УДК 004.056

Ю. В. Баришев, к. т. н.; М. М. Чайкін; О. В. Кохан

МЕТОД ТА ЗАСІБ ПІДВИЩЕННЯ СТІЙКОСТІ ЗРОЗУМІЛИХ КОРИСТУВАЧАМ ТЕКСТОВИХ ПАРОЛІВ

Наведено результати аналізу особливостей процесу автентифікації користувачів, які дозволили обґрунтувати доцільність використання паролів як факторів автентифікації. Розглянуто задачу генерування стійкого паролю користувачами, які можуть не мати для цього достатнього рівня компетентності в галузі кібербезпеки. Проаналізовано відомі підходи до генерування паролів, які можуть бути застосовними для розв'язання цієї задачі, зосереджуючи увагу на методах, що дозволяють генерування паролів, які зрозумілі користувачам, а відтак є простішими для запам'ятовування. Як спільний недолік відомих методів, було визначено використання навчальних вибірок або словників, які підбираються залежно від користувачів, що негативно впливає на масштабованість застосування методів. Запропоновано метод збільшення стійкості паролів, який базується на використанні слів, словотворів або словосполучень, які вводять користувачі, та їх подальшої модифікації. Запропоновані модифікації ґрунтуються на таблиці замін символів, яку пропонується формувати перед застосуванням методу. Зокрема, у цьому дослідженні було здійснено формування цієї таблиці на основі опитування користувачів. Описано спосіб, в якому було враховано отримані результати опитування під час реалізації замін символів. Для покращення гнучкості застосування методу було передбачено параметр керування — ймовірність модифікації певного символу з рядка, запропонованого користувачем як пароль. Наведено алгоритм, який дозволяє реалізувати запропонований метод. З метою обґрунтування здійсненності було описано розроблений платформонезалежний програмний засіб, який реалізує запропонований метод. Наведено результати тестування засобу. На основі аналізу наведених результатів продемонстровано особливості методу та вплив таблиці замін на вихідну множину потенційних паролів, отриманих внаслідок роботи застосунку. З проведеного дослідження було зроблено висновки та окреслено перспективи подальшого розвитку цього дослідження.

Ключові слова: *пароль, стійкість, автентифікація, зручний для користувачів, таблиця замін.*

Вступ

В інформаційних системах виникає необхідність наявності високого рівня захисту для забезпечення конфіденційності, цілісності та доступності інформації, яка зберігається, передається або обробляється в цих системах. Для цього використовується автентифікація – процедура встановлення факту підтвердження належності користувачеві пред'явленого ним ідентифікатора [1]. Об'єкти, на основі пред'явлення яких відбувається встановлення цього факту, називають факторами автентифікації [1, 2]. В загальному випадку як фактор автентифікації можливо використовувати будь-яку доступну лише користувачеві інформацію, яка повинна унікально його характеризувати.

Попри, на перший погляд, різноманіття у виборі шляхів автентифікації користувача застосування більшості потенційних факторів автентифікації для окремої практичної задачі суттєво обмежуються вимогами до зручності користувача або практичної можливості їх інтеграції до інформаційних систем, які вже розроблені і водночас не враховували вимоги кібербезпеки достатньою мірою на етапі свого проєктування [2 – 4]. Саме тому, з практичних міркувань [1, 5] найчастіше використовують саме паролі як фактор автентифікації. Однак спільним недоліком для автентифікації на основі паролів є необхідність розв'язання задачі пошуку компромісу поміж стійкістю пароля та складністю його запам'ятовування [1, 5, 6]. Паролі, згенеровані на основі випадкових або псевдовипадкових послідовностей [7], дозволяють забезпечити високий рівень стійкості, однак складність їх запам'ятовування спричиняє проблеми пов'язані з необхідністю користувачам створювати "резервні копії" паролів і невідповідними умовами їх подальшого зберігання [8] через те, що автентифікацію користувачеві протягом робочого дня необхідно проходити багато разів [4, 9]. Відповідно,

такий спосіб генерування породжує низку проблем, пов'язаних із людським фактором, які можна розв'язувати лише організаційними заходами, які внесуть ще більше впливу людського фактору. Саме тому, паролі повинні бути зрозумілими користувачам, і як наслідок, зручними для запам'ятовування.

