

Программа для ЭВМ

«Leak-search»

Инструкция по установке

Оглавление

Оглавление	2
1. Введение	3
2. Системные требования	3
2.1 Серверное программное обеспечение.....	3
2.2 Аппаратные характеристики серверов.....	3
3. Обзор процесса установки	3
4. Установка	4
4.1 Установка Docker	4
4.2 Установка и первый запуск сервиса «Leak-search»	6
4.3 Удаление сервиса «Leak-search»	9
5. Контактная информация производителя программного продукта	10
5.1 Юридическая информация	10
5.2 Контактная информация службы технической поддержки	10
Приложение 1. Перечень компонентов docker-контейнеров	10
Приложение 2. Создание аккаунта GitHub	12
П.2-1 Регистрация аккаунта GitHub	12
П.2-2 Создание API ключа	15

1. Введение

Настоящая инструкция описывает процесс установки программного продукта «Программа для ЭВМ «Leak-search»», представляющего собой сервис автоматизированного поиска (далее по тексту также Leak-search, Система, Сервис, сервис «Leak-search», программный продукт).

2. Системные требования

2.1 Серверное программное обеспечение

Развертывание сервиса «Leak-search» возможно на любой Linux-совместимой ОС с версией ядра 4.13+ с установленным ПО Docker. По состоянию на текущий момент для развертывания подойдут актуальные версии Debian(10, buster), Astra Linux.

Минимально необходимые версии компонентов Docker:

- *docker 19.03.+;*
- *docker-compose 1.25.+.*

2.2 Аппаратные характеристики серверов

Для установки сервиса «Leak-search» рекомендуется использование следующих ресурсов:

- **Сервер:** 2-core CPU, 16 Гбайт ОЗУ, 20 Гбайт дискового пространства.
- Также для установки необходимо подключение к сети Интернет.

3. Обзор процесса установки

Сервис «Leak-search» устанавливается путем запуска одного установочного скрипта (*leak-search-deploy.sh*) в дистрибутиве.

Ссылка для скачивания дистрибутива - <https://static.leak-search.com/download/leak-search.zip>

В состав дистрибутива входят следующие компоненты:

Docker-контейнеры:

- *leaks.master.backend* — внутренняя часть сервиса «Leak-search» (собственная разработка);
- *leaks.master.frontend* — web-интерфейс сервиса «Leak-search» (собственная разработка);
- *leaks.master.searching_tasks* — запуск поисковых задач (собственная разработка);
- *leaks.master.start_task* — поставщик данных для поискового узла;
- *leaks.master.gist_search_task* — инициатор заданий для поиска по сервису gist.github.com
- *leaks.master.gist_result_task* — сборщик результатов поиска по сервису gist.github.com

- *leaks.worker.git_tasks* — поисковый узел по сервису gist.github.com (собственная разработка);
- *leaks.master.git_search_task* — инициатор заданий для поиска по сервису github.com
- *leaks.master.git_result_task* — сборщик результатов поиска по сервису github.com
- *leaks.worker.git_tasks* — поисковый узел по сервису github.com (собственная разработка);
- *leaks.master.postgres* — СУБД PostgreSQL;
- *leaks.master.rabbitmq* — брокер сообщений Rabbit MQ;
- *leaks.master.redis* — кэширующая СУБД для брокера сообщений;
- *leaks.master.schedule* — процесс, запускающий поиск утечек по расписанию;
- *leaks.worker.proxy* — прокси-сервер для работы поисковых узлов.

Перечисленные контейнеры упакованы в архив *leak-search.tar.gz*

Дополнительные компоненты:

- *leak-search-deploy.sh* — скрипт запуска установки (собственная разработка);
- *leak-search-deploy.yml* — плейбук для развёртывания программного продукта через систему docker-compose (собственная разработка);
- *leak-search.env* — файл конфигурации установщика (собственная разработка);
- *leak-search-remove.sh* — скрипт удаления Leak-search (собственная разработка).

