# A Vulnerability Assessment on the Parental Control Mobile Applications' Security: Status based on the OWASP Security Requirements

**Eric B. Blancaflor, Gerardine Anne J. Anson, Angela Mae V. Encinas, Kiel Cedrick T. Huplo, Mark Anthony V. Marin, Stephany Lhaime G. Zamora**
School of Information Technology
Mapúa University
Makati City, Philippines
ebblancaflor@mapua.edu.ph , gajanson@mymail.mapua.edu.ph,
amvencinas@mymail.mapua.edu.ph, kcthuplo@mymail.mapua.edu.ph,
mavmarin@mymail.mapua.edu.ph, slgzamora@mymail.mapua.edu.ph

## Abstract

Parental control software and hardware are the most common solution for regulating their children's online activities. Consequently, while this may ease their worries about internet use regulation, it also introduces grave security and privacy threats. This study was able to analyze the underlying vulnerabilities present in commonly used parental control mobile applications by parents and propose recommendations from the identified vulnerabilities. This quantitative research case study used a vulnerability assessment method utilizing the Quixxi Security tool to yield a descriptive analysis of the results. FamilyTime, FamilyLink, and OurPact are the identified commonly used parental control application, and they have ten vulnerabilities common with each other.

## Keywords

Cybersecurity, Vulnerability Assessment, Parental Mobile Application, OWASP, MASVS

## 1. Introduction

Minors are constantly being exposed to different dangers with navigating the internet unsupervised, they use the internet as their means to express themselves completely unaware that the mobile applications that they are using are leaving them vulnerable to attackers. Due to the children's exposure to the Internet, there is a higher possibility that they would encounter brutality, corrupt practices, or exploitation (Council of Europe [CoE], n.d). Ellis (2020) listed the best free parental control software applications there is in the market that promotes child safety when browsing the internet, which also allows their parents to not worry about them encountering inappropriate content. According to published research on children protection on the internet by Denić et. al (2017), ten percent of its respondents, who are parents, used software to control their children's access to web-addresses. As much as these applications can protect the children's safety, it is also riddled with vulnerabilities that fail to secure the private information of its users that would be used to manipulate as well as steal user data by the attackers (Anurag, 2018).

The Open Web Application Security Project or OWASP (2020) provided information to developers regarding common coding are being used to identify potential vulnerabilities and suggested procedures on how to secure code by application security experts. Olekss (2020) stated that codes are written by high-skilled developers also have a bug wherein some software has been reported to have fifteen to fifty bugs per one thousand lines of code. Insecure data storage, a common issue, is found in 76 percent of mobile applications. This kind of issue puts sensitive and confidential data, such as financial information and personal data, at risk. An article published by Positive Technologies (2019) presented that thirty-eight percent (38%) of existing mobile applications for iOS users are at high-risk vulnerabilities and forty-three percent (43%) for Android users. With the use of malware, eighty-nine percent (89%) of vulnerabilities can be exploited when users install a mobile application with the ignorance of security implications.

### 1.1 Objectives

This study was able to assess the vulnerabilities of the parental control mobile application security status based on the OWASP security requirements. Specifically, this study was able to:

1. determine the three most frequently used parental control application;
2. scan vulnerabilities of each of the most used parental control application with the use of the Quixxi Security tool; and
3. identify which vulnerabilities are common between the three parental control applications.

### 1.2 Scope and Limitation

The research mainly focused on performing a vulnerability assessment on the top three most used parental monitoring tools based on the OWASP security requirements. The parental control tools are the top three most used mobile applications that parents are using based on the gathered survey. The vulnerability assessment was done with the utilization of the Quixxi Security tool.

The result of the vulnerability assessment is limited to identifying the total scanned and detected vulnerabilities, the severity level of threats, and assessment status. It is also restricted in determining the general security issues, exploits information, and common weakness types of the vulnerabilities detected.

## 2. Literature Review

### 2.1 Parental Control Tools

A study by Saravid et.al (2018) describes parental control tools as applications or software implemented in operating systems of different devices that restrict the abuse of some software like games and other unnecessary or inappropriate websites that are not for children. By using parental control, parents could encourage decent behavior and avoid such bad behaviors that the children can adapt.

