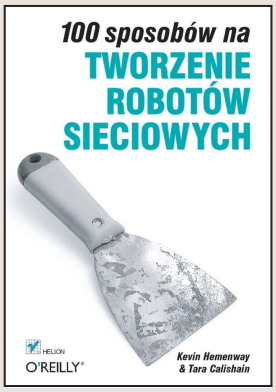
100 sposobów na tworzenie robotów sieciowych



Autorzy: Kevin Hemenway, Tara Calishain
Tłumaczenie: Tomasz Żmijewski
ISBN: 83-7361-452-4
Tytuł oryginału: Spidering Hacks
100 Industrial-Strength Tips& Tools
Format: B5, stron: 384

Internet to niewyczerpane źródło informacji. Zwykle jednak znalezienie w gąszczu witryn i danych tego, co jest naprawdę potrzebne, jest zadaniem niezwykle trudnym. Wyszukiwarki internetowe, nawet te najpopularniejsze i najbardziej efektywne w działaniu, nie zawsze są odpowiednim rozwiązaniem. W takich przypadkach trzeba poszukać innego sposobu, a nawet opracować własną metodę przeszukiwania sieci.

Książka „100 sposobów na tworzenie robotów sieciowych" może służyć pomocą w wielu takich właśnie przypadkach. Przedstawia sposoby tworzenia i wykorzystywania „pająków" przeszukujących sieć pod kątem określonych zasobów. W książce poruszono następujące zagadnienia:

- Uwarunkowania prawne dotyczące korzystania z programów przeszukujących sieć
- Narzędzia do tworzenia robotów oraz wyszukiwania danych
- Sposoby wyszukiwania i pobierania plików multimedialnych
- Przeszukiwanie sieciowych baz danych
- Metody zapisywania zbiorów danych i aktualizowania ich
- Tworzenie zbiorów danych łatwych do przeszukiwania przez innych

W książce przedstawiono techniki tworzenia kompletnych programów przeszukujących sieć (pająków), umożliwiających odnalezienie wszystkich potrzebnych danych.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

ROZDZIAŁ PIERWSZY

Łagodne wprowadzenie

Sposoby 1. – 7.

W sieci są obecnie ponad trzy miliardy stron, więc każdy żeglarz cyberprzestrzeni wcześniej czy później musi zadać sobie dwa pytania: gdzie znaleźć coś wartościowego i co można z tym zrobić? Każdy ma własne pojęcie o tym, co jest wartościowe, większość ludzi ma też różne dobre pomysły o tym, jak takie rzeczy wykorzystać. Są takie zakątki sieci, w których przekształcanie danych do innych celów prowadzi do nagłych olśnień, a to z kolei staje się początkiem nagłej eksplozji nowych danych.

Z punktu widzenia sieci dopiero niedawno pojawiły się pierwsze narzędzia ułatwiające korzystanie z powszechnie dostępnych danych. Kiedy serwis Google udostępnił swoje serwisy wyszukiwawcze za pośrednictwem interfejsów API (zajrzyj do książki *Google Hacks*), podobny krok szybko zrobił Amazon.com (zajrzyj do *Amazon Hacks*); w jednym i drugim wypadku owocem tych decyzji było pojawienie się nowych narzędzi. W tym krótkim i przyjemnym rozdziale przedstawimy Czytelnikom elegancką sztukę zbierania danych i używania pająków: czym ona jest, a czym nie jest, co jest zwykle dozwolone, a co niesie ze sobą ryzyko. Powiemy, jak szukać alternatywnych metod dostępu do interesujących danych i jak pozyskać sobie przychylność autorów witryn (a tak naprawdę to przekazać im pewną wiedzę), którzy na naszą automatyzację patrzą niechętnie.

SPOSÓB 1.

Pająki i zbieranie danych: kurs na zderzenie

Odpowiemy tu na pytania: dlaczego i w jakim celu instaluje się pająki i zbiera dane.

Istnieje mnóstwo rozmaitych programów służących do zbierania i odsiewania informacji, łączenia ze sobą zasobów oraz porównywania danych — liczba tych programów stale rośnie. Ludzie to tylko część znacznie większego i zautomatyzowanego równania. Jednak mimo takiej obfitości istniejących programów, podstawowe zasady ich działania pozostają niezmienne.

Pająki to programy wędrujące po sieci, służące do zbierania informacji. W dziennikach witryn sieciowych można znaleźć programy `User-Agent`, takie jak `Googlebot`, `Scooter` czy `MSNbot`. Są to właśnie pająki lub, jak wolą je nazywać z angielska niektórzy, *boty*.

W książce tej będziemy stale mówić o pająkach i programach zbierających — czym różnią się jedne od drugich? Ogólnie rzecz biorąc, jedne i drugie wędrują po internecie i zbierają różne dane. Na potrzeby tej książki najlepiej traktować *pająki* jako programy pobierające

całe strony, pliki lub ich zbiory, zaś *programy zbierające* jako programy służące do pobierania wybranych fragmentów danych z całych plików. Przykładowo, **jeden z pająków** [Sposób 44.], omawianych w tej książce, przeznaczony jest do pobierania całych pakietów (kolekcji) wiadomości z Yahoo! Group, a następnie do przekształcania ich w pliki poczty elektronicznej, tak aby można było odczytywać je w programie pocztowym. Z kolei **jeden z programów zbierających** [Sposób 76.] służy do pobierania ze stron danych o rozkładzie jazdy pociągów. Pająki poruszają się po hiperłączach, zbierając dane, natomiast programy zbierające wybierają ze stron informacje. Jedne i drugie są zwykle używane łącznie: można używać programu wykorzystującego pająka do poruszania się po łączach, by potem za pomocą programu zbierającego wybierać jedynie interesujące dane.

Czemu pająki?

Poznając jakąkolwiek technologię czy sposób użycia technologii, zawsze dobrze jest zadać sobie ważne pytanie: dlaczego? Dlaczego trudzić się pająkami? Po co marnować czas na pisanie pająka, testowanie go, uzyskiwanie zezwolenia od właściciela strony na jego użycie, udostępnianie go innym i opiekę nad jego kodem? Otóż warto: kiedy raz zacznie się stosować pająki, potem trudno wyobrazić sobie, jak można było się dotąd bez nich obejść.

Uzyskiwanie automatycznego dostępu do zasobów
: Oczywiście, można codziennie odwiedzać każdą interesującą nas stronę, ale czy nie prościej byłoby napisać odpowiedni program, który wybrałby jedynie interesujące nas dane? Jeśli pająk zwróci wyniki najczęściej wykonywanego poszukiwania w Google, można zaoszczędzić mnóstwo czasu, energii i uniknąć wielokrotnego powtarzania tej samej pracy. Im bardziej prace będą zautomatyzowane, tym więcej czasu można będzie poświęcić używaniu tych danych.

Zbieranie informacji i prezentowanie ich w formacie alternatywnym
: Wyniki badań marketingowych, zbierane w formie wyników, zwracane przez wyszukiwarkę, można **zaimportować do arkusza Microsoft Excel w celu późniejszej ich prezentacji czy analizowania** [Sposób 93.] Można skopiować swoje ulubione archiwa Yahoo! Groups w takiej postaci, którą **program pocztowy** będzie w stanie **odczytać,** tak jak odczytuje **każdy inny folder pocztowy** [Sposób 43.] Można na bieżąco śledzić zawartość najciekawszych witryn **bez konieczności płacenia za kolejne wizyty** [Sposób 81.] Kiedy już mamy w ręku interesujące nas dane, można je przekształcać, zmieniać ich układ, formatować wedle woli.