Якщо розглядати використання паролів, які несуть змістове навантаження для користувачів, то внаслідок принципу мінімізації осіб, ознайомлених з конфіденційною інформацією [6], розв'язання цієї задачі покладається на користувачів, які, в більшості випадків, не мають достатньої компетентності або бажання для її розв'язання [4, 9], а не на офіцера чи відповідну службу з кібербезпеки підприємства або установи, наприклад, СУІБ – службу управління інформаційною безпекою в банках [10], які є фахівцями у цій галузі. Таким чином виникає суперечність, з одного боку паролі повинні мати високий рівень стійкості, з іншого – цю стійкість повинні забезпечувати персони, які не мають для цього достатнього рівня компетентності в галузі кібербезпеки [3, 4]. Саме тому, актуально розробити інструмент, який без додаткового розголошення самого пароля дозволить користувачам підвищити його стійкість, залишаючи пароль зрозумілим, а відтак – легшим для запам'ятовування.

Метою цього дослідження є підвищення стійкості паролів, отриманих на основі слів та словосполучень з природної мови людей.

Для досягнення мети необхідно розв'язати такі задачі:

– проаналізувати відомі методи автентифікації користувачів;

– розробити метод підвищення стійкості паролів;

– реалізувати метод у вигляді засобу.

Аналіз підходів до автентифікації користувачів

У програмних застосунках, операційних системах, базах даних тощо виникає необхідність наявності високого рівня захисту для забезпечення конфіденційності інформації. Для цього використовується розподіл доступу користувачів до інформаційних ресурсів. Ключовим елементом для реалізації цього розподілу доступу є автентифікація користувачів [1]. Внаслідок виконання процедури автентифікації відбувається підтвердження належності користувачеві пред'явленого ним ідентифікатора. Для підтвердження можна використовувати будь-яку іншу секретну інформацію, яка повинна бути доступна лише користувачеві, або унікально його характеризувати. Відповідно виділяють такі типи факторів автентифікації користувачів [1, 3, 6]:

– знання чого-небудь;

– володіння чим-небудь;

– на основі біометричних характеристик;

– на основі місця розташування.

Найбільш популярним підходом до автентифікації є використання текстових паролів як факторів автентифікації [1, 2, 5, 6], яке належить до категорії факторів автентифікації користувачів на основі знання чого-небудь. Поширеність використання паролів обумовлена низкою прагматичних міркувань [1, 9]:

– використання паролів не потребує вкладання ресурсів у додаткове апаратне забезпечення інформаційних систем, яке необхідне при використанні інших груп факторів автентифікації користувачів, чим водночас покращує масштабованість інформаційної системи, в якій відбувається автентифікація;

– низька складність додаткового програмного забезпечення, що обумовлює низьку вартість розробки відповідних програмних модулів порівняно з відповідними модулями для інших видів автентифікації;

– наявність найкращих практик розробки модулів парольної автентифікації, оскільки це найбільш поширений та найбільш досліджений метод автентифікації.

При такому підході, коли користувача $user_i$ із асоційованим до нього паролем $password_i$ реєструють в системі, пароль гешують та зберігають отримане геш-значення на стороні, яка здійснює автентифікацію [5, 11], що захищає від зловмисних дій з боку адміністраторів. Таким чином, в інформаційній системі в базі даних користувачів *UserDB* повинна зберігатись інформація, яка є відображенням (mapping) ідентифікаторів зареєстрованих користувачів та загешованих паролів, які ним відповідають:

$$UserDB = Map\big(User, hash(Password)\big)$$

Оскільки до *UserDB* є доступною для адміністраторів інформаційної системи, тому існує загроза витоку загешованих паролів. Гешування стає на заваді швидкому зламу паролів, однак за наявності відповідних ресурсів зловмисники можуть побудувати прообраз пароля користувача. Відповідно для керування ризиками виникає необхідність періодичної зміни паролів. Як показують дослідження [9], навіть, ті з користувачів, які обізнані з базовими поняттями кібергігієни, часто нехтують останньою заради власної зручності. При цьому через згаданий вище принцип мінімізації осіб, ознайомлених з конфіденційною інформацією [6], виконати контроль адекватності нового паролю є проблематичним.

