SDK категоризации доменов

- Модуль категоризации доменов для C, C++
- Модуль категоризации доменов для Python

Модуль категоризации доменов для C, C++

Эта библиотека позволяет решать следующие задачи:

1. Находить категории заданного URL с помощью поиска в локальной базе данных, находящейся на конечном устройстве пользователя (ПК, маршрутизатор, встраиваемое устройство). В этом случае подключение к интернету не нужно и никаких сетевых запросов не выполняется. Эта база данных называется локальной базой данных.
2. Находить категории заданного URL с помощью обращения по протоколу HTTPS к облаку компании SkyDNS. Так как выполняется сетевой HTTPS-запрос, для реализации этого функционала требуется, во-первых, работоспособное подключение к интернету, и, во-вторых, наличие логина и пароля к соответствующему сервису SkyDNS. Но в этом случае наличие на компьютере пользователя локальной опорной базы данных не нужно.
3. Сохранять в локальном дисковом кэше категории URL, полученные из облака SkyDNS для последующего более быстрого доступа к ним. Данные, сохранённые в дисковом кэше, переживают перезагрузку компьютера и перезапуск приложения, использующего этот кэш.
4. Кэшировать в оперативной памяти компьютера полученные категории данного URL для ускорения последующего их поиска. Размеры кэша задаются при сборке решения.
5. Получать с сайта SkyDNS и устанавливать регулярные обновления к локальной базе данных тематических категорий сайтов. На текущий момент доступно только скачивание полностью базы данных (все категории ~1.6Гб, возможны более мелкие кастомные сборки).

Решение реализовано на языке программирования Си и предоставляет своим пользователям интерфейс в виде функций на этом языке программирования. Решение может быть также интегрировано в проект на языке Python (имеются примеры интеграции в Си и Python решения).

Можно выдать категории на английском и русском языках (задаётся на этапе компиляции библиотеки).

Интеграция SDK в конечный продукт

SDK достаточно просто интегрировать в конечный программный продукт, разрабатываемый на языке программирования Си или Си++. Также существует возможность интегрировать его в конечное решение на Python.

Для использования библиотеки необходимо получить исходники библиотеки (тут репозиторий), сконфигурировать библиотеку, собрать и установить.

Конфигурирование библиотеки url2cat

Для конфигурирования и сборки библиотеки используется утилита **cmake** (версии 3.15 или выше).

Также понадобятся библиотеки (для Ubuntu 18+): **openssl-1.1.1**, **sqlite3**.

1. Зайдите в каталог с исходниками библиотеки
2. Сконфигурируйте библиотеку, используя команду:

```
cmake -S . -B build_release -DCMAKE_BUILD_TYPE=Release
> -DURL2CAT_SERVER=yes -DURL2CAT_DATABASE=yes -DURL2CAT_LOCALE=ru
> -DURL2CAT_LIBRARY=static
```

Изменяя следующие переменные можно настроить функции библиотеки:

- `URL2CAT_SERVER` - запросы к серверу SkyDNS (yes, no);
- `URL2CAT_DATABASE` - запросы к локальной БД (yes , no);
- `URL2CAT_LOCALE` - формат отображения категорий (ru, en);
- `URL2CAT_LIBRARY` - тип создаваемой библиотеки (static, shared);
- `URL2CAT_MAX_NUMBER_CATEGORY` - количество определяемых категорий (по умолчанию 5);
- `URL2CAT_HASH_TYPE` - тип хэша (MD5, SHA256, SHA512, по умолчанию MD5);
- `URL2CAT_HASH_LEN` - используемая длина хэша (full (хэш не обрезается перед поиском по базе), half (берётся 8 первых Байт хэша), по умолчанию half).

3. Соберите библиотеку, используя команду:

```
cmake --build build_release
```

4. Скопируйте следующие файлы библиотеки в свой проект:

- `build_release/lib/liburl2cat.a` или `build_release/lib/liburl2cat.so`;
- `include/url2cat.h`.

