

Пароли: стойкость, политики назначения и аудит

Пароли представляют собой наиболее распространенный инструмент, используемый для получения доступа к информационным ресурсам. Однако не всегда его использование оправдано с точки зрения компьютерной безопасности. Какие же условия необходимо выполнить, если требуется обеспечить надежную парольную аутентификацию пользователей?

А. Беленко, криптоаналитик
Компания «Элкомсофт»

Введение

Пароли традиционно являются наиболее распространенным и доступным средством аутентификации. Для пользователей это один из самых удобных вариантов, который, к тому же, не требует наличия какого-либо дополнительного оборудования или специальных навыков. Все что требуется – это придумать и запомнить стойкий пароль (или несколько паролей). Кажется, это совсем не сложно? К сожалению, суровая реальность вносит свои коррективы в этот процесс.

Согласно проведенному американским Ponemon Institute опросу¹, 88 % респондентов в течение последних двух лет забывали свой пароль хотя бы один раз. Часто сотрудники компаний считают политику безопасности в отношении паролей излишней, и требование менять пароль каждые 30 дней вызывает у них бурю негодования. В таких случаях в качестве нового пароля используется немного видоизмененный старый, к которому добавляется «0», «1» или текущий год.

К сожалению, с точки зрения компьютерной безопасности паро-

льную аутентификацию далеко не всегда можно назвать удачным выбором. Как и многие технические решения, она страдает от двух проблем: человеческого фактора и технического несовершенства.

- **Человеческий фактор** (ошибки пользователей). Многие люди не могут (или попросту не хотят, не видят смысла) запоминать стойкие пароли, поскольку это объективно сложно. Поэтому они используют простые, нестойкие пароли, или, если их все же заставляют использовать стойкие, записывают их на ковриках для мышки, на обороте клавиатуры или на стикерах, приклеенных к монитору.
- **Техническое несовершенство** (ошибки разработчиков), чаще всего заключающееся в ошибках, допущенных на этапах проектирования и/или реализации программного обеспечения, осуществляющего проверки паролей.

Стойкость паролей

Мерой стойкости паролей традиционно является *энтропия* – мера неопределенности, – измеряемая обычно в битах. Энтропия в 1 бит соот-



¹ Those Pesky Passwords, http://www.csoonline.com/article/221733/Those_Pesky_Passwords.

ветствует неопределенности выбора из двух паролей, в 2 бита – из 4 паролей, в 3 бита – из 8 паролей и т. д. Энтропия в N бит соответствует неопределенности выбора из 2^N паролей.

В случае случайных паролей (например, сгенерированных с помощью генератора случайных чисел) энтропия вычисляется достаточно просто: она равна логарифму по основанию два количества возможных паролей для заданных параметров. Так, для случайного пароля длиной N символов, составленного из алфавита, содержащего M букв, энтропия будет равна $E = \log_2 M^N$. Значения энтропии для некоторых длин паролей и наборов символов представлены в табл. 1.

Если же пароль сгенерирован не беспристрастным генератором случайных чисел, а человеком, то вычислить его энтропию гораздо труднее. Самым распространенным подходом к подсчету энтропии в этом случае является подход, который был предложен американским институтом NIST²:

- энтропия первого символа пароля составляет 4 бита;
- энтропия следующих семи символов пароля составляет 2 бита на символ;
- энтропия символов с 9-го по 20-й составляет 1,5 бита на символ;
- все последующие символы имеют энтропию 1 бит на символ;
- если пароль содержит символы верхнего регистра и неалфавитные символы, то его энтропия увеличивается на 6 бит.

Что влияет на стойкость пароля кроме самого пароля?