### 2.2 The OWASP's Mobile Application Security Verification Standard (MASVS)

The Open Web Application Security Project® or OWASP (2020) is a nonprofit organization that aims to establish foundations of security of software. They are one of the sources of developers, security analysts, and technologists to secure the web through tools and resources, community and networking, and education and training.

OWASP established the Mobile Application Security Verification Standard or the OWASP-MASVS (2016). This is a set of security requirements for mobile apps. It is usually used in application development, mobile application penetration testing, procurement, etc. The MASVS is based and a parallel project under the OWASP Mobile Security Testing Guide. The following are the security requirements under the MASVS:

- V1. Architecture, Design, and Threat Modeling Requirements: This requirement list pertains to the architecture and design of the application.
- V2. Data Storage and Privacy Requirements: This requirement list pertains to user data protection, especially, sensitive data such as user credentials.
- V3. Cryptography Requirements: This requirement list intends to implement cryptographic standards to further protect stored data.
- V4. Authentication and Session Management Requirements: This requirement list is for the security of users logging into remote services.
- V5. Network Communication Requirements: This guarantees the integrity and confidentiality of information exchange between apps and services within a network or over the Internet.
- V6. Platform Interaction Requirements: The usage of standard APIs and secure components is the main focus of this requirement list.
- V7. Code Quality and Build Setting Requirements: This requirement list focuses on secure coding practices in application development.
- V8. Resilience Requirements: This requirement list focuses on threat modeling and assessment of the effectiveness of the protection of the application against obfuscations. This is the security requirement to avert reverse-engineering attacks.

### 2.3 Common Vulnerabilities and Exposures (CVE)

The Common Vulnerabilities and Exposure or CVE (2016) is a system that provides a list of computer and information security vulnerabilities and exposures that aims to provide common names for publicly known problems. CVE reports can come from a vendor or any user that can discover a security flaw from a computer or open-source software. A CVE entry includes the CVE ID which provides a brief description of the security vulnerability or exposure, and references, which can include links to vulnerability reports and advisories.

### 2.4 Common Weakness Enumeration (CWE)

Common Weakness Enumeration or CWE (2020) is a collection of software and hardware weakness or vulnerabilities which is a community developed. The CWE's can be described and discussed in hardware and software in a common language by developers, also, it can analyze and check for weaknesses present in hardware and software products. It can also help developers prevent vulnerabilities of software and hardware before their deployment. The CWE List consists of weak types of both software and hardware which can also be downloaded and viewed entirely.

### 2.4 Quixxi Vulnerability Scanning Tool

The Quixxi (2020) security tool is a useful scanning tool that can provide a detailed analysis of a mobile application. The main purpose of Quixxi is to enable organizations to develop, secure, operate, and sustain applications that can be trusted. It can detect critical risks and descriptively explain detected vulnerabilities. What makes Quixxi excellent is that it is automated in terms of scanning and generating reports. Quixxi performs a static analysis of mobile applications, an APK (Android Package), or an IPA (iOS App Store Package) format.

Quixxi is an OWASP-inspired vulnerability scanner. It analyzes the security areas and requirements based on the OWASP Mobile Security Project guidelines, which were previously discussed in the OWASP's Mobile Application Security Verification Standard (MASVS). In this part of the study, the recently conducted research will be presented as well as its relevance to this research (Quixxi, 2020).

### 2.5 Privacy Report Card for Parental Control Solutions

This study was able to conduct the first comprehensive study that aims to analyze different parental control software and hardware solutions. The researchers analyzed the applications by deploying a set of privacy and security tests. The applications chosen for this study were popular representatives of parental control applications available for download on Windows and Android OS.

Notable findings of the study revealed the uncovered vulnerabilities on each parental control application. The cross-platform comprehensive analysis of the selected application revealed the systematic problems in the deployment of the parental control application in terms of privacy and security which poses a grave threat to children's online and real-world safety.

