УДК 65.07

Черкесова Ж. Ж.
студентка 4 курса кафедры «Информационная безопасность»
института информатики, электроники и компьютерных технологий
КБГУ
Россия, г. Нальчик
Карданов З. С.
студент 4 курса кафедры «Информационная безопасность»
института информатики, электроники и компьютерных технологий
КБГУ
Россия, г.Нальчик

МЕТОДЫ ХРАНЕНИЯ ПАРОЛЕЙ

Аннотация: *Данная статья посвящена исследованиям, лежащим в области информационной безопасности, и касается изучения методов хранения паролей. В статье определены основные методы хранения паролей.*

Ключевые слова: *информационная безопасность, пароль, хранение, хэш, шифрование.*

Abstract: *This article is devoted to research in the field of information security, and concerns the study of methods for storing passwords. The article defines the main methods of storing passwords.*

Key words: *information security, password, storing, hash, encryption.*

Методы хранения паролей:

1) В открытом виде;
2) В виде хэш -значения;
3) В зашифрованном виде.

1) С одной стороны это просто ужасно. Только представьте себе, что злоумышленник получил доступ к вашему компьютеру всего на одну минуту

и этого будет достаточно, чтобы все пароли оказались у него. Так хранят пароли, например, браузеры Google Chrome и Firefox. Почему разработчики, для которых вопрос защиты информации является одним из ключевых, поступают именно так? На самом деле причина такого решения вполне понятна: браузер не может предусмотреть все возможные способы проверки пароля, поэтому недостаточно хранить хэш или какую-то другую информацию. Всегда найдется сайт, который для аутентификации выберет новую хэш функцию или вообще предоставит свою.

Для того чтобы не совсем очернять указанные браузеры, стоит отметить, что в них реализована функция «Мастер - пароля». Используя мастер-пароль, все хранящиеся пароли зашифровываются и хранятся уже в зашифрованном виде. Если пользователь захочет воспользоваться паролем из базы, то ему придется ввести мастер-пароль, обычно один раз в течение сессии работы с браузером. Такое хранение более безопасно, но возможна худшая ситуация — скомпрометировав такой пароль, пользователь лишается всех своих паролей.

В качестве примера программы, которая просто хранит пароль в открытом виде — достаточно популярный интернет - мессенжер Pidgin. В ответ на вопрос, почему пароль храниться именно так, авторы программы отвечают: «If someone else can access your files and you can’t trust them not to misuse stored sensitive data, don’t store the sensitive data.» («Если кто-то еще имеет доступ к вашим файлам, и вы не можете положиться на то, что он не будет использовать ваши конфиденциальные данные, то не стоит их вообще хранить.»).

Плюсы хранения паролей в открытом виде:

1. Подтверждение подлинности (проверка, что пара логина и пароля совпадает с парой в таблице) очень простое – сравниваются строки.
2. Забытые пароли можно восстановить – пароль легкодоступен, если указан логин.

Минусы хранения паролей в открытом виде:

1. Во-первых, любой с доступом к файлу (или могущий производить выборку из таблицы) получает немедленный доступ ко всем паролям. Работник с законным доступом к файлу может распечатать файл или отправить информацию по электронной почте, и – вуаля! – все пароли скомпрометированы.

2. Вторая проблема заключается в том, что во время обмена аутентификационной информацией пароль видим в сети. Если везде не используется защищенная связь, пароль будет виден при прохождении по сети. Например, даже если веб - приложение использует SSL (уровень защищённых сокетов) для отправки пароля, пароль по-прежнему виден, когда сервер веб - приложения выбирает информацию из удаленной базы данных. Результаты запроса передаются по сети незашифрованными.

2) Криптографическая хеш-функция является необратимой функцией. Хеш-функция принимает входные данные любой длины и генерирует уникальные выходные данные постоянной длины. Например, если пароль (любой длины) хешируется криптографической хеш-функцией MD5, результатом будет 128 битное число, однозначно соответствующее паролю. Криптографические хеши работают не только на паролях – если криптографический хеш двух файлов идентичен, то два файла идентичны.

В последние годы в связи с ростом вычислительных мощностей некоторые криптографические хеш-функции больше не рекомендуется использовать (MD4, MD5, SHA1). Однако их допустимо применять для хеширования паролей. Или же измените код так, чтобы он использовал SHA2.

При хранении хешированных паролей пароль хешируется (прогоняется через алгоритм хеширования), и полученный хеш хранится вместо пароля. Чтобы сравнить пароли, захешируйте указанный пароль с помощью той же самой хеш-функции и сравните результаты. Если хеши совпадают, пароли совпадают.

Прелесть необратимой функции заключается в том, что невозможно вычислить пароль на основе хеша. Хешированные пароли неустойчивы к атаке перебором – при наличии словаря и хеша пароля хакер может вычислить хеши всех слов в словаре, сравнить слова с хешем пароля и узнать пароль. Надежные пароли (содержащие буквы, цифры и специальные символы) помогают защититься от атак перебором.

3) Лучший подход к хранению паролей (и единственная обоснованная альтернатива, если пользователям надо иметь возможность восстанавливать пароли) – шифрование паролей перед их сохранением.

Данный подход основан на обладании тайной. Тайной является алгоритм шифрования или ключ, используемый вместе с современным алгоритмом шифрования.

Шифрование паролей – обратимая операция. Тайна используется для искажения пароля, и та же самая тайна может быть использована для восстановления оригинального пароля. Когда пользователь задает пароль, сохраненный пароль расшифровывается с помощью тайны, и пароли сравниваются. Альтернативный подход – зашифровать предоставленный пароль с помощью тайны и сравнить две искаженные версии – совпадение показывает, что предоставлен правильный пароль.

Плюсы шифрования с помощью тайны:

1. Забытый или утерянный пароль можно восстановить.
2. Только одну тайну (алгоритм или ключ) надо хранить безопасно.
3. Для многопользовательских распределенных приложений при использовании шифрования надо передавать незашифрованный пароль (для проверки) или надо передавать тайну для выполнения аутентификации в клиенте.

Минусы шифрования паролей:

1. Если тайна скомпрометирована, все пароли скомпрометированы. Если у кого-то есть доступ к тайне и к хранилищу паролей, все пароли могут быть расшифрованы!

2. Только доступа к хранилищу паролей достаточно, чтобы предоставить информацию о паролях, так как все пароли шифруются с помощью одинакового алгоритма. Если два пользователя имеют одинаковый зашифрованный пароль, они также должны иметь одинаковый пароль. Хитрые хакеры с доступом к хранилищу паролей могут создать пользователей с известными паролями и проверить на наличие других пользователей с таким же паролем. Такой тип атаки является разновидностью атаки с известным открытым текстом. Такие атаки можно остановить с помощью соли (смотри ниже).

3. При использовании блочного шифра длина пароля должна храниться в составе зашифрованного пароля. Надо хранить длину, потому что блочные шифры всегда дают блок зашифрованного текста фиксированного размера. Если длина пароля не зашифрована (например, если хранится как столбец в таблице), информация очень полезна для взломщиков паролей. Знание точной длины пароля сильно упрощает угадывание пароля.

Использованные источники:

1. Галатенко В.А. Основы информационной безопасности [Электронный ресурс] — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 266 с.
2. Фаронов А.Е. Основы информационной безопасности при работе на компьютере [Электронный ресурс] — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 154 с.

3. Учебно-методическое пособие по дисциплине Методы и средства защиты компьютерной информации [Электронный ресурс] — М.: Московский технический университет связи и информатики, 2016.— 32 с
4. Басалова Г.В. Основы криптографии [Электронный ресурс] — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 282 с.