|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное бюджетное образовательное учреждение высшего образования  «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» |
| Методические рекомендации по проекту № 42  Мобильное приложение Ак Барс  Для:  «Команды бекенд-разработчиков» |
|  |
| Выполнили:  Студенты 3 курса  Группы 181-352  Захаров В.А.  Констанитов Д.А. |

Москва – 2020

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код** | **Дата обнаружения** | **Описание** | **Способы устранения** | **Уровень опасности (Оценка по CVSS 3.0)** |
| BDU:2020-05726 | 05.03.2020 | Уязвимость библиотеки Django для языка программирования Python связана с недостаточной защитой структуры SQL-запроса. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, выполнить произвольный код | [Метод-1](https://www.djangoproject.com/weblog/2020/mar/04/security-releases/) | 8,8 |
| BDU:2020-01767 | 02.08.2019 | Уязвимость модуля HTMLParser функции django.utils.html.strip\_tags фреймворка для веб-разработки Django связана с ошибкой медленного оценивания больших входных данных, которые содержат большие последовательности неполных HTML объектов. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании | Для Django:  Обновление программного обеспечения до 2:2.2.4-1 или более поздней версии | 7,5 |
| BDU:2020-01459 | 15.12.2019 | Уязвимость фреймворка для веб-приложений Django связана с некорректной проверкой электронного адреса после трансформации символов Юникода в регистр, что дате возможность отправить токен сброса. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, оказать воздействие на целостность данных | Для Django:  Обновление программного обеспечения до 2:2.2.10-1 или более поздней версии | 7,5 |
| BDU:2019-01260 | 04.01.2019 | Уязвимость библиотеки Django для языка программирования Python связана с недостаточной нейтрализацией специальных элементов в выходных данных, используемых нижестоящим компонентом. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании | Для python-django:  Обновление программного обеспечения до более поздней версии | 6,5 |
| BDU:2019-03594 | 26.07.2019 | Уязвимость функций django.contrib.postgres.fields.HStoreField и django.contrib.postgres.fields.JSONField фреймворка для веб-разработки Django связана с ошибкой преобразовании и поиска ключей, а так же поиска индексов. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, оказать воздействие на целостность данных, получить несанкционированный доступ к защищаемой информации, а также вызвать отказ в обслуживании | Для Django:  Обновление программного обеспечения до 2:2.2.4-1 или более поздней версии | 9,8 |
| BDU:2020-01759 | 02.08.2019 | Уязвимость функции jango.utils.encoding.uri\_to\_iri фреймворка для веб-разработки Django связана с неправильным кодированием последовательностей октетов UTF-8, которое может привести к чрезмерному использованию памяти из-за рекурсии. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании | Для python-django:  Обновление программного обеспечения до более поздней версии | 7,5 |
| BDU:2020-01765 | 02.08.2019 | Уязвимость функции django.utils.text.Truncator методов chars() и words() фреймворка для веб-разработки Django связана с ошибочной передачей значения True аргументу html. Эксплуатация уязвимости может позволить нарушителю, действующему удаленно, вызвать отказ в обслуживании | Для Django:  Обновление программного обеспечения до 2:2.2.4-1 или более поздней версии | 7,5 |

**Распространенные угрозы для фреймворка «Jango»**

**Межсайтовый скриптингs (XSS)**

XSS (Cross-Site Scripting — «межсайтовый скриптинг») — тип атаки на веб-системы, заключающийся во внедрении в выдаваемую веб-системой страницу вредоносного кода (который будет выполнен на компьютере пользователя при открытии им этой страницы) и взаимодействии этого кода с веб-сервером злоумышленника. Является разновидностью атаки «Внедрение кода».

Специфика подобных атак заключается в том, что вредоносный код может использовать авторизацию пользователя в веб-системе для получения к ней расширенного доступа или для получения авторизационных данных пользователя. Вредоносный код может быть вставлен в страницу как через уязвимость в веб-сервере, так и через уязвимость на компьютере пользователяs.

**Межсайтовая подделка запроса (CSRF)**

CSRF (cross-site request forgery — «межсайтовая подделка запроса», также известна как XSRF) — вид атак на посетителей веб-сайтов, использующий недостатки протокола HTTP. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер (например, на сервер платёжной системы), осуществляющий некую вредоносную операцию (например, перевод денег на счёт злоумышленника). Для осуществления данной атаки жертва должна быть аутентифицирована на том сервере, на который отправляется запрос, и этот запрос не должен требовать какого-либо подтверждения со стороны пользователя, которое не может быть проигнорировано или подделано атакующим скриптом.

Основное применение CSRF — вынуждение выполнения каких-либо действий на уязвимом сайте от лица жертвы (изменение пароля, секретного вопроса для восстановления пароля, почты, добавление администратора и т. д.). Также с помощью CSRF возможна эксплуатация отражённых XSS, обнаруженных на другом сервере.

