Join the Discussion
Connect

Log Parser Lizard

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle), USA Chapter

Prerequisites

—Windows
—Microsoft Log Parser 2.2[1]
—Microsoft.Net 3.5[2]

At RSA Conference 2012, I gave a presentation called "Evil through the Lens of Web Logs." This presentation is built on research I'm conducting for a SANS Gold paper for graduate school and pays particular attention to SQL Injection and Remote File Include attacks. One of the tools discussed as very useful for analysis tactics is Log Parser Lizard.[3] You're probably familiar with Log Parser, but I'll bet you didn't there was a great GUI-based tool with which to leverage its raw power with ease.

Log Parser Lizard is the brainchild of Dimce Kuzmanov, a Macedonian software engineer, who started Lizard Labs in 1998. In 2006 while also working as a part-time sysadmin on financial systems, Dimce recognized that he was using Log Parser on a daily basis for creating reports, analyzing logs, automatic error reporting, transferring data with txt files, etc. Over time his collection of queries became unmanageable and difficult to maintain, so he created Log Parser Lizard (LPL) for his personal use, and having benefited from free software himself, he wanted to release a useful freeware product to give back to the community. While LPL very successfully harnesses Log Parser's capabilities, Dimce firmly believes that as a great UI it help users learn and organize their queries with less effort. When he added log4net and regex input support, the Log Parser community really began to embrace LPL.

LPL releases are a bit sporadic, usually based on a few new features, bug or code fixes, and future releases are planned but not with a known frequency. Today LPL has a user base of about 2000 installations each month, based on trend analysis for the last three years and approximately 80,000 users worldwide.

The current production release of LPL is 2.1 and features include:

- Ability to organize queries along with an improved source code editor that includes enhanced source navigation and analysis capability, syntax-highlighting, automatic source code completion, method insight, undo/redo, bookmarks, and more.
- Support for Facebook Query Language (FQL). This feature was introduced to help Facebook developers organize their queries.
- Code snippets (code templates) and constants. Log Parser Lizard also supports "constants" binding to static/shared properties from Microsoft.Net.
- Numerous other user-interface features including advanced grid with filtering and grouping as well as support for charts without requiring a Microsoft Office installation as is a dependency for a standalone instance of Log Parser.
- Support for printing and exporting results to Excel and PDF documents (for registered users ($26.51 USD)).
- Support for inline VB.Net code to create LogParser SQL queries.

Inline VB.net support allows you to drop your code between <% and %> marks; it will then be executed and the resulting string will be replaced in the query. Lizard Labs believes this feature will be very useful for LPL users. Before parsing logs you can move-copy-rename files, download via FTP, shutdown IIS, etc. You can also use .Net data types like DateTime for arithmetic operations and/or System.Environment settings in query parameters.

As I write this I'm testing the beta for LPL 2.5 and the new feature set includes:

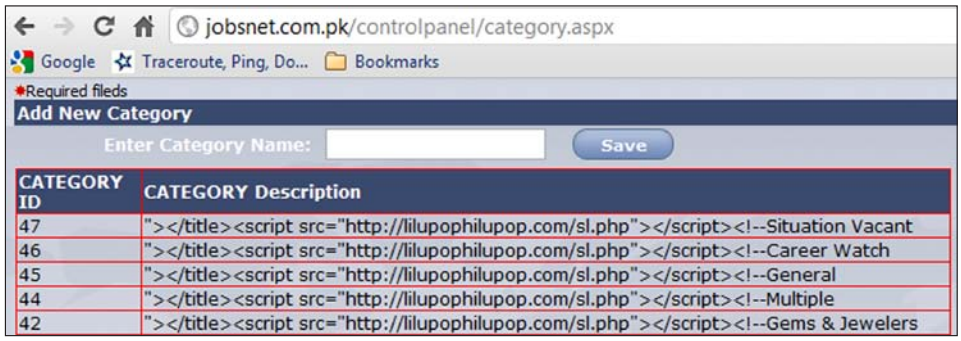
- Conditional field formatting (color, font, size, image) to identify required information. As an example, you can set the conditions to change error colors to red, warnings to yellow, etc., or highlight a specific field if it contains a string value of interest.
- Store and organize queries in SQL Server database for ease of use among multiple users and computers in an organization as well as backups, auditing, and all other benefits that database storage allows.
- Excel-style row filtering.
- Ability to add columns with Excel-style formulas (with most Excel functions) and support for exporting in Excel 2007 format (more than 65365 rows).

