УДК 004.056.52

О.О. Тимченко, Львів

ВІДДАЛЕНА АУТЕНТИФІКАЦІЯ НА ОСНОВІ ТОТР-АЛГОРИТМУ

One of the methods for implementing reliable remote authentication systems is considered. Method of generating one-time passwords for authenticating users based on time and secret key. Its reliability, weaknesses and vulnerabilities has been analyzed. It is shown that the introduction of the proposed method allows to substantially strengthen the protection of authentication systems.

Keywords: remote authentication, generation of one-time passwords.

Розглядається один з методів реалізації надійних систем віддаленої аутентифікації – метод створення одноразових паролей для аутентифікації користувачів на основі часу та секретного ключа, проаналізовані його надійність, слабкості та вразливості. Показано, що впровадження запропонованого способу дозволяє суттєво підсилити захист систем аутентифікації.

Ключові слова: віддалена аутентифікація, побудова одноразових паролей

Вступ

Аутентифікація користувача – це перевірка, чи дійсно користувач є тим, за кого він себе видає. Різні методи аутентифікації необхідні, фактично, у всіх системах обмеження і розмежування доступу до даних – як розподілених, так і призначених для захисту окремого комп'ютера або мобільного пристрою.

Для коректної аутентифікації користувача необхідно, щоб користувач пред'явив аутентифікаційну інформацію – деякі унікальні дані, якими повинен володіти тільки він і ніхто інший.

Існує три основних типи аутентифікаційної інформації:

1. користувач знає якусь інформацію (наприклад, своє ім'я (логін) та пароль);
2. користувач володіє носієм інформації, що міститься якусь інформацію про власника, або має спеціальні характеристики (USB-накопичувач, цифровий токен, тощо);
3. аутентифіційна інформація є невід'ємною частиною користувача (відбиток пальця або будь-яка інша біометрична інформація.

Найбільш поширеним із способів є перший. Більшість мережевих систем використовують цей підхід для аутентифікації користувачів. Проте разом із розвитком обчислювальних можливостей сучасних серверів зростають і хакерські можливості у сфері крадіжок паролів та злому персональних даних. Розглянемо посилення захисту системи шляхом впровадження двоетапної валідації паролю за допомогою розробленого ТОТР алгоритму.

Теоретична частина

TOTP (Time-based One-Time Password Algorith) – алгоритм побудови одноразових паролів для захищеної односторонньої аутентифікації, коли сервер виконує перевірку справжності клієнта. Алгоритм є покращення алгоритму HOTP (HMAC – Based One-time Password Algorith), у якому пароль генерується на основі поточного часу, а не інкрементованого параметру. Ця версія алгоритму дозволяє будувати ідентичні одноразові паролі на сервері і пристрої, з якого ініціюється спроба аутентифікації без прямого зв'язку між ними. HMAC (hash-based message authentication code) – тобто аутентифікаційне повідомлення захешоване за допомогою однієї із стандартних хеш функцій, що використовується для перевірки дійсності особи користувача.

Хеш-функція дозволяє перетворювати дані будь-якої довжини у короткий "цифровий відбиток пальця". Довжина значення цієї функції не залежить від довжини тексту вхідного параметра. Наприклад, у випадку алгоритму SHA-1 довжина такого відбитку становить 160 біт. Прикладами хеш функцій є MD5, SHA-1, ГОСТ_Р_34.11-94.

Застосування та впровадження TOTP

Для детального розбору та імплементації алгоритму розглянемо послідовність дій які відбуваються для генерації, а також застосування одноразового паролю.

Рис.1. Отримання секретного ключа клієнтом через захищений канал

На рис. 1 зображено процес ініціалізації зв'язку між сервером та користувачем.

1. Сервер генерує секретний ключ K, представлений як проміжна байтова стрічка. Цей ключ передається по захищеному приватному каналу користувачу, який ініціює процес зв'язку.
2. На цьому етапі також перевіряється відповідність UNIX-часу між сервером та користувачем. Якщо час співпадає, тоді обирається початок відліку T0 (по замовчуванню, обирають як час з моменту початку UNIX-ери.
3. Обирається інтервал часу T1. Він буде використовуватись для обрахунку інкрементованого кроку значення лічильника C.
4. Обирається хеш-функція, яка буде використана (по замовчуванню застосовується SHA-1).