Agregowanie osobnych źródeł danych
: Żadna witryna nie jest samotną wyspą, choć nie jest to tak oczywiste, jeśli wziąć pod uwagę trudność ręcznego integrowania danych z różnych witryn. Użycie pająków automatyzuje to żmudne zadanie — na przykład pomaga analizować **zmieniające się stopniowo wyniki Google** [Sposób 93.] czy **łączyć uzupełniające się dane** [Sposób 69.] z różnych dzienników w pojedynczy pakiet RSS. Pająki można przystosować do zbierania danych zarówno z różnych źródeł, jak i z jednego źródła w różnych okresach.

Łączenie możliwości różnych witryn
Wiele osób ma swoje ulubione wyszukiwarki, które jednak nie posiadają wszystkich potrzebnych funkcji. Braki te może uzupełnić inna wyszukiwarka, która z kolei może nie spełniać innych wymagań. Pająk może **powiązać możliwości obu takich programów** [Sposób 48.] przez zadanie zapytania jednej przeglądarce i przekazanie uzyskanych odpowiedzi drugiej.

Znajdowanie i zbieranie informacji określonego typu
Być może najpierw trzeba będzie przeszukać wyniki wyszukiwania; w naszym imieniu pająk może wykonywać zapytanie przez **wypełnianie formularzy i odsiewanie potrzebnych wyników** [Sposób 51.]

Wykonywanie typowych zadań administratora witryny
Codziennymi, rutynowymi zadaniami administratora może zająć się pająk. To on może służyć do sprawdzania kodu HTML, aby zapewnić jego zgodność ze standardami, do sprawdzania czystości kodu (*http://tidy.sourceforge.net/*), poprawności hiperłączy, a także braku łączy do zakazanych treści.

Więcej informacji o pająkach, robotach, pełzaczach i programach zbierających dane znaleźć można w witrynie Web Robot FAQ (często zadawane pytania na temat sieciowych robotów) dostępnej pod adresem *http://www.robotstxt.org/wc/faq.html*.

SPOSÓB 2.

Zasady obowiązujące pająka i jego użytkownika

Wybrane zasady dotyczące pisania własnych dobrze ułożonych pająków.

Aby tworzony pająk był skuteczny i tak przydatny, jak tylko to możliwe, trzeba podczas jego tworzenia pamiętać o kilku zasadniczych kwestiach.

Nie bądźmy zbyt wybredni

Użycie pająka oznacza pobieranie informacji z witryny. Aby pobrać informacje, trzeba przebić się przez mieszaninę różnych znaczników: HTML, XML, zwykły tekst lub jeszcze inne formaty danych. Delikatnie mówiąc, nie jest to nauka ścisła. Jeśli zmieni się choć jeden znacznik czy zmieni się cokolwiek w pliku formatującym, pająk prawdopodobnie przestanie działać i będzie wymagał zrobienia w nim poprawek. Na szczęście większość witryn nie jest gruntownie przebudowywana co sześć miesięcy, jak to kiedyś bywało, ale i tak witryny zmieniają się na tyle często, aby trzeba było to brać pod uwagę.

Aby zredukować wrażliwość naszych narzędzi na wszelkie zmiany, podczas zbierania danych ze strony należy używać możliwie mało danych brzegowych. *Dane brzegowe* to otoczka interesujących nas treści: znaczniki, nadmiarowe informacje, spacje, znaki nowego wiersza i tak dalej. Na przykład tytuł każdej witryny wygląda mniej więcej tak:

```
<title>To jest tytuł</title>
```

Jeśli właśnie tytuł nas interesuje, to danymi brzegowymi są znaczniki `<title>` i `</title>`.

Regularnie należy sprawdzać wyniki uzyskiwane od pająka, aby **upewnić się, że są one zgodne z oczekiwaniami** [Sposób 31.]. Możliwie szybko należy uwzględniać wszelkie zmiany, aby się nie pogubić, poza tym, projektując pająka, **należy uczynić go możliwie elastycznym** [Sposób 32.]

Nie ograniczajmy swoich zbiorów danych

To, że pracujemy ze stronami sieciowymi, nie oznacza, że musimy się ograniczać do analizowania dokumentów HTML. Ograniczenie się jedynie do stron sieciowych oznacza od razu ograniczenie dostępnego zbioru danych; w sieci są przecież także obrazy, dźwięki, filmy, pliki PDF, pliki tekstowe — wszystkie one warte są tego, aby rozważyć dołączenie ich do swojej kolekcji.

Nie wyważajmy otwartych drzwi

Chętnie myślimy, że to, co robimy, jest jedyne w swoim rodzaju i niepowtarzalne, ale niewykluczone, że wcześniej ktoś już tworzył podobne pająki i pobierał dane z tych samych lub podobnych witryn, zostawiając swoje ślady w postaci kodu, czystych danych czy instrukcji.

CPAN (*http://www.cpan.org*) — Comprehensive Perl Archive Network (ang. *Największa sieć z archiwalnymi zasobami Perla*) — to prawdziwa skarbnica modułów Perla pozwalających programować w internecie, przeglądać tekst w poszukiwaniu danych, przekształcać zebrane zbiory danych — wszystko to może być wykorzystane przy tworzeniu własnych pająków. Moduły te są dostępne za darmo: wystarczy je pobrać, a następnie używać, modyfikować i udoskonalać. Kto wie, być może, kończąc tworzenie naszego pająka, będziemy mieli w swoim dorobku kilka nowych modułów, które będziemy mogli udostępnić komuś innemu.

Przed rozpoczęciem kodowania należy przejrzeć podaną witrynę, aby się upewnić, że nie poświęcamy mnóstwa czasu i wysiłku w stworzenie czegoś, co już jest gotowe. Jeśli co rano chcemy mieć w skrzynce pocztowej prognozę pogody, warto sprawdzić witrynę lokalnej gazety czy jakąś witrynę poświęconą pogodzie (jak *http://www.weather.com*); prawdopodobnie jest tam potrzebna usługa. Jeśli potrzebna jest zawartość witryny w formie pakietu RSS, natomiast w witrynie tej nie widać pomarańczowego przycisku „XML", warto tej zawartości poszukać w Google (`rss site:example.com ( filetype:rss | filetype:xml | filetype:rdf )`) lub w witrynie Syndic8 (*http://www.syndic8.com*).

Następnie można się oczywiście skontaktować z właścicielem witryny, pytając, czy dostępna jest dana usługa czy format danych. Być może to właśnie pytanie przekona właściciela, że posiadanie pakietu RSS lub sieciowej usługi API z treścią witryny jest dobrym pomysłem.

W podrozdziale **„Co dalej?"** [Sposób 100.] podana została większa ilość adresów miejsc poświęconych pobieraniu zasobów.

Wskazówki dla użytkownika

Tak jak ważne jest przestrzeganie pewnych zasad podczas programowania pająka, tak ważne jest przestrzeganie zasad podczas jego projektowania.

Wybieranie najbardziej strukturalnej postaci danych. Pliki HTML są w niewielkim stopniu ustrukturyzowane, gdyż koncentrują się na formie danych, a nie na ich treści. Często witryny występują w różnych postaciach; warto spytać o wersję XHTML czy XML, która jest czystsza i ma ściślej określoną strukturę. Uproszczona postać XML, RSS występuje wszędzie.

Analiza HTML, jeśli jest konieczna, powinna być możliwie prosta. Jeśli potrzebne informacje są dostępne jedynie w formie HTML, warto poszukać wersji tekstowej lub wersji do wydruku; wersje te zwykle mają prostszy kod HTML, wyższy wskaźnik proporcji treści do prezentacji i w mniejszym stopniu podlegają zmianie podczas reorganizacji strony.