1 http://www.microsoft.com/downloads/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en.

2 http://www.microsoft.com/downloads/en/details.aspx?FamilyId=333325fd-ae52-4e35-b531-508d977d32a6&displaylang=en.

3 http://lizard-labs.net/log_parser_lizard.aspx.

Figure 1 – Lilupophilupop victim site



What would a *toolsmith* article be without a tool roadmap, so let's not break a good habit, eh? LPL 3.0 will likely include out-of-the-box queries for IIS web reports (as in other commercial log analysis products), support for query execution scheduling, reports sent via email from LPL, command line support, a query builder tool, text file input format (where a single file is one record and fields can be extracted with RegEx or with Log Parser functions), and improved log4net input format. As with most of the tools we discuss, Dimce is certainly open to good ideas for the product and welcomes feedback and ideas from the user community. In total fantasy land the future of LPL may even include queries "in the cloud," an LPL ASP.net web app that can be installed right on the server, a web service supporting LPL, mobile apps that can use this service, and a global query dictionary that users can submit, comment, and rate the queries. "The future's so bright, I gotta wear shades." Whoa, 80s flashback, sorry.

Using Log Parser Lizard

Installing Log Parser Lizard is so straightforward it doesn't even warrant a section. Ensure you have Log Parser and .Net 3.5 installed, then execute the LPL installer. Finito.

As described above, I've been working on research for a paper which includes analysis of a mass SQL injection attack, well described in detail this past December by Mark Hofman on the SANS Internet Storm Center Diary.[4] In addition to Mark's analysis, this popular post included many comments and replies from readers who had suffered or noted the attack in their logs and even some helpful folks who submitted log samples. You likely remember the LizaMoon attack, and the Lilupophilupop attack was quite similar. In both cases, injected sites offered a URL that then caused redirection to a fake antivirus offering. Specifically, `</title><script src="hxxp://lilupophilupop.com/sl.php"></script>` was embedded in victim sites where sl.php bounced you to the likes of `hxxp://ift72hbot.rr.nu`, then on to rogue AV. I actually had to look up the .rr.nu TLD; it's the Republic of Moldova and has been implicated recently in massive SPAM campaigns as well as the current WordPress hacks (as of this writing).

Figure 1 represents a victim site still exhibiting typical signs of compromise.

Victim sites were most often running ASP.net apps on IIS with MS-SQL back-ends. It was quickly learned that a few identifying traits of the Lilupophilupop attack included the fact that a rather large hex blob was evident in IIS logs. I've always found that checking logs for 500 errors when analyzing for SQL injection attacks can typically point you down the right path. Using a log file submitted by an ISC reader (anonymized for obvious reasons), I first built a query to seek ASP application errors from a default query included in LPL. I launched LPL, clicked *IIS Logs*, then *ASP App Errors*, replaced `#IISW3C#` in the `FROM` statement with the path to my anonymized log file, and finally clicked *Run Query* as seen in figure 2. Email me if you'd like me send you the log file so you can experiment for yourself.

Using this query, including `FROM D:\logs\lilupophilupop\ex111201anon.log WHERE (sc-status = 500) AND (cs-uri-stem LIKE '%.asp')`, prior to being aware of *lilupophi-*

