دانشکده مهندسی کامپیوتر

پروژه پایانی

امنیت سیستم های کامپیوتری

استاد گرامی: دکترابولفضل دیانت

گردآورندگان: روزبه غزوی

محمد حسین قفقازیان

بهار ۱۴۰۲

بخش ۱

Web Crawler

۱.۱ Web Crawler و Web Scraper چیست؟

وب کرالینگ (Web Crawling) و وب اسکرپینگ (Web Scraping) دو تکنیک مختلف هستند که در حوزه جمع‌آوری اطلاعات از اینترنت استفاده می‌شوند.

وب کرالینگ یک فرآیند اتوماتیک است که توسط برنامه‌ها یا ربات‌ها اجرا می‌شود تا اطلاعات را از صفحات وب مختلف جمع‌آوری کند. این فرآیند شبیه به عملکرد موتورهای جستجو مانند گوگل است که به صورت مداوم صفحات وب را اسکن می‌کند تا اطلاعات جدید را کشف کند و در پایگاه داده خود به‌روزرسانی کند. وب کرالرها معمولاً اطلاعات از صفحات وب را براساس لینک‌های موجود در صفحات وب دنبال می‌کنند و اطلاعات را به‌طور خودکار و مجدداً جمع‌آوری می‌کنند.

وب اسکرپینگ از طرفی، فرآیندی است که در آن اطلاعات خاصی از صفحات وب استخراج یا استخراج می‌شود. در این فرآیند، برنامه‌ها یا ابزارهای وب اطلاعات خاصی را از صفحات وب به‌طور مستقیم استخراج می‌کنند. اطلاعاتی مانند متن، تصاویر، جداول، لیست‌ها و داده‌های ساختاری از صفحات وب به‌دست می‌آید. وب اسکرپینگ معمولاً برای تحلیل داده‌ها، استخراج اطلاعات تحقیقاتی و کشف الگوها در داده‌ها استفاده می‌شود.

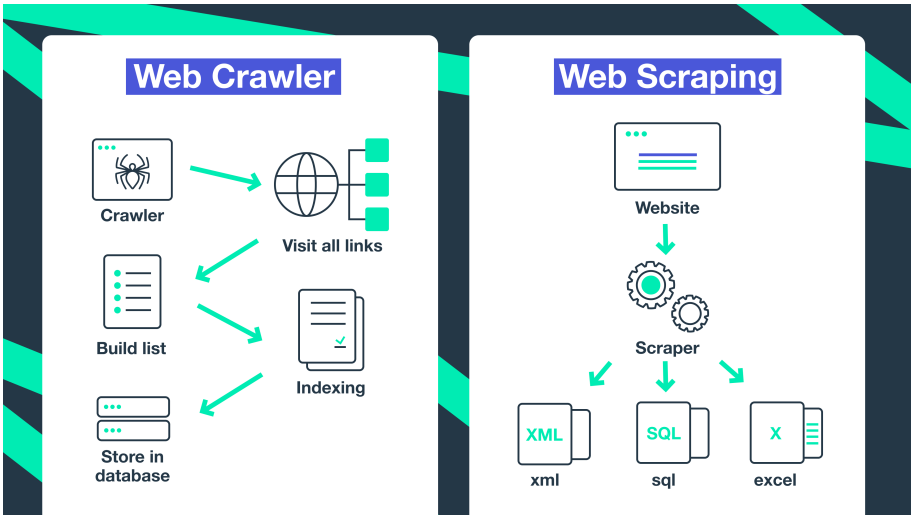
۲.۱ تفاوت ها

هدف: وب کرالینگ به‌صورت اتوماتیک صفحات وب را جستجو می‌کند و لینک‌ها و اطلاعاتی را که برای به‌روزرسانی پایگاه داده مورد نیاز است جمع‌آوری می‌کند. وب اسکرپینگ به‌صورت دقیق تر اطلاعات خاصی را از صفحات وب استخراج می‌کند که ممکن است در تحلیل داده‌ها و استفاده‌های دیگر مفید باشد.

روش عملکرد: وب کرالینگ با دنبال کردن لینک‌ها و لغوی‌های صفحات وب اطلاعات را جمع‌آوری می‌کند. وب اسکرپینگ به‌صورت مستقیم داده‌ها را از صفحات وب استخراج می‌کند.

محتوا: وب کرالینگ عمدتاً لینک‌ها و متن‌ها را از صفحات وب به‌دست می‌آورد. وب اسکرپینگ می‌تواند محتوای متنی، تصاویر، داده‌های جدولی و داده‌های ساختاری را استخراج کند.

به‌طور خلاصه، وب کرالینگ فرآیند جمع‌آوری اطلاعات از صفحات وب است، درحالی‌که وب اسکرپینگ تمرکز بیشتری بر استخراج اطلاعات خاص از صفحات وب دارد. هر دو تکنیک به‌طور گسترده‌ای در جمع‌آوری داده‌ها و اطلاعات از اینترنت و برای مقاصد مختلفی مورد استفاده قرار می‌گیرند.



بخش ۲

Dns Spoofing

دی‌ان‌اس اسپوفینگ یا DNS Spoofing (همچنین به نام تقلب دی‌ان‌اس یا DNS Cache Poisoning شناخته می‌شود) یک تکنیک حمله سایبری است که در آن حمله‌کنندگان ترافیک شبکه را متلاشی کرده و تغییر مسیر دی‌ان‌اس را به گونه‌ای جعل می‌کنند که کاربران به سایت‌های مخرب یا تقلبی هدایت می‌شوند.

در یک حمله دی‌ان‌اس اسپوفینگ، حمله‌کننده اطلاعات جعلی در مورد نام‌های دامنه و آدرس‌های آی‌پی در سرورهای دی‌ان‌اس پخش می کند. این اطلاعات جعلی با هدف جایگزینی داده‌های واقعی دی‌ان‌اس قرار داده می‌شوند. از این رو، کاربران که با سرور دی‌ان‌اس ارتباط بر قرار می کنند و می‌خواهند یک نام دامنه را به آدرس آی‌پی متناظر ترجمه کنند، داده‌های جعلی را دریافت می‌کنند و نتیجه متفاوتی دریافت می‌کنند. این تغییر در داده‌های دی‌ان‌اس باعث می‌شود که کاربران به سرورهای مختلفی هدایت شوند، از جمله سرورهایی که توسط حمله‌کنندگان کنترل می‌شوند.

