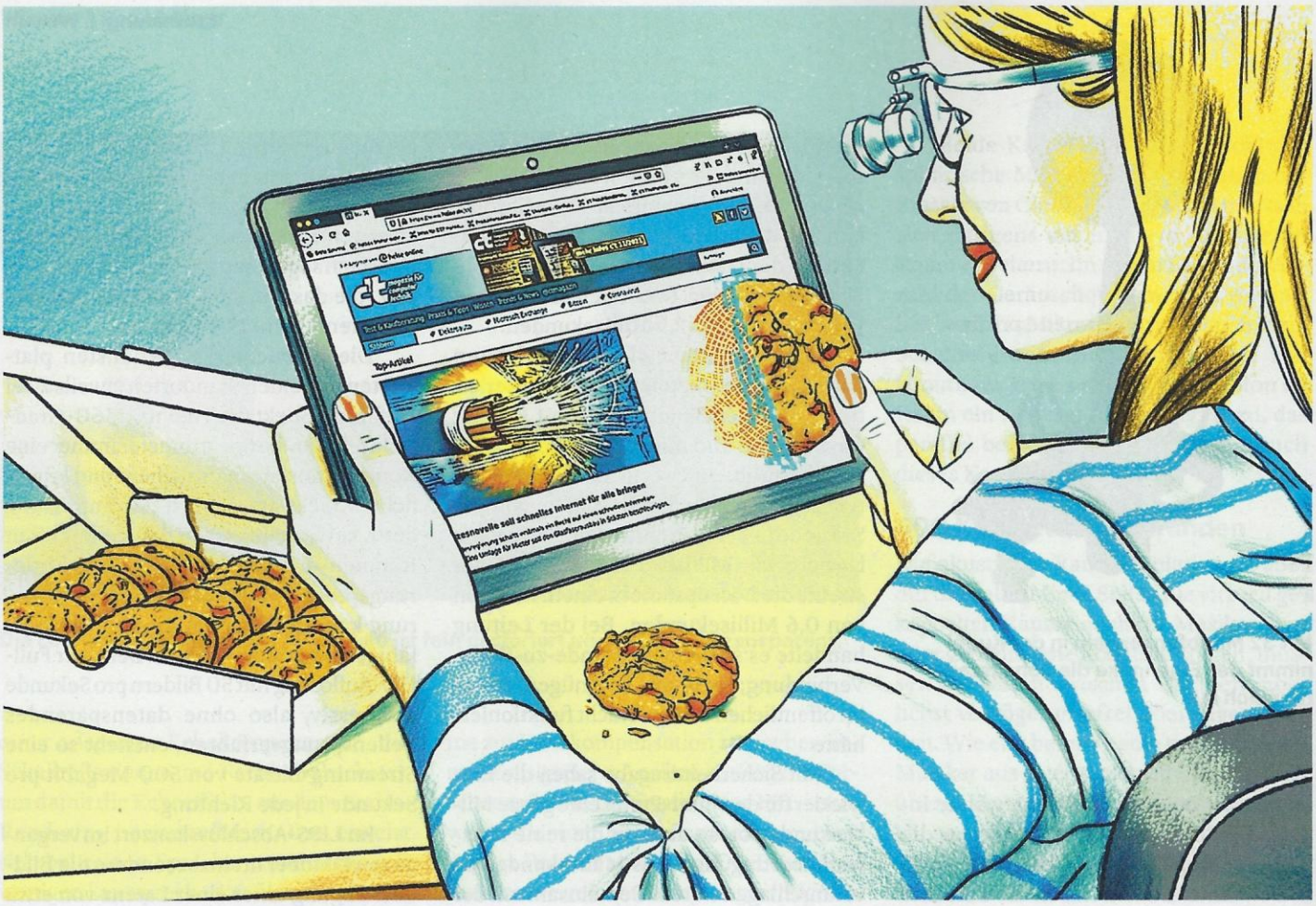


Bild: Thomas Kuhlenbeck

Angebissen

Mit Curl Webseiten anzapfen und Dateien herunterladen

Websites als Datenquellen zu nutzen ist mitunter schwierig, oft sind Logins oder Cookies Voraussetzung für den Zugriff auf die gewünschten Informationen. Mit dem Open-Source-Kommandozeilentool Curl gelingt das trotzdem.

