



alexkbs 5 января 2017 в 06:55

Let's Encrypt и nginx: настройка в Debian и Ubuntu

Настройка Linux, Nginx, Серверное администрирование

Tutorial



Если вдруг вся эта история прошла мимо вас, Let's Encrypt — центр сертификации от не существующий при поддержке EFF и многих компаний, взявшей на себя миссию дать лк для сайтов и серверов. Сертификаты от Let's Encrypt уже используются на более чем 10

Кроме очевидной бесплатности у сертификатов от Let's Encrypt есть особое, отсутствующ у сертификационных центров, достоинство: если вы однажды получили сертификат от Le навсегда. Не нужно раз в год-два вручную обновлять сертификаты. Не нужно вообще вс Получил, настроил и забыл!

Внимательный читатель сразу захочет возразить: как же так, ведь известно что сертификаты выдаются со сроком действия в три месяца? Всё дело в автоматическом обновлении сертификатов, которое возможно при полном отсутствии действий со стороны человека.

Организации автоматического обновления сертификатов в статье уделено пристальное внимание, с тем чтобы вы могли в полной мере оценить это принципиальное преимущество Let's Encrypt.

Почему эта статья

На сайте EFF есть [краткие инструкции по использованию Certbot](#), рекомендуемой программы для получения сертификатов, но они скорей рассчитаны на тех, кто заходит на свой сервер по SSH лишь по острой необходимости. Более [подробная документация](#) тоже есть, но пока всю ее прочитаешь и найдешь всё то, что действительно нужно знать... К тому же, в ней не рассмотрены некоторые важные стратегические вопросы использования сертификатов.

Очевидно, нужна короткая и понятная инструкция для тех, кто привычен к серверной консоли, но хочет во всём разобраться без излишних трат времени.

Содержание

Из этой статьи вы узнаете...

1. Как установить и настроить Certbot для регулярного использования.
2. Что требуется от nginx и как настроить nginx для получения сертификатов.
3. [Как получать сертификаты](#) и как проверить полученный сертификат.
4. Как установить сертификат от Let's Encrypt в nginx.
5. Как автоматически обновлять сертификаты.

Реклама

 2 230 P	 2 820 P
 3 080 P	 2 020 P
 2 000 P	 2 790 P
 1 860 P	

Caveat emptor

Всё знаете про SNI? Читайте сразу про установку.

В инструкциях ниже я исхожу из того что ваши сайты будут использовать SNI. Это расширение протокола TLS позволяет браузерам сообщить желаемое имя сайта до получения и проверки SSL сертификата от сервера. Благодаря SNI вы можете разместить сколько угодно сайтов за HTTPS на одном IP. Но не всё так просто — иначе бы зачем я об этом писал?

Есть ряд старых браузеров в принципе не поддерживающих SNI. В их число входят любые версии IE в уже заброшенном Windows XP, встроенный браузер в Android 2.3 и 2.2 из 2010 года, а также некоторые другие более экзотические браузеры и библиотеки типа Java версии 1.6 и Python до версии 2.7.9.

Если вы всё-таки хотите чтобы ваш сайт открывался в IE в Windows XP, то одним отказом от SNI эта проблема не решается. Нужно специальным образом подбирать шифры, уже отказываясь от forward secrecy и рискуя получить низкую оценку от SSL Labs. Как можно догадаться, этот вопрос заслуживает отдельного обсуждения хотя бы потому что пользователям IE под XP можно посочувствовать — у них уже не открывается половина интернета!

Еще год назад от перехода на SNI вас могла бы удерживать ограниченная поддержка этой технологии некоторыми поисковыми ботами типа Bing, но сейчас, с появлением десятков сайтов с бесплатными сертификатами от Cloudflare, что без SNI не открываются, бот Bing (что легко проверить), и боты других основных поисковиков, пришли в согласие с реальностью. Сейчас за это можно не волноваться. Отмечу, что у Googlebot таких проблем не было никогда.

Другим поводом для волнений могут быть различные средства доступа к API вашего сайта. Если у вас давно есть API, то есть небольшой шанс что среди ваших клиентов есть какие-то, использующие устаревшие версии Java или Python. Если у вас таких нет, то не о чем переживать. Если же есть — мои соболезнования.

Почему лучше рассчитывать на SNI?

1. Это просто. Вам не нужно постоянно держать в голове факты о выданных сертификатах. Для какого домена сертификат был выдан первым. К какому сертификату нужно добавлять еще домены. И так далее... Ни о чем таком со SNI не нужно думать.
2. Секреты остаются секретами. Если у вас для всех доменов один сертификат, то любой сможет очень легко увидеть весь список, независимо от вашего желания. Если же для каждого сайта свой сертификат, то такой проблемы нет.

Например, так можно посмотреть домены в сертификате Тематических Медиа:

```
true | openssl s_client -showcerts -connect habrahabr.ru:443 2>&1 |  
openssl x509 -text | grep -o 'DNS:[^,]*' | cut -f2 -d:
```

На момент написания статьи эта команда выведет подробный список всевозможных доменов ТМ:

```
habrastorage.org  
api.geektimes.ru  
api.habrahabr.ru  
geektimes.ru  
habrahabr.ru  
id.tmtm.ru  
lab.geektimes.ru  
m.geektimes.ru  
m.habrahabr.ru  
special.geektimes.ru  
special.habrahabr.ru  
www.geektimes.ru  
www.habrahabr.ru
```

Никакой секретности и никаких тайн. Вы этого хотите?

Установка Certbot

Если вы читаете этот текст из будущего, когда Certbot уже есть в Debian stable и Ubuntu без обиняков и оговорок, то всё просто:

```
apt-get install certbot
```

Либо используйте aptitude или другой пакетный менеджер вашего дистрибутива.

Установка в Jessie

Если у вас еще в ходу актуальный на конец 2016 года Debian stable "jessie", то всё лишь немного сложнее.

1. Нужно подключить Debian Backports, добавив строчку в /etc/apt/sources.list:

```
deb http://ftp.debian.org/debian/ jessie-backports main contrib non-free
```

2. Теперь можно устанавливать с указанием источника:

```
apt-get update
apt-get install certbot -t jessie-backports
```

(Раздел актуален пока только *stretch* не стал stable.)