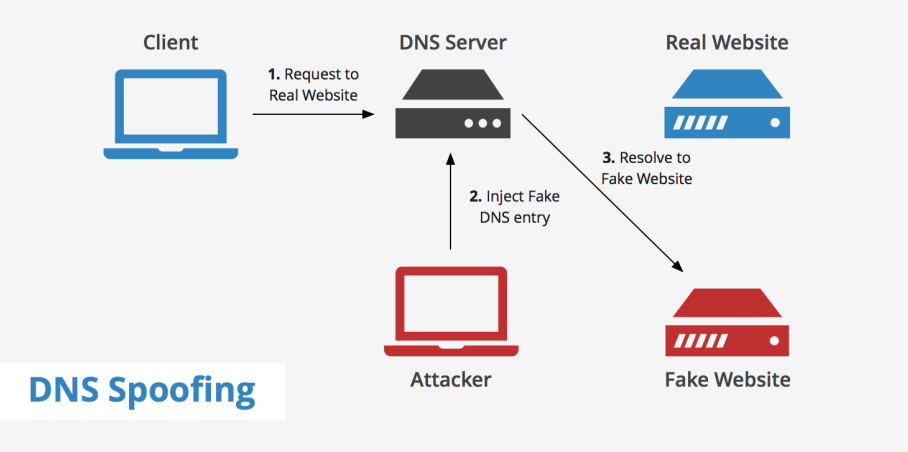
دی‌ان‌اس اسپوفینگ می‌تواند منجر به حملات مختلفی شود، از جمله:

هدایت کاربران به صفحات تقلبی: حمله‌کنندگان می‌توانند کاربران را به صفحات تقلبی هدایت کنند که شبیه به صفحات واقعی به نظر می‌رسند. این حمله معمولاً برای سرقت اطلاعات کاربری (مانند نام‌کاربری و رمز عبور) یا اطلاعات مالی استفاده می‌شود.

مسموم‌سازی حافظه کش دی‌ان‌اس: حمله‌کنندگان می‌توانند حافظه کش دی‌ان‌اس را با داده‌های جعلی مسموم کنند تا موجب اشتباهات و خطاهایی در نتایج جستجوها و ارتباطات شبکه شود.

حملات MITM (Man-in-the-Middle) : با استفاده از دی‌ان‌اس اسپوفینگ، حمله‌کنندگان می‌توانند محتوای ارتباطات بین کاربران و سرورها را مانند یک شخص در میانه کنترل کنند و اطلاعات حساس را زائل سازند.

برای مواجهه با حملات دی‌ان‌اس اسپوفینگ، بسیاری از سیستم‌ها از روش‌های امنیتی مانند DNSSEC (DNS Security Extensions) استفاده می‌کنند که امکان تأیید هویت و اصالت داده‌های دی‌ان‌اس را فراهم می‌کند.



بخش ۳

روش های تشخیص نفوذ

تشخیص وب کرالرها یا ربات‌های وب یک وظیفه مهم برای مالکان و مدیران وبسایت است تا از استفاده عادلانه از منابع محافظت کنند، در مقابل خطرات امنیتی محتمل اقدام کنند و اطلاعات مفیدی درباره بازدیدکنندگان سایت جمع‌آوری کنند. برای شناسایی وب کرالرها می‌توانید از روش‌های زیر استفاده کنید:

تجزیه و تحلیل User-Agent : وب کرالرها معمولاً با ارسال هدر User-Agent در درخواست‌های HTTP خود هویت خود را معرفی می‌کنند. با این حال، این هدر می‌تواند توسط ربات‌های مخرب تقلب شود، بنابراین کاملاً قابل اعتماد نیست. ربات‌های معتبر عبارات کلیدی خاصی را در رشته User-Agent خود قرار می‌دهند، مانند "Googlebot" برای کرالر گوگل یا "Bingbot" برای کرالر بینگ. می‌توانید هدرهای User-Agent ورودی را تجزیه و با رشته‌های User-Agent شناخته شده ربات‌ها مقایسه کنید تا وب کرالرها را تشخیص دهید.

تجزیه و تحلیل آدرس IP : می‌توانید لیستی از آدرس‌های IP مرتبط با وب کرالرهای شناخته شده نگهداری کرده و آدرس‌های IP درخواست‌های ورودی را با این لیست مقایسه کنید. با این حال، این روش به طور کامل اعتبار ندارد زیرا برخی از وب کرالرها از آدرس‌های IP پویا یا سرورهای پراکسی استفاده می‌کنند.

محدودیت نرخ: وب کرالرهای معتبر نرخ درخواست را رعایت می‌کنند و دستورات robots.txt را رعایت می‌کنند. می‌توانید قوانین محدودیت نرخ را پیاده‌سازی کنید تا تعداد درخواست‌هایی که از یک آدرس IP یا

User-Agent خاص می‌آید را کنترل کنید. این می‌تواند به شناسایی ربات‌ها که از محدودیت‌های تعیین شده فراتر می‌روند، کمک کند.

چالش‌های Captcha : پیاده‌سازی چالش‌های Captcha بر روی برخی از صفحات یا فرم‌ها به تمایز بین کاربران انسان و ربات‌ها کمک می‌کند. ربات‌های وب کرالر معتبر ممکن است در لیست سفید قرار بگیرند و کاربران تنها هنگام تشخیص رفتار مشکوک به حل چالش Captcha دعوت می‌شوند.

تشخیص JavaScript : برخی از ربات‌ها JavaScript را پردازش نمی‌کنند در حالی که اکثر مرورگرهای مدرن این کار را انجام می‌دهند. با اضافه کردن تست‌های مبتنی بر JavaScript به صفحات وب خود می‌توانید ربات‌هایی که JavaScript را به درستی اجرا نمی‌کنند را شناسایی کنید.

تجزیه و تحلیل رفتار: رفتار کاربران را نظارت کرده و الگوها را تجزیه و تحلیل کنید. ربات‌ها تمایل دارند مسیرهای ناوبری خاصی را دنبال کنند و رفتارهای مشخصی را نشان می‌دهند، مانند دسترسی به چندین صفحه به سرعت یا دنبال کردن پیوندها به یک شیوه پیش‌بینی‌پذیر.

آنالیز وب آنالیتیک: با استفاده از ابزارهایی مانند Google Analyticsداده‌های ترافیک وب را تحلیل کرده و الگوهایی که با وب کرالرها سازگار هستند را شناسایی کنید.

لیست سیاه‌گذاری: اگر وب کرالرها یا ربات‌های مخرب را شناسایی کردید، می‌توانید آدرس‌های IP یا رشته‌های User-Agent آنها را به لیست سیاه‌گذاری اضافه کرده و اجازه دسترسی آنها به وبسایت‌تان را محدود کنید.

