Программа для ЭВМ
Vul-Search

Руководство пользователя

2023

Оглавление

1. Общие сведения *2*

1.1. Назначение 2

1.2. Системные требования 2

1.3. Общие сведения о Сервисе 2

2. Описание графического интерфейса *2*

2.1. Общее описание интерфейса 2

2.2. Описание структуры базы данных 4

2.3. Использование строки поиска 8

2.4. Использование предопределенных фильтров 5

3. Описание REST-API *5*

3.1. API-ключ 5

3.2. Поиск в базе данных 6

3.3. Поиск общедоступных эксплойтов 9

3.4. Получение уязвимостей/эксплойтов по названию и версии программного обеспечения 11

3.5. Получение уязвимостей по продукту CPE и версии 12

3.6. Получение ссылок на уязвимости 13

3.7. Загрузить всю коллекцию баз данных 14

3.8. Аудит хостов Windows на предмет установленных KB 14

3.9. Аудит установленных KB's и программного обеспечения 15

3.10. Получение данных о иерархии Windows KB 16

3.11. Получение списка обновлений Windows KB со ссылками для загрузки 16

3.12. Аудит узлов Linux на наличие уязвимостей (на основе RPM/DEB) 17

3.13. Загрузить данные об уязвимостях заданного дистрибутива Linux 18

1. Общие сведения

1.1. Назначение

Программный продукт – сервис Vul-Search (далее Сервис) предназначен для повышения уровня защищенности пользователей и принадлежащих им информационных систем от угроз информационной безопасности, а также автоматизации мониторинга состояния защищенности информационных систем пользователей.

1.2. Системные требования

Минимальные требования для работы с Сервисом:

- Минимальные технические требования
  - 1 CPU – Intel Pentium4 и выше или AMD Athlon II и выше;
  - ОЗУ - 2 Гбайт и выше (для 64-разрядных систем — 4 Гбайт);
  - 400 МБ дискового пространства;
  - Сетевое подключение 10 Мбит.
- Операционные системы:
  - MS Windows 7, 8, 8.1, 10, 11;
  - Linux — (любой современный дистрибутив, например, AstraLinux, Ubuntu, Debian);
  - MacOS X: Macbook Pro (2011г. и позднее), Macbook Air (2012 г. и позднее).
- Приложения:
  - Интернет-браузеры: Яндекс.Браузер, Edge, Mozilla FireFox, Google Chrome, Safari актуальных версий.

1.3. Общие сведения о Сервисе

Сервис предназначен для обеспечения возможности получения актуальной информации об опубликованных уязвимостях используемого программного обеспечения, а также для обеспечения возможности контроля наличия уязвимостей программного обеспечения, использующегося внутри информационных систем пользователя.

2. Описание графического интерфейса

2.1. Общее описание интерфейса

Основное назначение графического интерфейса пользователя – поиск информации об уязвимостях программного обеспечения. Эта функция может быть выполнена двумя способами – с помощью поисковой строки и предустановленных фильтров.

Для обращения к поисковой строке необходимо перейти по ссылке https://vul-search.ru, ввести свой запрос в поле Поиск (Рисунок 1) и нажать кнопку Искать.