Ubuntu версий ниже 16.10 (yakkety)

```
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install --upgrade letsencrypt
```

Дальше везде вместо certbot используйте letsencrypt.

Другой дистрибутив

Если у вас какой-то другой дистрибутив, то дополнительные инструкции по установке есть [на официальном сайте Certbot](#). Если обходиться без пакетного менеджера, то обычно установка сводится к...

```
wget -O /usr/local/bin/certbot-auto https://dl.eff.org/certbot-auto
chmod +x /usr/local/bin/certbot-auto
ln -s /usr/local/bin/certbot-auto /usr/local/bin/certbot
```

Везде ниже вместо команды certbot можно использовать команду certbot-auto.

Certbot и webroot

Мы будем получать сертификаты по методу *webroot* без перенастройки или остановки веб-сервера, под которым подразумевается nginx. Нам нужен какой-то каталог, в который certbot будет писать свои файлы, и какой должен быть доступен из сети удостоверяющему серверу согласно протокола ACME.

Чтобы не писать каждый раз длинную строку из опций, а еще лучше — не вспоминать о них, запишем основные настройки в файл конфигурации, который certbot ожидает найти в /etc/letsencrypt/cli.ini:

```
authenticator = webroot
webroot-path = /var/www/html
post-hook = service nginx reload
text = True
```

Последняя директива нужна чтобы избавить нас от прелестей и красотей ncurses, что нужно чтобы вы могли сравнить вывод команд здесь, в этой статье, и у себя.

Также нам нужно мягко перезагрузить nginx (без перерыва в обслуживании) при успешном обновлении сертификатов. Ничего не мешает в этот же момент перезапустить и другие сервисы вроде Postfix, использующие полученные сертификаты. Команды указываются через точку с запятой.

Если точка с запятой вызывает ошибку

Если вы видите такую ошибку:

```
letsencrypt: error: Unexpected line 14 in /etc/letsencrypt/cli.ini: post-hook = service nginx reload; serv
ice postfix reload
```

То вам нужно обновить python-configargparse. Ошибка была исправлена в 0.11.0.

Что будет делать Certbot

Ожидается что certbot будет создавать необходимые для проверки прав на домен файлы в подкаталогах ниже по иерархии к указанному. Вроде таких:

```
/var/www/html/.well-known/acme-challenge/example.html
```

Эти файлы должны будут быть доступны из сети на целевом домене по крайней мере по HTTP:

```
http://www.example.com/.well-known/acme-challenge/example.html
```

Для следующих проверок создадим какой-то такой файл:

```
mkdir -p /var/www/html/.well-known/acme-challenge
echo Success > /var/www/html/.well-known/acme-challenge/example.html
```

Регистрация в Let's Encrypt

Регистрацию нужно сделать только один раз:

```
certbot register --email me@example.com
```

Здесь ничего сложного.

Подготовим nginx к получению сертификатов

В общем случае для получения сертификата необходимо во всех блоках server добавить следующий блок до других блоков location:

```
location /.well-known {
    root /var/www/html;
```

```
}
```

Понятно, что вписывать для каждого сайта такой блок явно — это моветон, потому создадим файл `/etc/nginx/acme` с содержанием блока выше.

```
# cat /etc/nginx/acme
location /.well-known {
    root /var/www/html;
}
```

Затем для каждого домена и поддомена, для которых нужно получить сертификаты, в блоке `server` перед всеми блоками `location` укажем:

```
include acme;
```

Хосты-редиректоры (например, с голого домена на `www`) можно пропустить. ACME сервер обязан учитывать стандартную переадресацию. Подробнее об этом ниже.

Перезагрузим `nginx` и проверим что наш тестовый файл виден:

```
# service nginx reload
# curl -L http://www.example.com/.well-known/acme-challenge/example.html
Success
```

После проверки лучше удалить тестовый файл — `certbot` любит удалять за собой всё лишнее, а такой файл будет мешать и вызывать сообщение об ошибке (Unable to clean up challenge directory).

```
rm /var/www/html/.well-known/acme-challenge/example.html
```

Теперь у нас всё готово чтобы **получить наш первый сертификат**.

О переадресации с кодами 301 и 302

Как было уже сказано, ACME сервер Boulder учитывает переадресацию с кодами 301 и 302. В этом смысле не имеет значения где, в конечном счете, находятся файлы, требуемые для прохождения проверок. Переадресация возможна даже на нестандартные порты, без ограничений по конечному протоколу HTTP или HTTPS. Сами Let's Encrypt рекомендуют использовать переадресацию для создания единой точки проверки прав на домены.

Если вы можете получить эти файлы с помощью `curl` с ограничением в десять переадресаций, то и Boulder эти файлы увидит. Не должно быть никаких ограничений по IP адресам.

```
curl --location --max-redirs 10 http://example.com/.well-known/acme-challenge/example.html
```

Это удобно если у вас сложная структура переадресаций между разными версиями сайтов. Должно быть достаточно подключить тот блок с `location` только на основном сайте для получения сертификатов для всех остальных.

```
$ curl --head --silent --location --max-redirs 10 http://somewhere.example.net/... | grep ^HTTP
HTTP/1.1 301 Moved Permanently
HTTP/1.1 301 Moved Permanently
HTTP/1.1 200 OK
```

Проверка всегда начинается с запроса по протоколу HTTP на 80 порту.

Если у вас уже всё зашифровано...

Если у вас уже все сайты работают по HTTPS, то вся схема будет работать если у вас настроен переадресующий сервер на 80 порту, сохраняющий `$request_uri` в ответе.

Другое дело что можно сократить путь и подключить наш блок с `location` в умолчальном сервере для 80 порта, который делает переадресацию на HTTPS. Тогда не нужно будет ничего дописывать в конфиги отдельных сайтов.

Пример конфигурации такого переадресующего всё-подряд-на-HTTPS сервера:

```
server {
    listen server.example.com:80 default_server;

    include acme;

    location / {
        return 301 https://$host$request_uri;
    }
}
```

Такой конфиг стоит определить в `/etc/nginx/conf.d/default.conf`, в стороне от конфигов конкретных сайтов.

Сервер запускаем явно на внешнем IP чтобы не перенастраивать Apache на другой порт. Если для вас это не проблема, то указание имени сервера в директиве `listen` можно пропустить.

