第6章　Apache httpdのログの設定

Webサーバは、クライアント（いわゆる「Webブラウザ」など）からのリクエストに応じて、さまざまなオブジェクト（多くの場合はHTMLや画像など）をHTTPというプロトコルで送り返すサーバです。

本章では、Webサーバの概要に触れたのち、「Apache httpd」（Version 2.4系）を例にとり、Webサーバのログ分析を行うのにあたり必要なログの設定について紹介します。ここでいう「ログ」とは基本的にアクセスログのことを指します。

6.1　Webサーバの概要

代表的なWebサーバ製品

代表的なWebサーバとしては古くは「CERN httpd」や「NCSA httpd」などが挙げられますが、現在主流なのは「Apache httpd」、「Microsoft IIS」、「Nginx」などです（ほかにもいろいろあります）。本書執筆時点でのシェアでは、やはりApache系が多いと言えます。ログのフォーマットに関しては、Apache httpd、Nginxはcombined形式、Microsoft IISはw3c形式が使われることが多いようです（設定でカスタマイズできます）。フォーマットはそれぞれ異なるものの、タイムスタンプ、リモートホスト、リクエスト内容、HTTPステータスなど、記録される／できる基本的な情報は、おおむね似ていると言えます。

Webサーバのしくみ

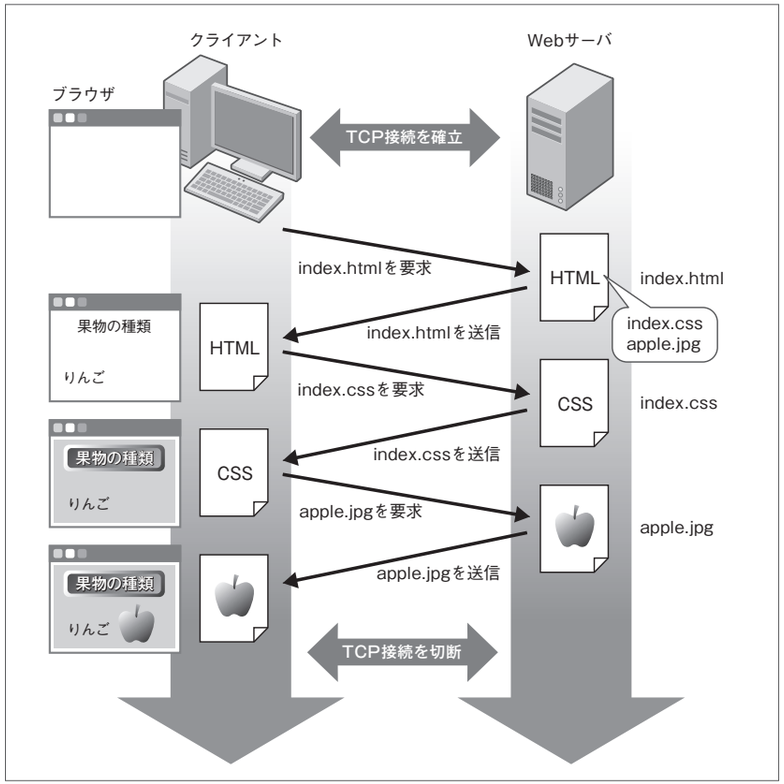
Webサーバの基本的なしくみは次のとおりです（**図6-1**）。

①クライアントはWebサーバとの間でTCPコネクションを確立したのち、HTTPでほしいオブジェクトを要求する（HTTPリクエスト）。オブジェクトとは、たとえばHTMLファイルやスタイルシート（CSS）、画像ファイルやサーバサイドで動的に生成されたコンテンツなどである

②Webサーバは要求されたリクエストに合わせたオブジェクトを、HTTPでクライアントに返す（HTTPレスポンス）

③返却されたオブジェクトの中には、さらに別のオブジェクトを必要とする場合があり、クライアントはさらにHTTPリクエストを行い、必要なオブジェクトを取得する。HTMLの中で、スタイルシート（CSS）、画像、JavaScriptなどを必要とする場合などがこれにあたる

図6-1 ▶ Webサーバとクライアント（ブラウザ）が通信する様子[注1]



6.2　設定ファイルとおもな設定項目

Apache httpdにおいてログの設定は、デフォルトではhttpd.confの中に記載されています。設定として必要なものは大きく分けて次の2つがあります。

・ログを取るためのモジュールを組み込む（ロードする）設定
・ログそのものの設定

順に説明します。

注1　HTTP keep-alive（1回のTCP接続で複数のHTTPリクエストを処理する）が有効な場合を想定した概念図の一例です（オブジェクトのリクエスト順や同時接続数についての正確な表現は意図していません）。keep-aliveが無効な場合は毎回TCP接続を切断し、都度TCP接続を確立しなおします。