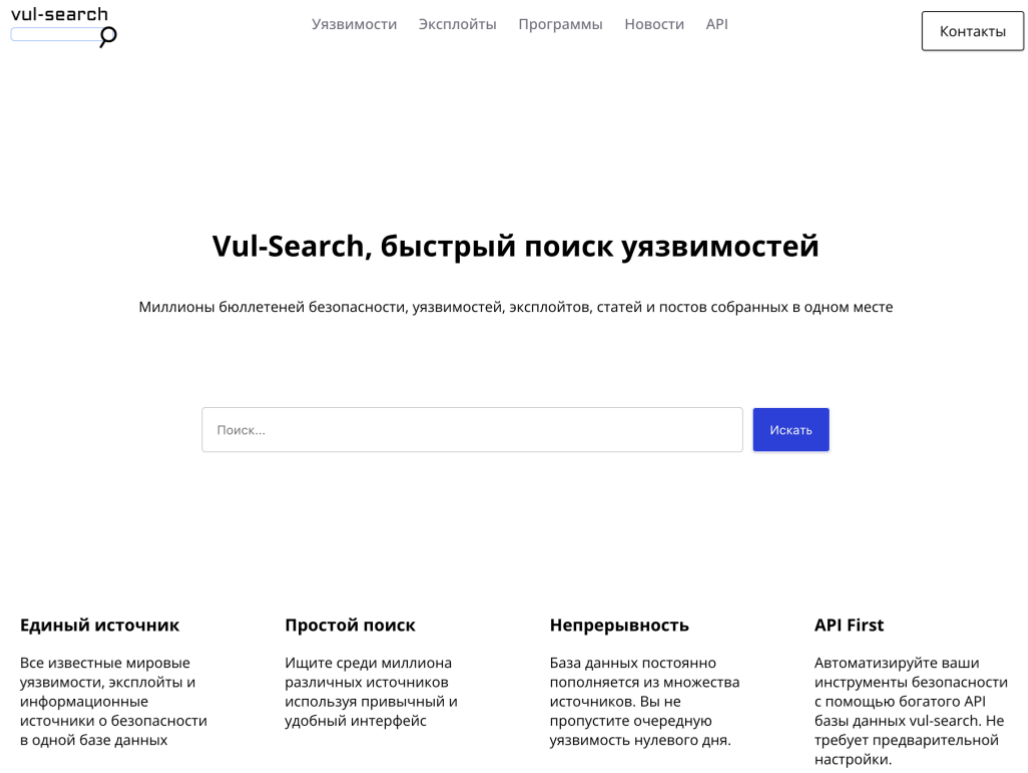

Рисунок 1

Для использования предопределенных фильтров (Рисунок 2) перейдите по одной из ссылок вверху страницы:

- Уязвимости;
- Эксплойты;
- Программы.

Или введите запрос в поисковую строку.