Відомий підхід до генерування паролів на основі часткових відомостей про нього з метою відновлення доступу користувачів до облікових записів [12]. Цей підхід ґрунтується на основі методів штучного інтелекту. Попри те, що початково він не був орієнтований саме на генерування нових паролів, він також може бути застосовним для досягнення мети цього дослідження, хоча і з певними недоліками, пов'язаними з іншим початковим призначенням. Напрям із використання підходів штучного інтелекту виглядає перспективним при використанні навчальних вибірок, які враховуватимуть особливості та персональний досвід користувача або певної групи користувачів. Водночас підготовка навчальної вибірки таким чином, щоб згенеровані паролі були зрозумілими і легкими для запам'ятовування широкому колу користувачів, зменшуватиме кількість паролів, які можуть викликати певні асоціації у кожного з користувачів.

В роботі [13] пропонуються підходи, які базуються на замінах, перестановках та словниках зі словами, що можуть викликати певні асоціації у користувачів. Залежно від схеми отримані користувачами паролі можуть бути як простими для запам'ятовування (наприклад, слово "AMAZON" перетворюється у пароль "amzon"), так і складними (наприклад, слово "AMAZON" перетворюється у пароль "NHTFHT"). Запропоновані схеми [13] можуть стати в нагоді, зокрема, працівникам СУІБ. Однак в межах задачі цього дослідження, яке спрямоване на користувачів, які не мають спеціальної освіти в галузі кібербезпеки, такі підходи не стануть в нагоді. Крім того, використання сталих словників є вразливим місцем при такому підході.

Таким чином, попри доцільність використання паролів як факторів автентифікації користувачів, їх застосування ускладнене відсутністю методів генерування стійких паролів, орієнтованих на користувачів без спеціалізованої підготовки.

Метод підвищення стійкості паролів

Аналіз показав, що метод парольної авторизації є зручним та широко використовується в системах перевірки прав користувача. Однак такий метод передбачає наявність певних характеристик секретного слова користувача. Однією з головних таких характеристик є стійкість пароля та його зручність у використанні. Наявність спеціальних символів, літер різного регістру, цифр покращує стійкість паролів та водночас для звичайного користувача стає важчим запам'ятовування таких ключових слів. З іншого боку, використання простих, передбачуваних слів та символьних комбінацій негативно позначиться на стійкості парольної фрази. Тому пропонується метод генерування паролів, що дозволить поєднати ці дві властивості та дасть змогу користувачам на основі слова, яке б вони воліли

використовувати як пароль, отримувати варіанти більш стійких паролів.

Запропонований метод передбачає таку попередню підготовку, а саме формування таблиці замін літер на символи. У таблиці 1 наведено фрагмент такої таблиці, яка використовувалась під час практичної реалізації в межах цього дослідження.

Таблиця 1

Приклад таблиці заміни символів

Літера	Символи для заміни
A	@; 4; (L; /\
B	8; I3; |3;
C	<; (
H	#; /-/;]-[; }{
...	...

Після того, як таблиця замін була сформована з урахуванням особливостей мови, природної для користувачів, автори пропонують провести опитування користувачів щодо їх більш бажаного для них варіанту заміни. Зокрема у цьому дослідженні було сформовано анкету для онлайн-опитування, фрагмент якої наведено на рис. 1.