مهم است به یاد داشته باشید که این روش‌ها تشخیص وب کرالرها را ممکن می‌سازند، اما کامل نیستند. برخی از ربات‌های پیشرفته می‌توانند رفتاری شبیه به رفتار انسان داشته باشند و از تکنیک‌های پیچیده‌تر برای جلوگیری از شناسایی استفاده کنند. به‌روزرسانی و بهینه‌سازی دوره‌ای روش‌های شناسایی شما از اهمیت بالایی برخوردار است تا همواره در مقابل فناوری‌های ربات‌ها قرار داشته باشید.

- روش ترکیبی Hilstone برای تشخیص web crawlerهای مشکوک

استفاده از تجزیه و تحلیل دستی و ایستا[1] در داده‌های لاگ‌شده (بر اساس آدرس‌های IP بازدید شده) می‌تواند کار فشرده‌ای باشد و هزینه‌های بالا و سربار زیادی را به همراه داشته باشد. راه‌حل ترکیبی Hillstone از یک مکانیسم مدل‌سازی رفتاری خودآموز اختصاصی استفاده می‌کند که در شناسایی web crawlerهای کند مؤثرتر است. همچنین تجزیه و تحلیل‌های آماری را برای شناسایی خودکار crawler وب پیچیده و مشکوک را برای مدیر[2] ارائه می‌دهد.

این روش از تجزیه و تحلیل داده‌های ثبت آماری استفاده می‌کند و مهم‌تر از آن، بر مدل‌سازی رفتاری برای شناسایی crawler وب مشکوک تمرکز می‌کند. ثابت شده است که این کار در شناسایی crawler وب پیچیده و مخرب و همچنین crawlerهای کندی که مستعد از دست‌دادن ردیابی هستند مؤثر است. در این رویکرد ترکیبی، مجموعه‌ای از ویژگی‌های رفتاری L3-L7 از پیش تعریف‌شده، داده‌ها را در صفحه داده، نظارت و جمع‌آوری می‌کنند، که سپس با استفاده از الگوریتم‌های یادگیری ماشین که به‌طور دوره‌ای این ویژگی‌های رفتاری را یاد می‌گیرند.

به صورت پشت سر هم، داده‌های ثبت‌شده ترافیک سطح شبکه و application جمع‌آوری شده در دوره‌های زمانی خاص نیز پردازش، مرتب‌سازی، فیلتر و تجزیه و تحلیل می‌شوند.

بر اساس نتایج پیش‌بینی‌کننده مدل‌سازی رفتاری و تجزیه و تحلیل آماری از داده‌های گزارش، مجموعه‌ای از قوانین همبستگی(correlation) برای همبستگی نتایج مربوطه از ماژول‌های تشخیص مختلف تعریف شده‌اند. آن‌ها برای شناسایی آن دسته از آدرس‌های IP استفاده می‌شوند که در مقایسه با آدرس‌های IP که دارای رفتار عادی دسترسی به وب و browsing هستند، "غیر عادی" هستند. نتیجه نهایی یک رویداد تهدید طبقه‌بندی شده است که در پایگاه داده رویداد تهدید ذخیره می‌شود.

- DataDome ابزاری برای تشخیص crawlerها به‌صورت برخط:

[1] Static
[2] Admin

DataDome به هر درخواستی که به اپلیکیشن موبایل، وب‌سایت و API می‌رسد نگاه می‌کند و آن را با یک پایگاه داده‌ای از الگوهای عظیم درون حافظه مقایسه می‌کند. از طریق ترکیبی از هوش مصنوعی و یادگیری ماشین، نرم افزار تشخیص web scraping در کمتر از 2 میلی ثانیه تصمیم می‌گیرد که آیا درخواست از یک انسان است یا یک ربات.

- User Agent detection – Hello, my Name is Googlebot:

هنگامی که در حال مرور وب هستیم، ممکن است گاهی اوقات احساس ناشناس بودن کنیم. با این حال مرورگر ما هرگز این کار را نمی‌کند. هر درخواستی که ارائه می‌کند باید با نام خودش امضا شود که به آن User Agent گفته می‌شود. به عنوان مثال؛ .Chrome/67.0.3396.79 Safari/537.36

ربات‌ها همچنین دارای user agent منحصر به فردی هستند، به عنوان مثال، نام زیر متعلق به نسخه دسکتاپ Googlebot است:

Googlebot: Mozilla/5.0(compatible;Googlebot/2.1;
+http://www.google.com/bot.html)

- آنالیز لاگ‌های سرور:

این فایل‌ها ذخیره‌ای از تمام درخواست‌های ارائه شده به سرور، از جمله بازدیدکنندگان و موتورهای جستجو و سایر ربات‌ها هستند. اگر به دلایلی، بخشی از سایت ایندکس نشده باشد، زیرا توسط robots.txt مسدود شده است، اما در گزارش‌های خود، می‌توانیم بازدیدهایی از آن قسمت را مشاهده کنیم که توسط یک scraper بوجود آمده‌است که به robots.txt اهمیتی نمی‌دهد. از کجا بفهمیم که این ربات یک GoogleBot است یا یک مزاحم؟

اگر بخواهیم منبع درخواست را تشخیص دهیم، باید آدرس IP را که درخواست از آن ارسال شده است بررسی شود.

دروغ گفتن در مورد آن سخت است. می توان از یک سرور پروکسی DNS استفاده کرد و IP واقعی را مخفی کرد، اما IP پروکسی را آشکار می‌کند.

دو روش برای تایید IP وجود دارد:

برخی از موتورهای جستجو لیست یا محدوده IP را ارائه می دهند. می توانیم crawler را با تطبیق IP آن با لیست ارائه شده تأیید کنیم:

IP Lists and Ranges

برخی از crawlerهای موتورهای جستجو لیستی از آدرس IPهای ایستا و یا محدوده آدرس را مشخص می‌کنند. مزیت مقایسه IPهای crawler و این لیست، این است که می‌توان آن را به صورت اتوماتیک انجام داد بخصوص اگر چک‌کردن فضای بزرگی داشته باشد. متاسفانه ممکن است که آدرس IP در آینده تغییر پیدا کند که در ان صورت این روش مناسب نخواهد بود.

برخی از موتورهای جستجو که لیست IPها را فراهم می‌کنند:

- Google,
- Bing,
- DuckDuckGo.

می توانیم برای اتصال آدرس IP به نام دامنه، یک جستجوی DNS انجام دهیم:

برای ربات هایی که لیست IP رسمی ارائه نمی دهند، باید یک جستجوی DNS انجام دهیم تا منشاء آنها را بررسی کنیم. این روش همچنین در صورت تغییر آدرس های IP در آینده ضروری است.