Von Tim Schürmann

Daten aus Webseiten abzuschöpfen ist heute kompliziert, denn Website-Betreiber möchten ihre Schätze nur selten teilen und ergreifen verschiedene Maßnahmen, um ein Abfischen von Informationen zu verhindern. Genügte es früher meist, die gewünschte URL mit einem beliebigen Download-Tool abzurufen, überprüfen Webserver heute häufig den vorgeblichen Browsertyp oder erfordern Cookies und Logins. Mit Curl können Sie auch Daten oder Dateien solcher Websites herunterladen, aber auch auf viele andere Serverdienste zugreifen, denn Curl unterstützt neben HTTP und HTTPS etliche weitere Protokolle, darunter IMAP(S), SMB, (S)FTP und SCP.

c't Hardcore

Curl gibt es für Windows, macOS und Linux. Unter Linux gehört Curl oft zur Standardinstallation oder ist in den Standard-Paket-Repositories der Distribution enthalten, für macOS, Windows und viele andere Plattformen finden Sie auf der Projektseite curl.se eine Liste mit Download-Links, teilweise zu externen Anbietern. Die Parameter, und davon benötigt man mitunter etliche gleichzeitig, sind für alle Plattformen gleich.

Im einfachsten Fall rufen Sie Curl mit der herunterzuladenden URL als einzigen Parameter auf, etwa unserem Security-RSS-Feed:

```
curl "https://www.heise.de/security/↵
↳rss/news.rdf"
```

Ohne weitere Angabe gibt Curl die Daten im Terminal aus. Mit dem Parameter `-O` speichert Curl die Datei unter dem Namen, den sie auf dem Webserver trägt – in diesem Fall unter news.rdf im aktuellen Verzeichnis. Außerdem erscheint im Terminal eine Download-Statistik.

Hinter dem Parameter `-o` oder `--output` können Sie einen abweichenden Dateinamen oder Pfad angeben. Das ist vor allem dann sinnvoll, wenn die Download-URL etwa zu einer PHP-Datei führt und verschiedene

Get-Parameter enthält – diese würde Curl bei `-O` in den Dateinamen übernehmen. Damit etwaige Sonderzeichen in der URL keinen Ärger bereiten, sollten Sie sie in Anführungszeichen einschließen.

Für jeden kurzen Parameter wie `-O` oder `-o` gibt es auch einen entsprechenden Parameter in langer Schreibweise, hier `--remote-name` und `--output`. Für andere Parameter existiert nur die Langform. Gibt es beide Formen, so können Sie sich aussuchen, welche Sie verwenden.

Wenn Sie eine größere Datei wie zum Beispiel das ISO-Image von Ubuntu 20.04 herunterladen und der Download plötzlich abbricht, kann Curl zu einem späteren Zeitpunkt den Download fortsetzen. Dazu rufen Sie das Tool nach dem Abbruch mit dem Parameter `-C -` auf. Achten Sie dabei auf die Großschreibung und die Minuszeichen:

```
curl -C - "https://releases.ubuntu
↳.com/20.04.2.0/ubuntu-20.04.2.0-
↳desktop-amd64.iso"
```

Bei HTTPS-Verbindungen achtet Curl unter anderem darauf, dass das Serverzertifikat zur Domain passt und nicht abgelaufen ist. Stimmt etwas nicht, bricht Curl ab. Um nähere Hinweise auf die Ursache zu bekommen, können Sie Curl mit dem Parameter `-v` geschwätzig machen; der Parameter `-k` ignoriert alle Zertifikatfehler und versucht mit allen Mitteln, die gewünschten Daten herunterzuladen.

