National College of Ireland

Preventing Remote control Smishing on Android OS

MSc Research Project
MSc in Cybersecurity

Yejin Lee
Student ID: X20227809

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Yejin Lee
Student ID:	X20227809
Programme:	MSc in Cyber Security
Module:	Academic Internship
Supervisor:	Michael Pantridge
Submission Due Date:	14/09/2023
Project Title:	Preventing Remote control Smishing on Android OS
Word Count:	6,687 **Page Count:** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: .....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Preventing Remote control Smishing on Android OS

Yejin Lee
X20227809

Abstract

Smishing is a combination of SMS and phishing. In this study, we explore and propose remote control smishing prevention in the Android OS environment. The purpose of this study is to implement and analyse three main things. This study analyses smishing cases using an Android remote control application, Supervised Learning Classification, and extracting top keywords of smishing through text mining. It is a vulnerability discovery and prevention plan suggestion through analysis of remote-control applications released in the Android Google Store through list-up. This study was conducted using a spam message public data set and investigated 56 smishing data, and We found 30 keywords. This study analyses the latest cases of smishing and proposes measures from both user and technical aspects.

Keywords: Preventing smishing, Android OS, Remote control smishing, SMS

1 Introduction

Smishing through a smartphone remote control application is emerging as a new method. As the number of smartphone users increases, smishing crimes using mobile devices continue to increase (Kamau, 2022). Smishing combines SMS and phishing (Jain, 2019). Smishing aims to steal funds from users and is intended to obtain information such as contacts, phone versions, photos, and other information stored on mobile devices (Yeboah-Boateng, 2014). The problem with remote Smishing is that the attacker completely controls the user's device. It is more difficult to track it immediately after an accident because it removes all the tools and records used for the attack and messages between the attacker and the user after data steal.

Details of general remote access control smishing are shown in Figure 1. Remote smishing begins with sending a URL link containing malicious code to the user. And when the remote file is executed on the user's device, the attacker steals the credentials from the device. The attacker then induces the victim to directly install an application that allows remote access. Then, when the attacker runs the Malware remote-control app installed by the smartphone user, the user's smartphone receives a remote-control command from the C&C server (Mishra, 2023). In the process, the attacker can steal all of the user's data, including contact information, location information, unblock messages, recordings, photo, and remote control of banking information.

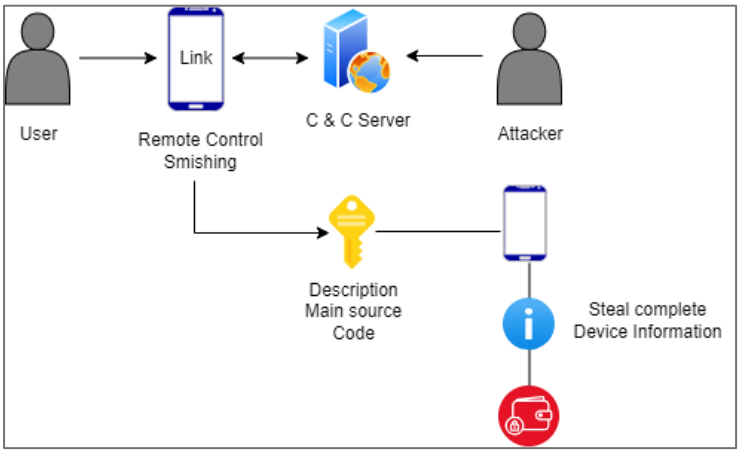

Figure 1 Smartphone URL remote-control smishing details

In addition, advanced remote smishing sometimes requires advance preparation by the attacker. It starts with the attacker gathering and analysing the target's information. It is initially sent as an SMS message to the target and then asked to move to a SNS message such as WhatsApp, Facebook Messenger, and Instagram. Recently, attackers have been using common remote applications such as TeamViewer, AirDroid, Anydesk, and VNC to take over a user's device. Remote control applications used by attackers mainly use vulnerable apk-type files, but recently, smishing cases using applications with remote control functions distributed on Google Play Store are increasing. And there have been cases in which attackers create fake digital wedding invitations and send them to their targets.

