ГЕНЕРАЦИЯ ПАРОЛЕЙ ПОЛЬЗОВАТЕЛЕЙ В КЛИЕНТ-СЕРВЕРНЫХ СИСТЕМАХ

Ф.Х. Холиярова
Самаркандский филиал Ташкентского университета информационных технологий, доцент
e-mail: fer.xoli.xafi@gmail.com
Дж. Салайдинов
Самаркандский филиал Ташкентского университета информационных технологий, магистрант
e-mail: salaydinovjamshid@gmail.com

Annotatsiya: В статье представлены результаты исследования по разработке модели генератора псевдослучайных чисел для усиления аутентификации в клиент-серверных системах.

Ключевые слова: клиент-серверные системы, аутентификация, одноразовый пароль, генераторы псевдослучайных чисел, секретный ключ, синхропосылка, хэширование.

В современном мире информационные системы различной значимости и охвата стали неотъемлемой частью инфраструктуры государства, бизнеса, и общества в широком смысле этого слова. С каждым днем в информационные системы переносится все больше защищаемой информации. Современные технологии в информационном пространстве обеспечивают новые возможности ведения бизнеса, государственной и общественной деятельности, и вместе с тем создают значительные требования в обеспечении информационной безопасности с целью защиты этой информации. До 70 % из них — случаи несанкционированного получения прав доступа, хищении и передачи конфиденциальной информации пользователей информационных систем. Это становится возможным в связи несовершенством технологий авторизации и аутентификации пользователей информационных систем. Совершенствование методов систем учета доступа и регистрации пользователей является одним из главных приоритетов совершенствования информационных систем. Основными процедурами регистрации пользователей в информационных системах являются процедура идентификации — получение ответа на вопрос «Кто Вы?» и аутентификации

— доказательства того, что «Вы именно тот, кем представляетесь». Несанкционированное получение прав доступа злоумышленником к информационной системе связано, в первую очередь, с нарушением прохождения процедуры аутентификации.

Процесс регистрации пользователя в информационной системе состоит из трех взаимосвязанных, выполняемых последовательно процедур: идентификации, аутентификации и авторизации. Идентификация — это процедура распознавания субъекта по его идентификатору. В процессе регистрации 5 субъект предъявляет системе свой идентификатор и она проверяет его наличие в своей базе данных. Субъекты с известными системе идентификаторами считаются легальными, остальные субъекты относятся к нелегальным. Аутентификация (установление подлинности) — процедура проверки принадлежности субъекту доступа предъявленного им идентификатора и подтверждение его подлинности. Другими словами, аутентификация заключается в проверке: является ли подключающийся субъект тем, за кого он себя выдает. Авторизация — процедура предоставления определённому лицу или группе лиц прав на выполнение определённых действий; после прохождения им процедуры аутентификации. Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции — это значит, что он имеет на неё право.

На сегодняшний момент аутентификация с помощью логина и пароля является наиболее распространенной, прежде всего, благодаря простоте использования. Однако данный вид аутентификации имеет некоторые недостатки. В отличие от случайно формируемых криптографических ключей, пароли пользователя бывает возможно подобрать из-за безответственного отношения подавляющего большинства пользователей к генерированию пароля. Выбор пользователями легко угадываемых паролей встречается чаще всего. Существуют и свободно доступны различные утилиты подбора паролей, в том числе, специализированные для конкретных широко распространенных программных средств. Пароль может быть подсмотрен или перехвачен при вводе. Начинающие пользователи персональных компьютеров используют один и тот же пароль для большинства своих аккаунтов в сети, для защиты учетной записи. Это одна из самых распространённых ошибок, так как злоумышленники могут скомпрометировать его и, разумеется, воспользоваться им. В связи с этим, пользователи должны задумываться о том, чтобы всегда устанавливать (или хотя бы стараться) разные пароли на учетные

записи, аккаунты в Интернете. Кроме того, одна из теорий, подтвержденных на практике уже не один раз гласит, что пароль периодически нужно менять. 6 Периодическая смена паролей позволит более эффективно защитить аккаунт и конфиденциальную информацию на нем, а кроме того, это усложнит процесс взлома злоумышленникам, так как старая информация, которая возможно, была ими найдена уже будет неактуальной.