Niezależnie od tego, jakie będzie ostatecznie źródło danych, należy analizować możliwie mało kodu HTML — tylko tyle, ile jest niezbędne do zlokalizowania odpowiednich danych. Im mniej kodu HTML, tym mniej wrażliwy będzie pająk. Więcej informacji na ten temat podano w **„Anatomii strony HTML"** [Sposób 3.]

Dobór właściwych narzędzi. Czy stronę należy analizować za pomocą wyrażeń regularnych? A może lepiej zastosować bardziej uniwersalne narzędzie jak **WWW::Mechanize** [Sposób 22.] czy **HTML::TokeParser** [Sposób 20.]? W dużym stopniu zależy to od interesujących nas danych oraz od konstrukcji kodu HTML. Czy jest on pisany ręcznie i nie zachowuje ustalonych konwencji, czy jest generowany przez jakieś narzędzie i przez to zawiera powtarzające się wzory? Należy wybrać najprostszą i najmniej podatną na awarię metodę, która jest dostępna; zresztą odporność na awarie jest z tych dwóch rzeczy ważniejsza.

Nie wchodź w szkodę. Stworzony przez nas skrypt może być wyjątkowo wyrafinowany, ale nie ma to żadnego znaczenia, jeśli witryna, w której chcemy go uruchamiać, na to nie pozwoli. Przed zabrnięciem w taką sytuację trzeba sprawdzić, czy w interesującej nas witrynie możliwe jest korzystanie z pająków i dążyć do **jak najmniejszego zużycia pasma i zasobów** [Sposób 16.] Więcej informacji na te tematy, z uwzględnieniem kwestii prawnych, Czytelnik znajdzie w podrozdziałach **„Ostrożnie z użyciem pająka"** [Sposób 6.] oraz **„Poszanowanie dla pliku robots.txt"** [Sposób 17.]

Dobór właściwego identyfikatora. Zapisując identyfikator dla swojego pająka, trzeba wybrać taki, który jasno określi możliwości pająka: jakie informacje ma zbierać i do czego jest używany. Nie trzeba pisać powieści: wystarczy jedno zdanie. Identyfikatory takie nazywane są *agentami użytkownika* (`User-Agents`) i ustawianiem ich zajmiemy się w podrozdziale **„Dodawanie do żądania nagłówków HTTP"** [Sposób 11.]

Cokolwiek się robi, nie należy podszywać się pod istniejące pająki, jak Googlebot, ani nadawać identyfikatora, który łatwo jest pomylić z istniejącym pająkiem. Nie dość, że pająk będzie od razu traktowany jako podejrzany, to łatwo o kłopoty ze strony Google

czy innej firmy, pod którą się podszywamy. Ewentualne konsekwencje takiego postępowania omówimy w podrozdziale **„Ostrożnie z użyciem pająka”** [Sposób 6.]

Udostępnianie informacji o pająku. Warto przygotować stronę internetową z informacjami o stworzonym pająku oraz z danymi kontaktowymi. Strona ta powinna być oczywiście widoczna w ulubionej przeglądarce. W podrozdziale **„Rejestrowanie pająka”** [Sposób 4.] omówione zostaną sposoby informowania innych o istnieniu pająka.

Nie wymagaj zbyt wiele. Stworzenie nawet najdoskonalszej aplikacji od chwili pojawienia się w Google programu PageRank niewiele znaczy, ponieważ to i tak administrator strony decyduje o prawach dostępu do niej czy do jej poszczególnych obszarów. Należy uprzejmie prosić o taki dostęp, nie zaś kategorycznie go żądać. Dzielmy się swoimi doświadczeniami: być może warto nawet przedstawić stworzony kod! W końcu przecież chcemy pobierać z ich strony informacje. Udostępnienie programu w zamian za informacje jest uczciwym postawieniem sprawy.

Wskazówki dla pająka

Pisząc pająka, trzeba pamiętać o pewnych zasadach.

Poszanowanie dla pliku robots.txt. Plik *robots.txt* znajduje się w katalogu głównym witryny; stanowi on informację dla pająków o tym, do czego na danym serwerze mogą mieć dostęp. Plik ten może nawet spowodować, że niektóre pająki będą musiały opuścić stronę bez zaglądania gdziekolwiek. Wielu administratorów witryn wykorzystuje sposób traktowania tego pliku przez pająka jako papierek lakmusowy: jeśli pająk ignoruje *robots.txt*, zwykle jest wyrzucany ze strony. Szczegółowe porady na ten temat znajdują się w podrozdziale **„Poszanowanie dla pliku robots.txt”** [Sposób 17.].

Poza plikiem *robots.txt* istnieje jeszcze znacznik META Robots (*http://www.robotstxt.og/wc/exclusion.htmlk#meta*), który podaje instrukcje dla pająków dotyczące indeksowania dla poszczególnych stron. Protokół znacznika META Robots nie jest nawet w przybliżeniu tak uniwersalny jak plik *robots.txt* i jest znacznie rzadziej uwzględniany przez pająki.

Ograniczanie obciążenia. Nawet jeśli zawartość witryny jest doskonała i chcielibyśmy dla naszej aplikacji pozyskać jak najwięcej danych, nie warto być zachłannym. Jeśli pająk próbuje w krótkim czasie pobrać zbyt wiele danych — dziesiątki czy nawet setki stron na sekundę — może to się w widoczny sposób odbić na dopuszczalnej przepustowości strony oraz na możliwości dostępu do tej strony przez innych użytkowników. Można wtedy usłyszeć „Ten kretyński pająk zapchał mi cały serwer i wszystko działało koszmarnie wolno!”

Nie ma jakichś określonych norm, jak szybko pająk może pobierać strony, aby nie być uznanym za uciążliwego intruza. Uczestnicy forum WebmasterWorld.com najczęściej mówili o jednym, dwóch żądaniach na sekundę.

WebmasterWorld.com (*http://www.webmasterworld.com*) to działające w internecie forum miłośników wyszukiwarek i administratorów witryn z całego świata. Odbywa się tam wiele dyskusji na istotne tematy. Co najważniejsze, głos zabierają przedstawiciele niektórych wyszukiwarek i liczących się witryn.

Niestety, wydaje się, że łatwiej jest powiedzieć, czego nie można zaakceptować, niż podać jakiś rozsądny limit. Dla osób cierpliwych jedno czy dwa żądania na sekundę wystarczą; intensywniejsze pobieranie danych może spowodować furię właściciela strony. Zrobić można wszystko, byle był na to czas; jeśli dane nie są potrzebne natychmiast, stopniowym ich pobieraniem może zająć się pająk. Więcej szczegółów na ten temat znajduje się w podrozdziale **„Ograniczanie zajmowanego pasma”** [Sposób 16.]

Tyle, ile trzeba i wtedy, kiedy trzeba. Nadmierne pobieranie to po prostu branie więcej danych niż trzeba i przez to zajmowanie szerszego pasma niż jest to niezbędne. Jeśli potrzebna jest strona, należy pobrać stronę. Nie wolno wtedy pobierać całego katalogu ani (niech Bóg broni!) całej witryny.

To samo dotyczy czasu. Nie należy pobierać strony częściej niż jest to konieczne. Jeśli program z danymi z takiej strony będzie uruchamiany raz dziennie, wystarczy te dane raz dziennie pobrać. Nie zaleca się pobierania danych częściej niż raz na godzinę, jeśli nie jest to absolutnie niezbędne (choć i tak wymagana jest wtedy zgoda właściciela strony).

SPOSÓB 3.

Anatomia strony HTML

Aby umieć pobierać dane, nie wystarczy umieć programować; trzeba jeszcze znać HTML i znać różne rodzaje plików związanych z witrynami sieciowymi.