As the number of mobile users increases, many preliminary studies have already been conducted that smishing crimes are increasing (Harnett, 2023). Most previous studies have discussed the effects of malware and URL smishing. According to App Data Report 2023, users can access 2.87 million Android apps on the Android Google Store (Petroc, 2023). Moreover, 4.18 million malicious apps were discovered, and about 11,500 of the latest harmful apps were discovered daily. Smishing fraud increased by 328% in 2020. (Choi, 2021).

The study aims to propose a smishing crime analysis and prevention method using a remote-control app released on the Android Google Store. Therefore, the questions in this study are as follows: How to Prevent Remote Control Smishing in Android Operating Systems?

In this study, qualitative research and case analysis methods are followed. A related contribution of this study is remote control smishing prevention. In particular, this study is meaningful in that it can predict and prevent future remote app smishing crimes in the Android environment. Therefore, this study is expected to be used as an Android remote control smishing detection application through this study and to be used as a guideline and future research on matters to be considered when developing applications in the future.

The study is structured in the following way. Section 2 analyses the Literature Review of existing remote-control smishing studies with Related work. It also analyses the Literature Review of remote smishing in traditional Android environments. Section 3 describes the Research Methodology. Section 4 describes the technical framework and implementation in the Design Specification phase. Section 5 provides a smishing implementation solution from a user and technology perspective. Finally, sections 6 and 7 provide assessments and results for the study. We would also like to propose Future Work based on this study.

2 Literature Review

In this section, we will reinforce the need for this study through a Literature Review. To this end, we compose a total of 4 sections to find relevant research literature and write a review. Also, this section reviewed the 25 most relevant literature for our research literature review. In addition, by writing a review of the research, we justify the need for this research question by analysing the strengths and weaknesses of the existing literature together. In particular, we found the need to study technical aspects and remote smishing prevention methods that users can apply immediately. Therefore, in this study, we propose to present and use technical measures to recognize that remote control applications are running.

2.1 Typical smishing characteristics

In the study Diksha Goel and Jain(2018), the principle of common smishing attacks is that an attacker sends an SMS containing a malicious URL to the user. Subsequently, the user submits credentials on the fake login page, and the attacker claims that sensitive data information of the user is available. In addition, Park and Seo (2007) research has been conducted on the differences between phishing and smishing attack principles and attack methods. The advantage of this research literature is that it helps to understand principals by explaining the sequence of common URL smishing attack methods in mobile environments. However, since this literature focus on general smishing, more is needed to explain the recently-occurring remote control smishing problem.

Yeboah and Amanor (2014) present the characteristics of a typical smishing attack. This study explains the difference between traditional phishing attacks and modern phishing attacks. In addition, Mishara and Soni (2023) explained smishing by dividing it into three aspects: smishing through URL, e-mail ID, and phone number. However, the smishing methods presented in this study have limitations that cannot explain the remote-controlled smishing we want to suggest.

Harnett and Jones (2023) describe the short message service SMS spam to explain a typical smishing attack. In Verma and Shri (2022), in smishing, the attacker mainly contacts the target via SMS and text and mainly imitates a known entity. It also predicts that the number of smartphone users will continue to increase, suggesting that smishing will increase accordingly. These studies used to suggest that short messages in smishing attacks and that mobile smishing filtering strategies are still in their infancy.

A study by Ahmed (2020) demonstrated that smishing attackers primarily use psychological tricks to force users to obey. In addition, Breda (2017) research revealed that smishing uses hunting more often, where the attacker does not maintain contact within the network and waits for a user response. However, in recent smishing, this literature has limitations in that attackers actively contact users using psychological tricks.

