Управление паролями в UNIX.

Контролировать доступ к машине администратор может в первую очередь путем обеспечения выбора пользователями правильных паролей.

Чем больше системных администраторов будет знать о том, как работают эти инструментальные средства, тем лучше они смогут защитить свои системы.

Обсуждается управление паролями в UNIX, в том числе используемые системы шифрования, правила создания и проверки паролей, сокрытие и устаревание паролей. А также о том, как работает система и какими методами злоумышленникам удается эти средства защиты обойти.

Контролировать доступ к машине администратор может в первую очередь путем обеспечения выбора пользователями правильных паролей. Часто это наиболее слабое звено в любой вычислительной среде, где пользователям предоставлено право задавать пароль самим. Как ни удивительно, но во многих системах вся власть над паролями отдается в руки пользователей. При противоположном подходе, когда пользователи никак не влияют на создание паролей, они часто записывают их где попало, в силу чего вся цепочка защиты сразу рассыпается.

1. ХРАНЕНИЕ ПАРОЛЕЙ

Пароли могут храниться в самых разных местах в зависимости от того, какой вариант UNIX используется и как сконфигурирована система. В **Таблице 1** перечислены самые распространенные системы и указано, где в них могут размещаться пароли. «Теневые», или «скрытые», файлы, т. е. файлы паролей, закрытые для чтения или записи, появились в целях решения проблемы доступности зашифрованных паролей. Поскольку они были доступны для всех пользователей, а алгоритм шифрования широко известен, это открывало возможность написания программ эффективного подбора паролей, получивших название «взломщики паролей». По этой причине зашифрованные пароли стали храниться в скрытых файлах для обеспечения жесткого контроля за тем, какие пользователи и какие системные службы могут их видеть.

ОС	Без сокрытия	С сокрытием
HP-UX	/etc/passwd	/tcb/files/auth
Solaris	/etc/passwd	/etc/shadow
Linux	/etc/passwd	/etc/shadow
AIX	/etc/passwd	/etc/security/passwd
Free BSD	/etc/passwd	/etc/master.passwd

2. ФОРМАТ ФАЙЛА /etc/passwd.

Форматы файла паролей во всех реализациях UNIX согласованы, но, как будет показано далее, этого нельзя сказать о скрытых файлах. Каждый пользователь UNIX должен иметь запись в файле паролей. Однако клиент-серверные приложения не всегда требуют, чтобы пользователь имел бюджет уровня UNIX, когда ему не предоставляется доступ на уровне UNIX, поскольку приложение само выполняет аутентификацию пользователей и осуществляет управление ими. Тем не менее некоторые приложения используют методы аутентификации на уровне UNIX. Но это зависит от конкретного приложения, и данная тема выходит за рамки этой статьи.

Каждая запись в файле паролей /etc/passwd имеет формат

userid:password:UID:GID:Comment:Home

directory:shell

и выглядит, к примеру, следующим образом:

chare:x:500:500:Chris Hare:/home/chare:/bin/bash

Поле «идентификатор пользователя» (chare) содержит реальное название бюджета пользователя. Поле пароль (x) будет содержать либо реальный зашифрованный пароль, либо его обозначение (как в данном примере). Наличие последнего свидетельствует об использовании скрытого файла. Поле «номер пользователя» (500) представляет собой уникальный идентификатор, с помощью которого система различает пользователей. Номер группы (500)

позволяет определить, к какой группе принадлежит пользователь и какая ему соответствует запись в файле /etc/group. Комментарии или поле GECOS (Chris Hare) могут содержать любую текстовую информацию, необходимую для идентификации. Как правило, здесь записывается имя пользователя и другие данные, например номер телефона или название отдела. Домашний каталог (/home/chare) указывает, куда помещается пользователь в момент первоначальной регистрации в системе, а исходный командный процессор (/bin/bash) определяет, какой интерпретатор команд применяется.

3. МЕТОДЫ ШИФРОВАНИЯ ПАРОЛЕЙ

3.1. Незашифрованные пароли.

Одно время пароли UNIX хранились в незашифрованном формате, и только администратор и системное программное обеспечение имели доступ к данному файлу. Проблема возникает при редактировании файла /etc/passwd. Поскольку большинство редакторов создают для редактирования временный файл, то во время этого процесса к нему может обратиться кто угодно и узнать пароль для всех бюджетов. В результате пароли стали шифровать с помощью алгоритма однонаправленного шифрования, и в этом случае в системе хранятся зашифрованные значения. Однако надежность защиты полностью определяется выбранным методом шифрования.