The study was able to develop an experimental framework for analyzing and evaluating parental control applications. The developed framework was utilized to provide the first comprehensive study of parental control on different platforms. This paper was also able to present the suggested proof-of-concept exploits scenarios of each identified vulnerability. The experimental framework presented in this study may serve as a guide in the current study for evaluating parental control applications in terms of their safety and security.

### 2.6 Angel or Devil? A Privacy Study of Mobile Parental Control Apps

This study aimed to analyze and assess various privacy risks on the regulatory compliance of the 46 different parental control applications. Specifically, the study aims to determine if parental control applications have vulnerabilities in terms of dangerous permissions, having 3rd party libraries, and private data dissemination.

The study searched for parental control applications and eliminated those that do not implement parental control functionalities until a total of 61 parental control apps is found. The researchers also discarded applications that have not been updated since 2016 and resulted in a final dataset of 419 versions of 46 parental control applications. The study used a static analysis wherein parsing of its android manifest file was done to understand the high-level behavior without analyzing the binary code. Dynamic analysis was also carried out to collect actual evidence on the dissemination of personal data.

The findings of the static and dynamic analysis show that the 46 applications in this study, the average tends to request more dangerous permission compared to the top 150 applications in the Google Play Store. Moreover, several dangerous permissions-protected methods are invoked embedded third-part libraries only. In the transmission of personal data, a total of 11% of the applications exploit personal data in the clear. 34% of the applications tend to send and gather personal information without the consent of the user. Moreover, 72% of the applications share user's data with third parties, including online advertisements and analytics services, without a proper indication of their presence in their privacy policies.

The study summarized that parental control applications lack in terms of their transparency on their user and lacks compliance with the regulatory requirements. It calls into question the exposure of the users to privacy risks that are present in the application.

## 3. Methods

This research used a quantitative case study procedure to conclude findings. Certain formulas and measures were used to derive results from the data that has been collected. The case study was done by using the vulnerability assessment procedure to have an explanatory and descriptive analysis of the subject of this research.

The Quixxi (2020) Security tool was used to identify the general security issues, their severity, assessment status, and their corresponding exploit definition and software weakness type. The severity level has three levels: HIGH, MEDIUM, and LOW. The automated assessment status has only binary values: FAIL or PASS. The security assessments that were marked as FAILED in the assessment status indicate the vulnerability of the scanned application.

For the treatment of data, a tally system was utilized. The score of an item has been determined based on how many occurrences it has in the data set. Tallying is used to determine the top three used parental control software and the common vulnerabilities between those three. Since the FAILED assessments indicate the vulnerabilities of the mobile app, the fail rate has been determined and was computed using the formula:

$$FAIL\ RATE = \frac{Total\ Failed\ Tests}{Total\ Vulnerabilities}$$

Additionally, for the statistical validation, the mean of the fail rate of each of the top three applications was computed. The means of fail rate of each severity level has been identified and the overall fail rate has been identified. The values of means generally represent the quantitative value of fail rate which pertains to the vulnerabilities of the parental control mobile applications. The overall fail rate was determined whether to accept or reject the directional hypothesis *"the parental control mobile applications are secured with a fail rate average of less than or equal to 40%".*

## 4. Data Collection

A questionnaire through Google Forms is the initial instrument for this research that aims to gather information on how many parents are using parental control tools, and which parental control software they are using. The top three most used software that has been gathered is the focus of the software vulnerability assessment. The respondents for the initial questionnaire are random parents or guardians with a child or children that access the Internet. Their response was treated as the input to identify the top three most used parental control mobile applications.

The software vulnerability assessment of the parental monitoring tools was done through Quixxi (2020) Security. This is an automated web-based vulnerability scanning tool that provides a detailed analysis of the risks of a mobile application. Quixxi is using Open Web Application Security Project (OWASP) security requirements and guidelines. Quixxi checks the static and runtime behaviors of an app and identifies its severity level, associated risk, and security flaws. Out of the 8 Mobile AppSec Verification Standard (MASVS) requirements, only 6 are being checked by the Quixxi tool with a total of 30 vulnerability assessment items. The Quixxi Security scanning tool checks the security requirements of OWASP's MASVS except for V1 and the V4.

