**Виды уязвимостей для базы данных PostgreSQL (ФСТЭК РОССИИ)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код** | **Дата обнаружения** | **Описание** | **Способы устранения** | **Уровень опасности (Оценка по CVSS 3.0)** |
| BDU:2020-05467 | 12.11.2020 | Уязвимость компонента client системы управления базами данных PostgreSQL связана с использованием криптографических алгоритмов, содержащих дефекты. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, реализовать атаку типа «человек посередине» | [Метод-1](https://www.postgresql.org/support/security/) | 8,1 |
| BDU:2020-05466 | 12.11.2020 | Уязвимость реализации мета-команды «gset» системы управления базами данных PostgreSQL связана с ошибками переключения контекста привилегий. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, выполнить произвольный код | [Метод-1](https://www.postgresql.org/about/news/postgresql-131-125-1110-1015-9620-and-9524-released-2111/) | 7,5 |
| BDU:2020-05465 | 12.11.2020 | Уязвимость компонента core server системы управления базами данных PostgreSQL связана с недостаточной защитой структуры запроса SQL. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, оказать влияние на целостность, доступность и конфиденциальность данных | [Метод-1](https://www.postgresql.org/support/security/) | 8,8 |
| BDU:2020-04047 | 29.04.2020 | Уязвимость системы управления базами данных PostgreSQL связана с выходом за границы памяти. Эксплуатация уязвимости может позволить нарушителю вызвать отказ в обслуживании | Обновление программного обеспечения до 9.6.10-astrase6 или более поздней версии | 4,3 |
| BDU:2020-01320 | 20.11.2019 | Уязвимость скрипта pg\_ctlcluster пакета postgresql-common связана с небезопасным управлением привилегиями. Эксплуатация уязвимости может позволить нарушителю повысить свои привилегии | [Метод-1](https://blog.mirch.io/2019/11/15/cve-2019-3466-debian-ubuntu-pg_ctlcluster-privilege-escalation/) | 7,8 |
| BDU:2019-03221 | 08.08.2019 | Уязвимость функции SECURITY DEFINER системы управления базами данных PostgreSQL связана с непринятием мер по защите структуры запроса SQL. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, выполнить произвольные SQL команды | [Метод-1](https://git.postgresql.org/gitweb/?p=postgresql.git;a=commitdiff;h=8673743)  [Метод-2](https://git.postgresql.org/gitweb/?p=postgresql.git;a=commitdiff;h=752fa3d) | 7,5 |
| BDU:2019-02385 | 20.06.2019 | Множественные уязвимости системы управления базами данных PostgreSQL вызваны переполнением буфера на стеке. Эксплуатация уязвимостей может позволить нарушителю, действующему удалённо, выполнить произвольный код | Обновление программного обеспечения до более поздней версии | 7,5 |

