УДК 004.056.53

К.А. Тюрин, Р.В. Сёмин

АНАЛИЗ СТОЙКОСТИ ПАРОЛЬНЫХ ФРАЗ НА ОСНОВЕ ИНФОРМАЦИОННОЙ ЭНТРОПИИ

Известно, что несанкционированный доступ к важным информационным системам влечет за собой убытки для многих компаний, и эти убытки с каждым годом увеличиваются. Несмотря на сколь угодно надежные технические средства защиты, безопасность любой многопользовательской информационной системы подвержена влиянию человеческого фактора. Исследовав характер этого влияния, злоумышленник может оптимизировать процедуру атаки на информационную систему, а значит, особенности влияния человеческого фактора и возможности использования сведений о них при проведении атак являются важными предметами для изучения. В большинстве случаев аутентификация осуществляется либо при помощи метода однофакторной аутентификации по паролю, либо данный метод входит в состав используемого (как правило, методы двухфакторной аутентификации используют ввод постоянного пароля как один из факторов). Исследования показывают, что зачастую пользователи применяют в качестве паролей информативные последовательности символов. Этот факт объясняется тем, что такие пароли проще запоминаются человеком. Информационная составляющая используемых паролей может определяться присутствием в них фрагментов слов естественных языков, расположением символов на клавиатуре и т. п. Информативность делает пароли не вполне случайными, а значит, априорные данные о статистических особенностях языка используемых на практике паролей могут существенно ускорить процесс их перебора в ходе атак на информационные системы. Таким образом, в случае, когда пароль создается пользователем, а не генерируется автоматически, возникает специфическая угроза безопасности, обусловленная человеческим фактором. Существует множество способов реализации этой угрозы при совершении попыток несанкционированного доступа. Изучение таких способов позволяет выработать правила, соблюдение которых предотвращает такого рода угрозы. Рассматриваются существующие способы атак на системы аутентификации, а также предлагается модификация существующего алгоритма, оптимизирующего подбор пароля на основании априорных данных о статистике реального использования паролей, характеризуемых содержанием ненулевой информации. Приведен пример работы предлагаемого алгоритма и анализ его эффективности по сравнению с существующими аналогами. Алгоритм может быть использован для тестирования безопасности парольных систем, пентестов.

Информационная безопасность; энтропия; атака по словарю.

K.A. Turin, R.V. Semin

ANALYSIS OF PASSPHRASES RESISTANCE BASED ON INFORMATION ENTROPY

As is known, unauthorized access to critical information systems causes damage to many companies, and this damage grows year by year. Even if the technical protection is high, the human factor has a great influence on security of multi-user information system. The attacker is able to optimize the procedure of attacks on information systems by investigating the nature of this influence. This means that the influence of the human factor and the possibility of using user's information for the attacks are important for researching. In most cases, information systems use the single-factor password authentication or some other method that includes this one (most two-factor authentication methods use the permanent password as one of the factors). Investigations show that users often use informative sequence of characters as their passwords. This is explained by the fact that such passwords are easier to remember. Information component allow passwords to contain fragments of natural languages words, keyboard layouts, and so on. Informative part makes passwords not completely random. This means that statistical information about the features of the used in practice passwords language can accelerate the process of information systems cracking. Thus, when a password

is not automatically generated and created by the user, there is a specific threat because of the human factor. There are many ways to implement this threat for the unauthorized access attempts. The research of these methods allows developing rules that prevent such threats. In this paper, we review existent methods of authentication systems cracking and present the modification of the algorithm that optimizes the cracking of a password based on using of information about the statistics of the actual use of passwords. In addition, paper contains the concrete example of algorithm work and analysis of its efficiency. The algorithm can be used for testing the security of password systems, pentesting.

Information security; entropy; dictionary attack.