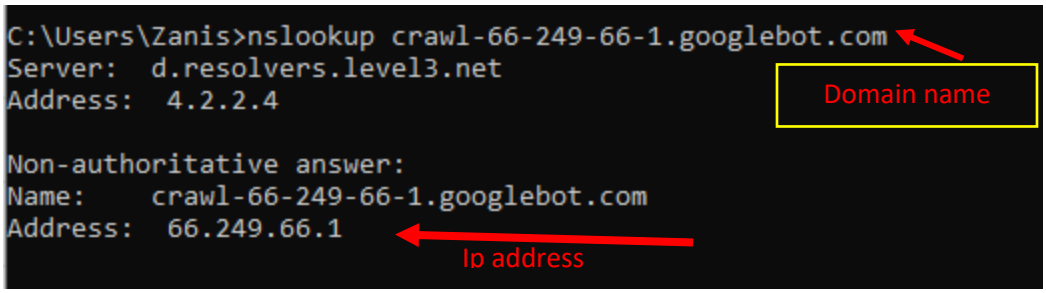
جستجوی DNS روشی برای اتصال دامنه به آدرس IP است. باید با یک آدرس IP درخواست شروع کنیم، و سپس سعی کنیم که دامنه مبدا آن را تعیین کنیم.

اولین مرحله در این فرآیند، جستجوی معکوس DNS نامیده می‌شود که در آن از سرور بخواهیم تا خود را با نام دامنه معرفی کند.

اگر دستور nslookup را در cmd با IP درخواست، ارزیابی کنیم و نام دامنه را بخوانیم باید به دامنه صحیح ختم شود. دامنه صحیح برای Googlebot، Googlebot.com است.

جستجوی نام کافی نیست. برای اطمینان از تأیید صحت، باید ریشه نام‌ها را در انتهای نام خودش داشته باشد. به عنوان مثال، دامنه ای به نام googlebot.com.imascam.se قطعاً به یک Googlebot معتبر تعلق ندارد. برای اطمینان می‌توان یک تغییر مسیر از سرور scam خود به سرور معتبر Googlebot راه اندازی کنیم. در این صورت، اگر از سرور نام آن را بپرسید، دامنه Googlebot مناسب را دریافت خواهیم کرد.

می توان این کار را با استفاده از دستورات زیر انجام دهیم.



اگر آدرس IP بالا با آدرس IP درخواست یکی باشد مشکلی وجود ندارد.

لیستی از محبوب‌ترین دامنه‌های crawlerها:

Service Name	Domain Name
Baidu	*.crawl.baidu.com
Baidu	*.crawl.baidu.jp
Bing	*.search.msn.com
Googlebot	*.google.com
Googlebot	*.googlebot.com
Yahoo	*.crawl.yahoo.net

whitelist:

محدوده IP منتشر شده ممکن است در آینده تغییر کند. چنین لیستی مطمئناً در برخی از تنظیمات سرور باقی خواهد ماند و آنها را در برابر فریب در آینده آسیب‌پذیر می‌کند.

نباید از روش جستجو برای هر درخواستی استفاده کنیم این کار زمان تا اولین بایت (TTFB) را از بین می‌برد و در نهایت سرعت وب سایت را کاهش می‌دهد. بنابراین باید یک لیست سفید موقت IP ایجاد کنیم.

ایده اصلی این است که وقتی درخواستی از user agent، Googlebot دریافت می‌شود، ابتدا لیست سفید بررسی کند، اگر در لیست باشد پس یک Googlebot معتبر است.

در مواردی که درخواست از یک آدرس IP می‌آید که در لیست سفید نیست، با nslookup چک می‌کنیم که اگر آدرس مثبت تأیید شود، وارد لیست سفید می‌شود.

لیست سفید موقتی است. باید به صورت دوره ای تمام آدرس‌های IP را حذف یا دوباره بررسی شوند. اگر درخواست‌های نادرست که دریافت می‌کنیم زیاد باشند بهتر است یک لیست سیاه هم بسازیم تا بدون انجام جستجوی DNS، چنین درخواست‌هایی را رد کنیم.

Web crawlerها کارهای شگفت انگیز زیادی برای ما انجام می‌دهند. آنها به عنوان کتابداران اینترنت عمل می‌کنند و گردش کار را خودکار می‌کنند. اما گاهی اوقات می‌توانند برای یک کسب و کار مشکل ایجاد کنند. هنگام استخراج داده‌ها از وب‌سایت‌ها، crawlerها می‌توانند آمار وب شرکت را منحرف کنند و باعث کندشدن (استفاده از پهنای باند) سایت و حتی خرابی آن شوند. ربات‌های crawler وب نیز برای تصاحب حساب‌ها و کلاهبرداری ترک استفاده می‌شوند. و به همین دلیل است که به محافظت anti-crawler نیاز است. روش‌های زیر تعدادی از این ضد crawlerها را بیان می‌کنند:

تشخیص ویژگی های مبتنی بر زمان:

وقتی سرور stampهای زمانی گزارش‌های یک جلسه را بررسی می‌کند، ممکن است چندین ویژگی برای تشخیص crawler به دست آورد. یک مثال این است که برخی از crawlerها ممکن است سریع‌تر از توانایی یک انسان به صفحات دسترسی پیدا کنند. یک crawler مخرب زره‌پوش ممکن است مرز بالای سرعت بازدید را از طریق دنباله و شکست کشف کند و سپس می تواند نرخ دانلود خود را زیر این آستانه تنظیم کند. با این حال، از آنجایی که crawlerهای توزیع‌شده یک مهاجم دارای الگوهای مبتنی بر زمان‌بندی مشابهی هستند، مدافعان ممکن است با تجزیه و تحلیل شباهت‌های سری زمانی هر کاربر، آنها را شناسایی کنند (Jacob et al. 2012).

تشخیص ویژگی های URL سرور:

می تواند URL های جلسه[3] یک کاربر را بررسی کند تا تصمیم بگیرد که آیا کاربر عادی است یا خیر. گزارش‌های دسترسی پیوسته که متعلق به یک کاربر است را بررسی می‌کنند. به عنوان مثال، برخی از crawlerها سعی می‌کنند URL ها را برای دسترسی آینده "حدس بزنند"، بنابراین در یک جلسه از URLهای موجود با نرخ بالا بازدید می‌کنند. مثال دیگر این است که وقتی crawlerها سعی می‌کنند از چندین بار بازدید از یک صفحه خودداری کنند، نرخ بازدید مجدد صفحات در یک جلسه پایین خواهد بود.