ログを取るためのモジュールを組み込む（ロードする）設定

Apache httpdではログを取る機能についても、モジュールを組み込むという形態をとっています。現状、Apache httpdはDSO（Dynamic Shared Object）が有効になった状態でコンパイル（ビルド）されている場合が多いと思いますので、httpd.confの中のLoadModule log_config_moduleで始まる記述を探し、有効になっていることを確認します（**リスト6-1**）。

リスト6-1▶log_config_moduleモジュールを組み込む設定例

```
LoadModule log_config_module modules/mod_log_config.so
```

このほかにも、ログに関するモジュールはありますが、ここでは説明を割愛します。

ログそのものの設定

ログそのものの設定は、**リスト6-1**で組み込んだ「log_config_module」モジュールの設定として、httpd.confの<IfModule log_config_module> 〜 </IfModule>内で記述されることが一般的です。ただし、バーチャルホストの設定などがある場合にはこの限りではありませんので、注意してください。

この中で設定できる項目（ディレクティブと呼びます）には、ログフォーマットを設定するLogFormatディレクティブ、ログの出力ファイル名などを設定するCustomLogディレクティブなどがあります。設定例を**リスト6-2**に示します。

リスト6-2▶log_config_module設定例

```
<IfModule log_config_module>
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined  ←❶
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    CustomLog "/var/log/httpd-access.log" combined  ←❷
</IfModule>
```

ログのフォーマットとしてよく知られているものとしては、「combinedログ形式」「commonログ形式」「w3cログ形式」などがあります。**リスト6-2**内の①の行ではcombinedログ形式が、その直後の行でcommonログ形式が定義されていることがわかります。

①のcombinedログ形式を例にとって、LogFormatディレクティブの設定の見方を説明します。ダブルクォーテーションで囲まれている部分（"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""）が定義しているフォーマットで、最後にあるcombinedがこのフォーマットに名付けられたニックネームです。ニックネームにはパーセント記号（%）が含まれるべきではないので、独自のニックネームを設定する場合などには注意してください。

CustomLogディレクティブは、ログの出力先ファイル名（パス名含む）および、その際に使用するログフォーマットのニックネームを指定します。**リスト6-2**内の②の例では、/var/log/httpd-access.logに、combinedというニックネームで定義されたフォーマットで、ログを出力する設定を行っています。

LogFormatディレクティブ自体はフォーマットに対するニックネームの定義だけしか行いませんので、実際に設定したフォーマットを使ってログ出力をするためには、CustomLogディレクティブと組み合わせて設定する必要があります（TransferLogディレクティブを使う設定方法もありますが、ここでは説明を割愛します）。

以下、combinedログ形式のフォーマット部分について見ていくことにしましょう。

6.3　combinedログ形式

図6-2がhttpd.confにおけるcombinedログ形式の書式指定の例です。それぞれの意味と出力例を参考にしてください。出力例は実際には1行です。

図6-2▶httpd.confにおけるcombinedログ形式

▼書式指定の例

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

▼書式指定の意味

書式	意味
%h	リモートホスト名
%l	identdからのリモートログ名（なければ「-」）
%u	リモートユーザ名（なければ「-」）
%t	日時
\"	ダブルクォーテーション（"）
%r	リクエストの最初の行
%>s	ステータスコード
%b	ヘッダを除く送信バイト数
%{*foo*}i	HTTPリクエストヘッダfooの内容

▼ログの出力例

```
192.168.60.108 - - [22/Mar/2015:03:17:15 +0900] "GET /www/index.php HTTP/1.1"
```

リモートホスト　リモートログ名　リモートユーザ　日時　リクエストの最初の行

```
200 46359 "http://192.168.60.107/www/index.php?main_page=checkout_shipping"
```

ステータス　送信バイト数　Refererヘッダの内容

```
"Mozilla/5.0 (X11; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0"
```

User-Agentヘッダの内容

%h

リモートホスト名が記録されます。Apacheの設定ファイル（通常、httpd.conf）における設定項目のHostnameLookupsがOffになっている場合は、IPアドレスで記録されます（デフォルト）。Onの場合は、サーバがホスト名を調べて記録しますが、サーバ負荷の観点で注意が必要な場合があります。クライアント－サーバ間にプロキシサーバがある場合などは、ここにはプロキシサーバの情報が記録される場合がありますので、必ずしもエンドユーザとしてのクライアントのホスト名やIPアドレスが記録されるとは限らないことに注意してください。