Введение. Современные системы однофакторной аутентификации можно рассматривать как «черный ящик», способный проверять корректность введенного пароля. Особый интерес представляют те системы, пользователями которых являются люди, т.е. такие, в которых при выборе пароля присутствует человеческий фактор. Благодаря компрометации таких систем зачастую существует доступ к выборке паролей некоторого подмножества пользователей систем. Анализ свойств такой выборки позволяет делать выводы о характере паролей, выбираемых пользователями. Однако выделение этих свойств и возможных способов их использования остается открытой задачей, что приводит к различным способам решения данной проблемы. Предложенный в статье алгоритм основан на использовании неравновероятного распределения символов на позициях и появления пар расположенных друг за другом символов в парольных фразах, составляемых пользователями.

Постановка задачи. Известными параметрами системы являются: A – алфавит допустимых символов, m – размер алфавита (m = |A|), n – число символов перебираемых паролей, X – функция распределения вероятностей символов, X^2 – функция распределения вероятностей биграмм, X_k – распределение вероятностей символов на k-й позиции в пароле. Необходимо разработать алгоритм, позволяющий подобрать корректный пароль, который заранее неизвестен. Требованием к алгоритму является минимизация математического ожидания количества времени, проходящего до достижения результата.

Обзор существующих решений. Существует достаточно много известных подходов к решению этой задачи [1, 2], характеризующихся различным уровнем практичности. Рассмотрим некоторые из них.

1. Метод грубой силы. Данный подход представляет собой последовательный перебор всевозможных паролей [3]. Пароли генерируются на каждой итерации без использования статистических особенностей. Таким образом, алгоритм требует O(n) памяти, так как в каждый момент времени в памяти находится только текущий пароль. Самый простой в реализации способ, который, однако, не использует никакой информации об используемых паролях. Асимптотическая оценка количества попыток до успеха равна $O(m^n)$. При этом математическое ожидание количества попыток равно половине числа всевозможных паролей ($M_{\text{гр}} = \frac{m^n}{2}$).

2. Использование словаря всех паролей. Для использования этого метода необходимо провести статистический анализ базы используемых на практике паролей. После этого составляется словарь фраз, которые отсортированы по вероятности их появления в данной базе. Для этого могут использоваться различные статистические данные, такие как вероятности появления символов, вероятности появления символов на конкретных позициях, вероятности появления пар символов и т.д. [4]. Пароли перебираются в порядке убывания данной вероятности [5].

Закон Ципфа [6] гласит, что многие языки имеют распределение слов, близкое к распределению Парето, т.е. $p_i = \frac{1}{c*i}$, где c – константа, равная $\sum_{i=1}^{m^n} \frac{1}{i}$. Тогда мы получим математическое ожидание числа итераций перебора $M_{сл} = \sum_{i=1}^{m^n} \frac{i}{c*i} = \frac{1}{c} * \sum_{i=1}^{m^n} \frac{i}{i} = \frac{m^n}{c}$. Так, для пароля длиной 8 символов, состоящего из букв латин-

ского алфавита в разных регистрах, цифр и специальных символов (всего 94 различных символа), $M_{сл} = \frac{94^8}{\sum_{i=1}^{94^8} \frac{1}{i}} \approx \frac{6*10^{15}}{36.9} \approx 1.6 * 10^{14}$, $M_{гр} \approx \frac{6*10^{15}}{2} = 3 * 10^{15}$. Таким образом, перебор по словарю даст улучшение в скорости по сравнению с методом грубой силы в 18,75 раза.

Потребляемая при использовании такого подхода память равна $O(nm^n)$, так как необходимо хранить весь словарь.

3. Радужные таблицы. Радужные таблицы также используют словарь всевозможных паролей, однако за счет предвычисления хэш-функций сокращают процесс перебора [7]. При этом потребление памяти имеет оценку $O(m^{2n/3})$. Недостатком данного метода является то, что для его использования необходимо знать используемый алгоритм хэширования. К тому же использование «соли» (некоторого случайного префикса, который добавляется к паролю перед хэшированием) позволяет защититься от данного метода атаки [8].

4. Перебор, основанный на использовании информационной энтропии [9]. Данный метод не использует словарь паролей как таковой. Однако он использует более общую информацию, полученную в результате его анализа, что позволяет не хранить весь словарь в памяти и при этом использовать его преимущества. Его описание представлено ниже.