Стойкость того или иного пароля должна рассматриваться только в контексте конкретной системы парольной аутентификации: пароль, являющийся стойким для одной системы, может оказаться совершенно не стойким при использовании другой. Это происходит из-за того, что разные системы в разной степени реализуют (или вовсе не реализуют)

Таблица 1

Длина пароля	Цифры (10) (1)	Латинские буквы без учета регистра (26)	Цифры и латинские буквы без учета регистра (36)	Латинские буквы с учетом регистра (52)	Цифры и латинские буквы с учетом регистра (62)	Цифры, латинские буквы и специальные символы (96)
6	19,9	28,2	31,0	34,2	35,7	39,5
7	23,3	32,9	36,2	39,9	41,7	46,1
8	26,6	37,6	41,4	45,6	47,6	52,7
9	29,9	42,3	46,5	51,3	53,6	59,3
10	33,2	47,0	51,7	57,0	59,5	65,8
11	36,5	51,7	56,9	62,7	65,5	72,4
12	39,9	56,4	62,0	68,4	71,5	79,0

Таблица 2

Где применяется	Количество итераций, алгоритм	Изменение эффективной энтропии пароля
WPA	4×4096 SHA-1 (PBKDF2)	+14 бит
MS Office 2007	50 000 SHA-1	+15,6 бит
WinRAR 3.0+	~100 000 SHA-1	+16,6 бит
PGP 9.0+	2×1024 SHA-1	+11 бит
Adobe Acrobat 5.0–8.0	50 MD5 + 20 RC4	+6,1 бит

механизмы противодействия атакам, направленным на взлом паролей, а также потому, что некоторые системы содержат ошибки или используют ненадежные алгоритмы. Основной способ противодействовать взлому паролей – искусственно замедлить процедуру их проверки. Действительно, займет ли проверка 10 наносекунд или 10 миллисекунд – для пользователя разница будет совершенно незаметной, а с точки зрения взлома скорость упадет очень существенно – со 100 миллионов до 100 паролей в секунду! Замедление обычно достигается за счет многократного вычисления криптографических функций, причем эти вычисления построены таким образом, чтобы атакующая сторона не могла проверить пароль без повторения вычислений (то есть недостаточно просто добавить вызов Sleep(10) в процедуре проверки пароля). Впервые такой вариант был предложен в 1997 году в работе «Secure Applications of Low-Entropy Keys»³. Системы, реализующие подобные замедления, увеличивают эффективную энтропию пароля на величину $\log_2 C$, где C – количество итераций при вычислении криптографического преоб-

зования. Примеры подобных систем приведены в табл. 2.

Как было сказано выше, системы могут содержать ошибки, снижающие ее стойкость.

Например, для шифрования данных может использоваться алгоритм с ключом недостаточной длины или сама процедура преобразования пароля может быть ненадежной. Так, все версии Microsoft Word и Excel, начиная с 97 и до 2003 включительно, для шифрования документов по умолчанию используют потоковый алгоритм RC4 с длиной ключа 40 бит. Дело в том, что до 1998 года в США существовали ограничения на экспорт программных продуктов, реализующих шифрование, в том числе длина ключа была ограничена 40 битами для симметричных и 512 битами для ассиметричных алгоритмов. Сейчас эти ограничения сняты, но для обеспечения совместности со старыми продуктами все их новые версии также поддерживают этот алгоритм защиты документов. При использовании коротких 40-разрядных ключей взломщик может не заниматься восстановлением пароля, а просто подобрать ключ (это занимает несколько секунд на

² NIST Special Publication SP 800-63 «Electronic Authentication Guideline», http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.

³ <http://www.schneier.com/paper-low-entropy.pdf>.

обычном компьютере). Стойкость системы определяется стойкостью самого слабого звена, поэтому если известно, что для шифрования используется 40-битный ключ, нет никакого смысла использовать пароль с большей энтропией.

Другой, ставший уже классическим, пример – алгоритм проверки паролей в операционной системе Windows NT/2000/XP/2003. Эти системы поддерживают два алгоритма – LM и NTLM. При создании пользователя или изменении его пароля эти системы вычисляют LM- и NTLM-хэши пароля и сохраняют его в базе данных учетных записей. Алгоритм вычисления LM-хэша выглядит следующим образом:

- символы пароля приводятся к верхнему регистру и преобразуются в OEM-кодировку;
- если пароль короче 14 символов, он дополняется пробелами до этой величины;
- полученная последовательность из 14 символов делится на две части по 7 символов;
- каждая из полученных частей используется в качестве ключа для шифрования фиксированного значения алгоритмом DES;
- полученные в результате шифрования блоки объединяются – это и есть LM-хэш пароля.