При регистрации в системе UNIX программа getty требует ввести имя пользователя и запускает программу входа в систему, а та, в свою очередь, запрашивает пароль, но не декодирует его. Фактически, программа /bin/login шифрует пароль, введенный пользователем, а затем сравнивает полученное значение с тем, которое хранится в /etc/passwd. Если данные совпадают, то пароль был введен правильно.

Самым популярным методом шифрования стало применение улучшенного стандарта на шифрование данных (Data Encryption Standard, DES). Сам стандарт DES представляет собой систему с симметричным ключом, т. е. один и тот же ключ служит и для шифрования, и для дешифровки. Если бы алгоритм не был усовершенствован, то два пользователя с одним и тем же паролем могли получить одно и то же зашифрованное значение. Данная проблема сохраняется и в улучшенной системе, если поле пароля одного из пользователей скопировать в поле пароля

другого пользователя. Однако в случае применения расширенного метода DES, даже если выбран одинаковый пароль, зашифрованные значения будут отличаться. Подробнее об этом рассказывается в следующем разделе.

3.2. ШИФРОВАНИЕ ПАРОЛЕЙ В UNIX С ПОМОЩЬЮ DES

Обращение к методу шифрования паролей в UNIX выполняется с помощью системного вызова crypt(3).

Истинное значение, которое хранится в /etc/passwd, получается путем использования пароля пользователя для шифрования 64-разрядного блока нулей с помощью вызова crypt(3). «Чистый текст» — это пароль пользователя, который является ключом данной операции. Итоговый «закодированный текст» представляет собой зашифрованный пароль. «Чистый текст» (также называемый незакодированным текстом) — исходное незашифрованное сообщение. Закодированный текст — сообщение после шифрования.

Алгоритм crypt(3) базируется на стандарте DES, разработанном в Национальном институте стандартов и технологий (National Institute of Standards and Technology, NIST). В обычном случае для шифрования исходного текста, в DES часто называемого «чистым текстом», применяется 56-разрядный ключ, состоящий, например, из восьми 7-разрядных символов. Этот «чистый текст», как правило, имеет в длину 64 бита. Вот почему UNIX распознает только первые восемь символов пароля, введенного пользователем. Полученный в итоге закодированный текст невозможно расшифровать, не зная значения исходного ключа.

При вызове crypt(3) в UNIX используется модифицированная версия указанного метода, при этом декларируется, что «чистый текст» преобразуется в блок нулей. Новое шифрование закодированного текста с помощью пароля пользователя в качестве ключа еще больше усложняет этот процесс. Так повторяется 25 раз, а затем полученные в итоге 64 разряда разделяются на 11 печатных символов и сохраняются в файле паролей.

Хотя исходные тексты crypt(3) могут предоставить многие производители (их распространение ограничивается территорией Соединенных Штатов), общедоступного метода для преобразования закодированного текста или зашифрованного значения обратно в исходный «чистый текст» не существует.

Роберт Моррис и Кен Томсон, первыми реализовавшие технологию crypt(3) в UNIX, опасались, что с появлением микросхем с поддержкой DES на аппаратном уровне защиту системы UNIX можно будет легко преодолеть. Чтобы ликвидировать эту угрозу, они предложили использовать «крупинку соли» — 12-разрядное число, применяемое для модификации результата работы DES. Это число может принимать значения от 0 до 4095. В итоге каждый возможный пароль можно представить в файле паролей одним из 4096 способов. Разные пользователи на одной и той же машине могут использовать один и тот же пароль, и никто, даже системный администратор, не будет об этом знать.

Когда запускается программа /bin/passwd для создания нового пароля, она выбирает значение «крупинки соли» в зависимости от времени дня, которое затем применяется для модификации пароля пользователя. Чтобы предотвратить маловероятную возможность задания другого значения «крупинки соли» при следующей регистрации пользователя, UNIX хранит эту величину в /etc/passwd. Фактически, данное значение представлено в виде первых двух символов зашифрованного пароля. Такой механизм гарантирует, что пароль может быть зашифрован снова и ему будет найдено соответствие.

Например, значение зашифрованного пароля

W1wEdEKQNtJbA

состоит из «крупинки соли» (W1) и самого 11-символьного зашифрованного пароля (wEdEKQNtJbA). В нашем примере пользователь вводит свой пароль в процессе регистрации, а «крупинка соли», т. е. W1, добавляется в процессе шифрования. После этого пароль сравнивается с зашифрованным значением, которое хранится в соответствующем файле (помните, что этот файл может быть скрытым). Если два значения совпадают, то пользователь получает доступ в систему. При вводе пароля, вопреки распространенному мнению, значение, хранящееся в системе, не декодируется.