Данный подход к перебору паролей основывается на последовательном генерировании и проверке корректности вариантов пароля таким образом, что генерируемая последовательность по своим статистическим свойствам близка к заранее обработанному частотному словарю встречающихся на практике паролей. При этом генерируемое множество включает в себя этот словарь как подмножество, а разработка алгоритма генерации подчинена, среди прочего, намерению генерировать последовательность, включающую в себя по возможности значительную часть частотного словаря как подпоследовательность.

В работе [10] был предложен следующий алгоритм генерации паролей, основанный на вышеизложенной идее и использующий словарь реально используемых паролей.

Фиксируем некоторую опорную позицию s. Поочередно фиксируем на ней отсортированные по убыванию вероятности символы исходного алфавита. Для каждого зафиксированного символа начинаем перебирать всевозможные его префиксы и суффиксы. Их перебор осуществляется при помощи деревьев, узлы которых являются символами, отсортированными по убыванию условной вероятности (деревья не обязательно хранить в памяти, они могут генерироваться динамически). Так, для сортировки дерева префиксов используются условные вероятности $P_{"b"|"a"} = \frac{N_{"ab"}}{N_{"a"} - N_{end"a"}}$, где $N_{"ab"}$ – количество биграмм "ab", $N_{"a"}$ – количество символов "a", $N_{end"a"}$ – количество паролей, которые заканчиваются символом "a". Для сортировки дерева суффиксов – $P_{"a"\uparrow"b"} = \frac{N_{"ab"}}{N_{"b"} - N_{begin"b"}}$, где $N_{"ab"}$ – количество биграмм "ab", $N_{"b"}$ – количество символов "b", $N_{begin"b"}$ – количество паролей, которые начинаются с символа "b".

Для оптимизации выбора опорной позиции предлагается следующая процедура. Для каждой позиции определим наименьшее число элементов, начиная с первого в упорядочении по убыванию, идущих подряд так, чтобы сумма их вероятностей была больше некоторого порогового значения. После этого выбирается позиция с наименьшим числом таких элементов. Этот способ позволяет достаточно хорошо выбрать опорную позицию, так как за счет фиксирования элемента с высокой частотой появления в данном месте мы увеличиваем вероятность перебираемых в первую очередь паролей.

Описанный алгоритм работает гораздо эффективнее метода грубой силы за счет того, что мат. ожидание числа попыток становится меньше благодаря использованию статистических особенностей. Эффективность по сравнению с использованием отсортированного словаря проявляется в том, что нет необходимости хранить весь словарь в памяти, его элементы генерируются на каждой итерации. Также данный алгоритм не привязан к методам проверки, как метод радужных таблиц, т.е. система может быть настоящим «черным ящиком» и её внутренняя структура может быть абсолютно неизвестна. За счет вышеупомянутых преимуществ данный алгоритм рекомендуется для использования в большинстве случаев, особенно когда существует ограничение на количество используемой памяти.

Однако выбор текущей позиции лишь по наибольшим вероятностям не гарантирует поведение алгоритма в условиях нахождения в данной позиции в дальнейшем. Проблема заключается в том, что опорная позиция фиксируется единожды и в дальнейшем не может быть изменена. Так, некоторое количество символов с меньшими вероятностями может иметь равномерное распределение, что вносит больший элемент случайности в выбор текущего пароля. Но и при неравномерном распределении вероятности показательность частот появления символов уменьшается.

Представлена модификация данного алгоритма, позволяющая избежать данного недостатка. Эта модификация заключается в динамическом выборе опорной позиции и фиксируемого на ней символа, что увеличивает эффективность предложенного подхода. Основным требованием к этой модификации является то, что динамический выбор позиции не должен приводить к повторному перебору уже проверенных паролей. Для достижения этого вводится дополнительный массив текущего индекса для каждой позиции (возможны и другие варианты реализации данной идеи, основанные на использовании иных структур данных).

