KPMG

Security test Candidate Corona apps

Ministry of Health, Welfare and Sport
A2000020142

Final report

—

14 May 2020

Contents

KPMG Advisory N.V.
Postbus 74500
1070 DB Amsterdam

Laan van Langerhuize 1 1186 DS
Amstelveen Telephone +31 (0)20 656
7890

Amstelveen, 14 May 2020

To the Ministerial Department of Health, Welfare and Sport

Dear Mr Roozendaal, Dear Ron,

In accordance with our proposal of 17 April 2020, reference A2000020142, we hereby send you our report on the work performed by us.

The purpose of the instruction given to us was to give you an insight into the extent to which sufficient attention has been paid to the security and reliability of the solutions as shown in the Appathon of 18 and 19 April 2020, consisting of one or more apps, back-end systems and the communication with these systems.

Nature of the engagement

The nature of the work means that we have not conducted an audit, review engagement or other assurance engagement. Hence, no assurance regarding the fairness of financial or other information can be derived from this report.

Our advice in this report is based only on the results of the agreed work and the results thereof. If we had carried out additional work, or had performed an audit, review or assurance engagement, other topics eligible for reporting could have been identified. We will not adjust our report in the event of future changes, legislative amendments or revisions in the legal and administrative interpretations of legislation and regulations.

Responsibility of the Ministerial Department of Health, Welfare and Sport

For the sake of completeness, we note that you are responsible for the correctness and completeness of the information made available to us within the context of the above work. We accept no responsibility for the quality, correctness or completeness of the information supplied to us.

Distribution of the report

The report is exclusively intended for you, the client. Without our express and prior written permission, you are not permitted to use this report, or parts thereof, for other purposes, to make it public and/or to disclose it to third parties. KPMG accepts no liability for the use of this report other than for which it was drawn up and made available to you, the client.

We would like to thank you for the open and constructive cooperation in the performance of our work and while preparing our report.

If you have any further questions, please do not hesitate to contact us.

Yours sincerely,

KPMG Advisory N.V.

Hermans

Ing. J.A.M. Hermans RE

Partner

Backgrounds of the security assessment

On 11 April 2020, the Ministry of Health, Welfare and Sport (hereinafter referred to as VWS), through the Invitation for smart digital solutions Corona (hereinafter referred to as "the Invitation") invited the market to submit proposals for the following four subjects:

1. Smart digital solutions that can contribute to source and contact tracing;

2. Smart digital solutions that can contribute to self-monitoring and remote guidance;

3. Other digital solutions that can contribute to the transition strategy;

4. Preconditions under which such solutions can be implemented.

With regard to part 1, seven parties were asked to participate in a so-called Appathon (which takes/took place on 18 and 19 April 2020), to allow VWS to determine whether and to what extent the solution meets the principles stated in the Invitation while maintaining transparency.

Parallel to this Appathon, VWS asked KPMG to analyse the solutions offered by the seven parties, to give an insight into the extent to which sufficient attention has been paid to the security and reliability of these solutions, consisting of one or more apps, back-end systems and the communication with these systems.

The security assessment that has been carried out per supplier consists of two parts, namely:

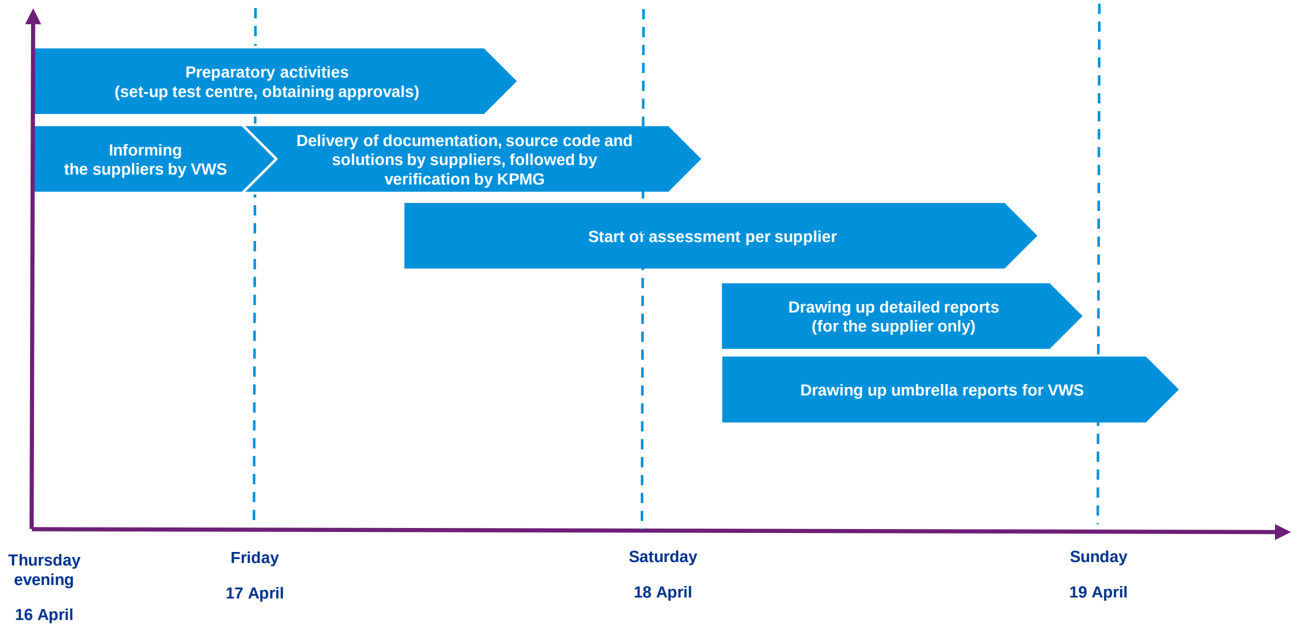
1. A limited initial penetration test.

2. A source code quick scan.

During the security assessment, a number of research questions were formulated for both parts, supported by a structured security assessment approach, to ensure comparability of the aspects to be analysed per supplier. These research questions are further specified in this report.

Backgrounds of the security assessment

The security assessment was characterised by the very short timeline, namely two days. The figure below shows the activities that took place during these two days.


Preparatory activities
(set-up test centre, obtaining approvals)
Informing
the suppliers by VWS
Delivery of documentation, source code and solutions by suppliers, followed by verification by KPMG
Start of assessment per supplier
Drawing up detailed reports
(for the supplier only)
Drawing up umbrella reports for VWS
Thursday evening
16 April
Friday
17 April
Saturday
18 April
Sunday
19 April

The findings from our security assessment will be presented in this report, grouped in the following categories:

- General findings
- Findings per research question

Research questions limited initial penetration test

KPMG applied the following research questions when conducting the limited initial penetration test. These research questions are underpinned on the test methodology used and consistently performed for all applications within scope, where possible.

Highest priority

1. Can a malicious person acquire sensitive information through unauthorised communication with the underlying infrastructure of application X?
2. Can a malicious person compromise or severely disrupt the underlying infrastructure of application X by injecting malicious code, as a result of which the availability, integrity and confidentiality of the system and data cannot be guaranteed?
3. Can a malicious person enter incorrect data or delete correct data by exploiting the underlying infrastructure of application X, as a result of which the data in the underlying infrastructure becomes less reliable, meaning it is no longer possible to properly determine which users have COVID19 infection and which do not?
4. Can a malicious person intercept and/or exploit the communication between application X and the underlying infrastructure (the channel itself or, for example, via the API)? This scenario carries a higher risk when used in public hotspots such as a guest Wi-Fi and/or public hotspots.
5. Can a malicious person obtain sensitive data from the storage space of application X on the end user's mobile device? This scenario can occur if the mobile device is vulnerable and application X stores sensitive data on the device (consciously or unconsciously). This can affect the privacy of the user, as well as possible other users (if data is also stored about them because these users were nearby, for example).
6. Can a malicious person compromise the communication between application X and the underlying infrastructure by exploiting channels other than the primary channel (the expected API interface)? Examples include additional channels such as e-mail, text messages or other TCP/UDP network interfaces.

Lowest priority

Research questions source code quick scan

KPMG applied the following research questions when carrying out the source code quick scan. These research questions were converted into a structured process in which automated and manual investigations are consistently performed for all applications within scope, where possible.

7. What is the general picture of the (technical) quality of the source code of the application?

8. Does the automatic tooling in the analysis reveal any real vulnerabilities regarding the reliability and security in the source code?

9. Does the source code contain other data outputs (including logging) that are not defined in the design?

10. Does the supplied software (both front-end and back-end) depend on external libraries for a proper operation? If so, are these libraries current, are they maintained and/or do they contain known vulnerabilities with regard to reliability and security?