3.3. ШИФРОВАНИЕ ПАРОЛЕЙ MD5

Linux стала первой операционной системой на базе UNIX, в которой был реализован Message Digest 5 (MD5) — метод шифрования паролей в процессе регистрации. Он основан на использовании системы подключаемых модулей идентификации (Pluggable Authenti-cation Module, PAM) для усовершенствования или замены процесса шифрования DES, который применяется по умолчанию. И это лишь часть возможностей PAM. Прежде чем рассмотреть алгоритм Message Digest 5, поговорим о том, для чего служит PAM.

Цель всех проектов PAM — удалить компоненты аутентификации из программного обеспечения определения привилегий. Такой подход позволяет системным администраторам выбирать предпочтительный для них модуль аутентификации во время исполнения, а не полагаться на выбор производителя ПО. В контексте нашего обсуждения модулями аутентификации могут быть стандартное шифрование UNIX DES, шифрование MD5, Kerberos или любой другой из подобных методов. Обсуждение самой среды PAM выходит за рамки данной статьи.

PAGE# 1 (10379:L0)

Тип модуля	Флаг управления	Путь	Аргументы
password	sufficient	/lib/security/pam_UNIX.so	nullok use_authtok md5 shadow

Рисунок 1. Файл /etc/pam.d/system.auth.

Файл system.auth в каталоге /etc/pam.d определяет применяемые методы аутентификации. Пример его показан на Рисунке 1. Поле «тип модуля» (Module Type) предназначено для обновления аутентификационных ключей (token), или паролей, связанных с пользователем. Как

правило, для каждого вида аутентификации служит свой модуль PAM. Поле «флаг управления» (Control Flag) указывает, как библиотека PAM будет реагировать в случае успешного или неудачного завершения процедуры аутентификации. Флаг управления «достаточно», как в данном случае, означает, что библиотека PAM подтвердила аутентификацию или обновление пароля.

PAM обеспечивает очень важное отличие. Поскольку данный подход предусматривает применение подключаемых модулей, то для предоставления доступа к компоненту приложение или сама операционная система может использовать несколько уровней аутентификации.

Путь доступа к модулю (module path) указывает местонахождение подключаемых модулей. Для каждого из них указываются свои аргументы (arguments), которые определяют, как данный модуль будет действовать. В приведенном примере нас интересуют аргументы md5 и shadow. Именно они сообщают PAM о необходимости применить пароли MD5 и создать скрытый файл паролей.

Алгоритм MD5 подробно описан в документе IETF RFC 1321. Он был разработан Роном Ривестом из RSA Data Security. Данный документ носит исключительно информационный характер, т. е. не имеет статуса стандарта Internet. Алгоритм Message Digest генерирует отпечаток, или дайджест сообщения, для пароля. На самом деле MD5 применяется в самых разных ситуациях, обычно для цифровых подписей, и базируется на алгоритме Message Digest 4 (MD4). Хотя MD4 работает очень быстро, он довольно уязвим. Несмотря на то что MD5 чуть медленнее, он реализует более надежную функцию шифрования и будет очень быстро работать на современных вычислительных платформах. Пользователи не заметят различия в скорости между методами шифрования паролей DES и MD5.

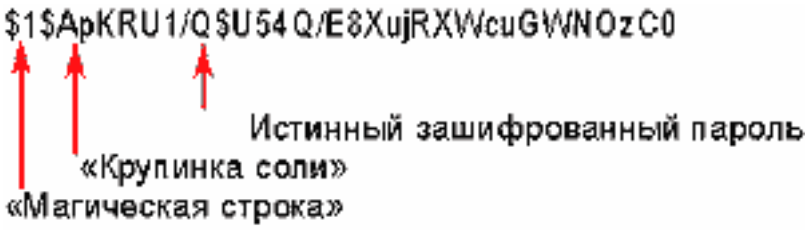

Рисунок 2. Создание паролей MD5 в Linux.

Реализация механизма паролей MD5 работает так же, как и реализация DES, с добавлением «крупинки соли» для модификации процесса шифрования. Пароль в реализации Linux показан на Рисунке 2.

«Магическая строка», применяемая в реализации MD5, идентифицирует пароль как зашифрованный пароль MD5, поэтому модуль PAM знает, какой процесс шифрования использовать при сравнении хранимого значения и введенного пользователем. «Крупинка соли», как и в случае реализации DES, служит для варьирования процесса шифрования MD5. При таком подходе два пользователя, имеющие один и тот же пароль, будут иметь разные зашифрованные значения.

