Concordia Institute for Information System Engineering (CIISE)
Concordia University

INSE 6120 Cryptographic Protocols and Network Security

Project Report

Privacy and Security Evaluation of Covid-19 related Apps and Web Services

Submitted to:

Professor Dr. Mohammad Mannan

Submitted By:

Student Name	Student ID
Deepinder Singh Sidhu	40109996
Harjant Singh	40110797
Omer Mujtaba	40137495

1. Major Findings:

1.1. Applications:

- **TousAntiCovid:** CAPTCHA API KEY is exposed. If this key is used for multiple purposes, then the attacker can misuse this key.
- **COVA Punjab**: Labour Key is hardcoded. Hard-coded cryptographic key allows malicious users to gain access to the account.
- **Hamagen**: Key hardcoded: "transistorsoft_key". A key probably related to transistor software which is a module for background location tracking and geofencing and is aware of battery and motion of the device.
- **Ito:** This is the only application which did not use sha1 with RSA and did not sign the APK with debug certificate.
- **PeduliLindungi:** Only this application asked for permission - android.permission.ACCESS_MEDIA_LOCATION
- **Rakning C-19:** Only this application asked for permission- android.permission.AUTHENTICATE_ACCOUNTS and implemented Microsoft Visual Studio App Center Analytics and Microsoft Visual Studio App Center Crashes and Bugsnag.
- **DOCANDU Covid Checker:** Only this application asked for permission- android.permission.RECORD_AUDIO.
- None of the applications asked the permission- android.permission.GET_TASKS
- **SM-COVID-19:** Only this application had the vulnerability CWE-919 and implemented AltBecon and OneSignal.
- **COVA-Punjab:** Only this application implemented the tracker Google Admob.
- COVID Alert, Smittestop, Koronavilkku, COVID Tracker Ireland, Immuni,Beat Covid Gibraltar: Do not leak any private data related to the user of the device.

1.2 Websites:

- Only 2/68 websites got A+ and 4/68 received an F grade from Immuniweb.
- Only 1/68 websites got A+ and 38/68 received an F grade from Mozilla observatory.
- Only one website named **https://ncov.moh.gov.vn** was found with an invalid certificate.
- Only one website named **https://covid.gobusiness.gov.sg** implemented a CSP header with unsafe sources inside style-src.
- Only one website named **https://covid-19.chinadaily.com.cn** had content visible via cross-origin resource sharing file or headers.
- No website had implemented HTTP public key pinning header which is also deprecated.
- Only 2 websites named **https://covid.gobusiness.gov.sg** and **https://www.covid19.gov.ph** implemented Referrer-Policy header that were set to no-referrer, same-origin, strict-origin or strict-origin-when-cross-origin.
- **Subresource integrity** header was not implemented in any website.

2. Applications tested:

We have analysed 22 mobile applications in total, both android and iOS based. The applications chosen are from various geographical locations in which most of the apps are provided by the government of a particular country for the pandemic. The names of the applications are as follows:

1. Coronavirus Australia
2. COVIDSafe
3. Stop Corona
4. COVIDAlert
5. Coronalert
6. eRouska
7. smittestop
8. Korona Vilkku
9. TousAnticovid
10. ito
11. Corona-Warn-App
12. Beat Covid Gibraltar
13. DOCANDU Covid Checker
14. GH COVID-19 Tracker App
15. VirusRadar
16. Ranking C-19
17. PeduliLindungi
18. COVA Punjab
19. COVID Tracker Ireland
20. Hamagen
21. Immuni
22. SM-COVID-19

3. Tool Used:

3.1. MobSF:

Mobile Security Framework or MobSF is a framework for Android/iOS/Windows based application testing which includes pen testing, malware analysis and security assessment. The tool helps in performing both Static and dynamic analysis.

3.1.1. Methodology:

We have used MobSF in macOS. The installation process takes a few minutes via terminal and then you can host the framework on the desired port. Before installation, there are certain requirements to be fulfilled on the mac which is clearly explained in the documentation.

For the static analysis, the APK format of the application is provided and it gives us the results of the scanning.

For the dynamic analysis, we used Genymotion as an emulator to test the network security in the run time environment. Dynamic analysis does not work on virtual machines.

