Анализ инструментов защиты данных пользователей на примере игровых компаний

Бродович Б.Е.

гр.160801

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Рабцевич В.В. – старший преподаватель каф.ИКТ

Аннотация. В материалах доклада рассматриваются проблемы безопасности игровых веб-приложений. Был проведен анализ существующих решений для защиты с учетом существующих видов атак.

Ключевые слова: кибербезопасность, игровое приложение, веб-приложение

Введение. Создателям цифровых видеоигр постоянно приходится сталкиваться со взломом своих продуктов, который базируется на взаимодействии с исходным кодом (например, путем реверсного инжиниринга) либо других составляющих. При этом прослеживается закономерность между популярностью товара компании и его вероятностью быть взломанными [1].

Основная часть.

Проведем анализ различных атак на пользовательскую базу данных. Согласно принципу Керкгоффса [2] при оценке надёжности необходимо предполагать, что злоумышленник знает об используемой системе шифрования всё, кроме ключей. В данном случае ключами являются пароли пользователей, которые хранятся в виде хешей, а сама СУБД (система управления базами данных), соответственно, была взломана и целиком попала к злоумышленнику.

Самый простой метод взлома это подбор пароля. Существует два подхода: это атака по словарю и brute-force атака (простой лексикографический перебор всех возможных вариантов). Процесс подбора пароля описанными методами показан ниже на рис. 1. От данных атак сложно защититься, но есть способы сделать их менее эффективными.

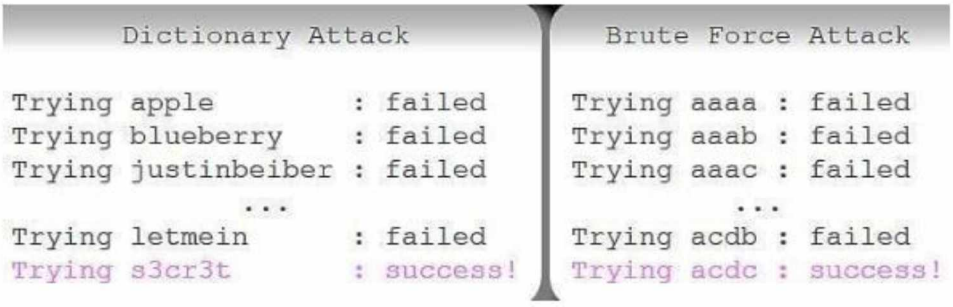

Рисунок.1. Взлом пароля с помощью атаки по словарю и brute-force атаки

Метод взлома через таблицу заключается в создании специальной таблицы, где заранее вычислены значения хешей для различных паролей. При взломе хеша происходит поиск соответствующего ему исходного пароля в этой таблице.

Обратный поиск по таблице — это эффективный метод взлома при больших объемах БД. Он позволяет злоумышленнику применять перебор паролей одновременно, создавая таблицу для поиска пользователей с соответствующими значениями хеша. Эффективность увеличивается, если несколько пользователей имеют одинаковые пароли.

Последний способ взлома – это использование так называемых радужных таблиц. Радужные таблицы – это вариант обычных таблиц, содержащих цепочки данных, полученные через хеширование и редукцию. Они требуют меньше памяти и могут взламывать пароли до восьми символов, зашифрованные с помощью MD5.

Анализ объекта защиты и выбор инструментов для обеспечения безопасности

Остановимся подробнее на особенностях данных, которые необходимо шифровать при разработке многопользовательской игры. В БД при регистрации нового пользователя заносится новый объект, содержащий хеш пароля, так называемую соль (затравку для функции хеширования) и токен пользователя – еще один хеш. Основная информация о пользователе — это логин, электронный адрес почты, хэш пароля (элемент passwordHash), соль хеша пароля (элемент r) и токен (элемент token).