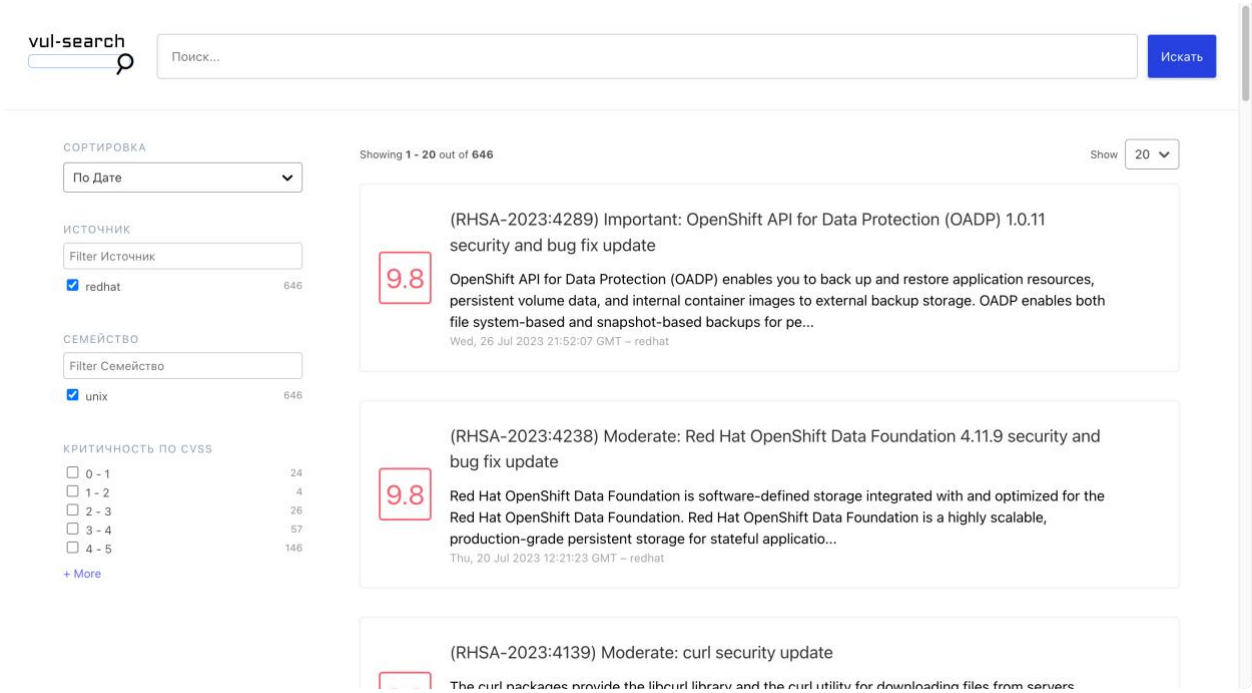

Рисунок 2

2.2. Описание структуры базы данных

Для эффективного поиска по базе данных уязвимостей пользователю следует иметь представление о структуре базы данных уязвимостей.

База данных имеет документно-ориентированную структуру. Каждый документ представляет собой набор именованных полей строкового, целочисленного, логического или составного типа. Причем поля с одинаковыми именами имеет одинаковый тип для всех документов в базе данных.

Все документы в базе объединяются в коллекции по признаку источника данных. Коллекции документов объединяются в семейства по признаку общих наборов полей.

Каждый документ включает следующий набор полей:

- id – уникальный идентификатор документа.
- type – название коллекции документов.
- bulletinFamily – название семейства документов.
- title – название документа.
- description – текстовое содержимое документа.
- published – дата публикации документа.
- modified – дата последнего изменения документа.
- href – ссылка на источник документа.

- references – список внешних ресурсов, на которые ссылается документ.
- cvelist – список идентификаторов CVE[1], упоминаемых в документе.

2.3. Использование предопределенных фильтров

Для упрощения поиска и дополнительной фильтрации можно использовать предопределенные фильтры на странице результатов.

Доступны следующие фильтры:

- источник документа,
- семейство документов,
- критичность по CVSS.

Также можно выбрать параметры сортировки:

- по релевантности,
- по дате,
- по уровню критичности по CVSS,
- по заголовку.

3. Описание REST-API

3.1. API-ключ

Для взаимодействия с REST API Vul-Search требуется API-ключ. Для получения ключа необходимо выполнить следующие действия:

1. Направить на электронную почту support@vul-search.ru запрос с указанием данных для регистрации API-ключа по форме:

Наименование организации	
№ Лицензионного договора с ООО «КИВИ Лаб»	
ФИО	

2. API-ключ будет выслан на электронную почту, с которой был получен запрос.

Для проверки валидности API-ключа выполните следующий запрос.

Запрос:

[1] Идентификатор уязвимости в соответствии с классификацией https://www.cve.org

```
POST /api/v3/apiKey/valid/
```

Входными данными могут быть:

- Строка запроса, например:

```
curl -XPOST 'https://Vul-Search.ru/api/v3/apiKey/valid/?keyID={API key}'
```

- JSON, например:

```
curl -XPOST --compressed https://Vul-Search.ru/api/v3/apiKey/valid -H
'Content-Type: application/json' -d '{
   "keyID": "{API key}"
}'
```

3.2. Поиск в базе данных

Функция поиска в базе данных аналогична поиску на сайте Vul-Search.

Для поиска информации в базе данных существуют следующие способы:

1. Поиск по ключевым словам с использованием синтаксиса языка Lucene.

Запрос:

```
POST /api/v3/search/lucene/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/lucene -H 'Content-Type: application/json' -d '{{
"query": "heartbleed",
"skip": 0,
"size": 10,
"fields": [
   "id",
   "title",
   "description",
   "type",
   "bulletinFamily",
   "cvss",
   "published",
```

```
    "modified",
    "lastseen",
    "href",
    "sourceHref",
    "sourceData",
    "cvelist"],
"apiKey": "{API key}"
}'
```

2. Получение полной информации о документе по id. Вы можете получить полную информацию о бюллетене по определенному идентификатору.