5. Обирається довжина токена, як результат виконання хеш-функції (по замовчуванню 6 символів).

Треба зауважити, що більшість сучасних реалізацій алгоритму на стороні сервера повертає користувачу ключ, закодований за допомогою 32-бітного кодування, тому клієнт, перед його використанням також має його декодувати. Якщо сервер у своїй відповіді на запит, разом із ключем K не повертає параметри T0 і T1 – це означає, що було обрано значення по замовчуванню.

Коли усі вхідні параметри є узгоджені токен генерується за наступною логікою:

1. Вирахувати значення лічильника C, як кількість інтервалів часу T1, що пройшли від початку відліку T0.
2. Вирахувати HMAC хеш H від K та C.
3. Означити останні 4 старші біти H і використовувати їх як зсув O.
4. Означити 4 байти із H починаючи з O старших байтів, не враховуючи старший біт, як цілочисельне I.
5. Токеном буде найменші N цифр з I за основою 10. Якщо результуюче число має менше цифр ніж N, то його доповнюють нулями зліва.

На рис. 2 наведено принцип застосування алгоритму на практиці. Легко бачити, що одночасно і сервер і пристрій користувача повинні вирахувати токен на основі відомих спільних даних (часу і секретного ключа). Коли сервер отримує токен він порівнює його із тим, що вирахував сам. Якщо токени співпали тоді користувач аутентифікується у системі успішно. Якщо ні – тоді користувачеві буде відмовлено у доступі до системи.

Часто на стороні сервера прораховується декілька токенів для значень лічильника C та C-1. Це робиться для того, щоб якщо відбулась затримка у момент між тим як токен був згенерований і тим як він був надісланий на сервер то ця похибка не завадила розпізнати дійсного користувача. Причин таких затримок може бути безліч: невелика різниця у часі на годиннику на пристрої та сервері, затримки швидкості передачі мережевих пакетів, або ж швидкість введення коду користувачем.

Надійність алгоритму

Системи захисту, спроектовані із використанням TOTP є дуже надійними. В більшості випадків вони є невразливі до поширених криптографічних атак, наприклад:

- Атака типу “людина посередині” (*Man-in-the-middle*). Зловмисник, що перехопив повідомлення із кодом не зможе підробити його, так як пароль у системі постійно змінюється. Ця властивість робить системи, захищені двоетапною аутентифікацією, невразливими для більшості хакерських атак.
- Атака повторного відтворення – одна з найбільш поширених атак, буде неневдалою, оскільки елементарне повторне надсилання раніше валідного коду не дозволить зловмисникам пройти аутентифікацію.