Использование этого массива позволит запоминать, какие символы, на каких позициях мы уже перебирали. Таким образом, когда мы переберем всевозможные префиксы и суффиксы для некоторого символа на данной позиции, мы увеличим счетчик данной позиции на единицу, и этот символ больше никогда в данном месте не окажется. Теперь алгоритм выглядит следующим образом.

Начальные значения элементов массива z – нули. Для каждой позиции k отсортируем символы по убыванию вероятности их нахождения на данном месте. Получившиеся массивы пар (символ, вероятность данного символа на данной позиции) назовем p^k, где k – номер позиции. Теперь на каждой итерации будем выбирать максимальный среди элементов $p^k_{z_k}$. Пусть t – позиция данного элемента, e – символ, $p(e)_k$ – вероятность его нахождения на позиции t. Тогда выбираем t как опорную позицию, после чего начинаем перебор префикса и суффикса по тем же принципам, что и в базовом алгоритме, за исключением того, что во время построения деревьев при выборе элемента для k-й позиции будут использоваться только символы, принадлежащие элементам $p^k_{z_k}.. p^k_{m-1}$. После того как были перебраны все префиксы и суффиксы, увеличиваем счетчик z_k на единицу. Таким образом, если когда-то на позиции k был зафиксирован символ a, то больше он никогда на данном положении зафиксирован не будет, что защищает данный алгоритм от перебора одних и тех же паролей.

Таким образом, в данном алгоритме мы будем фиксировать на позициях лишь те символы, которые появляются на них наиболее часто. К сожалению, статистика распределения символов с каждой новой итерацией теряет актуальность за счет того, что не используется статистика распределения символов на позициях при условии фиксации другого символа на некоторой позиции. Такое улучшение могло бы дать сильное преимущество данному алгоритму, однако такая статистика очень сложна для вычисления и слишком объемна.

В качестве последующего улучшения как базового алгоритма, так и данной его модификации, можно рассмотреть оптимизацию перебора префиксов и суффиксов. На данный момент перебор осуществляется простым образом: зафиксировав некоторый префикс, перебираются всевозможные суффиксы, после чего переходим к следующему префиксу и так далее. Возможное улучшение будет заключаться в следующем. Так как префиксы и суффиксы соответствуют листьям деревьев, можно простым и быстровычислимым образом однозначно сопоставить каждому листу натуральное число. Тогда порядок перебора пар (префикс, суффикс) будет выглядеть как (0,0), (0,1), (0,2),..., (0, s)... (r, s-1), (r, s). Улучшение заключается в следующем порядке перебора пар: (0,0), (0,1), (1,0), (1, 1), ..., (r, s). Мотивация такого улучшения заключается в том, что мы изначально считаем префиксы и суффиксы с меньшими номерами более удачными, поэтому перебор более вероятных вариантов вначале является обоснованным.

Пример работы алгоритма.

Пусть алфавит допустимых символов A состоит из трех элементов a, b, c. Длина пароля n = 5.

На первом этапе алгоритма отсортируем массивы p^k для каждого k по убыванию вероятности. Начальные значения элементов массива z – нули. Теперь выберем из всех $p^k_{z_k}$ элемент с максимальной вероятностью (рис. 1).

$$z = [0 \quad , \quad 0 \quad , \quad 0 \quad , \quad 0 \quad , \quad 0]$$

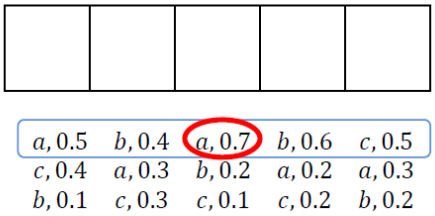

Рис. 1. Первый этап работы алгоритма

Выбираем данную позицию как опорную, фиксируем на ней символ выбранного элемента. Теперь проводим процесс перебора префикса и суффикса согласно вероятностям парного появления символов (рис. 2).