## 5. Results and Discussion

### 5.1 Most Frequently Used Parental Control Application

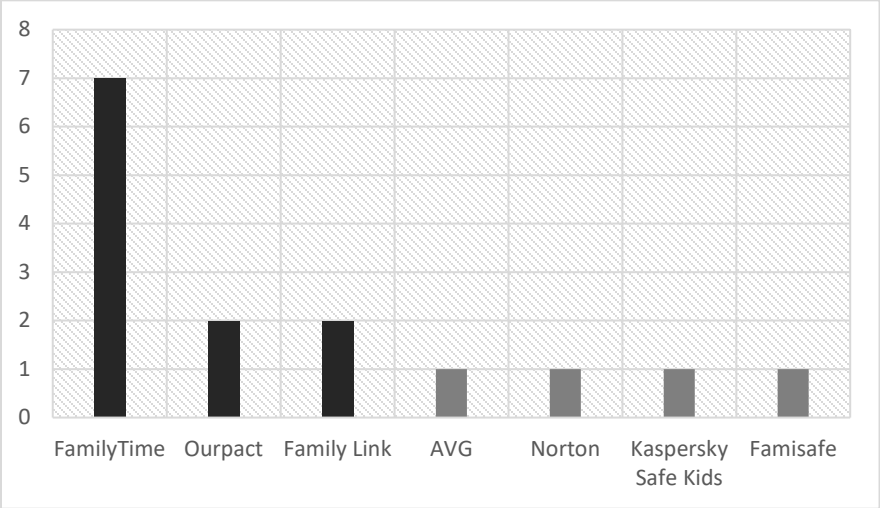


Figure 1. Most Frequently Use Parental Control Apps

As shown in Figure 1, out of the fifty participants, only thirteen of them were using parental control software. Out of those thirteen, it shows that 7 (53.8%) of the participants were using Family Time, 2 (15.4%) participants were using Google Family Link, 2 (5.4%) participants were using OurPact, 1 (7.7%) participant was using AVG, 1 (7.7%) participant was using Norton, 1 (7.7%) participant was using Kaspersky Safe Kids, 1 (7.7%) participant was using Famisafe as their parental control application for monitoring their children.

As a result, the top three most used parental control software are FamilyTime, OurPact, and Google Family Link. The APK files of these mobile applications were used as the samples for the vulnerability assessment using the Quixxi Security vulnerability scanning tool.

### 5.2 Vulnerabilities of the Most Used Parental Control Application

Table 1 shows the vulnerability assessment summary of the top three parental control applications, FamilyTime, Family Link, and Our Pact.

| ISSUE | SEVERITY | ASSESSMENT STATUSES | | |
|---|---|---|---|---|
| | | FamilyTime | Family Link | Ourpact |
| V2 Data Storage and Privacy | | | | |
| Unsafe files deletion | High | Fail | Fail | Fail |
| Missing Copy&Paste protection from EditText fields | Medium | Fail | Fail | Fail |
| Missing protection against screenshots & screen sharing | Medium | Fail | Fail | Fail |
| No blurring for the app in the background | Medium | Fail | Fail | Fail |
| ADB Backup allowed | Medium | Pass | Fail | Fail |
| Unsafe and deprecated files configuration | Low | Pass | Pass | Pass |
| V3 Cryptography | | | | |
| Weak Random Number Generator | Medium | Fail | Fail | Fail |
| Weak Hashing Algorithms | Medium | Fail | Fail | Pass |