3.1.2. Results: The results of the MobSF Scan are as follows: (See Appendix 1 for the detailed table of results).

I. Signer Certificate: A signer certificate is a certificate with the public key that is linked with some personal certificate. Its purpose is to verify personal certificates.

- **v1 signature scheme-** It is used to sign JAR of the APK. This was found in 100% of the applications.
- **is signed with SHA1withRSA-** In this RSA, a public key cryptosystem is combined with SHA1, a hash function, to sign the certificate. This was found in 95.4% of the applications.
- **signed with a debug certificate-** The debug certificate is used to sign apps that are being developed and tested. SHA-1 is used to create the certificate. This was found in 95.4% of the applications.

II. Permissions: Permissions allow an application to perform certain operations in an operating environment. They are built upon a central design keeping android security architecture in mind. By default, no application has permission to perform any operations that can damage other applications existing in the same environment, the operating system itself, or the user.

- **android.permission.ACCESS_MEDIA_LOCATION:** Permits an application to access any geographic locations using the user's shared media. This type of permission is considered dangerous. This permission was found in 4.5% of the applications.
- **android.permission.AUTHENTICATE_ACCOUNTS:** Allows an application to create accounts and set passwords. This permission was found in 4.5% of the applications.
- **android.permission.ACTIVITY_RECOGNITION:** Allows an app to receive updates of your activity level and has direct access to personal contact card. This permission was found in 9.09% of the applications.
- **android.permission.ACCESS_BACKGROUND_LOCATION**: Application is allowed to access location in the background. This permission is considered dangerous. This permission was found in 22.7% of the applications.
- **android.permission.CAMERA:** Application is allowed to access the camera and is considered dangerous. This permission was found in 31.8% of the applications.
- **android.permission.ACCESS_COARSE_LOCATION:** This is a dangerous permission that allows an application to access approximate location. This permission was found in 45.4% of the applications.
- **android.permission.READ_EXTERNAL_STORAGE:** A dangerous permission that allows an application to read from external storage. This permission was found in 18.1% of the applications.
- **android.permission.WRITE_EXTERNAL_STORAGE:** A dangerous permission that allows an application to write on external storage. This permission was found in 4.5% of the applications.

- **android.permission.RECORD_AUDIO:** Another dangerous permission to record audio by an application. This permission was found in 4.5% of the applications.
- **android.permission.GET_TASKS:** This permission is deprecated. This permission was found in 0% of the applications.
- **android.permission.ACCESS_FINE_LOCATION:** Allows an applications to access the fine location of the device and is considered dangerous. This permission was found in 45.4% of the applications.

III. **Vulnerabilities-** Vulnerability is a weakness inside an application due to its design that can make the system insecure to the attacks and can reveal the private data of a user.

- **CWE-919:** A weakness related to mobile applications. This vulnerability was found in 4.5% of the applications.
- **CWE-532**: This vulnerability is related to insertion of sensitive information into log file. This vulnerability was found in 90.9% of the applications.
- **CWE-330**: This security vulnerability relates to random numbers when random numbers are predictable and is critical from a security point of view. This vulnerability was found in 86.3% of the applications.
- **CWE-312:** Sensitive information is stored in cleartext in a resource that is accessible to third parties or an attacker. This vulnerability was found in 68.1% of the applications.
- **CWE-276:** File permissions allow installed files to be modified by anyone. This vulnerability was found in 72.7% of the applications.
- **CWE-327 (sha1)**: This vulnerability corresponds to the use of weak cryptographic algorithms. This vulnerability was found in 40.9% of the applications.
- **CWE-327 (ECB Mode in crypto encrypt):** This vulnerability was found in 9.09% of the applications.
- **CWE-327 (md5)**: This vulnerability was found in 18.1% of the applications.
- **CWE-89**: The SQL command elements are not properly taken care of which can lead to SQL injection. This vulnerability was found in 59.09% of the applications.
- **CWE-250:** Application executes operation at a privilege level higher than the level required, which gives birth to new weaknesses. This vulnerability was found in 9.09% of the applications.
- **CWE-295**: This vulnerability of incorrectly validating or not validating a certificate by an application. This vulnerability was found in 13.6% of the applications.