(В Приложении 1 к настоящей инструкции приведен перечень компонентов контейнеров, лицензий их распространения и адреса репозиториях этих компонентов в сети Интернет)

Установочный скрипт запускает docker-контейнеры. После успешного запуска на компьютере, на котором была установлена Система, будет доступен веб-интерфейс Сервиса «Leak-search», взаимодействовать с которым можно посредством любого браузера по адресу <http://localhost:8080>

4. Установка

4.1 Установка Docker

Обязательным предусловием для установки «Leak-search» является установка следующих пакетов:

- *docker 19.03.+;*
- *docker-compose 1.25.+.*

Детали их установки описаны в документах <https://docs.docker.com/engine/install/debian/> и <https://docs.docker.com/compose/install>.

Для дистрибутива Debian или Astra Linux установка производится следующими командами:

```
$ sudo apt-get update
$ sudo apt-get install -y \
apt-transport-https \
```

```

ca-certificates \
curl \
gnupg \
lsb-release

$ sudo mkdir -p /usr/share/keyrings/

$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

$ echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian buster stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

$ sudo apt-get update

$ sudo apt-get install -y docker-ce docker-ce-cli containerd.io

$ sudo curl -L \
  "https://github.com/docker/compose/releases/download/1.25.0/docker-compose-$(uname -s)-
$(uname -m)" -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose

```

Пользователь, под учетной записью которого будет производиться установка, должен быть участником группы ***docker***. Команда для проверки:

```

user@host:~$ groups
user docker

```

Если пользователь не является членом группы ***docker***, нужно выполнить следующую команду:

```
$ sudo usermod -aG docker USER_NAME
```

где ***USER_NAME*** — название учетной записи пользователя, от которого будет запускаться скрипт установки.

После этого необходимо выйти из учётной записи и зайти снова, для применения правил группы к пользователю.

4.2 Установка и первый запуск сервиса «Leak-search»

Для установки сервиса «Leak-search» выполните следующие шаги:

1. Скачайте архив с дистрибутивом программного продукта. Ссылка на архив – <https://static.leak-search.com/download/leak-search.zip>
2. Распакуйте скаченный архив любым удобным для вас способом в отдельный каталог, например, leak-search.
Внимание! Архив защищен паролем. Если у вас еще нет пароля, обратитесь в техподдержку по адресу support@leak-search.com.

```
$ unzip leak-search.zip -d leak-search
```

3. Войдите в каталог leak-search:

```
$ cd leak-search/
```

4. Исправьте в файле leak-search.env значение следующих переменных - github_login (имя созданного вами аккаунта на GitHub) и github_token (токен учетной записи), созданные вами, как описано в Приложении 2.
(Для удобства экспертизы была создана учетная запись, привязанная к предоставляемому для экспертизы экземпляру программного продукта.)

```
SERVICE_ACCOUNT=github_login  
SERVICE_TOKEN=github_token
```

Примечание: Пользователи Сервиса не занимаются созданием аккаунтов на внешних относительно Leak-search системах, поэтому не зависят от санкционных политик и ограничений. Все необходимые учетные записи создаются и периодически обновляются в рамках сопровождения Сервиса без заявок от клиентов.

5. Запустите скрипт leak-search-deploy.sh

```
$ ./leak-search-deploy.sh
```

Скрипт автоматически развернет все пятнадцать Docker-контейнеров.