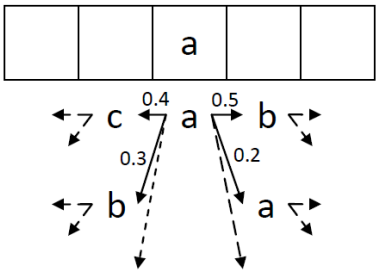

Рис. 2. Построение деревьев префикса и суффикса

Если на текущей итерации искомый пароль не был найден, увеличиваем элемент z_k на единицу и переходим к следующей. Так, снова выбираем опорную позицию и фиксируемый элемент (рис. 3).

$z = [0 \quad , \quad 0 \quad , \quad 1 \quad , \quad 0 \quad , \quad 0]$

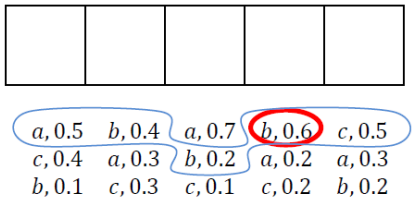

Рис. 3. Выбор нового опорного элемента

Продолжаем перебор, но с учетом элементов z_k, для того чтобы не попадать на уже перебранные пароли. Таким образом, элемент a больше не будет установлен на позицию с номером 2 (рис. 4).

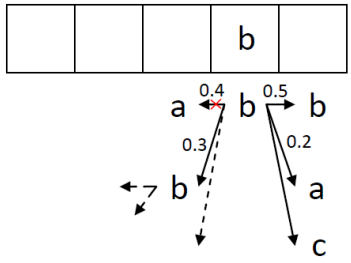

Рис. 4. Построение деревьев с учетом перебранных символов

На 5-й итерации процесс выбора опорной позиции и фиксируемого символа будет происходить так, как показано на рис. 5.

$z = [2 \quad , \quad 0 \quad , \quad 1 \quad , \quad 1 \quad , \quad 1]$

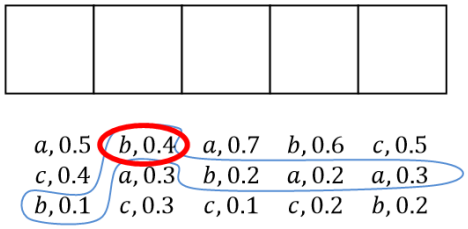

Рис. 5. Выбор опорного элемента после нескольких итераций

Оценка вычислительной сложности и занимаемой памяти.

Для работы данному алгоритму необходимы:

- вероятности «символ a перед b» – $O(m^2)$;
- вероятности «символ a после b» – $O(m^2)$;
- вероятности «символ a на позиции k» – O(m*n);
- массив счетчиков для каждой позиции – O(n);
- текущий подбираемый пароль – O(n).

Оцениваемая занимаемая память – O(m*q), где q = max(n, m).

Сложность выбора опорной позиции имеет значение O(n), так как заключается в поиске максимума в массиве размерностью n.

При обходе дерева префиксов и суффиксов требуется дополнительная проверка на то, допустим ли некий символ на данной позиции. При использовании массивов сложность проверки будет иметь значение O(n). Однако за счет использования иных структур данных или же дополнительных оптимизаций это время можно сократить до O(1).

Проведен эксперимент на скомпрометированной [11] базе парольных фраз. По известному набору паролей, число которых было около 100 000, была собрана исходная статистика. После этого для каждого пароля замерено, на какой итерации выбора опорной позиции данный пароль мог бы быть найден данным алгоритмом. Полученные данные проанализированы, и в результате оказалось, что на паролях, которые появляются с большей вероятностью, алгоритм дает улучшение около 16-ти символов (рис. 6). Это значит, что использование модифицированного алгоритма для данного случая может сэкономить в среднем $16m^{n-1}$ проверок, что эквивалентно полному перебору 16 паролей длиной на единицу меньше искомого по сравнению с исходным алгоритмом.

