yubico

OATH-HOTP

Yubico Best Practices Guide

Copyright

Trademarks

Disclaimer

Contact Information

Yubico Inc
420 Florence Street, Suite 200
Palo Alto, CA 94301
USA
yubi.co/contact

Document Release Date

February 11, 2016

yubico

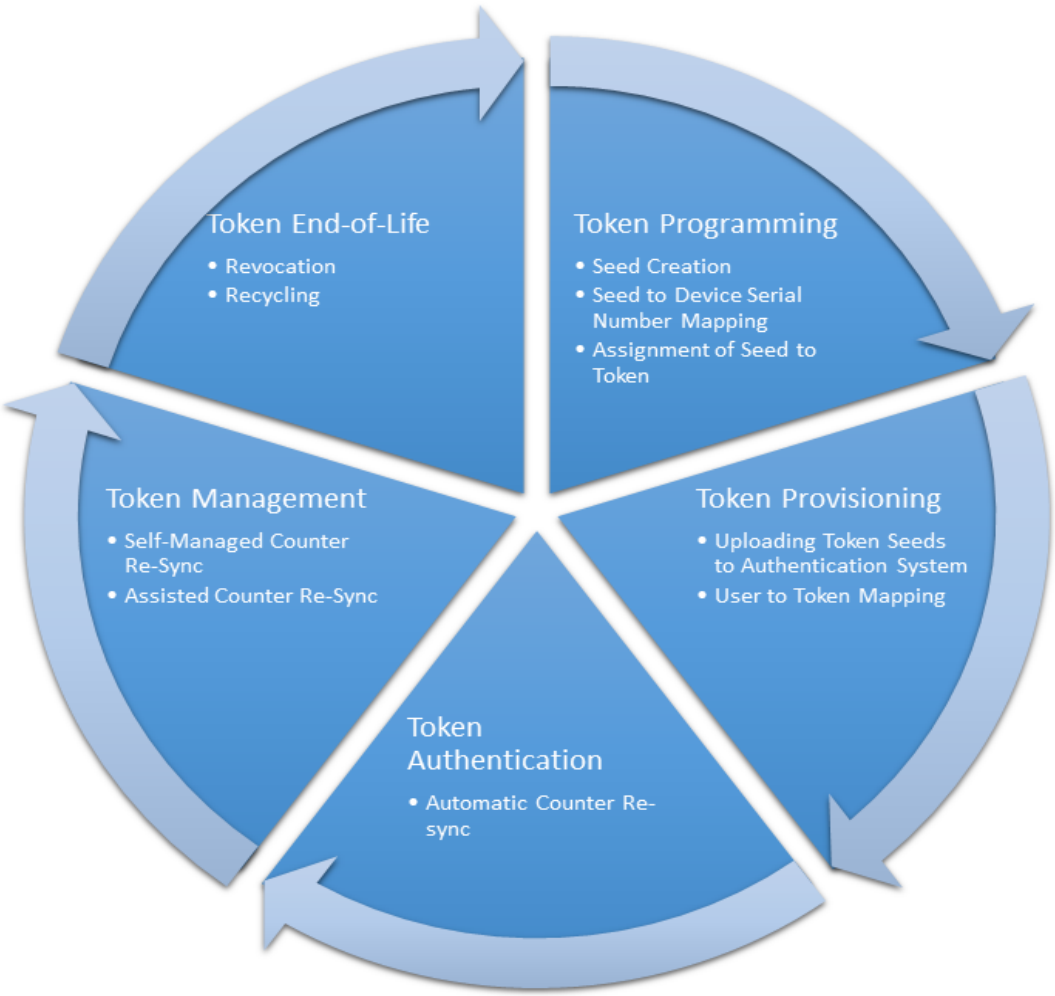
Contents

Introduction

Yubico has assisted in the deployment of authentication systems for many different organizations and with a variety of use cases. This document covers the specific lessons learned and describes the best practices established for deploying Open Authentication Initiative HMAC-based One-Time Password (OATH-HOTP) compliant authentication systems. OATH-HOTP is a standard algorithm for calculating one-time passwords based on a secret (a seed value) and a counter. The specifications for OATH-HOTP is published in IETF RFC 4226.

Tokens and Seeds Lifecycle

Each OATH-HOTP token follows a process for the entire lifecycle of the token, and each step of the lifecycle must be evaluated for inclusion in the processes deployed at the enterprise. The following diagram highlights the phases of the token lifecycle as well as the processes in each phase that must be evaluated.



OATH-HOTP Best Practices

In this section, each major stage of the token lifecycle is analyzed and recommendations are provided for consideration.

