Н. П. Цыганенко, Н. А. Жиляк
(БГТУ, Минск)

БЕЗОПАСНАЯ ФОРМА ХРАНЕНИЯ ПАРОЛЕЙ В БАЗЕ ДАННЫХ

Заводя учетную запись на каком-либо ресурсе в сети Интернет, пользователь меньше всего хочет, что бы ей могло воспользоваться постороннее лицо. Очевидно, что пароли пользователей в базе данных (БД) нельзя хранить в открытом виде, иначе злоумышленнику достаточно взломать БД и извлечь необходимую информацию.

На сегодняшний день большое количество Интернет-ресурсов хранят не сами пароли, а их хеши. Хеширование – преобразование по детерминированному алгоритму входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования называются хеш-функциями, а их результаты называют хешем.

Существуют различные алгоритмы хеширования (MD5, SHA-1, SHA-2, SHA-3), отличающиеся своими хеш-функциями и длиной выходного хеша. Наиболее популярным алгоритмом является MD5, его использует почти каждый интернет-ресурс. Но из-за некоторых недостатков его использование для данных целей не рекомендуется. Лучше воспользоваться одной из реализаций SHA. Так в SHA-3 используется алгоритм, который делает практически невозможным появление коллизий, что существенно повышает сложность подбора исходных данных [1].

Если злоумышленник украдет хеши паролей из БД и узнает алгоритм по которому они были получены, то сможет составить так называемую радужную таблицу для поиска паролей по хешам.

Радужная таблица – специальный вариант таблиц поиска, использующий механизм разумного компромисса между временем поиска и занимаемой памятью. По ним таблицам хакер может быстро узнать пароли, удовлетворяющие определенным условиям. Условия могут быть разные: только латинские буквы, буквы и символы, специальные символы.

Не бывает полностью надежных методов обеспечения безопасности какой-либо информации. Но можно организовать защиту таким образом, что злоумышленник откажется от взлома из-за большой трудоемкости.

Одним из методов затруднения процесса взлома является применение соли при хешировании. Соль – это уникальный набор случайных символов, которые добавляются к хешу пароля. Одним из наиболее часто встречаемых вариантов соли является строка символов переменной длины. После добавления соли полученная последовательность снова хешируется и в БД записывается хеш и соль. Если злоумышленник получит хеши паролей, соли и алгоритм получения хеша, то ему придется строить несколько радужных таблиц в лучшем случае. Таким образом, соль не влияет на устойчивость единичного пароля, а повышает криптостойкость системы в целом. Если злоумышленник получит только хеши паролей, то это очень сильно замедлит подбор даже единичного пароля.

Другим методом является замедление скорости получения хеша. Для этого пароль проходит через цикл функций хеширования. Злоумышленнику при составлении радужной таблицы требуется воспользоваться таким же циклом функций, что значительно замедлит ее составление. Главное в таком способе получения хеша то, чтобы скорость алгоритма не зависела от реализации. Например, есть алгоритм, который генерирует хеш входной последовательности за полсекунды. Для рядового пользователя такая задержка несущественна, но для взломщика составление радужных таблиц займёт слишком много времени и ресурсов [2].

При проектировании архитектуры и модели безопасности будущего Интернет-ресурса, необходимо находить баланс между ценностью хранимой информации и затратами на обеспечение ее безопасности. Главной рекомендацией является разработка собственных методов хеширования и обеспечение защиты от возможности получения их третьими лицами.

Литература

1. Петцольд, Ч. Код/ Чарльз Петцольд. – Москва: Русская редакция, 2001. – 512 с.

2. Фергюсон, Н. Практическая криптография/ Нильс Фергюсон, Билл Шнайер. – Москва: Вильямс, 2005 – 424 с.