Apache Access Log: A Detailed Introduction

June 21, 2021
by SentinelOne

What is the Apache access log? Why do you need it?

Well, a perfect use case for Apache access log can be like this: Apache access log can be used to examine a detailed log of who has been to you

include their IP address, their browser, the actual HTTP request itself, the response, and plenty more.

In this post, you'll get what's promised in the title: a detailed introduction to this type of logging. Being an introduction, the post won't venture into

knowledge of the fundamentals of the Apache access log.

We'll start out by giving you a basic definition of the Apache access log. We continue by explaining the nature of Apache, what it does, and where it

Then we'll show how to file the log file as well as how to search in it. Finally, we wrap up by covering how to customize the formatting of the log entr

Let's get started.

Apache Access Log: The Basics

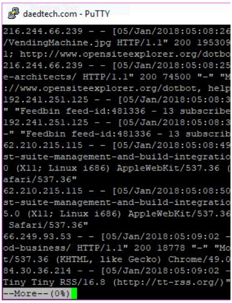
Well, at the broadest level, **the Apache access log is a source of information about who is accessing your website and how.**

But as you might expect, a lot more goes into it than just that.

After all, people visiting your website aren't like guests at your wedding, politely signing a registry to record their presence. They'll visit for a whole

And some of them will passively (or even actively) thwart information capture.

So, the Apache access log has a bit of nuance to it. And it's also a little…complicated at first glance.



```
216.244.66.239 - - [05/Jan/2018:05:08:26
/VendingMachine.jpg HTTP/1.1" 200 195309
l; http://www.opensiteexplorer.org/dotbo
216.244.66.239 - - [05/Jan/2018:05:08:25
e-architects/ HTTP/1.1" 200 74500 "-" "M
://www.opensiteexplorer.org/dotbot, help
192.241.251.125 - - [05/Jan/2018:05:08:3
" "Feedbin feed-id:481336 - 13 subscribe
192.241.251.125 - - [05/Jan/2018:05:08:3
-" "Feedbin feed-id:481336 - 13 subscrib
62.210.215.115 - - [05/Jan/2018:05:08:49
st-suite-management-and-build-integratio
0 (X11; Linux i686) AppleWebKit/537.36 (
afari/537.36"
62.210.215.115 - - [05/Jan/2018:05:08:50
st-suite-management-and-build-integratio
5.0 (X11; Linux i686) AppleWebKit/537.36
 Safari/537.36"
66.249.93.53 - - [05/Jan/2018:05:09:02 -
od-business/ HTTP/1.1" 200 18778 "-" "Mo
t/537.36 (KHTML, like Gecko) Chrome/49.0
84.30.36.214 - - [05/Jan/2018:05:09:02 -
Tiny Tiny RSS/16.8 (http://tt-rss.org/)"
--More--(0%)
```

But don't worry—demystifying it is the purpose of this post.

Apache Access Log: The What and Where

Having established the value proposition, let's get a little more specific about some things.

Apache is a widely used piece of software called a web server. The definition of server is always a little fuzzy since we use it to describe two relatec

Both exist to wait for and respond to requests, but they exist at different levels of abstraction.

> **[The Apache access log] simply has**
> **a list of all inbound requests…**

Apache is a server of the software variety. Specifically, it's a web server. This means that it lives to take in HTTP requests and give back HTTP resp

As any piece of complex software should, Apache leaves a lot of log files in its wake.

The **access log** is one of those files. It simply has a list of all inbound requests, formatted to allow you to consume them easily (and probably with a

The overwhelming majority of the time you see it in the wild, Apache runs on Linux. (You can port it to Windows, but that's not common.) And you'

This is not universal, though, because you can configure Apache to store the log wherever you want, and conventions may vary by Linux distributior

```
/var/log/httpd/access_log
/var/log/apache2/access.log
/var/log/httpd/access.log
```

You can read about locating the log file in more detail here.

Another important type of log file Apache uses is the **error log**. As its name suggests, this is where Apache records information about errors or othe

In practice, many of the so-called "errors" written to the log aren't that critical, but rather very minor incidents such 404 errors (i.e. a user requestin

Apache also records events that aren't even errors but might be of help down the line. For instance, warnings that could indicate potential problem

How to Read the Access Log

When you look at the screenshot above of an access log, it may seem intimidating.

Part of that is viewing it through the SSH shell with the text wrapping around. But there's legitimately a whole lot of information packed in there, wh

Really, though, it's just a series of log entries recorded one entry per line.

Each piece of information per entry is separated by a space. My log, in particular, is in the so-called "combined" format, which we'll discuss in more

So if you were to parse the access log, you would first tokenize it by line and then by spaces into a series of entries.

Or, to think of it another way, consider that it would be pretty straightforward to import the Apache access log to Excel and view it there.