Рассмотренный алгоритм имеет более высокую сложность по сравнению с методом полного перебора, однако эта сложность компенсируется тем, что в процессе подбора пароля «узким местом» становится не процесс генерации, а процесс проверки пароля на правильность [12]. В случае же оценки количества перебираемых паролей, предложенный алгоритм быстрее метода полного перебора в $\frac{c}{2}$ раз, где c – константа, равная $\sum_{i=1}^{m^n} \frac{1}{i}$.

Описанный алгоритм обладает рядом преимуществ в сравнении с методом радужных таблиц:

- применение алгоритма, использующего информационную энтропию парольных фраз, не требует какого-либо знания о внутреннем устройстве системы аутентификации;
- способы противодействия против метода радужных таблиц [13, 14] не влияют на работоспособность предлагаемого алгоритма.

Эффективность алгоритма близка к эффективности перебора по словарю (с точностью до перестановки слов в словаре в пределах некоторых кластеров), однако позволяет существенно сэкономить потребляемую память. Так, предложенный алгоритм потребляет O(m*q) памяти (где q = max(n, m)), в то время как словарь паролей требует хранения $O(nm^n)$ символов.

Рис. 6. Сравнение методов

Заключение. Освещается проблема стойкости парольных фраз. Рассмотрены основные существующие методы перебора паролей, таких как полный перебор, перебор по словарю и использование радужных таблиц, приведены оценки их эффективности, рассмотрены основные достоинства и недостатки.

Предложен новый алгоритм, основанный на использовании понятия информационной энтропии и являющийся модификацией существующего. Алгоритм хорошо показал себя на популярных парольных фразах [15, 16], полученных при помощи анализа скомпрометированных баз паролей [17].

Проведенный эксперимент показал эффективность данного алгоритма по сравнению с существующими. Так, он позволял получить прирост по скорости по сравнению с методами полного перебора и перебора на основе информационной энтропии, а также прирост по памяти по сравнению с методами перебора по словарю и при помощи радужных таблиц.

Использование предлагаемого алгоритма позволяет более полно соблюдать российские [18] и иностранные [19, 20] требования к безопасности информационных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Burnett M.* Perfect Password: Selection, Protection, Authentication // Syngress Publishing. – 2006. – P. 194.
2. *Заркумова Р.Н.* Исследование количественных характеристик системы парольной защиты информации // Сборник научных трудов НГТУ. – 2010. – № 2 (60). – С. 83-88.
3. *Снегуров А.В., Чакрян В.Х.* Анализ устойчивости ко взлому современных механизмов парольной защиты операционных систем // Восточно-Европейский журнал передовых технологий. – 2011. – Т. 2, № 10 (50). – С. 27-29.
4. *Марков Г.А.* К вопросу об определении стойкости парольных систем // Сборник трудов Третьей всероссийской НТК «Безопасные информационные технологии» / Под. ред. В.А. Матвеева. – М.: НИИ РЛ МГТУ им. Н.Э. Баумана, 2012. – С. 21-23.
5. *Гуфан К.Ю., Новосядлый В.А., Эдель Д.А.* Оценка стойкости парольных фраз к методам подбора // Открытое образование. – 2011. – № 2. – С. 127-130.
6. *Kechedzhy K.E., Usatenko O.V., Yampol'skii V.A.* Rank distributions of words in additive many-step Markov chains and the Zipf law // Phys. Rev. E. – 2005. – Vol. 72.
7. *Hellman M.* A cryptanalytic time-memory trade-off // IEEE Transactions on Information Theory. – 1980. – Vol. 26. – P. 401-406.
8. *Ferguson Neils.* Practical Cryptography // Indianapolis: John Wiley & Sons. – 2003. – P. 230-243.
9. *Гуфан К.Ю., Новосядлый В.А., Эдель Д.А.* О методах оценки стойкости парольных фраз // Материалы XIX научно-технической конференции «Методы и технические средства обеспечения безопасности информации», 5–10 июля 2010 г. – СПб.: Изд-во Политехн. ун-та, 2010. – С. 73-74.
10. *Марков Г.А.* Метрики стойкости парольной защиты // Молодежный научно-технический вестник. – 2013. – URL: http://www.cnpo.ru/doc/psw-metrics.pdf (дата обращения: 27.12.2014).
11. 1 000 000 уже неработающих паролей в открытом доступе. Как мы защищаем пользователей Яндекса. Режим доступа. – http://habrahabr.ru/company/yandex/blog/236007 (дата обращения: 15.12.2014).
12. *Беленко А.* Пароли: стойкость, политика назначения и аудит // Защита информации. Инсайд. – 2009. – № 1. – С. 61-64.
13. *Broder A., Mitzenmacher M.* Network applications of Bloom filters: A survey // In Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing. – 2002. – P. 636-646.
14. *Колодзей А.В.* Компромисс «время/память» в реконфигурируемых вычислительных системах // Известия ЮФУ. Технические науки. – 2014. – № 12 (161). – С. 46-52.
15. *Евтеев Д.* Анализ проблем парольной защиты в российских компаниях // ЗАО «Позитив Технолоджиз». – 2009. – 33 с. – URL: http://www.securitylab.ru/analytics/381943.php.
16. *Spafford E.H.* Opus: Preventing weak password choices // Computer and Security. – 1992. – № 11. – P. 273-278.