11. Has the source code been set up in line with our expectations based on the technical and functional documentation (for example, have described privacy mechanisms been implemented, does the scope fit the description, etc.)?

Limitations of the security assessment

Partial security assessment with limited depth

- During our security assessment, we have analysed the apps used, the back-end and the communication between the components at limited depth (due to the short timelines).
- In addition, we would like to point out that the manner of implementation of the entire concept and compliance with the correct management and security measures are decisive for being able to meet the security and privacy requirements required for these concepts. We therefore would like to emphasise that a comprehensive security and privacy investigation focused on the entire concept consisting of organisation, processes and technology is important before proceeding with full implementation.

Guaranteeing the anonymity of the suppliers and intended use

- The results of the examination are aimed at providing general insights to be used by the committee when questioning the suppliers. The results of the assessment are not intended as a selection tool. This report does not contain references to specific suppliers.

Given the very short completion time of the security assessment, we would like to point out the following limitations therein:

- A limited number of detailed test activities have not been performed, for reasons that include the late availability of the source code, the required test environment and the tools to be used for this test. Consequently, not all issues may have been identified. In our reports, situations in which no detailed testing activities have taken place have been marked as such.
- For a limited number of findings, no additional in-depth security assessment has been possible, as a result of which some of these findings could be described as “false-positives” after further investigation.
- Detailed findings of the penetration tests and source code reviews have initially not been validated with the supplier in detail ("hearing both sides") in accordance with the agreements with the client and suppliers. After publication of the first version of the initial version of the final report we have, upon request of the client, performed a validation of the detailed findings of the penetration tests and source code reviews. Based on this validation with the suppliers we have made minor adjustments to the detailed findings, to strengthen the factual observations. This has not provided cause to adjust the content of the general findings and observations in this report.
- We carried out the security assessment based on the documentation, source code, apps and back-end systems provided by the supplier. We have not been able to determine whether the supplied documentation and source code are complete. In addition, adjustments may have been made to the apps, back-ends or source code during our security assessment. These changes have not been included in the results of our security assessment.

Limitations of the security assessment

Limitations regarding the limited initial penetration tests:

- Testing iOS apps carries certain inherent limitations. For example, Apple has secured its operating system (iOS) in such a way that we cannot test everything (depending on the iOS version). Furthermore, we can only test to a limited extent and therefore not cover all versions/equipment combinations.
- Mobile devices have certain inherent vulnerabilities that are not investigated when conducting a penetration test on a mobile application. For example, certain unique identifiers of identified Bluetooth devices may be stored in the log files of mobile devices. Such inherent vulnerabilities are not investigated during an application penetration test.
- A security test can only demonstrate whether it is possible to breach existing security measures. It cannot be stated with certainty that implemented security measures cannot be breached. An attacker with an unlimited budget, knowledge and time will almost always succeed in breaking through security. The activities of KPMG further involve a time-related effort, which means that not all possible weaknesses can be tested exhaustively.
- A security test is conducted based on the weaknesses and attack techniques known to KPMG at that time. The result of a security test is therefore always a snapshot in time of the knowledge of weaknesses and attack techniques, as well as of the security measures taken.

Limitations regarding the source code quick scan:

- The source code quick scan is limited to the source code provided by the suppliers. Limited inquiries have been made as to whether the software was complete and everything had been supplied.
- All apps use additional standard software components (such as operating systems, database and web server software), network facilities and network protocols. Vulnerabilities in these components have not been mapped out.
- This concerns a time-boxed effort and the tooling used is not the same for every software platform. This means that not all issues may have been identified and that not all findings could be fully validated.

KPMG

General findings and observations

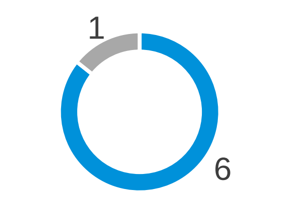
General findings - participating suppliers

Suppliers included in the security assessment

On the evening of 16 April 2020, seven participants of the Appathon were informed by VWS about their participation and the information needs of KPMG. KPMG has conducted an intake with the various suppliers and validated whether the required information (consisting of technical documentation, source code, app(s) and other required information) was provided by the supplier.

Based on the validation in which it was established that the necessary information could not be provided, it was decided to exclude one of the suppliers from further testing activities.