Если нужно получить сертификат для домена без сайта...

Типичный пример — сертификат для выделенных под SMTP или IMAP серверов, на которых вообще нет каких-то сайтов. Либо используйте универсальный переадресатор что выше, либо...

```
server {
    server_name smtp.example.com imap.example.com;
    listen server.example.com:80;

    include acme;

    location / {
        return 404;
    }
}
```

К сожалению, протокол ACME требует чтобы такой сервер был доступен во время каждой проверки. Это практически эквивалентно постоянной доступности, ввиду требования получения и обновления сертификатов без перезагрузки сервера. Не удаляйте такой конфиг после получения сертификата.

Если у вас только Apache2...

Если у вас Apache2, а перейти на всеми любимый nginx возможности нет, то... Добавьте следующие строчки в `/etc/apache2/conf-available/certbot.conf`:

```
Alias /.well-known/ /var/www/html/.well-known/
<Directory /var/www/html/.well-known/>
    Satisfy any
</Directory>
```

Затем

```
a2enconf certbot
mkdir -p /var/www/html/.well-known
```

```
service apache2 reload
```

И обязательно проверьте, так:

```
mkdir -p /var/www/html/.well-known/acme-challenge
echo Success > /var/www/html/.well-known/acme-challenge/example.html
curl -L http://localhost/.well-known/acme-challenge/example.html &&
rm /var/www/html/.well-known/acme-challenge/example.html
```

Существует много причин почему такая схема может у вас в Apache2 не заработать. Пары экранов текста не хватит чтобы описать их все. Не сердчайте — статья про nginx.

Получаем сертификаты

У Let's Encrypt есть лимиты на количество обращений за сертификатами, потому сначала попробуем получить необходимый сертификат в режиме для тестов:

```
certbot certonly --dry-run -d example.com -d www.example.com
```

В конце программа должна отчитаться об успешной работе:

```
The dry run was successful.
```

Теперь можно смело получать сертификат уже в самом деле. Не забудьте явно указать все необходимые поддомены, такие как `www`.

```
# certbot certonly -d example.com -d www.example.com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Starting new HTTPS connection (1): acme-v01.api.letsencrypt.org
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for example.com
http-01 challenge for www.example.com
Using the webroot path /var/www/html for all unmatched domains.
Waiting for verification...
Cleaning up challenges
Generating key (2048 bits): /etc/letsencrypt/keys/0001_key-certbot.pem
Creating CSR: /etc/letsencrypt/csr/0001_csr-certbot.pem
```

IMPORTANT NOTES:

- Congratulations! Your certificate and chain have been saved at /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will expire on 2017-04-01. To obtain a new or tweaked version of this certificate in the future, simply run certbot again. To non-interactively renew *all* of your certificates, run "certbot renew"

Ура! С получением сертификата закончено!

Если нужно добавить поддомен или домен в сертификат

Если вы вдруг забыли указать поддомен `www`, или вам нужно добавить другой домен или поддомен в сертификат (которых может быть до 100 в одном сертификате), то это легко сделать после получения сертификата. Просто запустите команду еще раз, добавив требуемое имя:

```
certbot certonly -d example.com -d www.example.com -d shop.example.com
```

Вам будет безальтернативно предложено добавить этот домен в сертификат. Если хочется избежать вопросов, то можно сразу указать одобряющий такое поведение ключ:

```
certbot certonly --expand -d example.com -d www.example.com -d shop.example.com
```

Операцию можно повторять.

Проверим полученный сертификат

Убедимся что полученный сертификат — именно тот, что нам нужен:

```
# openssl x509 -text -in /etc/letsencrypt/live/example.com/cert.pem
Certificate:
    Signature Algorithm: ...
    Validity
        Not Before: Jan  3 06:00:00 2017 GMT
        Not After : Apr  3 06:00:00 2017 GMT
    X509v3 extensions:
        ...
        X509v3 Subject Alternative Name:
            DNS:example.com, DNS:www.example.com
```

Или, если подробности вам не нужны:

```
cat /etc/letsencrypt/live/*/cert.pem | openssl x509 -text |
grep -o 'DNS:[^,]*' | cut -f2 -d:
```

Команда должна вывести список доменов в сертификате.

Установка и использование сертификатов

Certbot не перезаписывает сертификаты, а заменяет их ссылками на самые актуальные варианты сертификатов в определенном каталоге, одноименном с первым доменом сертификата (т.е. `cn`).

Давайте посмотрим что за файлы у нас есть:

```
# find /etc/letsencrypt/live/ -type l
/etc/letsencrypt/live/example.com/fullchain.pem
/etc/letsencrypt/live/example.com/chain.pem
/etc/letsencrypt/live/example.com/privkey.pem
/etc/letsencrypt/live/example.com/cert.pem
```

С этим знанием мы можем задать настройки SSL для nginx:

```
ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
```

Все потоки Разработка Администрирование Дизайн Менеджмент Маркетинг Научпоп



Войти

Регистрация

Как видите, `cert.pem` нигде в конфиге не используется, и это не ошибка. Для `nginx` он не нужен.

Полный рабочий пример конфига:

```
server {
    server_name www.example.com;
```



```
listen www.example.com:443 ssl; # default_server;
# выше можно добавить default_server для клиентов без SNI

ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;

ssl_stapling on;
ssl_stapling_verify on;
resolver 127.0.0.1 8.8.8.8;

# исключим возврат на http-версию сайта
add_header Strict-Transport-Security "max-age=31536000";

# явно "сломаем" все картинки с http://
add_header Content-Security-Policy "img-src https: data;; upgrade-insecure-requests";

# далее всё что вы обычно указываете
#location / {
#    proxy_pass ...;
#}
}
```

Конфиг для переадресации с голого домена без www:

```
server {
    server_name example.com;
    listen example.com:443 ssl;
    access_log off;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;

    ssl_stapling on;
    ssl_stapling_verify on;
    resolver 127.0.0.1 8.8.8.8;

    add_header Strict-Transport-Security "max-age=31536000";

    expires max;
    return 301 https://www.example.com$request_uri;
}
```

Подразумевается что вы используете какой-то локальный сервер для кеширования DNS запросов. Если это не так, то 127.0.0.1 в директиве resolver нужно заменить на IP используемого DNS сервера.