Dla osób dopiero zapoznających się ze światem pająków internetowych podjęcie decyzji, co należy pobrać i dlaczego, nie jest łatwe. Zdarza się pobierać zbyt dużo danych, zbyt mało lub pobierane są dane, które zmieniają się częściej niż powinny. Znajomość budowy plików HTML ułatwia wybieranie tylko tych danych, o które chodzi.

Pliki HTML to zwykłe pliki tekstowe, tyle że specjalnie sformatowane. Z takimi właśnie plikami zwykle mamy do czynienia, zajmując się pająkami: zarówno w tej książce, jak i w codziennej praktyce. Wprawdzie będziemy dalej zajmować się szukaniem i pobieraniem plików multimedialnych: obrazków, filmów, plików dźwiękowych, ale nie będziemy ich analizować ani wydobywać z nich żadnych danych.

Anatomia strony HTML

Stronę HTML można sformatować na niemalże tyle sposobów, ile stron jest w internecie. Jeśli mamy nakazać pająkowi wyszukiwać na stronie HTML interesujące nas dane, musimy wiedzieć, jak takie strony są zbudowane i jak mogą być ułożone w nich informacje.

Zasadniczo strona HTML wygląda tak:

```
<html>
<head>
```

```
    <title>
      Tytuł strony
    </title>
  </head>
  <body>
    Treść strony
  </body>
  </html>
```

I to tyle. Taki jest ogólny zarys 99% stron HTML znajdujących się w sieci. Strony te mogą być o wiele bardziej rozbudowane, ale ogólna zasada jest właśnie taka. Co to oznacza dla naszych pająków? Oznacza to, że tylko jeden fragment informacji jest wyraźnie oznakowany: tytuł strony. Jeśli potrzebny był nam tylko tytuł, znaleźliśmy to, o co nam chodziło.

Jeśli jednak potrzebne są nam dane z treści strony, na przykład nagłówek czy data, jeszcze sporo pracy przed nami. Niejednokrotnie treść strony zawiera kilka tabel, kod JavaScript i inne fragmenty kodu utrudniające dotarcie do tego, co nas interesuje; wszystkie te niedogodności w znacznie większym stopniu wiążą się z formatowaniem, a nie z organizacją danych. Z drugiej jednak strony język HTML zawiera pewne standardowe metody organizowania danych. Niektóre z nich powodują powiększenie pewnych informacji na ekranie jako nagłówków. Inne służą do zestawiania danych w listy. Zrozumienie sposobu działania tych metod ułatwi wybieranie informacji ukrytej głęboko w strukturze strony.

Treści nagłówkowe w znacznikach H

Istotne informacje na stronie (nagłówki, podtytuły, uwagi i tak dalej) są zwykle ujmowane znacznikami `<Hx>`, gdzie *x* to cyfra od `1` do `6`. Standardowo treść znacznika `<H1>` jest na ekranie największa, gdyż jest to nagłówek stojący w hierarchii najwyżej.

Czasami pobranie nagłówków pozwala zorientować się w zawartości tej strony, choć zależy to od jej konstrukcji. Jeśli na przykład pobieramy dane ze strony z najświeższymi wiadomościami i wiemy, że nagłówki zawsze są ujmowane znacznikami `<H2>`, zaś podtytuły znacznikami `<H4>`, można pobrać odpowiednie znaczniki i pobrać krótki spis treści artykułów bez konieczności analizy całego oznakowania. Tak naprawdę, jeśli wiadomo, że strona zawsze jest oznakowana w opisany sposób, można określić treść całej witryny na podstawie wymienionych znaczników, bez konieczności analizowania reszty stron.

Specjalne znaczniki HTML obsługujące listy

Nie każdy projektant witryny używa list do organizowania danych; niektórzy korzystają po prostu z kolejnych, numerowanych akapitów. Jednak listy zwykle tworzy się za pomocą specjalnych znaczników.

Listy uporządkowane (których pozycje są automatycznie numerowane) ograniczone są znacznikami `<ol>` i `</ol>`, natomiast każdy element listy ograniczony jest znacznikami `<li>` i `</li>`. W przypadku używania wyrażeń regularnych do pobierania informacji,

można wybrać całą treść spomiędzy <ol> i , przeanalizować poszczególne pary , wstawić ich zawartość do tablicy i dalej normalnie już przetwarzać. Oto przykład listy uporządkowanej:

```
<ol>
  <li>jajka</li>
  <li>mleko</li>
  <li>masło</li>
  <li>cukier</li>
</ol>
```

Listy nieuporządkowane są bardzo podobne do list uporządkowanych, ale zamiast liczb pokazywane są wyróżniki graficzne (zazwyczaj kółka), natomiast cała lista ujęta jest w parę znaczników zamiast .

Pliki inne niż HTML

Niektóre pliki inne niż HTML są równie zmienne jak pliki HTML, inne z kolei są lepiej opisane. Przykładowo, zwykłe pliki *.txt* (których jest w sieci mnóstwo) nie mają żadnych informacji formatujących, nawet tak elementarnych, jak oddzielenie tytułu od treści. Pliki te jednak często są łatwiejsze do analizy, gdyż nie mają mnóstwa kodu HTML utrudniającego ich interpretowanie.

Drugą skrajnością są pliki XML (XML to rozszerzalny język znaczników, ang. *eXtensible Markup Language*). Poszczególne części pliku XML są opisane znacznie lepiej niż ma to miejsce w HTML. RSS, format wymiany danych stanowiący uproszczoną formę XML, ma jasno określone części plików, takie jak tytuły, treść, łącza i informacje dodatkowe. W książce tej często używamy plików RSS; ściśle zdefiniowane części są łatwe do analizy i zapisu w Perlu. Zobacz **„Użycie XML::RSS do przekształcania danych”** [Sposób 94.]

Pierwsze, co trzeba zrobić przed pobraniem danych, to określenie typu pliku. Jeśli jest to zwykły plik tekstowy, *.txt*, niemożliwe będzie dokładne wskazanie interesujących nas treści. Jeśli jest to plik XML, możliwe będzie sięgnięcie do potrzebnych części za pomocą wyrażeń regularnych lub skorzystanie z jednego z wielu perlowych modułów obsługujących XML (jak *XML::Simple*, *XML::RSS* czy *XML::LibXML*).

XHTML: Hybryda XML i HTML

Z poprzednich przykładów wywnioskować można, że o ile w HTML duża część kodu dotyczy formatowania, to sposób zorganizowania danych na typowej stronie jest zdecydowanie słabiej opisany. Pojawił się już jednak standard XHTML (rozszerzalny język znaczników hipertekstowych, ang. *eXtensible HyperText Markup Language*). Być może w przyszłości XHTML zastąpi HTML. Sposób zapisu stron w nowym języku jest ściślej narzucony niż w HTML, zaś uzyskiwany kod jest prostszy w analizie.

SPOSÓB 4.

Rejestrowanie pająka

Jeśli planujemy użycie jakiegoś pająka, musimy się choćby w minimalnym stopniu upewnić, że łatwo go będzie zidentyfikować. Nawet najprostszy pająk może być przedmiotem intensywnego zainteresowania.

W internecie stale toczą się wojny: czy to między spamerami i antyspamerami, czy to między zwolennikami systemów wymiany plików i ich przeciwnikami. Mniej agresywna wojna toczy się między pająkami sieciowymi a właścicielami witryn, którzy nie życzą sobie działań pająków.

Każdy może być przeciwny analizowaniu jego stron przez pająki. Niestety, nie wszystkie pająki są tak przyjazne jak Googlebot — program indeksujący serwisu Google. Wiele pająków przemieszcza się po rozmaitych stronach w celu zbierania adresów poczty elektronicznej dla spamerów. Jeszcze inne nie stosują się do **zasad przyjaznego pobierania danych** [Sposób 2.] W tej sytuacji przez doświadczonych administratorów witryn pająki bywają traktowane bardzo podejrzliwie.