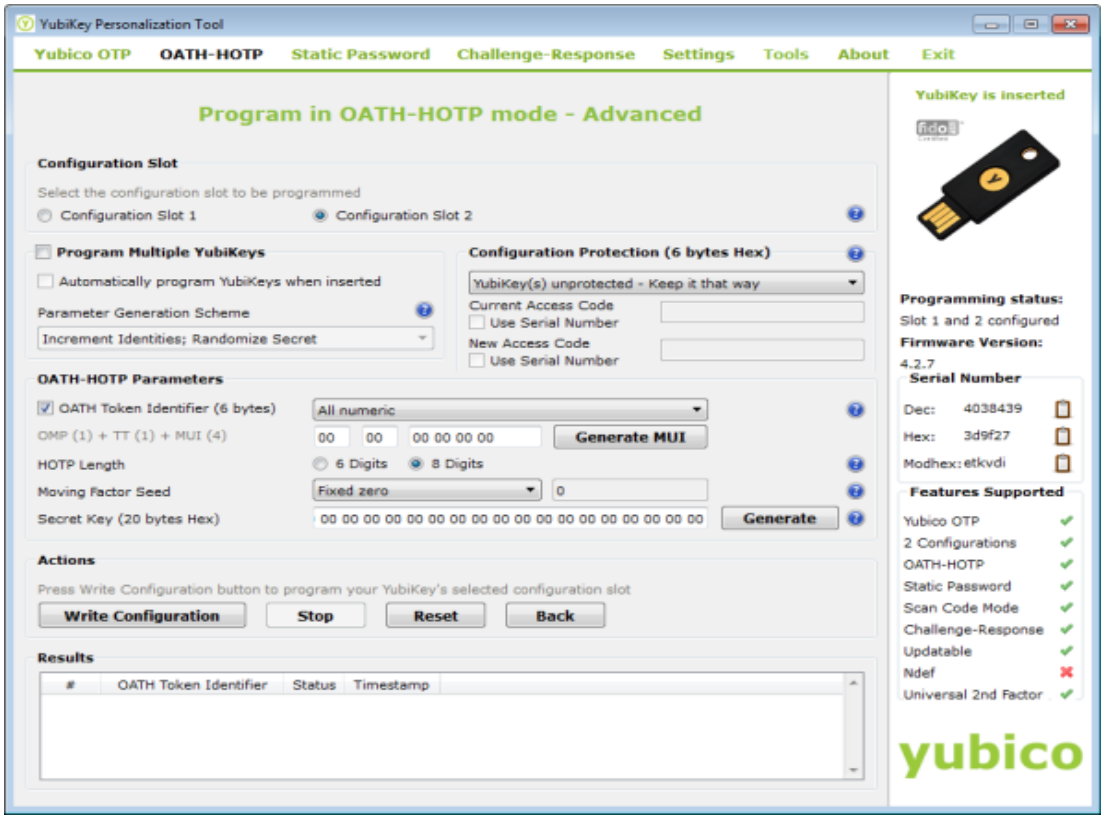
Token Programming

Token programming includes the process for creating the OATH seed, assigning the seed value to a device, and programming the token with the assigned seed value.

Business and Process Decisions

There are several business decisions that need to be made. Each of these business decisions will dictate how the programming and lifecycle will operate.

Option 1A: Customer purchases, programs, and manages the entire lifecycle of the YubiKey for their individual account or for their entire corporate account. The configuration and management of the YubiKeys is done with the YubiKey Personalization tool or by integrating the YubiKey APIs into existing identity management processes.



Option 1B: Customer purchases, programs, and manages the entire lifecycle of the YubiKey for their individual account or for their entire corporate account. The configuration and management of the YubiKeys is done with customized software for each specific implementation. An example of a customized

integration is integrating the configuration of the YubiKey into the badge provisioning processes at the physical Security Offices of an organization.

Option 2: Yubico sells branded OATH-HOTP programmed YubiKey (YubiKey VIP) and manages the programming and configuration of the key.

Option 3: Yubico wholesales the YubiKey to the service provider or reseller (Dell Defender) who can work with Yubico to program YubiKeys, but the service provider or reseller manages the YubiKey inventory.

Technical Decisions

The technical decisions in the token programing phase affect the provisioning and authentication phases of the token lifecycle.