Однако, в отличие от реализации DES, «крупинка соли» в процессе MD5 может состоять из восьми символов. От самого пароля ее отделяет знак «$», тем самым позволяя менять ее длину. Пароль в MD5 может содержать от 13 до 24 символов.

4. УПРАВЛЕНИЕ ПАРОЛЯМИ

Системный администратор, специалист по защите или системный аудитор могут использовать три основных способа контроля: сокрытие, устаревание и качество. В этом разделе мы поговорим о реализации сокрытия и устаревания паролей в самых распространенных системах.

4.1. Устаревание паролей.

Многие версии UNIX предлагают функции устаревания паролей для контроля за тем, когда пользователи должны менять свои пароли. Управление устареванием паролей ведется с помощью значений, добавляемых в файл паролей вслед за зашифрованным паролем. Эти величины определяют минимальный период времени, с момента последнего изменения, прежде чем пользователь сможет его поменять, и максимальный период — до окончания срока действия пароля. На Рисунке 3 дается графическое представление этой концепции.

Рисунок 3. Сроки действия пароля.

Если скрытый файл паролей не используется, то информация, управляющая устареванием пароля, хранится вместе с зашифрованным паролем как последовательность печатных символов. Если имеется скрытый файл паролей, то управление устареванием пароля ведется иначе и будет описано ниже в данной статье.

В UNIX, при традиционном подходе, элементы управления устареванием записываются после

пароля и отделяются от него запятой. Эти символы представляют:

максимальное число недель, в течение которых пароль имеет силу;
минимальное число недель, по прошествии которых пользователь может снова изменить свой пароль;
время последнего изменения пароля.

В **Таблице 2** перечислены представления символов и их значения.

Символ	.	/	0	1	…	9	A	…	Z	a	…	z
Недели	0	1	2	3	…	11	12	…	37	38	…	63

Механизм контроля устаревания распознает два специальных условия: одно заставляет пользователя изменить пароль при следующей регистрации в системе, а второе не разрешает пользователю его изменить.

Чтобы заставить пользователя изменить пароль, как и в случае создания нового пользователя, поле пароля модифицируется — к нему добавляется запятая, после которой указывается два значения, определяющие максимальный и минимальный периоды времени. Эти значения заставляют пользователя изменить пароль при следующей регистрации. После его смены «насильственная» управляющая информация удаляется из записи пароля.

Второй особый случай запрещает пользователю менять пароль. Данное условие реализуется за счет задания максимального значения, меньшего, чем минимальное. В этом случае при следующей попытке регистрации пользователю будет сообщено, что пароль не может быть

изменен.

Некоторые из более защищенных новых версий UNIX используют понятие «срок жизни пароля». Данный параметр определяет количество дней, в течение которых, по окончании периода времени действия пароля, пользователь все еще может изменить просроченный пароль. Как только завершится и этот период, бюджет становится недействительным. Механизм срока жизни пароля не препятствует пользователю изменить пароль, а затем, через какое-то время, вернуть старое значение. Лишь немногие версии системы UNIX отслеживают, какие именно пароли пользователь применял раньше. Конкретный процесс реализации механизма устаревания пароля зависит от версии. Чтобы реализовать его в своей системе, прочитайте системную документацию.

Программа pwexp.pl, представленная на Листинге 1, расширяет понятие истечения срока действия пароля, поэтому пользователи могут подготовиться к тому дню, когда система затребует новый пароль. Заметим, однако, что эта версия программы предназначена для стандартной версии UNIX System V, где не используются файлы скрытых паролей. (Листинги для данной статьи можно найти на Web-сайте журнала SysAdmin по адресу: **http://www.sysadminmag.com**.)

Программа pwexp.pl имеет дело с механизмом поддержки устаревания пароля, который реализован в версиях UNIX вплоть до System V Release 3.2. На этом уровне для усиления системной защиты прибегают к разным вариантам. В итоге было создано несколько различных методов управления скрытыми файлами. Производные BSD и AT&T System V Release 4 опираются на /etc/shadow, в то время как другие системы, в том числе SCO UNIX, разработанная Santa Cruz Operation, и HP-UX компании Hewlett-Packard реализовали Trusted Computing Base.

Следует отметить, что реализация SCO UNIX может использовать /etc/passwd, /etc/shadow или Trusted Computing Base в зависимости от уровня защиты системы. В реализации AIX компании IBM применяется другой механизм для хранения скрытых паролей и информации о возрасте паролей.