Dabei berücksichtigt Curl stets nur die explizit angegebenen Adressen: Geben Sie etwa eine HTML- oder PHP-Seite als URL an, so lädt Curl nur den HTML-Code dieser einen Seite herunter – und zwar ohne den Inhalt zu verändern. Etwaige dort referenzierte Bilder, CSS- und JavaScript-Dateien ruft Curl nicht ab, das Programm eignet sich also nicht dafür, eine Website lauffähig auf dem lokalen Rechner zu spiegeln. Dafür ist das Programm `wget` [1] besser geeignet.

Rangeschafft

Curl interpretiert die vom Server gelieferten Daten nicht, weshalb das Programm gern in Skripten benutzt wird, um die Rohdaten zu beschaffen [2] oder um von Hand im Terminal zu schauen, welche Daten ein Server tatsächlich liefert. Das muss nicht immer HTML-Code sein, der Dienst Wttr.in zum Beispiel liefert die aktuelle Wettervorhersage als hübsche ASCII-Grafik:

```
curl -H "Accept-Language: de" \
  "http://wttr.in/Hannover"
```

Der Parameter `-H` fügt in den HTTP-Header den Parameter `Accept-Language` ein und wählt so die gewünschte Sprache aus. Möchte man mehrere Parameter setzen, darf `-H` auch mehrfach benutzt werden.

Dass Curl die Daten unverändert ausgibt, nutzen manche Softwareprojekte, um mit einer einzigen Befehlszeile ein Installationsskript herunterzuladen und auszuführen, etwa für das CMS Lektor:

```
curl -sf "https://www.getlektor.com/
↳installer.py" | sudo python3
```

Curl ruft das Installationsskript `installer.py` ab und reicht es mittels Pipe an den Python-Interpreter weiter, der es ausführt. Der Curl-Aufruf nutzt aus, dass Sie mehrere Parameter zu einem kombinieren dürfen; `-sf` bedeutet also das Gleiche wie `-s` („silent", keine Download-Statistik ausgeben) und `-f` („fail silently", keine Fehlermeldungen ausgeben). Diese Praxis ist allerdings ziemlich riskant, ein Angreifer, der die Projektseite übernommen hat, könnte Ihnen so beliebige Befehle unterschieben – die dann mit Root-Rechten ausgeführt werden. Besser ist es, das Skript herunterzuladen, einen Blick hineinzuwerfen und es erst dann auszuführen:

```
curl -O "https://www.getlektor.com/
↳installer.py"
less installer.py
sudo python3 installer.py
```

Hinter einigen Internetadressen versteckt sich eine Weiterleitung. Ausprobieren können Sie das mit `curl "http://heise.de"`, als Rückgabe erhalten Sie anstelle der zu erwarteten Startseite nur einen Verweis auf `https://www.heise.de`. Mit `curl -L "http://heise.de"` folgt Curl der entsprechenden Weiterleitung und liefert die eigentliche Startseite von Heise zurück. Das klappt allerdings nur, wenn der Server die Weiterleitung über den korrekten HTTP-Status-Code meldet.

Grenzen einreißen

Curl unterstützt neben dem direkten Abruf der URL auch mehrere Formen von Proxies. Um Geofencing zu umgehen oder zu überprüfen, ob ein Webserver für verschiedene Länder unterschiedliche Inhalte liefert, können Sie Curl eine Verbindung über einen HTTP-Proxy etwa von freeproxylists.net herstellen lassen. Dazu verwenden Sie den Parameter `-x` gefolgt von IP-Adresse und Portnummer des Proxys:

```
curl -x 176.9.85.13:3128 \
  "https://www.showmyip.gr"
```

Das funktioniert in gleicher Weise mit lokalen Proxies, etwa in Unternehmen. Sie müssen lediglich die IP-Adresse anpassen. Ein weiterer Anwendungsfall für einen Proxy ist, wenn Sie Onion-URLs aus dem Darknet abrufen oder Ihre Identität schützen wollen. Dazu kann Curl den SOCKS5-Proxy benutzen, den der Tor-Browser automatisch bereitstellt, sobald er geöffnet wird. Mit folgendem Befehl kontaktiert Curl Facebooks Auftritt im Darknet:

```
curl -i --socks5-hostname \
  127.0.0.1:9150 \
  "http://facebookcorewwwi.onion"
```

Durch den Parameter `--socks5-hostname` benutzt Curl nicht nur das SOCKS5-Protokoll, sondern überlässt auch die Namensauflösung dem Tor-Browser. Das schützt nicht nur Ihre Privatsphäre, es ist bei Onion-Domains zwingend notwendig, da herkömmliche DNS-Server keine Onion-Domains kennen.

