*, November, 2002.

[16] Takada, T. and H. Koike, ''MieLog: A Highly Interactive Visual Log Browser Using Information Visualization and Statistical Analysis,'' *Proceedings of the 16th Systems Administration Conference (LISA '02)*, November, 2002.

[17] Finke, J., ''Monitoring Usage of Workstations with a Relational Database,'' *Proceedings of the 8th Systems Administration Conference (LISA '94)*, September, 1994.

[18] Internet2, *MW-E2ED Diagnostic Backplane Pilot Effort (ccBay)*, http://middleware.internet2.edu/e2ed/public/pilot/pilothome.html , Accessed 29 July, 2005.

[19] Conostix S. A., *IPFC (Inter Protocol Flexible Control)*, http://www.conostix.com/ipfc/ , Accessed 25 July, 2005.