2.2 Smishing detect

A study by In Jain and Gupta (2019) presented a URL analysis approach to detect smishing. In particular, this document showed that smishing can be detected with an accuracy of 98 points or more using a synthetic prime number oversampling technique. However, in the case of this remote control smishing, it does not send a malicious URL link. However, it induces the user to install a remote-control application with a message using social engineering techniques and then wholly control the device.

In Shravasti and Chavan (2021), they proposed content-based filtering, URL whitelisting, and blacklisting techniques to prevent typical smishing. The research is significant because whitelisting techniques allow us to identify trustworthy URLs. In addition, In Goel, D. and Jain (2018) this study provides a function for detecting smishing messages on mobile devices. In particular, there is an advantage in presenting an algorithm for identifying 19 suspicious keywords and detecting fake messages. However, more than this study is needed to explain the rapidly developing smishing attack in that it is proposed mainly on smishing attacks until 2017. In addition, Ansari et al., (2022) presented an idea to prevent attacks by providing essential knowledge training on attacks based on AI-based recognition. This part suggests a significant meaning for preventing remote smishing protection in the future from the viewpoint that humans cause most cyber security vulnerabilities.

In addition, Sandhya (2020), the 'Smishing Detector' for identifying smishing messages, Sandhya proposed a way to analyses users' malicious links in advance. However, more than identifying malicious code in URL links is needed to detect and prevent attacks in remote smishing attacks. For example, it should be possible to detect and prevent users and attackers from starting a conversation, which usually occurs in remote smishing. And In this study, we propose to find smishing features and top keywords, and to create and detect whitelists in advance in the Android environment.

A smishing attack prevention study by Joo (2017) also presented a detection method considering only specific keywords in the content of text messages and URL links. These research data suggest that users can avoid malicious links by detecting in advance when suspicious keywords are included in smishing. However, In Mambina and Michael (2022), Research even proactively restricting whitelisting or blacklisting has limitations as attackers change numbers periodically. In addition, in the case of remote controlled smishing that induces applications downloaded from the Google store, it is limited to explain only with the whitelist method proposed in the smishing detection study, so we need a multilateral analysis of remote smishing cases.

2.3 Remote control smishing

In Yatsyk and Shkelebei (2018), information can be accessed and controlled remotely by anyone anywhere in the world through the standardization of computer equipment and means of communication. Also, Koski (2021) research literature suggests that remote work contributes to increased phishing attacks. Once the remote control is executed, the victim's smartphone screen is exposed to the attacker as it is, and the attacker can control all input functions. This research leads to why it is essential to prevent remote control smishing in modern society in conducting this research.

In Alabdan (2020) Survey literature, the attacker initiates remote control when a drive-by download runs malware on a user's device. At this time, it is explained that remote control is performed using the command-and-control centre. As also seen in Chanti and Chithralekha (2022), smishing allows users to install malware on their devices, which runs in the attacker's background. This literature the danger of remote smishing and the possibility of attack development.

However, in remote control smishing, a URL malicious file installation method, the user steals all data such as contacts, location information, message blocking, recording, photo theft, and remote control of bank information through the remote-control app installed by the victim. In particular, since there is a need for more words and research on the latest trend, we need to investigate actual cases and study the attack principle and prevention method of remote control smishing.

2.4 Android OS vulnerability

In a study by Shahriar users are suggested to download approved applications from the google play store as a way for smishing mitigation. (2015). However, for remote smishing, an approved application download plan is not appropriate because the attack process is designed to allow users to download and use applications that are officially released on the Google Android store. Vulnerabilities in applications used for remote control smishing are mostly found in Android OS environments.

Hamandi argues for research supporting the reason. Android google play store allows the development and infection of SMS malicious code, according to Android SMS malware research literature (2013). In Kazmi et al., (2012), Android is possible without monitoring and signing certificates from Google when trying to publish an app to an application. In addition, In Mohamed et al. (2015) Android environment more vulnerable compared to the iOS environment. Also, in Sarkar et al., (2019) Android Vulnerability Study, there is a vulnerability in the Android OS that allows all Android developers to upload applications that can cause security threats to user devices to the Google Play store without going through strict security checks.