بررسی فیلدهای درخواست پروتکل HTTP:

[3]Session

مدیران وب سرور می‌توانند گزارش‌های سرورهای وب خود را بررسی کنند و چندین فیلد درخواست HTTP مانند ارجاع‌دهنده‌ها و کوکی‌ها را برای شناسایی درخواست‌های غیرعادی بررسی کنند. برخی از درخواست‌های خزنده‌ها این فیلدها را از دست می‌دهند در حالی که برخی از درخواست‌های دیگر در مقایسه با درخواست‌های کاربران عادی در این زمینه‌ها تفاوت‌های آشکاری دارند. یکی از فیلدهای معمول درخواست، User-Agent است. هر درخواست HTTP حاوی فیلد User-Agent است و می توانیم بگوییم که کدام نرم افزار از طرف کاربر مطابق این فیلد عمل می کند. به عنوان مثال، یکی از ربات‌های عامل کاربر گوگل «Googlebot/2.1» است، در حالی که عامل‌های کاربر عادی در بیشتر موارد نام مرورگرها هستند. اگرچه شناسایی خزنده های ساده با بررسی این فیلدها موثر است، خزنده های زره پوش می توانند این فیلدها را در درخواست های خود تغییر دهند تا از این نوع تشخیص فرار کنند.

چالش‌های CAPTCHA

برای جلوگیری از مسدودکردن یک کاربر عادی، چالش‌های CAPTCHA را به کاربران مشکوک اضافه می‌کنیم. هنگامی که یک کاربر CAPTCHA را به درستی وارد می‌کند، علامت مشکوک آن را حذف می‌کنیم. از آنجایی که انواع مختلفی از CAPTCHA وجود دارد، برای یک crawler سخت است که خود را برای شناسایی انواع CAPTCHA های پیشرفته آماده کند. در این صورت می‌توانیم crawler های واقعی را که نمی‌توانند به درستی پاسخ دهند شناسایی کنیم. از طرف دیگر، آن CAPTCHA ها می‌توانند به راحتی توسط کاربران عادی حل شوند. علاوه بر این می‌توان شمارنده دیگری را در «جدول اطلاعات کاربر» که در بخش «ایجاد جداول پایگاه داده» تنظیم کرد و این شمارنده تعداد دفعاتی را که یک کاربر CAPTCHA وارد می‌کند را ثبت می‌کند و اگر کاربر ماژول CAPTCHA را 3 بار در یک روز فعال کند مستقیماً مسدود می‌شود. با این شمارنده، حتی یک crawler می‌تواند تمام CAPTCHA های سیستم ما را تشخیص دهد، تا زمانی که ماژول تشخیص بلادرنگ کار می‌کند، در نهایت شناسایی می‌شود.

Heuristic detection

ماژول تشخیص اکتشافی تجزیه و تحلیل اولیه را بر روی ترافیک ورودی انجام می‌دهد و هدف آن کشف crawlerها بر اساس ویژگی های اصلی جریان ترافیک مانند ارجاع دهنده، کاربر-عامل و cookieهای تمام ترافیک ورودی است. علاوه بر این ویژگی‌های کلی، این ماژول همچنین یکپارچگی نشانگر URL را بررسی می‌کند، یک ویژگی جدید تشخیص اکتشافی که ما پیشنهاد کردیم. به طور خاص، پس از اینکه سرور نشانگر را از URL استخراج می کند، یکی از روش‌ها این است که می‌توان ابتدا شناسه بازدیدکننده را با اطلاعات ثبت شده در نشانگر URL مقایسه کند. اگر بازدیدکننده واقعی این صفحه همان بازدیدکننده‌ای نیست که در نشانگر URL ثبت شده

است، این ورودی گزارش را علامت‌گذاری می‌کنیم و این کاربر را به‌عنوان یک خزنده بالقوه که از پیوندهای به اشتراک‌گذاشته‌شده توسط دیگر خزنده‌ها بازدید می‌کند، علامت‌گذاری می‌کنیم. اگر کاربر در یک بازه زمانی چندین بار علامت‌گذاری شود، این کاربر را به‌عنوان یک crawler مشکوک علامت‌گذاری می‌کنیم و با یک CAPTCHA از آن درخواست می‌کنیم.

اگرچه تشخیص اکتشافی در بسیاری از سیستم‌های وب به کار گرفته شده‌است، شناسایی دقیق crawlerهای توزیع‌شده که آدرس‌های اینترنتی را برای خزیدن به اشتراک می‌گذارند هنوز یک چالش است. با ادغام نشانگر URL، ماژول تشخیص اکتشافی ما می‌تواند crawlerهای توزیع شده را با بررسی شناسه‌های کاربر در نشانگرها شناسایی کند.

تشخیص spambotها:

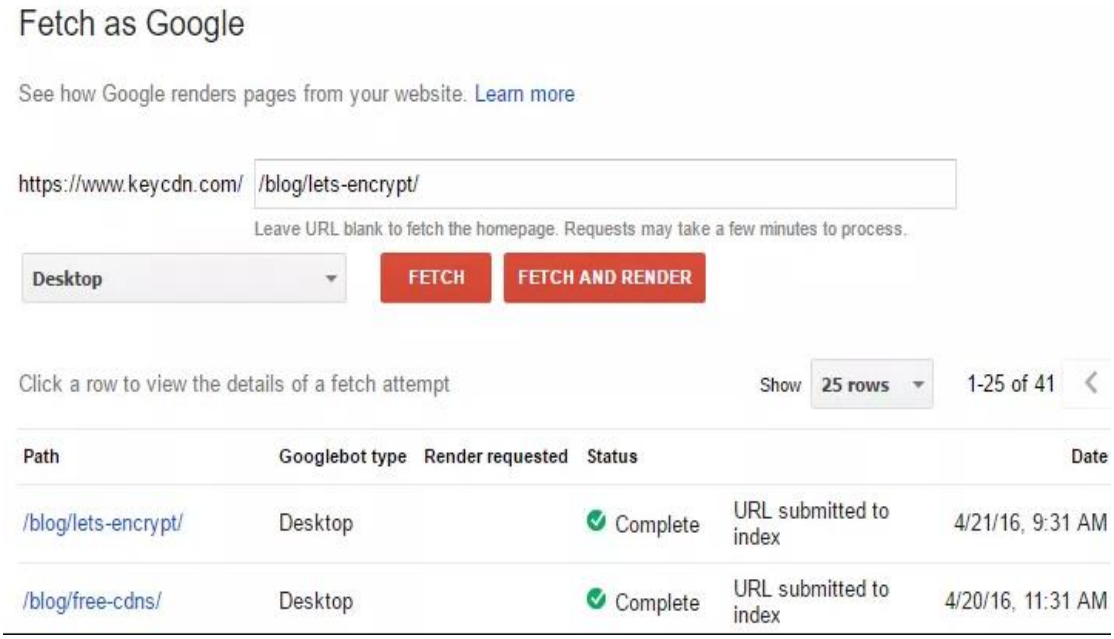
spambotها منحصر به فرد هستند و به گونه‌ای برنامه‌ریزی شده‌اند که مانند کاربران واقعی رفتار می‌کنند. بنابراین، شناسایی این spambotها به این سادگی نیست. مثلاً می‌توان به موارد زیر برای شناسایی آن‌ها اشاره کرد:

- ربات ها به طور مستقیم یا غیر مستقیم با زبان انگلیسی مرتبط هستند. آنها یاد می‌گیرند یا به گونه‌ای از قبل برنامه‌ریزی شده‌اند که می‌توان اشتباهات گرامری و املایی زیادی پیدا کرد.
- پیام ها از یک منبع غیرمنتظره خواهند بود. این پیام از طرف شخصی خواهد بود که اصلاً برای ما شناخته شده نیست، و این باید به عنوان یک پیام مشکوک در نظر گرفته شود که برای سرقت اطلاعات شخصی یا انتشار بدافزار انجام می‌شود.
- پیام بی ربط خواهد بود. ممکن است اسکرین شات ها و تصاویر مختلفی را ببینیم که به موضوع ربطی ندارد.

✓ ابزارهای crawling:

Googlebot: گوگل به عنوان بزرگترین موتور جستجوی جهان، برای فهرست کردن میلیاردها صفحه در اینترنت به crawlerهای وب متکی است. crawler Googlebot وب است که گوگل برای انجام این کار از آن استفاده می‌کند.

Googlebot دو نوع crawler است: یک desktop crawler که از شخصی که در حال مرور در رایانه است تقلید می‌کند و یک crawler تلفن همراه که عملکرد مشابه آیفون یا تلفن اندرویدی را انجام می‌دهد.



Bingbot: در سال 2010 توسط مایکروسافت ایجاد شد تا URL ها را اسکن و فهرست کند تا اطمینان حاصل شود که Bing نتایج موتور جستجوی مرتبط و به روز را برای کاربران پلتفرم ارائه می دهد.

Helium Scrapper: یک پکیج برای scraping که پشتیبانی مناسبی نیز دارد و فرمت داده به صورت XML، CSV، Excel، JSON، SQLite است. Proxy rotation دارد و می‌تواند تصاویر را و موارد خواسته‌شده دیگر را بلاک کند. (www.heliumscrapper.com)

ParseHub: یک web crawler است که داده‌ها را از وب سایت‌ها با استفاده از AJAX، JavaScript، cookies و ... جمع‌آوری می‌کند و با استفاده از یادگیری ماشین می‌تواند آنالیز کند و بخواند و داکیومنت‌های وب را به داده‌های خواسته شده تبدیل کند. می‌تواند داده‌ها را فرمت JSON و CSV تحویل دهد و همچنین روی Mac، ویندوز و لینوکس قابل انجام است.

Octoparse: یک ابزار web crawler مبتنی بر مشتری است که داده‌های وب را در صفحات گسترده قرار می‌دهد. این نرم افزار با رابط کاربر پسند نقطه و کلیک، به طور خاص برای غیر کدنویس‌ها ساخته شده است. می‌توان داده‌ها را به صورت برخط استخراج کرد.

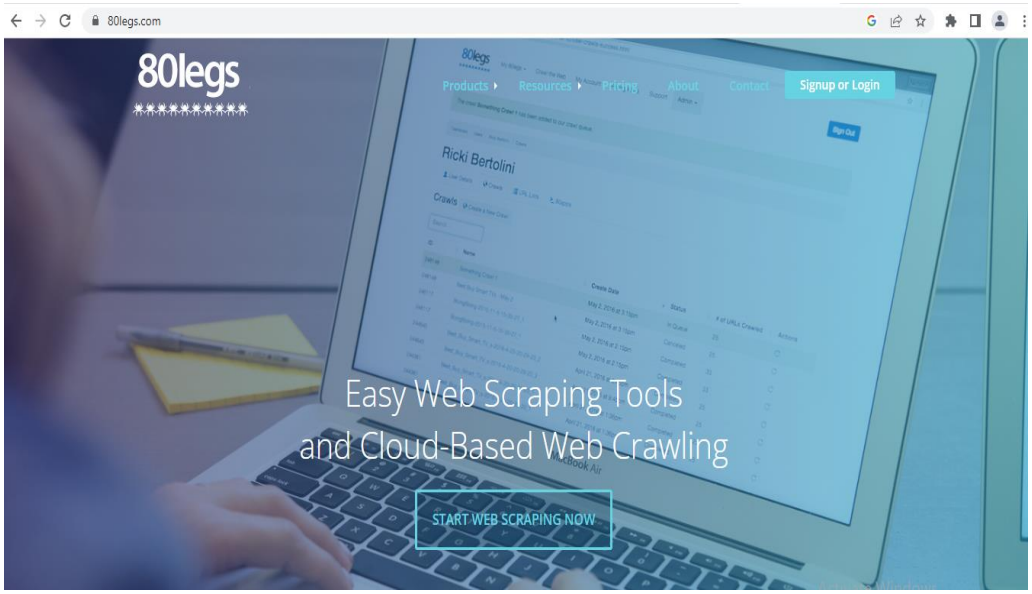
Slurp bot: نتایج جستجوی یاهو از crawler وب یاهو Slurp و crawler وب بینگ به دست آمده است، زیرا بسیاری از یاهو توسط بینگ تامین می‌شود. سایت‌ها باید به Yahoo Slurp دسترسی داشته باشند تا در نتایج جستجوی یاهو موبایل ظاهر شوند.

علاوه بر این، Slurp کارهای زیر را انجام می‌دهد:

- محتوا را از سایت‌های شریک جمع‌آوری می‌کند تا در سایت‌هایی مانند Yahoo News، Yahoo Finance و Yahoo Sports گنجانده شود. برای تأیید صحت و بهبود محتوای شخصی‌شده یاهو برای کاربران، به صفحاتی از سایت‌ها در سراسر وب دسترسی دارد.

