УДК 004.056.5 *2.3.6*

МЕТОД АУДИТА ПОЛЬЗОВАТЕЛЬСКИХ ПАРОЛЕЙ НА БАЗЕ АРХИТЕКТУРЫ НЕЙРОННОЙ СЕТИ LSTM

И.В. Аникин, Р.Р. Хаеров

Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
Российская Федерация, 420111, г. Казань, ул. К. Маркса, д. 10

Аннотация. Целью работы является повышение эффективности процедуры аудита «хорошо запоминаемых» пользовательских паролей за счет разработки метода реализации направленного их перебора от наиболее вероятных к наименее вероятным, с учетом вероятностей появления отдельных символов и их групп. Разработанный метод позволяет формировать списки паролей, упорядоченные согласно вероятности их встречаемости. Для этого использована архитектура нейронной сети LSTM (long short-term memory). Для оценки эффективности разработанного метода проведено сравнение позиций исследуемых паролей в эталонном (используемым при прямом переборе паролей) и упорядоченном (сформированном предложенным методом) списках. Показано, что предложенный метод аудита «хорошо запоминаемых» пользовательских паролей является более эффективным по сравнению с полным их перебором.

Ключевые слова: аутентификация, пользовательский пароль, аудит, нейронная сеть, LSTM.

Введение

Обеспечение информационной безопасности (ИБ) автоматизированных систем (АС) является одной из важнейших задач для устойчивого развития современных организаций. Решение данной задачи сводится к обеспечению конфиденциальности, целостности и доступности защищаемых активов исходя из перечня актуальных угроз [1]. К базовым подсистемам защиты информации, обеспечивающим выполнение вышеперечисленных свойств, относят: подсистему управления доступом, регистрации и учета событий, обеспечения целостности, криптографическую подсистему. Подсистема управления доступом выполняет функции по идентификации и аутентификации пользователей АС, по управлению доступом пользователей к объектам.

Идентификация и аутентификация пользователей является одной из первичных задач, решаемых системой защиты информации (СЗИ), и располагается на переднем крае обороны. Под идентификацией понимают [2] присвоение субъекту доступа уникального идентификатора, который предъявляется СЗИ при осуществлении доступа к объекту, то есть субъекту предлагается назвать себя. Под аутентификацией [2] понимают подтверждение субъектом доступа предъявленного идентификатора, проверка его подлинности и принадлежности именно данному пользователю. Аутентификация выполняется для устранения фальсификации на этапе идентификации, а информация, используемая для аутентификации, должна сохраняться в секрете.

Для решения задачи аутентификации пользователя наиболее часто применяются следующие методы: аутентификация по паролю, аутентификация с применением физического носителя, биометрическая аутентификация, многофакторная аутентификация и др. [3]. При этом метод аутентификации пользователя с использованием паролей является одним из основных и наиболее распространенных для СЗИ в связи с простотой своей реализации, отсутствием необходимости применения дополнительных аппаратных средств. Однако данный метод требует безопасного хранения и передачи паролей в СЗИ. Для решения последней задачи используют аппарат функций хэширования (в том числе криптографиче-

ских) [4]. Их применение позволяет обезопасить хранимые и передаваемые пароли от непосредственного изучения, а для получения доступа к паролям требуется полный перебор, который невозможно осуществить за приемлемое время при использовании паролей большой длины.

В данных условиях, одним из наиболее проблемных мест парольных методов аутентификации пользователей, является выбор «плохих» паролей, не являющихся случайными. Наиболее ярко выраженным случаем таких паролей являются дата рождения пользователя, «классические» клавиатурные комбинации либо пароли из словаря. Время подбора таких паролей может быть значительно сокращено и составлять доли секунды.

С другой стороны, применение пользователями СЗИ истинно случайных паролей также имеет ряд недостатков. В частности, такие пароли сложно запоминаемы, и их владельцы часто «сохраняют» копии паролей в записных книжках, телефонах и т.д. Для устранения последнего недостатка в настоящее время при проектировании СЗИ часто применяют генераторы «хорошо запоминаемых» паролей [5], в которых соблюдена «золотая середина» между истинной случайностью с одной стороны и близостью к естественному языку с другой. Примерами таких паролей являются *Apesna, Stindi, QoTwaz, bunque, naXkiv.*

Для оценки качества (аудита) применяемых пользовательских паролей могут применяться различные способы атаки на них, целью которых является установление факта возможности их подбора за приемлемое время. Наиболее известными инструментами аудита, позволяющими автоматизировать перебор паролей, являются L0phtCrack и SAMInside [8], выполняющие попытки подобрать пароль способами «грубой силы» либо атакой по словарю. Функцией аудита паролей наделено также большинство сканеров безопасности компьютерных сетей. Характеристика способов атаки и оценка их эффективности представлены в таблице 1.