**Виды угроз для серверов (ФСТЭК РОССИИ)**

| **№ п/п** | **Идентификатор** | **Описание угрозы** | **Источники угрозы** | **Объекты воздействия** | **Нарушаемые свойства безопасности информации** |
| --- | --- | --- | --- | --- | --- |
|  | УБИ.016 | Угроза заключается в возможности передачи нарушителем дискредитируемому браузеру запроса на доступ к файловой системе пользователя вместо URL-запроса. При этом браузер выполнит этот запрос с правами, которыми он был наделён при запуске, и передаст данные, полученные в результате выполнения этой операции, нарушителю.  Данная угроза обусловлена слабостями механизма проверки вводимых пользователем запросов, который не делает различий между запросами на доступ к файловой системе и URL-запросами.  Реализация данной угрозы возможна в случае наличия у нарушителя привилегий на отправку запросов браузеру, функционирующему в дискредитируемой системе | Внешний нарушитель со средним потенциалом | Сетевое программное обеспечение | К |
|  | УБИ.173 | Угроза заключается в возможности неправомерного осуществления нарушителем массовой рассылки коммерческих, политических, мошеннических и иных сообщений на веб-сервер без запроса со стороны дискредитируемых веб-серверов.  Данная угроза обусловлена уязвимостями механизмов фильтрации сообщений, поступающих из сети Интернет.  Реализация данной угрозы возможна при условии наличия в дискредитируемом веб-сервере активированного функционала, реализующего различные почтовые сервера, службы доставки мгновенных сообщений, блоги, форумы, аукционы веб-магазинов, онлайн-сервисы отправки SMS-сообщений, онлайн-сервисы голосования и др. | Внешний нарушитель с низким потенциалом | Сетевое программное обеспечение | Д |
|  | УБИ.153 | Угроза заключается в возможности осуществления нарушителем опосредованного деструктивного программного воздействия на дискредитируемую систему большим объёмом сетевого трафика, генерируемого сторонними серверами в ответ на сетевые запросы нарушителя, сформированные от имени дискредитируемой системы. Генерируемый сторонними серверами сетевой трафик значительно превышает объём сетевых запросов, формируемых нарушителем.  Данная угроза обусловлена слабостями мер межсетевого экранирования дискредитируемой информационной системы, мер контроля подлинности сетевых запросов на сторонних серверах, а также слабостями модели взаимодействия открытых систем.  Реализация данной угрозы возможна при условии наличия у нарушителя:  сведений о сторонних серверах с недостаточными мерами контроля подлинности сетевых запросов;  сведений о сетевом адресе дискредитируемой системы;  специального программного обеспечения, реализующего функции генерации сетевых пакетов | Внутренний нарушитель с низким потенциалом  Внешний нарушитель с низким потенциалом | Информационная система, сетевой узел, системное программное обеспечение, сетевое программное обеспечение | Д |

**Рекомендации по развертыванию и настройке базы данных «PostgreSQL»**

**Установите PostgreSQL с помощью команд:**

*sudo apt-get update*

*sudo apt-get install postgresql postgresql-contrib*

**Стандартные настройки PostgreSQL**

СУБД PostgreSQL, установленная из официального репозитория Ubuntu, по умолчанию прослушивает localhost. Этот стандартный параметр блокирует автоматическое прослушивание открытых интерфейсов. Его можно изменить переопределением listen\_addresses в файле postgresql.conf.

Кроме того, файл pg\_hba.conf поддерживает только локальный адрес замыкания и соединения с сокетами домена Unix/Linux. Потому сервер не будет принимать соединения от внешних хостов.

*# Put your actual configuration here*

*# ----------------------------------*

*#*

*# If you want to allow non-local connections, you need to add more*

*# "host" records. In that case you will also need to make PostgreSQL*

*# listen on a non-local interface via the listen\_addresses*

*# configuration parameter, or via the -i or -h command line switches.*

*# DO NOT DISABLE!*

*# If you change this first entry you will need to make sure that the*

*# database superuser can access the database using some other method.*

*# Noninteractive access to all databases is required during automatic*

*# maintenance (custom daily cronjobs, replication, and similar tasks).*

*#*

*# Database administrative login by Unix domain socket*

*local all postgres peer*

*# TYPE DATABASE USER ADDRESS METHOD*

*# "local" is for Unix domain socket connections only*

*local all all peer*

*# IPv4 local connections:*

*host all all 127.0.0.1/32 md5*

*# IPv6 local connections:*

*host all all ::1/128 md5*

Эти параметры блокируют прослушивание публичных интерфейсов. Используя такие настройки и включив брандмауэр, вы сможете полностью защитить сервер от взаимодействия с публичными интерфейсами.

**Настройка удалённых подключений**

На этапе производства и при работе с конфиденциальными данными рекомендуется шифровать трафик PostgreSQL с помощью SSL (в дополнение к брандмауэру или виртуальной частной сети, VPN).

Можно использовать менее сложный вариант настройки безопасности сервера: для этого нужно включить брандмауэр на сервере баз данных и ограничить доступ к нему до заведомо безопасных хостов.

**1: Создание пользователя и БД**

Для работы вам понадобится тестовая БД и пользователь. Подключитесь к администратору СУБД, postgres, с помощью клиента psql. Флаг –i запускает интерактивный режим, что позволяет ввести учётные данные пользователя postgres; флаг –u позволяет указать пользователя.