Из описания алгоритма видны его серьезные недостатки. Во-первых, происходит преобразование пароля к верхнему регистру, что значительно снижает его энтропию. Так, энтропия пароля длиной 7 символов, состоящего из латинских букв и цифр, будет снижена с 41,7 до 36,2 бит – более чем в 45 раз! Во-вторых, алгоритм ограничивает эффективную длину пароля семью символами – последние 7 символов пароля могут перебираться независимо от первых семи. Это очень серьезная проблема, из-за нее при использовании алгоритма LM нет смысла в применении паролей длиннее 7 символов – стойкость от этого не увеличится. Таким образом, данный алгоритм не надежен и по возможности стоит избегать его использования.

Из всего вышесказанного делаем логичный вывод: при использовании паролей необходимо знать осо-

бенности используемой системы аутентификации и учитывать эти особенности при работе с ней. Для большинства распространенных приложений (Windows, Linux, Microsoft Office) особенности парольной защиты хорошо исследованы и информация о них доступна в Интернете. Для менее распространенных операционных систем и приложений самый надежный способ получения информации об особенностях парольной аутентификации – найти квалифицированного специалиста, который проведет экспертизу системы, составит развернутый отчет и представит рекомендации по ее использованию.

Человеческий фактор и политика безопасности

Человек остается самым слабым звеном в парольной аутентификации. Исследования показывают, что более 60 % респондентов считают большим неудобством необходимость запоминать пароли, поэтому в качестве таковых часто используют номера телефонов, даты рождения, клички домашних животных, имена друзей-подруг, названия любимых фильмов или торговых марок. Пользователь полагает, что такая информация может быть известна только ему, в крайнем случае – узкому кругу доверенных лиц. И совершает большую ошибку. Вопросы человеческого фактора, как правило, решаются административными мерами через утверждение и введение в организации политики безопасности, в которой явно указаны требования к стойкости паролей, периодичности их смены, (не)допустимости повторения и т. д.

Для создания эффективной политики информационной безопасности необходимо быть уверенным в надежности программного обеспечения, реализующего парольную аутентификацию, знать присутствующие в нем ошибки и понимать его ограничения. Например, если утвержденная политика безопасности не предусматривает отключения алгоритма LM, то и задавать в той же политике минимальную длину пароля более 7 символов или требо-

вать присутствия в пароле символов верхнего и нижнего регистров совершенно бессмысленно. В данном случае слабым звеном является алгоритм LM, и политика безопасности должна это учитывать. На мой взгляд, этот пример очень хорошо показывает необходимость составления политики безопасности паролей с учетом результатов экспертизы, иначе ее эффективность оставит желать лучшего и кроме ложного чувства защищенности ничего не даст.

Типовые требования к паролям, формулируемые в политике безопасности, выглядят так:

- пароль должен быть не короче 8 символов;
- пароль должен содержать хотя бы одну букву и хотя бы одну цифру;
- пароль не должен существенно отличаться от предыдущего пароля;
- пароль не должен совпадать с идентификатором пользователя (логином);
- пароль не должен совпадать с осмысленным словом какого-либо языка;
- необходимо менять пароль не реже одного раза в 3 месяца;
- пароль не должен содержать имени, фамилии, отчества или инициалов пользователя;
- в качестве пароля нельзя использовать личные данные, такие как номера телефонов, номер паспорта или другого документа, номер автомобиля и т. п.;
- пароли для доступа к разным системам и службам должны различаться.

Аудит паролей

После утверждения политики безопасности паролей возникает очередная проблема – как контролировать ее выполнение? В ряде случаев можно настроить программное обеспечение таким образом, чтобы проверка пароля на соответствие политике безопасности происходила непосредственно при смене пароля пользователем, и пароли, не удовлетворяющие политике, не принимались бы системой. К сожалению, это не всегда реализуемо, поэтому традиционным способом контроля

соблюдения политики является парольный аудит. При аудите паролей проверяющая сторона ставит себя на место злоумышленника и пытается взломать максимальное количество паролей за определенное время (или уложившись в определенный бюджет). Чем больше паролей будет найдено – тем хуже дела у проверяемой организации.

В идеале аудит паролей должен выполняться как составная часть комплексной проверки безопасности, такой как *penetration testing*, так как его эффективность при этом максимальна. К сожалению, не все могут позволить себе регулярно проводить подобные проверки, поэтому зачастую парольный аудит осуществляется как отдельное мероприятие. Проведение такого аудита – задача не слишком сложная, особенно если учитывать значительный выбор специализированного программного обеспечения.