Таблица 1. Способы атаки на парольные системы

Способ атаки	Характеристика	Оценка эффективности	Примеры перебираемых паролей
Метод грубой силы (brute-force)	Полный последовательный перебор всевозможных комбинаций символов пароля, охватывая полное пространство решений (A^L, где A – мощность алфавита паролей, а L – длина паролей)	Способ применяется для оценки возможности подбора истинно случайных паролей и «хорошо запоминаемых» паролей. Время решения задачи в худшем случае: A^L / V сек., где V- скорость перебора паролей в сек.	pQrtHg xchAks и др.
Атака по словарю	Последовательный перебор всех паролей из заданного словаря, включающего наиболее часто используемые пароли	Способ эффективен для выявления «очень плохих» паролей, выбранных пользователем из заданного словаря. Время решения задачи в худшем случае: N / V, где N – количество строк словаря (на практике, существуют словари, включающие около 10^7 позиций, V – скорость перебора паролей в сек.	home, password и др.

Из представленной таблицы видно, что аудит хорошо запоминаемых паролей в настоящее время выполняется только методом грубой силы (brute-force), не учитывающим логику построения «хорошо запоминаемых» паролей, как правило используемых на практике, и в этом смысле осуществляющим избыточный перебор возможных вариантов. Для повышения эффективности аудита хорошо запоминаемых паролей необходимо осуществлять их «направленный» перебор – начиная с наиболее вероятных комбинаций символов, и заканчивая наименее вероятными. Тем самым, «направленный» перебор позволяет сократить пространство перебора «хорошо запоминаемых» паролей и выявлять среди них быстро подбираемые («плохие»), которые должны быть заменены. Бурное развитие в настоящее время интеллектуальных методов обработки текстовой информации, позволяет помочь в решении такой задачи.

Целью работы является повышение эффективности процедуры аудита «хорошо запоминаемых» пользовательских паролей за счет разработки метода реализации «направленного» их перебора от наиболее вероятных к наименее вероятным, с учетом вероятностей появления в них отдельных символов и их групп. В данной работе предлагается метод аудита пользовательских паролей, основанный на применении архитектуры нейронных сетей LSTM (long short-term memory).

Для достижения поставленной цели необходимо решить следующие задачи:

- формирование обучающей выборки для обучения нейронной сети;
- конкретизация нейросетевой архитектуры LSTM для реализации «направленного перебора» паролей;
- обучение предложенной нейросетевой архитектуры;
- формирование ранжированного (упорядоченного) списка «хорошо запоминаемых» пользовательских паролей для реализации направленного перебора;
- оценка эффективности направленного перебора путем выборочного сравнения исследуемых позиций «хорошо запоминаемых» паролей в ранжированном списке и списке, используемом при полном переборе.

Практическая ценность предложенного метода заключается в том, что направленный перебор позволяет более быстро по сравнению с brute-force подбирать наиболее вероятные «хорошо запоминаемые» пароли, выявляя тем самым уязвимые места в системе для своевременного их устранения.

Способы хранения и аудита пользовательских паролей. Генераторы хорошо запоминаемых паролей

Один из основных способов безопасного хранения пользовательских паролей в СЗИ АС основан на применении функций хэширования. При этом применяются две типовые схемы [6]. Обозначим через K – пользовательский пароль, S – символы привязки («соль»), а через H(K) – некоторую функцию хэширования сообщений. Первая типовая схема безопасного хранения пользовательских паролей использует хранение закрытой информации E=H(K) для пользовательской учетной записи, вместо открытой K. Вторая типовая схема предполагает применение «соленых» хэшей [7] и хранение информации E=H(S,K) вместо открытой K. Первая типовая схема безопасного хранения пользовательских паролей используется, например, в ОС Windows, вторая – в ОС семейства Linux. Применение «соли» для функций хэширования не позволяет распараллелить перебор паролей для учетных записей пользователей, а также обеспечивает уникальность хэша даже при выборе пользователями одинаковых паролей.