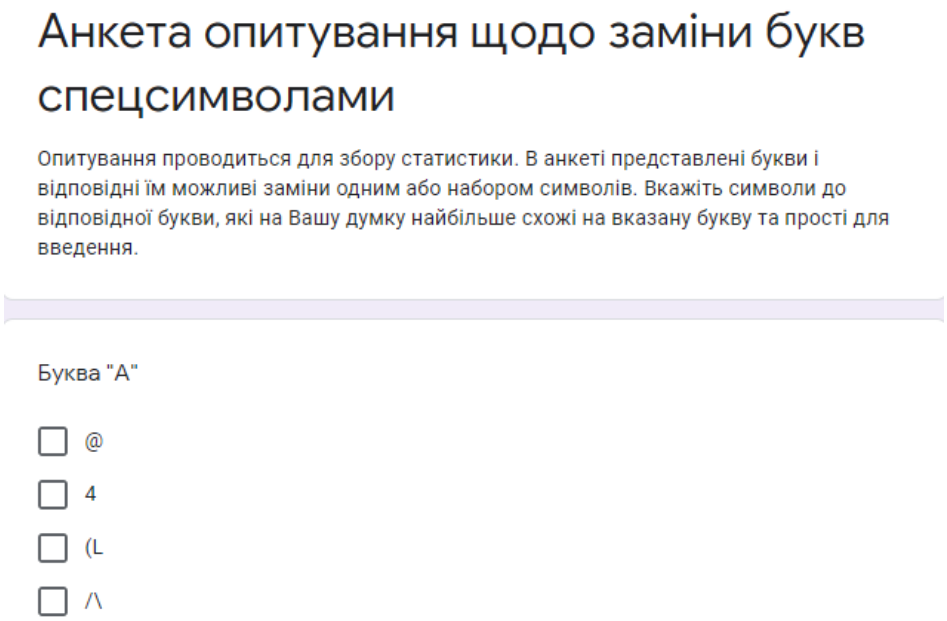

Рис. 1. Фрагмент анкети опитування

Внаслідок опитування отримуються результати бажаної заміни для користувачів. Зокрема на рис. 2 наведено результати, отримані в межах цього дослідження при відповіді на фрагмент анкети, наведений на рис. 1.

Як результат агрегації результатів опитування визначається ймовірність заміни кожної з літер, наявних у таблиці заміни, у випадку, якщо в процесі виконання методу виникне необхідність у такій заміні. Крім того, залежно від особливостей інформаційних систем, в яких передбачається автентифікація, перед виконанням методу необхідно визначити значення ймовірності заміни символу n. Варто зазначити, що у випадку, коли значення цієї ймовірності становитиме 0, то запропонований метод виродиться в класичний підхід, коли пароль обирає користувач.

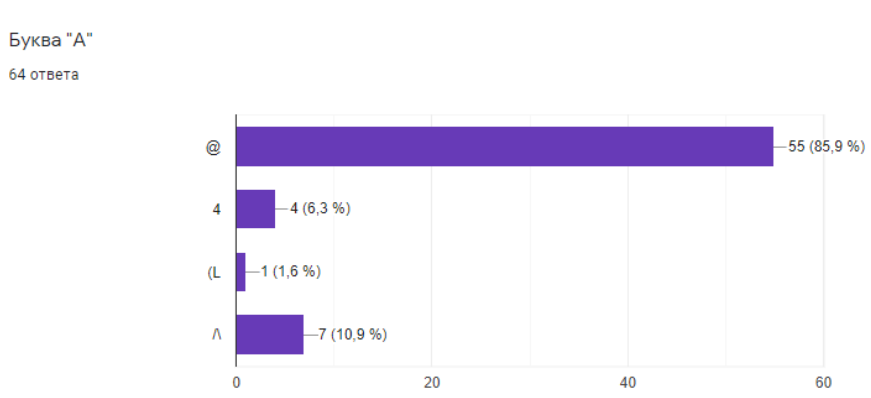

Рис. 2. Результати проведеного опитування

Після завершення попередньої підготовки передбачається виконання методу, який має такі кроки:

Крок 1. Користувач вводить слово або словосполучення, яке є бажаною для нього основою формування паролю.