Настройки шифров и прочее подобное (ssl_dhparam, ssl_session_cache) лучше держать вне конфигов отдельных серверов.

Perfect Forward Secrecy

Если вы переживаете что Certbot может утащить ключи от вашего сертификата не смотря на [открытые исходные коды](#), а значит, в теории, какие-то злодеи смогут расшифровать весь трафик, то спешу вас успокоить. Если для соединения с вашим сайтом используются шифры из семейств DHE и ECDHE, то утечка ключа не позволит расшифровать трафик. В этих шифрах ключ сертификата используется только для подтверждения подлинности, и не используется в качестве ключа для шифрования. Все современные браузеры поддерживают эти шифры.

Если для ECDHE на эллиптических кривых ничего не нужно делать, то для DHE можно было бы использовать усиленные параметры. Лучше всего будет отключить DHE вообще.

[Если по какой-то причине без DHE вы не можете обойтись](#)

Продление сертификатов

Сертификаты выдаются *на три месяца*. Не на полгода, не на год, а лишь на три месяца. Естественно это вызывает вопросы. Нужно ли проходить всю эту процедуру через три месяца? Нужно ли это делать всегда до истечения веков? Может стоит всё-таки вложиться в платный сертификат чтобы забыть об этом всем и не вспоминать пару лет?

Но нет, не спешите искать платежные средства! Как и было обещано в начале статьи, с обновлением сертификатов проблем нет.

Если у вас Debian, то нужно лишь дописать к вызову certbot в /etc/cron.d/certbot ключ `--allow-subset-of-names`:

```
# последняя строка в /etc/cron.d/certbot
# было certbot -q renew, а надо
certbot -q renew --allow-subset-of-names
```

Если у вас Debian и systemd, то посмотрите эти инструкции.

Если у вас не Debian или нет файла, то добавим в crontab от root одну лишь строчку (`sudo crontab -e`):

```
42 */12 * * * certbot renew --quiet --allow-subset-of-names
```

Согласно рекомендаций Let's Encrypt следует пытаться обновить сертификаты два раза в день. Делать это нужно в случайным образом выбранную минуту того часа, а значит вам нужно заменить 42 в этой строке на другое число в диапазоне между 0 и 59. Либо вы можете поступить так как это делается в /etc/cron.d/certbot.

Как это работает

В этой команде ключ `--allow-subset-of-names` нужен чтобы Certbot пытался получить сертификаты для частичного набора доменов.

Например, были у вас на сервере сайты `www.example.com` и `shop.example.com`, проходящие под одним сертификатом, но потом вы перенесли `shop.example.com` на другой сервер. Если такой ключ не указать, то Certbot упадет с ошибкой при попытке подтвердить владение `shop.example.com`, не получив для вас вообще никакого сертификата. Сертификат истечет и ваш сайт уйдет в оффлайн. С этим ключом вы всё же получите сертификаты хотя бы для частичного набора доменов, оставив ваши сайты в сети.

Вот и всё

Если вам близки по духу `tee` и `sed`, то есть гораздо более короткая инструкция по настройке связки Let's Encrypt и nginx, при условии корректно настроенного `hostname`. Только копируй команды и вставляй.

Нашли ошибку? Напишите в личку, пожалуйста.

Теги: letsencrypt, nginx, certbot, debian, ubuntu, apache2

Хабы: Настройка Linux, Nginx, Серверное администрирование

↑ +44 ↓ 703 201k 94 Поделиться



131,7

Карма

0,0

Рейтинг

Алексей @alexkbs

Инженер-программист

Сайт Twitter Github ICQ Telegram

ПОХОЖИЕ ПУБЛИКАЦИИ

5 апреля 2013 в 16:50

Версионность конфигураций серверов на базе debian/ubuntu

+4

8,7k

59

17

23 августа 2012 в 19:42

Шпаргалка начинающего Debian/Ubuntu администратора по управлению пакетами

+77

272k

1272

64

21 марта 2012 в 16:15

Спасительная флешка на основе дистрибутива Linux Debian/Ubuntu

+29

19,7k

291

49

ЗАКАЗЫ

- Доработка сайта на PHP, ZendFramework1. JScript Data Encryption

1 200 ₺ за час • 3 отклика • 12 просмотров
- Настроить в Windows 10 разные программы по разным интернет каналам

1 000 ₺ за час • 5 откликов • 29 просмотров
- Настройка яндекс директ под инфопродукт

7 000 ₺ за проект • 5 откликов • 24 просмотра
- Разработка приложения на React Native


80 000 ₺ за проект • 6 откликов • 46 просмотров
- Отредактировать три информационные статьи

1 300 ₺ за проект • 3 отклика • 45 просмотров
- Больше заказов на Хабр Фрилансе

Реклама

Комментарии 94

НЛО прилетело и опубликовало эту надпись здесь

 Zolg

5 января 2017 в 10:34

#

🔖

📄

🔗


↑

+3

↓

А зачем пересоздавать CSR при обновлении скртификата ?!?!!!

НЛО прилетело и опубликовало эту надпись здесь

 Zolg

5 января 2017 в 11:18

#

🔖

📄

🔗

↑

+2

↓

плюньте вы на этот certbot (хотя наверняка в нем это настраивается).
в асте-tipe путь к csr и сертификату — параметры командной строки. засунули в cron и все.



Taragolis 5 января 2017 в 12:19



↑ +2 ↓

Конфигурационные файлы для genew сертификата лежат в директории (актуально для Ubuntu, возможно в других дистрибутивах путь может быть другой) /etc/letsencrypt/genewal в них есть все параметры, которые указываются ключами для certbot (и которые описаны в man'e)



alexkbs 5 января 2017 в 13:07



↑ +1 ↓

Отлично! Туда же можно прописать allow-subset-of-names для сертификатов с временными доменами.



alexkbs 6 января 2017 в 07:20



↑ +1 ↓

Используйте шифры, дающие perfect forward secrecy, и не переживайте о ключах. От них никакого толка для желающих почитать трафик в случае использования DHE и ECDHE.



motienko 5 января 2017 в 09:27



↑ -2 ↓

У Let's Encrypt есть проблема — отсутствие аналогов (бесплатный сертификат по асме).