При использовании хороших функций хэширования, обладающих свойствами необратимости, рассеивания и чувствительности к изменениям, единственно возможным способом подбора пользовательского пароля по известной информации E, является полный перебор всевозможных значений K и сравнение полученных хэшей с E. При использовании

достаточной длины и алфавита паролей, данная задача не может быть решена за приемлемое время. В частности, выбрав длину пароля L=10 и алфавит, включающий в себя заглавные и малые английские буквы и цифры (A=2*26+10=62), время полного перебора паролей со скоростью $V=10^8$ паролей/сек составит $T = A^L/V \approx 266$ лет.

Однако, как отмечено ранее, в настоящее время хорошей практикой считается формирование не истинно случайных, а применение генераторов хорошо запоминаемых паролей. Использование такого подхода исключает формирование сложно запоминаемых паролей, которые владельцы будут вносить в записные книжки, телефоны, записывать на «стикеры» и т.д. Данные генераторы используют вероятностные методы, учитывающие вероятности появления символов и групп символов на конкретных позициях. Математический аппарат данных методов, как правило, основан на применении Марковских моделей [9] и контекстно-свободных грамматик [10]. В таблице 2 представлены наиболее известные генераторы запоминающихся паролей.

Таблица 2. Наиболее известные генераторы хорошо запоминаемых паролей

Наименование генератора	Ссылка
Garbler	github.com/michaelbironneau/garbler
ClaveSegura	www.clavesegura.org/ru/
Seotools	www.useotools.com/ru/memorable-password-generator
Omgopass	github.com/omgovich/omgopass
lastpass	www.lastpass.com/features/password-generator
Warpconduit	www.warpconduit.net/password-generator/
Gpw-strength	www.goskyhawk.com/pwd/gpw-strength.html
Pronounceable password generator	caseyjmorris.github.io/pronounceablePassword/
Mxtoolbox	mxtoolbox.com/PasswordGenerator/
Markov Chain algorithm	www.outsideopen.com/password/
GenPas	genpas.narod.ru/
Генератор удобно произносимых паролей	pr-cy.ru/password/
Abbrase	github.com/rmmh/abbrase

Несмотря на очевидное преимущество применения таких генераторов, отсутствие строгой случайности формируемых пользовательских паролей дает возможность сокращения пространства их перебора, а также возможность осуществления «направленного» перебора паролей от более вероятных к менее вероятным. Тем самым, сгенерированные пользовательские пароли могут стать небезопасными, их подбор может занять небольшой промежуток времени. Такие пароли должны быть оперативно выявлены и заменены.

Для аудита пользовательских паролей с учетом оценки вероятности появления отдельных символов и их групп, был разработан и реализован метод, основанный на архитектуре нейронных сетей LSTM.

Архитектура LSTM

LSTM представляет собой вид рекуррентных нейронных сетей с обратными связями и запоминанием информации [11]. Архитектура блока LSTM представлена на рис. 1 и включает в себя следующие компоненты: ячейку памяти (*c*), входной вентиль (*i*), выходной вентиль (*o*) и вентиль забывания (*f*). Входной вентиль осуществляет контроль за поступлением новых значений в ячейку памяти, выходной вентиль – контроль за тем, в какой степени значение, находящееся в памяти, используется при расчёте выходной функции активации для блока, вентиль забывания осуществляет контроль за сохранением информации в памяти с целью ее использования в будущих вычислениях. Индекс t отражает номер блока LSTM, значение x_t – входной вектор, h_t – выходной вектор, σ - функция активации на основе сигмоиды, tanh – функция активации на основе гиперболического тангенса.

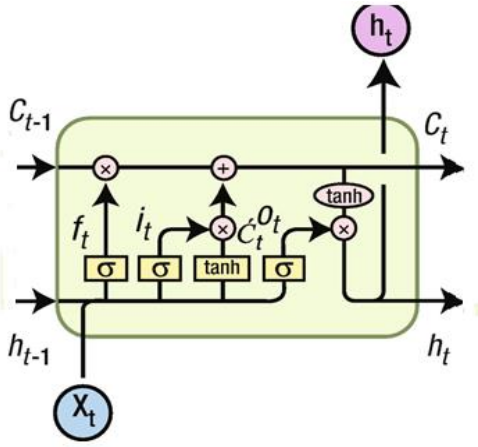

Рис.1. Архитектура блока LSTM

В нейросетевой архитектуре LSTM представленные на рис. 1 блоки объединяются в цепочки (рис. 2).