Приведу в качестве примера типичную последовательность действий при аудите паролей пользователей в домене Microsoft Windows. Во-первых, необходимо получить исходные данные – зашифрованные пароли, хранящиеся на контроллере домена. Сделать это можно, например, с помощью бесплатных утилит *pwdump*⁴ или *fgdump*⁵. Следующий шаг – непосредственно попытка взломать полученные зашифрованные пароли. Для этого можно воспользоваться бесплатными программами *Cain & Abel*⁶ (графический интерфейс) и *John The Ripper*⁷ (командная строка) или платными, такими как *Elcomsoft Proactive Password Auditor*⁸ (PPA) и др. Коммерческие продукты, как правило, обладают более широкими функциональными возможностями и более высоким быстродействием по сравнению с бесплатными аналогами. Этап перебора может быть очень ресурсоемким, а его продолжительность – зависеть от доступных аудиторам вычислительных ресурсов и от при-

нятой в организации политики безопасности. Как правило, он занимает от нескольких часов до нескольких суток. После завершения перебора необходимо проанализировать результаты, идентифицировать найденные пароли и сделать вывод о соответствии или несоответствии их принятой политике.

Однако такой обособленный аудит паролей не может дать целостной картины соблюдения или несоблюдения утвержденной политики. Если пользователь использует один и тот же пароль (пусть и удовлетворяющий требованиям политики) для входа в систему и для доступа к своему персональному ящику электронной почты, такому как *mail.ru*, подобное нарушение не может быть обнаружено при «обычном» аудите паролей. Между тем, это очень серьезная проблема – пароли, используемые в Интернете для доступа к почте, форумам и т. п. зачастую передаются в открытом, незашифрованном виде, и узнать их для подготовленного специалиста не составит труда. Другая проблема, не решаемая обособленным аудитом, – проверка пользователей на устойчивость к атакам из области социальной инженерии, при которых злоумышленник получает ценную информацию (в том числе пароли), не прибегая к использованию технических средств, а путем общения и убеждения. Эти и многие другие проблемы безопасности могут быть решены с помощью комплексных проверок безопасности, таких как *penetration testing*. При общем аудите безопасности проверяющий действует в модели, максимально приближенной к реальной атаке, и может использовать большинство «хакерских» приемов (кроме тех, которые способны привести к выводу из строя оборудования и/или нарушению бизнес-процессов).

Залог успеха парольного аудита – в его регулярности. Если политика безопасности требует от пользовате-

ля менять пароль каждые 30 дней, то имеет смысл проводить проверку системы на устойчивость к атакам с той же периодичностью. Столь же важно понимать, что пароль, считавшийся безопасным 5 лет назад, совершенно не обязательно будет таковым в реалиях дня сегодняшнего. Производительность компьютеров постоянно растет, и время (и стоимость), требуемое для взлома знаковой последовательности определенной длины, уменьшается. Увеличение производительности может быть как плавным и прогнозируемым (скажем, развитие процессоров в общем и целом подчиняется закону Мура), так и скачкообразным (как произошло год тому назад, когда для восстановления паролей начали использовать графические процессоры, что разом увеличило производительность в 10–20 раз). Это еще одна, и очень важная, причина, по которой аудит следует проводить регулярно.

Выводы

Итак, если требуется обеспечить надежную парольную аутентификацию пользователей, необходимо, чтобы выполнялись следующие условия:

- программное обеспечение, реализующее парольную аутентификацию, должно быть проверенным и не иметь ошибок, влияющих на безопасность, или, как минимум, эти ошибки должны быть известны;
- в организации должна быть утверждена политика безопасности, в которой четко сформулированы требования к паролям (их стойкости, повторяемости, частоте смены) и которая составлена с учетом анализа программного обеспечения;
- следует проводить регулярный аудит парольной безопасности на соответствие утвержденной политике, и эта политика должна время от времени пересматриваться. ■

⁴ <http://www.foofus.net/fizzgig/pwdump/>

⁵ <http://www.foofus.net/fizzgig/fgdump/>

⁶ <http://www.oxid.it/>

⁷ <http://www.openwall.com/john/>

⁸ <http://www.elcomsoft.ru/ppa.html>