Крок 2. Генерується псевдовипадкове число *rand_n*, яке масштабується відповідно до значень ймовірності *n* та відбувається порівняння. Якщо *rand_n* є меншим за *n*, то відбуватиметься заміна символу – крок 3, інакше виконується крок 4.

Крок 3. Генерується ще одне псевдовипадкове число *rand_s*, яке масштабується відповідно до значень ймовірності. Залежно від значення *rand_s* за принципом рулетки ("ширина" сектору залежить від результатів агрегації відповідей користувачів) визначається сценарій заміни.

Крок 4. Якщо кроки 2 та 3 були виконані для кожного символу, введеного користувачем, то виконання алгоритму завершується, інакше відбувається перехід до наступного символу і перехід до кроку 2.

З метою генерування випадкових чисел передбачається використання спеціалізованих співпроцесорів [14]. Як альтернатива генераторам випадкових чисел, у випадку неможливості їх використання або недовіри до них [15], доцільним є використання криптографічно стійких генераторів псевдовипадкових чисел [15, 16].

Для формалізації методу один з алгоритмів, що реалізує його, наведено на рис. 3.

Оскільки ці дослідження були спрямовані на користувачів, які можуть не мати достатньої компетентності для реалізації методу, на основі розробленого алгоритму (рис. 3), який реалізує запропонований метод, авторами було розроблено засіб, який дозволятиме використовувати запропонований метод на практиці.

Засіб підвищення стійкості паролів

З метою реалізації засобу підвищення стійкості паролів було обрано мову програмування Java через можливість використовувати застосунку на різних пристроях з різними операційними системами, що забезпечує Java, покращить масштабованість використання застосунку. На вхід засобу подаються дані, що вводить користувач, а саме ключове слово, словотвір або словосполучення, на основі якого буде генеруватися пароль. У випадку занадто короткого слова (менше 8 символів), користувачу буде виводитись попередження про недостатню довжину. Крім того, для адаптації під різні задачі в інтерфейсі передбачено поле для введення ймовірності заміни символів у парольній фразі (рис. 4).