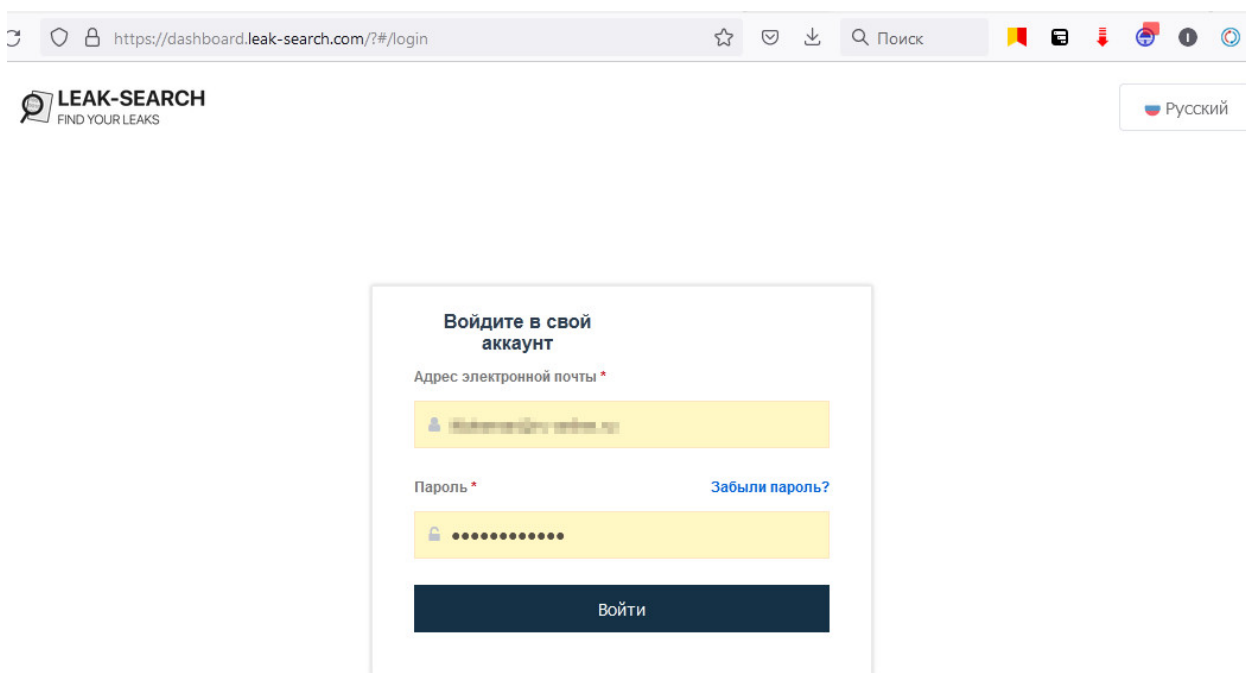
Об успехе развертывания свидетельствует значение «done» в конце каждой строки после наименования соответствующего контейнера.

```
Load Images
39982b2a789a: Loading layer [=====] 5.885MB/5.885MB
81c3a55b8d1a: Loading layer [=====] 108.2MB/108.2MB
ae1ac60f839e: Loading layer [=====] 7.66MB/7.66MB
f4bd89c91855: Loading layer [=====] 3.584kB/3.584kB
88a0dbee8560: Loading layer [=====] 2.048kB/2.048kB
77180ad23b7f: Loading layer [=====] 613.4kB/613.4kB
208096353844: Loading layer [=====] 321.7MB/321.7MB
e0a376392b6c: Loading layer [=====] 3.436MB/3.436MB
Loaded image: leak-search-frontend:2.8.113
6ead25fbf64c: Loading layer [=====] 203.7MB/203.7MB
Loaded image: leak-search-backend:2.8.113
d000633a5681: Loading layer [=====] 72.53MB/72.53MB
55ad5d8b8f0f: Loading layer [=====] 7.377MB/7.377MB
01a27e800166: Loading layer [=====] 28.01MB/28.01MB
8bee098d3f25: Loading layer [=====] 4.608kB/4.608kB
53df63ea34e2: Loading layer [=====] 10.07MB/10.07MB
c606c73960d7: Loading layer [=====] 350.2kB/350.2kB
8f5d5437ad4e: Loading layer [=====] 513.8MB/513.8MB
c7790994d2fa: Loading layer [=====] 2.048kB/2.048kB
7cf22a78db69: Loading layer [=====] 53.26MB/53.26MB
e77a32c50b21: Loading layer [=====] 64kB/64kB
c06bdd805c4d: Loading layer [=====] 3.072kB/3.072kB
Loaded image: leak-search-node:2.8.113
Launch Leak-Search
Creating volume "leaks.master.rabbit.data" with default driver
Creating volume "leaks.master.postgres.data" with default driver
Creating volume "leaks.master.redis.data" with default driver
Creating volume "leaks.master.logs.data" with default driver
Creating volume "leaks.worker.logs.data" with default driver
Creating volume "leaks.master.proxy.data" with default driver
Pulling postgres (postgres:11)...
11: Pulling from library/postgres
442547fc262c: Pull complete
99f9b0d37d63: Pull complete
673a0386621a: Pull complete
e6d442b05d6a: Pull complete
e9b9f3eeffa3: Pull complete
17b8a068ac5a: Pull complete
2916937b1163: Pull complete
ddee610c7782: Pull complete
a04348da5dce: Pull complete
d40e40562345: Pull complete
31fac142e00b: Pull complete
74db098f55a9: Pull complete
529ebe5ad1d1: Pull complete
Digest: sha256:147d3cd54a8ebc7692cc0d58b6cd04dd30d5dc5f89d6957f4df7357a9b864b76
Status: Downloaded newer image for postgres:11
Pulling redis (redis:5.0)...
5.0: Pulling from library/redis
a330b0cecb98: Already exists
14bfbab96d75: Pull complete
8b3e2d14a955: Pull complete
4a5768b461bc: Pull complete
59b264a02740: Pull complete
fe842fc77a96: Pull complete
Digest: sha256:6aa31467e60f57a15427ab847752cc075d3218f70ba5fb6c913c4b9c09b61c95
Status: Downloaded newer image for redis:5.0
Pulling rabbitmq (rabbitmq:3.7-management-alpine)...
3.7-management-alpine: Pulling from library/rabbitmq
df20fa9351a1: Pull complete
6a4ed0140701: Pull complete
5979a511b05f: Pull complete
7508e6c462e3: Pull complete
a5e057a63f72: Pull complete
c740f8e9d4eb: Pull complete
9039a2c6dbd4: Pull complete
9efc49b21a0d: Pull complete
4012d1250c55: Pull complete
Digest: sha256:e97f6bb68aab3f747f8a6f4dc52ed57401de91ca480a0be3a36e7476ac273169
Status: Downloaded newer image for rabbitmq:3.7-management-alpine
Pulling http_proxy (vimagick/tinyproxy:latest)...
latest: Pulling from vimagick/tinyproxy
5843afab3874: Pull complete
b351547d7b1c: Pull complete
Digest: sha256:72b441b95ee1e641af948f68f09492f9f795ead72b73954414e339168c98ad8c
Status: Downloaded newer image for vimagick/tinyproxy:latest
Creating leaks.worker.gist_tasks ... done
Creating leaks.master.redis ... done
Creating leaks.master.git_result_task ... done
Creating leaks.master.backend ... done
Creating leaks.master.schedule ... done
Creating leaks.master.start task ... done
Creating leaks.master.gist_search_task ... done
Creating leaks.worker.proxy ... done
Creating leaks.master.gist_result_task ... done
Creating leaks.master.rabbit ... done
Creating leaks.master.git_search_task ... done
Creating leaks.master.searching tasks ... done
Creating leaks.worker.git_tasks ... done
Creating leaks.master.frontend ... done
Creating leaks.master.postgres ... done
```