Der Parameter `-i` macht Curl gesprächiger, es zeigt dann den HTTP-Header der Serverantwort. Facebooks Hidden Service antwortet lediglich mit dem HTTP-Status-Code 301 („Moved Permanently") und der neuen URL – eine HTML-Umleitungsseite wie der Heise-Server liefert Facebook nicht, weshalb Curl ohne den Parameter `-i` gar nichts ausgeben würde.

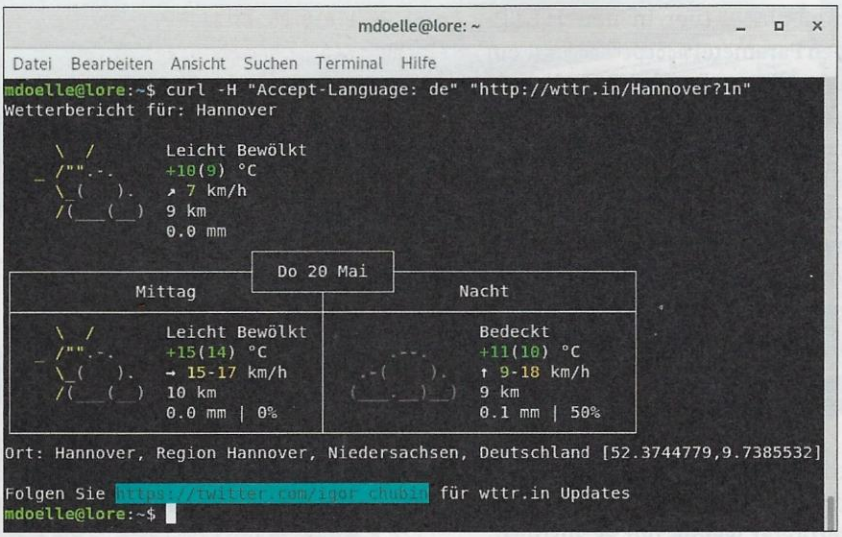
Curl maskiert

Bei manchen Webseiten macht es einen Unterschied, ob man sie im Browser öffnet oder mit Curl abruft. Etwa bei der Google-Suche, wenn Sie mit folgendem Curl-Aufruf nach dem Stichwort „Linux" suchen:

```
curl "https://www.google.de/
↳search?q=Linux"
```

Während Sie im Browser die Suchergebnisse sehen, verweigert Google Curl den Zugriff und liefert stattdessen nur den HTTP-Status-Code 403 („Forbidden"). Der Grund dafür ist, dass Google Curl am sogenannten User-Agent erkennt, mit dem sich der Browser beim Webserver zu erkennen gibt. Mit dem Parameter `-v` können Sie den gesamten Verbindungsaufbau verfolgen und finden dort unter anderem die HTTP-Anfrage von Curl, hier ein Auszug:

```
> GET /search?q=Linux HTTP/2
> Host: www.google.de
```



Der Wetterdienst Wttr.in liefert das aktuelle Wetter als farbige ASCII-Grafik, die Curl unverändert auf der Konsole ausgibt.

```
> User-Agent: curl/7.64.0
> Accept: */*
```

Curl weist sich anständig aus, was zur Ablehnung durch Google führt. Mit dem Parameter `-A` können Sie Curl anweisen, sich zum Beispiel als Firefox Version 86.0 unter Linux zu tarnen:

```
curl -A "Mozilla/5.0 (X11; Ubuntu; ↵
↳Linux x86_64; rv:86.0) Gecko/↵
↳20100101 Firefox/86.0" \
"https://www.google.de/search?q=Linux"
```

Links zu zwei Sammlung von User-Agent-Angaben diverser Browser und Mobilgeräte haben wir Ihnen auf ct.de/ycqs bereitgestellt.