Moreover, Google Play Protect, a malware protection function built into Android, can primarily protect the user's device. However, in In Giovannitti et al., (2019) literature, Google Play Protect needs more protection and better protection in terms of protection and usability. In addition, smishing using the remote-control function application officially released on the Google Play Store is limited because it is difficult to identify with the built-in surveillance function of Android mobile phones.

3 Research Methodology

In this study, We used for methodology Descriptive Analysis methodology. Classification-supervised learning analysis is an analysis technique. Analyse the smishing data set through classification. Also, in this study, we create a model by classifying words frequently included in spam messages as variables through Naive Bayes Classifier analysis. In addition, we want

to discover words and attack patterns related to remote control hidden in large-scale data and derive rules. These discovered patterns and rules can then be used to train machine learning models to detect remotely controlled smishing.

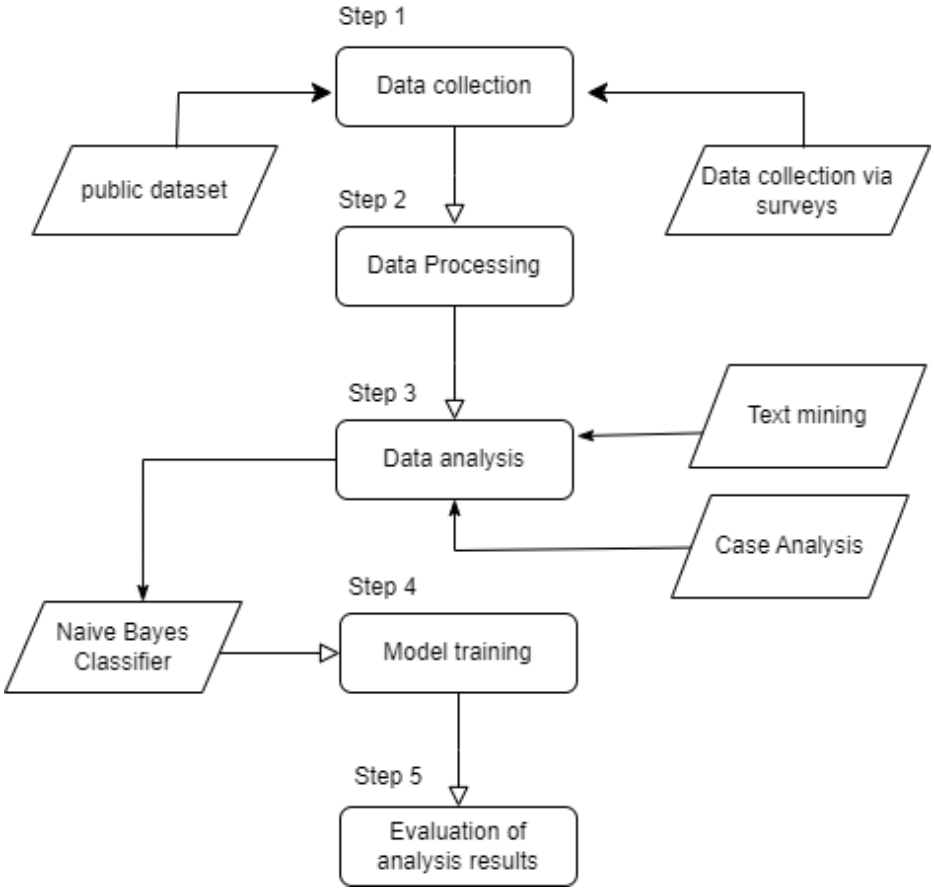

Figure 2 Research Methodology flow chart

3.1 Data Collection

To prevent remote control smishing, we chose a public data set in this study. The selected data set is the smishing The SMS Spam Collection (Almeida et al., 2012). We selected this data because it is 5572 in English and is labelled as spam or not, so it is suitable for this study. However, data preprocessing is required because this public set still contains unencoded messages.