Программа, представленная на Листинге 2 — pwexp2.pl, — печатает данные об изменениях и устаревании паролей для систем, где используется файл /etc/shadow. Ее необходимо запускать с привилегиями root, иначе файл /etc/shadow будет недоступен всем остальным пользователям. Результат выполнения команды pwexp2.pl показан на Рисунке 4.

```
#. /pwexp2. pl
Last Password Change
Username        Last Changed on      Can Change      Must Change
root            Wed Mar 28 2001          Now         Never
chare           Wed Mar 28 2001          Now         Mon Sep 24 2001
luciano         Thu Mar 29 2001          Now         Tue Sep 25 2001
sipes           Thu Mar 29 2001          Now         Tue Sep 25 2001
rsivanan        Thu Mar 29 2001          Now         Tue Sep 25 2001
seal            Fri  Mar 30 2001         Now         Wed Sep 26 2001
#
```

Рисунок 4. Результат выполнения команды pwexp2. pl.

4.2. СКРЫТЫЕ ФАЙЛЫ

Как уже отмечалось, существуют три основные версии скрытых файлов. Хотя форматы этих файлов отличаются, они служат одной цели: защитить действующие зашифрованные пароли и обеспечить дополнительный контроль, в том числе поддержку устаревания пароля.

Первый из трех основных форматов, файл /etc/shadow, приведен на Рисунке 4. Для каждого пользователя в этом файле должна присутствовать соответствующая запись /etc/shadow. Сразу выявить зашифрованный пароль невозможно, но пароль, закодированный с помощью DES, состоит, как правило, из 13 символов: 2 символа — это «крупинки соли», а 11 символов — сам пароль. Пример, показанный на Рисунке 5, — пароль, зашифрованный с помощью MD5.

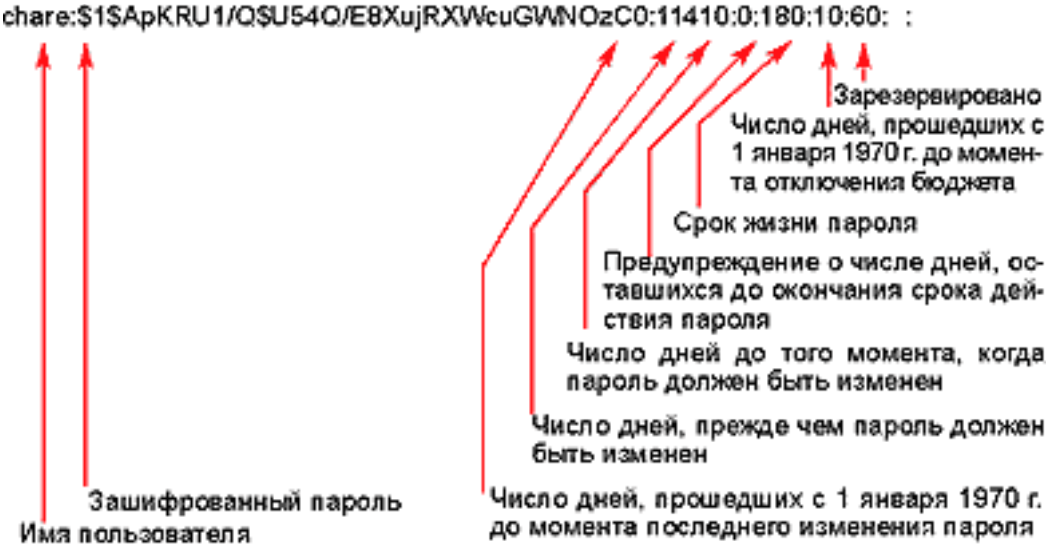

Рисунок 5. Файл /etc/shadow.

Третье поле указывает, сколько дней прошло с момента отсчета времени (1 января 1970 г.) до последней смены пароля пользователем. Четвертое поле — число дней, которые должны пройти до того момента, когда станет возможно сменить пароль. По умолчанию в большинстве инсталляций оно равно нулю, т. е. пользователь может сменить свой пароль в любое время.

Пятое поле определяет, сколько дней пароль останется действующим, прежде чем его придется заменить. Процедуры устаревания пароля добавляют этот период к дате последнего изменения пароля, чтобы определить, нужно ли запрашивать новый. Информация об устаревании пароля

практически бесполезна, если пользователь до истечения срока действия пароля не получает соответствующие уведомления. Шестое поле — число дней до истечения срока действия пароля, когда пользователь получает уведомление о необходимости его изменить.