Included suppliers



■ Information received

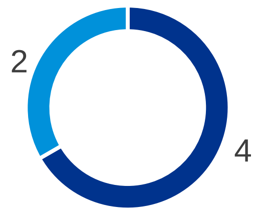
■ No information received

Therefore, no findings with regard to this supplier are presented in the remainder of this report.

Available apps related to the Operating System

During the validation of the data, it was further found that not all suppliers currently have an app for both the Android and iOS operating systems.

Currently available apps



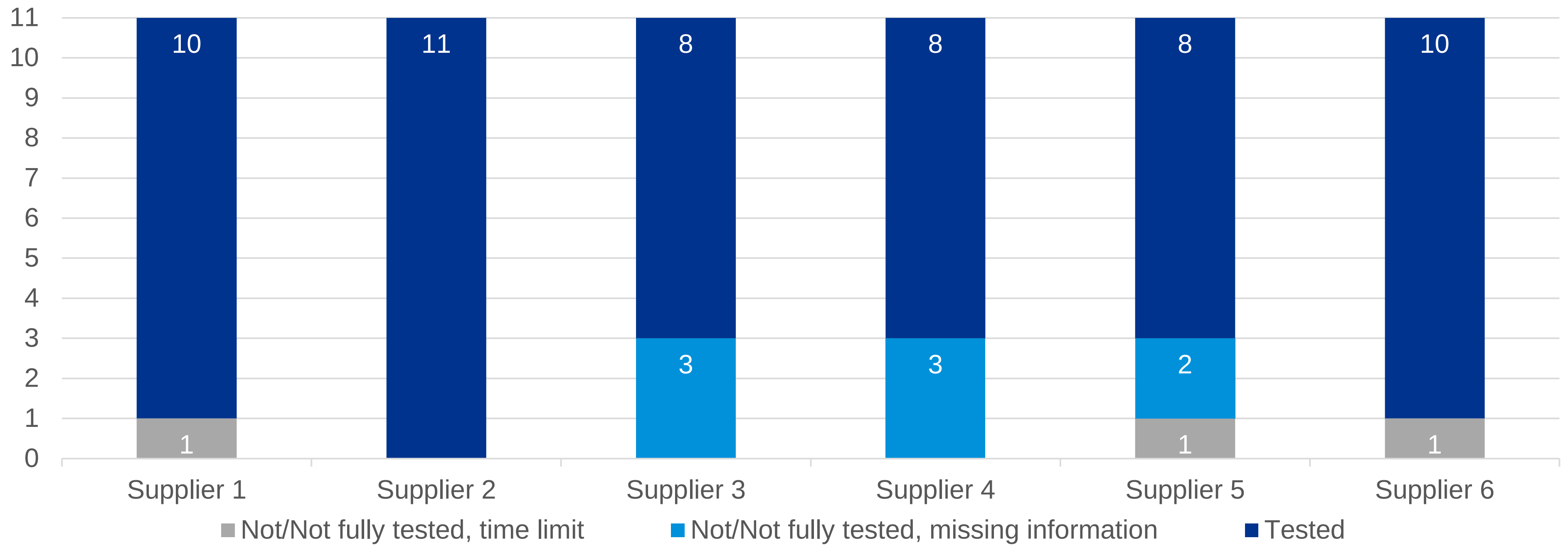
■ Both Android and iOS

There were no suppliers that only developed an iOS-based app.

General findings - research questions

As already explained in this report, we carried out the security assessment based on 11 research questions. During the security assessment we, for a number of suppliers, were unable to perform all test activities aimed at answering the research questions, due to the unavailability of all relevant information, the unavailability of the apps and partly due to the partial unavailability of test infrastructure and test tools, as well as a lack of time of the testers.

The figure below shows which research questions we have or have not been able to answer, per supplier.



** The numbering of suppliers used in this report does NOT correspond to the list of suppliers published by VWS*

General observations

General observations of the limited initial penetration tests

- In general, the developers of the applications have not applied secure coding principles, as a result of which well-known and therefore expected security measures have not been implemented. This also applies to the underlying infrastructure (misconfigurations, outdated software, exposed management interfaces, etc.). This creates (serious) vulnerabilities that could easily have been prevented.

- In addition to the above, we would like to point out specifically that several apps use hard-coded passwords, which are therefore clear text included as such in the legible source code. It turned out that these passwords enabled access to databases and/or the underlying infrastructure (including datasets outside the domain of the COVID application to be tested).

