**Bypassing two factor authentication by calculating the time-based one-time password**

Goal:

Bypassing two-factor QR code based authentication by extracting the final key from a handed QR Code through calculating the time-based one-time password (TOTP).

Abstract:

Bypassing the Two factor authentication (2-FA processes) like sending funds from banks and cryptobanks, using binance as an example with the usage of **OATHAuth**, The **OATHAuth** extension is a time-based one-time password (TOTP) creation service calculated based on the handed OR Code (1).

Main:

Further requirements to reach the 2FA of binance;

A VPN, brave browser and oathauth as well as a qrcode to base32 extractor, this can be accived by simply screenshotting the QR code and sending the QR code through any online service reading out QR codes. The classic login credentials, like username/email and password must be obtained.

A VPN is used to emulate the region and to make binance believe it is the user from the region, additionally a VM Ware with Ubuntu and braavebrowser was set up, the vpn was tunnelbear.

While the login regional lock can be bypassed by the vpn, the transfer of funds is not possible without setting up a 2-FA.

Instead of scanning the QR code with the smartphone, one takes a screenshot and sends it to the QRCode reader or extractor in base32.

The output key is processed with the following pseudocode using oathauth in Ubuntu Terminal (2):

$ oathauth –b totp ‘key’

where the outputkey extracted from the QR Code gets entered as the ‘key’ value, b is for base32 as output, handing over the result digit number (time beased one time passwort totp) that can be directly used to bypass 2-FA protection and to cash out funds in binance.

The output ist a 4-8 digit number. Without using a mobile phone or without using further requirements, the 2-FA was bypassed with a desktop PC and the mentioned software, to potentially eneble the worst possible, stealing funds from binance.

Since I discovered not a financial service not relying on 2FA by google authenticator, this error is global and critical, since **OATHAuth and related software creates the 4-8 digit number as plain output, just by resolving the key to base32 creating a TOTP working in the google authenticator and related software.** Since any kind of encryption in the creation of the TOTP creation by QR code is missing and cannot be implemented without handing over the code itself through the QR code on the open and there is currently no commercial workaround like light based encryption, the general method of 2-FA by QR code is deemed unsafe. Any account protected by QR code based 2-FA can be obtaind control of just by possession of mail and password of said account, biance.us is an example (3).

Further variations of the 2-FA bypass could be to deactivate the existing 2-FA and reactive 2-FA by setting up previously mentioned system or by activating the 2-FA on a e.g. binance account that was only used as cold wallet and is not constantly monitored.

Conclusion:

Further impact is on social media and logins with a 3. Party login possibility through google authenticator and other authenticator systems using TOTP without further security measures. Binance using TOTP authentication with base32 is not a security measurement.

Since it was proven, that every QR code based system has no 2-FA, everyone using a QR code based 2-FA system should switch from QR code verification to mail, SMS or proprietary application based confirmation with an encrypted TOTP beyond base32. This would require a picture/QR based encryption and an application on the second device that has a proprietary application decrypting the 4-8 digit TOTP out of a picture/QR Code, PGP could additionaly be used to set up a 3 Factor authentication.

Carlo D’Abbraccio

3.10.2021

1. <https://blog.separateconcerns.com/2020-04-03-2fa-pass-oathtool.html>
2. <https://github.com/wikimedia/mediawiki-extensions-OATHAuth>
3. [https://issuetracker.google.com/issues/201923171](https://deref-web.de/mail/client/E9nFOMSmML4/dereferrer/?redirectUrl=https%3A%2F%2Fissuetracker.google.com%2Fissues%2F201923171)