Token Identifiers: One of the most important elements of an OATH design that is enabled by the YubiKey is the usage of an OATH Token Identifier. The OATH Token Identifier has been a highly unused part of the OATH standard due to the effect on the user experience. However, since the YubiKey operates as a keyboard, the user experience of typing a Token Identifier plus a one-time password is reduced to a single touch. Token Identifiers greatly simplify the provisioning process due to each touch for a one-time-password including the token identifier. For more details on Token Identifiers, see Choosing a Token Identifier.

Access Codes: For security reasons, and to avoid accidental reprogramming, YubiKeys can be protected using a configuration protection access code. If the configuration protection access code is set, no one can reprogram the YubiKey unless the correct access code is provided during reprogramming. An important fact about the access code is that it is not possible to retrieve the access code from the YubiKey. Due to this fact, Yubico highly recommends users to record the access code for each YubiKey programmed. This can be achieved easily by ensuring that logging is enabled and the relevant logs are archived.

The following operations are supported when programming with access codes:

- **YubiKey(s) unprotected - Keep it that way**: Use this option if the YubiKey is currently unprotected and you want to keep it that way
- **YubiKey(s) unprotected - Enable protection**: Use this option if the YubiKey is currently unprotected and you want to enable the protection. When you select this option, you also need to provide a new access code.
- **YubiKey(s) protected - Disable protection**: Use this option if the YubiKey is currently protected and you want to disable the protection. When you select this option, you also need to provide the current access code.
- **YubiKey(s) protected - Keep it that way**: Use this option if the YubiKey is currently protected and you want to keep it that way. When you select this option, you also need to provide a new access code.
- **YubiKey(s) protected - Change access code**: Use this option if the YubiKey is currently protected and you want to change the access code. When you select this option, you also need to provide your current access code and a new access code.

Token Provisioning

The token provisioning process is the most important phase of the token lifecycle. Each token must be associated with the user account in either a manual or automated fashion. However, the process for associating a token to a user must be managed to ensure that only the correct tokens are associated with the correct users. For the token to be provisioned, the token seeds must be uploaded to the authentication system and the token must be associated to a user account.

Uploading Token Seeds to Authentication System

Each multi-factor authentication (MFA) system must have a process for uploading core token data into the system. This data includes the OATH seed value and the programmed counter value.

General Recommendation for Token Programming

- For minimal risk of deployment issues, set the OATH-HOTP initial counter to 0 or a fixed value.
- For a more secure system with a slightly higher risk of deployment issues, set the OATH-HOTP counter to a random value per device.

User to TokenID Binding

The process for associating a token to a user account is a major design point for the user experience and system security. There are two major categories of user to token binding: self-service management and centrally-administered management. Each category presents different decisions that enable different levels of assurance as well as different levels of friction for users and administrators.

Self-Service Management Methods

There are two methods for self-service registration: auto and manual token association. Auto token association works when a user with no previously registered tokens authenticates for the first time. Once the authentication occurs using the user name and password, the user is prompted to provide a one-time-password value from his YubiKey. The YubiKey outputs the OATH Token Identifier followed by the OATH Value. The authentication server then maps the TokenID to the user account. Once this mapping has occurred, the token is then registered for the user to move into the authentication phase.

Manual association requires the user to utilize an application to register a token. The token registration must be done before the user attempts to authenticate the first time. The application will pair the token ID to the account. If OATH Token Identifier is not used, the user will need to manually input the token serial number into the system during the registration phase.

Centrally-Administered Management Methods

The traditional method of administering tokens has always been to have an administrator or an agent for the administrator do the pairing of the token on behalf of the user. This model eliminates the need for electronic identity proofing processes as the administrator or agent is able to verify the identity of the user that the token is delivered to. Enabling the OATH Token Identifier can simplify the assignment of the token to a user. With the Token ID enabled, an administrator can insert the YubiKey, touch the button, and extract the Token Identifier from the result.

General Recommendations for User to TokenID Binding

The following are recommendations for creating a system that provides the best security without creating overwhelming process requirements:

- Allow each user to register multiple keys for backup and recovery purposes.
- Deploy the YubiKey 4 Nano on laptops and tablets with USB ports, for a seamless user experience for on boarding and authenticating with the YubiKey.
- Determine the level of identity assurance required before deploying this system.

Token Authentication

The purpose of deploying and managing tokens is to provide users a second factor for authentication. Authenticating with YubiKeys and the OATH-HOTP protocol is as simple as a touch for users. This simple touch of the YubiKey, whether intentional or errant, will produce a new HOTP password, which increments the counter on the YubiKey. If this touch is not intentional then the HOTP counter will increment on the YubiKey without incrementing the counter on the server side. The most important decision to make about the authentication phase is how far ahead the counter on the YubiKey can be from the server side and still authenticate. This window of acceptable counter values is call the look-ahead window.

