ISSN: 1859-2171
e-ISSN: 2615-9562
TNU Journal of Science and Technology 225(06): 479 - 485

NGHIÊN CỨU XÂY DỰNG HỆ THỐNG XÁC THỰC ĐA NHÂN TỐ CHO WEBSITE

Đặng Xuân Bảo*, Trần Thị Xuyên, Hoàng Thu Phương, Nguyễn Thị Hồng Hà
Học viện Kỹ thuật Mật mã

TÓM TẮT

Một trong những yêu cầu quan trọng của an toàn thông tin là xác thực danh tính của đối tượng được cấp quyền sử dụng các tài nguyên điện toán của hệ thống như truy nhập tài khoản, đọc và chỉnh sửa tài liệu đối với hệ thống thông tin của cơ quan nhà nước, hay thực hiện các giao dịch trực tuyến trong các hệ thống thương mại điện tử. Phương thức xác thực trực tuyến phổ biến nhất hiện nay là sử dụng mật khẩu. Tuy nhiên, trong bối cảnh hiện nay tính an toàn của phương pháp này không cao. Để tăng tính an toàn trong quá trình xác thực cần sử dụng mật khẩu với các yếu tố khác, hay xác thực đa nhân tố. Bài báo này sẽ giới thiệu một phương pháp xây dựng hệ thống xác thực đa nhân tố cho một website.

Từ khóa: *Xác thực; xác thực đa nhân tố; mã OTP; mã HOTP; mã TOTP; mã QR*

Ngày nhận bài: 14/4/2020; Ngày hoàn thiện: 30/5/2020; Ngày đăng: 31/5/2020

RESEARCH AND BUILDING MULTI FACTOR AUTHENTICATION SYSTEM FOR WEBSITE

Dang Xuan Bao*, Tran Thi Xuyen, Hoang Thu Phuong, Nguyen Thi Hong Ha
Academy of Cryptography Techniques

ABSTRACT

One of the most important requirements of information security is to authenticate the identity of the object, who authorized to use the system computing resources such as account access, read and edit documents for the information system of state agencies, or conduct online transactions in e-commerce systems. The most popular online authentication method is to use a password. However, in the current context, the safety of this method is not high. To increase the security for authentication process, the password should be used with other factors, or called multi-factor authentication. This article will introduce a method to build multi- factors authentication system for a website.

Keywords: *Authentication; multi-factor authentication; OTP code; HOTP code; TOTP code; QR code.*

Received: 14/4/2020; Revised: 30/5/2020; Published: 31/5/2020

* Corresponding author. *Email: dangxuanbao.attt@gmail.com*

1. Giới thiệu

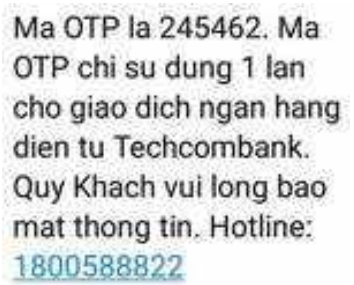
Xác thực là việc xác lập hoặc chứng thực một thực thể đáng tin cậy, có nghĩa là những thông tin do một người đưa ra hoặc về một cái gì đó là đúng đắn. Xác thực một đối tượng còn có nghĩa là công nhận nguồn gốc của đối tượng, còn xác thực một người thường bao gồm việc thẩm tra nhận dạng cá nhân của họ. Xác thực là khâu đặc biệt quan trọng để bảo đảm an toàn cho hoạt động của một hệ thống thông tin. Đó là một quy trình nhằm xác minh nhận dạng số của bên gửi thông tin trong liên lạc trao đổi, xử lý thông tin, chẳng hạn như một yêu cầu đăng nhập [1]. Mật khẩu đăng nhập là một dạng xác thực danh tính người dùng. Người dùng sẽ sử dụng nó để đăng nhập vào website, ứng dụng và dữ liệu. Nhưng kiểu xác thực danh tính truyền thống này dễ dàng bị hacker bẻ khóa. Để tăng tính an toàn cho quá trình xác thực, giải pháp xác thực đa nhân tố đã được sử dụng. Xác thực đa nhân tố là sự kết hợp tối thiểu của 2 trong 3 nhân tố sau đây [2]:

- Something that you have: Những gì mà chỉ bạn mới có. Chẳng hạn như thẻ thông minh, thiết bị token của ngân hàng.