Седьмое поле — период отсрочки, в течение которой пользователь может по-прежнему регистрироваться в системе, но перед запуском нужной ему оболочки он должен будет сменить пароль. Если за это время пароль изменен не будет, бюджет пользователя блокируется, после чего он не сможет войти в систему. Восьмое поле — число дней, в течение которых бюджет был блокирован, представленное как число дней с 1 января 1970 г. Последнее поле резервное и не используется в существующих реализациях скрытого файла.

4.3. Структура Trusted Computing Base.

Структура /tcb/files была взята из пакетов C2 Security, созданных Secure Ware, и реализована в HP-UX компании Hewlett-Packard и SCO UNIX компании Santa Cruz Operation. Это неотъемлемая часть Trusted Computing Base, поэтому и структура каталога начинается с /tcb. В указанном каталоге находится набор файлов, которые необходимы для реализации функций защиты C2. Большинство из них не относятся к теме данной статьи. Однако скрытые файлы создаются в каталоге /tcb/files/auth.

PAGE# 2 (10142:L0)

В этом каталоге названия большей части подкаталогов совпадают с буквами алфавита, что крайне важно, поскольку каждый пользователь имеет отдельный файл с информацией о нем. Местонахождение пользовательского файла определяется по первой букве имени пользователя. Например, файл bob хранится в /tcb/files/auth/b/bob, как показано на Рисунке 6.

В этом примере каждая запись снабжена комментариями.
Записей, имеющих вид #текст, в самом файле нет.

```
bob:u_name=bob:\          #Действительное имя пользователя
:u_id#1003:\              #Идентификатор пользователя
:u_pwd=MWUNe/9IrPqck:\    #Зашифрованный пароль
:u_type=general:\         #Тип пользователя
:u_succhg#746505937:\     #Последняя успешная попытка изменения пароля
:u_unsucchg#746506114:\   # Последняя неудачная попытка изменения пароля
:u_pswduser=bob:\         #
:u_suclog#747066756:\     #Последняя успешная регистрация
:u_suctty=tty02:\         #Последняя успешная регистрация в tty
:u_unsuclog#747150039:\   #Последняя неудачная попытка регистрации
:u_unsuctty=tty04:\       #Последняя неудачная попытка регистрации в tty
:u_numunsuclog#1:\        #Число неудачных попыток регистрации
:u_lock@:\                #
:chkent:                  #
```

Рисунок 6. Пример файла пользователя из Trusted Computing Base.

Как видно из рисунка, между файлом /etc/shadow и данными, находящимися в Trusted Computing Base, очень много схожего. Записи u_name и u_id обеспечивают соответствие между содержимым файла и конкретным пользователем из /etc/passwd. Здесь хранится зашифрованный пароль, хотя, как правило, используется традиционная форма шифрования DES. Как и /etc/shadow, записи TCB содержат информацию о том, когда пользователь последний раз изменил свой пароль, но, в отличие от /etc/shadow, в нем записывается также, когда была сделана последняя неудачная попытка сменить пароль.

В файле TCB также сохраняется дата последней удачной и последней неудачной регистрации и

указывается, какой терминал при этом был использован. Эти данные необходимы для печати последней регистрационной информации, когда пользователь завершает процесс регистрации. Наконец, файл также содержит число неудачных попыток регистрации, что позволяет отключить бюджет, когда превышено заданное пороговое значение.

Информация о сроках действия паролей сохраняется в другом файле в TCB по мере того, как формируется политика работы с файлами, и применяется ко всем пользователям в системе. Для сравнения файл /etc/shadow позволяет задавать разные значения для каждого пользователя и таким образом повысить уровень детализации при определении защиты бюджетов с бо?льшими привилегиями.

4.4. Скрытые пароли в AIX.

Система AIX компании IBM использует другую реализацию для создания скрытого файла. Скрытые пароли хранятся в /etc/security/passwd. Формат этого файла показан на Рисунке 7.

```
chare:
password =MGURSj.F056Dj
lastupdate =623078865
flags =ADMIN,NOCHECK
```

Рисунок 7. Фрагмент из файла /etc/security/passwd для операционной системы IBM AIX.

Файл /etc/security/password содержит такую же пользовательскую информацию, что и скрытый файл, но разделен на блоки, которые в IBM называют строфами (stanza). Каждая из них представляет собой данные, относящиеся к конкретному пользователю. Однако информация об устаревании пароля хранится в другом файле — /etc/security/user, который мы рассмотрим несколько позже. Пример записи /etc/security/passwd, показанный на Рисунке 7, иллюстрирует доступные для каждого пользователя параметры, описание которых приведено в **Таблице 3**.