Tak naprawdę sytuacja jest na tyle poważna, że nasz pająk może zostać po prostu zablokowany. Biorąc to pod uwagę, trzeba starannie dobrać nazwę dla pająka, rejestrując go w sieciowych bazach danych oraz dbając o to, że będzie dostatecznie dobrze postrzegany w sieci.

Tak na marginesie, wydawać się może, że tworzony przez nas pająk będzie zbyt mało istotny, aby w ogóle ktokolwiek mógł go zauważyć. Jednak w rzeczywistości prawdopodobnie wcale tak nie będzie. Witryny typu Webmaster World (*http://www.webmasterworld.com*) mają całe fora poświęcone identyfikowaniu i omawianiu pająków. Nie należy zakładać, że pająk będzie ignorowany tylko dlatego, że nie używa tysięcy działających non stop serwerów i nie analizuje dziennie milionów stron.

Nazwa dla pająka

Pająkowi należy dobrać taką nazwę, która powie coś o tym, czym jest ten program i czemu ma służyć. Nie jest zatem dobrą nazwą Botprzykladowy, lepszą już jest PobierzObrazkiZNews czy w angielskiej wersji NewsImageScraper. Gdy pająk jest kontynuacją wcześniejszego programu, warto w nazwie oznaczyć wersję, na przykład NewsImageScraper/1.03.

W przypadku używania wielu pająków warto zastanowić się nad nadaniem im wspólnej nazwy; na przykład Kevin mógłby nadać każdemu z nich nazwę zaczynającą się od disobeycom: disobeycomNewsImageScraper, disobeycomCamSpider, disobeycomRSSfeeds i tak dalej. Jeśli działanie pająków będzie zgodne z przyjętymi powszechnie zasadami, podejrzliwy administrator, widząc kolejnego pająka o podobnej nazwie, może spojrzeć na niego przychylniejszym okiem. Jeśli natomiast programy te „będą bezczelne", podobne nazwy ułatwią administratorom szybkie ich odrzucenie — bo takie pająki na to tylko zasługują.

Te rozważania mogą nasunąć myśl: czemu nie nazwać swojego pająka tak, jak nazwany jest pająk już istniejący? W końcu przecież większość witryn udostępnia swoje zasoby pająkowi Googlebot — czemu nie skorzystać z jego nazwy?

Jak wspomnieliśmy, jest to zły pomysł, i to z wielu powodów; między innymi dlatego, że właściciel pająka, którego nazwa zostanie użyta, prawdopodobnie zablokuje imitatora. Istnieją witryny, takie jak *http://www.iplists.com*, poświęcone śledzeniu adresów IP legalnych pająków — jest na przykład cała lista adresów związanych z pająkiem Googlebot. Poza tym, choć nie ma dotąd zbyt bogatego orzecznictwa dotyczącego podszywania się pod czyjeś pająki, to firma Google już wykazała, że nie będzie uprzejmie traktowała nikogo nadużywającego czy używającego bez zezwolenia nazwy Google.

Strona poświęcona pająkowi

Kiedy już stworzymy pająka, musimy go zarejestrować. Warto dla niego stworzyć stronę w internecie, tak aby ciekawscy i ostrożni administratorzy witryn mogli łatwo znaleźć informacje o nim. Strona taka powinna zawierać:

- Nazwę, jaka pojawia się w dziennikach systemowych (jako `User-Agent`).
- Krótkie omówienie, czemu pająk ma służyć i co robi (wraz z hiperłączem do zasobów przez pająka zbieranych, o ile są one publicznie dostępne).
- Dane kontaktowe programisty, który stworzył pająka.
- Informacje o tym, jak administratorzy mogą w zależności od potrzeb zablokować skrypt lub ułatwić jego działanie.

Gdzie rejestrować pająka

Kiedy już mamy stronę poświęconą naszemu pająkowi, trzeba tego pająka zarejestrować w dostępnych w sieci bazach danych. Po co? Gdyż administratorzy witryn mogą zacząć szukanie pająka w bazach danych, zamiast podawać jego nazwę w wyszukiwarkach. Mogą też na podstawie tych baz danych podejmować decyzje, którym pająkom pozwolić na działanie w swoich witrynach. Oto kilka baz danych, od których można zacząć:

Baza robotów sieciowych (http://www.robotstxt.org/wc/active.html)
: Bazę tę można przeglądać w różnych formatach. Dodanie pająka wymaga wypełnienia szablonu i wysłania informacji na wskazany adres poczty elektronicznej.

Wyszukiwarka robotów (http://www.jafsoft.com/searchengines/webbots.html)
: Programy `User-Agent` i pająki ułożone są tu według kategorii: roboty wyszukiwarek, przeglądarki, kontrolery hiperłączy i tak dalej wraz z listami oszustów; dodatkowo znajdują się tu komentarze od administratorów.

Lista programów User-Agent (http://www.psychedelix.com/agents.html)
: Ta baza danych podzielona jest na wiele stron i często jest aktualizowana. Proces dodawania nowych pozycji nie jest dokładnie określony, choć na dole wszystkich stron znajduje się adres poczty elektronicznej.

Baza danych programów User-Agent (http://www.icehousedesigns.com/useragents/)
: Baza zawiera prawie 300 agentów; można ją przeszukiwać na różne sposoby. Witryna zawiera adres poczty elektronicznej, na który można przesłać swojego pająka.

SPOSÓB 5.

Przedstawiamy się

Zamiast czekać, aż ktoś naszego pająka wykryje, niech pająk przedstawi się sam!

Niezależnie od tego, jak dyskretny i ostrożny jest nasz pająk, wcześniej czy później zostanie zauważony. Niektórzy administratorzy witryn zechcą sprawdzić, do czego pająk służy i zechcą uzyskać odpowiedzi na szereg innych pytań. Zamiast czekać na to, co się stanie, czemu nie wziąć spraw w swoje ręce i samemu się nie przedstawić? Zastanówmy się, jak można się do tego zabrać, jak swojego pająka popierać i jak informować o nim innych.

Nawiązywanie kontaktu

Skoro napisaliśmy doskonałego pająka, czemu się nim nie pochwalić w witrynie? W przypadku małych witryn jest to względnie proste: wystarczy odszukać łącze Feedback, Kontakt, About czy temu podobne. W przypadku większych witryn jednak znalezienie osoby odpowiedniej do nawiązania kontaktu staje się trudniejsze. Najpierw należy sprawdzić kontakty techniczne, dopiero potem kontakty zwykłe. Okazuje się, że najlepiej w miarę możności unikać kontaktów z działami *public relations* (PR). Wprawdzie łatwo się z nimi skontaktować, gdyż zwykle to ich adresy są najbardziej widoczne, ale osoby z tych działów najchętniej rozmawiają z prasą, poza tym rzadko mają na tyle dużą wiedzę techniczną, aby zrozumieć, o co ich pytamy (do osób z działów PR: prosimy tego nie traktować jako lekceważenia; i tak was kochamy; promujcie nadal książki wydawnictwa O'Reilly — buziaczki, autorzy).