IV. **Trackers:** Trackers are used to track the general data about the application behaviour or the user's behaviour in a subtle manner by capturing either the state of the application or the generic information of the user such as preferences of the user. Sometimes, the data collected by these trackers may lead to privacy leaks.

- **Google Admob:** It's an in-app advertising platform. This tracker collects information such as device information. This tracker was found in 4.5% of the applications.

- **Google CrashLytics:** It is a real time crash reporting system and provides useful tips to fix simple issues. It accesses your application's error events. This tracker was found in 27.2% of the applications.
- **Google Firebase Analytics:** It is a software that analyses your application and provides application usage and user engagement and also reports on 500 distinct events and user properties. This tracker was found in 45.4% of the applications.
- **Microsoft Visual Studio App Center Analytics:** It is similar to google firebase analytics that tracks user activities. This tracker was found in 4.5% of the applications.
- **Microsoft Visual Studio App Center Crashes:** It is very similar to google crashlytics and collects information about crashes and errors in your application and uploads them. This tracker was found in 4.5% of the applications.
- **AltBecon:** It is a specification defining the format of the advertisement broadcast message that Bluetooth Low Energy proximity beacons. This tracker was found in 4.5% of the applications.
- **OneSignal:** It is a push notification service. This tracker was found in 4.5% of the applications.
- **Bugsnag:** Bugsnag is an error monitoring and reporting service for mobile apps. This tracker was found in 4.5% of the applications.

V. **Miscellaneous-** Some important findings are discussed in this section.
- **TousAntiCovid:** CAPTCHA API KEY is exposed. If this key is used for multiple purposes, then the attacker can misuse this key.
- **COVA Punjab**: Labour Key is hardcoded. Hard-coded cryptographic key allows malicious users to gain access to the account.
- **Hamagen**: Key hardcoded: "transistorsoft_key". A key probably related to transistor software which is a module for background location tracking and geofencing and is aware of battery and motion of the device.
- **Ito:** This is the only application which did not use sha1 with RSA and did not sign the APK with debug certificate.
- **PeduliLindungi:** Only this application asked for permission - android.permission.ACCESS_MEDIA_LOCATION
- **Rakning C-19:** Only this application asked for permission- android.permission.AUTHENTICATE_ACCOUNTS and implemented Microsoft Visual Studio App Center Analytics and Microsoft Visual Studio App Center Crashes and Bugsnag.
- **DOCANDU Covid Checker:** Only this application asked for permission- android.permission.RECORD_AUDIO.
- None of the applications asked the permission- android.permission.GET_TASKS
- **SM-COVID-19:** Only this application had the vulnerability CWE-919 and implemented AltBecon and OneSignal.
- **COVA-Punjab:** Only this application implemented the tracker Google Admob.

3.2. MITM analysis: For the MITM flow analysis, mitmproxy which is a free open-source interactive HTTPS proxy was used. We can use mitmproxy to intercept requests, modify requests and replay requests. This can be used in three ways.

- **mitmproxy** is an interactive, SSL/TLS-capable intercepting proxy with a console interface for HTTP/1, HTTP/2, and WebSockets.
- **mitmweb** is a web-based interface for mitmproxy.
- **mitmdump** is the command-line version of mitmproxy. Think tcpdump for HTTP.

3.2.1. Methodology: We used mitmproxy in macOS which can be installed through terminal using brew install mitmproxy. To analyse the packets sent and received by the applications, the IP address of mac is set as the Wi-Fi proxy IP for a mobile device so that all the traffic can be seen on the PC. Also, we have used both android and iOS mobile devices to analyse the traffic.

3.2.2. Results:

I. **COVIDAlert, Smittestop, Koronavilkku, COVID Tracker Ireland, Immuni,Beat Covid Gibraltar**: Do not leak any private data related to the user of the device. This is because all of these applications use random Bluetooth IDs to identify the same application nearby and whether a person is diagnosed with COVID-19. These IDs are random and change a certain amount of time and thus does not reveal any private data of the user.

II. **COVID Safe App:** This application initially sent 4 requests to the internet. It revealed the android version and the browser, and its version being used on the device. Moreover, the Cross-Origin-Resource-Policy was set to cross-origin and the X-XSS-Protection was not set.