همچنین می‌توان از extensionهایی بر روی مرورگر بهره برد همانند:

Scraper: یک افزونه کروم با ویژگی های استخراج داده محدود است اما برای انجام تحقیقات برخط مفید است. همچنین اجازه می‌دهد تا داده‌ها را به صفحات گسترده Google صادر کنیم. این ابزار برای افراد مبتدی و متخصص در نظر گرفته شده است. می توان به راحتی داده‌ها را در کلیپ بورد کپی کنیم یا با استفاده از OAuth در صفحات گسترده ذخیره کنیم. Scraper می‌تواند به طور خودکار XPathها را برای تعریف URLها برای crawling ایجاد کند. خدمات crawling همه جانبه را ارائه نمی دهد، اما اکثر مردم به هر حال نیازی به مقابله با پیکربندی‌های نامرتب ندارند.



یک نمونه web crawler

✓ متدهای crawling

رفتار یک crawler وب نتیجه ترکیبی از سیاست‌ها است:

- یک خط مشی انتخاب که صفحاتی که باید دانلود شوند را بیان می‌کند،
- یک خط مشی بازدید مجدد که بیان می‌کند چه زمانی باید تغییرات در صفحات را بررسی کرد،
- یک خط مشی ادب که نحوه جلوگیری از بارگذاری بیش از حد وب سایت‌ها را بیان می‌کند.
- یک خط مشی موازی سازی که نحوه هماهنگ‌کردن crawlerهای وب توزیع شده را بیان می‌کند.

Web crawling روند ساخت یک مجموعه از web pageهاست که از یک URL ابتدایی شروع می‌شود و به‌طور بازگشتی صفحات مربوط را با با لینک‌های دیگر پیدا می‌کنند.

گام‌ها در web crawling:

1. Crawler creates and maintain a list of URLs to be processed.

2. The crawler selects a URL from the list (based on strategy) marks it as “crawled” and it fetches the webpage from that URL. The page is processed and links and content are extracted. This processing can be as simple as just extracting links using a regular expression (regex) that matches all the occurrences of tags in html.

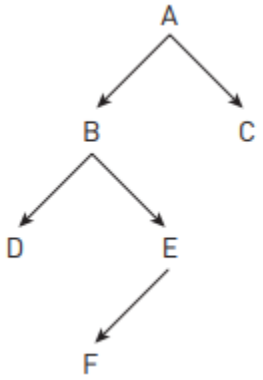
3. If the content has already been seen, it is discarded. If not, it is added to the collection of webpages to be further processed.

4. For each URL in the new set of URLs identified on the page, a verification is made to ensure that the URL has not been seen before, that the page exists, and can be crawled. If all these filters are passed the URL is added to the list of URLs at Step 1 and the crawler goes back to Step 2

Traversal strategies

از آنجایی که وب یک گراف است پس می‌توان از الگوریتم‌های پیمودن گراف استفاده کرد همانند BFS، زمانی که یک صفحه به آن بدهیم، تمام URLهایی که می‌توان از صفحه اول پیمود را جمع‌آوری می‌کنیم.

در روش دیگر می‌توان از DFS استفاده کرد به این صورت که ابتدا یک صفحه وب به آن می‌دهیم یک URL از صفحه پیدا می‌کنیم و صفحاتی که از آن URL پیدا می‌شوند را پردازش می‌کنیم. همه پردازش‌ها تا حالت بن‌بست ادامه دارند.



Breadth-first traversal order: A, B, C, D, E, F
Depth-first traversal order: A, B, D, E, F, C

Crawler Politeness

بیشتر وب سایت‌ها یک قانون مشخص دارند که چه crawlerهایی می‌توانند سایت را بپیمایند و چه بخش‌هایی می‌تواند پیمایش شود. ۲ روش برای گذاشتن قانون وجود دارد: ۱. Robots.txt فایلی که در root وب سایت وجود دارد مثلاً www.cnn.com/robots.txt. ۲. از طریق meta tags که در HTML صفحات موجود است. <meta naem="robots" content="index.nofollow"> که بیان می‌کند این صفحه می‌تواند crawl شود اما لینک‌های صفحه را نمی‌توان دنبال کرد.

محدود کردن لینک‌های دنبال شده

یک crawler ممکن است فقط بخواهد صفحات HTML را جستجو کند و از سایر انواع MIME اجتناب کند. به منظور درخواست فقط منابع HTML، یک خزنده ممکن است یک درخواست HTTP HEAD برای تعیین نوع MIME یک منبع وب قبل از درخواست کل منبع با درخواست GET ارائه دهد. برای جلوگیری از درخواست‌های متعدد HEAD، یک خزنده ممکن است URL را بررسی کند و فقط در صورتی درخواست منبع کند که URL با کاراکترهای خاصی مانند html، htm.، asp.، aspx.، php، jsp.، jspx. یا / ختم شود. . این استراتژی ممکن است باعث شود که بسیاری از منابع وب HTML ناخواسته نادیده گرفته شوند.

برخی از crawler ها همچنین ممکن است از درخواست منابعی که دارای "?" هستند اجتناب کنند. در آنها (به صورت پویا تولید می‌شوند) به منظور جلوگیری از spider trap که ممکن است باعث شود crawler تعداد بی نهایت URL را از یک وب سایت دانلود کند. اگر سایت از بازنویسی URL برای ساده سازی URL های خود استفاده کند، این استراتژی غیرقابل اعتماد است.