17. *Bonneau J.* Guessing human-chosen secrets // Technical Report UCAM-CL-TR-819. – 2012. – P. 161.
18. *Марков А.С., Цирлов В.Л., Барабанов А.В.* Методы оценки несоответствия средств защиты информации. – М.: Радио и связь, 2012. – 192 с.
19. Information Assurance Implementation // Department of Defense Instruction 8500.2. – 2003. – 102 p.
20. PCI DSS Requirements and Security Assessment Procedures. Version 2.0. PCI Security Standards Council LLC – 2010. – 75 p.

REFERENCES

1. *Burnett M.* Perfect Password: Selection, Protection, Authentication, *Syngress Publishing*, 2006, pp. 194.
2. *Zarkumova R.N.* Issledovanie kolichestvennykh kharakteristik sistemy parol'noy zashchity informatsii [The study of quantitative characteristics of the system password protection information], *Sbornik nauchnykh trudov NGTU* [Proceedings of the NSTU], 2010, No. 2 (60), pp. 83-88.
3. *Snegurov A.V. Chakryan V.Kh.* Analiz ustoychivosti ko vzlomu sovremennykh mekhanizmov parol'noy zashchity operatsionnykh sistem [Analysis of the resistance to cracking modern mechanisms of password protection of operating systems], *Vostochno-Evropeyskiy zhurnal peredovykh tekhnologiy* [Eastern-European Journal of Eenterprise Technologies], 2011, Vol. 2, No. 10 (50), pp. 27-29.
4. *Markov G.A.* K voprosu ob opredelenii stoykosti parol'nykh sistem [To the question of determining the resistance of password systems], *Sbornik trudov Tret'ey vserossiyskoy NTK «Bezopasnye informatsionnye tekhnologii»* [Proceedings of the Third all-Russian research Institute of Secure information technology"], Under ed. V.A. Matveeva. Moscow: NII RL MGTU im. N.E. Baumana, 2012, pp. 21-23.
5. *Gufan K.Yu., Novosyadlyy V.A., Edel' D.A.* Otsenka stoykosti parol'nykh fraz k metodam podbora [The evaluation of resistance passphrases to the methods of selection], *Otkrytoe obrazovanie* [Open Education], 2011, No. 2, pp. 127-130.
6. *Kechedzhy K.E., Usatenko O.V., Yampol'skii V.A.* Rank distributions of words in additive many-step Markov chains and the Zipf law, *Phys. Rev. E*, 2005, Vol. 72.
7. *Hellman M.* A cryptanalytic time-memory trade-off, *IEEE Transactions on Information Theory*, 1980, Vol. 26, pp. 401-406.
8. *Ferguson Neils*. Practical Cryptography, *Indianapolis: John Wiley & Sons*, 2003, pp. 230-243.
9. *Gufan K.Yu., Novosyadlyy V.A., Edel' D.A.* O metodakh otsenki stoykosti parol'nykh fraz [Methods for assessing passphrases], *Materialy XIX nauchno-tekhnicheskoy konferentsii «Metody i tekhnicheskie sredstva obespecheniya bezopasnosti informatsii» 5-10 iyulya 2010 g* [The materials of the XIX scientific and technical conference "Methods and technical tools of information security" 5-10 July 2010]. St. Petersburg: Izd-vo Politekhn. un-ta, 2010, pp. 73-74.
10. *Markov G.A.* Metriki stoykosti parol'noy zashchity [Metrics password], *Molodezhnyy nauchno-tekhnicheskiy vestnik* [Youth Scientific and Technical Bulletin], 2013. Available at: http://www.cnpo.ru/doc/psw-metrics.pdf (Accessed 27 December 2014).
11. 1 000 000 uzhe nerabotayushchikh paroley v otkrytom dostupe. Kak my zashchishchaem pol'zovateley Yandeksa [1 000 000 already broken passwords in open access. How do we protect users of Yandex]. Available at: http://habrahabr.ru/company/yandex/blog/236007 (Accessed 15 December 2014).
12. *Belenko A.* Paroli: stoykost', politika naznacheniya i audit [Passwords: resistance, assignment policy and audit], *Zashchita informatsii. Insayd* [Protection of Information. Inside], 2009, No. 1, pp. 61-64.
13. *Broder A., Mitzenmacher M.* Network applications of Bloom filters: A survey, *In Proc. of the 40th Annual Allerton Conference on Communication, Control, and Computing*, 2002, pp. 636-646.
14. *Kolodzey A.V.* Kompromiss «vremya/pamyat'» v rekonfiguriruemykh vychislitel'nykh sistemakh [Time-memory trade-off on reconfigurable computer systems], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 12 (161), pp. 46-52.
15. *Evteev D.* Analiz problem parol'noy zashchity v rossiyskikh kompaniyakh [Analysis of the problems of password protection in the Russian companies], *ZAO «Pozitiv Tekhnolodzhiz»* [CJSC "Positive technologies"], 2009, 33 p. Available at: http://www.securitylab.ru/analytics/381943.php.