Использование библиотеки в проекте

Вначале использования библиотеки нужно ее инициализировать с использованием структуры `s_url2cat_setting`. Структура имеет следующие поля:

- `cache_size` - размер кэша в байтах (если установлен 0 кэш не используется);

- `db_name` - имя базы данных;
- `db_download_scheme` - протокол для обновления БД;
- `db_download_host` - хост обновления БД;
- `db_download_port` - порт обновления БД;
- `db_download_path` - путь обновления БД;
- `db_download_user` - логин для обновления БД;
- `db_download_password` - пароль для обновления БД;
- `server_scheme` - протокол подключения к серверу SkyDNS;
- `server_host` - хост подключения к серверу SkyDNS;
- `server_port` - порт подключения к серверу SkyDNS;
- `server_path` - путь подключения к серверу SkyDNS;
- `server_user` - логин подключения к серверу SkyDNS;
- `server_password` - пароль подключения к серверу SkyDNS.

Для инициализации используется функция: `url2cat_init(s_url2cat_setting * setting)`

Для получения категории используется функция: `url2cat_category(char * url, size_t len_url, s_url2cat_category * * category, size_t * number_category)`

где `s_url2cat_category` структура, имеющая следующие поля:

- `type` - номер категории;
- `type_name` - название категории.

После использования библиотеки нужно ее деинициализировать, используя функцию: `url2cat_deinit()`

При использовании библиотеки можно обновить БД, используя функцию: `url2cat_database_update(s_url2cat_setting * setting)`

При использовании библиотеки можно отправить домен на перекатегоризацию, используя функцию: `url2cat_recategory(char * url, size_t url_size, char * category_name, size_t category_name_size)`

Примечание. В каталоге `example/` исходников библиотеки url2cat есть примеры интеграции в простые решения на Си и Python.

Приложения

Приложение 1. Справочник категорий

В настоящее время поддерживаются такие категории:

id	Категория
2	Неизвестные сайты

3	Malware
4	Phishing & Typosquatting
5	Онлайн-реклама и баннеры
6	Наркотики
7	Грубость, матершина, непристойность
8	Плагиат и рефераты
9	Запаркованные домены
10	Агрессия, расизм, терроризм
11	Прокси и анонимайзеры
12	Botnets & C2C
13	Сайты для взрослых
14	Алкоголь и табак
15	Знакомства
16	Порнография и секс
17	Астрология
18	Казино, лотереи, тотализаторы
20	Торренты и P2P-сети
21	Файловые архивы
22	Фильмы и видео онлайн
23	Радио и музыка онлайн
24	Фотогалереи
25	Content Delivery Networks
26	Чаты и мессенджеры
27	Форумы
28	Компьютерные игры
29	Социальные сети
30	Досуг и развлечения
32	Автомобили и транспорт
33	Блоги и персональные сайты
34	Корпоративные сайты
35	Интернет-магазины
36	Образование и учебные учреждения

37	Финансы и финансовые учреждения
38	Правительство
39	Здоровье и здравоохранение
40	Юмор
41	Работа и найм
42	Войска и вооружения
43	Политика, общество, закон
44	Новости и СМИ
45	Некоммерческие организации
46	Порталы
47	Религия и атеизм
48	Поисковые системы
49	Компьютеры и Интернет
50	Спорт
51	Наука и технологии
52	Туризм
53	Дом, семья, хобби
54	Торговля и покупки
55	Искусство
56	Веб-почта
57	Недвижимость
58	Доски объявлений
59	Бизнес, экономика, маркетинг
60	Сайты для детей
63	Трекинг и Аналитика
66	Cryptojacking
67	Интернет-библиотеки
69	Аниме
70	DGA
71	Ransomware
72	ИИ Чат-боты
100	Без контента

Для зарубежного рынка добавлены категории:

id	Категория
19	Child Sexual Abuse (IWF)
31	German Youth Protection
65	Child Sexual Abuse (Arachnid)

Приложение 2. Канонический вид домена (или URL’а)

Один URL может быть представлен не в одной уникальной форме, для представления одного ресурса, например, punny-код и кириллица в домене, висячий слеш, и т.д.

Было бы нерационально включать в базу категоризации все эти различные возможности, как минимум для некоторых интеграций базы, например, в небольшие бюджетные роутеры.