Zolg 5 января 2017 в 10:33



↑ +1 ↓

imho certbot слишком перегружен, рекомендую асме-tiny.



surefire 5 января 2017 в 11:07



↑ 0 ↓

или асме-client



motienko 5 января 2017 в 11:23



↑ 0 ↓

у certbot есть отличная фишка — он запускается как http сервер, т.е. можно просто проксировать запросы well known к нему



Zolg 5 января 2017 в 14:36



↑ +5 ↓

на вкус на цвет: наличие в системе *еще одного* http сервера представляется фишкой сомнительной отличности



HunterNNm 5 января 2017 в 10:43



↑ +1 ↓

Я бы отметил еще, что certboot пока не умеет idn, хоть и заявлен. Так что владельцы доменов.рф,.рус пока не могут получить для них сертификат. Хотя тот же <https://github.com/Neilpang/acme.sh> это делает отлично



alexkbs 5 января 2017 в 10:50



↑ 0 ↓

Даже если указывать домен в Punycode?



HunterNNm 5 января 2017 в 10:54



↑ 0 ↓

Да. На гитхабе в issue обещали поправить в начале декабря, но так и не исправили. Пишет, что не знает доменную зону xn--p1ai. В acme.sh исправили оперативно



motienko 5 января 2017 в 11:22



↑ 0 ↓

в гите есть ветка, где проверку на idn выпилили, но пока оно не в master



HunterNNm 5 января 2017 в 11:40



↑ 0 ↓

Я и ее пробовал — та же беда. Может за последние недели 2-3 чего поменяли



motienko 5 января 2017 в 16:00



↑ 0 ↓

```
(venv)$certbot --version  
certbot 0.10.0.dev0
```

выпускает сертификаты для РФ



alexkbs 5 января 2017 в 11:43



↑ 0 ↓

В certbot#3614 пишут что вот-вот всё это будет.



anakhorein 5 января 2017 в 16:00



↑ 0 ↓

Вообще-т уже 16 дней как сделали, сам неделю назад накатил на рф, пруф.



Vladekk 5 января 2017 в 12:14



↑ +2 ↓

А если вы освоили докер, то можно практически ничего не настраивать, если использовать <https://github.com/jwilder/nginx-proxy> и <https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion> только указать имена хостов и мейл для регистрации сертификатов.



antonvn 5 января 2017 в 12:38



↑ +4 ↓

Для меня все восторги по поводу letsencrypt заканчиваются на необходимости поддерживать SSL не только на веб фронтах. Вы удивитесь, но в мире есть не только nginx и апач! Есть tomcat, винда с iis, exchange, микротики, разношерстное телекоммуникационное оборудование, iLo серваков, админки СХД и куча других сервисов и железок, куда сертификат можно поставить только вручную.



splav_asv 5 января 2017 в 12:56



↑ +3 ↓

Приземлять HTTPS на nginx, а дальше http.



GennPen 5 января 2017 в 12:58



↑ +1 ↓

Под винду с iis есть клиент letsencrypt-win-simple, который прописывается в планировщик.



motienko 5 января 2017 в 13:26



↑ +1 ↓

для «внутреннего» оборудования удобнее сделать свой CA, подписать сертификаты на 10 лет и добавить этот CA в браузеры сотрудников



foxmuldercp 6 января 2017 в 01:42



↑ 0 ↓

А в нормальных виндовых доменах CA и так есть, и ещё куча плюшек, от шифрования почты до vpn доступа по сертификату.



zelenin 5 января 2017 в 16:34



↑ 0 ↓

Ну ок. И в чем проблема?



Zolg 5 января 2017 в 20:52



↑ +2 ↓

А в чем проблема-то? Http используется только для подтверждения владения доменом, где вы дальше используете полученный сертификат — ваше дело.

Мы мх'ы все letsencrypt'ом отшили, обновляется автоматом ессно.

Для интРАнет сервисов логичнее свой CA поднять



navion 8 января 2017 в 15:56



↑ +1 ↓

Есть непубличные сервисы доступные определённому списку ip-адресов и/или через IPSec-туннель. К ним не сможет достигаться бот LE (его адреса зачем-то скрывают), а проверка через DNS не поддерживает наш NS и придётся делать это вручную.

Плюс всякие железки, вроде ASA с AnyConnect или контроллера Wi-Fi с captive-порталом, где тоже нужен нормальный сертификат, но нет встроенного клиента.

LE отличная штука для публичных сайтов, а для всего остального мне проще купить сертификат у GGSSL за \$10 и забыть про это на 3 года.



Erelecano 8 января 2017 в 17:43



↑ 0 ↓

> его адреса зачем-то скрывают

Правда что ли? А как можно скрыть от меня адреса с которых к моему веб-серверу обращаются? Вы глупости говорите.

Интересно вам этот ггssl заплатил за рекламу или вы добровольно и бесплатно рекламируете какую-то сомнительную шарашку у которой и не факт, что ЦА в доверенных основных систем?



navion 9 января 2017 в 17:22



↑ +1 ↓

| А как можно скрыть от меня адреса с которых к моему веб-серверу обращаются?

Вот только знать его надо перед подключением и не обязательно следующая попытка будет с того же адреса.

| Интересно вам этот ггssl заплатил за рекламу или вы добровольно и бесплатно рекламируете какую-то сомнительную шарашку у которой и не факт, что ЦА в доверенных основных систем?

У них брендированные сертификаты от Comodo по самым низкий ценам. Ещё они рассказывают всякое интересное тут.



ademaro 5 января 2017 в 13:43



↑ +2 ↓

Чтобы заработал Forward Secrecy я ещё добавляю в конфиг nginx'a

```
ssl_dhparam /etc/nginx/ssl/dhparam.pem;
```

Сгенерировать ключ можно так:

```
openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
```

Заодно можно включить http2 (добавив опцию в директиву listen):

```
listen 443 ssl http2;
```

Ну и ещё можем добавить

```
ssl_stapling on;
```

Этим мы позволяем серверу прикреплять OCSP-ответы, тем самым уменьшая время загрузки страниц у пользователей.

