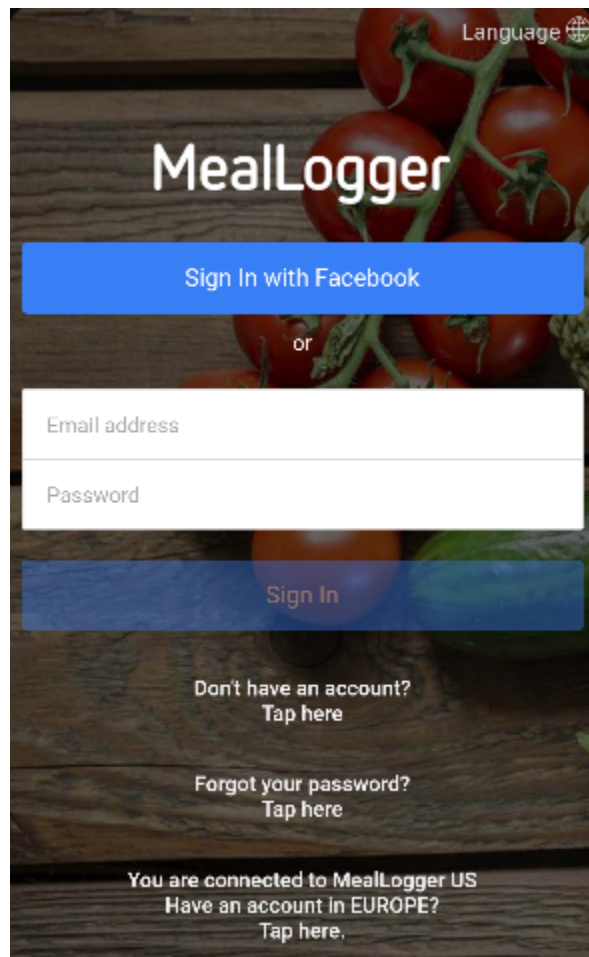


Module 11 Assignment

MSTG-AUTH-1

For authentication purposes, there are two main ways of logging into the Meallogger application: standard login and Facebook sign-in (shown below). Interestingly, the Facebook sign-in functionality seems to not work, at least in the emulator. Other notable functions include creating an account and password retrieval. Since Meallogger is an application that only has functionality following login, authentication is in fact required for access to remote resources. Thus, the Meallogger application passes this standard.

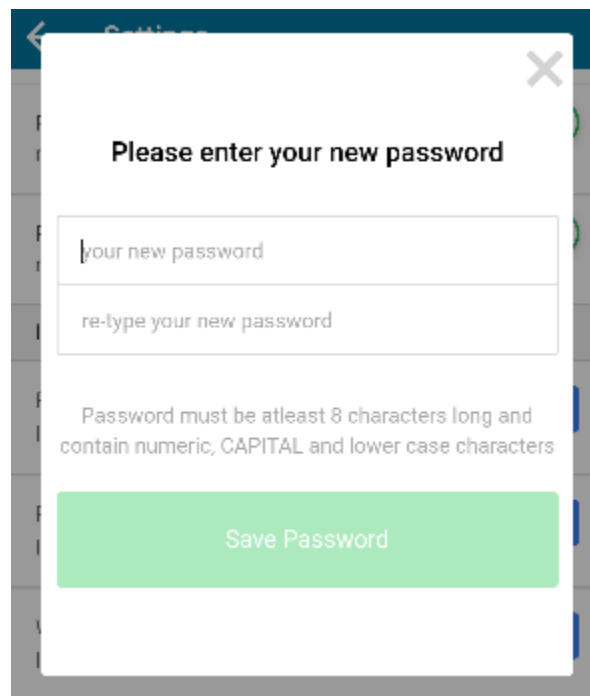


MSTG-AUTH-4

For this standard, the Meallogger application does have ways of terminating a session. If the user chooses to logout from within the application, then they will exit the session and return to the login page. From the login page, the user is not able to access any features of the app without logging in again. However, it is important to note that there is no other way to terminate a session. Simply minimizing or exiting the application does not seem to end the current session, but this is a common feature among many modern applications and does not pose an immediate problem. Since the user cannot access any remote or local information from the application without being logged in, the Meallogger application ultimately passes this standard.

MSTG-AUTH-5

Regarding this standard, the Meallogger application does enforce a password policy for authentication purposes. While OWASP recommends a minimum password length of 10 characters, 8 characters is not too far off. Furthermore, in relation to password complexity, the password policy does meet three out of four rules provided by OWASP, namely: at least one uppercase character, one lowercase character, and one digit. For these reasons, the Meallogger application ultimately passes this standard.



