УДК 004.4

Способы защиты информации при создании сайтов

Палаш Борис Викторович
Хакасский государственный университет им. Н.Ф. Катанова
Студент

Научный руководитель:
Голубничий Артем Александрович
Хакасский государственный университет им. Н.Ф. Катанова
старший преподаватель кафедры программного обеспечения вычислительной техники и автоматизированных систем

Аннотация
При разработке сайтов важной задачей является обеспечение безопасности данных пользователей. В данной статье будут рассмотрены простые способы защиты от самых популярных методов несанкционированного доступа.
Ключевые слова: web, создание сайтов, sql, инъекции, защита

Ways to protect information when creating websites

Palash Boris Viktorovich
Katanov Khakass State University
student

Scientific adviser:
Golubnichiy Artem Aleksandrovich
Katanov Khakass State University
senior lecturer department of computing software and automated systems

Abstract
When developing websites, an important task is to ensure the security of user data. This article will discuss simple ways to protect against the most popular methods of unauthorized access.
Keywords: web, website development, sql, injection, protection

При разработке сайтов важной задачей является обеспечение безопасности данных пользователей. В данной статье будут рассмотрены простые способы защиты от самых популярных методов несанкционированного доступа.

Для этого необходимо рассмотреть самые основные способы незаконного получения информации.

Первым и самым популярным методов является SQL injection – это способ взлома, при котором самыми подверженными частями являются

POST и GET запросы в которые помимо основных данных добавляются дополнительные, искажающие sql запрос. Самым простым способом обнаружения такой уязвимости, является наличие специальной ошибки при отправке данных одним из методов, и наличии в них специального символа. Для примера был взят тестовый сайт 0861.ru с возможность детального просмотра новостей которые открываются по GET запросу типа «0861.ru/?news=1». При переходе по данному адресу пользователю будет показана подробная информация по новости с id равным единице. В случае добавления специального символа «’» в запрос, сайт начнет выдавать предупреждение, показанное на рисунке 1.1 это, будет сигналом к тому, что в данном случае есть уязвимость.

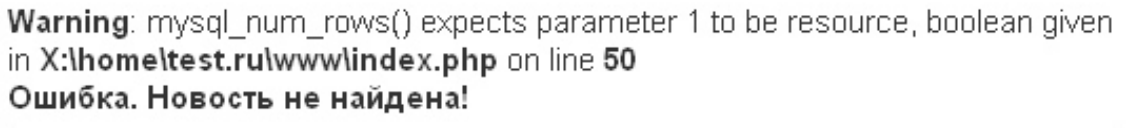

Рисунок 1.1 – пример ошибки в случае SQL уязвимости

Видя такую ошибку, злоумышленник понимает, что данный запрос не проходит специальные проверки и может использовать для изменения, получения или загрузки необходимых данный в базу сайта.

Самый простой способом защиты информации от такого рода взлома это проверка всех входящих данных и отсеивание ненужных параметров. Для этого в языке программирования PHP существует функция «mysqli_real_escape_string» предназначенная для экранирования всех специальных символов для последующего использования в sql запросе [1].

Второй способ защиты — это экранирование самой переменной в sql запросе и использование подготовленных выражений. Экранирование переменной в самом sql запросе является более простым вариантом чем использование специальных функций языка php и хорошо подходит для целочисленных данных. Пример такого экранирования представлен на рисунке 1.2.

Рисунок 1.2 – пример экранирования переменной в sql запросе

Использование подготовленных выражений является еще одним способом защиты информации. Основным принципом работы таких выражений является создание шаблона sql запроса и выполнение его после передачи входных параметров.

Главным преимуществом подготовленных выражений, помимо обеспечения безопасности, является уменьшение кода, так как данный шаблон достаточно подготовить единожды и можно использовать не определенное количество раз передавая разные входные параметры как показано на рисунке 1.3.

```
$username=mysqli_real_escape_string($con, $username);
$stmt = $con->prepare( query: "SELECT * FROM account WHERE `username`=?")
$stmt->bind_param( types: "s", &var1: $username);
$stmt->execute();
$stmt->store_result();
```

Рисунок 1.3 – пример подготовленного выражения

Единственной проблемой таких выражений является потребление ресурсов сервера, где располагается сайт, так как существует необходимость хранить шаблоны всех действующих выражений. При использовании же обычных sql запросов шаблон стирается сразу после использования.

Вторым наиболее популярным методом получения несанкционированного доступа к информации является метод перебора паролей и пользователей. Хотя данный метод и кажется довольно простым, он остаётся довольно популярным по сей день. Большинство начинающих разработчиков сайтов мало времени уделяют данному вопросу и считают его малозначимым. С данной проблемой разработчики борются с момента появления самых первых сайтов с возможностью авторизации.

Самый простой и древний способ — это использование паролей большой длины, разных регистров и специальных символов. На данный момент такой способ защиты сильно устарел, компьютеры сильно шагнули вперед и могут перебирать по десятку и сотне тысяч различных комбинаций паролей в час. А различные прокси сервера делают бесполезными различные блокировки переборов по ip и другим параметрам.

Одним из современных методов защиты от перебора это наличие специализированного ключа, который невозможно симулировать программно.

Один из таких генераторов ключей является «Google recaptcha», данный сервис предоставляет возможность размещения уникального ключа, Представленного на рисунке 1.4, на сайте клиента для ввода которого требуется человек [2].

Рисунок 1.4 – пример ключа, генерируемого сайтом «Google recaptcha»

Использование такого способа проверки передаваемых данных на сайт значительно замедляет взлом с использованием перебора, что дает администратору сайта вовремя среагировать на потенциальную угрозу.

Используя данные способы защиты информации вместе, разработчик в состоянии обеспечить минимальный набор проверок для обеспечения минимальной защиты сайта. Что для малых проектов может стать более чем достаточным вариантом.

Библиографический список

1. База знаний URL: https://www.php.net (дата обращения 04.05.2019).
2. reCAPTCHA URL: https://www.google.com/recaptcha (дата обращения 04.05.2019).