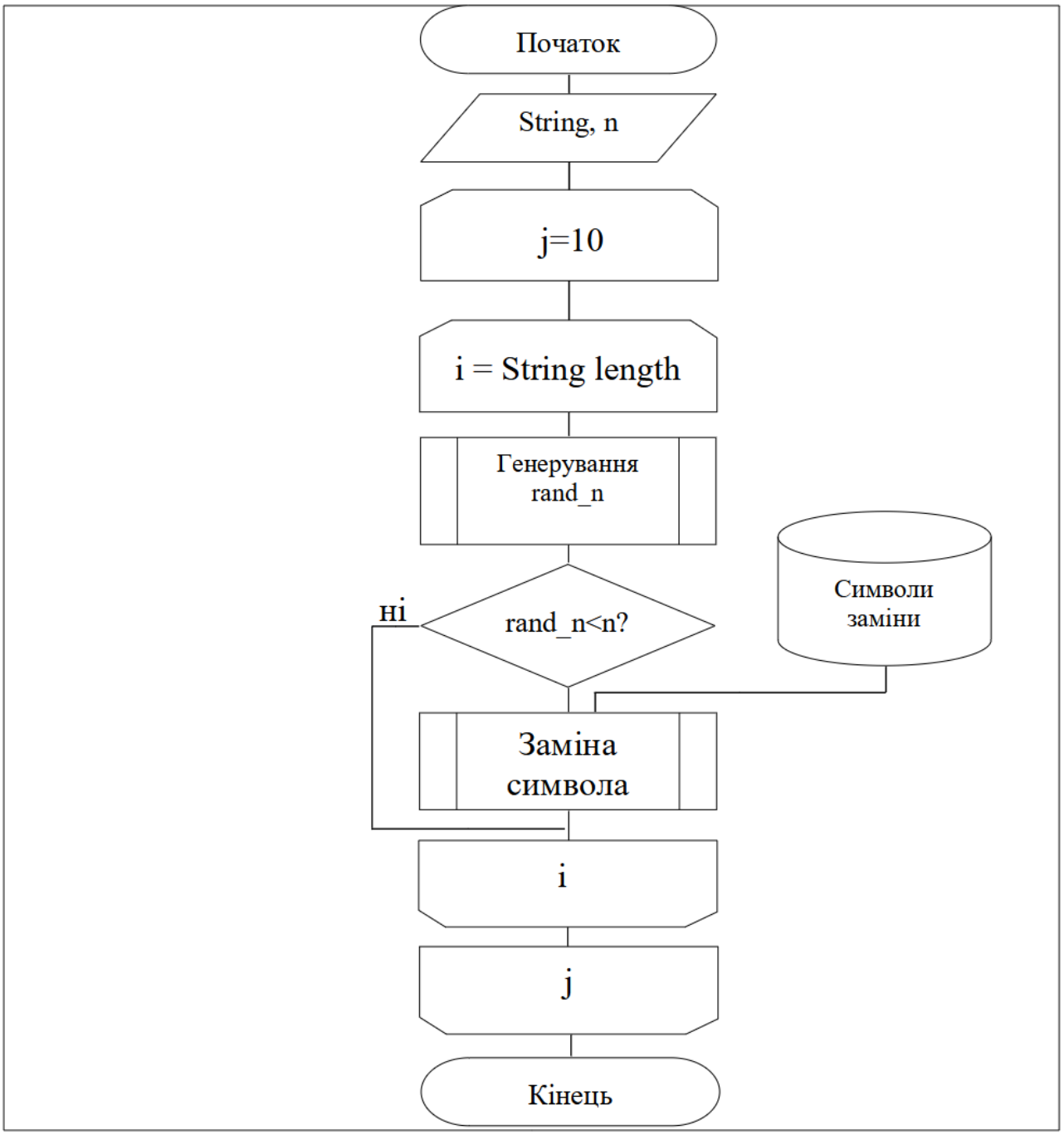

Рис. 3. Алгоритм підвищення стійкості паролів

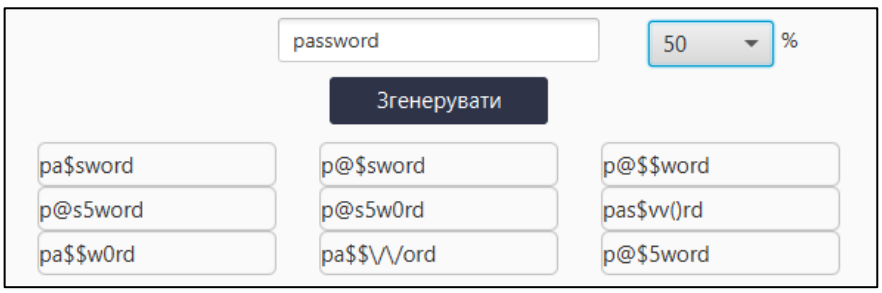

Рис. 4. Результати генерування паролів за допомогою засобу

Заміна кожного символу буде відбуватися у циклі, кількість повторів якого залежить від

довжини введеного слова, де визначатиметься чи буде замінена літера на символ з таблиці замін.

Процес заміни побудований на генераторі псевдовипадкових чисел, межі генерування якого в подальшому масштабуються у проміжку від 0 до 100 за допомогою методу генерування псевдовипадкових чисел, описаному в роботі [16]. У випадку, коли користувач введе частку заміни символів (у наведеному прикладі на рис. 4 це значення встановлене на рівні 50%) і згенероване число буде з проміжку від 0 до 50, відповідна літера у слові буде замінюватися на спеціальний символ.