Jeśli naprawdę trudno znaleźć jakikolwiek rozsądny kontakt, można spróbować poniższych trzech kroków:

1. Wiele witryn, szczególnie poświęconych zagadnieniom technicznym, ma pracowników zajmujących się dziennikami. Warto sprawdzić, czy uda się te dzienniki znaleźć za pomocą wyszukiwarki Google. Jeśli na przykład interesują nas pracownicy Yahoo!, dobrze sprawdzi się zapytanie `"work for yahoo" (weblog | blog)`. Czasami można się z tymi właśnie osobami skontaktować i dogadać się, wtedy oni są w stanie przekazać list do osoby władnej prośbę spełnić lub jakoś inaczej odpowiedzieć.
2. W 99,9% przypadków zadziała adres *webmaster@* (na przykład *webmaster@przykladowa.witryna.com*). Nie zawsze jednak można zakładać, że osoba, korzystająca z tej skrzynki, czyta ją częściej niż raz na miesiąc, a bywa jeszcze gorzej.
3. Jeśli już nic nie działa, nie sposób znaleźć adresów poczty elektronicznej, a listy wysyłane na adres *webmaster@* wracają z powrotem, warto zajrzeć do witryny poświęconej rejestracji domen, jak *http://www.whois.org*. Zwykle można tam znaleźć przy adresie domeny jakiś adres kontaktowy, choć znowu nie ma żadnych gwarancji, że skrzynka tak jest w ogóle sprawdzana, a nawet że nie została już usunięta. Poza tym pamiętać trzeba, że działa to jedynie na poziomie domen najwyższego poziomu. Innymi słowy być może uda się uzyskać kontakt z adresem *www.przykladowy.com*, ale już nie *www.przykladowy.com/zasob/*.

Popieranie swojego pająka

Teraz, kiedy mamy już adres kontaktowy, należy przekazać nań jakieś argumenty przemawiające za naszym pająkiem. Jeśli jasno opiszemy, co pająk robi, to świetnie. Jednak może okazać się, że trzeba zakodować przykład do pokazania administratorowi. W przypadku gdy nasz rozmówca nie jest znawcą Perla, być może warto stworzyć działającą po stronie klienta wersję skryptu narzędziem *Perl2Exe* (*http://www.indigostar.com/perl2exe.htm*) lub *PAR* (*http://search.cpan.org/author/AUTRIJUS/PAR*) i taką wersję wysłać jako testową.

Nie wahajmy się udostępnić naszego kodu. Wyjaśnijmy, jak działa. Podajmy przykładowe wyniki. Jeśli kod się spodoba, zaproponujmy rozpowszechnianie go z witryny, o którą nam chodzi! Pamiętajmy, że wszyscy administratorzy, niezajmujący się programowaniem, spodziewają się stwierdzenia typu: „Cześć, napisałem ten program i on na Twojej stronie robi to i tamto. Czy nie masz nic przeciwko temu, abym go użył?" Jasne jest, że administrator będzie oczekiwał pełnych wyjaśnień i pewnych gwarancji.

Pająk powinien być widoczny

Kolejnym dobrym sposobem zapewnienia, że inni będą wiedzieli o naszym pająku, jest **zawarcie w klauzuli `User-Agent` pająka danych kontaktowych** [Sposób 11.] Może to być adres poczty elektronicznej lub adres strony. Trzeba pamiętać potem o sprawdzaniu tego adresu i zapewnieniu, że znajdą się pod nim oczekiwane informacje.

Kwestie prawne

Mimo nawiązania kontaktów, uzyskania pozwolenia i udostępnienia mnóstwa informacji o pająku w sieci, jego działanie nadal może budzić pewne wątpliwości. Czy pająk jest legalny? Czy użycie go nie pociągnie za sobą kłopotów?

Jeśli chodzi o prawa dotyczące sieci, to istnieje jeszcze wiele kwestii otwartych i sędziowie, eksperci i naukowcy, nie mówiąc już o zwykłych użytkownikach, nie są zgodni co do wielu zagadnień. Uzyskanie pozwolenia i przestrzeganie jego warunków pozwoli uniknąć licznych kłopotów, szczególnie w przypadku małych witryn (utrzymywanych przez pojedyncze osoby, a nie przez wielkie korporacje). Jeśli działamy bez uzyskania pozwolenia, natomiast warunki użycia witryny nie są wyraźnie określone, ryzyko stosowania pająka jest już większe. Podobne ryzyko istnieje zwykle w przypadku działania w witrynach, w których nie zapytaliśmy o pozwolenie, a które oferują interfejs API i mają jasno określone zasady użycia (jak Google).

Warunki prawne używania internetu stale się zmieniają: medium to jest po prostu zbyt nowe, aby mogły istnieć w nim niezmienne warunki, określające, co jest dopuszczalne, a co nie. Nie chodzi tylko o to, jak pająk działa, ale też o to, co może zbierać. Autorzy pragną ostrzec, że stosowanie jednego z opisanych w książce sposobów wykorzystywania pająka nie oznacza, że nie wiąże się to z takim czy innym ryzykiem i że żaden administrator nie uzna tego za naruszenie jego praw lub praw innych podmiotów.

Trzeba używać zdrowego rozsądku (niewątpliwie nierozsądne jest pobranie wszystkiego z jakiejś witryny, umieszczenie tego w swojej i uważanie, że wszystko jest w porządku). Trzeba też prosić o pozwolenie — najgorsze, co się może zdarzyć, to odmowa. W przypadku naprawdę poważnych wątpliwości pozostaje porozmawiać z dobrym prawnikiem.

SPOSÓB 6.

Ostrożnie z użyciem pająka

Tu i tam pojawiają się ciekawe dane. Zanim po nie sięgniemy, sprawdźmy, jak można z danej witryny korzystać.

Ponieważ celem naszej książki jest pokazanie, jak pobierać dane niedostępne dla interfejsu API, czasami może się okazać, że działamy w szarej strefie. Oto kilka porad, które będą pomocne w uniknięciu zablokowania nas lub nawet zaskarżenia.

Być może pewnego dnia po odwiedzeniu jakiejś strony znajdziemy na niej wspaniałe dane, które bardzo chcielibyśmy posiąść. Zanim jednak zabierzemy się za ich pobieranie, warto rozejrzeć się za zasadami dopuszczalnego użycia (ang. *Acceptable Use Policy*, AUP) lub warunkami świadczenia usług (ang. *Terms of Service*, TOS); czasami mogą to być też warunki użycia (ang. *Terms of Use*, TOU). W takich dokumentach przeczytać można, co jest w witrynie dopuszczalne i co wolno zrobić z danymi z tej witryny. Zwykle na dole strony znajduje się hiperłącze do strony z informacjami o prawach autorskich. Odpowiednie hiperłącze w witrynie Yahoo! nazywa się Terms of Reference i jest przy samym końcu strony głównej, natomiast w witrynie Google hiperłącze to znajduje się na dole strony About. Jeśli odpowiedniego łącza nie uda się znaleźć na stronie głównej, warto przeszukać wszelkie strony About. Czasami witryny, szczególnie te mniejsze, nie mają w ogóle odpowiednich zapisów, należy więc skontaktować się z administratorem, niemalże zawsze dostępnym pod adresem *webmaster@nazwa.witryny.com*, i zapytać o zgodę.

Tak więc mamy już AUP czy TOS. Czego właściwie szukamy? Czegokolwiek, co dotyczy używania pająków i automatycznego zbierania danych. W przypadku aukcji eBay wszystko jest jasne, gdyż wynika z poniższego wyjątku z umowy:

> *Użytkownicy zgadzają się nie używać robotów, pająków, programów zbierających dane ani innych automatów do korzystania z Witryny w żadnym wypadku, o ile nie uzyskają naszej pisemnej zgody.*

Jasne, prawda? Czasami jednak nie jest to tak wyraźnie opisane. Niektóre umowy nie mają żadnych odniesień do pająków czy programów zbierających dane. W takich wypadkach należy skontaktować się z administratorem lub pracownikami technicznymi i spytać.

Niedobry pajączek, a sio!