*sudo -i -u postgres psql*

Теперь можно создать пользователя и пароль для него.

*CREATE USER 8host WITH PASSWORD 'password';*

**Примечание**: Замените условное имя пользователя и пароль более надёжными учётными данными.

Теперь нужно создать БД и передать все права на неё новому пользователю. С точки зрения безопасности пользователи должны иметь тот уровень доступа, который необходим им для работы, и только к необходимым ресурсам. То есть, если пользователь должен читать файлы service\_1, не нужно давать ему права на запись и исполнение файлов service\_1 и service\_2

*CREATE DATABASE 8hostdb OWNER 8host;*

Вы получите подтверждение:

*CREATE DATABASE*

Закройте оболочку СУБД:

*\q*

**2: Настройка UFW**

Для наилучшей безопасности, рекомендуется использовать встроенный брандмауэр UFW. Для этого его необходимо правильно настроить.

Проверьте состояние брандмауэра:

*sudo ufw status*

**Примечание**: Если брандмауэр отключен (inactive), включите его:

*sudo ufw enable*

После этого снова запросите состояние брандмауэра. Чтобы разблокировать SSH, введите:

*sudo ufw allow OpenSSH*

На данный момент брандмауэр поддерживает только сервис OpenSSH.

Теперь нужно разблокировать доступ к порту PostgreSQL и ограничить его только до заведомо безопасных хостов.

Приведенная ниже команда добавит правило, которое разблокирует порт PostgreSQL по умолчанию, 5432. Если вы изменили этот порт, не забудьте обновить его в следующей команде. Также вместо client\_ip\_address укажите IP-адрес клиента, которому нужен доступ к хосту. В случае необходимости повторно запустите эту команду, чтобы добавить другие IP-адреса безопасных клиентов.

*sudo ufw allow from client\_ip\_address to any port 5432*

Теперь нужно настроить PostgreSQL для прослушивания внешнего IP-адреса. Для этого нужно добавить две настройки: вход для подключения к хосту в файл pg\_hba.conf и параметр listen\_addresses в postgresql.conf.

**3: Настройка доступ к хосту**

**Добавьте настройки в файл pg\_hba.conf.**

*sudo nano /etc/postgresql/\*/main/pg\_hba.conf*

Добавьте строки host под закомментированным блоком, который описывает настройки нелокальных соединений. Также нужно добавить внешний IP-адрес сервера базы данных, чтобы быстро проверить настройку брандмауэра.

*host 8hostdb 8host client\_ip\_address/32 md5*

**4: Настройка listen\_addresses**

Теперь нужно настроить прослушивание адресов в файле postgresql.conf:

*sudo nano /etc/postgresql/9.5/main/postgresql.conf*

Найдите строку listen\_addresses и определите в ней адреса, которые нужно прослушивать. Укажите имя хоста или IP-адрес сервера БД. Убедитесь, что используете внешний IP-адрес сервера БД, а не клиента.

*#listen\_addresses = 'localhost' # what IP address(es) to listen on;*

*listen\_addresses = 'localhost,server\_ip\_address'*

Сохраните и закройте файл.

**5: Настройка логирования действий**

Данное действие позволит получить детальную информацию, в случае инцидентов.

Откройте конфигурационный файл:

*sudo vim /etc/postgresql/9.\*/main/postgresql.conf*

Допишите или раскомментируйте:

*log\_line\_prefix = '%t %c %u ' # time sessionid user*

*log\_statement = 'all'*

Управлять некоторыми параметрами можно прямо из сессии с СУБД. Например включение подробного

логирования:

*SELECT set\_config('log\_statement', 'all', true);*

По умолчанию, log-файлы хранятся по пути */var/log/postgresql/\**

**6: Перезапуск PostgreSQL**

Чтобы обновить настройки СУБД, перезапустите PostgreSQL.

*sudo systemctl restart postgresql*

Утилита systemctl не выводит результат некоторых команд. Чтобы убедиться, что сервер успешно перезапущен, запросите его состояние:

*sudo systemctl status postgresql*

Убедитесь, что сервис запущен.