Сообщение об успешной установке сервиса «Leak-search»

6. После выполнения действий 1-4 из пункта 4.2 данной инструкции, приложение будет доступно по адресу <http://localhost:8080/#login>
7. Зайдите в браузер по адресу <http://localhost:8080/#login> и убедитесь, что сервис «Leak-search» доступен для работы. Более подробно программный продукт описан в Руководстве пользователя.

Доступ в сервис «Leak-search» осуществляется по выданным Владелец Системы логину и паролю:



Экран авторизации в сервисе «Leak-search»

Примечание. Проверить состояние запущенных контейнеров можно командой «docker ps».

```
$ docker ps
```


CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
caae2bd4a78	postgres:11	"docker-entrypoint.s..."	29 seconds ago	Up 28 seconds		leaks.master.postgres
9a1ac729785	leak-search-frontend:2.8.113	"docker-entrypoint.s..."	29 seconds ago	Up 28 seconds		leaks.master.frontend
8a5d6ee676e	leak-search-node:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.worker.git_tasks
8aa79ebf75e	rabbitmq:3.7-management-alpine	"docker-entrypoint.s..."	29 seconds ago	Up 28 seconds		leaks.master.rabbit
ac601d68845	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.searching_tasks
3080b4be1e3	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.git_search_task
8a77c8b8cdd8	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.git_result_task
5b5b4724658	vimagic/tinyproxy	"tinyproxy -d"	29 seconds ago	Up 28 seconds	8888/tcp, 0.0.0.0:45785->45785/tcp, :::45785->45785/tcp	leaks.worker.proxy
99f7bc1204b	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.git_result_task
7dfda555f7e3	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.git_search_task
78eb1a43870	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.start_task
2a354458641	leak-search-backend:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.master.schedule
2a1b1d64676	leak-search-backend:2.8.113	"/entrypoint.backen..."	29 seconds ago	Up 28 seconds		leaks.master.backend
1448bc788724	leak-search-node:2.8.113	"/entrypoint.celery..."	29 seconds ago	Up 28 seconds		leaks.worker.git_tasks
7fb99c54b7c4	redis:5.0	"docker-entrypoint.s..."	29 seconds ago	Up 28 seconds		leaks.master.redis

