

Technical White Paper

October 17, 2003 www.TechPathways.com

Christopher L. T. Brown, CISSP

clbrown@techpathways.com

The Art of Key Word Searching

Abstract

Keyword searching in computer forensics can make or break an investigation. Choosing the wrong search terms may cause you to miss vital evidence, or may return so many hits that you spend hours looking for a needle in a haystack to find any real evidence. Given the size of today's computer disks, each search can take many hours, causing delays and frustration unless your search terms are well chosen. This paper is intended to show how a thoughtful approach to keyword selection along with a few simple operators will often net the desired results quickly and with fewer CPU cycles.

Background

Searching for evidence in computer forensics has long been thought of as a difficult task requiring hours of thought to create complex GREP expressions. GREP is an acronym for "global regular expression print" (sometimes also referred to as "globbing regular expression print"). When GREP was created in the 1970's users needed a way to search for regular expressions (mathematical formula) in thousands of lines of programming code without a graphical user interface. While it's hard to argue with the power of GREP, most of today's computer forensics needs can be fulfilled by keyword searching using a few simple operators. Using GREP in most computer forensics cases is like using a Monster Truck to run an NASCAR race. Newer keyword techniques, specifically developed for searching large lexicons are far more efficient and are as effective for use in computer forensics.

For those who are somewhat familiar with GREP, you may have seen something similar to this before `grep "([^\()]*a" file.c`. In this example GREP would return any line from file.c containing a pair of parentheses that are innermost and are followed by the letter "a". This is precisely the type of search GREP was created for; finding complex regular expressions (mathematical expressions) within large amounts of source code. From a computer forensics standpoint this type of search is overkill when typically the investigator is looking for text within a specific lexicon. Despite the overkill even today many computer forensics tutorials and formalized

multi-day classes focus on how to create a complex searches as shown above. Even more time is spent perfecting the student's use of regular expression syntax in an effort to reduce user mistakes. Before you know it the student is staying up all night creating complex expressions to find every instance of the word "DOG" without the word "CAT" falling within the following 8 characters after the word "DOG". Of course once the student is back at the lab they quickly begin impressing their colleagues by applying the same syntax focused thought process towards each search.

In the aforementioned scenario the student has fallen into a trap focusing on the process and tool rather than the primary focus... **what** they are looking for and **where** do they need to look?

What are you looking for?

In most cases investigators are looking for references to a Person, Place or thing

One of the easiest ways to narrow down the search results is to search for "whole words" or EXACTLY rather than words LIKE or "words containing". As an example without searching for specific criteria most search systems will default to LIKE behavior so inputting "Chris" will normally return all instances of "Chris", "Christopher" and "Christine". Using a simple operator such as "whole words" or EXACTLY can significantly reduce the unwanted search returns.

Case sensitivity is another operator which can help reduce the number of unwanted results. In a situation where you are looking for a person named "Richard Green" searching for "Green" with case sensitivity turned on would help reduce the color green search hits.

In other cases the investigator may be looking for a specific document for which they know the contents or maybe they are looking for a specific phrase. Using multiple word phrases is one of the best ways to fine tune searches to eliminate unwanted search hits. This is an area where investigators are often tempted to create complex GREG expressions when a simple exact phrase will do the job. Just as using case sensitivity in the example above would have helped limit the color green returns searching for the *phrase* "Richard Green" would also help eliminate any undesirable search results.

Looking for unique and misspellings is another helpful tool in finding specific documents. In a recent case where a bank robbery suspect's computer was seized, there was very little incriminating evidence. Searching for misspellings common to several of the demand notes handed to the tellers quickly netted fragments of the demand notes that had been typed and printed, but never "saved" on the seized computer.

Boolean logic

Only three operators are necessary to add a great deal of power to the search operators we have already discussed. The three operators are the Boolean logic operators AND, OR and NOT. The beauty of the three Boolean logic operators is that they do just what you would

expect. Searching for “dog AND cat” would result in all documents containing both the search terms dog and cat. Search for “dog NOT cat” would result in all documents containing dog but not documents which also contained cat. Searching for “dog OR cat” would result in all documents which contained either dog or cat. Most search implementations use an implied AND operator when more than one search term is given in a phrase.

Nested Searches

Another way to approach searching is to perform nested searching, that is to say search for all documents with “Richard Green” then search the resulting documents for yet a second search term such as “money”. While you can accomplish this type of nested search in a single regular expression, often the results of the first search are just as interesting as the second.

Searching for a broad set of search terms is often the starting point in civil discovery where the first set is searched then a second search term to help eliminate privileged information is used to determine responsive documents.

Where are you searching?

Using today's tools in civil discovery as well as criminal computer forensics investigators have a much more expansive view of the hard disks than normal users. In addition to having the ability to look at the file system as the user may have seen it, the investigator has the capability to look at recoverable deleted files, hidden files, un-partitioned disk slack, unallocated clusters and in some cases even files within the hardware protected area. Given so many views of the data, it makes sense that they may want to search the data based on any level of these views.

Some may argue that since searching every bit on the disk at the lowest sector level will undoubtedly return all the data, that this approach should always be taken. Time and Search Tuning are two reasons to consider compartmentalizing search areas. Additionally the level of information needed may differ greatly between Corporate Investigations, Civil and Criminal cases. Some areas in which users may desire to compartmentalize their searches are:

- The entire disk at the bit level including unallocated space
- All files within the file system
- Specific files within the file system
- Only deleted files

Conclusion

As you can see by just a few simple operators you can greatly reduce search complexity and find what you are looking for fast. Even when you do find one of the rare situations where you need a little more power adding one or two Boolean logic operators will most likely solve the situation. While GREP is a powerful tool and undoubtedly going to be around for some time to come, most computer forensics focused searching can be accomplished by keeping an eye on what you are looking for, where you think it is and a few simple operators.