Сейчас существует множество алгоритмов хеширования данных, каждый из которых имеет свои плюсы и минусы. Наиболее популярные из них:

1. **SHA-256 (Secure Hash Algorithm 256-bit):** этот алгоритм является наиболее часто используемым. Он создает 256-битное хеш-значение фиксированной длины и известен своей безопасностью и скоростью.
2. **SHA-1 (Secure Hash Algorithm 1):** SHA-1 также широко применяется, но стоит отметить, что он менее безопасен, так как использует 160-битное хеш-значение.
3. **MD5 (Message Digest Algorithm 5):** MD5 является одним из наиболее известных алгоритмов хеширования и широко используется для проверки целостности данных.

Для обеспечения безопасности при использовании функций хеширования рекомендуется добавлять затравку (salt), которая представляет собой случайную строку, добавляемую перед или после хешируемого сообщения (порядок не важен). Применение затравки обеспечивает ряд преимуществ, таких как: защита от атак по таблицам, усложнение перебора, повышение криптостойкости.

Таким образом, злоумышленник не может предугадать, какая затравка будет использована в каждом хеше пароля. Это влечет за собой невозможность построения радужных и иных таблиц для комбинаций всех возможных паролей со всеми возможными

затравками. Стоит также отметить, что затравку нет смысла шифровать так как это усложнит процесс аутентификации пользователей с точки зрения приложения.

Для повышения уровня защищенности при использовании пары «логин - пароль» в игровых приложениях, было решено создавать для каждого пользователя по токену. Токен в данном конкретном случае – это зашифрованная пара: логин и хеш пароля.

При регистрации пользователю выдается специальный хеш – секретный API ключ, который не следует разглашать. Этот ключ используется в каждом запросе к сервису, таким образом идентифицируя клиента.

Для обеспечения безопасности RESTful запросов используется защищенное соединение и передача токена вместо «логина-хеша пароля». Токен создается однажды: при регистрации и не меняется при смене пароля. Для его создания применяются алгоритмы с техникой «key stretching», увеличивающей криптостойкость даже слабых паролей.

Формирование политики безопасности:

1) **Минимальная длина пароля:** важно ограничивать минимальную длину пароля. В сетевых приложениях, включая игровые, слишком короткие пароли не обеспечивают надежную защиту. Обычно рекомендуется устанавливать минимум в 8 символов, что является разумным компромиссом между безопасностью и удобством для пользователей.

2) **Политика парольной защиты**:

а) Объединение ошибок в логине и пароле: при попытках авторизации следует рассматривать ошибку в логине и ошибку в пароле как единую пару. Это предотвращает перебор логинов и паролей через интерфейс.

б) Восстановление пароля через временные токены: при запросе на восстановление пароля пользователю следует предоставить ссылку с уникальным временным токеном. Токен должен быть привязан к конкретному аккаунту, чтобы злоумышленник не мог изменить чужой пароль. Время действия токена должно быть ограничено (например, 15 минутами).

в) Аннулирование токена после успешной авторизации: когда пользователь восстановил пароль, токен должен быть аннулирован. Это предотвращает его повторное использование.

г) Не отправлять новый пароль по почте: никогда не следует отправлять новый пароль пользователю по электронной почте. Вместо этого используйте временные токены для восстановления пароля.

д) Использование новой затравки для хеширования нового пароля: при изменении пароля рекомендуется обновлять затравку для хеширования. Это повышает безопасность хранения паролей в базе данных.

Заключение. Таким образом, в статье рассматриваются проблемы повышения защищенности игрового веб-приложения с учетом различных видов атак на пользовательскую БД и проведен анализ существующих решений для обеспечения надежной парольной защиты.

Список литературы

1. *Информационная безопасность в игровой индустрии [Электронный ресурс].. – Режим доступа : https://habr.com/ru/articles/695912/. – Дата доступа : 27.10.2022.*

2. *Н.Е.Губенко, А.В.Сирант Сравнительный анализ и выбор алгоритмов хеширования для организации парольной защиты игровых приложений /Н.Е.Губенко, А.В.Сирант. – Донецк: ДНТУ.*

Analysis of user data protection tools using the example of gaming companies

Brodovich B.E.

gr.160801

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Rabcevich V.V.

Annotation. The materials of the report address the security issues of a gaming web application. An analysis of existing security solutions was carried out, taking into account existing types of attacks.

Keywords: cybersecurity, gaming application, web application