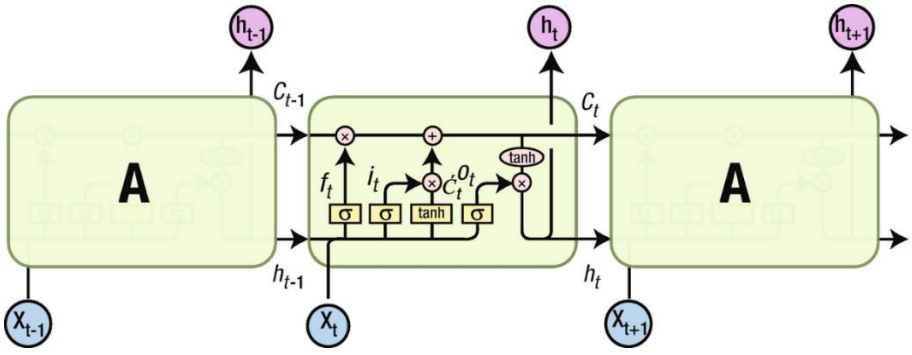

Рис. 2. Архитектура нейронной сети LSTM

Дальнейшим развитием данной архитектуры является bidirectional LSTM (рис.3) [12] и stacked bidirectional LSTM [13], реализующая глубокий двусторонний анализ входной информации с применением множества слоев.

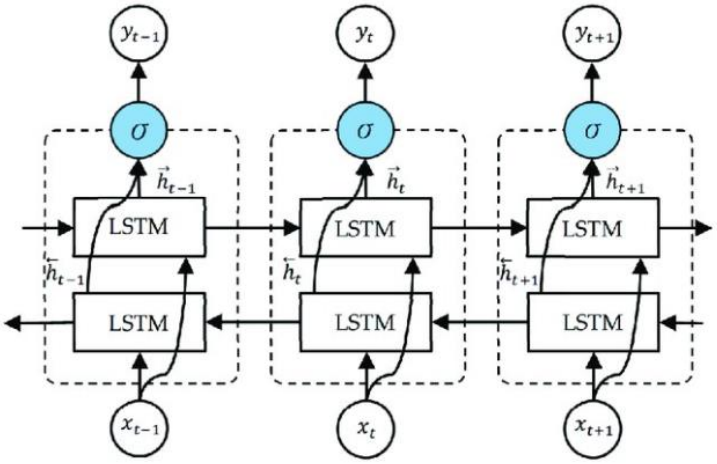

Рис. 3. Архитектура двунаправленой LSTM

Обучение нейросетевой архитектуры LSTM осуществляется с помощью классического алгоритма обратного распространения ошибки [14].

LSTM архитектуры нашли большое практическое применение для решения задач обработки текстовой информации: классификации [15], генерации синтетических текстов [16], машинном переводе [17], QA (вопросно-ответных) систем [18] и т.д. В связи с этим, актуальным видится применение данных архитектур для решения задачи аудита хорошо запоминаемых пользовательских паролей.

Метод аудита «хорошо запоминаемых» пользовательских паролей с применением архитектуры нейронной сети LSTM

Метод аудита «хорошо запоминаемых» пользовательских паролей предполагает формирование упорядоченного их перечня, от наиболее вероятных к наименее вероятным, с учетом вероятностей появления конкретных групп символов. Данный перечень может быть использован для осуществления «направленного» перебора паролей, вместо обычного brute-force, проверяя в первую очередь наиболее вероятные пользовательские пароли, а в последнюю очередь – наименее вероятные.

Формирование обучающей выборки

Для формирования обучающей выборки использовались два из представленных в таблице 1 генераторов паролей: Garbler и Pronounceable password generator. С помощью каждого из них сформировано по 1 млн паролей (всего 2 млн). Использована следующая конфигурация формируемых паролей – алфавит включает в себя заглавные и малые английские буквы, длина паролей равна шести символам. Из сформированного списка были исключены повторяющиеся пароли. Ниже представлены примеры сгенерированных паролей:

Apesna
Stindi
QoTwaz
bunque
naXkiv

Применяемая нейросетевая модель осуществляет прогнозирование вероятностей появления следующих символов пароля, путем анализа нескольких предыдущих. В связи с этим, обучающая выборка представляла собой результат обработки сформированной последовательности паролей методом скользящего окна [19]. Использована ширина окна равная двум символам, горизонт планирования – один символ (следующий за исследуемыми двумя). В частности, из сформированного пароля QoTwas была сформирована следующая последовательность для обучающей выборки нейронной сети (таблица 3):

Таблица 3. Элемент обучающей выборки для пароля QoTwas

Вход модели	Выход модели
/Q	O
Qo	T
oT	W
Tw	A
wa	s

Знак “/” в первой строке говорит о том, что следующий символ пароля будет первым.