Сообщения о состоянии запущенных контейнеров сервиса «Leak-search»

4.3 Удаление сервиса «Leak-search»

Чтобы удалить сервис «Leak-search» запустите скрипт *leak-search-remove.sh*

```
$ ./leak-search-remove.sh
```

[illegible]

Сообщение о завершении операции удаления сервиса «Leak-search»

5. Контактная информация производителя программного продукта

5.1 Юридическая информация

Информация о юридическом лице компании:

Название компании: Общество с ограниченной ответственностью «Биллинг Онлайн Решения».

Юр. адрес: 620014, Свердловская обл., г. Екатеринбург, ул. 8 Марта, стр. 32а, этаж/пом. 5/1-7, 6/1-5.

ОГРН 1196658065955

ИНН 6671099510

5.2 Контактная информация службы технической поддержки

Связаться со специалистами службы технической поддержки можно по электронной почте и/или телефону:

Email: support@leak-search.com

Тел.: 8 800 600 07 27

Приложение 1. Перечень компонентов docker-контейнеров.

Наименование	Лицензия	Ссылка на репозиторий
Linux	GPL v2	https://github.com/torvalds/linux
docker-slim	Apache License Version 2.0	https://github.com/docker-slim/docker-slim
PostgreSQL	Postgresql license	https://git.postgresql.org/gitweb/?p=postgresql.git;a=summary
RabbitMQ	MPL 2.0	https://github.com/rabbitmq/rabbitmq-server
Redis	3-Clause BSD license	https://github.com/redis/redis
Python	PSF license + BSD Zero clause license	https://github.com/python/cpython
Django	3-Clause BSD license	https://github.com/django/django
Django REST Framework	3-Clause BSD license	https://github.com/encode/django-rest-framework
django-filter	3-Clause BSD license	https://github.com/carltongibson/django-filter
django-otp	2-Clause BSD license	https://github.com/django-otp/django-otp
beautifulsoup	MIT License	https://code.launchpad.net/beautifulsoup/
celery	3-Clause BSD license	https://github.com/celery/celery
lxml	3-Clause BSD license	https://github.com/lxml/lxml
markupsafe	3-Clause BSD license	https://github.com/pallets/markupsafe
phonenumbers	Apache License Version 2.0	https://github.com/daviddrysdale/python-phonenumbers