%l

（もしidentdから提供されていれば）リモートログ名が記録されます[注2]。mod_identが存在し、IdentityCheckディレクティブがOnに設定されていない限り、ダッシュ“-”が記録されます。

%u

リクエストが認証されたものである場合のリモートユーザ名が記録されます。いわゆるBASIC認証やDigest認証のユーザ名です。HTTPステータスが401（unauthorized）の場合は、実際には正しくない（認証されていない）リモートユーザ名が記録される場合もあり得ます。リモートユーザ名がない場合はダッシュ“-”が記録されます。

%t

サーバがクライアントからのリクエストを受け取った日時が記録されます。フォーマットは[日/月/年:時:分:秒 タイムゾーン]となっており、年が4桁の数字、月が3文字（英語表記）、日、時、分、秒が2桁の数字、タイムゾーンはUTCからの差分（時差）を“+”もしくは“-”と4桁の数字の組み合わせになっています。たとえば、日本時間の場合はUTCからの時差は＋9時間なので、+0900となります。

\"%r\"

クライアントからのリクエスト行が記録されます。内容はメソッド、クエリストリング付のURL、およびプロトコルですが、各々がスペースで区切られているため、全体をダブルクォーテーション（\"）でくくっています。

%>s

サーバから（最終的に）クライアントに送り返される際のステータスコードです。「200 OK」や「404 Not Found」などは聞いたことがあるかもしれませんが、それらの数字が記録されます。

注2　クライアントマシン上で、IDENTプロトコル（RFC 1413）で応答するidentdや似たようなものが動いている場合に、得られた各コネクションのリモートユーザの名前がRFC 1413に準拠した形で記録されるのですが、IDENTプロトコル自体があまり使われなくなったため、出番はあまりないかもしれません。

%b

サーバからクライアントに送り返されるオブジェクトのサイズ（バイト数）が記録されます。レスポンスヘッダのサイズは含まれません。クライアントに何もコンテンツが送り返されない場合は“-”が記録されます。もし、“-”ではなく“0”と記録したい場合は、%bの代わりに%Bを使ってください。

\"%{Referer}i\"

クライアントがサーバに送信してきたリクエストの中のRefererヘッダの内容が記録されます。当該リクエストの参照元、いわゆるこのページに来る前に見ていたリンク元のURLが入っている場合が一般的です。空白文字を含む可能性があるので、ダブルクォーテーション（\"）でくくることでログのパース（構文解析）がしやすくなっています。

\"%{User-Agent}i\"

クライアントがサーバに送信してきたリクエストの中のUser-Agentヘッダの内容が記録されます。クライアントのユーザエージェント情報（ブラウザ名など）が入っている場合が一般的です。空白文字を含む可能性があるので、ダブルクォーテーション（\"）でくくることでログのパース（構文解析）がしやすくなっています。

上記のほかにも、設定できる（＝ログとして記録できる）情報はいろいろありますので、ログ形式をカスタマイズしたい場合などは次のURLを参考にしてみてください。ただし、combinedログ形式を前提としているツールなどが運用ですでに使われている場合などは、事前に十分確認し、ログ形式の変更は注意深く行ってください。

Apache モジュール mod_log_config——カスタムログ書式

・日本語　https://httpd.apache.org/docs/2.4/ja/mod/mod_log_config.html#formats
・英語　https://httpd.apache.org/docs/2.4/en/mod/mod_log_config.html#formats

6.4　分析に必要なログ項目

ログに出力すべき項目

ログ分析をするにあたり、そもそもログを取得していないというのは論外ですが、ログがあっても、そこに必要な情報が記録されていなかったり、誤った情報が記録されていたりしては、当然、ログ分析を効率良く行うことなどできません。ここでは、Apache httpdのアクセスログを例に、ログに出力すべき項目を整理します。

いつ（When）：日時

アクセスがいつ発生したのかを表す日時は非常に重要です。また、単に記録されているだけでなく、その時刻の正確さも重要です。とくに複数のサーバを運用している場合、NTPなどを用いて時刻の同期をとっておくことが大切です。時刻がずれていると、サーバのログ同士を突合させる場合に、非常に苦労することになります。

Apache httpdではログ書式で%tを指定すると日時が記録されます。よく使われるcombinedログ形式にも含まれています。タイムゾーンについても、とくに複数サーバでの運用の際には注意が必要です。

誰が（Who）：ユーザID

Webサービスにとって、アクセスしてきているのが誰なのかは、非常に重要です。実際の利用者個人を特定できることが望ましいですが、Webサービスでは非現実的なので、通常はサービスが払い出した（あるいは利用者が登録した）ユーザIDで代用します。

