1. **High/Medium Risk -** We noted that the allowed characters for Names as described on page 22 of “Documentatie 112 NL App - PEN Test.docx” is not effectively enforced. We were able to use the name “Ruben!” and “KPMG@”.
   Additionally, if enforced effectively, we do not see a reason to allow the characters ‘,’ ‘.’ in the name of the app user. Please note that the name is not communicated to the backend in the current test setup. It might be that the input validation occurs on the backend. Still, we recommend to enforce this on the client side as well.
2. **Medium Risk -** We noted limited documentation in relation to the security of the application and the accompanying backend. The documentation received is focused around the functional working of the application.
   What is missing currently are the mitigating controls from a documentation perspective, given our risk analysis meeting we had before. In this meeting the highest risk identified was ‘script kiddies’. What controls are in place from a security perspective to mitigate this risk? Same goes for example about file upload; what data types are allowed? Procedures, descriptions and controls in relation to security is noted in the delivered documentation.
3. **Medium Risk -** The Android Manifest shows that ‘android:usesCleartextTraffic’ is set to True. This means that the app intends to use cleartext network traffic protocols. From the current setup we are not able to identify what protocols that will be, put most probably HTTP.
4. **Medium Risk (Under investigation still) -** We noted that the Android application exports activities that can be activated, whilst not being enrolled via the telephone number validation check. It seems that this would allow an attacker to register a fake telephone number, since the validation activity can be circumvented. Note that this observation is still under investigation.
5. **Medium Risk** - From the documentation received, it seems that internal communication is unencrypted. For example, the communication between the ‘AML POST Service’ and the ‘Meldkamer Service Bus’ uses HTTP as well as between ‘GVS: SOAP XML’ and ‘Meldkamer Service Bus’.
6. **Medium Risk (Under investigation still) -** The iOS application includes references towards URL’s that are not mentioned in the provided documentation. These URL’s are http://royapps.com/ and http://patrickpiemonte.com. Currently it not yet clear what these domains are used for by the application. Additionally, the ‘roysapps’ domain is not registered yet and can thus pose a security threat in the future.
7. **Low Risk -** From the documentation received, it seems that the **‘**appSessieID’ is communicated via a GET request parameter during a chat between client and server. GET request parameters can be stored on multiple locations. This includes browsers, webserver logs, referrer-headers and reverse proxies. A malicious user with access to one of those sources can abuse the ID. Note that ID abuse will further be investigated during the ‘ketentest’.
8. **Low Risk** - Acceptance API is only blocked on the application layer. On the network layer, the webserver ports are still enabled. Generally speaking, servers operating in an test or acceptance environment are less hardened.
   For example, we often see less protective measures in place (e.g. WAF) at acceptance environments compared to production environments. Within the timeframe of this test we were not able to identify weaknesses due to this.