III. **TousAnti COVID:** This application initially sent 6 requests to the internet. It revealed the android version and the browser, and its version being used on the device. Moreover, for most of the connections, the Cross-Origin-Resource-Policy was set to cross-origin and the X-XSS-Protection was not set.

IV. **COVA Punjab:** This application initially sent 7 requests to the internet. One of the requests, GET https://firebase-settings.crashlytics.com HTTP/1.1 revealed X-CRASHLYTICS-DEVELOPER-TOKEN as 470fa2b4ae81cd56ecbcda9735803434cec591fa and an attacker accessing this token immediately can get sensitive information about the user on the developer's behalf.

V. **SM-COVID-19:** This application initially sent 34 requests to the internet. Firstly, this indicated that this application is consuming more bandwidth than the similar applications. Also, it revealed X-CRASHLYTICS-DEVELOPER-TOKEN: 470fa2b4ae81cd56ecbcda9735803434cec591fa. According to Crashlytics terms of Service, Crashlytics collects data such as Device state information, Unique device identifiers, Location data, Usage data, Email address (depending on how the developer implements Crashlytics). This sort of data can constitute personal information. It reveals information about individual people, and can, in theory, be linked to them. It also reveals how individuals use your app.

4. Web services testing (WEBSITES):

We evaluated 66 web services related to covid-19 from various geographical regions. Most of the websites are government owned and used as a tracker for the pandemic and some can be used to get appointments of the tests and other services. The list of websites is attached in the appendix (See Appendix 2).

5. Tool Used:

5.1.Immuniweb: ImmuniWeb is an online tool which is also known as an AI platform for Application and Website Security testing. The community edition of Immuniweb is free which offers static and dynamic analysis of both Android/iOS apps and web services with the use of OWASP Top 10 vulnerabilities.

5.1.1. Methodology: We used immuniweb for the website security testing which includes the GDPR and PCI DSS test, CSP and HTTP headers check and Website CMS security testing.

5.1.2. Analysis (See Appendix 2):

I. **Software Security Analysis**: In this, ImmuniWeb analyses the software of the website and then reports any issues if present. Immuniweb uses the concept of fingerprinting in order to analyse the software security. The percentage of websites that passed the software security analysis or in other words those websites which have no software security issues are 57.8%.

II. **GDPR Compliance**: General Data Protection Regulation or GDPR is a regulation that mandates these websites to safeguard the personal data and privacy of European Union citizens for requests and responses to and from clients and servers within EU states. For example, if a website needs to be GDPR compliant it needs to keep in private the basic identity information such as name, address, web data such as location, IP address, cookie data etc. ImmuniWeb basically checks for privacy policy of the website; website software security that can lead to privacy leak; SSL/TLS traffic encryption; cookie configuration; and cookie disclaimer. The percentage of websites that are GDPR compliant are 15.6%.

