ИСПОЛЬЗОВАНИЕ «СОЛИ» ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ХЕШИРОВАНИЯ

И.Ю. Изгачёв, М.А. Аниховский, В.А. Ковалёв, А.В. Галковский

В настоящее время хеширование используется для хранения паролей пользователей веб-приложений. Тем не менее регулярно конфиденциальные данные «утекают» в сеть в результате взломов аккаунтов пользователей веб-сервисов [1].

Хеширование – это процесс трансформации произвольного массива данных в строку фиксированной длины. Этот процесс необратим, т.е. невозможно восстановить данные по хешу. Это свойство хешей позволяет использовать их для хранения паролей, ведь захешированное значение невозможно получить, однако возможно проверить его на соответствие с введенной строкой. Однако, злоумышленник может подобрать необходимый для авторизации хеш используя таблицы поиска – таблицы со списком наиболее распространенных паролей и соответствующими хешами. Также используются радужные таблицы [2], которые используют дополнительную память для более эффективного поиска необходимого пароля. Атаки с использованием подобных таблиц эффективны по причине того, что процесс хеширования для каждого пароля происходит одинаково, т. е. значения хеша для двух идентичных паролей будут совпадать. Для рандомизации значений хеша необходимо перед хешированием добавить к паролю строку из случайных символов («соль»).

Использование «соли» делает атаку с помощью таблиц поиска и радужных таблиц неосуществимой, однако для этого «соль» должна быть уникальной для каждого пароля [3]. Этого можно добиться формированием «соли» с помощью криптографически безопасного генератора псевдослучайных чисел, использующего, например, текущее системное время [4]. Также в качестве дополнительной меры безопасности можно использовать секретный ключ, добавляемый к итоговому хешу. Такой ключ необходимо сохранять на отдельном защищенном сервере.

Литература

1. CSO Online [Электронный ресурс]. – Режим доступа: https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html. – Дата доступа: 12.04.2021

2. Dafydd Stuttard, Marcus Pinto, «The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws»

3. Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse. The Mobile Application Hacker’s Handbook.

4. David Litchfield, Chris Anley, John Heasman, Bill Grindlay. The Database Hacker’s Handbook: Defending Database Servers.