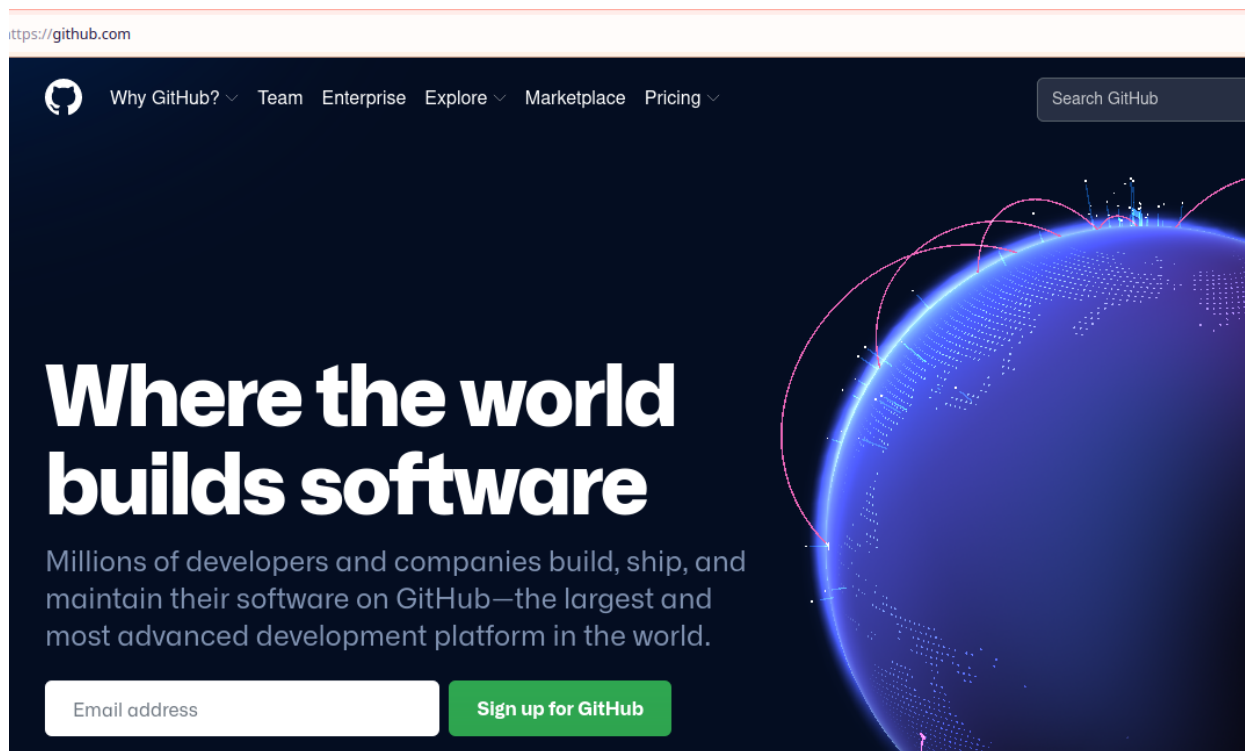
Наименование	Лицензия	Ссылка на репозиторий
prettytable	3-Clause BSD license	https://github.com/jazzband/prettytable
psycpg2	LGPL	https://github.com/psycpg/psycpg2
pyTelegramBotAPI	GPL-2.0 License	https://github.com/eternnoir/pyTelegramBotAPI
python-dateutil	PSF license	https://github.com/paxan/python-dateutil
python-telegram-handler	MIT License	https://github.com/sashgorokhov/python-telegram-handler
qrcode	3-Clause BSD license	https://github.com/lincolnloop/python-qrcode
requests	Apache License Version 2.0	https://github.com/psf/requests
redis	MIT License	https://github.com/andymccurdy/redis-py
aiohttp	Apache License Version 2.0	https://github.com/aio-libs/aiohttp
pygments	2-Clause BSD license	https://github.com/pygments/pygments
elasticsearch	Apache License Version 2.0	https://github.com/elastic/elasticsearch-py
babel	3-Clause BSD license	http://babel.pocoo.org/en/latest/
pytest	MIT License	https://docs.pytest.org/en/6.2.x/contents.html
pytest-xdist	MIT License	https://github.com/pytest-dev/pytest-xdist
pytest-django	3-Clause BSD license	https://github.com/pytest-dev/pytest-django
pyotp	MIT License	https://github.com/pyauth/pyotp
node.js	MIT License	https://github.com/nodejs/node
vue.js	MIT License	https://github.com/vuejs/vue
vue-avatar	MIT License	https://github.com/eliap/vue-avatar
vue-chartjs	MIT License	https://github.com/apertureless/vue-chartjs
vue-datetime	MIT License	https://github.com/mariomka/vue-datetime
vue-router	MIT License	https://github.com/vuejs/vue-router
vue-spinner	MIT License	https://github.com/greyby/vue-spinner
weekstart	MIT License	https://github.com/gamtiq/weekstart
js-cookie	MIT License	https://github.com/js-cookie/js-cookie
axios	MIT License	https://github.com/axios/axios
font-awesome	Font Awesome Free License	https://github.com/FortAwesome/Font-Awesome
Intro.js	GNU GPL	https://github.com/usablica/intro.js
chart.js	MIT License	https://github.com/chartjs/Chart.js
element-ui	MIT License	https://github.com/ElementUI/lib
vue-i18n	MIT License	https://github.com/kazupon/vue-i18n
vue-i18n-loader	MIT License	https://github.com/intlify/vue-i18n-loader
tinypoxy	GPL-2.0 License	https://github.com/tinypoxy

Приложение 2. Создание аккаунта GitHub.