- Something that you are: Những gì thuộc về sinh trắc của bạn. Ví dụ như đồng tử, tròng mắt, dấu vân tay hay giọng nói của bạn...

- Something that you know: Những gì mà chỉ bạn mới biết. Là những thông tin mà một mình bạn tạo ra như: tên đăng nhập, mật khẩu đăng nhập, tên người thân của bạn, ngôi trường cấp 3 của bạn,...

Xác thực đa nhân tố thường hay được sử dụng trong lĩnh vực như kinh doanh online, ngân hàng, mạng xã hội... Hiện nay, trong xác thực đa nhân tố, nhân tố đầu tiên được sử dụng thường là mật khẩu do người dùng cài đặt, còn nhân tố thứ hai thường là mật khẩu sử dụng một lần (OTP). Đối với mật khẩu sử dụng một lần, mã OTP (hình 1) và Token vật lý (hình 2) là hai lựa chọn được sử dụng nhiều nhất hiện nay [3].



Hình 1. *Mã OTP được gửi qua tin nhắn khi thực hiện giao dịch ngân hàng điện tử Techcombank*

Hình 2. *Token key sinh mã OTP khi thực hiện giao dịch ngân hàng điện tử Techcombank*

Tuy nhiên, chi phí khi sử dụng mã OTP là rất lớn (phí tin nhắn, phí thiết bị). Vì vậy, trong bài báo này chúng tôi trình bày giải pháp xây dựng xác thực đa nhân tố sử dụng hệ thống phần mềm, giúp tiết kiệm chi phí cho quá trình xác thực đa nhân tố. Thực nghiệm được tiến hành trên một website giả định.

2. Giải pháp và công nghệ thực hiện

Hiện nay có hai cách sinh mã OTP là sinh theo phương pháp đồng bộ bộ đếm (RFC-4226) và sinh theo phương pháp đồng bộ thời gian (RFC-6238). Theo RFC-4226, thuật toán sinh OTP dựa trên đồng bộ bộ đếm được gọi là HMAC-Based One-Time Password Algorithm (HOTP), với

$$HOTP(K,C) = Truncate(HMAC\_SHA\text{-}1(K,C)) \quad (1)$$

Trong đó:

- K: Giá trị chia sẻ bí mật giữa Client và Server.

- C: Bộ đếm đã được đồng bộ giữa Client và Server, C có độ dài 8 bytes.

- Truncate(): Hàm tách chuỗi, thực hiện việc trích xuất kết quả từ hàm băm Hash để có được mật khẩu OTP.

- HMAC_SHA-1(K,C) là một hàm tính toán dựa trên thuật toán HMAC kết hợp với hàm băm SHA-1 của giá trị K và bộ đếm C.

$$HMAC\_SHA\text{-}1(K,C)=SHA\text{-}1(K\oplus C1\ldots \| SHA\text{-}1(K\oplus C2\ldots \| C)) \quad (2)$$

với:

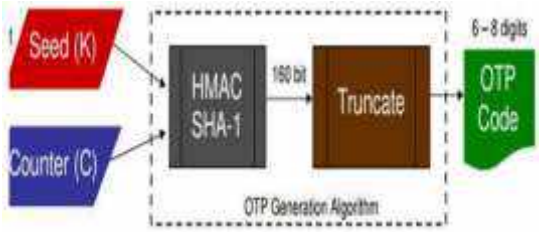
- C1 = 0x36 (36 trong hệ thập lục phân).

- C2 = 0x5c (5C trong hệ thập lục phân).