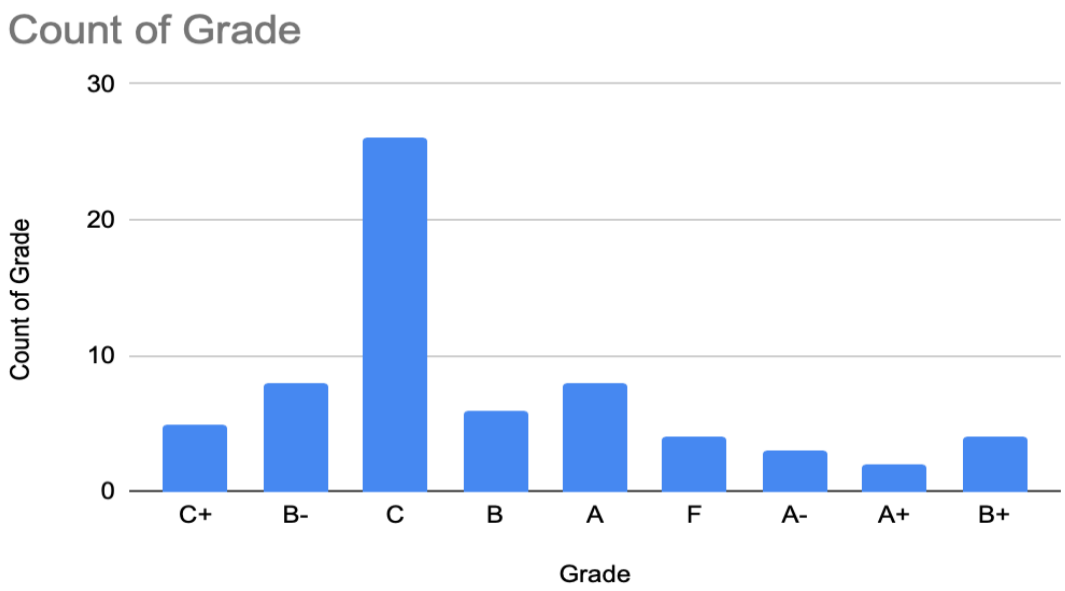
III. **PCI DSS compliance**: The Payment Card Industry Data Security Standard or PCI DSS is a set of security standards that requires all companies to provide secure environment for the credit card information if they want to accept, process, store or transmit credit card information. The percentage of websites that are PCI DSS compliant are 14.6%.

IV. **CSP**: Content Security Policy or CSP is a layer of security that detects few forms of attacks such as Cross Site Scripting (XSS) and data injection attacks and also helps the website to mitigate them. The percentage of websites that have CSP are 10.9%.

V. **HTTP header security**: This checks for HTTP headers related to security and privacy such as Strict-Transport-Security, X-Frame-Options, X-XXS-Protection, X-Content-Type-Options, Expect-CT etc. Public-Key-Pins, Public-Key-Pins-Report-Only,

Permissions-Policy, Access-Control-Allow-Origin. The percentage of websites that have HTTP header security are 48.4 %.

VI. **Miscellaneous:** Only 2/68 websites got A+ and 4/68 received an F grade from immuniweb.



5.2.Mozilla Observatory: Mozilla observatory is a platform for the developers to test their website and configure it safely and securely. The Observatory tests for cross-site scripting attacks, man-in-the-middle attacks, cross-domain information leakage, cookie compromise, content delivery network compromise, and improperly issued certificates but it does not test for outdated software versions, improper policies for password creation, SQL injection vulnerabilities and many others.

5.2.1. Methodology: We analysed the same websites with more properties included that were represented by immuniweb in clubbed format in order to gain deeper insight of these web services.

5.2.2. Analysis and Results (See Appendix 3):

I. **HTTP Observatory-** The findings related to HTTP connections of these websites are reported here.

- **CSP (Content Security Policy):** A layer of security that detects attacks such as Cross Site Scripting (XSS) and data injection attacks and also helps the website to mitigate them. CSP was implemented in 1.5% of the websites.
- **Cookies:** An HTTP cookie is data stored on the user's computer by the web browser while browsing a website. They are used to remember information such as names, addresses, passwords, and payment card numbers etc. Security vulnerabilities may allow a cookie's data to be read by a hacker. The security of a cookie depends on the security of the website and the web browser, and whether the cookie data is encrypted. No cookies are detected for 35 websites. Among the remaining websites

38.7% of the websites all cookies use the Secure flag, and all session cookies use the HttpOnly flag. The remaining websites did not use secure flag over HTTP.