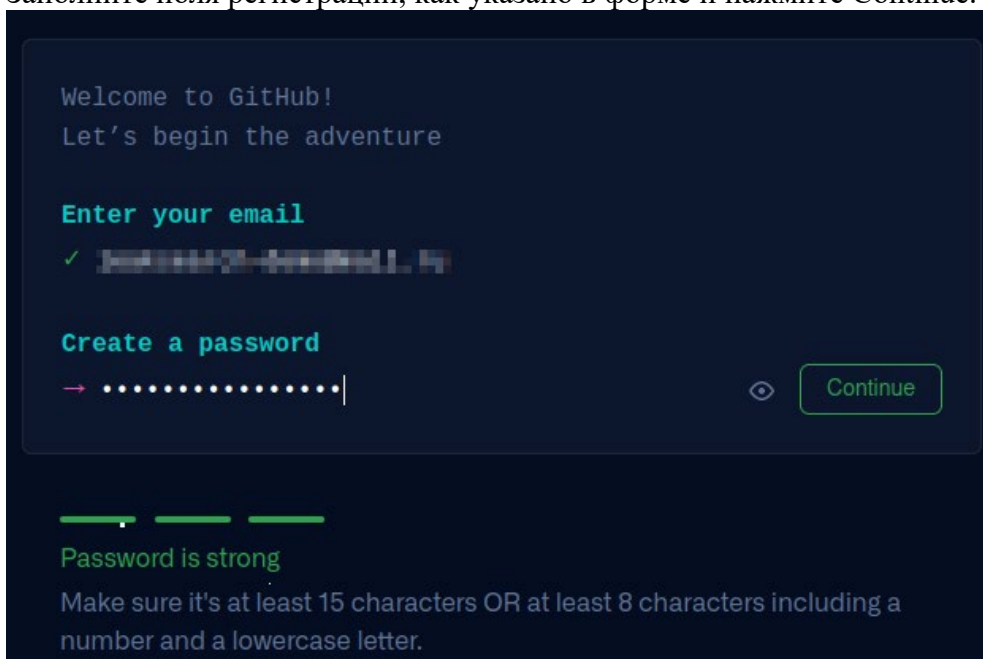
Для осуществления первоначальной настройки и успешного поиска утечек в сервисе GitHub необходимо завести учетную запись в этом сервисе и получить API-ключ.

П.2-1 Регистрация аккаунта GitHub

Перейдите на <https://github.com> и нажмите кнопку «Sign Up for GitHub».



Заполните поля регистрации, как указано в форме и нажмите Continue.



Далее введите значение капчи и нажмите «Create Account».

После этого введите код подтверждения, полученный в письме, пришедшем на электронный адрес, указанный при регистрации.

ue

You're almost done!

We sent a launch code to [redacted email]

→ Enter code

[Six empty input boxes for the code]

Didn't get your email? [Resend the code](#) or [update your email address](#).

После этого Вы будете перенаправлены на страницу настройки персонализации вашего аккаунта. Пропустите её, нажав «Skip Personalization», для перехода в панель управления GitHub.

https://github.com/join/welcome



Welcome to GitHub

We are glad you're here.

How many team members will be working with you?

This will help us guide you to the tools that are best suited for your projects.

Just me

2 - 5

5 - 10

10 - 20

20 - 50

50+

Are you a student or teacher?