Giá trị mật khẩu được tính theo công thức:

$$HOTP = HOTP(K,C) \bmod 10^d \quad (3)$$

Trong đó: d là số chữ số của OTP, thông thường một mật khẩu OTP sinh ra có độ dài từ 6 đến 8 chữ số.



Hình 3. *Sơ đồ mô tả thuật toán HOTP sinh mật khẩu OTP*

Tiêu chuẩn RFC 6238 dựa vào thuật toán Time-based One Time Password (TOTP). Nó có phương thức hoạt động tương tự HOTP đã trình bày ở mục trên. TOTP cũng có hai nhân tố là khoá chia sẻ và một bộ đếm (giống như giá trị K và C trong HOTP), nhưng bộ đếm của TOTP hoạt động khác so với HOTP. Cụ thể thuật toán TOTP sinh mật khẩu OTP dựa theo thời gian thì giá trị T về thời gian được tính như sau:

$$T = (Tcurrent\_unix\_time - T0) / X \quad (4)$$

Trong công thức (4) ở trên:

- Tcurrent_unix_time là giá trị thời gian hiện tại được tính theo thời gian Unix (được tính từ thời điểm của Unix Epoch là ngày 01/01/1970 theo UTC (giờ chuẩn quốc tế).

- T0: Là giá trị thời gian ban đầu (thường chọn T0=0).

- X: Là bước thời gian, đây là tham số quyết định thời gian hợp lệ của mật khẩu OTP.

- T là kết quả tính (đã lấy phần nguyên) từ công thức tính toán trên.

Thuật toán TOTP dựa trên thuật toán HOTP (RFC-4226) thay giá trị đếm (C) bằng giá trị thời gian (T):

$$TOTP = HOTP(K, T) \quad (5)$$

Đối với thuật toán TOTP độ dài của mật khẩu OTP được tính như sau:

$$TOTP = TOTP(K, T) \bmod 10^d \quad (6)$$

Trong công thức (6): d là số chữ số của mật khẩu OTP. Cũng giống với HOTP, TOTP thông thường cũng có độ dài từ 6 đến 8 ký tự.

So sánh hai phương pháp này, phương pháp sinh mã OTP theo đồng bộ thời gian an toàn hơn vì đảm bảo việc mã OTP được sinh ra chỉ có tác dụng trong một thời gian ngắn chọn trước, còn đối với mã OTP được sinh ra theo phương pháp đồng bộ đếm thời gian tác dụng phụ thuộc vào số sự kiện đăng nhập và có thể tồn tại một khoảng thời gian dài, dẫn đến việc có thể bị tìm ra [4]. Vì vậy, trong nghiên cứu này lựa chọn phương pháp sinh mã OTP theo đồng bộ thời gian (RFC-6238). Với phương pháp đồng bộ thời gian, mã OTP được sinh theo thuật toán Time-based One Time Password [5]. Trong thuật toán Time-based One Time Password việc lựa chọn tham số X là rất quan trọng, ảnh hưởng đến hoạt động của hệ thống. Nếu X quá lớn sẽ giúp kẻ tấn công dò ra được mã OTP, nếu X quá bé sẽ gây khó khăn cho người dùng. Trong RFC-6238 thời gian X mặc định là 30 giây. Để đảm bảo tính bí mật của khóa chia sẻ, việc gửi khóa chia sẻ được thực hiện thông qua mã QR [6].

3. Thiết kế xây dựng hệ thống

Cấu trúc hệ thống

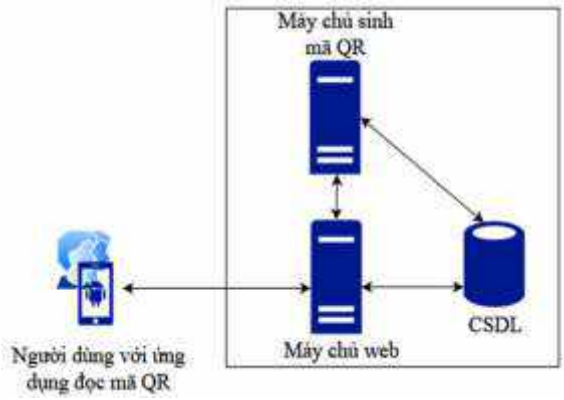
Như vậy, hệ thống xác thực đa nhân tố cho website bao gồm những thành phần sau đây: máy chủ sinh mã QR; máy chủ web; máy chủ cơ sở dữ liệu; ứng dụng đọc mã QR và sinh mã OTP trên điện thoại của người dùng (hình 4).