Lesehilfe

Die Angabe des User-Agent macht Curl-Befehlszeilen sehr lang und nur noch schwer durchschaubar. Solche häufig benötigten Angaben können Sie in einer oder – etwa für unterschiedliche User-Agents – mehreren Konfigurationsdateien speichern und Curl diese dann über den Parameter `-K` einlesen lassen. Um mehrere Konfigurationsdateien zu kombinieren, geben Sie `-K` mehrfach an.

Eine Konfigurationsdatei darf beliebige Curl-Parameter in der gleichen Form enthalten, wie man sie auch im Terminal eingeben würde – mit einer Ausnahme: Während Sie unter macOS und Linux auf der Kommandozeile Zeichenketten in Hochkommas einschließen dürfen, müssen Sie in Konfigurationsdateien Anführungszeichen verwenden, um etwaige Leerzeichen wie im User-Agent zu schützen. Eine Konfigurationsdatei `cFirefox` mit Firefox als User-Agent sieht deshalb folgendermaßen aus:

```
-A "Mozilla/5.0 (X11; Ubuntu; Linux↵
↳ x86_64; rv:86.0) Gecko/20100101↵
↳ Firefox/86.0"
```

Damit wird der Aufruf der Google-Suche sehr viel übersichtlicher:

```
curl -G -d "q=Linux" -d "hl=en" \
-K cFirefox \
"https://www.google.de/search"
```

Wenn Sie Curl keine Konfigurationsdatei übergeben, sucht es beim Aufruf automatisch nach der Datei `.curlrc` in Ihrem Heimatverzeichnis. In dieser Konfigurationsdatei können Sie Parameter verstauen, die Curl bei jedem Aufruf ausführen soll.

Formularkönig

Die Google-URLs aus den beiden letzten Beispielen enthielten eine implizite HTTP Get Request mit dem Suchbegriff, durch das Fragezeichen von der eigentlichen URL der Google-Suche abgetrennt. Damit wird eine entsprechende Eingabe im Suchformular auf Googles Einstiegsseite simuliert. Curl beherrscht den Umgang mit Formularen aber auch nativ, Sie müssen die Get Request nicht von Hand in die URL integrieren, zum Beispiel für zwei Werte:

```
curl -G -d "q=Linux" -d "hl=en"
-A "Mozilla/5.0 (X11; Ubuntu; ↵
↳Linux x86_64; rv:86.0) Gecko/↵
↳20100101 Firefox/86.0" \
"https://www.google.de/search"
```

Der Parameter `-G` veranlasst Curl, aus den hinter den Parametern `-d` angegebenen Werten – einmal der Suchbegriff „Linux“ und einmal die Ausgabesprache Englisch – einen HTTP-GET-Request aufzubauen. Ohne den Parameter würde Curl sonst einen HTTP-POST-Request senden. Sie können `-d` mehrfach verwenden, Curl kümmert sich automatisch um die korrekte Übermittlung aller Daten. Wenn die Daten Sonderzeichen enthalten, sollten Sie anstelle von `-d` die Option `--data-urlencode` für GET-Requests verwenden, damit Curl die Daten so umwandelt, wie das auch ein Browser tun würde.

Der größte Vorteil des Parameters `-d` liegt darin, dass Curl die Daten automatisch passend für das verwendete Protokoll konvertiert. So können Sie etwa im Smart Home via MQTT eine Shelly-Plug-Steckdose einschalten:

```
curl -d "turn=on" \
"mqtt://192.168.2.61/relay/0"
```

Ob Curl das MQTT-Protokoll unterstützt, hängt aber davon ab, ob das Programm und die Bibliothek libcurl entsprechend übersetzt wurden – Standard ist das nicht. Auf den meisten Rechnern wird Curl deshalb den Aufruf mit einer entsprechenden Fehlermeldung verweigern. Dann lässt sich der Schaltvorgang aber oft auch per HTTP mit der gleichen URL durchführen.