Как уже упоминалось выше, в файле /etc/security/passwd отсутствуют данные об устаревании пароля, дате последнего изменения и другая информация, традиционно размещаемая в файле /etc/shadow. Она находится в /etc/security/user (см. Рисунок 7).

```
chare:
   login =true
   rlogin =false
   ttys =/dev/console
   sugroups =security,!staff
   expires =0531010090
   tpath =on
   admin =true
   account_locked =true
   auth1 =SYSTEM,METH2;dhs
```

Рисунок 8. Фрагмент из файла /etc/security/user для операционной системы IBM AIX.

Как и файл /etc/security/password, сведения о пользователе разделены на строфы. В файле содержатся самые разные данные, но для нас наибольший интерес представляют записи, показанные в **Таблице 4**, касающиеся устаревания пароля и элементов управления.

Файл /etc/security/user предлагает самые широкие возможности управления — как устареванием пароля, так и его качеством. Руководители компании и аудиторы должны стремиться обеспечить самый серьезный контроль качества пароля, чтобы не спровоцировать несанкционированный доступ в систему. При этом высокое качество достигается не за счет реализации произвольным образом сгенерированных паролей, а благодаря обучению пользователей и использованию соответствующих системных проверок для их контроля.

5. КАЧЕСТВО ПАРОЛЯ

Чтобы гарантировать надежную защиту паролей, недостаточно только сделать их длиннее шести символов. Всеми силами следует избегать использования в качестве паролей следующих данных:

ваше имя;
имя вашей супруги/супруга;
имя вашего ребенка;
имя вашего домашнего животного;
имена друзей, членов семьи или коллег;
имена персонажей мультфильмов или фантастических героев;
все указанные выше имена с циклически переставленными в них буквами;
название операционной системы, которую вы используете;
названия предметов, видимых с вашего рабочего места;
название улицы, на которой вы живете;
номер паспорта или водительских прав;
дата рождения;
популярные слова, такие, как wizard, gandalf, guru и т. п.;
имя любого пользователя;
любое слово, которое можно найти в словаре;
географическое название;
любое имя собственное;
простые шаблоны, вроде qwerty или abcdefg;
слова и фразы из телепередач или фильмов наподобие NCC-1701;

слова, состоящие из одинаковых букв.

Проверка качества важна для того, чтобы гарантировать надежность пароля как элемента управления. Она должна проводиться при задании пароля. Проверку можно выполнить и после данной операции с помощью таких программ, как crack, но это далеко не так эффективно, как считают многие. Контроль должен базироваться на серии правил, которые следует адаптировать к требованиям конкретной организации. Не все операционные системы или среды приложений предлагают такой уровень функциональности в своей архитектуре. Тем не менее крайне важно использовать подобные функции там, где это возможно. Однако в некоторых ситуациях ценность проведенной проверки почти ничтожна из-за природы самого приложения. Например, если приложение позволяет проверить пароли по словарю, но при этом не позволяет выполнить дополнительную проверку на базе правил, то контроль ценен настолько, насколько надежен словарь.

Причинами реализации правил для пароля во время задания последнего могут быть следующие:

пароли можно взломать с помощью грубой силы;
пароли можно взломать путем поиска в словаре;
пароли могут передаваться в явном виде по незащищенной сети;
пароли часто совместно используются членами одной группы;
пользователи выбирают легко угадываемые пароли.

Поскольку перечисленные ситуации вполне реальны, роль проверки при создании паролей возрастает. Однако не все системы предлагают правила проверки. А в тех системах, где такая возможность присутствует, нужно использовать все варианты проверки, в том числе соответствие

шаблонов и подстановку символов. В среде, где это сделать невозможно, применяются три дополнительных метода: замена самой программы обслуживания паролей, «детективное расследование» и определение политики.

Предпочтительнее всего заменить программу /bin/passwd на одну из тех, которые поддерживают улучшенные правила проверки, поскольку приобретенные преимущества со временем оправдают первоначальные затраты на реализацию. Кроме того, несколько таких программ размещены в Internet, поэтому организации не нужно самой заниматься разработкой.

Определение политики защиты требует, чтобы пользователи были информированы о требованиях выбора высококачественных паролей. Однако пользователи есть пользователи, и не стоит полностью на них полагаться. Следовательно, необходимо с помощью функций поиска таких программ, как crack, находить низкокачественные пароли и добиваться их замены. Кроме того, нужно постоянно требовать от производителей приложения или операционной системы UNIX обеспечить необходимую функциональность.

Осталось только ответить на вопрос: «Что такое пароль хорошего качества?»