Общее количество элементов обучающей выборки – 10 млн значений. Данная выборка была поделена на обучающую (75%) и тестовую (25%). В связи с тем, что нейросетевая модель требует задания числовых знаний на своем входе и формирует числовые значения на выходе, применено числовое кодирование входов и выходов нейросетевой модели.

Конкретизация нейросетевой архитектуры

Нейросетевая архитектура, используемая для формирования упорядоченного перечня пользовательских паролей, представлена на рис. 4.

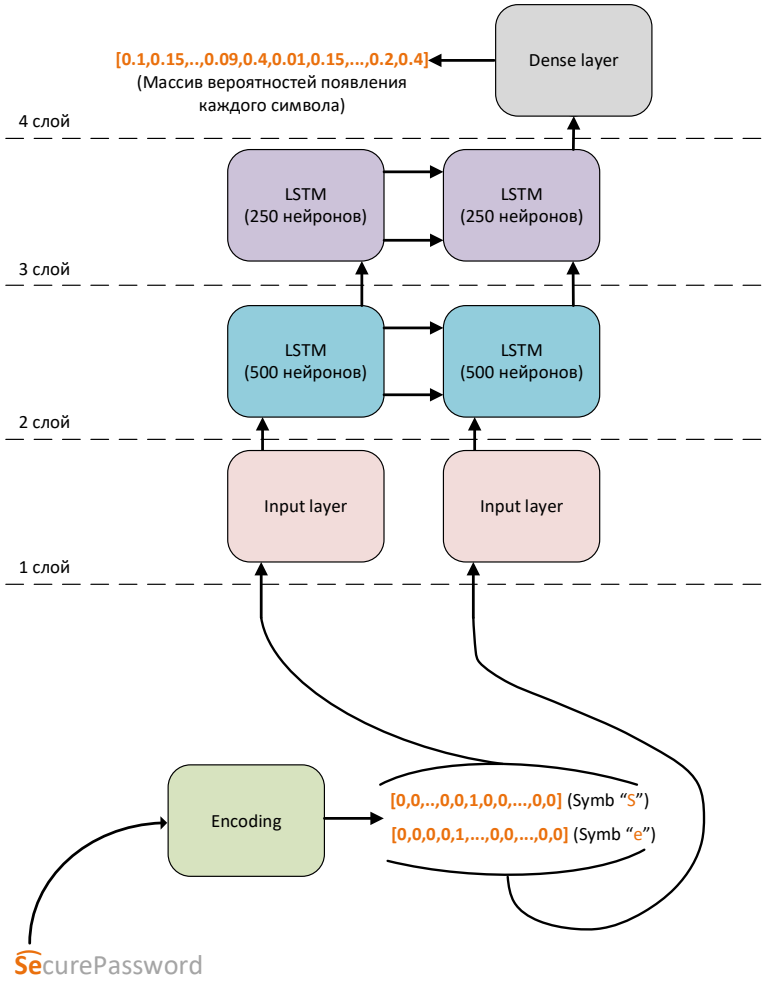

Рис. 4. Архитектура нейронной сети на базе LSTM, используемая для формирования упорядоченного перечня пользовательских паролей

Представленная архитектура нейронной сети включает в себя входной слой, два внутренних слоя (по 500 и 250 блоков LSTM соответственно), а также Dense layer, используемый для формирования массива с вероятностями появления каждого из символов в качестве следующего. При обучении нейросетевой модели использовались следующие параметры: метод оптимизации ADAM, количество эпох обучения – 100, dropout = 0,5.

Особенности применения нейросетевой модели

При заданном первом символе пароля, последовательное применение предложенной нейросетевой модели позволяет формировать векторы вероятностей появления последующих символов. Данный процесс применительно к формированию 4-го символа пароля наглядно представлен на рис. 5. Итоговая вероятность полученного пароля определяется как произведение вероятностей появления отдельных символов.