«Клиент-сервер» — современная технология, которая обеспечивает сетевое взаимодействие «запрос» — «ответ» путем распределения нагрузки и заданий между теми сетевыми ресурсами, которые предоставляют услуги («серверы») и теми, которые их используют («клиенты»). Речь идет о программном обеспечении, размещенном на одной или нескольких вычислительных машинах. Любой веб-сайт, или приложение в Интернет работает на сервере, а его пользователи являются клиентами. Социальные сети (Фейсбук, ВК и пр.), сайты электронной коммерции (Amazon, Озон и др.), мобильные приложения (Instagram и т. д.), устройства Интернета вещей (умные колонки или смарт-часы) работают на основе клиент-серверной архитектуры. Практически любая корпоративная сеть или ИТ-система предприятия, как правило, строится по архитектуре «клиент-сервер». В настоящее время парольная аутентификация является наиболее распространенной, благодаря своему единственному достоинству — простоте использования. Однако, часто встречаются легко предугадываемые пароли по причине небрежности пользователей при их формировании, из-за чего пароли можно легко подобрать. После получения злоумышленником многоразового пароля субъекта, он имеет постоянный доступ к взломанным конфиденциальным сведениям. Эта проблема решается применением одноразовых паролей, которые действительны только для одного входа в систему и при каждом следующем запросе доступа будет необходим новый пароль. Аутентификация с одноразовым паролем обладает устойчивостью к атаке анализа сетевых пакетов, что дает ей значительное преимущество перед запоминаемыми паролями. Технологии использования одноразовых паролей можно разделить на следующие: Использование генератора псевдослучайных чисел, единого для субъекта и системы. В данном случае используется генератор псевдослучайных чисел с одинаковым значением для субъекта и для системы. Использование временных меток вместе с системой единого времени. Аутентификация основана на генерации случайных чисел через определенные временные интервалы. Использование базы случайных паролей,

единой для субъекта и для системы. Основан на единой базе паролей для субъекта и системы и высокоточной синхронизации между ними, при этом каждый пароль из набора может быть использован только один раз. Можно сделать вывод, что по сравнению с использованием многоразовых паролей одноразовые пароли предоставляют более высокую степень защиты. [1].

Настоящий стандарт определяет криптографические алгоритмы генерации псевдослучайных чисел, для которых характерно использование секретного ключа и уникальной синхропосылки, в результате чего генерируемые числа трудно предугадать или повторить. Ключ алгоритма генерации может быть известен только одной стороне, тогда псевдослучайные числа можно использовать для построения секретных параметров владельца ключа. Если ключ алгоритма генерации будет известен нескольким сторонам, то стороны могут использовать алгоритм для построения общих секретных параметров. Стандартом предусмотрено использование трех типов криптографических алгоритмов генерации псевдослучайных чисел [2]: — алгоритм выработки имитовставки в режиме HMAC (Hash-based Message Authentication Code), — алгоритм генерации псевдослучайных чисел в режиме счетчика, — алгоритм генерации псевдослучайных чисел в режиме HMAC Как уже было установлено, одноразовые пароли предназначены для усиления аутентификации в клиент-серверных системах: кроме обычного долговременного (статического) пароля клиент предъявляет серверу дополнительный пароль, срок действия которого ограничен определенным сеансом аутентификации или промежутком времени. Даже если противник узнает пароль текущего сеанса или промежутка, он не сможет использовать его в следующем. Аутентификация может быть двусторонней: после успешной аутентификации клиента сервер генерирует новый одноразовый пароль и предъявляет его клиенту. Стороны генерируют одноразовый пароль R , комбинируя общий секретный ключ K с уникальной синхропосылкой. Ключ K должен вырабатываться без возможности предсказания, распространяться с соблюдением мер конфиденциальности и храниться в секрете. Ключом является двоичное слово фиксированной или произвольной длины. Была разработана схема аутентификации клиент-серверной системы, которая представлена на рис. 1.