Что бы не прописывать это каждый раз, можно поменять шаблон certbot'a в файле

```
/etc/letsencrypt/options-ssl-nginx.conf
```



Acuna 5 января 2017 в 14:44



↑ 0 ↓

Затем для каждого домена и поддомена, для которых нужно получить сертификаты, в блоке server перед всеми блоками location укажем

А что в случае, если используются автоподдомены, например языковые?



alexkbs 5 января 2017 в 15:19

↑ 0 ↓

Они же указываются в директиве server_name, ведь так?



Acuna 5 января 2017 в 17:13

↑ 0 ↓

Не так) Имел ввиду именно генерацию динамических автоподдоменов.



motienko 5 января 2017 в 15:54

↑ 0 ↓

надо указывать весь список, wildcard сертификаты не выпускаются



Acuna 5 января 2017 в 17:14

↑ +1 ↓

Ууууу, ясно, большое спасибо, это реально по сути камушек в их огороде)



Borz 6 января 2017 в 00:43

↑ 0 ↓

у вас много языков на уровне автоподдоменов?



Acuna 6 января 2017 в 14:19

↑ 0 ↓

Видите ли-с в чем дело. Изначально в проекте предусматривали мультиязычность, и для простоты придумали систему, где все слова хранятся в простых текстовых файлах, которые очень легко переводить гугловским API, поэтому теперь в него можно с легкостью добавить любой язык, ибо для этого достаточно просто перевести языковой файл русского на любой другой язык, который поддерживает Google Translate. При таком раскладе стало очень интересно в числе прочих языков иметь всякие паджаби и латынь. Да это блажь, понимаю, однако лично я не вижу смысла урезать список всех языков до двух-трех только потому, что Let's Encrypt с какого-то перепугу не выдает Wildcard.



alexkbs 6 января 2017 в 15:45

↑ 0 ↓

Не выдает, но почему вам это мешает? Так как одним сертификатом не обойтись (у Google Translate больше 100 языков), лучше, из соображений размера сертификата, сразу получить отдельные под каждый код языка.

```
for langcode in $(cat ISO_639-1.txt); do
    certbot certonly -d $langcode.example.ru
done
```

Другой вопрос если вам нужно чтобы без SNI все работало. Только только \$300 в год на *.example.ru или уход от идеи поддоменов решат вопрос.



Acuna 7 января 2017 в 17:09

↑ +1 ↓

Спасибо, интересно. Хотя с другой стороны на GoGetSSL можно взять полноценный Wildcard и за немногим больше 70\$ кстати.

P. S. Кто бы мог подумать, что тема неожиданно превратится в настоящий Тостер) В таких ситуациях я нахожу весьма нужным возможность заносить комментарии в избранное.



alexkbs 7 января 2017 в 17:26

↑ +2 ↓

Тожe подумайте: может вам вообще не стоит так делать. Вдруг вам когда-нибудь захочется различать en_US и en_GB и en_AU.



Acuna 11 января 2017 в 20:23



0



Ахах, и такие предложения поступали, человека при этом чуть из окна не выкинули и зареклись никогда это не использовать, что бы там ни было. Ибо что-то, но использовать Английский (Австралия) — это уже какое-то извращение)



Borz 6 января 2017 в 20:58



0



тогда настройте hook, который при запросе "нового" языка будет расширять сертификат, добавляя в него поддомен и потом уже осуществлять обычный ответ с сертификатом. Не думаю, что это будет слишком большой оверхед



Acuna 7 января 2017 в 12:37



0



Да, благодарю, как раз ниже посоветовали оптимальное решение такого рода.



motienko 7 января 2017 в 00:42



0



в соседней ветке посоветовали вот это <https://caddyserver.com/>



Acuna 7 января 2017 в 17:05



0



Вы не поверите, как раз некоторое время назад рассматривал его, правда с с другой стороны (как замена Апачу, а лучше вообще NGINX), он действительно на удивление хорош, однако на англоязычном SO говорили, что пока он подойдет только для тестирования на локале, типа с нагрузкой в продакшене он не справится, поэтому и о замене им неншних решений говорить пока весьма рано. К сожалению не смог найти сходу ссылку на этот вопрос, однако очевидно, что фигни там не посоветуют, ибо за такое можно неслабо словить минусов (это, к тому же, практически единственная область, где их там можно словить).



Acuna 6 января 2017 в 15:08



0



P. S. Да и может у нас необходимость иметь десятки языков, может мы Красный Крест какой, почему нет?))



Ereleciano 6 января 2017 в 20:06



+1



<https://github.com/GUI/lu-resty-auto-ssl>

И генерируйте на лету, при обращении к домену :)



Acuna 7 января 2017 в 12:36



0



Ого! Вкуснотища какая, у меня аж слюни потекли)



Ereleciano 8 января 2017 в 02:02



-1



Приятного аппетита! Для Ubuntu 14.04 LTS у меня где-то рра есть для openresty под это дело, вдруг понадобится.



Acuna 11 января 2017 в 20:20



0



Ахах, ну да, спасибо) У меня (ну не у меня, конечно. На сервере) Дебиан (уже Джесси), однако я не знаю какой сюрприз может подготовить мне судьба, поэтому на решение для Убунты я бы тоже глянул бы... Да и для общественности можно оставить на сохранение.

Влепившему минус Вашему комменту я бы порекомендовал бы попытаться высказать Вам его претензии лично при встрече. Уверен, он даже из дома побоиться выйти, куда уж там что-либо высказывать...



Ereleciano 11 января 2017 в 20:24



0



<https://launchpad.net/~ernillew/+archive/ubuntu/openresty-for-le>

Ну собственно можете взять оттуда исходники пакета и собрать для Jessie себе, все будет попроще. Просто почему-то OpenResty не пакеты в дистрах, вот пришлось самому это сделать, пока было нужно.

А на минусы я внимания не обращаю, по моим комментариям есть какой-то анонимный фанат, ходит и минусит, фиг бы с ним. Мои комментарии полезны коллегам, а остальное меня не волнует.



Acuna 18 января 2017 в 18:05



↑ -1 ↓

Ясно, благодарю!