In addition, to understand the keywords and patterns of the latest smishing in more depth, spam messages are additionally collected from anonymous National College of Ireland students through Google questionnaires, and research is conducted. The dataset does not include mail or voice spam but includes the latest types of SMS, Instagram, WhatsApp, and SNS messages.

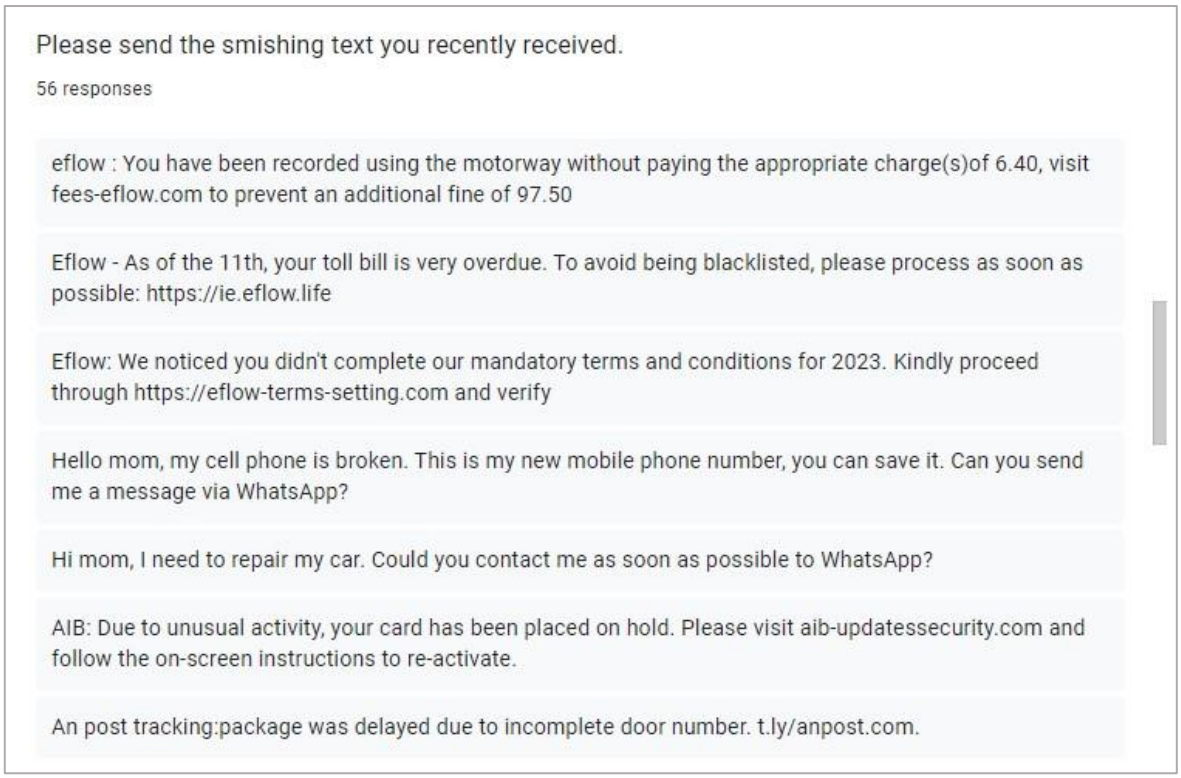

Figure 3 Recently Smishing messages

3.2 Data Processing

In order to use the data appropriately for research, we first pre-process the data. To do this, we remove punctuation to change all uppercase letters to lowercase letters and remove all special characters, punctuation marks, numbers, and control characters. In addition, after removing some invalid data from 72 spam messages collected from students, 56 spam message data of remote smishing were also obtained. Similarly, we removed unnecessary elements such as redundant messages, numbers, and special characters for additionally obtained data. In addition, we classify the dataset into a training set and a test set and model it based on the Naive Bayes Classifier after the implementation stage.

