2FA Using Passwords and Time-based One-time Passwords

Ward M. Zahran

April 30, 2024

University of Jordan

Prelude

Table of Contents

Introduction

Project Goals

This project implements TOTP as a second factor for traditional password authentication.

Presentation Goals

- Showcase how TOTP and its ancestor HOTP work.
- Showcase my implementation of 2FA using TOTP.

Why talk about HOTP?

TOTP is an extention of HOTP so understanding the latter is paramount to understanding the former.

Explaining HOTP and TOTP

Understanding HOTP (RFC 4226)

HOTP, short for HMAC-Based One-Time Password, is an algorithm which uses:

- A symetric secret key (K).
- A hash function (H) the default being SHA-1.
- A counter value (C) in order to generate one time passwords.
- A HOTP value length (D) (6–10, default is 6, and 6–8 is recommended).

These values must be agreed upon by both parties, in our case the webapp and the user.

How HOTP works mathmatically.

$$HOTPvalue = HOTP(K, C) mod 10^D$$
$$HOTP(K, C) = truncate(HMACH(K, C))$$
$$truncate(MAC) = extract31(MAC, MAC[(198 + 4) : (198 + 7)])$$
$$extract31(MAC, i) = MAC[(i8 + 1) : (i8 + 481)]$$

How HOTP works in human words.

1. We generate a HOTP value from the HTOP(K, C) modulo 10 to the power of the key length D.
2. The HOTP function first generates the HMAC using K and C and then truncates it.
3. The truncate function does some math on the hash and then produces the value.
4. Once the value generated both parties increment the counter C independently.

How TOTP works.

TOTP uses the same exact algorithm but instead of a counter we use a new value based on time which we will call T.

So our algorithm ends up being:
$HOTPvalue = HOTP(K, T) mod 10^D$

How to calculate T.

Per RFC 6238:

T = (Current Unix time - T0) / X

where:

- T0 is is the Unix time to start counting time steps (default value is 0, i.e., the Unix epoch). Meaning an agreed upon time from which to start counting the time.
- X is the time step in seconds (default value X = 30 seconds). Meaning the amount of time before each code changes.

The Project

Why TOTP over HOTP?

It is easier as I nor the user need to keep track of a counter.

How My Project Works.

1. After user registration and first login the user is prompted with a QR code and an input box.
2. The user scans the code via an authenticator application either on their phone or desktop and then inputs the code that appears on it as to verify that they scanned the QR and that they successfully aquired the secret key.
3. On each login after that the user is prompted with a box to input the number on their application to verify their identity.
4. If the user fails to enter the OTP three times then the user will be forwarded back to the login page as to disable an attacker from brute forcing the OTP in case he has aquired a users password.

Live Demo

Now for a showcase.