- The underlying infrastructure (including the API) can often be exploited without the necessary identification/authentication. In those instances where security is in place, it is often easy to circumvent due to a vulnerable implementation, the failure to validate, for example, security certificates or validating these sufficiently, or the use of self-signed certificates. In many instances, this leads to access to confidential data and/or the possibility to submit incorrect data.

- The design of the applications is not always based on the principle of storing as little sensitive data on the end user's phone as possible. In addition, data that is stored is stored without encryption.

- For the provided track and trace functionality Android phones require a relatively large number of access permissions, including some that could be seen as strange requests, such as access to microphone or photos. The desired permissions can be explained from the desired functionality, but the end user is likely to distrust and not accept this.

General observations - continued

General observations of the source code quick scan

- The supplied software features a considerable diversity in the degree to which it is "ready for use": however, none of the systems is "finished". This also prevents a proper comparative assessment of the findings; for example, where a mediocre hashing algorithm is used in one application component, this functionality is altogether lacking in other solutions.

- Technical documentation about the components has been largely absent. The source code investigators spent a lot of time figuring out how the components were put together, how they had to be compiled (if necessary) and how they functioned.

- General measures to promote code quality have been found to only a limited extent. We have detected deviations from coding standards and magic literals (an anti-pattern using numbers and strings directly in the code). Furthermore, the source codes were found to have few unit tests and limited inline documentation.

Findings per research question

In response to the research questions on the following pages we have provided findings that are applicable to one or multiple of the suppliers.

Research question 1

1. Can a malicious person acquire sensitive information through unauthorised communication with the underlying infrastructure of application X?

Supplier	Test
1	✓
2	✓
3	—
4	—
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- In general, the developers of the applications have not applied secure coding principles, as a result of which well-known and therefore expected security measures have not been implemented. This also applies to the underlying infrastructure (misconfigurations, outdated software, exposed management interfaces, etc.). This creates (serious) vulnerabilities that could easily have been prevented.
- The use of hard-coded passwords that are therefore clear text and included as such in the legible source code. This allowed access to sensitive data.
- The underlying infrastructure (including the API) can often be exploited without the necessary identification/authentication. In those instances where security is in place, it is often easy to circumvent due to a vulnerable implementation, the failure to validate, for example, security certificates or validating these sufficiently, or the use of self-signed certificates. In many instances, this leads to access to confidential data and/or the possibility to enter incorrect data.
- A malicious party can manipulate the underlying infrastructure because there is no or very limited control of the data that is sent to/from this infrastructure.

Research question 2

2. Can a malicious person compromise or severely disrupt the underlying infrastructure of application X by injecting a malicious code, as a result of which the availability, integrity and confidentiality of the system and data cannot be guaranteed?

Supplier	Test
1	X
2	✓
3	✓
4	—
5	—
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- The underlying infrastructure contains various (serious) vulnerabilities that can be exploited, giving an attacker access to the data (ability to read and/or manipulate) and also possible access to the underlying operating system. These include:
  - Unsafe configuration of operating systems, databases and (application) services
  - Outdated software
  - Management interfaces exposed towards the Internet (admin portals, databases, etc.)
- Also, a malicious party can manipulate the underlying infrastructure because there is no or very limited control of the data sanitisation (e.g. user input / system output validation).

Research question 3

3. Can a malicious person enter incorrect data or delete correct data by exploiting the underlying infrastructure of application X, as a result of which the data in the underlying infrastructure becomes less reliable, meaning it is no longer possible to properly determine which users have COVID19 infection and which do not?

Supplier	Test
1	✓
2	✓
3	—
4	—
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- The underlying infrastructure (including the API) can often be exploited without the necessary identification/authentication. In those instances where security is in place, it is often easy to circumvent due to a vulnerable implementation or the failure to validate, for example, security certificates or validating these sufficiently. This leads to access to confidential data and/or the possibility to enter incorrect data.
- In general, the developers of the applications have not applied secure coding principles, as a result of which well-known and therefore expected security measures have not been implemented. This also applies to the underlying infrastructure (misconfigurations, outdated software, exposed management interfaces, etc.). This creates (serious) vulnerabilities that could easily have been prevented.

Research question 4

4. Can a malicious person intercept and/or exploit the communication between application X and the underlying infrastructure (the channel itself or, for example, via the API)? This scenario carries a higher risk when used in public hotspots such as a guest Wi-Fi and/or public hotspots.