3.3 Text mining and keyword analysis

Also, we used KH coder3 open-source software to find keyword variables that frequently appear in spam messages. (KHcoder, 2023). As a tool for Classification quantitative content analysis and text mining, we used this software to analyse the frequency of keywords in the data of 803 classified spam messages. After that, in this study, we analyse what percentage of smishing characters can be prevented after adding text keywords to the keyword list on two Android mobile devices.

3.4 Classification

In addition, this study uses Classification supervised classification analysis technique. It also uses the Naive Bayes Classifier Algorithm and an open-source software tool for smishing analysis. The Naive Bayes Classifier Algorithm is one of the methods of classification

analysis. It is used as a primary analysis tool for this study because it is an algorithm that can quickly and easily analyse large amounts of data based on the frequency of keywords. Python is used as the primary language in this study, and Jupyter Notebook is used as a basis for the analysis required for the study. Also, in Naive Bayes Classifier, we use a code to prevent Laplace smoothing.

3.5 Remote control smishing case analysis

In this study, we analysed cases of remote control smishing. In general, smishing cases using remotely controlled malicious applications such as URLs and malware are not analysed. Instead, since finding the latest research on the case analysis of smishing cases using typical applications is difficult, we thought it was necessary for future research, so we covered it in this study. At this time, in the case of remote smishing, the attacker stole the credentials using an application released on Android. Through this, the sequence of remote smishing and the approach to users were investigated. In addition, applications that can be used for remote smishing were further investigated.

3.6 Evaluation of analysis results

The evaluation of the analysis results is divided into three major categories. First, block rates are measured on 3 Android devices to evaluate text mining and keyword analysis after setting the identified top keywords to a block list. Supervised classification analysis evaluation also evaluates the model's score after dividing it into training and test data sets. Finally, vulnerabilities are evaluated after analysing real remote-control cases in the Android environment.

4 Design Specification

For the design specification, we can present the following three steps in this study. Based on this architecture, we intend to derive comprehensive results after analysing keyword analysis, classification analysis, and actual cases of smishing in the implementation and evaluation stages of Chapters 5 and 6. In this study, we also use 30 high-frequency keywords, which can be used to identify spam first. Naive Bayes Classifier Algorithm For classification analysis, the data collected and preprocessed in Chapter 3 is used.

4.1 Dataset settings

The data set uses 5628 data by utilizing The SMS Spam Collection and additionally secured 56 spam data. When loading the dataset package, we used the Scikit-Learn library in Python (scikitlearn, 2023). Ham means that it is not a spam message, and spam means a message that can harm a user's device, remote smishing, or a message containing a malicious link.

Label	Text (Count)
Ham	4825
Spam	803

Table 1 Research Dataset

4.2 Parameter setting

In this study, in classification, we use Bayesian filtering to set parameters and also utilize Bayesian algorithms to calculate the probability that a message is spam based on the occurrence of certain words or phrases. We also use content analysis as a parameter in keyword extraction. In addition, we will extract 30 keywords from spam messages using KH coder3 open source and check their characteristics. In addition, we will extract the top 14 keywords from 56 spam messages with a high possibility of remote smishing and check their characteristics.

4.3 Naive Bayes Classifier Algorithm

In general, the model of the Navie Bayesian Classifier Algorithm is as follows. P(A|B) means the Posterior Probability of the Hypothesis given that the Evidence is True.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

P(B) in the denominator means Prior Probability that the evidence is true. In the numerator, P(B|A) means Likelihood of the evidence given that the Hypothesis is True, and P(A) means Prior Probability of the Hypothesis.

