

OATH Seed Conversion

Contents

- [1 Introduction](#)
- [2 Pre-requisites](#)
- [3 Python Script](#)
- [4 Microsoft MFA Seed File Formatting Requirements](#)

Introduction

This article explains how to convert the OATH Hard Token seeds from Base16 Hexadecimal to Base 32. This can provide ultimate flexibility for your Hardware token investment, if you intend to import the OATH tokens to other systems in conjunction with using them on the Swivel Secure platform.

Pre-requisites

- Swivel Secure OATH Token seeds in their original distribution format (*.txt file and fields separated by space)
- Some ability to use Python scripts
- Python version 3.7.3
- Python IDLE
- Microsoft Excel or Notepad for CSV file formatting and preparation

Python Script

```
import base64
import codecs
import csv

with open('C:\\Users\\admin\\Desktop\\seeds.txt','rt') as input, open('C:\\Users\\admin\\Desktop\\seeds32.txt','w') as output:

    csvin = csv.reader(input, delimiter=' ')
    csvout = csv.writer(output, delimiter=',')

    for row in csvin:
        hex = row[1]
        b32 = base64.b32encode(codecs.decode(hex, 'hex'))
        b32decoded = b32.decode("utf-8")
        csvout.writerow([row[0]] + [b32decoded])
```

Microsoft MFA Seed File Formatting Requirements

Note: If you want a stronger authentication solution, consider deploying [Sentry SSO with Office 365](#) to take advantage of [Authcontrol v4 Sentry SSO and Adaptive Authentication](#), with Single Sign On to your other corporate applications - instead of using Microsoft MFA.

Prepare a CSV file in the following format:

```
upn,serial number,secret key,timeinterval,manufacturer,model
```

e.g.

```
daniel.croft@swivelcloud.net,2000123456789,QFA56K3C5CGFDQWCJDCNNJHGXYD2DDC,60,SwivelSecure,TOTP
```

Note that timeinterval should be 60 and that manufacturer and model are just arbitrary text and can be anything.

Import your seed file into Azure AD under **Azure Portal > Azure Active Directory > MFA Server > OATH tokens**. With the seeds converted to Base 32 using the above Python script you should then be able to successfully activate and use the tokens.