Потому, наиболее рациональной кажется схема, по которой URL’ы ресурсов в базе хранятся в каноническом виде, а значит, перед запросом списка категорий, URL требуется каноникализировать, то есть получить каноническое представление URL’а.

Единый указатель ресурса URL состоит из нескольких частей, но нас интересует только два из них:

- Доменное имя (`domain`);
- Путь (`path`).

Рассмотрим на примере случайного URL:

`directory.google.com/example/test.php?key=value&one=1`

где домен это:

`directory.google.com`

а это путь URL:

`/example/test.php?key=value&one=1`

Поскольку URL указывает на ресурс, он может быть записан в различных формах и различными способами. Для прямого поиска в базе надо привести все разнообразные формы URL указывающие на один ресурс к одному каноничному виду. Это очень важно для получения правильной категоризации запрашиваемого URL.

Мы используем стандартный вариант каноникализации URL из проекта Google Safe Browsing с версией API больше 2. В этом проекте описаны техники для каноникализации идентификатора ресурса, с примерами и алгоритмами которого можно ознакомится на

странице.

Примеры каноникализации:

Исходный URL	Каноничный URL
`http://evil.com/foo-bar-baz`	`http://evil.com/foo`
`http://host/%25%32%35`	`http://host/%25`
`http://evil.com/foo-bar-baz`	`http://evil.com/foo`
`http://test.com/kek/../`	`http://test.com/`

Каноникализация в контексте библиотеки url2cat

Описанным выше образом мы получим каноничную форму URL. Но из-за природы URL нельзя положиться на один URL. Для поиска наилучшего соответствия ему в базе требуется сформировать производные URL, путем поочередного отбрасывания частей первичного URL с левого и правого края.

Рассмотрим на примере случайного URL:

`directory.google.com/example/test.php?key=value&one=1`

Производные URL будут следующими:

`directory.google.com/example/test.php`

`directory.google.com/example/`

`directory.google.com/`

`google.com/example/test.php?key=value&one=1`

`google.com/example/test.php`

`google.com/example/`

`google.com/`

Получившиеся URL требуется проверить по базе.

Приложение 3. Локальная база

Один из способов получения категорий URL состоит в использовании локальной базы данных, которую предоставляет своим клиентам компания SkyDNS. Эта база данных ежедневно обновляется и потому всегда содержит актуальную информацию о сайтах в интернете. В настоящее время эта база данных поставляется в формате sqlite3.

Обновление базы

База распространяется в двух видах: файл-бинарник (sqlite3-база) и патч (набор sql-конструкций для приведения имеющейся sqlite3-базы в актуальное состояние).

Источник для получения базы в бинарном виде:
`https://url2cat.skydns.ru/pubfilter/grandbase.db`.

Источник для получения базы в виде патчей:
`https://url2cat.skydns.ru/api/v1/update/<user_version>`, где <user_version> это PRAGMA параметр текущей базы. Внутри получаемого патча, первой строкой указывается новая версия этого параметра и если она совпадает с запрошенной, обновление не требуется.

Посмотреть текущую версию базы (PRAGMA-параметр user_version) можно командой (Linux):

```
xxd -l 4 -s 60 grandbase.db
```

Процесс обновления представляет из себя GET-запрос к указанному источнику с параметрами BASIC-авторизации.

Описание базы

Таблица result

Таблица «**result**» содержит хешированные записи URL и их категоризацию.

Схема базы данных:

Название	Данные
domain_hash	хэш от домена
path_hash	хэш от пути
cat_id	список категорий

С первичным ключом по полям `domain_hash`, `path_hash`.

Данные в поле `cat_id` хранятся в виде blob массива, для стандартизации перечисляемого типа, где каждая категория хранится в виде unsigned short.

Таблица cat

Таблица «**cat**» содержит список записей с названиями и идентификаторами категорий. В зависимости от требований клиента база SkyDNS Octo может поставляться с различным числом категорий с более детальной категоризацией или уникальными именами категорий.

Схема базы данных:

Название	Данные

locale	идентификатор локализации
cat_id	идентификатор категории
name	название категории

С первичным составным ключом `locale`, `cat_id`.

