И.М. Бурлака;
С.В. Пилькевич,
доктор технических наук

ИССЛЕДОВАНИЕ СТОЙКОСТИ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ В ПРИЛОЖЕНИИ К ЗАДАЧАМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В статье рассматриваются результаты экспериментального исследования стойкости криптографических хэш-функций применительно к защите парольной информации при задействовании вычислительных систем различной конфигурации. Проведено исследование современных рекомендаций, реализуемых в рамках политики безопасности, которая регламентирует вопросы безопасного создания и хранения паролей, а также влияния трудоемкости хеш-функции на работу информационной системы.

Ключевые слова: криптографическая хэш-функция, парольная информация, словарь, радужная таблица, идентификация пользователя, SSD накопитель.

ВВЕДЕНИЕ

Одной из традиционных сфер применения хэш-функций являются процедуры идентификации объектов для защиты паролей. Впервые идея хранения паролей в хэш-форме была предложена Робертом Моррисом для операционной системы UNIX [1]. Его алгоритм, известный как crypt, использовал алгоритм DES и 12-битный случайный вектор для снижения риска перебора пароля по словарю. Данный подход в той или иной форме применяется и в современных информационных системах: администраторы не могут узнать пароли пользователей, но при этом пользователи с помощью своих паролей получают доступ в систему, располагающую хэшем пароля пользователя.

Взлом пароля является одним из распространенных типов атак на информационные системы, использующие аутентификацию по паролю или паре «имя пользователя-пароль». Суть атаки сводится к завладению злоумышленником паролем пользователя, имеющего право входить в систему. Привлекательность атаки для злоумышленника состоит в том, что при успешном получении пароля он гарантированно получает все права пользователя, учетная запись которого была скомпрометирована. Кроме того, вход под существующей учетной записью обычно вызывает меньше подозрений у системных администраторов.

В данной работе рассматривается ситуация, при которой злоумышленнику удалось перехватить трафик или получить доступ к хранилищу (файлу, базе данных), содержащему хэши паролей. Целью исследования является разработка рекомендаций по выбору пароля, обеспечивающего его стойкость по отношению к взлому и применению нарушителем различных методов криптографического анализа используемой в информационной системе хэш-функции, а также влияние хеш-функции на работу информационной системы.

СОВРЕМЕННЫЕ АТАКИ НА ПАРОЛЬНУЮ ИНФОРМАЦИЮ

В рамках проведенного исследования были рассмотрены следующие виды атак:

1. Метод полного перебора на GPU и CPU.
2. Атака по словарю.
3. Использование радужных таблиц.

Метод полного перебора относится к классу методов поиска решения исчерпыванием всевозможных вариантов. Сложность полного перебора зависит от мощности множества всех

возможных решений задачи. Если пространство решений очень велико, то полный перебор может не дать результатов в обозримом временном интервале. Криптографические атаки, основанные на методе полного перебора, являются наиболее универсальными, но и самыми долгими.

Перебор по словарю является адаптированным вариантом метода полного перебора, осуществляемым относительно сокращенного пространства гипотетических решений – паролей определенного вида и длины, объединенных в словарь по принципу наибольшей частоты использования пользователями.

Радужная таблица представляет собой специальный вариант таблиц поиска для обращения криптографических хэш-функций, использующий механизм разумного компромисса между временем поиска по таблице и занимаемой памятью. Основная идея, эксплуатируемая радужными таблицами, состоит в том, что для всех распространенных и коротких паролей, не содержащих соль, нарушитель может заранее подсчитать значения хэшей и сохранить их в таблице. Это позволяет быстро найти совпадение в заранее сформированной таблице. Но чем длиннее пароль, тем больше таблица и тем больше памяти необходимо для ене хранения. Альтернативным вариантом является хранение только первых элементов цепочек хэшей. В результате требуется больше вычислений для поиска пароля, но значительно уменьшается количество требуемой памяти. Таким образом, радужные таблицы являются улучшенным вариантом данного метода, позволяя избежать коллизий.

Перечисленные виды атак не являются новыми, однако практическая сторона их применения постоянно претерпевает изменения, зачастую значительные.

1. Метод полного перебора хорошо поддается распараллеливанию, что открывает широкие перспективы по применению многопроцессорных вычислительных систем.
2. Результаты анализа многочисленных утечек персональных данных и реквизитов доступа пользователей позволяют формировать актуальные чрезвычайно полные словари, используемые при одноименной атаке.
3. Основные недостатки радужных таблиц – высокая стоимость и большой объем накопителя, используемого для хранения таблиц. Они стремительно нивелируются за счет удешевления оборудования. Кроме того, открываются новые возможности, обусловленные применением передовых технологий хранения данных – переход с HDD на SSD накопители.

ПРОВЕДЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

В качестве базового алгоритма бесключевого преобразования данных использована хэш-функция MD5. Выбор обусловлен простотой ее реализации и своеобразной инвариантностью по отношению к вычислительной платформе, на которой она выполняется.

Тестирование производительности метода полного перебора пароля на CPU и GPU осуществлялось на основе программы, реализованной на языке программирования python. Рассматривалась зависимость времени выполнения операций от длины парольной фразы и от мощности используемого алфавита.

Сравнение динамики изменения параметров для CPU и для GPU дает закономерный результат – большую оперативность вычислений на GPU. Причины этого заключаются в том, что скорость вычисления хэш-функции для CPU примерно в 10 раз меньше, чем у GPU (для CPU 260.9 MH/s, а для GPU 3620.6 MH/s). Вместе с тем CPU не предназначены для решения подобных задач, в отличие от GPU, которые изначально создавались для обработки изображений, где требуется выполнение большого количества простейших операций.

Экспериментальная вычислительная платформа, реализующая возможности GPU, построена на основе видеокарты Nvidia GeForce GTX 960M, выпущенной пять лет назад. Но даже такое, несколько устаревшее, оборудование обладает высокой производительностью (тенденции последних лет показывают ежегодный прирост производительности видеокарт более

чем вдвое). Результаты сравнения скорости криптографических преобразований представлены в табл. 1.

Таблица 1

Сравнение скорости выполнения хэш-функций на базе видеокарты Nvidia GeForce GTX 960M

№ п/п	Название хэш-функции	Скорость вычисления хэшей, миллионов в секунду	По отношению к MD5
1	MD5	3753.9	1
2	SHA1	1268.6	0.34
3	SHA2-256	452.7	0.12
4	SHA2-512	153.1	0.04
5	NTLM	6799.2	1.81
6	LM	3823.0	1.02
7	bcrypt	0.001773	47230
8	ГОСТ Р 34.11-2012	2.7694	1355

Таким образом, данные по производительности, полученные для различных хэш-функций, могут быть сопоставлены и, при необходимости, конвертированы друг в друга. Вместе с тем с увеличением трудоемкости хеш-функции увеличивается время обработки каждой авторизации в систему. Поэтому требуется более мощное оборудование, чтоб уменьшить это влияние.

Специфика атаки по словарю состоит в том, что такой метод использует своеобразную уязвимость системы защиты информации, обусловленную человеческим фактором. Пользователи предпочитают применять короткие семантически окрашенные пароли, им свойственно лениться и редко менять пароли. В результате имеется возможность составления баз данных наиболее часто используемых паролей – так называемых словарей.

Время перебора указанным методом существенно меньше, чем методом полного перебора. Например, самый популярный словарь rockyou содержит 14 344 391 слов, а размер словаря, состоящего из всех слов длиной шесть символов и с алфавитом из 63 букв, равен 62 523 502 209 слов. Фактография исследований, проводимых на протяжении последних 30 лет [2–5], показывает, что процент встречаемости паролей в словарях превышает 30%.

Программа RainBowCrack представляет собой одну из наиболее удачных современных реализаций метода, использующего радужные таблицы. Как и в предыдущих случаях, критичными для объема памяти, занимаемого таблицей, являются длина слов и размер алфавита.

Для взлома сложных паролей (длиной более 10 символов при 100 символах в алфавите) требуется большой объем таблицы (более 10 ТБ). Но это намного меньше, чем простая таблица хэшей паролей (более 86 ТБ). Отметим, что применение данного метода требует определенного объема оперативной памяти, так как туда осуществляется отображение фрагментов таблиц с жесткого диска компьютера. Это обстоятельство приводит к тому, что большое количество времени уходит на загрузку данных с диска (около 98% времени работы программы).

Представляется, что оперативность выполнения сортировки и поиска можно увеличить за счет использования SSD вместо HDD. Для проверки этой гипотезы был проведен вычислительный эксперимент. В качестве исходных данных использованы радужные таблицы [6] для алфавита из 25 символов с максимальной длиной 9 символов, и набор из шести хэшей, выбранных таким образом, чтобы один из них не содержался в таблицах. Это позволяет измерить максимальное время работы и задействованный объем памяти (табл. 2).

Таблица 2

Сравнение параметров работы радужных таблиц на SSD и HDD

Параметры работы метода	Тип устройства хранения данных	
	HDD	SSD
Полное время перебора, сек.	846.03	267.99
Время прохождения цепи, сек.	53.14	52.86
Время чтения с устройства (диска), сек.	840.61	250.28
Количество цепочек хешей, сек.	6498700000	
Скорость чтения цепочек хешей, миллионов в секунду	13.17	12.29

Основные параметры примерно одинаковы, за исключением времени чтения информации с носителя. Скорость работы на SSD увеличилась в 3 раза, что существенно улучшило результат - 267.99 секунд против 846.03 секунд.