Searching the Log

Okay, so now you understand the conceptual formatting of the access log and how to read it.

But I'm guessing you're probably not going to grab a cup of coffee, sit down, and start reading it like a newspaper.

So, how do you work with this thing? How do you interpret it?

Probably the most common way you're going to use an access log is with simple search.

Probably the most common way you're going to use the access log, especially at first, is with simple search.

In the past, I've written a guide to searching files with grep and regex, and these access logs make an excellent candidate for that.

For instance, you might do a search for all lines containing 404 to see all instances of users requesting non-existent pages.

If that's not sufficient, you can also open the logs with a text editor or with Excel. This is a little more involved and may suffer when your files are p

And finally, you might also consume these logs via a more sophisticated, broader log management strategy by folding them into a log aggregation s

Conceptually, this is somewhat like the aforementioned "put it into Excel" concept. You leverage tooling that parses the logs for you and puts them

Customizing the Access Log Format

You also have options beyond just how you consume the log file. If you want (and have the correct permissions to change settings in Apache), you

You can get an awful lot of information about requests to your site and the people making them.

Consider the "combined" format we've mentioned earlier. We'll now describe it in more detail.

The following string is what you'd use to represent the combined format in the Apache settings:

```
"%h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-agent}i""
```

Each of those represents a variable to the log formatting utility.

In your Apache configuration files, you can specify the variables you want to see in the access log.

A detailed treatment of these configuration options is beyond the scope of this post, but we'll give you a quick overview of the options we've just us

- **%h** – Remote host (client IP address)
- **%l** – User identity, or dash, if none (often not used)
- **%u** – Username, via HTTP authentication, or dash if not used
- **%t** – Timestamp of when Apache received the HTTP request
- **"%r** – The actual request itself from the client
- **%>s** – The status code Apache returns in response to the request
- **%b** – The size of the request in bytes.
- **"%{Referer}i"** – Referrer header, or dash if not used (In other words, did they click a URL on another site to come to your site)
- **"%{User-agent}i** – User agent (contains information about the requester's browser/OS/etc)

As you can see, this is powerful stuff. You can get an awful lot of information about requests to your site and the people making them.

Logging Efficiently Within Apache

When it comes to logging, it's important you have a strategy. Don't log "just because". Instead, implement a well-thought approach to logging, whic

Web server logging is certainly no exception. If you don't show your Apache's log files—especially the access and error logs—the care they need, yo harder for you to find the relevant piece of information you need in the future.

In the same vein, it's also very valuable to employ some strategy to reduce the amount of noise that makes it to the log files. That's what we'll see r

Conditional Logging

One of the best ways to keep log files manageable and relevant is to not log information that isn't valuable. That contributes to keeping the signal-t

Conditional logging is exactly what its name suggests: to log based on a set of conditions, or criteria. For instance, you might want to remove your o

This an example of how you can exclude requests from a given IP from the access log:

The following example excludes requests from a given IP, and also the requests for the robots.txt file while logging all remaining requests:

```
SetEnvIf Remote_Addr "" dontlog # Ignore requests from a given IP SetEnvIf Request_URI "^/robots.txt$" dontl
```

Graceful Restarts

Another essential component of a sound logging approach is log rotation. This is necessary when log files grow to a huge size. In order to keep then

Apache has an interesting feature that facilitates log rotation called graceful restarts. A graceful restart allows Apache to reload its configuration wi running processes to compress or move the old ones.

Piped Logs

However, it's possible to rotate logs without restarts, even graceful ones. To do that, Apache leverages the power of piping. Instead of just recordin from the Apache process itself, so it doesn't need to be stopped.

To perform the log rotation itself, you have several options. You can write your own log rotating program. Better yet, you can use an existing prograr

Understand the Opportunity Here

How you handle access to all of this information should go beyond just satisfying simple curiosity.

This information helps form the backbone of a good production strategy, and you should take advantage of it directly or indirectly.

You can mine the access log for all sorts of information. Are you seeing an unusual number of errors or improper requests? Is there a particular sit

This is just a tiny sampling of the sorts of questions you can answer.

So take a look through the Apache access log and get comfortable with it. Then, once you're comfortable with it, I suggest you develop a strategy fc

After that, you can think of taking your strategy to a new level by adopting a log management solution, such as Scalyr. A platform like Scalyr will hel

Scalyr can help you aggregate your log files from disparate sources into a centralized location. That way, you can analyze them, search them, and a

Give Scalyr a try today.

Like this article? Follow us on LinkedIn, Twitter, YouTube or Facebook to see the content we post.

Read more about Cyber Security

- SQL vs NoSQL Performance: Where One Outperforms the Other

©2023 SentinelOne, All Rights Reserved.