Student

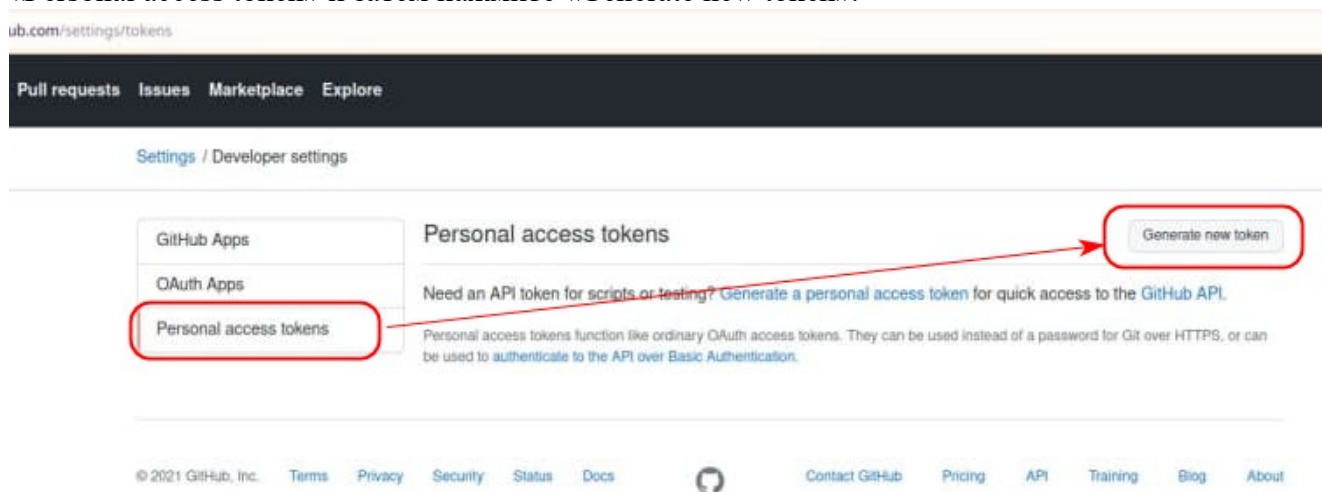
Teacher

Continue

Skip personalization

П.2-2 Создание API ключа

Перейдите на страницу <https://github.com/settings/tokens>, щелкните мышью на поле «Personal access token» и затем нажмите «Generate new token».



Заполните поле Note как показано на рисунке и в поле Expiration выберите «No Expiration», остальные поля оставьте незаполненными и нажмите «Generate token»

thub.com/settings/tokens/new

Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

Personal access tokens

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Generate new token

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

thub.com/settings/tokens/new

Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

leak-search personal access token

What's this token for?

Expiration *

No expiration ⌵ The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

☐ repo	Full control of private repositories
☐ repo:status	Access commit status
☐ repo_deployment	Access deployment status
☐ public_repo	Access public repositories
☐ repo:invite	Access repository invitations
☐ security_events	Read and write security events
☐ workflow	Update GitHub Action workflows
☐ write:packages	Upload packages to GitHub Package Registry
☐ read:packages	Download packages from GitHub Package Registry
☐ delete:packages	Delete packages from GitHub Package Registry
☐ admin:org	Full control of orgs and teams, read and write org projects
☐ write:org	Read and write org and team membership, read and write org projects
☐ read:org	Read org and team membership, read org projects

https://github.com/settings/tokens/new

☐ read:org	Read org and team membership, read org projects
☐ admin:public_key	Full control of user public keys
☐ write:public_key	Write user public keys
☐ read:public_key	Read user public keys
☐ admin:repo_hook	Full control of repository hooks
☐ write:repo_hook	Write repository hooks
☐ read:repo_hook	Read repository hooks
☐ admin:org_hook	Full control of organization hooks
☐ gist	Create gists
☐ notifications	Access notifications
☐ user	Update ALL user data
☐ read:user	Read ALL user profile data
☐ user:email	Access user email addresses (read-only)
☐ user:follow	Follow and unfollow users
☐ delete_repo	Delete repositories
☐ write:discussion	Read and write team discussions
☐ read:discussion	Read team discussions
☐ admin:enterprise	Full control of enterprises
☐ manage_runners:enterprise	Manage enterprise runners and runner-groups
☐ manage_billing:enterprise	Read and write enterprise billing data
☐ read:enterprise	Read enterprise profile data
☐ admin:gpg_key	Full control of public user GPG keys (Developer Preview)
☐ write:gpg_key	Write public user GPG keys
☐ read:gpg_key	Read public user GPG keys

Generate token Cancel

Скопируйте ваш токен и сохраните его, он понадобится позднее.

https://github.com/settings/tokens

Pull requests Issues Marketplace Explore

Settings / Developer settings

GitHub Apps
OAuth Apps
Personal access tokens

Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓Copy
Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About