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	X
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- The application and the underlying infrastructure do not check the identification and authentication of the sender/receiver (not even in the case where certificates are used, because so-called certificate pinning is not applied) or check this to a limited extent only. This allows an attacker easy access to the communication through a man-in-the-middle attack. These kinds of attacks are easy to set up at public hotspots.
- Weak encryption is used in the communication between the app and the underlying infrastructure.

Research question 5

5. Can a malicious person obtain sensitive data from the storage space of application X on the end user's mobile device? This scenario can occur if the mobile device is vulnerable and application X stores sensitive data on the device (consciously or unconsciously). This can affect the privacy of the user, as well as possible other users (if data is also stored about them because these users were nearby, for example).

Supplier	Test
1	✓
2	✓
3	—
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- The design of the applications is not always based on the principle of storing as little sensitive data on the end user's phone as possible. In addition, data that is stored is stored without encryption.
- Also, not all applications are designed/implemented with an adequate password policy.

Research question 6

6. Can a malicious person compromise the communication between application X and the underlying infrastructure by exploiting channels other than the primary channel (the expected API interface)? Examples include additional channels such as e-mail, text messaging or other TCP/UDP network interfaces.

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	—
6	X

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- In a few cases, the applications and the underlying infrastructure use additional channels (in addition to the primary channel, the API). Using these additional channels carries a risk of additional vulnerabilities, as well as the risk that another party may correlate the data to groups or even individuals. For example, due to traceability of identification data of the telephone (text messages, Bluetooth, UUID, telephone numbers, etc.) or other information that can be traced/correlated.

Research question 7

7. What is the general picture of the (technical) quality of the source code of the application?

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- The applications are generally small and have only limited technical debt (the expected effort required to implement a solution to the findings).
- Bar one, we were able to analyse the supplied application components with tools, which meant that the investigators could also compile the source code where necessary.
- Still, few measures were found that promote code quality. Consequently, quite a few deviations from the coding standards were found, including magic literals. Also, limited use is made of unit testing. In practically all solutions, “Todo’s” and “Fixme’s” were found in the inline documentation, indicating that the solution is not finished. The difference in technical quality between prototypes and near-mature implementations is clearly noticeable.

Research question 8

8. Does the automatic tooling in the “car wash” reveal any real vulnerabilities regarding the reliability and security in the source code?

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- Vulnerabilities requiring attention have been found in all solutions. The vulnerabilities found in the source code have been documented and reported to the suppliers.
- We further note that in a number of cases, no (or limited) authentication was implemented in the communication with the back-end application and that in some cases, other security measures were missing in the source code as well.

Research question 9

9. Does the source code contain other data outputs (including logging) that are not defined in the design?

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- No data outputs were found that were not expected.
- There is, in particular, location data available in the apps which is sometimes even passed on to the back-end system, but not processed (yet); this can harm users' confidence in the app. In addition, in theory it is possible that the supplier will still process this data in the future without the user being informed thereof. After all, the permission to make the data accessible from the app has already given at the initial installation.
- A point of attention is that in some solutions, the error handling is not properly implemented; this carries the risks that error messages with sensitive data may be displayed on screens, for example.

Research question 10

10. Does the supplied software (both front-end and back-end) depend on external libraries for a proper operation? If so, are these libraries current, are they maintained and/or do they contain known vulnerabilities with regard to reliability and security?

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- Some solutions use frameworks and libraries. One solution uses the latest versions and automated methods (dependency managers) to remain up to date. Other solutions use outdated versions; some of these are also known to contain vulnerabilities.

Research question 11

11. Has the source code been set up in line with our expectations based on the technical and functional documentation (for example, have described privacy mechanisms been implemented, does the scope fit the description, etc.)?

Supplier	Test
1	✓
2	✓
3	✓
4	✓
5	✓
6	✓

✓ Tested

X Not/not fully tested, time limit

— Not/not fully tested, missing information

- In general, very limited documentation has been found; although in general, the rough functionality has been found as understood from the pitches. However, in some of the solutions, not all stated functionality has been implemented yet.
- The expected roll-out of the API for exchanging Bluetooth identifiers by Apple and Google is in some cases the reason for the lack of functionality (as yet).
- Bar one solution, which works with a telephone number, user data is intended to be passed on anonymously. Specific privacy mechanisms have not been found.