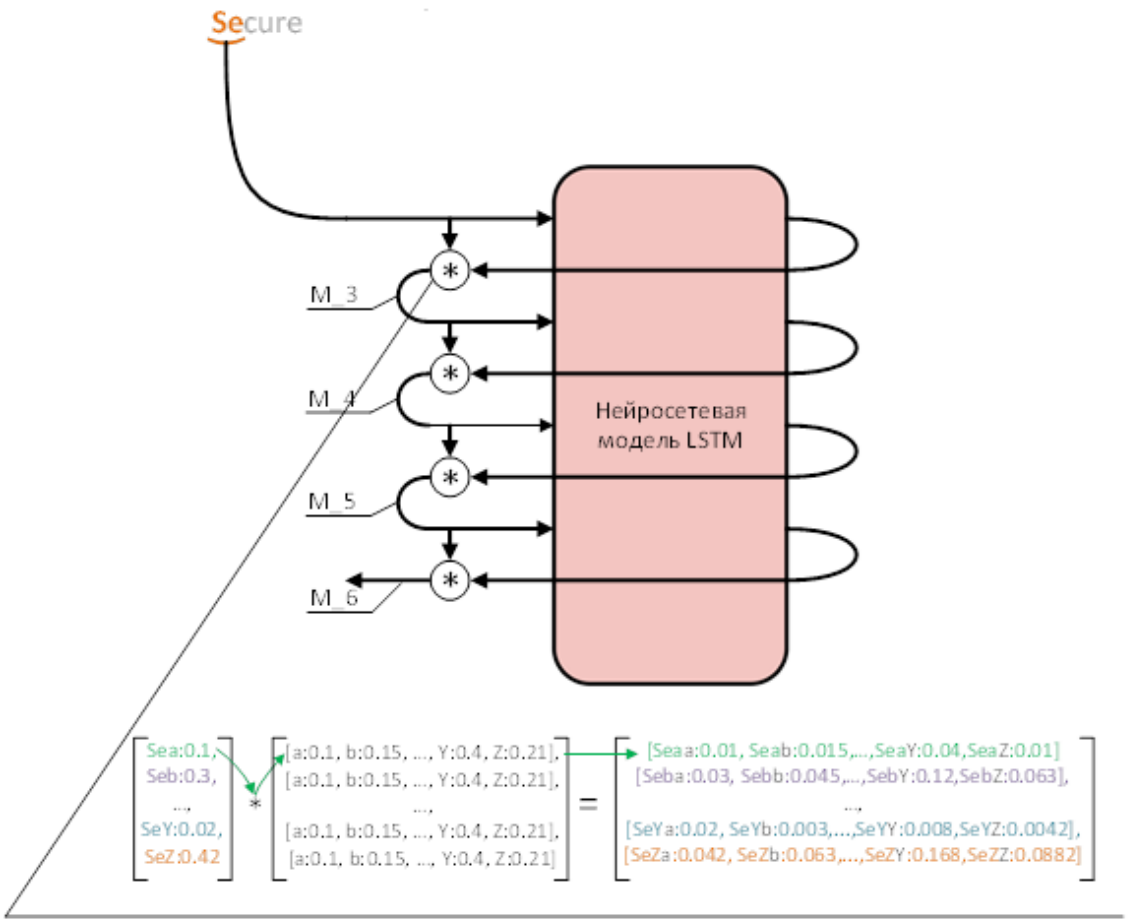

Рис. 5. Пример формирования 4-го символа пароля по известным 3

Финальным этапом, после формирования 6-го символа пароля, является сортировка полученного массива паролей по убыванию вероятностей. Тем самым формируется ранжированный (упорядоченный) по вероятностям список «хорошо запоминаемых» паролей.

Предложенный метод аудита пользовательских паролей был реализован на языке программирования Python, в среде разработки Jupyter Notebook. Построение нейросетевой модели осуществлялось с помощью библиотеки tensorflow.keras. Для работы с данными использовались библиотеки numpy и pandas.

Экспериментальные исследования

Оценка эффективности работы представленного метода осуществлялась на тестовой выборке, составляющей 25% от общего числа паролей. Эффективность метода определялась его преимуществом перед полным перебором паролей. Для этого определялись следующие показатели каждого из пароля К тестовой выборки (таблица 4):

- позиция пароля К в ранжированном списке (сформированном нейросетевой моделью);
- позиция пароля К в эталонном списке (в случае полного перебора паролей);
- вероятность пароля К в ранжированном списке;
- вероятность пароля К в эталонном списке (имеет одно и то же значение для полного перебора паролей).