Apache httpdではログ書式で%uを指定するとユーザIDを記録できるのですが、これはHTTPの認証を用いた場合のみ使えます。最近では、FORMでユーザIDとパスワードをPOSTしてログイン処理を行うことがほとんどですので、この方式は使えません。mod_dumpioなどを使えばPOSTデータも記録できますが、ログが膨れ上がるので、現実的ではありません。Webサーバの裏で認証を司るアプリケーションがある場合は、そのアプリケーションがログに出力し、セッションIDや時刻などでアクセスログとひも付けるのが現実的と言えるでしょう。

どこから（Where）：ソースIPアドレス

実際に利用者がどの国や地域からアクセスしているのかを特定できることが望ましいですが、これもまた非現実的なので、代わりにクライアントのソースIPアドレスを記録します。IPアドレスがわかれば、GeoIP[注3]などの情報を用いておおよその国／地域を特定したり、whoisの情報などから利用者が使用しているISP（Internet Service Provider）を特定したりできます。

Apache httpdではログ書式で%hを指定することでリモートホストを記録できます。これもcombinedログ形式に含まれています。

また、今後はHTML5のGeolocation APIを用いて、位置情報をWebアプリケーションの側で取得／記録するようになるかもしれません。WebサーバよりもWebアプリケーションで記録するログ項目として使われ得る情報です。

何を用いて（How）：User-Agent

クライアントのUser-Agentを記録しておくことで、たとえば分析時にサーチエンジンのクローラー（ロボット）からのアクセスを除外する、といったことが可能になります。

Apache httpdではログ書式で%{User-Agent}iを指定することで、HTTPヘッダからUser-Agentヘッダの情報を記録できます。これもcombinedログ形式に含まれています。

注3　MaxMind社が提供している、IPアドレスから地理位置情報などの情報を得るためのデータベースサービス。

また、これを一歩進めて、JavaScriptなどを用いてクライアントの情報（画面解像度やフォント、OS種別やインストール済みプラグインなど）を取得／記録することで、より細かく端末を識別しようとするデバイスフィンガープリントという技術もあります。これもWebサーバよりもWebアプリケーションで記録するログ項目として使われ得る情報です。

何をして（What）：リクエストURL

利用者がWebサービス上で何を行ったかを最も的確に表すのが、リクエストURLです。URLのパスによって、どのようなファイルにアクセスしたのか、またクエリストリングによって、どのようなパラメータがWebアプリケーションに渡されたのかを把握することができます。クエリストリングとはリクエストURLの末尾に?マークに続けて*名前=値*の形式で記述した文字列のことで、Webアプリケーションなどに対してパラメータを渡す際に使用される方式の1つです。&で区切って複数のパラメータを記述でき、たとえば、`http://www.example.com/foo/bar.cgi?name1=value1&name2=value2`の場合、クエリストリングは「`name1=value1&name2=value2`」となり、&や=で分解することで、`name1`（値は`value1`）と`name2`（値は`value2`）という2つのパラメータを知ることができます。

Apache httpdではログ書式で%rを指定することでクエリストリングを含むリクエストURLを記録できます。これもcombinedログ形式に含まれています。

ログ書式にはクエリストリングを含まない%Uという指定もあるのですが、パラメータというのはWebアプリケーションの動作を決定する重要な変数であり、また、脆弱性を狙う多くの攻撃がパラメータに攻撃パターンを仕込んでいることから、通常は%r でクエリストリングも含めて記録することをお勧めします。

また、本来であればPOSTで渡されるパラメータも記録する価値のある情報なのですが、先述のとおり、mod_dumpioなどを用いないとPOSTデータを記録することは残念ながらできません。

どうなった（What）：処理結果

処理結果を端的に表すものとして、HTTPのレスポンスステータスコードがあります。「404 Not Found」の急増で、サーバへの偵察行動を察知した経験がある方もいらっしゃるのではないでしょうか。

Apache httpdではログ書式で%>sを指定することで、（内部リダイレクトされた場合は最後の）ステータスコードを記録できます。これもcombinedログ形式に含まれています。

本来であれば、処理結果としてクライアントに返した全データを記録しておくことが望ましいですが、ログが膨れあがることを考えると現実的ではありません。代わりに、応答データの異常を判断する最小限の材料として、レスポンスのサイズを記録するというのが考えられます。サイズだけでも、本来データが送られるべきところでサイズが0であったり、逆に通常ではありえない大量のデータ送信が行われていたりといった事象から、異常に気づくことができます。

Apache httpdではログ書式で%bを指定することで、ヘッダ以外の送信されたバイト数を記録できます。これもcombinedログ形式に含まれています。