Идентификаторы из поля `cat_id` таблицы «**result**» являются внешним ключом на эту таблицу. Поле `locale` содержит идентификатор локализации языка, на котором записано название категории в поле `name`. Поле `cat_id` содержит числовой идентификатор категории, данные `cat_id` не являются последовательными. Поле `name` содержит локализованные названия категорий.

Присутствие в таблице записей на английском языке в любой локализации не считается ошибкой, а указывает на переведенные категории.

Приложение 4. API категоризации

Другой способ получения категорий URL состоит в использовании API категоризации. Получение списка категорий осуществляется GET-запросом на URL одного из серверов авторизации с применением BASIC-авторизации.

Сервера API категоризации

SkyDNS:

`https://z.api.skydns.ru/` - анонимный бесплатный сервер

`https://x.api.skydns.ru/` - сервер с авторизацией

На бесплатных серверах ограничение в 10 запросов в минуту.

Запрос на категоризацию

Формат запроса, на примере бесплатного сервера SkyDNS:

`https://z.api.skydns.ru/domain/pornhub.com`

Ответом сервера будет json вида:

```
{
    "category": [13, 16, 64],
    "bad": true,
    "category_name": [
        "Сайты для взрослых", "Порнография и секс", "Реестр запрещенных сайтов"
    ]
}
```

Модуль категоризации доменов для Python

Модуль skydns_url2cat на языке программирования Python

Модуль skydns_url2cat предоставляет функционал для доступа к базе данных с внутренней каноникализацией запросов.

Для работы с модулем ? ??????? pip установите его на каждую целевую систему. После установки будут доступны команды:

- skydns-url2cat - для разовой проверки урла по заданной базе
- skydns-url2cat-update - для обновления базы через сервера SkyDNS

Программный интерфейс

Пример использования модуля:

```
>>> import skydns_url2cat

>>> skydns_url2cat.init(path_to_skydns_db)
True
>>> skydns_url2cat.lookup('google.com')
('google.com/', (48,))
```

> База sqlite работает нестабильно на сетевых файловых системах.

> Даже для чтения базы требуются права на запись, это особенность используемого режима работы базы.

Перечитывание базы

При создании сервиса проверки ресурсов по урловой базе выгоднее держать открытое соединение на блок операций, а не создавать подключение к базе на каждый запрос. Но это создает проблемы для обновления базы, для обхода этого включайте перечитывание базы.

Автоматическое перечитывание

Для включения режима автоматического перечитывания базы, после инициализации модуля, вызовите метод *auto_refresh*

```
>>> import skydns_url2cat

>>> skydns_url2cat.init(path_to_skydns_db)
True
>>> skydns_url2cat.auto_refresh()
>>> skydns_url2cat.lookup('google.com')
('google.com/', (48,))
```

После этого модуль будет самостоятельно отслеживать изменение файла и будет производить перечитывание самостоятельно.

Обновление

Для обновления запустите команду

```
$ skydns-url2cat-update <path_to_db> <username> <password>
```

При этом произойдет проверка доступности обновлений, скачивание и применение обновления к указанной базе.

Настроить периодический запуск скрипта можно при помощи cron.

Модуль skydns_url2cat

exception `skydns_url2cat.reader.` `DatabaseError`

class `skydns_url2cat.reader.` `Reader`

Класс для чтения содержимого базы

`configure`(*path username=None*, *password=None*)

Настраивает объект, задает путь до базы данных и авторизационные данные

`lookup`(*url*)

Функция для получения категории урла

Возвращает кортеж из 2 элементов:

1. Урл, который найден в базе
2. Список id категорий

Если урл не найден или невозможно разобрать, первый элемент будет None.

Exception:	UrlParseError
Параметры:	**url** – str
Результат:	tuple

`lookup_with_names` (*url*)

Функция для получения категории урла. В отличие от метода lookup возвращает список имен категорий.

`category` (*i*)

Возвращает имя категории

`categories` ()

Возвращает доступные категории

`languages` ()

Возвращает доступные локализации

`category_for_lang` (*lang*, *i*)

Возвращает имя категории в заданной локализации

`categories_for_lang` (*lang*)

Возвращает категории в заданной локализации

`rollback` ()

Инициализировать автоматические перечитывание