БЕЗОПАСНОСТЬ ХЕШИРОВАНИЯ ПАРОЛЕЙ В ОБЛАКЕ

Курманбаев Е.А., к.ф.-м.н., доцент
Бейсебаева Ж.Е., магистр
Казахский гуманитарно-юридический инновационный университет г. Семей Республики Казахстан

Наиболее распространенными средствами аутентификации являются системы аутентификации на основе паролей [1]. Сотрудник ежедневно использует несколько паролей для всех приложений и систем, над которыми он / она может работать для работодателя. Предприятия тратят огромные деньги не только на хранение этих паролей, но и на обеспечение безопасности хранения этих паролей. Особенно, когда организация имеет дело с огромным количеством клиентов; им очень трудно создавать, поддерживать и распространять эти пароли по сети для целей аутентификации, авторизации или учета. Таким образом, система аутентификации на основе паролей имеет много проблем безопасности в относительно относительно защищенных существующих инфраструктурах [2]. Чтобы преодолеть возможные проблемы безопасности при хранении и распределении пароля по сети, пароль часто запускают с помощью криптографической хэш-функции, чтобы получить эквивалентный дайджест пароля, который хранится вместе с другими учетными данными пользователя в базе данных. Когда пользователи пытаются войти в систему с паролем, ввод вычисляется с помощью той же хеш-функции, чтобы сравнить с дайджестом того же пароля, который был сохранен в базах данных. Одним из свойств криптографической хеш-функции является ее необратимая односторонняя функция, которая означает, что практически невозможно вернуть пароль из самого дайджеста. С другой стороны, многие широко используемые функции хеширования, такие как MD5, SHA-1 и т. Д., Были разработаны в середине 90-х годов. Одна из слабых сторон наиболее широко используемой хеш-функции MD5 заключается в том, что злоумышленник может создать два идентичных дайджеста для двух разных входов, которые в области криптографии называются хэш-коллизиями [3]. Фактически, вероятность того, что злоумышленник найдет пароль из дампа хеша, пропорциональна количеству выполняемой им работы и способности предсказать распределение характеристик пароля. Более того, с появлением облачных вычислений и 8 новых мощных графических процессоров (GPU), злоумышленник теперь имеет все возможности для расшифровки паролей из хэша по своему усмотрению. С последних десятилетий в области графического процессора (GPU) происходит значительное развитие. Графический процессор очень подходит для выполнения параллельных задач, а также для вычисления проблем, связанных с плавающей запятой. В современных графических процессорах взлом паролей происходит в десять раз быстрее, чем взлом паролей в центральном процессоре (ЦП) [4]. В нашей статье мы покажем, как злоумышленник может использовать существующие облачные сервисы для взлома паролей из примеров хеш-паролей с использованием высокопроизводительных вычислительных ресурсов. Мы считаем, что наша

работа является одним из немногих исследований, посвященных возможностям вычислений на GPU с использованием сервисов облачных вычислений, таких как веб-сервисы Amazon, и широте выполняемых испытаний. Наша статья структурирована таким образом, что в первых нескольких разделах мы предоставим читателю некоторую предысторию обо всех возможных атаках, связанных с паролями, где в средней части мы настроили нашу тестовую среду в AWS с образцом хеш-файла пароля другого популярного хэш-алгоритма, используются сегодня, чтобы взломать. В конце мы сравниваем результаты и получаем время взлома из нашего тестового сценария. Затем мы сравниваем результаты теста и предлагаем использовать скрипт генератора паролей и другие возможные способы создания надежных паролей для использования онлайн-сервисов.

2. Природа и значение проблемы

Из-за недавнего взлома и публичного раскрытия частной информации (пароли пользователей) от нескольких крупных профильных организаций, таких как LinkedIn, E-Harmony и Yahoo, в течение последних 5 лет ставит серьезные вопросы не только о безопасности систем аутентификации в этих высококлассных организациях. но также и аспекты безопасности их методов хранения паролей в их базах данных.

Таким образом, в этой статье представленная работа демонстрирует, насколько эффективна технология взлома паролей на основе графического процессора против методов хеширования, и дает представление о том, почему выбор надежных, сложных паролей наряду с функцией хеширования медленных компьютеров будет держать злоумышленников в страхе при одновременном использовании GPU вычислительная мощность облачных вычислительных ресурсов. Мы полагаем, что наша модель может быть применена к любой форме сетевых ресурсов GPU / CPU и дает аналогичные результаты для определения надежности хэша пароля любого типа. Предыдущая работа обычно выполнялась на основе отдельных экземпляров вычислительных ресурсов, но мы считаем, что наша среда обеспечивает более современную распределенную мощность графического процессора, доступную в настоящее время в облаке.

Список литературы:

1. O’Gorman, L. (2003) Comparing Passwords, Tokens, and Biometrics for User Authentication. Proceedings of the IEEE, 91, 2021-2040.
2. Adams, A., Sasse, M.A. and Lunt, P. (1997) Making Passwords Secure and Usable. In: Thimbleby, H., et al., Eds., Proceedings of HCI on People and Computers XII, Springer-Verlag, London, 1-19. (2016) MD5 Message Digest Algorithm Hash Collision Weakness. Securityfocus.com.
3. Qiu, W.D., Gong, Z., Guo, Y.D., Liu, B.Z., Tang, X. and Yuan, Y. (2016) GPU-Based High-Performance Password Recovery Technique for Hash Functions. ResearchGate.

4. Thompson, C.J., Hahn, S. and Oskin, M. (2002) Using Modern Graphics Architectures for General-Purpose Computing: A Framework and Analysis. Proceedings of the 35th annual ACM/IEEE International Symposium on Microarchitecture, Istanbul, 18-22 November 2002, 306-317.