Як видно з рис. 4, засіб пропонує користувачу низку альтернативних паролів, які через застосування спеціальних символів та цифр на додачу до літер збільшують множину паролів, які необхідно перебирати зловмисникові під час атак на пароль. Саме завдяки цій властивості відбувається збільшення стійкості отриманих паролів порівняно з введеним словом “password”. Крім того, з рис. 4 видно як працює таблиця замін. Зокрема літера “a” внаслідок абсолютної переваги варіанту заміни “@” замінилась під час виконання алгоритму лише таким чином (хоча мала шанс і на інші варіанти заміни), в той час, як літера “o” у наведених результатах у деяких випадках була замінена на “0”, а в одному була замінена на “()”. Літери “p”, “r”, “w” та “d” не були змінені, оскільки були відсутніми у таблиці замін, сформованій в межах цього дослідження.

Таким чином, для визначення рівня збільшення стійкості пароля доцільно проаналізувати отриманий результат. Слово “password”, яке складається з 8 літер англійської абетки, які написані у нижньому регістрі, у випадку, коли зловмисник знає довжину пароля, реалізація атаки грубої сили вимагатиме перебору $(26)^8 \approx 2 \cdot 10^{11}$ комбінацій літер. Розглянемо один з найгірших для запропонованого методу випадків, коли зловмисник володіє інформацією щодо можливих замін символів, які використовуються у запропонованому методі, та, як і раніше, знає кількість літер оригінального слова. Після формування комбінації з літер зловмиснику необхідно застосувати всі можливі варіанти з таблиці замін. В конкретній реалізації засобу було використано таблицю замін для 12 літер, кожна з яких має в середньому 2,83 варіанти замін (у випадку, коли було прийнято рішення щодо заміни). Таким чином, потужність множини комбінацій, які необхідно перебрати зловмиснику становить $\left(14 + 12 \cdot (1 + 2{,}83)\right)^8 \approx 1{,}68 \cdot 10^{14}$ можливих варіантів паролів. Отже для типу атаки, що розглядається, кількість комбінацій, які необхідно перебрати зловмиснику, зросла у 840 раз внаслідок застосування запропонованого методу.

Висновки

Внаслідок проведеного аналізу показано актуальну задачу генерування стійких паролів, що повинні розв’язувати користувачі, які можуть не мати для цього достатньої компетентності в галузі кібербезпеки. Для розв’язання цієї задачі розроблено засіб, який ґрунтується на запропонованому методі підвищення стійкості паролів. Завдяки розробленому засобу користувачі матимуть змогу підвищити стійкість паролів, які вони обирають.

В подальшому планується удосконалити запропоновані метод та засіб шляхом надання можливості користувачеві впливати на довжину пароля, отриманого після обробки, а також визначення аналітичних формул для обчислення рівня підвищення стійкості паролів залежно від параметрів модифікації, які задає користувач.

СПИСОК ЛІТЕРАТУРИ

1. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. Учебное пособие для вузов / [Афанасьев А. А., Веденьев Л. Т., Воронцов А. А. и др.] ; под ред. А. А. Шелупанова, С. Л. Груздева. – Москва, Горячая линия-Телеком, 2012. – 550 с.

2. Баришев Ю. В. Метод автентифікації віддалених користувачів для мережевих сервісів. / Ю. В. Баришев,

В. А. Каплун // Інформаційні технології та комп'ютерна інженерія. – 2014. – № 2. – С. 13 – 17.

3. Dasgupta, Dipankar. Advances in User Authentication [Електронний ресурс] / Dasgupta, Dipankar, Roy, Arunava, Nag Abhijit // Springer. – 2017. – 360 p. – Режим доступу : https://www.researchgate.net/publication/334559194_Advances_in_User_Authentication.

4. Evaluation of user authentication methods in the gadget-free world [Електронний ресурс] / Halunen Kimmo, Häikiö Juha, Vallivaara Visa // Pervasive and Mobile Computing. – 2017. – DOI : 10.1016/j.pmcj.2017.06.017. – Режим доступу : https://www.researchgate.net/publication/318241904_Evaluation_of_user_authentication_methods_in_the_gadget-free_world.