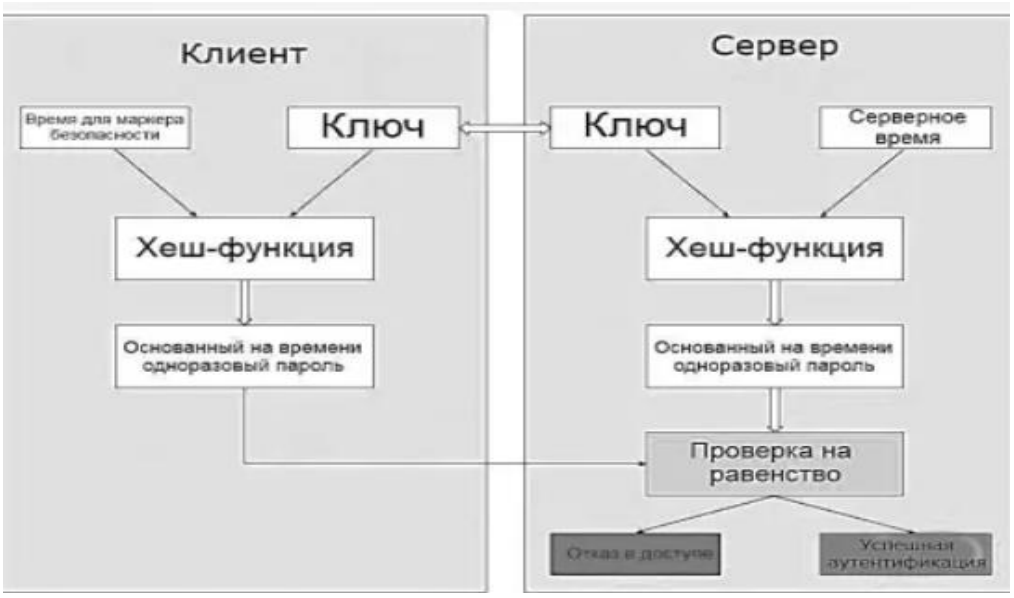

Рис. 1. Схема аутентификации клиент-серверной системы

В зависимости от способа формирования синхропосылки стандартом определены три режима (механизма) генерации паролей: HOTP (HMAC-based One-Time Password), TOTP (Time –based One-Time Password) и OCRA (OATH Challenge-Resp onse Algorithms). В данном исследовании был выбран режим TOTP, в котором синхропосылка представляет собой округленную отметку текущего времени [3]. Входными данными алгоритма генерации одноразовых паролей в режиме TOTP являются: — количество d ∈ {6,7,8} цифр в пароле; — секретный ключ K ∈ {0,1}8*; — округленная отметка T текущего времени — неотрицательное целое число. Выходными данными является одноразовый пароль R ∈{0, 1,..., 10d −1}. Алгоритм генерации пароля в режиме TOTP состоит в выполнении следующих шагов: W ←⟨ T ⟩64. Y ←hmac [*h*](K , W 7 ‖ W 6 ‖... ‖ W 0). R ←otp-dt (d, Y). Возвратить R . На основании вышесказанного была разработана программа «PRNG.TOTP», алгоритм которой представлен на рис. 2.

Рис. 2. Алгоритм программы «PRNG.TOTP»

Программа была написана на языке программирования Python, который является современным, многофункциональным и универсальным, широко используемым языком программирования с низкой времязатратностью на написание скрипта, простым и логичным синтаксисом. Ниже представлен скрипт разработанной программы «PRNG.TOTP»:

```
import math
 a = 6
 from datetime import datetime
current_datetime = datetime.now()
vremechko = current_datetime.year + current_datetime.month +
import random
p = 0
for i in range(3):
p = p + random.randint(-1000000, 1000000)
c = hash(hash(vremechko) + math.fabs(hash(math.fabs(p))))
while c > 999999:
```

```
c = c // 10 + random.randint(0,10)
print(int(c))
```

Вызов и загрузка программы осуществляется после загрузки общего программного обеспечения автоматически. После определения текущего времени, осуществляется генерация секретного ключа (посредством импорта библиотеки функций генератора псевдослучайных чисел) и хэширование его с текущим временем. В нашем случае секретный ключ составляет 6 знаков в длину.

Литература:

1. А.В. Васильков, И.А. Васильков. Безопасность и управление доступом в ИС. Учеб. пособие. М.:ФОРУМ:ИНФРА – МБ, 2019.- 368 с.
2. Islomov S. Z. et al. New authentication scheme for cloud computing //Journal of Advanced Research in Dynamical and Control Systems. – 2018. – Т. 10. – №. 10. – С. 2316-2319.
3. Djumayev S. N. et al. WEB SAYTLARNI HIMOYALASHDA PAROLLARNI GENERATSIYA QILUVCHI TIZIM //PEDAGOGS jurnali. – 2022. – Т. 23. – №. 1. – С. 144-149.
4. Djumayev S. N., Narzullayeva N. U., Umarova M. F. ISH YURITISH HUJJATLARINI TAYYORLASHNI AVTOMATLASHTIRUVCHI AXBOROT TIZIMINI YARATISH BOSQICHLARI //PEDAGOGS jurnali. – 2022. – Т. 23. – №. 1. – С. 138-143.
5. M’Raihi D., Machani S., Pei M., Rudel l J. TOTP: Time-Based One-Time Password Algorithm. Request for Comments: 6238, 2011