5 Implementation

In Chapter 5 implementation part to achieve the research goal, we include implementation to prevent remote control smishing in the Android environment. This study aims to provide insights into remote control smishing prevention from both user and technology aspects. In the figure below, in this study, we implement smishing measures in three directions: keyword detection through text mining, Naive Bayes Classifier model, and application vulnerability prediction and analysis through a remote-control smishing case study.

5.1 Spam message text mining and keyword detection

To identify the general characteristics of spam messages and for text mining, we added 56 previously collected spam data with a high probability of remote smishing together with SMS Spam Collection Dataset public data using the KH-Coder software tool and merged them from 5,572 data to 5628 data. We used 778 data for text extraction. Then, we used 803 data

classified as spam labels out of 5628 data to identify only keywords in spam messages. We extracted 30 top keywords from a total of 23,000 characters. These words are representative keywords frequently used in smishing and remote smishing and can be used as a primary keyword filtering prevention function. In the iOS environment, there is a problem that the spam blocking application released due to security issues in the message function does not operate smoothly.

Moreover, although iOS has a Filter Unknown Senders function, it does not provide a function to set only specific keywords to be blocked. However, the Android environment provides a filtering function when a user enters a specific phrase or keyword list in the setting category. Although this is simple, it is a method that can be considered if it is difficult to use the available spam-blocking applications because it automatically blocks based on keywords in the background of the Android device.

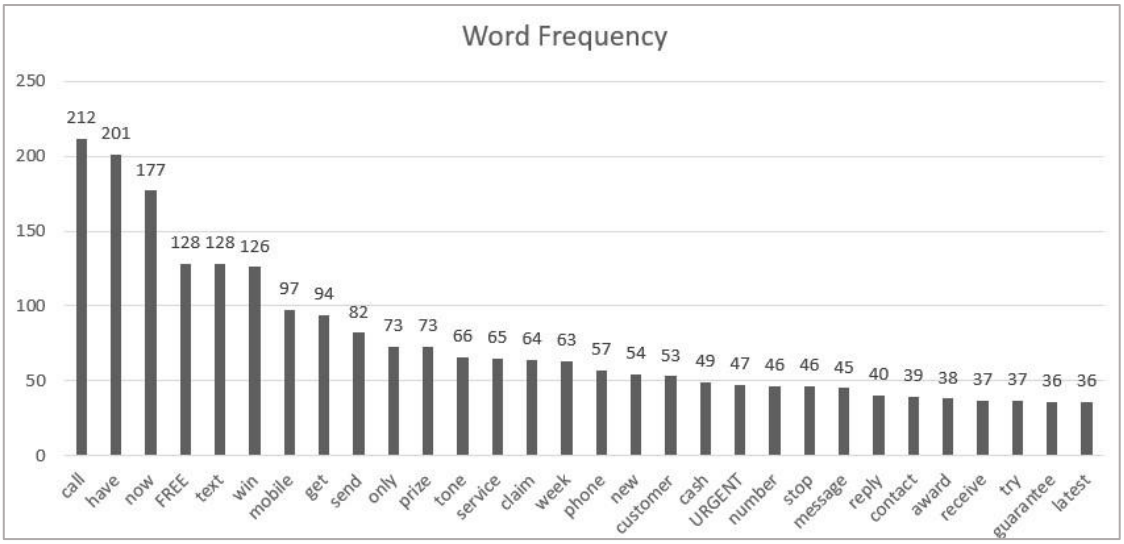

Figure 4 Spam message text mining

In addition, when the top 14 words and characters extracted from 56 newly investigated spam messages with a high possibility of remote control. The reason for this is that the latest smishing data is significantly smaller than that of existing public datasets, so we further investigated separately to identify the latest characteristics. Also, The attacker used a smishing technique in which the attacker requested a reply to the newly changed mobile phone number from the target. We found a total of 9 as a result of removing duplicate words from the entire dataset. The attackers also asked to minimize sending and receiving contacts via SMS and to contact users via WhatsApp social media messenger.