16. *Spafford E.H.* Opus: Preventing weak password choices, *Computer and Security*, 1992, No. 11, pp. 273-278.
17. *Bonneau J.* Guessing human-chosen secrets, *Technical Report UCAM-CL-TR-819*, 2012, pp. 161.
18. *Markov A.S., Tsirlov V.L., Barabanov A.V.* Metody otsenki nesootvetstviya sredstv zashchity informatsii [Methods to evaluate inconsistency means of information protection]. Moscow: Radio i svyaz', 2012, 192 p.
19. Information Assurance Implementation, *Department of Defense Instruction 8500.2*, 2003, 102 p.
20. PCI DSS Requirements and Security Assessment Procedures. Version 2.0. PCI Security Standards Council LLC, 2010, 75 p.

Статью рекомендовал к опубликованию к.ф.-м.н. В.М. Деундяк.

Тюрин Кай Андреевич – Федеральное государственное автономное научное учреждение "Научно-исследовательский институт "Специализированные вычислительные устройства защиты и автоматика" (ФГАНУ НИИ "Спецвузавтоматика"); e-mail: k.turin@niisva.org; 344002, г. Ростов-на-Дону, пер. Газетный, 51; тел.: +79885639315; лаборант.

Сёмин Роман Вячеславович – e-mail: r.semin@niisva.org; тел.: +79885181165; научный сотрудник.

Turin Kay Andreevich – "Federal State Autonomous Scientific Establishment "Scientific Research Institute "Specialized Security Computing Devices and Automation" (FSASE SRI "Spetsvuzavtomatika"); e-mail: k.turin@niisva.org; 344002, Rostov-on-Don, Gazetnyi per., 51; phone: +79885639315; assistant.

Semin Roman Vyacheslavovich – e-mail: r.semin@niisva.org; phone: +79885181165; researcher.