СПОСОБЫ ЗАЩИТЫ ПАРОЛЬНОЙ ИНФОРМАЦИИ ОТ КРИПТОГРАФИЧЕСКИХ АТАК

Из анализа методов можно сделать вывод, что для восстановления пароля самым быстрым является метод атаки по словарю, но он относится к классу вероятностных. Поэтому для однозначного получения результата лучше всего использовать метод полного перебора на GPU либо радужные таблицы. На практике реализация каждого из этих подходов потребует существенных финансовых затрат (618 млн. рублей за систему хранения данных для радужных таблиц или ASIC-майнеры для полного перебора на GPU). В то же время, если нарушитель располагает достаточным временем (более 46 ч) для проведения атаки, то доступность оборудования и осуществление атакующих воздействий становятся вполне реальными. В результате возникает необходимость разрабатывать новые способы защиты от подобных видов атак.

Защита парольной информации базируется на грамотном выборе политики безопасности, регламентирующей вопросы создания и хранения паролей. В настоящее время наиболее актуальными являются следующие рекомендации:

1. Использование криптографически стойких хэш-функций (например, SHA-3 или ГОСТ Р 34.11-2012). При выборе более трудоемкой хеш-функции также возрастает требование к оборудованию либо увеличивается время отклика от информационной системы. Для того чтобы не было такой проблемы требуется выбирать оптимальную хеш-функцию, которая обеспечить защиту и не будет перегружать систему.
2. Использование соли – подход, предполагающий увеличение пароля путем присоединения к нему (в начало или конец) случайной символьной последовательности, которая удлиняет пароль, а также делает его нетипичным. Таким образом, можно обезопасить информационную систему от словарной атаки и от атаки с использованием радужных таблиц. Кроме того, это увеличивает время для метода полного перебора. Отметим, что нельзя использовать короткую и повторяющуюся соль.
3. Использование сложных и нетипичных паролей.
4. Неоднократное использование хэш-функции (также с использованием соли) и/или комбинирование различных хэш-функций, например: md5(sha1(пароль)); md5(md5(соль) + + md5(пароль)); sha1(str_rot13(пароль + соль)); md5(sha1(md5(md5(пароль) + sha1(пароль)) + md5(пароль))).

Отдельного рассмотрения заслуживают изменения в рекомендациях NIST для политик безопасности [7]:

1) отказ от частой смены пароля, так как пользователи часто забывают пароли либо пишут их на бумаге и оставляют на рабочем месте;

2) проверка отсутствия пароля в базах данных популярных паролей;

3) отсутствие ограничений на использование любых символов в пароле;

4) использование парольной фразы вместо одного слова (комбинации символов), например, пароль из четырех случайных слов «рыба стол насос слон».

Приведенные способы защиты не смогут обезопасить систему на 100%, но значительно усложнят процедуру подбора пароля злоумышленником.

ЗАКЛЮЧЕНИЕ

Современные методы по восстановлению паролей продолжают совершенствоваться – прогресс в данной области обусловлен, главным образом, развитием программных и аппаратных технологий, а не прорывными достижениями криптографического анализа.

Стоимость оборудования, необходимого для преодоления процедур идентификации объектов, неуклонно снижается, и оно становится доступным для широкого круга злоумышленников. Поэтому необходимо использовать современные методы защиты паролей и внимательно строить политику безопасности, опираясь на изложенные выше рекомендации.

Список используемых источников

1. *Morris R., Grampp F.T.* UNIX Operating System Security // AT&T Bell Laboratories Technical Journal, 63, part 2, #8. – October 1984. – P. 1649–1672.

2. The memorability and security of passwords – some empirical results / J. Yan [et al.] // Technical reports published by the University of Cambridge Computer Laboratory, September 2000. – URL: https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-500.pdf (дата обращения: 20.04.2021).

3. *Spafford E.H.* Observations on reusable password choices. In Proceedings of the 3rd Security Symposium. Usenix. – September 1992. – URL: http://ftp.cerias.purdue.edu/ pub/papers/gene-spafford/spaf-OPUS-observe.pdf (дата обращения: 20.04.2021).

4. *Hunt T.* The 773 Million Record "Collection #1" Data Breach // www.troyhunt.com. – January 2019. – URL: https://www.troyhunt.com/the-773-million-record-collection-1-data-reach/ (дата обращения: 20.04.2021).

5. *Klein D.V.* Foiling the cracker: A survey of and improvements to password security // Programming and Computer Software. – 1992. – № 17.

6. Distributed RainbowTable Project «Free Rainbow Tables». – URL: https:// freerainbowtables.com/ (дата обращения: 20.04.2021).

7. NIST Special Publication 800-63C Digital Identity Guidelines. – URL: https://pages.nist.gov/800-63-3/sp800-63c.html/ (дата обращения: 20.04.2021).