| | | | | |
|---|---|---|---|---|
| Vulnerable SQLite Database | Low | Fail | Fail | Pass |
| Missing SQLite PRAGMA key protection | Medium | Pass | Pass | Pass |
| Unsecure cryptographic protocols | Medium | Pass | Pass | Pass |
| Strings Security based on Base64 Encoding | Low | Pass | Pass | Pass |
| V5 Network Communication | | | | |
| Unsafe Trust Manager implementation | High | Fail | Fail | Fail |
| Missing Certificate Pinning | High | Fail | Pass | Pass |
| Deprecated implementation of SSL Sockets | High | Pass | Pass | Pass |
| Unsafe HTTP Host scheme for implementing HTTPS connections | High | Pass | Pass | Pass |
| V6 Platform Interactions | | | | |
| Improper Export of Android Activities | High | Fail | Fail | Fail |
| Improper Export of Android Services | High | Fail | Fail | Fail |
| Improper Export of Android Broadcast Receiver | High | Fail | Fail | Pass |
| Improper Export of Android Content Providers | High | Pass | Fail | Pass |
| Fragment Injection | High | Pass | Pass | Pass |
| Command injection Vulnerability | Medium | Pass | Pass | Pass |
| The unsafe protection level for custom permissions | Low | Pass | Pass | Pass |
| V7 Code Quality and Build Settings | | | | |
| Debugging information provision | Medium | Fail | Fail | Fail |
| Debuggable App | Medium | Pass | Pass | Pass |
| Improper implementation of SSLErrorHandler class | Medium | Pass | Pass | Pass |
| The missing check for the download source | Low | Pass | Pass | Pass |
| V8 Resilience Requirements | | | | |
| Missing Native [C, C++] Code | Medium | Fail | Fail | Fail |
| The app allowed to run on a rooted device | Low | Fail | Fail | Fail |
| The app allowed to run in an emulator | Low | Pass | Pass | Pass |

Table 1. The Vulnerability Assessment Summary of the Top Three Parental Control Apps

As presented in Table 1, there are numerous failed assessments observable in every application tested. In the vulnerability assessment performed in the application FamilyTime, out of 10 issues with a severity level of 'High', there are six 6 assessments that have an assessment status of 'Fail' which yields a 60% fail rate. While out of the 13 issues with a severity level of 'Medium', there are seven 7 assessments that have an assessment status of 'Fail' that results in a 53.85% fail rate. Lastly, out of 7 issues with a severity level of 'Low', there are 2 assessments that have an assessment status of 'Fail' that earns a 28.57% fail rate. There is a total number of 30 scanned vulnerability assessments, and out of the 30 vulnerability assessments, 15 issues have an assessment status of 'Fail' which yields a 50.00% of the fail rate.

For the vulnerability assessment and scanning for the application Family Link, out of the 10 discovered issues with a severity level of 'High', 6 have an assessment status of 'Fail' that results in a 60% fail rate. Out of the 13 discovered issues with a severity level of 'Medium', 7 has an assessment status of 'Fail' that results in a 53.85% fail rate. And out of the 7 discovered issues with a severity level of 'Low', 3 have an assessment status of 'Fail' which yields a fail rate of 42.86%. The total number of vulnerabilities scanned from Family Link is thirty 30 and 16 issues had an assessment status of 'Fail' that results in a 53.33% fail rate.

For the last parental control application, the application OurPact was discovered to have 4 assessments which had an assessment status of 'Fail' out of 10 issues with a severity level of 'High', which results in a fail rate of 40.00%. Out of 13 vulnerability assessments with a severity level of 'Medium', 7 has an assessment status of 'Fail' that yields a 53.85% fail rate as well. Lastly, out of the 7 issues with a severity level of 'Low', only one had an assessment status of 'Fail' that results in a fail rate of 14.28%. Only 12 out of the 30 vulnerability assessments were detected as 'Fail' which is a 40% overall fail rate.

### 5.3 Common Vulnerabilities Between the Three Parental Control Applications

| Vulnerabilities Detected | Count |
|---|---|
| **Unsafe files deletion** | **3** |
| **No blurring for the app in the background** | **3** |
| **Missing protection against screenshots & screen sharing** | **3** |
| **Missing *Copy&Paste* protection from *EditText* field** | **3** |
| **Unsafe Trust Manager implementation** | **3** |
| **Weak Random Number Generator** | **3** |
| **Improper Export of Android Activities** | **3** |
| **Improper Export of Android Services** | **3** |
| **The app allowed to run on a rooted device** | **3** |
| **Debugging Information Provision** | **3** |
| ADB Backup allowed | 2 |
| Weak Hashing Algorithms | 2 |
| Vulnerable SQLite Database | 2 |
| Improper Export of Android Broadcast Receiver | 2 |
| Missing Native [C, C++] Code | 2 |
| The app allowed to run in an emulator | 1 |
| Missing Certificate Pinning | 1 |
| Improper Export of Android Content Providers | 1 |