- Máy chủ sinh mã QR làm nhiệm vụ sinh khóa chia sẻ, đóng gói khóa chia vào một mã QR và gửi tới ứng dụng đọc mã QR trên máy người dùng thông qua giao diện trên máy chủ web.

- Ứng dụng đọc mã QR trên máy người dùng đọc mã QR từ máy chủ gửi đến, lấy khóa chia sẻ và sinh mã OTP theo RFC-6238. Mã OTP được sinh ra sẽ sử dụng trong quá trình đăng nhập của người dùng.

- Máy chủ web tiến hành xác thực người dùng bằng mật khẩu và mã OTP

- Máy chủ cơ sở dữ liệu lưu dữ liệu của từng người dùng như tên đăng nhập, mật khẩu, mã QR...



Hình 4. *Hệ thống xác thực đa nhân tố cho website*

4. Tích hợp hệ thống token mềm trong quá trình xác thực người dùng

Modul xác thực được xây dựng dựa trên sự kết hợp đăng nhập bằng username và mật khẩu cộng với việc xác thực mã OPT.

Người dùng sẽ phải đăng kí tài khoản với hệ thống thương mại điện tử. Sau đó người dùng có thể tùy chỉnh bật tắt tính năng xác thực hai lớp trong phần cài đặt.

Khi bật tính năng xác thực hai lớp phía server sẽ tạo ra một mã QR có dạng như sau:

optauth://totp/<**Host**>%3A<**Username**>?secret=<**xxxxxxxxx**>&digits=<**x**>&issuer=<**Host**> (7)

Trong đó

- Host: sẽ là tên miền của hệ thống.

- Username: tên đăng nhập của người dùng.

- serect: là khóa bí mật được hệ thống tạo ra.

Để có thể thực hiện việc xác thực bằng mã QR, đầu tiên người dùng cần bật tính năng này. Sau khi bật thành công máy chủ sẽ sinh một mã QR có dạng (7) khác nhau cho mỗi người dùng. Người dùng sử dụng phần mềm đọc mã QR trên điện thoại để nhận khóa bí mật làm tham số đầu vào cho quá trình sinh mã OTP.

Trong phạm vi bài báo này phần mềm đọc mã QR được sử dụng là Google Autheticator [7]. Sau đó người dùng sử dụng mật khẩu và mã OTP được sinh ra để tiến hành xác thực khi đăng nhập. Trình tự hành động của người dùng được nêu trong hình 5.

Với lần đăng nhập đầu tiên vào website, người dùng chỉ sử dụng mật khẩu. Trong lần đăng nhập này người dùng gửi yêu cầu bật tính năng xác thực hai lớp. Khi tính năng xác thực hai lớp được bật trên website, người dùng sẽ nhận được một mã QR do máy chủ sinh mã QR sinh ra qua giao diện của website. Người dùng sử dụng ứng dụng quét mã QR để đọc mã này và sinh mã OTP theo RFC-6235 và sử dụng cho các lần đăng nhập tiếp theo. Từ lần đăng nhập thứ hai người dùng cần sử dụng cả mật khẩu và mã OTP được sinh ra trên ứng dụng Google Autheticator để tiến hành xác thực.

Về phía máy chủ website, khi nhận được yêu cầu sử dụng tính năng xác thực từ phía người dùng, gửi yêu cầu này đến máy chủ sinh mã QR. Máy chủ sinh mã QR tiến hành sinh mã QR cho từng người dùng theo RFC-623(xóa phần chữ đỏ này). Mã QR này được gửi tới giao diện website và lưu vào cơ sở dữ liệu. Khi người dùng đăng nhập lại sẽ tiến hành kiểm tra mật khẩu và mã OTP của người dùng (hình 6).