**SQL инъекции**

Внедрение SQL-кода (SQL injection) — один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Внедрение SQL, в зависимости от типа используемой СУБД и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах.

**Кликджекинг**

Кликджекинг (Clickjacking) — механизм обмана пользователей интернета, при котором злоумышленник может получить доступ к конфиденциальной информации или даже получить доступ к компьютеру пользователя, заманив его на внешне безобидную страницу или внедрив вредоносный код на безопасную страницу. Принцип основан на том, что поверх видимой страницы располагается невидимый слой, в который и загружается нужная злоумышленнику страница, при этом элемент управления (кнопка, ссылка), необходимый для осуществления требуемого действия, совмещается с видимой ссылкой или кнопкой, нажатие на которую ожидается от пользователя. Возможны различные применения технологии — от подписки на ресурс в социальной сети до кражи конфиденциальной информации и совершения покупок в интернет-магазинах за чужой счёт.

**Методы устранения распространенных угроз для фреймворка «Jango»**

**Межсайтовый скриптингs (XSS)**

Использовать систему шаблонов Django, которая защищает от большинства XSS атак, экранируя определенные символы, считающиеся "опасными" в HTML.

**Межсайтовая подделка запроса (CSRF)**

Механизм защиты заключается в добавлении тега шаблона {% csrf\_token %} в вашу форму. Этот токен будет отображен в вашем HTML как показано ниже, со значением, уникальным для каждого запрашивающего форму пользователя.

<input type='hidden' name='csrfmiddlewaretoken' value='0QRWHnYVg776y2l66mcvZqp8alrv4lb8S8lZ4ZJUWGZFA5VHrVfL2mpH29YZ39PW' />

Django генерирует уникальный для пользователя/браузера токен и отклоняет все формы, которые не содержат его или содержат его неверное значение.

**SQL инъекции**

Не использовать прямые SQL запросы в коде. Вместо этого, использовать ORM, который обеспечивает экранирование и безопасность выполнения запросов.

**Кликджекинг**

Использовать встроенную в Django защиту от кликджекинга в виде промежуточного програмного обеспечения (middleware) X-Frame-Options, который поддерживается браузерами и может запретить отображение страницы внутри <iframe>.

**Общие советы по использованию фреймворка «Jango»**

1. Wappalyzer - это расширение, которое идентифицирует программное обеспечение, работающее на нашем веб-сайте, что позволяет некоторым ботам использовать известные ошибки, прежде чем вы сможете исправить ваш веб-сайт, по этой причине рекомендуется защитить веб-сайт от этого типа расширений. В нашем случае django обнаруживается «csrfmiddlewaretoken», имя которого можно изменить и информацию о котором можно расширить, следуя последующему руководству hide-django-from-wappalyzer.

2. Одна из сильных сторон django - это панель администрирования, к которой можно получить доступ, используя URL-адрес по умолчанию \* / admin. Чтобы скрыть эту панель и избежать атак методом перебора, можно изменить этот URL-адрес. Для этого достаточно изменить файл urls.py и подставить:

# Default admin url

url(r'^admin/', admin.site.urls),

# Replace with following url

url(r'^my\_secure\_admin/', admin.site.urls),

3. Другой способ защиты нашей панели администрирования - сделать ее доступной только из сети, в которой хранится сервер, так как следует помнить, что эта панель не предназначена для конечных пользователей.

4. Обеспечить деактивацию DEBUG в производственной среде, иначе может быть предоставлен большой объем информации посетителям с плохими намерениями. Помните, что если вы оставите DEBUG равным False, вы должны настроить переменную ALLOWED\_HOSTS.

5. Использование защиты CRSF в формах, которые изменяют данные, с помощью встроенного метода промежуточного программного обеспечения CSRF «Промежуточное ПО CSRF и тег шаблона обеспечивают простую в использовании защиту от подделок межсайтовых запросов», для пользователя необходимо активировать его в настройках в качестве еще одного промежуточного программного обеспечения: 'django.middleware.csrf.CsrfViewMiddleware' и использовать тег шаблона в формах:

<form action="" method="post">{% csrf\_token %}

6. Проверять все данные, которые мы получаем в формах Django.

7. Разрешить доступ к просмотрам страниц, таким как «Логин» и «Админ», используя только протокол https.

8. При подозрении, что наш сайт подвергается атаке, можно использовать использовать такие методы, как: django-admin-honeypot, который создает нам поддельного администратора, сохранит журнал и уведомит администраторов о неудачных попытках доступа.

9. Использование шаблонов django вместо обычного html, так как это защитит нас от большинства XSS-атак.

10. Один из способов проверить имеющиеся ошибки безопасности - это использовать команду:

# With which we will obtain a detailed report about different errors and how to improve them.

python manage.py check --deploy

11. Другой способ проверить сайт Django, нет доступа к нашему серверу: ponycheckup, который предоставляет отчет об основных проблемах безопасности и способах их решения.