**Дополнительные рекомендации**

Рекомендуется рассмотреть несколько дополнительных вариантов повышения безопасности **PostgreSQL** сервера:

* Присваивать роли пользователям для управления доступом в PostgreSQL.
* Использовать SSL-сертификат для PostgreSQL, который позволит шифровать данные.
* Использовать SSH-туннель для PostgreSQL, который защитит подключения клиентов, несовместимых с SSL.
* Использовать парольные политики в виде расширений ([1](https://github.com/postgres/postgres/tree/master/contrib/passwordcheck), [2](https://github.com/ykrysko/passwordcheck), [3](https://github.com/michaelpq/pg_plugins/tree/master/passwordcheck_extra), [4](https://github.com/devrimgunduz/passwordcheck_cracklib), [5](https://github.com/asio/password))

**Виды уязвимостей для Astra Linux (ФСТЭК РОССИИ)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код** | **Дата обнаружения** | **Описание** | **Способы устранения** | **Уровень опасности (Оценка по CVSS 3.0)** |
| BDU:2020-02012 | 29.01.2020 | Уязвимость системы управления ЕПП службы Astra Linux Directory (ALD) связана с ошибкой авторизации для локального пользователя. Эксплуатация уязвимости может позволить нарушителю получить доступ к конфиденциальным данным, а также вызвать отказ в обслуживании | Для Astra Linux:  Обновление программного обеспечения (пакета ald) до 1.7.81+ci20191120131074686 или более поздней версии | 5,7 |
| BDU:2020-02010 | 29.01.2020 | Уязвимость инструмента для ввода Astra Linux SE/CE в домен MS Windows AD astra-winbind связана с ошибкой входа пользователя в домен. Эксплуатация уязвимости может позволить нарушителю получить доступ к конфиденциальным данным, нарушить их целостность, а также вызвать отказ в обслуживании | Для Astra Linux:  Обновление программного обеспечения (пакета astra-winbind) до 1.30 или более поздней версии | 4,8 |
| BDU:2020-02007 | 29.01.2020 | Уязвимость системы управления ЕПП службы Astra Linux Directory (ALD) связана с ошибкой выходом операции за границы памяти. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, получить доступ к конфиденциальным данным, нарушить их целостность, а также вызвать отказ в обслуживании | Для Astra Linux:  Обновление программного обеспечения (пакета fly-adminprinter) до 1.99.12 или более поздней версии | 6,3 |
| BDU:2020-03660 | 23.05.2020 | Уязвимость системы администрирования astra-openvpn-server связана с ошибкой при обработке данных. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании, путем изменения стандартного пути сохранения файлов | Для Astra Linux:  Обновление программного обеспечения (пакета fly-admin-openvpn-server) до 0.3.07 или более поздней версии | 5,3 |
| BDU:2020-03658 | 17.04.2020 | Уязвимость настройки безопасности astra-safepolicy связана с недостатком механизма проверки вводимых данных. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании | Для Astra Linux:  Обновление программного обеспечения (пакета astra-safepolicy) до 1.0.88+ci10 или более поздней версии | 7,5 |
| BDU:2019-04069 | 26.10.2019 | Уязвимость компоненты сервиса FreeIPA операционной системы Astra Linux связана с ошибкой при обработке содержания дескрипторов файла. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, выполнить произвольный код или вызвать отказ в обслуживании | Для Astra Linux:  Обновление програмного обеспечения (пакета astra-freeipa) до 1.32 или более поздней версии | 9,1 |

**Рекомендации по развертыванию и настройке Astra Linux**

**Установка и обновление программного обеспечения**

Устанавливайте только минимально необходимое ПО. Установка и настройка только по делу и с помощью установщика yum и dnf. Проверьте все установленное ПО и удалите ненужные пакеты:

*yum list installed*

*yum list packageName*

*yum remove packageName*

Используйте только **официальные** и доверенные репозитории пакетов.

Не используйте **нешифрованные** протоколы FTP, Telnet, Rlogin, Rsh.