The look-ahead window is the window that defines the acceptable values for the OATH-HOTP authentication algorithm. For example, if the look-ahead window is 10, and the current value for the server side counter is 100, then any one-time-password generated from the YubiKey with a counter value from 100 to 110 would be valid.

The key to keeping the system working with as minimal potential issues as possible is to utilize the look-ahead window calculation to synchronize the server-side counter value. For example, if the server has the counter set at 100 and the YubiKey is set to 104, when the server verifies that the password is generated with a counter value of 104, the server then sets the counter value to 105.

By using this automatic resync on each successful authentication, the number of times that the token will need to be manually resynchronized will be minimized. Most implementations of HOTP have a look-ahead window of 10 to 15. Yubico recommends a look-ahead window for authentication to be no more than 25.

When the HOTP counter exceeds the look-ahead window, attempting to authenticate even with valid OTPs will fail. In such a use case, the HOTP token will need to be replaced or the counter values resynchronized between server and token.

Token Counter Management

One of the shortcomings in the OATH-HOTP protocol is that the counter on the authenticator can get out of sync with the look-ahead window defined for authentication. There are two primary models for resolving the resynchronization problem: self-service and assisted. The self-service model allows for a user to provide sufficient authentication requirement to re-sync the token. The assisted model requires a user to work with an administrator to reconfigure the token. The self-service model exposes a web application for potential abuse, whereas the assisted model creates an opportunity for the help desk to be exposed to social engineering attacks.

The strategy for handling the counter management problems is driven by two major decision points: what is the cost of a help desk call, and what is the risk of a self-service reset? These questions often lead to the decision to encourage the registration of multiple authentication methods with the same assurance level. This registration allows for a user to use an existing, registered authentication token to resolve their own issues. The backup token reduces the downtime for users, as well as calls to the help desk.

When resynchronizing an HOTP token with counter values outside, the user should be required to submit the token ID of their HOTP device, followed by two or three sequentially generated OTP codes. The codes will be checked against the codes within the resynchronization window for matching sequential codes. If the codes are found, the counter value on the server for the HOTP token will be updated to the counter for the final OTP code.

Most implementations of HOTP have a look-ahead window for resynchronization of 50 to 80. Yubico recommends a look-ahead window of no more than 100. If the token falls outside of the look-ahead window for resynchronization the token, Yubico recommends an investigation as to how that token counter was that far out of sync before extending the window to allow the re-sync to occur.

Alternative methods of self-service password reset include:

- Email link for time-limited counter resync
- Email one-time password
- Alternative MFA method (TOTP, SMS, etc.)

Token End-of-Life

The end-of-life cycle for OATH tokens is the revocation and potential recycling of a token.

Token Revocation

There are several use cases that would cause an OATH-HOTP token to be revoked. To build a secure system users should be enabled to report missing tokens either through the help desk or through a web application. Reporting a lost token should be a part of corporate policy, and users should be educated so there is prompt reporting of the lost token.

These use cases include (but are not limited to):

- user account is no longer active on that system
- user account is deleted from that system because they are no longer an employee
- user reports the device lost or stolen
- user intends to register a new device
- The process for revoking a token is simple and irrevocable. When a token is marked as revoked, the seed and counter values should be deleted from the authentication system. This ensures that the token

cannot be reused or re-assigned without reprogramming. This step ensures that under no circumstances can a compromised token be reintroduced into inventory without being reprogrammed.

Token Recycling

A use case enabled by YubiKeys is the reprogramming of a YubiKey. If the YubiKey has completed the entire lifecycle, the YubiKey can be reprogrammed and re-issued without requiring the acquisition of a new token. To reprogram a YubiKey, the configuration protection access code must either be known or not enforced. For more information on the access code, see Token Programming.

Choosing a Token Identifier

The YubiKey supports the Class A Token Identifier Specification as outlined by https://openauthentication.org/token-specs/. The general format of the 12 character (6 bytes) OATH Token Identifier is as follows: `[MM][TT][UUUUUUUU]`, where:

MM	OATH Manufacturer prefix (OMP)	A 2 character prefix, assigned by OATH. The Yubico OMP has a ModHex value of "ub"
TT	Token Type (TT)	A 2 character token type, assigned by the manufacturer
UUUUUUUU	Manufacturer Unique Identifier (MUI)	8 characters that uniquely identify the token for a given manufacturer and token type

Token Type Identifiers

Following is the allocated token type identifier for OATH-HOTP service implemented with YubiKeys.

Token Type (TT)	**Description**
HE	Symantec VIP Service