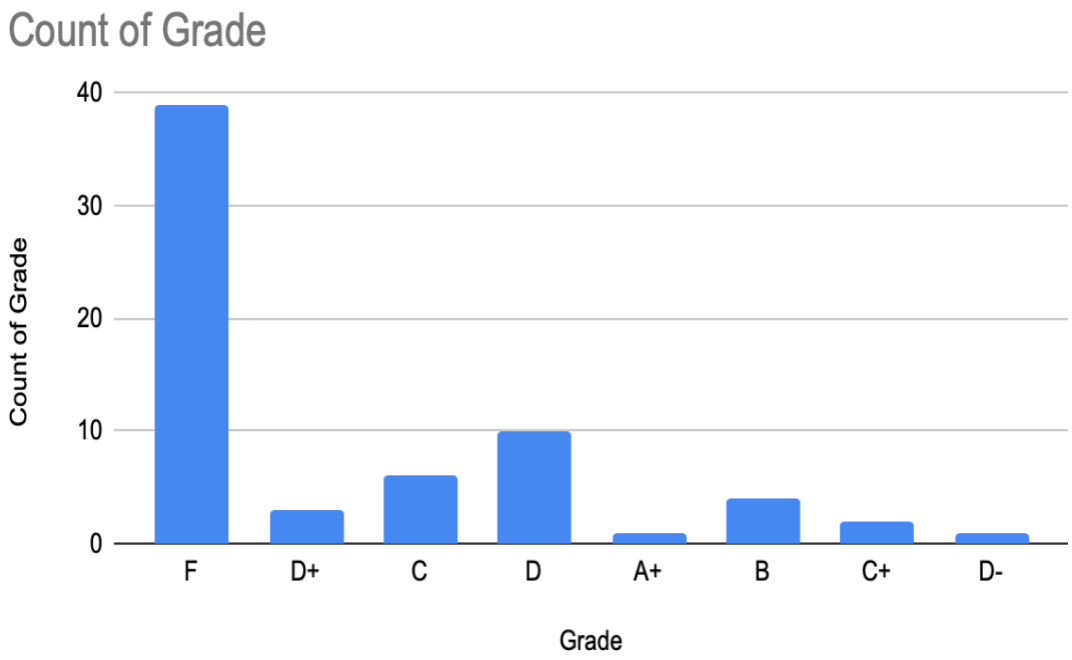
- **Cross-Origin Resource Sharing:** Cross-Origin Resource Sharing is an HTTP-header that allows a server to indicate any other domains other than its own to permit browser the loading of resources. In context of security, browsers restrict cross-origin HTTP requests initiated from scripts. In 98.4% of the websites, content is not visible via cross-origin resource sharing file or headers which is good.
- **HTTP public key pinning:** HTTP Public Key Pinning is a security feature that tells a client to associate a public key with a particular web server to minimize the threat of MITM attacks by forging of certificates. It has been deprecated in modern browsers. This was found in 0% of the websites.
- **HTTP strict transport Security:** The HTTP Strict-Transport-Security response header sent by a website tells the browser that it should only be connected using HTTPS protocol instead of an HTTP protocol. This header was implemented in 33.3% of the websites.
- **Redirection:** HTTP allows servers to redirect a client request to a different location which can be used by remote attackers to redirect users to arbitrary web sites. A fairly good percentage that is 78.7% of the websites were redirected HTTPS websites rest were redirected to an HTTP website.
- **Referrer Policy:** The HTTP referrer is an HTTP header field that identifies the address of the webpage linked to the resource. By checking the referrer, the webpage can see where the request came from. The Referrer-Policy HTTP header controls how much referrer information should be sent with HTTP referrer header. If this Referrer-Policy HTTP header is not set properly then this may lead to private data leaks. Referrer-Policy header set to no-referrer, same-origin, strict-origin or strict-origin-when-cross-origin in only 3% of the websites.
- **Subresource integrity:** Subresource Integrity (SRI) is a security feature that enables browsers to verify that resources they fetch are not manipulated by matching a hash of a fetched resource. This header was implemented in 0% of the websites.
- **X-Content-Type-Options:** It is a marker in response header of a server that shows that the MIME types in the Content-Type headers should not be changed. MIME type indicates the nature and format of a document or file. This way we can prevent MIME type sniffing. X-Content-Type-Options header was set to "nosniff" in 31.8% of the websites.
- **X-Frame options:** It is an HTTP response header used to tell the browser whether it should be allowed to render a page in a <frame>, <iframe>, <embed> or <object> hence preventing click-jacking attacks and ensure that their content is not embedded into other sites. This header was implemented in 33.3% of the websites.
- **X-XSS-Protection:** The HTTP X-XSS-Protection response header is a feature that prevents pages from loading when reflected cross-site scripting attack is detected. This header was implemented in 30.3% of the websites. X-XSS-Protection header is usually set to "1; mode=block" in these websites.

