GL Gary Linker Created on July 27, 2022

Authenticator app not working with sha-256 and sha-512 hash algorithm

I'm trying to set up 2FA with a service which has SHA-256 configured for TOTP. When i add Microsoft Authenticator app, and try to verify, the codes are always flagged as invalid. The same is true when the service is configured with SHA-512. The only way the authenticator app works is when SHA-1 is configured. I'm told SHA-1 cannot be used because it does not meet the security standards (NIST) and has not been compliant since 2015. Has anyone else run into this? When will Microsoft catch up and adhere to the latest security standards?

This thread is locked. You can vote as helpful, but you cannot reply or subscribe to this thread.

I have the same question (54) Report abuse

Question Info

Last updated June 28, 2024

Views 6,219

Applies to:

Microsoft 365 and Office / Subscr
For business / Security and comp

Replies (2)

Anesu Mwamuka MSFT
Microsoft Agent | Moderator

Replied on July 28, 2022

Report abuse

Hi Gary Linker,

Greetings

Thank you for posting in Microsoft Community forum.

Based on your description I understand you have a concern regarding Authenticator app not working with sha-256 and sha-512 hash algorithm. Please correct me if I am wrong.

I have done some further tests and research to try achieving your goal to enable sha-256 and sha-512 hash algorithm. kindly note Microsoft recommends usage of SHA256 as the algorithm for signing tokens as it is more secure than SHA1 but SHA1 still remains a supported option.

Kindly find this documentation for reference: change the token-signing algorithm (sha1 or sha256). However, this feature in not available yet for SHA-512.

Given the situation we'd like to invite you to leave feedback to our related team where the engineers cherish your valuable ideas much Azure Community ,As you know many products' services have been designed based on customers' constructive ideas and comments.

Your kind understanding is highly appreciated. Thank you for your cooperation.

Sincerely,

Anesu | Microsoft Community Moderator

****Note: In the event that you're unable to reply to this thread, please ensure that your Email address is verified in the Community Website by clicking on Your Account Name > "My Profile" > "Edit Profile" > Add your Email Address > tick "Receive email notifications" checkbox > click on "Save".****

* Beware of Scammers posting fake Support Numbers here.

1 person found this reply helpful · **Was this reply helpful?** Yes No

IT-Pro79

Replied on October 24, 2022

Report abuse

↳ In reply to Anesu Mwamuka MSFT's post on July 28, 2022

Sorry Anesu, you totally misunderstood the OP's question/point.

This has absolutely nothing to do with AD FS lest AD FS Token Signing! The generation of a TOTP-seed-device-relationship of an MFA/2FA service/product like LinOTP and then the generation of an usually 6-digit code in an authenticator app is nothing AD FS concerns itself with. AD FS via authentication extensions can offer text input to the user and forward such a code to an MFA/2FA backend system but the generation and verification of such a TOTP token code according to RFC 6238 is nothing AD FS concerns itself with. The token signing and encryption you are talking about belongs to the OAuth process between browser, IdP (AD FS) and the respective website for securely transmitting information and claims in the form of JWT tokens in JSON form. This is a step down AFTER the user has authenticated by whatever primary and secondary means and is totally unrelated to an OTP code from an MFA system.

@Gary: No, it seems as of October 24th 2022 there is still no support. I have Android 10 with current app updates but when using an SHA-256-based TOTP code on my web server system it fails although the QR code scanned correctly and code is shown in the MS Authenticator app. So I assume MS Auth App is always interpreting the seed as a Google Authenticator conformant code (which is 6-digit/30 sec/SHA-1). To be fair, I'm currently researching this, SHA-1 might not be that of a problem in the TOTP because the situation is very different than regular use of SHA-1 like in encryption or digital signing. TOTP bases on HOTP which is defined as SHA-1 and RFC 6238 states that SHA-256/SHA-512 *may* be used but this is no *must use*. Sadly I'm no cryptography expert but looking around I can somewhat understand why most aren't in panic mode yet. It also seems to be much more important that the MFA/2FA backend handles invalid input with a proper sliding window and re-generation to make any brute-force attack infeasible.
Since I have a myriad of services from Twitter over Twitch to MS on my MS Auth App makes me assume the the majority is currently still using SHA-1 or they have some clever failback mechanisms in the backend. So if you're forced to fulfill some compliance standards you can only go the route to use another app or two apps. Though I'd really love to see some cryptoexperts statement because I have a feeling that all those auditors just react on SHA-1 as a red flag (and I also would regarding web server certificates et al) and not out of sound cryptographic reasoning for that particular use case.
You may find these postings helpful especially that from Adrian Ho:

https://www.quora.com/Why-is-the-SHA1-algorithm-still-being-used-with-2FA-codes-instead-of-SHA2?share=1

Regards, Markus

MCITP Windows Server 2008 (R2) Enterprise Administrator

11 people found this reply helpful · Was this reply helpful? Yes No