Sie haben Post

Curl ist ein regelrechtes Sprachtalent, Sie können damit sogar Ihre E-Mails per POP3 oder IMAP einsehen. Das folgende Beispiel schaut nach, ob es in der Inbox Ihres IMAP-Postfachs bei Web.de eine neue Mail gibt:

```
curl "imaps://imap.web.de:993" \
  --user "benutzer:passwort" \
  -X "EXAMINE INBOX"
```

Den Benutzernamen und das Passwort müssen Sie gegen Ihre Zugangsdaten austauschen – und falls Sie einen anderen Mailanbieter verwenden, auch die Adresse des IMAP-Servers. Die Zugangsdaten über die Kommandozeile mitzugeben ist aber keine gute Idee, denn so sind sie in der Prozessliste für andere Benutzer sichtbar.

Sie sollten deshalb auch für die Zugangsdaten eine Curl-Konfigurationsdatei anlegen, etwa mit dem Namen cImap-Cred. Dort tragen Sie den Parameter

`--user` nebst Zugangsdaten ein und machen die Datei per `chmod 600 cImapCred` nur für Sie selbst lesbar. Anschließend können Sie in Curl-Befehlszeilen `-K cImapCred` anstelle der Login-Daten benutzen.

Gibt es ungelesene Mails („UNSEEN“) auf dem IMAP-Server, erwähnt er dies in seiner Antwort auf den IMAP-Befehl `EXAMINE INBOX` und liefert die Nummer der ersten ungelesenen Mail zurück, hier ein Auszug:

```
* 8064 EXISTS
* 0 RECENT
* OK [UNSEEN 8062] First unseen.
```

Die Mail Nummer 8062 ist die erste ungelesene, insgesamt gibt es 8064 Mails in der Inbox. Diese Informationen könnten Sie etwa auf einem Raspberry Pi in einem Skript nutzen, um bei neuen Mails eine am GPIO-Port angeschlossene LED einzuschalten – oder um anschließend die Mail Nummer 8062 abzurufen:

```
curl "imaps://imap.web.de:993/INBOX;↵
↳MAILINDEX=8062" -K cImapCred
```

Unter ct.de/ycqs finden Sie einen Link zum RFC 3501, in dem die Befehle des IMAP-Protokolls beschrieben sind, größtenteils mit Beispielen. Sie können mit Curl aber nicht nur Mails abrufen, sondern über das SMTP-Protokoll, das Curl ebenfalls beherrscht, auch welche versenden:

```
curl "smtps://smtp.web.de:587" \
  -K cImapCred \
  --mail-from "benutzer@web.de" \
  --mail-rcpt "ct@ct.de" \
  --upload-file mail.txt
```

Die Parameter für SMTP sind selbsterklärend, den Inhalt der E-Mail speichern Sie vorab in der Datei mail.txt. Sie könnten aber genauso gut täglich eine Log-Datei per Cron-Job verschicken lassen.

Kekse für Daten

Eine besonders interessante Funktion von Curl ist, Cookies über mehrere Aufrufe hinweg zu speichern und wiederzuverwenden: Viele Websites verwenden Cookies, um sicherzustellen, dass sich ein Benutzer zuvor eingeloggt hat, etwa auf einer Wordpress-Seite. Solange der beim Login gespeicherte Cookie immer wieder präsentiert wird, ist der Zugriff etwa auf interne Beiträge gestattet.