II. **TLS Observatory:** The findings related to TLS connections of these websites are reported here.

- **Valid Certificate:** Valid certificate indicates that a certificate authority (CA) has verified that the web address of a particular organization. Valid Certificate was found in 98.4% of the websites.
- **TLS 1.0:** TLS 1.0 is vulnerable to man-in-the-middle attacks which can lead to privacy leaks of data sent between a website and a browser. This was found in 27.6% of the websites.

III. **Miscellaneous-** The other findings are listed here.

- Only 1/68 websites got A+ and 38/68 received an F grade from Mozilla observatory.
- Only one website named **https://ncov.moh.gov.vn** was found with an invalid certificate.
- Only one website named **https://covid.gobusiness.gov.sg** implemented a CSP header with unsafe sources inside style-src.
- Only one website named **https://covid-19.chinadaily.com.cn** had content visible via cross-origin resource sharing file or headers.
- No website had implemented HTTP public key pinning header which is also deprecated.
- Only 2 websites named **https://covid.gobusiness.gov.sg** and **https://www.covid19.gov.ph** implemented Referrer-Policy header that were set to no-referrer, same-origin, strict-origin or strict-origin-when-cross-origin.
- **Subresource integrity** header was not implemented in any website.



6. Challenges faced:

I. For MobSF, some of the applications were region based which couldn't be installed on an emulator. So, we were not able to perform dynamic analyses on all of the applications.

II. For mitmproxy, most of the applications uses Bluetooth low energy to interact with peer applications on other devices and have minimal to no interaction with the internet after they are installed. Also, the information shared with other applications using Bluetooth are pseudorandom numbers that tell nothing specific to the user. This contributed to the less data for analysis.

III. Also, some applications require a local phone number to receive an OTP and setup the app. So, we were not able to track run-time traffic of those apps.

IV. We were not able to modify or replay the requests because server-side attacks were non-goal, and we didn't find a way to modify responses in mitmproxy.

7. Conclusion:

Overall, for covid-19 related apps, nothing critical was found. However, there were vulnerabilities and loopholes in the websites and applications that could lead to privacy leaks. A deeper investigation is required to understand the side effects of these security vulnerabilities and loopholes in context to privacy of the user.

8. References:

1) https://www.termsfeed.com/blog/crashlytics-privacy-policy/
2) Information Exposure [CWE-200]. In https://www.immuniweb.com/vulnerability/information-exposure.html, March 9, 2020
3) CWE List Version 4.0. In https://cwe.mitre.org/data/index.html.
4) Google Analytics. In https://firebase.google.com/docs/analytics.
5) Wikipedia contributors, "Google Analytics," Wikipedia, The Free Encyclopedia, (accessed April 22, 2020)
6) A Beginner's Guide to Facebook Insights. In https://neilpatel.com/blog/guide-to-facebook-insights/.
7) Firebase Crashlytics. In https://firebase.google.com/docs/crashlytics.
8) Adobe buys behavioral data management platform demdex In https://techcrunch.com/2011/01/18/adobe-buysbehavioral-data-management-platform-demdex/.
9) Wikipedia contributors. "ImmuniWeb." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 22 Jan. 2020. Web. 22 Apr. 2020
10) GDPR Compliance and Application Security. In https://www.immuniweb.com/compliance/gdpr/.
11) The Positive and Negative Implications of GDPR In https://www.timedatasecurity.com/blogs/the-positive-andnegative-implications-of-gdpr.
12) The Easy Way to Manage and Maintain PCI Security Controls. In https://www.pcicomplianceguide.org/. April 6,2020.
13) Content Security Policy (CSP). In https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP.
14) Insecure Cookies. In http://kb.enprobe.io/vulnerabilities/insecure-cookies.html.
15) 95% of HTTPS servers vulnerable to trivial MITM attacks In https://news.netcraft.com/archives/2016/03/17/95- of-https-servers-vulnerable-to-trivial-mitm-attacks.html.
16) HTTP headers In https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers.
17) Web Security In https://infosec.mozilla.org/guidelines/websecurity.

9. Appendix:

1) MobSF Scan worksheet

2) ImmuniWeb Scan Worksheet

3) Mozilla Observatory Scan Worksheet

4) MobSF scan reports attached in a separate folder.