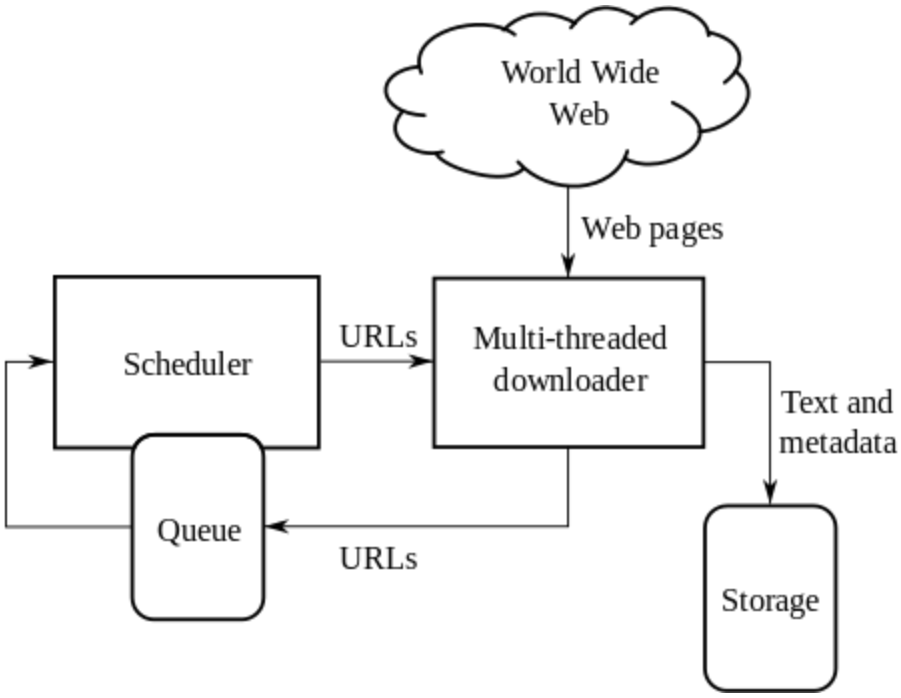
URL Normalization

crawler ها معمولاً نوعی از عادی سازی URL را انجام می دهند تا از برخی از crawlerها قصد دارند تا حد امکان منابع را از یک وب سایت خاص بارگیری/آپلود کنند. بنابراین crawler صعودی مسیر معرفی شد که به هر مسیری در هر URL که قصد crawling آن را دارد صعود می‌کند. به عنوان مثال، هنگامی که یک URL seed http://llama.org/hamster/monkey/page.html به ما داده می‌شود، سعی می‌کند /hamster/monkey/، /hamster، و "/ "را crawl کند.

crawlکردن یک منبع یکسان بیش از یک بار باید جلوگیری شود. اصطلاح URL normalization به فرآیند اصلاح و استانداردسازی URL به شیوه‌ای ثابت اشاره دارد. انواع مختلفی از عادی سازی وجود دارد که ممکن است انجام شود، از جمله تبدیل URL ها به حروف کوچک، حذف "." و بخش های ".."، و اضافه کردن "/" انتهایی به جزء مسیر غیر خالی.

Path-ascending crawling

هر crawler باید علاوه بر استراتژی‌هایی که بالا گفته شد، یک معماری نیز داشته باشد:

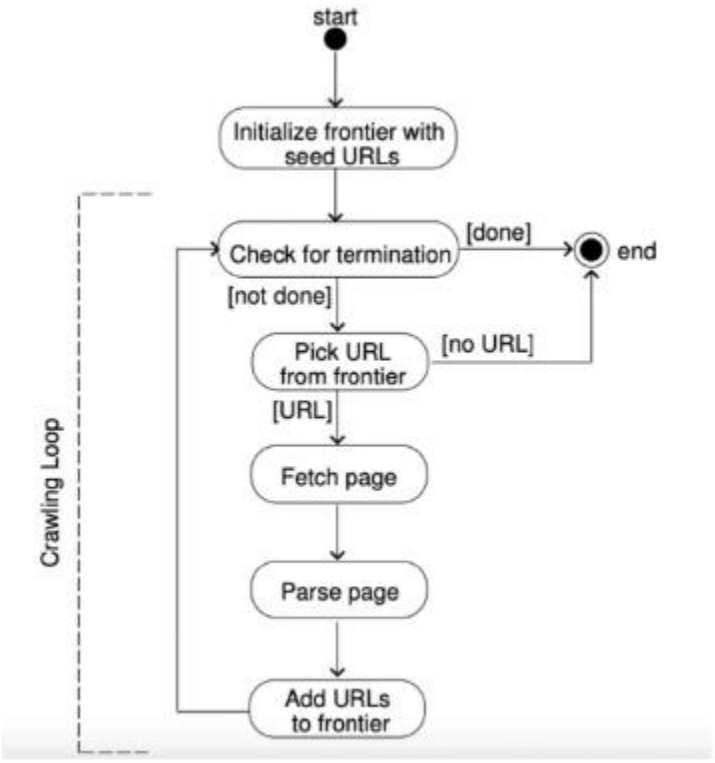


در هر معماری موارد زیر بیان می‌شود:

- URL seed : که به عنوان URL initiator نیز شناخته می‌شود، ورودی‌ای است که crawlerهای وب برای شروع فرآیندهای نمایه‌سازی و خزیدن از آن استفاده می‌کنند.
- مرز URL: مرز خزیدن شامل خطمشی‌ها و قوانینی است که یک خزنده وب باید هنگام بازدید از وب‌سایت‌ها رعایت کند. Crawl وب بر اساس خط مشی های مرزی تصمیم می‌گیرد از کدام صفحات بازدید کند.
- Crawl frontier اولویت‌های متفاوتی را به هر URL اختصاص می‌دهد (به عنوان مثال نشانی‌های اینترنتی با اولویت بالا و با اولویت پایین) تا به خزنده اطلاع دهد که از کدام صفحات بعدی بازدید کند و چند بار باید از صفحه بازدید شود.
- واکشی و رندر URL(ها): مرز URL به گیرنده اطلاع می دهد که کدام URL را باید برای بازیابی اطلاعات مورد نیاز از منبع خود درخواست کند. سپس crawler وب URL های واکشی شده را به منظور نمایش محتوای وب در صفحه مشتری ارائه می‌دهد.
- پردازش محتوا: هنگامی که محتوای صفحه وب crawl شده ارائه شد، دانلود شده و برای استفاده بیشتر در فضای ذخیره سازی ذخیره می شود. محتوای دانلود شده می‌تواند حاوی صفحات تکراری، بدافزار و غیره باشد.

فیلتر کردن URL: فیلتر کردن URL فرآیند حذف یا مسدود کردن URL های خاص از بارگیری در دستگاه کاربر به دلایل خاص است. هنگامی که فیلتر URL همه URL های موجود در فضای ذخیره سازی را بررسی می‌کند، URL های مجاز را به دانلود کننده URL ارسال می‌کند.

بارگیری URL: بارگیری URL تعیین می‌کند که آیا یک crawler وب یک URL را دیده است یا خیر. اگر دانلودکننده URL با URL هایی روبرو شود که هنوز دیده نشده‌اند، آنها را به مرز URL ارسال می‌کند تا دیده شوند.



مراجع

https://www.onely.com/blog/detect-verify-crawlers/

https://en.wikipedia.org/wiki/Web_crawler

https://netcorecloud.com/tutorials/spambots-complete-guide/

https://datadome.co/bot-protection-online-fraud-prevention/protect-your-business-customers-against-web-scraping/

https://www.hillstonenet.com/blog/a-hybrid-approach-to-detect-malicious-web-crawlers/

https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0023-1

https://cheq.ai/blog/what-are-the-different-types-of-anti-crawler-protection/#:~:text=When%20extracting%20data%20from%20websites,you%20need%20anti%2Dcrawler%20protection.

https://research.aimultiple.com/web-crawler/

https://www.octoparse.com/blog/top-20-web-crawling-tools-for-extracting-web-data#

https://www.keycdn.com/blog/web-crawlers

https://kinsta.com/blog/crawler-list/