Пароль должен удовлетворять следующим критериям:

иметь в длину как минимум восемь символов;
содержать как прописные, так и строчные символы;
включать в себя по крайней мере одно число;
иметь хотя бы один специальный символ;
не базироваться на словарном слове.

Вопросу о качестве пароля посвящено множество статей и дискуссий, но данный обзор посвящен совсем другой теме.

6. ПРОВЕРКА С ПОМОЩЬЮ ВЗЛОМЩИКА ПАРОЛЯ

Взломщик пароля — это программа, которая пытается «подобрать» зашифрованные пароли, хранящиеся в файле /etc/passwd, сравнивая их со словами из словаря. Насколько успешным будет результат работы такой программы — зависит от ресурсов центрального процессора, качества словаря и того, сделал ли пользователь копию /etc/passwd.

С развитием программного обеспечения интерактивного взлома паролей UNIX и роста вычислительной мощности, которую может использовать такое усовершенствованное программное обеспечение взлома, качество пароля становится все важнее. К примеру, средний настольный компьютер (с процессором Pentium младшего класса и операционной системой Windows 95) способен выполнять поиск со скоростью 50 тыс. паролей в секунду.

Рассмотрим следующие примеры.

Пароль из шести символов с прописными и строчными буквами и цифрами можно взломать за 626, или 56 800 235 584, попыток, т. е. на это потребуется 15,78 ч.
Пароль из семи символов с прописными и строчными буквами и цифрами можно взломать за 627, или 3521614606208, попыток, что эквивалентно 41 дню.
Пароль из восьми символов с прописными и строчными буквами и цифрами можно взломать за 628, или 2 183 401 055 849e+14, попыток. Процесс займет приблизительно 7 лет.
Пароль из восьми символов, в котором используются только прописные или только строчные символы, можно взломать за 268, или 208 827 064 576, попыток - около 24 дней.
Пароль из восьми символов (или PIN), в котором использованы только цифры, можно

взломать за 108, или 100 млн попыток, т. е. на это уйдет всего 100 с.

Таким образом, взлом пароля на современных вычислительных платформах займет не так уж много времени.

Если оценить время, которое на это требуется, становится очевидным, почему злоумышленники сначала пытаются заполучить системный файл паролей, а затем обработать его в поисках «простых» паролей. Программу для взлома написать достаточно легко: примерно 60 строк на языке Си или 40 строк на Perl. Если, стремясь гарантировать, что в вашей системе используются высококачественные пароли, вы решите написать подобную программу, имейте в виду, что тем самым можно навлечь на себя беду. Поскольку она может быть украдена и впоследствии использована для получения доступа к другим машинам, то, при должной ее эффективности, вы можете еще больше ослабить защиту своего компьютера.

Одна из наиболее популярных программ взлома паролей UNIX — crack — написана Алексом Моффеттом, в основу которой заложен поиск в словарях слов и выражений на английском (или другом) языке. Эти средства пробуют различные перестановки слов и фраз (в обратном порядке, добавляя в начале и в конце цифру, объединяя вместе два коротких слова, подставляя противоположное по значению и т. д.) параллельно с более простыми эвристиками, такими, как даты, перестановка букв регистрационного имени, почтовые индексы и определенные распространенные фразы. Некоторые инструментальные средства работают очень быстро и выполняют поиск словарных слов, например ferreted, в считанные минуты.

PAGE# 3 (10579:L0)

Хотя взлом пароля может стать способом определения эффективности и качества паролей, выбранных пользователями, лучше все-таки полнее использовать возможности логичной среды защиты UNIX. Помните, что, будучи предоставлены самим себе, пользователи выбирают пароли, которые легко запоминаются, и, как следствие, их можно быстро подобрать, а это ослабляет общий уровень защиты, над созданием и поддержкой которого работает организация.

7. РЕЗЮМЕ

Хотя механизм паролей в UNIX порождает массу споров и страдает от повторяющихся атак, он обеспечивает надежный контроль для предотвращения несанкционированного доступа к системе. Это утверждение верно, если элементы управления регистрацией корректно настроены с целью обеспечения высококачественных паролей и они оцениваются при создании, задаются ограничения на число неудачных попыток доступа, условия блокирования бюджетов и сроки устаревания пароля. Крайне важно гарантировать, чтобы реальные зашифрованные пароли не были видны пользователям, не имеющим прав администраторов, а для ftp необходимо установить соответствующие ограничения, которые не позволяют загружать их всем подряд. Наконец, прежде чем использовать любое программное обеспечение оценки паролей, например решения Oracle, получите разрешение у руководства на его применение.