Запрос:

```
POST /api/v3/search/id/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/id -H 'Content-Type: application/json' -d '{{
"id": "CVE-2017-14174",
fields": ["*"],
"apiKey": "{API key}"
}'
```

3. Получение полной информации о списке документов по id. Если вам нужна полная информация о списке бюллетеней, просто укажите список необходимых идентификаторов документов, данные о которых вы хотите получить:

Запрос:

```
POST /api/v3/search/id/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/id -H 'Content-Type: application/json' -d '{{
"id": [
    "CVE-2017-14174",
    "CVE-2016-1175"],
"fields": ["*"],
```

```
"apiKey": "{API key}"
}'
```

Каждый из этих вызовов может использовать следующие параметры для регулирования количества получаемой информации:

- "skip" и "size" позволяют организовать получение данных порциями, где size – размер порции, skip – смещение относительно начала очереди документов. Значения по умолчанию: 0 и 20 соответственно.
- "fields" позволяет получать только необходимую часть полей документа. В качестве значения принимает список названий полей. Значение по умолчанию:

```
default_fields = ["id","title","description","type","bulletinFamily","cvss","published","modified","lastseen","href","sourceHref","sourceData","cvelist"]
```

3.2.1. Использование сложных запросов в базу данных

Строка поиска предназначена для передачи запросов в базу данных. Поисковая строка позволяет выполнять поиск по ключевым словам или с помощью языка текстовых запросов Apache Lucene.

Для простого поиска можно ввести набор ключевых слов, система вернет список документов, которые содержат указанные ключевые слова в полях title и description.

С помощью языка запросов Apache Lucene можно отбирать документы по значениям полей. Для этого надо указать название поля и значение, разделенные двоеточием. Например:

```
type:nmap order:published
```

Также язык запросов Apache Lucene позволяет использовать логические операторы – AND OR, NOT и группировку. Например:

- Уязвимости вашего сайта по информации Openbugbounty:

```
type:openbugbounty AND title:"your-domain-here.com" AND openbugbounty.patchStatus:unpatched
```

- Уязвимости Firefox:

```
affectedSoftware.name:"firefox" OR affectedPackage.packageName:"firefox" OR cpe:*firefox* order:published
```

- Уязвимости конкретной версии ПО, например Nginx 1.11.0:

```
affectedSoftware.name:nginx AND affectedSoftware.version:"1.11.0"
```

- Поиск уязвимостей Centos по названию, начинающемуся со слов "Critical" или "Important" и NETWORK в CVSS vector:

```
(type:centos AND (title:"Critical" OR title:"Important") AND cvss.vector:"AV:NETWORK") order:published
```

Для числовых полей доступен поиск по диапазону значений. Например:

- Уязвимости Microsoft из каталога CVE, для которых CVSS находится в диапазоне от 6 до 10, отсортированные по дате:

```
(type:cve AND cvss.score:[6 TO 10] AND description:"Microsoft") order:published
```

Подробнее с синтаксисом языка Apache Lucene можно ознакомиться в официальной документации по ссылке - https://lucene.apache.org/core/2_9_4/queryparsersyntax.html.

3.3. Поиск общедоступных эксплойтов

С помощью этого метода можно получить общедоступные эксплойты из базы данных Vul-Search, указав идентификатор уязвимости или программного обеспечения:

Запрос:

```
POST /api/v3/search/lucene/
```

Пример запроса для идентификатора ПО:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/lucene/ -H 'Content-Type: application/json' -d '{{
"query": "wordpress 4.7.0",
"skip": 0,
"size": 100,
"fields": [
   "id",
```

```
    "title",
    "description",
    "type",
    "bulletinFamily",
    "cvss",
    "published",
    "modified",
    "lastseen",
    "href",
    "sourceHref",
    "sourceData",
    "cvelist",
    "sourceData"],
"apiKey": "{API key}"
}'
```

Пример запроса для CVE:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/lucene/ -H 'Content-Type: application/json' -d '{
"query": "CVE-2019-0708",
"skip": 0,
"size": 100,
"fields": [
    "id",
    "title",
    "description", "type",
    "bulletinFamily",
    "cvss",
    "published",
    "modified",
    "lastseen",
    "href",
    "sourceHref",
```

```
   "sourceData",
   "cvelist",
   "sourceData"],
"apiKey": "{API key}"
}'
```