**Отключайте** неиспользуемые сервисы на своем сервере, если в данный момент удаление сервиса вам не подходит.

Чтобы проверить список всех сервисов, используйте команду:

*systemctl list-unit-files --type=service*

Чтобы отключить и убрать службу из автозагрузки в CentOS используется systemctl:

*systemctl stop service*

*systemctl disable service*

Всегда держите в актуальном состоянии установленное программное обеспечение на вашем сервере. Вовремя обновленное ПО, защитит вас от известных уязвимостей. Вы можете настроить автоматическое обновление системы, чтобы каждый раз не выполнять это вручную.

*yum update — обновление системы*

**Защита SSH, авторизация по ключам**

Для безопасной авторизации на сервере, используйте приватный ключ. Сгенерируйте ключ с помощью утилиты ssh-keygen:

*ssh-keygen*

осле генерации, вы можете подключиться с помощью данного ключа к серверу, через ssh-клиент.

Создайте дополнительного пользователя и выполняйте команды через sudo.

*sudo groupadd sudo* – создать группу sudo

*adduser webmaster* – создать пользователя

*passwd webmaster* – поменять пароль

*usermod -aG sudo webmaster* — добавить пользователя в группу sudo

В файле /etc/sudoers добавьте строку:

*%sudo ALL=(ALL:ALL) ALL*

Отключите в настройка ssh-сервера, авторизацию через root и по логину/паролю:

*nano /etc/ssh/sshd\_config*

И изменяем значение указанных ниже строк на:

*PermitRootLogin no*

*PasswordAuthentication no*

*ChallengeResponseAuthentication no*

*UsePAM no*

*AuthenticationMethods publickey*

*PubkeyAuthentication yes*

Смените стандартный порт для ssh. Порт по умолчанию для ssh, в первую очередь будет подвержен взлому.

Чтобы поменять стандартный порт на ssh, в конфигурационном файле /etc/ssh/sshd\_config замените значение в строке:

*Port 22 # Заменить на любой незанятый порт*

**Регулярная смена паролей в Linux**

Если вы все же используете пароли для авторизации в Linux, то настройте срок действия пароля через утилиту chage.

Чтобы проверить, сколько дней будет действовать пароль для любого пользователя, используйте команду:

*chage -l root*

Чтобы изменить срок действия пароля (например 9 дней), воспользуйтесь командой:

*chage -M 9 root*

Для **блокировки** пользователей при вводе неправильного пароля, воспользуйтесь утилитой fail2ban (она позволяет реализовать аналог групповых политик блокировки учетных записией в Windows). Fail2ban позволяет защитить сервер Linux и службы от подбора паролей.

Для установки **fail2ban** выполните:

*yum install epel-release fail2ban -y*

Конфигурационный файл fail2ban расположен в /etc/fail2ban/jail.conf. Настройки, которые мы затронем:

bantime = 600 — время бана при блокировке

maxretry = 3 – количество попыток неправильного ввода пароля, после которых пользователь будет заблокирован

findtime = 600 – временной отрезок, в которой нужно авторизоваться пользователю.

Остальные настройки используются по желанию. Дополнительно вы можете настроить отправку отчетов к себе на почту.

Создайте файл /etc/fail2ban/jail.local и добавьте туда следующие строки:

*[DEFAULT]*

*ignoreip = 127.0.0.1/24*

*[ssh-iptables]*

*enabled = true*

*port = ssh*

*filter = sshd*

*action = iptables[name=sshd, port=ssh, protocol=tcp]*

*logpath = /var/log/secure*

После изменения конфигурационного файла, перезапустите сервис.

После чего сохраните файл и перезапустите сервис **fail2ban**:

*systemctl restart fail2ban*

Таким образом, мы настроили блокировку пользователей для **ssh.**

Таким образом можно настраивать блокировку и для других сервисов.

**Повышение защиты сервера с помощью SELinux**

SELinux имеет 3 **режима** работы:

* Enforcing
* Permissive
* Disabled

В режиме **enforsing** SELinux применяет свою политику в системе и следит за несанкционированным доступом со стороны пользователей. Все попытки регистрируются в логах.