А так да, и сам иногда попадаю под раздачу. Тут имеется большое количество таких людей, и очень жаль на самом деле, что так получилось, что самая большая концентрация во всем интернете находится как раз на Хабре/Гиктаймсе. Возможно, всему виной возможность отрицательного голосования за комментарии. Ни на каком другом ресурсе нет такой возможности по подленькому показать свое отношение к человеку, не показывая своего лица. Поэтому имеем что имеем. Я сам программист, и в таких ситуациях думаю только об одном: «Хоть бы этот человек не был программистом, дабы не очернять эту светлую профессию»)



alexkuzko 5 января 2017 в 17:09



↑ 0 ↓

Может кто уже такое делал (в принципе, не должно быть особо сложно — но я сейчас гриппую, не могу собрать мозги в кучку...):

- 1) есть nginx фронт на публичном IP
- 2) есть nginx back на публичном IP
- 3) есть amazon route53
- 4) основной трафик идет на 1 и с него проксируется на 2
- 5) если 1 упал, то route53 обновляет DNS и запросы идут уже на 2

Вопрос в том как правильно запрашивать сертификаты ведь нужно их иметь и на 1 и на 2.

Допустимо ли иметь разные сертификаты (и позволит ли LE) для одних и тех же доменов, причем если в нормальном режиме, то оба запроса будут проходить через 1 (с точки зрения LE).

Пока что обдумываю вариант с редиректами на hostname 1 и 2 чтобы гарантированно направлять ACME на них, но это не отвечает на вопрос позволит ли LE два (а если три?!) сертификата для одного и того же домена и не отзовет ли он предыдущий сразу после выдачи/обновления последнего...



Acuna 5 января 2017 в 18:00



↑ +3 ↓

Это Вам на Тостер надо. Seriously. Вероятность получить ответ в комментариях к статьям крайне мала. Она есть, конечно, однако это может быть только случайный прохожий, который сталкивался с этой проблемой. На Тостере же люди специально подписываются на теги конкретно по технологиям.



Zolg 5 января 2017 в 20:57



↑ 0 ↓

Емнип, при выпуске нового сертификата старый автоматически НЕ отзывается.



Erelecano 5 января 2017 в 21:36



↑ +1 ↓

LE позволяет выпускать сертификаты совершенно спокойно.

У меня есть несколько фронтов с nginx'ами(RR DNS) на которых я таки на каждом получал сертификат отдельно, плюс на бэке с nginx'ом же свой сертификат для того же домена. Все прекрасно работает. При получении нового старый никто не отзывает автоматом, можете не беспокоиться.



Prototik 7 января 2017 в 10:42



↑ 0 ↓

Можно ещё просто скадывать сертификаты в сетевую шару, откуда они будут доступны нужным серверам, а автоматизацию выдачи, соо-но, оставить только в одном месте.



aib 5 января 2017 в 20:23



↑ -1 ↓

А это только меня напрягает, что у нас появился центр, в котором будут храниться все private ключи и который, со временем, за счет бесплатности, перетянет к себе пол интернета?



motienko 5 января 2017 в 21:28



↑ +2 ↓

он не хранит приватные ключи.
но «единая точка отказа» напрягает.



motienko 5 января 2017 в 21:35



↑ 0 ↓

Может быть и такое развитие событий:

- 1) браузеры в 2017 году начинают считать небезопасным сайты без шифрования (это уже запланировано);
- 2) все, у кого нет возможности купить сертификат, соскакивают на letsencrypt;
- 3) через годик он меняет условия выдачи бесплатных сертификатов (срок действия, либо платность, либо вообще пропадает);
- 4) у продавцов сертификатов — profit



Erelecano 5 января 2017 в 21:51



↑ +1 ↓

Про платность и пропажу вы просто бредите, потому что не соизволили почитать кто стоит за LE. А за LE стоит Mozilla, стоит Cisco, стоит Google, стоят OVH и Vultr, и так далее.

<https://letsencrypt.org/sponsors/>



motienko 5 января 2017 в 21:56



↑ 0 ↓

Конечно, прочитал еще до того, как сделал первый сертификат :)
Но это не отменяет вышесказанное.
Вот когда появится 2-3 таких CA, будет заметно интереснее.

ИМНО целью стоит внедрить повсеместное шифрование, 1) чтобы не все спецслужбы заглядывали в трафик, 2) чтобы запустить http2.



Erelecano 5 января 2017 в 22:00



↑ +1 ↓

Создать CA который попадет в доверенные в системах и браузерах сложно, а потому второй такой появится вряд ли. За LE стоят все крупные игроки рынка, каких только можно представить, некому встать за вторым таким же.

Безусловно целью был повсеместный переход на http/2 и соответственно https(как мы помним http/2 без ssl'a не существует), такой переход лично я поддерживаю двумя руками и с момента, как LE вышел из беты я по умолчанию всем клиентам делаю https с сертификатом от LE.



motienko 5 января 2017 в 23:10



↑ 0 ↓

Есть надежда, что со временем допилят проверку через DNS aka DANE (rfc7671), тогда для простого https не нужен будет letsencrypt



Erelecano 5 января 2017 в 23:16



↑ +1 ↓

Я в какой-то момент перестал верить, что DANE допилят, если честно. А LE реально стало спасением, автоматическое обновление сертификатов позволило мне просто забыть о том, что надо получать новые, тащить их на сервер, укладывать куда-то(не важно руками или оркестрацией какой-нибудь). Единственное что, я считаю, что certbot странен и неудобен, использую <https://github.com/hlandau/acme> и очень доволен.



alexkbs 6 января 2017 в 07:23



↑ 0 ↓

Чтобы спецслужбы не заглядывали вот так просто в ваш трафик, используйте DHE и ECDHE. Эта рекомендация верна для любых центров выдачи сертификатов.



TerAnYu 6 января 2017 в 00:46



↑ +2 ↓

Для Ubuntu 16.04 есть ppa, а то с сайта и с репозитория ставится только letsencrypt.

[PPA Certbot](#)

**Dil0ng** 6 января 2017 в 10:54 ++

0 ↓

У меня почему то на многих сайтах, где стоит сертификат от letsencrypt, доктор веб на некоторых компах блокируют сайты как ненадежный сертификат/ошибка сертификата, хотя сам сертификат установлен корректно и не имеет ошибок. проверял

**alexkuzko** 6 января 2017 в 11:26 #

↑ 0 ↓

Возможно не вся цепочка доверия установлена?