Table 2. Vulnerability Occurrence Tally

Table 2 demonstrates the common vulnerabilities between FamilyTime, FamilyLink, and OurPact. These common vulnerabilities were revealed by tallying the occurrence of each identified issue between the three parental control applications.

Out of the 10 common vulnerabilities, 4 or 40% are having a 'High' severity level. Five or 50% of them are having a 'Medium' severity level. Lastly, only one or 10% of the ten is having a 'Low' severity level. These ten common vulnerabilities between the three most used mobile parental apps will be the focus of the recommendations in terms of mitigation suggestions.

### 5.4 Proposed Improvements

The following are the recommendations based on the CWE list Version 4.2 (2020) for the common vulnerabilities between the three most used applications:

- The issue of unsafe file deletion, missing Copy&Paste protection from EditText fields, missing protection against screenshots & screen sharing, and no blurring for the app in the background can be mitigated in the Architecture and Design phase by separating privileges. The issue has a likelihood of "High" in exploitation, that is why it is required to divide the system into categories, so that safe areas wherein trust boundaries can be drawn. Also, not allowing sensitive information or data to go beyond the trust boundary and appropriate division of the system must be ensured in building the system's design. Designers and architects are obliged to rely on the principle of least privilege for their decision when it is appropriate in dropping and using system privileges (CWE 2020).

- The issues of improper export of Android Activities and improper export of Android Services can be mitigated by four phases. In the Build and Compilation phase, marking components with android: exported="false" in the application manifest must be conducted, if the application doesn't need to be shared by other applications. If the

usage of exported components between related apps under a control is intended, the utilization of android: protection Level= "signature" in the XML manifest will limit access to applications signed. The strategy of this phase is Attack Surface Reduction. According to the CWE (2020) code CWE-200 which deals with the exposure of sensitive information to an unauthorized actor, the phases Build and Compilation; Architecture and Design is the limiting of Content Provider permissions (read/write) as appropriate. The strategies in this method phase are Attack Surface Reduction and Separation of Privilege.

- The issue of debugging information provision can be alleviated in the implementation phase by not leaving the debug statements, which could be used in the source code for execution. Also, all the debug information must be eradicated before releasing the software. Moreover, in the Architecture and Design phase, the system must be divided into categories to have safe areas trust boundaries wherein it can be unambiguously drawn. Sensitive information must not be allowed to go beyond the trust boundary and must always be careful in the interaction with a compartment out of the safe area. A standard established by CWE (2020) in the code CWE-926, designers and architects must always rely on the principle of least privilege when deciding on using and dropping appropriate system provisions.

- One of the potential mitigations for the security issue regarding the unsafe Trust Manager implementation can be alleviated during the Architecture and Design, and Implementation phase. This phase is the managing and checking of certificates for ensuring that the data are encrypted with the predetermined owner's public key. Another potential mitigation for this issue the Implementation phase. In this phase, before the certificate is pinned, the needed properties of the certificate should be completely validated and guaranteed, if certificate pinning is being utilized (CWE, 2020)

- The issue about the weak random number generation can be mitigated by performing two phases. The phases are Architecture and Design and Implementation. According to CWE (2020), the Architecture and Design phase is the specification of a true random number generator for cryptographic algorithms. The Implementation phase is the implementation of a true random number generator for cryptographic algorithms.