Appendices

Scope

as described in our proposal dated 17 April
reference number A2000020142

Scope

Scope of the limited initial penetration test

As for the limited initial penetration test, we apply a scenario-based approach in which we analyse vulnerabilities based on the OWASP Mobile Top 10 (version 2016) and the OWASP Top 10 (version 2017). The scope of the initial penetration test consists of the following three components:

- COVID-19 app. We will analyse the risks 'M2: Insecure Data Storage', and 'M5: Insufficient Cryptography', among other risks, to determine the extent to which sensitive information is stored in an unsafe manner.

- Communication between the app and back-end. We will analyse the risks 'M3: Insecure Communication' and 'M5: Insufficient Cryptography', among other risks. We focus on the scenario that an attacker can access the end user's local network, access the end user's mobile device or act as a man-in-the-middle.

- Back-end interface. We will analyse the risks 'M10:Extraneous Functionality', 'A1: Injection', and 'A3: Sensitive Data Exposure', among other risks. We focus on the unauthorised unlocking of data from the back-end, the unauthorised modification of data in the back-end and making the back-end inaccessible (with the exception of Distributed Denial-of-Service (DDoS) attacks).

Scope of the source code quick scan

The source code quick scan will focus on the source code provided by the supplier and other artefacts, including documentation. In view of the desired timeframe, no inquiries will be made about apparently missing elements; these elements will be reported as a detailed finding. The quality of the underlying software components, such as operating systems, database and web server software, is not part of the scope.

Approach

as described in our proposal dated17 April
reference number A2000020142

Approach

We submit the COVID-19 app, the associated back-end and the communication between the app and the back-end to a ***limited initial penetration test***, in accordance with the white box principle and a maximum given lead time of 24 hours. This means, among other things, that we obtain extensive access to documentation, system configuration and other requested information, in advance. We will not conduct detailed system configuration reviews, but will use this information to efficiently conduct our tests.

We at the same time form a picture of what an attacker could do if this information is not available to him. Based on this information, we determine which scenarios are most relevant to test during the penetration test. Based on these scenarios, we complete the following three phases of our penetration test:

1. *Identification scan phase*: performing several identification scans on the back-end environment. This step provides us with detailed information about which ports and services are active.

2. *Vulnerabilities scan*: we use efficient tools that identify known vulnerabilities in systems. These tools and scans are applied to the back-end environment. Manual scans for any vulnerabilities are also performed, both on an infrastructure and an application level, aimed at the app, the back-end and the communication between the app and the back-end.

3. *Exploitation*: We carry out manual checks to determine whether the vulnerabilities identified in the previous steps are actually present (false positive verification). We further conduct manual tests for vulnerabilities, exploiting them where possible. This way, we can determine the impact of the vulnerabilities.

In the ***source code quick scan***, with the emphasis on Reliability and a maximum given lead time of 24 hours, we will inspect the supplied source code of both the front and (if applicable) the back-end application. Using automatic tooling, we will determine commonly used software metrics such as the number of lines of code (LOC), Complexity and Duplication. Based on the available tooling for the specific development platform, we will obtain findings on Reliability, Security and Maintainability, if possible. With the focus of this research in mind, we will check the *Reliability* and *Security* findings manually.

We will further check whether the software depends on external libraries, whether these libraries are current and whether these libraries are known to have reliability and security issues.

Finally, we will check whether the source code is in line with our expectations based on the technical and functional documentation and whether there are data outputs or interfaces (including log files) other than those specified. As part of our efforts, we will pay attention to whether the described privacy mechanisms have been implemented. If cryptographic algorithms are used (within the supplied software), it will be checked whether these algorithms are frequently used, tested and current.

Ing. J.A.M. Hermans RE

Partner

KPMG Advisory N.V.
Tel: + 31 20 656 8394
Mob: + 31 6 51 366 389 hermans.john@kpmg.nl

ir. R. Heil MSC CISSP GICSP CISA

Partner

KPMG Advisory N.V.
Tel: +31 20 656 8033
Mob: +31 6 51 369 785
heil.ronald@kpmg.nl

Drs. J.M.A.Koedijk CISA CISM

Partner

KPMG Advisory N.V.
Tel: + 31 20 656 8251
Mob: + 31 6 22 903 688
koedijk.joost@kpmg.nl

KPMG on social media

KPMG app