В режиме **permissive** политика SELinux не применяется, но вся информация фиксируется в логах, после чего вы можете их проанализировать. Данный режим полезен при настройке и отладке системы.

И **disabled** соответственно вообще отключает SELinux и ее политика не применяется вообще.

**Конфигурационный** **файл** SELinux — /etc/selinux/config

Проверить статус: *sestatus*

Рекомендуется настроить SELinux, исходя из задач, выполняемых на сервере, используя официальную [документацию](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/index).

**Настройка брандмауэра Firewalld**

Откройте только минимально необходимый набор портов на сервере, которые действительно нужны для работы. Например, для работы web-серверов достаточно открыть 80 и 443. При работе с **PostgreSQL** Отключите возможность подключения с удаленных серверов на порт 5432 (если ваша БД используется только локально, не включена в кластер Galera и не испольуется в сценариях репликации).

Чтобы проверить открытые порты на сервере, используйте утилиту **netstat**:

*netstat -tulpn*

**Удаление X Windows**

**Не используйте** на своем сервере систему **X** **Windows**. Удалите все ПО, которое связанно с этой системой, нет необходимости использовать это на Linux сервере:

*yum group remove "GNOME Desktop"*

*yum group remove "KDE Plasma Workspaces"*

*yum group remove "Server with GUI"*

*yum group remove "MATE Desktop"*

**Защита ядра Linux**

Используйте настройки в файле /etc/sysctl.conf для обеспечения дополнительной безопасности во время загрузки ядра Linux.

Включите **execshield**:

*kernel.exec-shield=1*

**Отключите** IP-маршрутизацию:

*net.ipv4.conf.all.accept\_source\_route=0*

Включите **защиту** от **спуфинга**

*net.ipv4.conf.all.rp\_filter=1*

**Включите** игнорирование **широковещательных** **запросов**

*net.ipv4.icmp\_echo\_ignore\_broadcasts=1*

*net.ipv4.icmp\_ignore\_bogus\_error\_messages=1*

**Включите** регистрацию **подозрительных** пакетов

*net.ipv4.conf.all.log\_martians = 1*

**Права на файлы в Linux**

Запустите поиск файлов без пользователя и группы, такие файлы потенциально несут угрозу и могут быть использованы злоумышленниками. Чтобы найти такие файлы, используйте команду:

*find /директория -xdev \( -nouser -o -nogroup \) -print*

Если файлы будут найдены, измените на них владельца и группу.

Установите флаг chattr на нужные файлы, чтобы защитить их от модификации. Ни один пользователь, не сможет изменить файл, пока данный флаг установлен. Например:

*chattr +i /etc/mysript.sh*

Не устанавливайте слишком высокие права на директории и файлы, например, стандартные права для web:

* 755 для директорий
* 644 для файлов

В некоторых случаях, могут быть исключения, но всегда внимательно подходите к этому вопросу. Запретите пользователям выставлять разрешение 777. Вы можете найти такие файлы:

*find /home -type f -perm 777*

**Защита Web-сервера**

Если вы используете сервер под web, позаботьтесь о настройке web-сервисов. Закройте просмотр содержимого директорий через Options -Indexes, а также добавьте X-frame.

*Header always append X-Frame-Options SAMEORIGIN*

Данный параметр, запретит открывать страницу во фрейме, это предотвратит встраивание контента вашего сайта на другие сайты.

Отключите показ версии вашего web-сервера.

Для **nginx** в конфигурационном файле в секцию http пропишите:

*server\_tokens off;*

Так же настройте лимиты по выполнению скриптов и их размер, это обезопасит вас от мелких атак.

Установите на свой сайт и почту SSL сертификат, чтобы соединение в браузере и при отправке почты, **шифровалось**.

**Резервное копирование сервера Linux**

Необходимо настроить регулярное резервное копирование, чтобы в случае инцидентов, критически важная информация оставалась доступной.