5. Multilayer Access for Database Protection [Електронний ресурс] / Olesia Voitovych, Leonid Kupershtein, Vitalii Lukichov, Ivan Mikityuk // International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). – 2018. – P. 474 – 478. – Режим доступу : https://ieeexplore.ieee.org/abstract/document/8632152.

6. Лужецький В. А. Основи інформаційної безпеки : навч. посіб. / В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович. – Вінниця : ВНТУ, 2013. – 221 с.

7. Roebuck Kevin. Random Password Generators: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity / Kevin Roebuck. – Vendors : Lightning Source Incorporated, 2011. – 426 p.

8. A very weak and widespread password is written on a sticker / st. [Електронний ресурс] / Katherine Parker // Secure Networkers. Blog Space. – June 29, 2020. – Режим доступу : https://securenetworkers.com/a-very-weak-and-widespread-password-is-written-on-a-sticker-st/.

9. Password Security : What Users Know and What They Actually Do [Електронний ресурс] / Shannon Riley // Usability News. – 2006. – Vol. 8, Issue 1. – Режим доступу : https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.597.5846&rep=rep1&type=pdf.

10. ДСТУ ISO/IEC 27000:2019 Інформаційні технології. Методи захисту. Системи керування інформаційною безпекою. Огляд і словник термінів (ISO/IEC 27000:2018, IDT). [Чинний від 01.11.2019]. [Електронний ресурс] – Режим доступу : http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=85795.

11. Дискреційна модель та метод розмежування прав доступу до розподілених інформаційних ресурсів [Електронний ресурс] / Ю. В. Баришев, В. А. Каплун, К. В. Неуйміна // Наукові праці ВНТУ. – 2017. – № 2. – Режим доступа до журн. : https://praci.vntu.edu.ua/index.php/praci/article/view/506/501.

12. Advances in Password Recovery Using Generative Deep Learning Techniques. Artificial Neural Networks and Machine Learning / D. Biesner, K. Cvejoski, B. Georgiev [et al.] // ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia. – September 14–17, 2021. – Proceedings, Part III. – P. 15 – 27.

13. Publishable Humanly Usable Secure Password Creation Schemas. Proceedings [Електронний ресурс] / Manuel Blum, Santosh Vempala // The Third AAAI Conference on Human Computation and Crowdsourcing (HCOMP-15). – 2015. – Режим доступу : https://www.aaai.org/ocs/index.php/HCOMP/HCOMP15/paper/viewFile/11587/11430.

14. Intel Digital Random Number Generator (DRNG) : Software Implementation Guide [Електронний ресурс] / Revision 1.1. – 2012. – Режим доступу : https://www.intel.com/content/dam/develop/external/us/en/documents/441-intel-r-drng-software-implementation-guide-final-aug7.pdf.

15. Randomness generation [Електронний ресурс] / Daniel J. Bernstein, Tanja Lange. – 16 May 2014. – Режим доступу : https://cr.yp.to/talks/2014.05.16/slides-dan+tanja-20140516-4x3.pdf.

16. Баришев Ю. В. Методи формування псевдовипадкових чисел для псевдонедетермінованих геш-функцій / Ю. В. Баришев, Т. А. Кравчук // Тези доповідей Третьої міжнародної науково-практичної конференції "Інформаційні технології та взаємодії", м. Київ, 8-10 листопада 2016 року. – Київ, Видавничо-поліграфічний центр "Київський університет", 2016. – С. 207 – 208.

Стаття надійшла до редакції 18.06.2022.
Стаття пройшла рецензування 22.06.2022.

Баришев Юрій Володимирович – к. т. н., доцент кафедри захисту інформації.
Вінницький національний технічний університет.

Чайкін Михайло Михайлович – аспірант.
Інститут проблем моделювання в енергетиці Г. Є. Пухова. Національна академія наук України.

Кохан Олександр Володимирович – студент групи 1БС-20м, факультет інформаційних технологій та комп'ютерної інженерії.
Вінницький національний технічний університет.