Таблица 4. Выборка по показателям эффективности

№	Позиция в ранжированном списке, сформированном нейронной сетью	Позиция в эталонном списке brute-force	Вероятность в ранжированном списке, сформированном нейронной сетью	Вероятность в эталонном списке brute-force
1.	858 996	4 570 239	2*10-5	1,4*10-9
2.	793 813	1 040 248	3,1*10-4	1,4*10-9
3.	2 043 134	1 580 485	4,4*10-7	1,4*10-9
4.	1 959 465	5 583 234	1,2*10-7	1,4*10-9
…				

Видим, что в общем случае позиция проверяемого пароля в ранжированном списке находится выше, чем его позиция в списке brute-force, то есть пароль будет тестироваться раньше обычного. Тем самым, предлагаемый метод аудита пользовательских паролей является более эффективным, по сравнению с обычным brute-force.

Обсуждение полученных результатов

Проведенные эксперименты продемонстрировали предполагаемый факт - то, что за счет учета характера распределения символов в «хорошо запоминаемых» паролях, их позиция в ранжированном списке при реализации направленного перебора будет выше, чем позиция в эталонном списке brute-force. Таким образом, «направленный» перебор позволяет более быстро по сравнению с brute-force подбирать наиболее вероятные «хорошо запоминаемые» пароли. *Практическая ценность* предложенного метода заключается в том, что в данном случае реализуется более быстрый подбор «плохих» паролей, тем самым повышая эффективность поиска уязвимых мест в системе защиты для своевременного их устранения.

Условия применимости предложенного метода

Предложенный метод будет эффективен только при решении задачи аудита «хорошо запоминаемых» паролей, имеющих не равновероятное распределение своих символов. Как только, в качестве паролей пользователями будут выбираться истинно случайные последовательности, предлагаемый метод будет менее эффективен по сравнению с классическим brute-force, так как он ставит подобные пароли в конец списка.

Кроме этого, предлагаемый метод, очевидно, будет намного менее эффективен при «атаке по словарю», реализующей последовательный перебор всех паролей из заданного словаря, включающего наиболее часто используемые пароли. Наиболее полные словари включают в себя не более 10^7 слов. Учитывая скорость перебора паролей *V* современных

средств аудита паролей (порядка 4 млн паролей в секунду на PC со средними характеристиками), предлагаемый метод можно рассматривать не как замену, а как дополнение к словарной атаке.

Таким образом, можно рекомендовать следующую последовательность применения методов аудита паролей в компьютерной системе:

1. Поиск «очень плохих» паролей путем атаки по словарю. В данном случае производится проверка пользовательских паролей на наличие их заданном словаре.
2. Поиск «плохих» паролей путем реализации направленного перебора паролей предложенным методом.

В заключении необходимо отметить, что в работе не исследовались национальные и половозрастные особенности выбора пользовательских паролей. Учет данных особенностей приведет к существенному перераспределению паролей в ранжированном списке и падению эффективности предложенного метода. Принимая во внимание высокую актуальность проведения такого исследования с одной стороны, и сложность его проведения с другой, авторы оставляют данный вопрос на перспективу.

Заключение

Предложенный метод эффективен только при решении задачи аудита «хорошо запоминаемых» паролей, имеющих неравновероятное распределение своих символов. Кроме этого, предлагаемый метод, очевидно, будет намного менее эффективен при «атаке по словарю», реализующей последовательный перебор всех паролей из заданного словаря, включающего наиболее часто используемые пароли. Таким образом, предлагаемый метод можно рассматривать к применению в дополнение к словарной атаке.

Предложенный метод позволит более адекватно оценить риски нарушения информационной безопасности и при необходимости снизить их путем выбора более безопасных вариантов паролей. В качестве дальнейшей перспективы, планируется проведение подобных исследований с более сложными нейросетевыми архитектурами обработки текстовой информации, построенных на базе Transformer [20], а также исследование национальных и половозрастных особенности выбора пользовательских паролей

Список литературы

1. Аникин И.В. Управление рисками информационной безопасности: учебное пособие / И.В. Аникин. – Казань: Редакционно-издательский центр «Школа», 2018. – 160 с.

2. Шаньгин В.Ф. Информационная безопасность и защита информации / В.Ф. Шаньгин. – М.: ДМК Пресс, 2017. – 702 с.