[Примеры настройки резервного копирования для разных сервисов.](https://winitpro.ru/index.php/2019/09/13/linux-backup-scripts-to-cloud-storage/)

**Рекомендации по развертыванию и настройке Nginx**

**Отключить любые нежелательные модули nginx**

Чтобы отключить определенные модули, вам нужно перекомпилировать nginx. Мы рекомендуем отключить все ненужные модули, так как это минимизирует риск потенциальных атак за счет ограничения разрешенных операций.

Для этого используйте опцию configure во время установки. В примере ниже мы отключаем модуль autoindex, который генерирует автоматические списки каталогов, а затем перекомпилируем nginx.

*# ./configure --without-http\_autoindex\_module*

*# make*

*# make install*

**Отключить nginx server\_tokens**

По умолчанию директива server\_tokens в nginx отображает номер версии nginx. Этот номер непосредственно виден на всех автоматически генерируемых страницах ошибок, и также присутствует во всех ответах HTTP в заголовке сервера.

Это может привести к раскрытию информации — неавторизованный пользователь может получить информацию о используемой вами версии nginx. Вы должны отключить директиву server\_tokens в файле конфигурации nginx, установив *server\_tokens off*.

**Контроль ресурсов и ограничения**

Чтобы предотвратить потенциальные DoS-атаки на nginx, вы можете установить ограничения на размер буфера для всех клиентов. Вы можете сделать это в файле конфигурации nginx, используя следующие директивы:

* client\_body\_buffer\_size – используйте эту директиву, чтобы указать размер буфера тела запроса клиента. Значение по умолчанию — 8 КБ или 16 КБ, но рекомендуется установить его равным 1 КБ: client\_body\_buffer\_size 1k.
* client\_header\_buffer\_size – используйте эту директиву, чтобы указать размер буфера заголовка для заголовка запроса клиента. Размер буфера 1k подходит для большинства запросов.
* client\_max\_body\_size – используйте эту директиву, чтобы указать максимально допустимый размер тела для клиентского запроса. Директивы 1k должно быть достаточно, но вам нужно увеличить ее, если вы получаете загрузку файлов методом POST.
* large\_client\_header\_buffers – используйте эту директиву, чтобы указать максимальное количество и размер буферов, которые будут использоваться для чтения больших заголовков клиентских запросов. Директива large\_client\_header\_buffers 2 1k устанавливает максимальное количество буферов в 2, каждый с максимальным размером 1k. Эта директива будет принимать URI данных размером 2 КБ.

**Отключить любые нежелательные методы HTTP**

Рекомендуется отключить любые методы HTTP, которые не используются и которые не обязательно должны быть реализованы на веб-сервере. Если вы добавите следующее условие в блок местоположения файла конфигурации виртуального хоста nginx, сервер разрешит только методы GET, HEAD и POST и отфильтрует такие методы, как DELETE и TRACE.

*location / {*

*limit\_except GET HEAD POST { deny all; }*

*}*

**Установить ModSecurity для веб-сервера nginx**

ModSecurity — это модуль с открытым исходным кодом, который работает как брандмауэр веб-приложения (WAF). Его функции включают в себя фильтрацию, маскировку идентификатора сервера и предотвращение атак с нулевым байтом. Модуль также позволяет выполнять мониторинг трафика в режиме реального времени.

**Установка и настройка access и error logs nginx**

Журналы access и error logs nginx включены по умолчанию и расположены соответственно в logs/error.log и logs/access.log. Если вы хотите изменить местоположение, вы можете использовать директиву error\_log в файле конфигурации nginx. Вы также можете использовать эту директиву, чтобы указать журналы, которые будут записываться в соответствии с их уровнем безопасности. Например, уровень crit заставит nginx регистрировать критические проблемы и все проблемы, которые имеют более высокий уровень, чем уровень crit. Чтобы установить уровень crit, установите директиву error\_log следующим образом:

*error\_log logs/error.log crit;*

Вы можете найти полный список уровней error\_log в [официальной документации nginx.](http://nginx.org/en/docs/http/ngx_http_log_module.html#log_format)

Вы также можете изменить директиву access\_log в файле конфигурации nginx, указав расположение не по умолчанию для журналов доступа. Наконец, вы можете использовать директиву log\_format для настройки формата записываемых сообщений, как описано в документации по nginx.