Settings

Please enter your new password

your new password

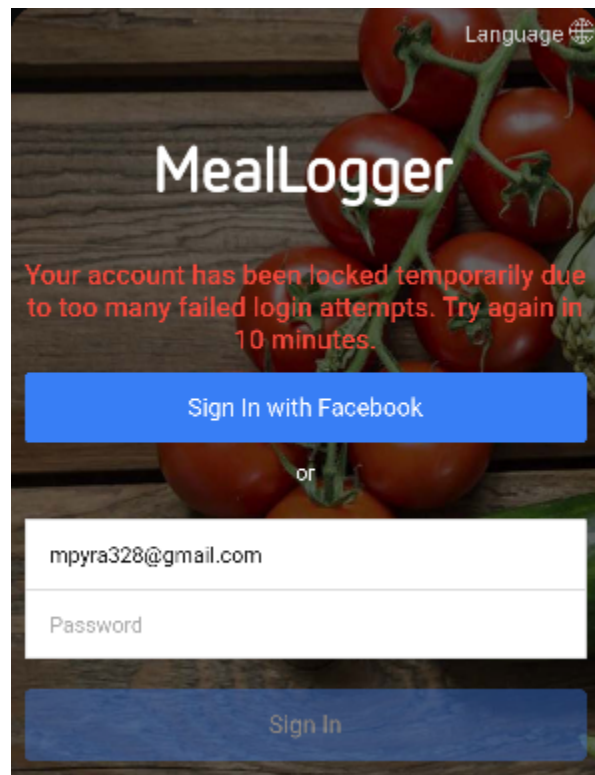
re-type your new password

Password must be atleast 8 characters long and contain numeric, CAPITAL and lower case characters

Save Password

MSTG-AUTH-6

For this standard, the application does in fact have login throttling. After five failed login attempts, the account gets locked for 10 minutes (shown below). As expected, by submitting the correct credentials, the account is still locked within the 10 minute duration. After the 10 minutes are over, his number doesn't increase, but rather resets after a successful login followed by five failed logins. Overall, since the Meallogger application implements a mechanism against an excessive number of credential submissions, the app passes this standard.



MSTG-AUTH-7

For this standard, there seems to be no termination of the application after any amount of time. Leaving the application open over several hours yielded no results, and the application functioned normally after that extended period of time. Essentially, the access tokens never expired and the session was never invalidated at any point. For these reasons, the Meallogger application fails this standard.

MSTG-AUTH-9

For this standard, the Meallogger application has no implementation of two-factor authentication. More specifically, past the first factor of a password check, there are no signs of a one-time password system or hardware/software tokens. Thus, the Meallogger application fails this standard.

MSTG-AUTH-10

Regarding this standard, the Meallogger application has no implementation of step-up authentication for sensitive transactions. However, there are no in-app purchases and no way to link payment/bank account information, so the Meallogger application passes this standard by default.

MSTG-AUTH-11

For this standard, it is clear that the Meallogger application makes no attempt to inform users of activities with their account. Furthermore, the application has no features regarding a device list or configuration logs. Even when changing the password within the application, there is no email sent to the user; the only communication that the app makes with the user is when resetting their password from the login screen. For these reasons, the Meallogger application ultimately fails this standard.

MSTG-CRYPTO-1

Regarding hardcoded secrets, the MobSF report notes 4 possible secrets, all related to Facebook's device authentication instructions (shown below). Interestingly, by looking closer at the source code in jadx, these instances are not apparent, and the files themselves do not contain such information.

HARDCODED SECRETS

POSSIBLE SECRETS
"com_facebook_device_auth_instructions" : "facebook.com/deviceXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
"com_facebook_device_auth_instructions" : "XXfacebook.com/device>XXXXXXXXXXXX"
"com_facebook_device_auth_instructions" : "XXXfacebook.com/deviceXXXXXXXXXX"

POSSIBLE SECRETS
"com_facebook_device_auth_instructions" : "000facebook.com/device0000000000000000"

However, MobSF further notes that other files outside of Facebook authentication also contain hardcoded information. Thus, the combination of these two findings means that the Meallogger application ultimately fails this standard.

Files may contain hardcoded sensitive information like usernames, passwords, keys etc.	warning	CWE: CWE-312: Cleartext Storage of Sensitive Information OWASP Top 10: M9: Reverse Engineering OWASP MASVS: MSTG-STORAGE-14	bolts/MeasurementEvent.java com/adobe/phonegap/push/GCMIntentService.java com/adobe/phonegap/push/PushConstants.java com/bumptech/glide/load/engine/EngineKey.java com/desmond/squarecamera/CameraFragment.java com/desmond/squarecamera/EditSavePhotoFragment.java io/intercom/com/bumptech/glide/load/engine/EngineKey.java rx/internal/schedulers/NewThreadWorker.java
--	---------	---	--

MSTG-CRYPTO-2, 3, 4

In terms of cryptography, the Meallogger application actually meets and follows the majority of encryption guidelines provided by OWASP. The MobSF report only notes one major problem, which relates to outdated hashing services. However, this issue is a very large one, as insufficient hashing can lead to other, bigger problems along the line. For this reason, the Meallogger application fails this standard.

FCS_COP.1.1(2)	Selection-Based Security Functional Requirements	Cryptographic Operation - Hashing	The application perform cryptographic hashing services not in accordance with FCS_COP.1.1(2) and uses the cryptographic algorithm RC2/RC4/MD4/MD5.
----------------	--	-----------------------------------	--

MSTG-CRYPTO-6

The MobSF report clearly notes that the application uses an insecure Random Number Generator (shown below). Thus, the Meallogger application fails this standard.

The App uses an insecure Random Number Generator.	warning	Random Values OWASP Top 10: M5: Insufficient Cryptography OWASP MASVS: MSTG-CRYPTO-6	com/adobe/phonegap/push/GCMIntentService.java com/plugin/datepickers/DatePickerPlugin.java
---	---------	--	---