### 5.4 Validation

To decide whether to accept the stated directional hypothesis of this case study, the mean of the fail rate in every severity level is determined and the overall fail rate for every mobile app is presented in Table 2.

| SEVERITY LEVELS | Family Time | Family Link | Ourpact | Mean |
|---|---|---|---|---|
| **High** | 60.00% | 60.00% | 40.00% | **53.33%** |
| **Medium** | 53.85% | 53.85% | 53.85% | **53.85%** |
| **Low** | 28.57% | 42.85% | 14.28% | **28.57%** |
| **Overall** | 50.00% | 53.33% | 40.00% | **47.78%** |

Table 3. Fail Rate Percentage and Mean of the Parental Control Apps

It shows in Table 3 that the mean failure rate of the HIGH, MEDIUM and LOW severity levels are 53.33%, 53.85%, and 28.57%, respectively. This suggests that most of the vulnerabilities detected in the parental control application have a severity level of 'High' and 'Medium' and the vulnerability tests with a severity level of 'Low' are mostly secured. The overall fail rate (vulnerability) mean is 47.78% and this indicates that almost half of the vulnerability tests were failed by the top three parental control mobile applications. This rejects the directional hypothesis and indicates that the top three most used parental control mobile applications are not secured because the overall fail rate mean is more than 40%.

## 6. Conclusion

At the end of the study, the vulnerabilities of the parental control mobile applications were able to be identified using the Quixxi Security tool. The said scanning tool used six OWASP security requirements (MASVS) and using this standard, the specific issues and vulnerabilities, and their corresponding assessment statuses were identified.

Based on the survey on the parents, the top three most used parental control applications were revealed, and these are Family Time, Family Link, and OurPact. All these three have their set of security issues detected, ranging from the severity level, 'Low' to 'High'. The total number of vulnerabilities detected in the three mentioned applications ranges from 12 to 16. Out of those vulnerabilities, they have 10 vulnerabilities that are common to each other which are mostly having 'Moderate' to 'High' severity levels. It can be decided that the top three most used parental control mobile applications are not that secured based on the OWASP's MASVS standards because of their overall fail rate.

## References

Anurag., 10 Biggest Risks to Mobile Apps Security, Available: https://www.newgenapps.com/blog/10-biggest-risks-to-mobile-apps-security/, October 2018.

Common Weakness Enumeration (CWE)., About CWE, Available: https://cwe.mitre.org/about/index.html/, August 2020

Common Weakness Enumeration, CWE List Version 4.2, Available: https://cwe.mitre.org/data/index.html/, December 2020

Common Weakness Enumeration, CWE-200: Exposure of Sensitive Information to an Unauthorized Actor, Available: https://cwe.mitre.org/data/definitions/200.html/, August 2020

Common Weakness Enumeration, CWE-295: Improper Certificate Validation, Available: https://cwe.mitre.org/data/definitions/295.html/, July 2006

Common Weakness Enumeration, CWE-926: Improper Export of Android Application Components, Available: https://cwe.mitre.org/data/definitions/926.html/, July 2013

Council of Europe. n.d. Children and the Internet: Protection and participation, Available: https://www.coe.int/en/web/portal/children-and-the-internet/, October 2020

Denić, N., Nešić, N., Radojičić, M., Petković, D., and Stevanović, M., A contribution to the research of children protection in use of internet, Technical Gazette, Vol. 24 pp. 525-533, 2017.

Ellis, C, The best free parental control software and apps 2020, Available: https://www.techradar.com/best/parental-control/, November 2020

Faessler, F., Alexander, G., Crete-Nishihata, Hilts, A., Kim K., Safer Without Korean Child Monitoring and Filtering Apps, Available: https://citizenlab.ca/2017/09/safer-without-korean-child-monitoring-filtering-apps/ September 2017.

Feal, Á., Calciati, P., Vallina-Rodriguez, N., Troncoso, C., & Gorla, A., Angel or Devil? A Privacy Study of Mobile Parental Control Apps, *Proceedings on Privacy Enhancing Technologies*, 2020, pp. 314-335.

Lemos, R., 6 ways to eliminate the most common security #fails in mobile apps, Available: https://techbeacon.com/security/6-ways-eliminate-most-common-security-fails-mobile-apps/, October 2020.