**Мониторинг журналов Access и Error**

Если вы постоянно отслеживаете и управляете лог-файлами nginx, вы можете лучше понимать запросы, отправленные на ваш веб-сервер, а также заметить любые обнаруженные ошибки. Это поможет вам обнаружить любые попытки атаки, а также определить, что вы можете сделать, чтобы оптимизировать производительность сервера.

Вы можете использовать инструменты управления журналами, такие как **logrotate**, чтобы обновлять и сжимать старые журналы и освобождать дисковое пространство. Кроме того, модуль ngx\_http\_stub\_status\_module предоставляет доступ к основной информации о состоянии

**Настройте Nginx для включения заголовков безопасности**

Чтобы дополнительно защитить свой веб-сервер nginx, вы можете добавить несколько различных заголовков HTTP.

**X-Frame-Options**

Заголовок HTTP-ответа X-Frame-Options, используется чтобы указать, разрешено ли браузеру отображать страницу в frame или iframe . Это может предотвратить атаки с помощью **clickjacking**.

Для этого добавьте следующий параметр в файл конфигурации nginx в разделе server:

*add\_header X-Frame-Options "SAMEORIGIN";*

**Strict-Transport-Security**

HTTP Strict Transport Security (HSTS) — это метод, используемый веб-сайтами для объявления того, что доступ к ним должен осуществляться только через безопасное соединение (HTTPS). Если веб-сайт объявляет политику HSTS, браузер должен отклонить все HTTP-соединения и запретить пользователям принимать незащищенные SSL-сертификаты. Чтобы добавить заголовок HSTS на ваш сервер nginx, вы можете добавить следующую директиву в ваш раздел сервера:

*add\_header Strict-Transport-Security "max-age=31536000; includeSubdomains; preload";*

**CSP and X-XSS-Protection**

Политика безопасности контента (CSP) защищает ваш веб-сервер от определенных типов атак, включая атаки с использованием Cross-site Scripting (XSS) и атаки с использованием data injection. Вы можете реализовать CSP, добавив следующий пример заголовка Content-Security-Policy (обратите внимание, что фактический заголовок должен быть настроен в соответствии с вашими уникальными требованиями):

*add\_header Content-Security-Policy "default-src 'self' http: https: data: blob: 'unsafe-inline'" always;*

Заголовок HTTP X-XSS-Protection поддерживается IE и Safari и не требуется для современных браузеров, если у вас есть строгая политика безопасности содержимого. Однако, чтобы предотвратить XSS в случае более старых браузеров (которые еще не поддерживают CSP), вы можете добавить заголовок X-XSS Protection в раздел вашего сервера:

*add\_header X-XSS-Protection "1; mode=block";*

**Настройте SSL и Cipher Suites**

Конфигурация nginx по умолчанию позволяет использовать небезопасные старые версии протокола TLS (согласно официальной документации: ssl\_protocols TLSv1 TLSv1.1 TLSv1.2). Это может привести к таким атакам, как атака BEAST. Поэтому мы рекомендуем вам не использовать старые протоколы TLS и изменять конфигурацию для поддержки только новых, безопасных версий TLS.

Для этого добавьте следующую директиву в раздел server файла конфигурации nginx:

*ssl\_protocols TLSv1.2 TLSv1.3;*

Так же, рекомендуется добавить строку:

*ssl\_prefer\_server\_ciphers on;*

Эта директива позволит принять решение о том, какие шифры использовать, на стороне сервера, а не на стороне клиента.

**Регулярно обновляйте ваш сервер**

Обновления также часто включают новые функции безопасности и улучшения. На сайте **nginx.org** вы можете найти рекомендации по безопасности в специальном разделе и новости о последних обновлениях на главной странице.

**Проверьте свою конфигурацию с Gixy**

**Gixy** — это инструмент с открытым исходным кодом, который позволяет вам проверить ваш веб-сервер nginx на наличие типичных ошибок конфигурации. После того, как вы подготовите свою конфигурацию nginx, всегда полезно проверить ее с помощью Gixy.