Hình 5. *Sơ đồ thuật toán hoạt động người dùng khi đăng nhập*

Hình 6. *Sơ đồ thuật toán hoạt động của máy chủ khi xác thực*

4.1. Thiết kế các chức năng của máy chủ sinh mã QR

Để tạo mã QR trong nghiên cứu sử dụng framework mã nguồn mở Django (thư viện django otp) [8]. Đoạn code tạo mã QR và gửi trả về cho client như sau:

```
    def get(self, request, *args, **kwargs):
    # Get the data from the session
    try:
        key =
self.request.session[self.session_key_name]
    except KeyError:
        raise Http404()
    # Get data for qrcode
    image_factory_string = getattr(settings,
'TWO_FACTOR_QR_FACTORY',
self.default_qr_factory)
    image_factory =
import_string(image_factory_string)
    content_type =
self.image_content_types[image_factory.kind
]
    try:
        username =
self.request.user.get_username()
    except AttributeError:
        username = self.request.user.username
    otpauth_url =
get_otpauth_url(accountname=username,
issuer=self.get_issuer(),secret=key,
digits=totp_digits())
    # Make and return QR code
    img = qrcode.make(otpauth_url,
image_factory=image_factory)
    resp = HttpResponse(content_type=con
                tent_type)
                img.save(resp)
                    return resp
```

4.2. Tính năng tạo mã OTP trên Google Authenticator

Người dùng sử dụng ứng dụng Google Authenticator đã được cài trên điện thoại của mình để quét mã QR. Hai tham số shared secret key và thời gian thực được kết nối với

nhau và đưa qua hàm băm SHA-1. Đầu ra sẽ là mã TOTP được chuyển về dạng dễ sử dụng cho người dùng gồm 6 ký tự. Tiến hành nhập mã OTP và gửi lên phía server, nếu mã OTP vừa nhập trùng với mã OTP được sinh trên server thì server sẽ tiến hành lưu trạng thái xác thực hai lớp của tài khoản này đã được bật ở cơ sở dữ liệu.

Đoạn code mẫu tạo ra mã OTP trong Google Autheticator:

original_secret = xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx

secret = BASE32_DECODE(TO_UPPERCASE(REMOVE_SPACES(original_secret)))

input = CURRENT_UNIX_TIME() / 30

hmac = SHA1(secret + SHA1(secret + input))

four_bytes = hmac[LAST_BYTE(hmac):LAST_BYTE(hmac) + 4]

large_integer = INT(four_bytes)

small_integer = large_integer % 1,000,000.

5. Kết quả thử nghiệm

Môi trường yêu cầu để cài đặt và triển khai hệ thống là: Hệ điều hành Linux, đã cài đặt Python 3.6, Django 1.11, cơ sở dữ liệu SQLite. Yêu cầu phần cứng: ổ cứng trống 50Mb, RAM từ 4Gb.