**Envek** 6 января 2017 в 12:15 #

↑ +1 ↓

Возможно, доктор веб имеет своё личное хранилище доверенных корневых/промежуточных сертификатов, куда корневые сертификаты от Let's Encrypt ещё не попали. Если SSL Test на этих сайтах не ругается на Chain Issues, то надо пинать поддержку доктора веба.

**Bluefox** 8 января 2017 в 12:19 #

↑ 0 ↓

А можно ли запустить chalange сервер не на 80м порту? У меня там бежит апач и при получении сертификатов раз в 3 месяца приходится останавливать апач, обновлять сертификаты, запускать апач. Было бы удобно если бы асме поддерживал альтернативные порты, например 81, 1880, 8080, 443

**alexkbs** 8 января 2017 в 12:55 #

↑ 0 ↓

Запустить можно, но ACME сервер все равно пойдет смотреть на 80-й.

**Bluefox** 8 января 2017 в 13:44 #

↑ 0 ↓

Ну в общем то я это имел ввиду. Можно ли заставить асме смотреть не 80.

**alexkbs** 8 января 2017 в 14:23 #

↑ 0 ↓

Не нужно. Используйте webroot и так:

```
Alias /.well-known/acme-challenge /var/www/html/.well-known/acme-challenge

<Directory "/var/www/html/.well-known/acme-challenge">
    Options None
    AllowOverride None
    Require all granted
    AddDefaultCharset off
</Directory>
```

В остальном все то же самое что в инструкциях в посте.

**Acid_Jack** 26 января 2017 в 11:27 #

↑ 0 ↓

Жаль. Я хотел повесить nginx'овский редирект 301 на 81-й порт, но не вышло. Всё равно придётся отдавать 80-й порт под верфикацию.

Хотя фраза «Переадресация возможна даже на нестандартные порты, без ограничений по конечному протоколу HTTP или HTTPS.» очень обнадежила.

P.S. Использую certbot 0.9.3 на Debian.

Может в 0.10.x можно переопределить порт, на который должен стучаться ACME?

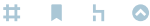
**alexkbs** 26 января 2017 в 12:08 #

↑ 0 ↓

С какой ошибкой не выходит?



Acid_Jack 26 января 2017 в 12:11



0 ↓

Failed authorization procedure. ***.*** (http-01): urn:acme:error:connection :: The server could not connect to the client to verify the domain :: Could not connect to validation-server.***.***:81

Если вешаю редирект на 80-й, всё ОК.



alexkbs 26 января 2017 в 12:13



↑ 0 ↓

Вы сами можете файлы по этому порту получить через curl? Как здесь

А если не с localhost, а откуда-то из сети?



Acid_Jack 26 января 2017 в 12:18



↑ 0 ↓

Да, могу. Запросы извне тоже нормально обрабатываются.

Думаю, что если порыбачить tcpdump'ом, то я увижу запросы от ACME-сервера на 80-м.



Acid_Jack 26 января 2017 в 13:02



↑ 0 ↓

А вот сейчас не понял. После энной попытки верификация прошла. Более того, убрал редирект, погасил nginx на сервере проверки и всё равно работает. Ничего не понимаю.



alexkbs 26 января 2017 в 13:11



↑ 0 ↓

ACME-сервер всегда сначала спрашивает на 80 порту. Куда вы его потом редиректите — ваше дело.



Acid_Jack 26 января 2017 в 13:45



↑ 0 ↓

Думаю, закавыка в редиректе nginx.

Использовал два идентичных варианта:

```
rewrite ^/.well-known/acme-challenge/(.*)$ http://validation-server.***.***:81$request_uri?permanent;
```

```
location ^~ /.well-known/acme-challenge/ {  
    return 301 http://validation-server.***.***:81$request_uri;  
}
```

Через curl запросы успешно доходят до validation-server, а от ACME болт.

Перевешиваю nginx на validation-server на 80-й (или standalone-режим на 80-й порт) и убираю ':81' из редиректа — всё работает.

Возможно, надо не редиректить, а проксировать запросы.



Erelecano 8 января 2017 в 13:16



↑ 0 ↓

А кто вам мешает использовать режим webroot?



Bluefox 8 января 2017 в 13:47



↑ 0 ↓

Спасибо за идею :)



Erelecano 8 января 2017 в 13:52



↑ +2 ↓

А, вообще, рекомендую в качестве клиента <https://github.com/hlandau/acme> и режим проксирования. Все же certbot убог.

НЛО прилетело и опубликовало эту надпись здесь



olen 11 января 2017 в 19:11



↑ 0 ↓

Спасибо за полезную и понятную статью.

А нет ли возможности параметризовать эти строки?

```
ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
```

...

Чтобы вместо example.com подставлялся соответствующий домен?



alexkbs 12 января 2017 в 03:42



↑ 0 ↓

Если вы хотите чтобы это происходило динамически, для любых доменов, то нет — так не получится сделать в nginx.



Erelecano 14 января 2017 в 17:23



↑ +1 ↓

nginx не принимает свою встроенную переменную `$server_name` в этих параметрах, так же, как не принимает ее в `error_log` (а вот в `access_log` принимает).

На самом деле очень не хватает такой возможности, но насколько я понимаю ее добавление утяжелит жизнь nginx'а (хотя лично на своих проектах я бы nginx с патчем дающим такую возможность поставил бы, это изрядно упростило бы жизнь).



motienko 16 января 2017 в 13:48



↑ 0 ↓

выше было как вариант



Erelecano 16 января 2017 в 16:39



↑ +1 ↓

Как вы могли заметить там мой же комментарий и есть.
Все же это несколько разные вещи, хотя и близкие.



Levhav 1 февраля 2018 в 05:43



↑ 0 ↓

Если вы читаете этот текст из будущего, когда Certbot уже есть в Debian stable и Ubuntu без обиняков и оговорок, то всё просто

Похоже что debian 9 и есть то будущее которого мы ждали.

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки

Неделя

Месяц

Docker is deprecated — и как теперь быть?

+68

32k

91

58

Анатомия GNU/Linux

+116

18,7k

272

64

PickPoint оценила, сколько товара украли из взломанных постаматов

+31

11,6k

5

45

Типовые ошибки Python-разработчиков на собеседованиях

+23

12,5k

102

37

Карты, деньги, Data Science: изучаем нескучные банковские данные [КВЕСТ]

Мегалост

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	Мерч