Инструменты командной строки для Web

Веб-разработка*

Жизнь веб-разработчика омрачена сложностями. Особенно неприятно, когда источник этих сложностей неизвестен. То ли это проблема с отправкой запроса, то ли с ответом, то ли со сторонней библиотекой, то ли внешний API глючит? Существует куча различных прилад, способных упростить нам жизнь. Вот некоторые инструменты командной строки, которые лично я считаю бесценными.

CURL

cURL — программа для передачи данных по различным протоколам, похожая на wget. Основное отличие в том, что по умолчанию wget сохраняет в файл, а cURL выводит в командную строку. Так можно очень просто посмотреть контент веб-сайта. Например, вот как быстро получить свой текущий внешний IP:

```
$ curl ifconfig.me
93.96.141.93
```

Параметры *-i* (показывать заголовки) и *-I* (показывать только заголовки) делают cURL отличным инструментом для дебаггинга HTTP-ответов и анализа того, что конкретно сервер вам отправляет:

```
$ curl -I habrahabr.ru
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 18 Aug 2011 14:15:36 GMT
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Keep-alive: timeout=25
```

Параметр *-L* тоже полезный, он заставляет cURL автоматически следовать по редиректам. cURL поддерживает HTTP-аутентификацию, cookies, туннелирование через HTTP-прокси, ручные настройки в заголовках и многое, многое другое.

Siege

Siege — инструмент для нагрузочного тестирования. Плюс, у него есть удобная опция *-g*, которая очень похожа на *curl –iL*, но вдобавок показывает вам ещё и заголовки http-запроса.

Вот пример с google.com (некоторые заголовки удалены для краткости):

```
$ siege -g www.google.com
GET / HTTP/1.1
Host: www.google.com
User-Agent: JoeDog/1.00 [en] (X11; I; Siege 2.70)
Connection: close

HTTP/1.1 302 Found
Location: http://www.google.co.uk/
Content-Type: text/html; charset=UTF-8
Server: gws
Content-Length: 221
Connection: close

GET / HTTP/1.1
Host: www.google.co.uk
User-Agent: JoeDog/1.00 [en] (X11; I; Siege 2.70)
Connection: close

HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
X-XSS-Protection: 1; mode=block
Connection: close
```

Но для чего Siege действительно великолепно подходит, так это для нагрузочного тестирования. Как и апачевский бенчмарк *ab*, он может отправить множество параллельных запросов к сайту и посмотреть, как он справляется с трафиком. В следующем примере показано, как мы тестируем Google с помощью 20 запросов в течение 30 секунд, после чего выводится результат:

```
$ siege -c20 www.google.co.uk -b -t30s
...
Lifting the server siege...      done.
Transactions:                   1400 hits
Availability:                 100.00 %
Elapsed time:                  29.22 secs
Data transferred:              13.32 MB
Response time:                  0.41 secs
Transaction rate:              47.91 trans/sec
Throughput:                     0.46 MB/sec
Concurrency:                   19.53
Successful transactions:        1400
Failed transactions:               0
Longest transaction:            4.08
Shortest transaction:           0.08
```

Одна из самых полезных функций Siege — то, что он может работать не только с одним адресом, но и со списком URL’ов из файла. Это отлично подходит для нагрузочного тестирования, потому что можно моделировать реальный трафик на сайте, а не просто жать один и тот же URL снова и снова. Например, вот как использовать Siege, чтобы нагрузить сервер, используя адреса из вашего лога Apache:

```
$ cut -d ' ' -f7 /var/log/apache2/access.log > urls.txt
$ siege -c<concurreny rate> -b -f urls.txt
```

Ngrep

Для серьёзного анализа трафика существует Wireshark с тысячами настроек, фильтров и конфигураций. Есть также версия для командной строки *tshark*. Но для простых задач функционал Wireshark я считаю избыточным. Так что до тех пор, пока мне не нужно мощное оружие, я использую ngrep. Он позволяет делать с сетевыми пакетами то же самое, что *grep* делает с файлами.

Для веб-трафика вы почти всегда захотите использовать параметр *-W*, чтобы сохранить форматирование строк, а также параметр *-q*, который скрывает избыточную информацию о неподходящих пакетах. Вот пример команды, которая перехватывает все пакеты с командой GET или POST:

```
ngrep -q -W byline "^(GET|POST) .*"
```

Вы можете добавить дополнительный фильтр для пакетов, например, по заданному хосту, IP-адресу или порту. Вот фильтр для всего входящего и исходящего трафика на google.com, порт 80, который содержит слово “search”.

```
ngrep -q -W byline "search" host www.google.com and port 80
```