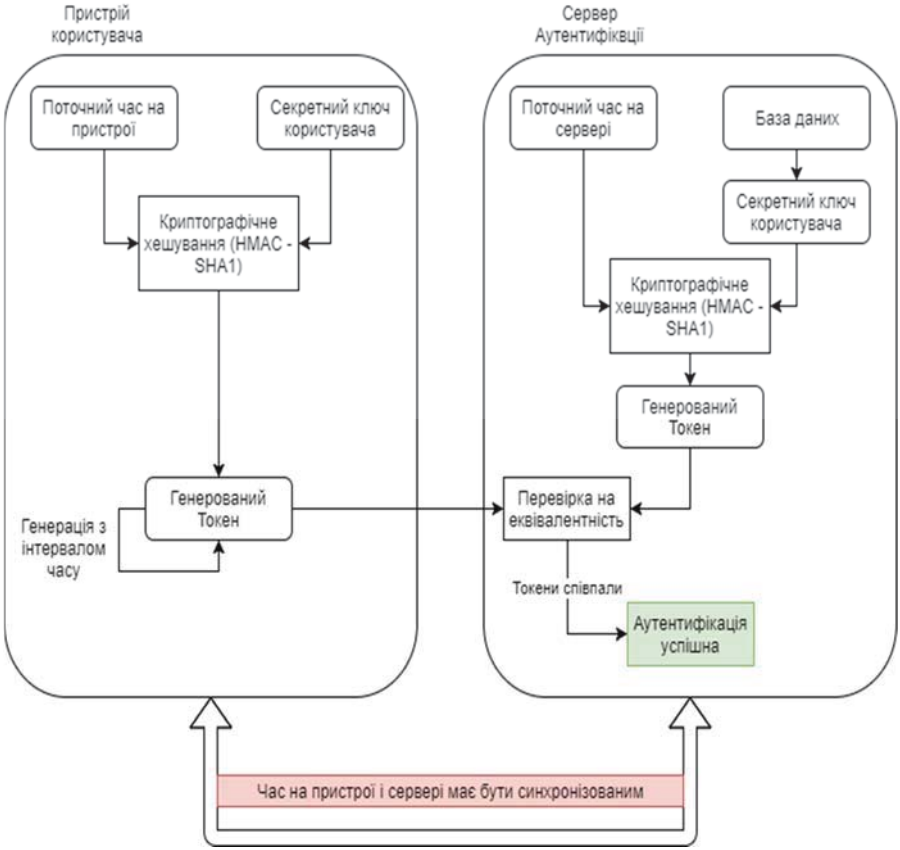

Рис. 2. Пояснення застосування алгоритму

Слабкості та вразливості

TOTP код може бути викрадений через мережу так само як і простий пароль. Проте, для того щоб цей код зберігав валідність він має бути викрадений у реальному часі і використаний на протязі невеликого інтервалу часу.

Хакер, що викрав спільний секретний ключ може власноруч генерувати валідні TOTP коди. Це веде до повної компроментації системи захисту. Це може бути особливо небезпечно, коли хакеру вдається отримати ключ до системи аутентифікації великої бази даних.

Реалізації алгоритму що не обмежують кількість спроб надсилання даних на сервер є вразливими до простого перебору кодів.

Також існує вразливість пов'язана із синхронізацією годинників сервера та користувача, оскільки існує ризик розсинхронізації часу між системами.

Оскільки основним параметром для підрахунку значень лічильника є час, то при неспівпадінні, усі спроби користувача на аутентифікацію завершаться невдачею. Слід додати, що вірогідність того, що така ситуація станеться є дуже низькою.

Висновки

Двоетапна аутентифікація є не тільки дуже надійним сучасним способом захисту доступу до користувацької системи або бази даних, а ще і простим у своїй реалізації. Імплементація за поясненим алгоритмом дозволяє за допомогою малозатратних обрахунків проводити генерацію і валідацію токенів тимчасових паролів у реальному часі. Проаналізувавши сучасні аутентифікаційні системи та відомі способи їх злому можна зробити висновок, що впровадження запропонованого способу дозволяє суттєво підсилити їх захист.

1. https://tools.ietf.org/html/rfc2104
2. https://tools.ietf.org/html/rfc4226
3. http://nathschmidt.net/breakdown-hmac-based-one-time-passwords.html
4. https://blogs.forgerock.org/petermajor/2014/02/one-time-passwords-hotp-and-totp/
5. https://tools.ietf.org/rfc/rfc6238.txt
6. https://www.bytemag.ru/articles/detail.php?ID=9101

Поступила 1.02.2018р.

УДК 004.352, 655.2

Б.М. Гавриш [1], к.т.н., ст. викл, О.В. Тимченко [1, 2], д.т.н., професор,
Р.О.Кульчицький [1], аспірант, О.Є. Семенова[3], асист. каф. ІТВС

ОСОБЛИВОСТІ ПОБУДОВИ НЕЙРОМЕРЕЖЕВИХ СИСТЕМ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

Розглядаються особливості розпізнавання образів із застосуванням штучних нейронних мереж, можливості їх використання та обмеження. Показана можливість створення нейромережевої технології розпізнавання зображень текстів лінгвістичним методом.

Ключові слова: розпізнавання зображень, штучні нейронні мережі, створення граматики

[1], Українська академія друкарства
[2] Uniwersytet Warmińsko-Mazurski w Olsztynie
[3] Національний університет «Львівська політехніка»