Mannan, M. and Youssef, A., Privacy report card for parental control solutions, Available: https://www.priv.gc.ca/en/opc-actions-and-decisions/research/funding-for-privacy-research-and-knowledge-translation/completed-contributions-program-projects/2019-2020/p_2019-20_02/, August 2020

Olekss, J., Vulnerabilities – if you can't beat them, hide them, Available: https://www.intertrust.com/blog/vulnerabilities-if-you-cant-beat-them-hide-them/ May 2020.

OWASP., OWASP Mobile Application Security Verification Standard, Available: https://github.com/OWASP/owasp-masvs/, March 2020

OWASP., What is CVE?, Available: https://github.com/RedHatProductSecurity/CVE-HOWTO/, November 2016

OWASP., Who is the OWASP Foundation? Available: https://owasp.org/, October 2020

Positive Technologies., Vulnerabilities and threats in mobile applications, 2019. Available: https://www.ptsecurity.com/ww-en/analytics/mobile-application-security-threats-and-vulnerabilities-2019/, June 2019.

Quixxi Security, How Secure is your mobile app?, Available: https://quixxisecurity.com/scan/, October 2020

Red Hat. 2020. What is CVE?, Available: https://www.redhat.com/en/topics/security/what-is-cve/, October 2020.

Saravid, A., Suchaad L., Mashiko, K., Ismail, N., and Abidin, M., Blockchain Use in Home Automation for Children Incentives in Parental Control, *Proceedings of the 2018 International Conference on Machine Learning and Machine Intelligence (MLMI2018)*, New York, NY, USA, September 2018, pp. 50–53.

## Biographies

**Eric Blancaflor** is an Associate Professor of Mapua University, Philippines. He earned B.S. in Electronics Engineering from Mapua University, Masters in Engineering major in Computer Engineering in the University of the City of Manila and currently working on his dissertation study as a requirement for the degree Doctor of Technology in Technological University of the Philippines. He has published conference papers related to IT systems, network design and security.

**Gerardine Anne J. Anson** is a high school graduate from Siena College of Taytay. She is also an on-going third-year student taking up a degree in Information Technology at Mapúa University. During her senior year in high school, she and her co-researchers conducted studies entitled, "Correlation on the Employee's Usage of the IoT Software to their Security" and "Perception of PERAA Company Employees on their Security in Using IoT (TeamViewer)".

**Angela Mae V. Encinas** is a third-year student taking the specialization track in Cybersecurity under the Bachelor of Science in Information Technology (BS-IT) program at Mapua University. She graduated high school at Centro Escolar University wherein during her education, conducted a study along with her classmates in the STEM track entitled "Incidence of Water-Borne Bacteria in Makati City".

**Kiel Cedrick T. Huplo** graduated from the University of Saint Louis in his high school and is currently a third-year college student under the course of Bachelor of Science in Information Technology (BSIT) at Mapua University. He is currently taking his specialization in Cybersecurity, along with his fellow co-authors. In his junior year in secondary school, with the help of his co-researchers, they conducted a study entitled "Evaluation of Atis Seeds, Leaves, And Barks (Annona Squamosa) Against Various Types of Pests" and while in his senior year he and his co-researchers conducted the study entitled "Library Utilization of Senior High School Students".

**Mark Anthony V. Marin** completed his high school education at Bulacan State University – Laboratory High School, under the Science, Technology, Engineering, and Mathematics (STEM) strand. He is currently a third-year student in Mapúa University under the program, Bachelor of Science in Information Technology, and specializing in Cybersecurity. Before taking the path of Information Technology and Cybersecurity, in his last year of being a STEM student, he was one of the researchers of the study "EdiBOWL: Alternative Edible Kitchenware".

**Stephany Lhaime G. Zamora** is a high school graduate from Divine Light Academy. During her secondary education, her track strand is STEM where she, along with co-researchers, conducted a study entitled "The Effectiveness of Potassium Alum Sulfate as an Alternative Flame Retardant on Fabrics". She is currently a third-year student from Mapúa University taking up B.S. in Information Technology and is specializing in Cybersecurity. She is a member of Youth for Christ (YFC) in Mapúa Makati.