Um falls erforderlich eine neue Session anzulegen und das Cookie für das anschließende Login zu speichern, sollten Sie zuerst die Login-Seite des Wordpress-CMS abrufen:

```
curl -K cFirefox -c cCookies \
  "https://example.com/login/"
```

Der Parameter `-c` gefolgt von dem Dateinamen cCookies bedeutet, dass Curl sämtliche Cookies in dieser Datei ablegt. Die meisten Wordpress-Installationen verwenden nur leicht modifizierte Standard-Templates, sodass das Wordpress-Login fast überall identisch ist: Der Benutzername gehört in das Formularfeld `log` und das Passwort in das Feld `pwd`. Diese Daten erwartet Wordpress jedoch üblicherweise im JSON-Format, hier ein Beispiel:

```
curl -K cFirefox -c cCookies \
-b cCookies -d "{{\"log\":↵
↳\"benutzername\",\"pwd\":↵
↳\"passwort\"}}" \
"https://example.com/wp-login.php"
```

Mit dem Parameter `-b` gefolgt vom Dateinamen präsentiert Curl dem Webserver die bereits dort gespeicherten Cookies und speichert, dank `-c`, weitere Cookies in derselben Datei.

Indem Sie die Cookie-Parameter `-c` und `-b` auch bei allen Folgeaufrufen verwenden, können Sie mittels Curl das gesamte Wordpress-CMS nach nützlichen Daten durchstöbern. Allerdings sind Curl Grenzen gesetzt, etwa wenn Webseiten umfangreiche JavaScript-Funktionen oder gar Captchas benutzen, um Data-Mining durch Skripte und Bots gezielt zu unterbinden. Dann bleibt Ihnen nur, die Daten im Browser von Hand einzusammeln.

(mid@ct.de) **ct**

Literatur

[1] Tim Schürmann, Kleiner Datensauger, Dateien mit wget herunterladen, c't 8/2017, S. 168

[2] Tim Schürmann, Ausbaufähig, Conky mit eigenen Lua-Skripten erweitern, c't 7/2019, S. 158

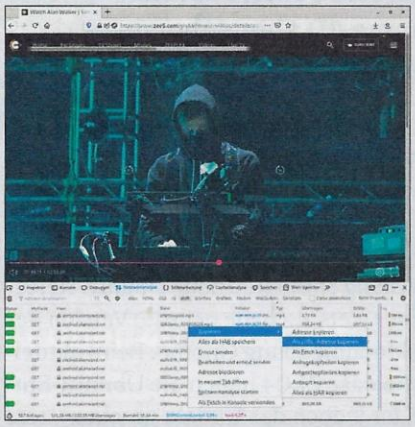
User-Agents: ct.de/ycqs

Schützenhilfe vom Browser

In den Entwicklerwerkzeugen von Firefox und Chrome können Sie für alle Elemente einer Internetseite passende Curl-Befehle erzeugen, mit denen Sie sie direkt auf der Kommandozeile herunterladen können. Sie müssen den Befehl also nicht mühsam von Hand zusammenbauen. Diese praktische Funktion ist allerdings bei bei den Browsern gut versteckt.

Im Firefox öffnen Sie zunächst mit Strg+Umschalt+E die Netzwerkanalyse der Entwicklerwerkzeuge. Sobald Sie eine Webseite aufrufen, protokolliert der Browser alle Seitenaufrufe am unteren Bildschirmrand. Indem Sie mit der rechten Maustaste auf einen Dateinamen klicken und dann aus dem Kontextmenü „Kopieren/Als cURL-Adresse kopieren“ auswählen, erhalten Sie eine komplette Curl-Befehlszeile zum Abruf eben dieser Datei. Das ist zum Beispiel auf exotischen Videoplattformen praktisch, die von youtube-dl nicht unterstützt werden.

In Chrome ist die Vorgehensweise ähnlich; darin öffnen Sie mit Strg+Umschalt+I die Entwicklertools und wechseln ins Register „Network“, bevor Sie die URL eingeben und abrufen. Klicken Sie dann in der Liste der abgerufenen Dateien mit der rechten Maustaste auf den gewünschten Eintrag und wählen Sie aus dem Kontextmenü „Copy/Copy as cURL“, um die Curl-Befehlszeile zu erhalten.



Die Curl-Befehlszeile aus Firefox erleichtert es drastisch, etwa einzelne Video- und Audiodateien von exotischen Videoplattformen herunterzuladen.