Nawet jeśli stosujemy się do obowiązujących warunków korzystania ze strony, może okazać się, że nasz pająk powoduje problemy. Jest kilka powodów, dla których pająk, mimo że działa zgodnie z literą prawa, może być nie do przyjęcia dla właścicieli stron. Na przykład w witrynie może być umieszczony zakaz dalszego rozpowszechniania jej treści w internecie. Wtedy pojawia się nasz pająk i pobiera dane w formie RSS. Pakiet

RSS formalnie nie jest stroną sieciową, ale autorzy witryny i tak mogą takie działania uznać za niedopuszczalne. Nic nie zakazuje właścicielom takiej witryny zmienić zapisy TOS, tak aby uniemożliwić działanie pająka z jednoczesnym wysłaniem nam zakazu dalszego prowadzenia naszej działalności.

Pomińmy na chwilę wszystkie te zastrzeżenia. Nie namawiamy oczywiście nikogo do naruszania warunków świadczenia usług, wykłócania się z prawnikami i tak dalej. Warunki świadczenia usług czemuś przecież służą; zwykle są w nich opisane zasady, których przestrzeganie pozwala na utrzymanie strony. Cokolwiek robi nasz pająk, musi to robić tak, aby nie utrudniać normalnego funkcjonowania wykorzystywanej strony. Jeśli pająk pobierze wszystkie informacje z witryny utrzymywanej z reklam, niemożliwe będzie dalsze korzystanie z tej metody finansowania, a wówczas co się stanie? Strona zniknie. Nie będzie strony, więc nasz pająk też nie będzie miał gdzie działać.

Wprawdzie rzadko ma to związek z używaniem pająków, ale przypomnijmy, że już od dawna panuje zgoda co do tego, że *framing* danych jest prawnie niedopuszczalny. Framing danych polega na tym, że czyjaś witryna jest umieszczana w cudzej ramce (w wyniku tego czyjeś dane pojawiają się pod inną marką). Strona zwykle zawiera reklamy, z których ktoś się utrzymuje; pobieranie treści strony pająkiem i wstawianie we własne strony z ramkami jest niewątpliwie niedopuszczalne i nie należy tego robić.

Naruszanie praw autorskich

Nawet nie powinniśmy o tym mówić, ale formalności musi stać się zadość. Jeśli używamy pająka po to, aby w naszej witrynie umieścić czyjąś własność intelektualną, naruszamy prawo. Niechby pająk najściślej jak tylko można przestrzegał warunków świadczenia usług i trzymał się wszystkich ogólnie przyjętych zasad, to jego użycie w takim celu byłoby nielegalne. W takim wypadku pająka nie daje się poprawić, gdyż problem nie leży w kodzie. Lepiej zastanowić się nad celem stosowania skryptu. Więcej informacji o prawach autorskich i własności intelektualnej w sieci znaleźć można w dzienniku Lawrence'a Lessiga dostępnym pod adresem *http://www.lessig.org/blog/* (Lessig jest profesorem prawa Szkoły Prawniczej w Stanford), w witrynie fundacji Electronic Frontier Foundation (*http://www.eff.org*) oraz Copyfight (*http://www.copyfight.org/*).

Agregowanie danych

Agregowanie danych polega na zbieraniu danych z różnych źródeł i zestawianiu ich wszystkich razem. Wyobraźmy sobie witrynę zawierającą ceny biletów różnych linii lotniczych albo witrynę umożliwiającą porównywanie cen z różnych księgarni internetowych. W internecie działają już serwisy skupiające różne dane; stanowią one swoistą szarą strefę internetowej etykiety. Niektóre firmy wyraźnie nie życzą sobie gromadzenia ich danych i porównywania ich z danymi z innych stron (na przykład aby porównywać ceny sklepowe), dla innych firm nie ma to żadnego znaczenia. Istnieją firmy, które podpisują umowy określające zasady skupiania ich informacji! Rzadko tego typu informacje ujmowane są w warunkach świadczenia usług, więc w razie wątpliwości trzeba pytać.

Wywiad gospodarczy

Właściciele niektórych witryn mają za złe, że ich konkurenci pobierają za pomocą pająków dane dostępne publicznie, przez dowolną przeglądarkę, i wykorzystują je do uzyskania przewagi na rynku. Można się z takim stanowiskiem zgadzać lub nie, ale pozostaje faktem, że tego typu działania były już przedmiotem sporów prawnych; za użycie takiego pająka firma eBay oskarżyła Bidder's Edge (*http://pub.bna.com/lw/21200.htm*).

Możliwe konsekwencje nadużyć pająków

Co się stanie, jeśli napiszemy pająka działającego niezgodnie z przyjętymi normami i wypuścimy go w świat? Jest kilka możliwości. Wiele witryn po prostu zablokuje nasz adres IP. Dawniej Google blokowało grupy adresów IP, próbując za pomocą standardowego, automatycznego procesu wyłapywać wszystkie przypadki naruszenia TOS. Inną możliwą konsekwencją jest wysłanie listu z żądaniem zaprzestania danej działalności; w zależności od udzielonej przez nas odpowiedzi konflikt może przybrać różne formy, włącznie z procesem sądowym.

Grożą więc nam nie tylko straty związane z przegranymi procesami cywilnymi, ale w przypadkach szczególnie drastycznych również grzywny, a nawet kara więzienia, ponieważ niektóre działania dotyczące publikacji w sieci podlegają przepisom ogólniejszym (np. zasadom prawa autorskiego) lub normom prawa karnego.

Napisanie źle wychowanego pająka rzadko powoduje wizytę policji, chyba że jest to „stworzenie" wyjątkowo paskudne, na przykład powodujące zalew witryny danymi lub, mówiąc inaczej, uniemożliwiające jej normalną działalność (jest to atak typu DoS, *denial of service* — odmowa dostępu). Jednak, abstrahując już od honorariów prawników, zmarnowanego czasu i ewentualnych kar finansowych, sam proces może być dostatecznie nieprzyjemny, aby unikać pisania źle „zachowujących się" pająków.

Nadążanie za prawem

Aby być na bieżąco z zagadnieniami, związanymi z prawnymi aspektami pobierania informacji, warto użyć wyszukiwarki Blawg (*http://blawg.detod.com/*), która indeksuje jedynie dzienniki sieciowe poświęcone kwestiom prawnym. Można zastosować takie hasła, jak `spider`, `scraper` czy `spider lawsuit`. Osoby szczególnie zainteresowane tym tematem powinny wiedzieć, że wyniki działania Blawg dostępne są też w formie pakietów RSS, które mogą być używane w zwykłych systemach zbierających i prezentujących wiadomości. Można, korzystając ze sposobów podanych w tej książce, uruchomić własne pakiety RSS dotyczące własności intelektualnej.

Inne miejsca, w których można znaleźć aktualne informacje o stanie prawnym, to: Slashdot (*http://slashdot.org/search.pl?topic=123*), popularne miejsce spotkań różnego rodzaju dziwaków; Electronic Freedom Foundation (*http://www.eff.org*) — fundacja, której strony poświęcone są prawu cyfrowemu, oraz działająca w Harwardzkiej Szkole Prawa organizacja Berkman Center for Internet & Society (*http://cyber.law.harvard.edu/home/*), publikująca program badawczy poświęcony cyberprzestrzeni i związanym z nią zagadnieniom.

SPOSÓB 7.

Szukanie wzorców w identyfikatorach

Jeśli okaże się, że w interesującej nas sieciowej bazie danych lub kolekcji zasobów wykorzystywane są niepowtarzalne numery identyfikacyjne, można rozszerzyć jej funkcje przez połączenie jej z innymi witrynami i wartościami identyfikującymi.