3. Сабанов А.Г. Идентификация и аутентификация в цифровом мире / А.Г. Сабанов, А.А. Шелупанов. – М.: Горячая Линия – Телеком, 2022. – 356 с.

4. Панфилов Ю.Б. Формирование функции хэширования паролей, стойкой к ускоренному перебору значений / Ю.Б.Панфилов // Журнал радиоэлектроники. – 2018. – №3. – С.12-23.

5. Walia K.S. An empirical analysis on the usability and security of passwords / K.S. Walia, S. Shenoy, Y. Cheng // 21st IEEE international conference on information reuse and integration for data science. – 2020. – P.1-8.

6. Хорев П.Б. Программно-аппаратная защита информации: учебное пособие. – М.: Форум, 2019. – 352 с.

7. Бирюков А.А. Информационная безопасность: защита и нападение / А.А. Бирюков.– М.: ДМК Пресс, 2022. – 472 с.

8. Скембрей Д. Секреты хакеров. Безопасность Windows Server 2003 – готовые решения / Д. Скембрей, С. Мак-Клар. – М.: Издательский дом Вильямс, 2004. – 512 с.

9. Ma J., Yang W., Luo M. A Study of Probabilistic Password Models / J. Ma, W. Yang, M. Luo // Security and Privacy. – 2014. – P. 689-704.

10. Weir M. Aggarwal S., Medeiros B. D. Password Cracking Using Probabilistic Context-Free Grammars / M. Weir, S. Aggarwal, B.D. Medeiros // Security and Privacy. – 2009. – P. 391-405.

11. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computations. – 1997. - 9 (8). – P. 1735-1780.

12. Graves A. Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition / A. Graves, S. Fernandes, J. Schmidhuber // Proceedings of Artificial Neural Networks: Formal Models and Their Applications, 2005. – P. 799-804.

13. Haag K. Bidirectional LSTM Networks Employing Stacked Bottleneck Features for Expressive Speech-Driven Head Motion Synthesis / K. Haag, H. Shimodaora // Proceedings of International Conference on Intelligent Virtual Agents, 2016. – P. 198-207.

14. Шолле Ф. Глубокое обучение на Python / Ф. Шолле. – СПб: Питер, 2022. - 400 с.

15. Liu P. Recurrent Neural Network for Text Classification with Multi-Task Learning / P. Liu, X. Qiu, X. Huang // Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016. – P. 2873-2879.

16. Dhal I. Text Generation Using Long Short-TermMemory Networks / I. Dhal, S.Vashisth, S.Saraswat // Micro-Electronics and Telecommunication Engineering, 2020. - P. 649-657.

17. Ramaiah M. Study of Neural Machine Translation With Long Short Term Memory Techniques / M. Ramaiah, D. Datta, R. Agarwal, V. Chandrasekaran // Deep Learning Research Applications for Natural Language Processing, 2023. – P. 65-88.

18. Tomer M. Question Answering System Using LSTM and Keyword Generation / M. Tomer, M. Kumar // Advances in Information Communication Technology and Computing. – 2021. – P. 271-281.

19. Fedorova A.A., Anikin I.V., Beliautsou V.A. Prediction vehicle's speed with using artificial neural networks / A.A. Fedorova, I.V. Anikin, V.A. Beliautsou // Proceedings - 2020 International Russian Automation Conference RusAutoCon, 2020. – P.11-15.

20. Lin T. A survey of transformers / T. Lin, Y. Wang, X. Liu, X. Qiu // AI Open, 2022. – 3. – P.111-132.

METHOD FOR USER'S PASSWORDS AUDIT BASED ON LSTM NEURAL NETWORK

I.V. Anikin, R.R. Haerov

Kazan National Research Technical University named after A. N. Tupolev-KAI
10, st. Karl Marx, Kazan, 420111, Russian Federation

Annotation. The aim of the work is to increase the efficiency of the audit procedure for "well-remembered" user passwords by developing a method for implementing their directed enumeration from the most probable to the least probable, taking into account the probabilities of occurrence of individual characters and their groups in them. The suggested method makes generates lists of passwords ordered according to their probability of occurrence. We used LSTM neural network for that. To evaluate the effectiveness of the suggested method, we compared the positions of passwords in the ordered (formed by the suggested method) and reference (used in brute-force) lists. We showed that the suggested method of auditing user passwords is more efficient than brute-force.

Keywords: authentication, password, audit, neural network, LSTM.

Статья представлена в редакцию 13 июня 2023 г.