Đầu tiên, cần khởi chạy máy chủ sinh mã QR (hình 7).

```
→ example git:(master) ✗ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 24, 2019 - 09:53:05
Django version 3.0, using settings 'example.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Hình 7. *Hệ thống token mềm*

Tiếp theo, người dùng yêu cầu bật tính năng xác thực hai yếu tố (hình 8).

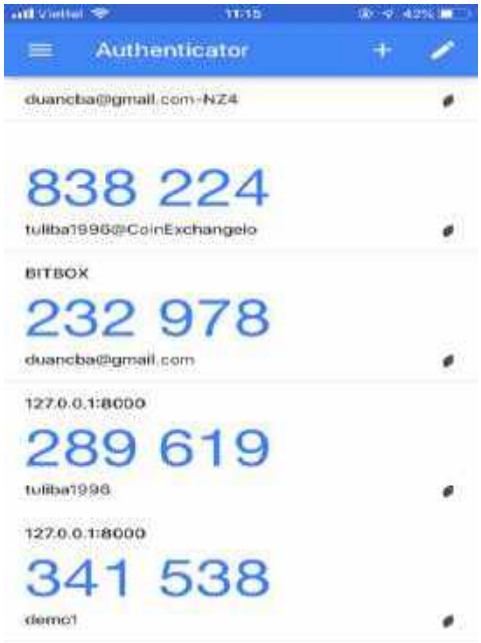
Hình 8. *Bật tính năng xác thực hai lớp*

Sau khi bật tính năng xác thực hai yếu tố, máy chủ sinh mã QR sẽ sinh một mã QR cho người dùng (hình 9).

Sau đó người dùng sử dụng Google Authenticator để đọc mã QR và sinh mã OTP. Kết quả sinh mã OTP như trong hình 10.

Tiến hành đăng nhập sử dụng mật khẩu (hình 11) và mã OTP (hình 12).

Hình 9. *Mã QR được sinh ra*



Hình 10. *Mã OTP được tạo ra*

Hình 11. *Giao diện đăng nhập*

Hình 12. *Sử dụng mã OTP để xác thực*

6. Kết luận

Nghiên cứu đã trình bày sự cần thiết của vấn đề xác thực đa nhân tố trong đảm bảo an toàn thông tin hiện nay. Trong đó xác thực đa nhân tố sử dụng mã OTP là rất phổ biến hiện nay. Kết quả của nghiên cứu là đã phân tích, thiết kế và xây dựng thành công hệ thống xác thực đa nhân tố sử dụng mã OTP cho một website cụ thể. Hệ thống chạy ổn định, có đáp ứng được nhu cầu bình thường của người dùng. Hệ thống đề xuất có thể tích hợp vào các trang web, các ứng dụng thương mại điện tử như shopee, lazada... để thực hiện chức năng xác thực người dùng cũng như xác thực hành động của người dùng (đặt hàng, đăng sản phẩm, hủy đơn hàng...). Hệ thống đề xuất cũng có thể tích hợp vào hệ thống máy chủ của các ngân hàng để xác thực các giao dịch trực tuyến, tuy nhiên cần xây dựng một ứng dụng sinh mã OTP thay thế cho Google Authenticator để đảm bảo các chuẩn an toàn riêng trong lĩnh vực thanh toán điện tử.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1]. M. S. Merkow, and J. Breithaupt, *Information Security Principles and Practices.* NJ, Prentice Hall, 2005.

[2]. E. Gilman, and D. Barth, *Zero Trust Networks: Building Secure Systems in Untrusted Networks*, 1st edition, Califonia: O'Reilly Media, 2017.

[3]. Le Phuong, "The bank has simultaneously changed the way to receive OTP codes from today", Jul. 01, 2019. [Online]. Available: https://bnews.vn/ngan-hang-dong-loat-doi-cach-nhan-ma-otp-tu-hom-nay/126768.html. [Accessed Jan. 11, 2020].

[4]. C. J. Wu, and J. D. Irwin, *Introduction to Computer Networks and Cybersecurity*. Florida: CRC Press, 2017.

[5]. Internet Engineering Task Force, "TOTP: Time-Based One-Time Password Algorithm". *Internet Engineering Task Force*, RFC6238, 2011. [Online]. Available: https://tools.ietf.org/html/rfc6238. [Accessed Jan. 11, 2020].

[6]. J. R. Vacca, *Computer and Information Security Handbook*. Massachusetts: Morgan Kaufmann, 2017.

[7]. Google, "Google Authenticator OpenSource", Dec. 06, 2018. [Online]. Available: https://github.com/google/google-authenticator. [Accessed Jan. 11, 2020].

[8]. P. Sagerson, "Django-otp", Aug. 24, 2019. [Online]. Available: https://django-otp-official.readthedocs.io/en/ stable. [Accessed Jan. 11, 2020].