Пример с методом поиска:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/lucene/ -H 'Content-Type: application/json' -d '{
"query": "bulletinFamily:exploit AND wordpress 4.7.0",
"skip": 0,
"size": 10,
"fields": [
   "id",
   "title",
   "description",
   "type",
   "bulletinFamily",
   "cvss",
   "published",
   "modified",
   "lastseen",
   "href",
   "sourceHref",
   "sourceData",
   "cvelist"],
"apiKey": "{API key}"
}'
```

3.4. Получение уязвимостей/эксплойтов по названию и версии программного обеспечения

Уязвимости для программного обеспечения + версия:

Запрос:

```
POST /api/v3/burp/softwareapi/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/burp/softwareapi/ -H 'Content-Type: application/json' -d '{
"software": "webkit2gtk",
"version": "2.32.3-1",
"type": "software",
"maxVulnerabilities": 10,
"apiKey": "{API key}"
}'
```

Аналогично можно задать список ПО для аудита: название программы + версия.
Запрос:

```
POST /api/v3/burp/packages/
```

Пример запроса (ограничение запрашиваемого списка 500):

```
curl -XPOST https://Vul-Search.ru/api/v3/burp/packages/ -H 'Content-Type: application/json' -d '{
"os": "",
"osVersion": "",
"packages": [{ "software": "apache tomcat", "version": "6.0.12" }],
"apiKey": "{API key}"
}'
```

3.5. Получение уязвимостей по продукту CPE и версии

Вы можете получить все уязвимости, указав продукт CPE + версию (строку):

Запрос:

```
POST /api/v3/burp/softwareapi/
```

Пример запроса для идентификатора CPE:

```
curl -XPOST https://Vul-Search.ru/api/v3/burp/softwareapi/ -H 'Content-Type: application/json' -d '{
```

```
"software": "cpe:/a:cybozu:garoon:4.2.1",
"version": "4.2.1",
"type": "cpe",
"maxVulnerabilities": 50,
"apiKey": "{API key}"
}'
```

Также можно получить все уязвимости, указав список продуктов с помощью CPE: vendor + имя производителя + версия (строка).

Запрос:

```
POST /api/v3/burp/packages/.
```

Пример запроса (ограничение запрашиваемого списка 500):

```
curl -XPOST https://Vul-Search.ru/api/v3/burp/packages/ -H 'Content-Type: application/json' -d '{
"os": "",
"osVersion": "",
"packages": [{ "software": "apache tomcat", "version": "6.0.12" }],
"apiKey": "{API key}"
}'
```

3.6. Получение ссылок на уязвимости

Получение всех бюллетеней по идентификатору:

Запрос:

```
POST /api/v3/search/id/
```

Пример запроса:

```
curl -POST --compressed https://Vul-Search.ru/api/v3/search/id/ -H 'Content-Type: application/json' -d '{{
"id": "CVE-2014-0160",
"fields": [
   "id",
   "title",
   "description",
```

```
    "type",
    "bulletinFamily",
    "cvss",
    "published",
    "modified",
    "lastseen",
    "href",
    "sourceHref",
    "sourceData",
    "cvelist"],
"references": "true",
"apiKey": "{API key}"}'
```

3.7. Загрузить всю коллекцию баз данных

Загрузить всю коллекцию баз данных и работать с данными локально. Эта функция доступна платным подписчикам:

Запрос:

```
GET /api/v3/archive/collection
```

Пример строки запроса:

```
curl -GET https://Vul-Search.ru/api/v3/archive/collection/?type=cve&datefrom=1950-01-
```

3.8. Аудит хостов Windows на предмет установленных KB

Использует быстрый аудит хостов Windows на предмет установленных KB безопасности. Для аудита Windows требуется версия ОС + список установленных обновлений (KB). Версию ОС и список установленных обновлений можно получить с помощью команды

```
systeminfo
```

Запрос:

```
POST /api/v3/audit/kb/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/audit/kb/ -H 'Content-Type: application/json' -d '{
"os": "Windows Server 2012 R2",
"kbList": ["KB4072650", "KB2959936", "KB2894856", "KB2896496"],
"apiKey": "{API key}"
}'
```

3.9. Аудит установленных KB's и программного обеспечения

Используйте быстрый аудит Windows-хостов на предмет установленных KB и программ безопасности. Для аудита Windows требуется версия ОС, список установленных обновлений (KB) и список установленного ПО.

Запрос:

```
POST /api/v3/audit/winaudit/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/audit/kb/ -H 'Content-Type: application/json' -d '{{
"os": "Windows Server 2012 R2",
"kbList": ["KB4072650", "KB2959936", "KB2894856", "KB2896496"],
"software": [{"software": "7-Zip 19.00 (x64)", "version": "19.00"}, {"software": "Git", "version": "2.33.0.2"},
  {"software": "Notepad++ (64-bit x64)", "версия": "8.4.6"}, {"software": "Microsoft 365 - en-us", "version": "16.0.15726.20202"},
  {"software": "Microsoft 365 - ru-ru", "version": "16.0.15726.20202"},
  {"software": "Microsoft OneDrive", "версия": "22.227.1030.0001"},
  {"software": "Total Commander 64-bit", "версия": "10.00"},
  {"software": "Microsoft .NET AppHost Pack - 6.0.0 (x64_x86)", "версия": "48.3.31210"},
  {"software": "Microsoft .NET Host FX Resolver - 6.0.0 (x64)", "version": "48.3.31210"},
  {"software": "VMware Player", "version": "16.2.4"}, {"software": "Foxit PDF Reader", "version": "12.0.1.12430"}]
}'
```

3.10. Получение данных о иерархии Windows KB

Обратитесь к этому методу, если после предыдущего метода вам потребуется дополнительная информация о KB. Информация об иерархии будет возвращена в виде словаря с двумя полями: 'superseeds', 'parentseeds'.

- superseeds означает "какие KB включает данный KB", т.е. стоящие ниже в иерархии.
- parentseeds означает "какие KB содержат данный KB", т.е. стоящие выше по иерархии.

Запрос:

```
POST /api/v3/search/id/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/id/ -H 'Content-Type: application/json' -d '{{
"id": "KB4524135",
"fields": ["superseeds", "parentseeds"],
"apiKey": "{API key}"
}'
```

3.11. Получение списка обновлений Windows KB со ссылками для загрузки

Запрос:

```
POST /api/v3/search/lucene/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/search/lucene/ -H 'Content-Type: application/json' -d '{ {
"query": "type:msupdate AND kb:(KB4524135)",
"skip": 0,
"size": 100,
"fields": [
  "id",
  "title",
  "description",
```

```
    "type",
    "bulletinFamily",
    "cvss",
    "published",
    "modified",
    "lastseen",
    "href",
    "sourceHref",
    "sourceData",
    "cvelist"],
"apiKey": "{API key}"
}'
```

3.12. Аудит узлов Linux на наличие уязвимостей (на основе RPM/DEB)

Для проведения аудита Linux необходимы версии пакетов, установленных в системе.

Вы можете получить список установленных пакетов для любой ОС на базе RPM с помощью следующей команды:

```
rpm -qa --qf '%{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}\\n'.
```

Запрос:

```
POST /api/v3/audit/audit/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/audit/audit/ -H 'Content-Type: application/json' -d '{{
"os": "centos",
"version": "7",
"package": ["glibc-common-2.17-157.el7_3.5.x86_64"],
"apiKey": "{API key}"
}'
```

Вы можете получить список установленных пакетов для любой ОС на базе DEB с помощью следующей команды:

```
dpkg-query -W -f='${Package} ${Version} ${Architecture}\\n'.
```

Запрос:

```
POST /api/v3/audit/audit/
```

Пример запроса:

```
curl -XPOST https://Vul-Search.ru/api/v3/audit/audit/ -H 'Content-Type: application/json' -d '{{
"os": "debian",
"version": "8",
"package": ["uno-libs3 4.3.3-2+deb8u7 amd64"],
"apiKey": "{API key}"
}'
```

3.13. Загрузить данные об уязвимостях заданного дистрибутива Linux

Вы можете загрузить данные об оценке уязвимостей Linux (на основе RPM/DEB) и работать с ними локально для ускорения обработки.

Запрос:

```
GET /api/v3/archive/distributive/
```

Пример запроса:

```
curl -GET https://vulners.com/api/v3/archive/distributive/?os=CentOS&version=7&apiKey={API key}
```