Niektóre dostępne w sieci kolekcje danych są po prostu dużymi zbiorami zestawionymi w jednym miejscu, zorganizowanymi za pomocą programu bazodanowego lub wyszukiwarki. Kolekcje takie nie wykorzystują żadnych numerów identyfikujących, które ułatwiłyby ustalenie w nich jakiejś struktury. Jednak nie zawsze tak jest.

W miarę jak coraz więcej bibliotek udostępnia w sieci swoje zbiory, coraz więcej rekordów i stron ma swoje niepowtarzalne numery identyfikacyjne.

Cóż z tego? Otóż to, że kiedy witryna używa jakiejś metody identyfikującej swoje informacje, zrozumiałej dla innych witryn, korzystając z tejże metody można pobierać dane ze wszystkich tych witryn. Załóżmy na przykład, że chcemy zwiedzić Stany Zjednoczone, grając w golfa, ale obawiamy się zanieczyszczenia środowiska, wobec czego grać chcemy jedynie w obszarach czystych ekologicznie. Można byłoby napisać skrypt przeszukujący pola golfowe, dostępne pod adresem *http://www.golfcourses.com*, pobierający kody pocztowe znalezionych pól i sprawdzający te kody w witrynie *http://www.scorecard.org* w celu znalezienia najbardziej (lub najmniej) zanieczyszczonych okolic.

Przykład jest niepoważny, ale pokazuje, jak za pomocą niepowtarzalnego identyfikatora (tutaj kodu pocztowego) można powiązać ze sobą dwie sieciowe bazy danych, opisujące pola golfowe i stopień zanieczyszczenia różnych miejsc.

Ogólnie rzecz biorąc, dane w sieci mogą być zorganizowane trojako:

- W formie doraźnie ustalanych systemów klasyfikacji w ramach kolekcji.
- Jako systemy klasyfikacji korzystające z ogólnie przyjętych hierarchii danych z kolekcji.
- Jako systemy klasyfikacji identyfikujące dokumenty z wielu różnych kolekcji.

Doraźnie ustalane systemy klasyfikacji

Doraźnie ustalane systemy klasyfikacji albo nie są oparte na ogólnie przyjętych hierarchiach, albo do takich hierarchii jedynie luźno nawiązują. Jeśli dziesięć fotografii otrzyma niepowtarzalne kody zależne od tego, co fotografie te przedstawiają i zależne od zawartości w nich niebieskiej składowej koloru, mamy już doraźnie ustalony system klasyfikacji.

Przydatność doraźnie ustalanych systemów klasyfikacji jest ograniczona; nie można tych kodów używać w innych witrynach. Być może uda się wykryć w nich jakieś wzorce, które pozwolą pobrać duże ilości danych, ale może się to też nie udać (innymi słowy pliki oznaczone kodami *10A*, *10B*, *10C* i *10D* mogą być przydatne, natomiast pliki z kodami *CMSH113*, *LFFD917* i *MDFS214* już nie).

Systemy klasyfikacji oparte na ogólnie przyjętych hierarchiach

Najbardziej naturalne przykłady systemów klasyfikacji, wykorzystujących ogólnie przyjęte hierarchie, to katalogi biblioteczne oparte na klasyfikacji dziesiętnej Deweya, klasyfikacji Biblioteki Kongresu czy innej ogólnie przyjętej.

Systemy takie mają swoje zalety i swoje wady. Załóżmy, że szukamy książki *Google Hacks* na Uniwersytecie w Tulsa. Okazuje się, że numer LOC tej książki to ZA4251.G66 C3 2003. Wstawiając teraz ten numer do Google, otrzymamy około 13 odpowiedzi. Zatem jest dobrze: wyniki pochodzą z różnych bibliotek. Okazuje się, że mogliśmy wykonać takie zapytanie w Google, znajdując inne biblioteki mające *Google Hacks* i całemu temu pomysłowi **nadać postać skryptu** [Sposób 65.]

Pokazaliśmy zaletę, ale jest i wada: w ten sposób nie wyszukamy wszystkich bibliotek mających żądaną książkę. Inne biblioteki mają inne systemy klasyfikacji, jeśli więc potrzebna jest lista wszystkich bibliotek, nie można ograniczyć się tylko do opisanej metody. Jednak liczba znalezionych tak bibliotek w wielu wypadkach może być wystarczająca.

Systemy klasyfikacji identyfikujące dokumenty z wielu różnych kolekcji

Poza systemami klasyfikacji, opartymi na ogólnie przyjętych hierarchiach, istnieją jeszcze systemy wykorzystujące numery identyfikacyjne powszechnie uznawane i stosowane. Przykładami takich systemów mogą być:

ISBN (International Standard Book Number, Międzynarodowy Standardowy Numer Książki)
: Jak nietrudno zgadnąć, jest to system identyfikacji książek. Podobne numery przypisano czasopismom, muzyce, raportom naukowym i tak dalej. Numery ISBN pojawiają się wszędzie tam, gdzie wyliczane są książki: od katalogów bibliotecznych po księgarnię Amazon.com.

NIP (Numer Identyfikacji Podatkowej)
: Używany przez urzędy skarbowe. Numer ten pojawia się na wszelkich zeznaniach podatkowych, a także na wielu innych dokumentach, takich jak akty notarialne, zaświadczenia i tak dalej.

Kod pocztowy
: Służy Poczcie Polskiej do jednoznacznego identyfikowania obszarów.

Jest to zaledwie kilka przykładów z wielu powszechnie stosowanych systemów numeracji. Można oczywiście pójść dalej, podając takie cechy jednoznacznie identyfikujące obiekty, jak długość i szerokość geograficzna, numery stosowane w biznesie i administracji czy systemy kodowania obszarów. Cała sztuka polega na takim dobraniu systemów identyfikujących, które będą wymagały możliwie mało dodatkowych informacji do zadziałania w pająku. `"918"` to trzycyfrowy łańcuch dający w wyszukiwarce mnóstwo wyników, z których bardzo wiele nie będzie powiązanych z interesującymi nas danymi. Może się zatem okazać, że nie sposób wykluczyć w pająku ze znalezionego zbioru wyników zbędnych.

Z drugiej strony długie numery identyfikacyjne, takie jak katalog numerów LOC czy ISBN, będą dawały znacznie mniej lub wcale nie będą dawały błędnych wyników wyszukiwania. Im dłuższy i bardziej skomplikowany jest numer identyfikacyjny, tym lepiej nadaje się do automatycznego wyszukiwania i pobierania danych.

Wybrane duże zbiory z identyfikatorami

W sieci istnieje sporo miejsc wykorzystujących niepowtarzalne numery identyfikujące „rozumiane" przez wiele witryn. Oto kilka przykładowych:

Amazon.com (http://www.amazon.com), Abebooks (http://www.abebooks.com)
: Witryny te używają numerów ISBN. Połączenie danych z obu witryn pozwoli znaleźć najtańsze książki.

The International Standard Serial Number Register (http://www.issn.org)
: Chcąc skorzystać z tej witryny, trzeba się w niej zarejestrować, ale dostępne są darmowe konta próbne. Numery ISSN nadawane są zarówno czasopismom internetowym, jak i papierowym.

Poczta USA (http://www.usps.com)
: W tej witrynie można wyszukiwać zarówno standardowe, jak i dziewięciocyfrowe kody pocztowe USA; rozszerzone, dziewięciocyfrowe kody pozwalają dokładniej określać obszar oraz ułatwiają odrzucanie niepożądanych wyników szukania danych przez pająka.

GuideStar, baza danych dla organizacji typu *non-profit* oferuje stronę umożliwiającą wyszukiwanie danych według numerów EIN (numery pracownicze w USA; *http://www.guidestar.org/search/*). Także wiele innych amerykańskich biznesowych baz danych umożliwia wyszukiwanie według EIN.