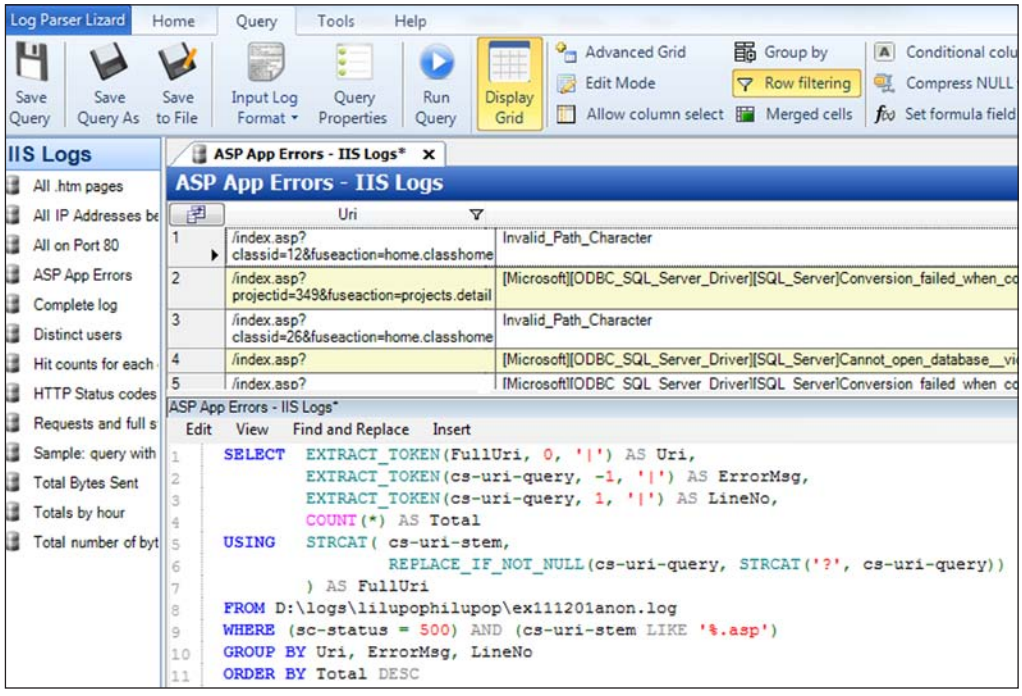
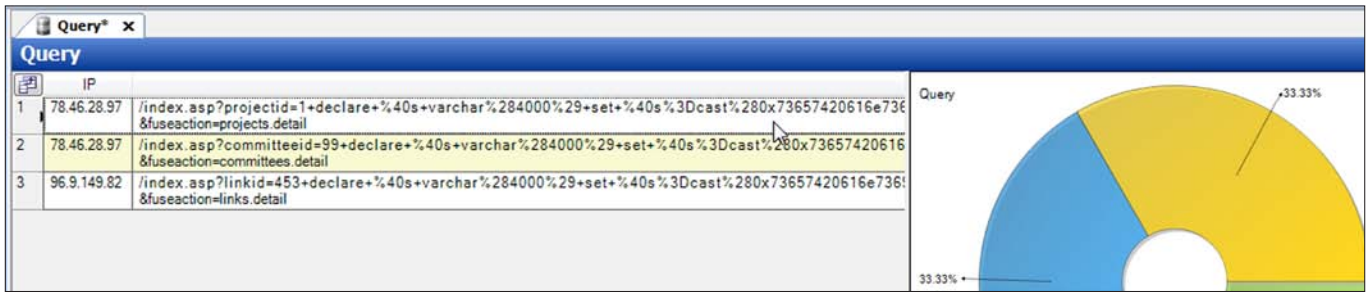


Figure 2 –LPL parsing error messages

4 http://isc.sans.edu/diary.html?storyid=12127.

Figure 3 – LPL parsing DECLARE statements



lupop as a keyword or part of an injected URL, would have immediately narrowed the search vectors.

Also common to attacks of this nature might be a `DECLARE` statement (defines variable(s)) visible in logs. A query as seen in figure 3 produced three results that included a `DECLARE` statement followed by a `CAST` (converts an expression of one data type to another) statement wherein an attempt to pass the hex blob to the backend was noted.

I clicked in one the results from 78.46.28.97, chose *Select All*, then *Copy*, and dropped the content to a text editor. I then grabbed the hex from just after the `CAST` statement to just prior to the `AS VARCHAR` statement and copied into a Burp Suite decoder window and chose *decode as ascii hex*.

Figure 4 shows the converted attack string.

Long and short of it, the attack loops through all columns in all tables and updates their values by adding JavaScript to point to `hxxp://lilupophilupop.com/sl.php`.

This took all of five to ten minutes with LPL and a little experimentation. Yes, you can do all of this with Log Parser at the command line, but if you're looking for strong query management, tidy reporting exports including charts, and downright convenience, LPL is the way to go.

In conclusion

Log Parser Lizard is one of those indispensable tools that treads lightly on your system but offers a huge bang for the buck. Free or $26? Puhleeze. Keep in mind that while I used an IIS log sample for the article you can throw LPL at generic XML, CSV, TSV and W3C based logs all day long. Download it and put it to good use right away. Dimce would love to hear from you, and I look forward to hearing your success stories.

Ping me via email if you have questions (russ at holisticinfosec dot org).

Cheers…until next month.

Acknowledgements

—Dimce Kuzmanov, lead developer and founder, Lizard Labs

About the Author

Russ McRee leads the incident management and penetration testing functions for Microsoft's Online Services Security team. He advocates a holistic approach to information security via holisticinfosec.org and volunteers as a handler for the SANS Internet Storm Center. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

Figure 4 – Burp decoder converts hex