Figure 5 Remote Control Smishing message text mining

5.2 Message Detection Function Supervised Learning Classification

For keyword identification and smishing detection to prevent remote smishing in Android, we use the Naive Bayes Classifier model to prevent smishing. We chose the Naive Bayes classification model. because it can quickly and accurately classify a large amount of data compared to other algorithmic models (VanderPlas, J, 2016). Also, the most important thing to detect and prevent smishing is to be able to quickly filter even if the dataset is continuously added later.

Moreover, the collected data sets are converted into vectors based on the frequency of words using the scikit-learn library CountVectorizer provided by Python. The model is then trained using MultinomialNB, a naive Bayes classifier for multinomial models. Analyse the text by specifying two labels, spam message and ham, in the dataset. In addition, the machine learning algorithm model can be tested according to the frequency, and results can be derived. For model training, training and performance evaluation data are distributed at a ratio of 0.7 and 0.3, respectively. Then, using the distributed data, model learning, performance evaluation, and result analysis are derived.

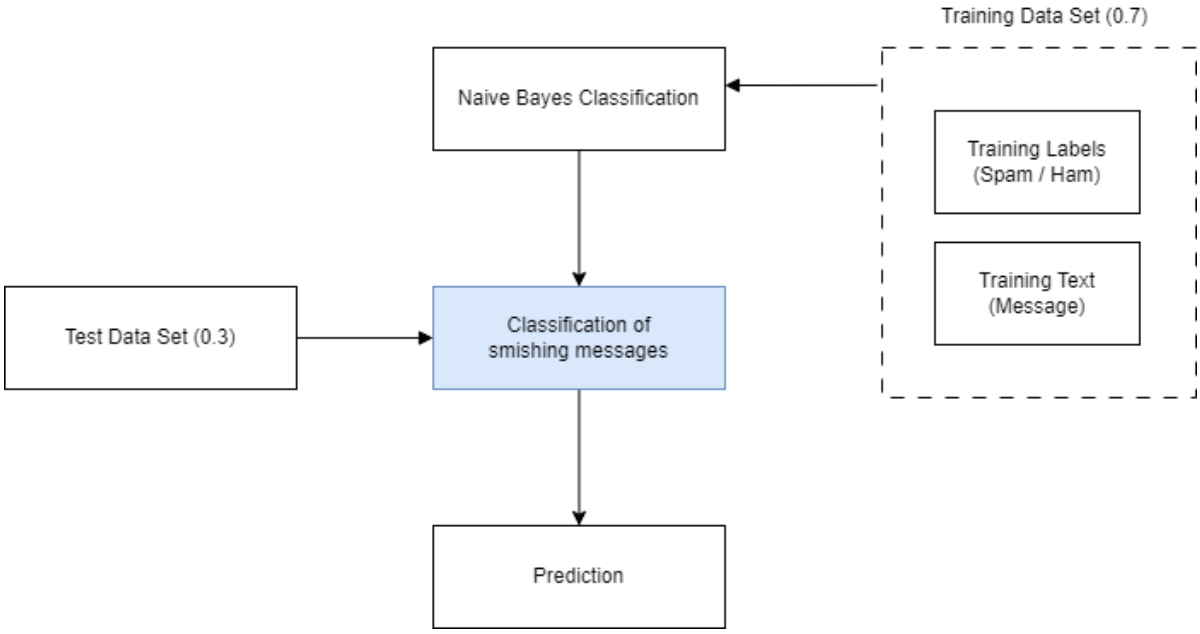

Figure 6 Naive Bayes Classifier

5.3 Analysis of Android application cases with remote control function

To prevent smishing from the user's point of view, we analyses how remote control smishing occurred in the Android environment based on actual cases in this study. Recently, cyber-attack methods have become increasingly sophisticated to the extent that smishing, which inserts a link to a digital wedding invitation and installs remote malicious code on a target's mobile device, has appeared. The example below is a remote-control smishing case in the Android environment using an application released on Google Play Store that occurred in October 2022.

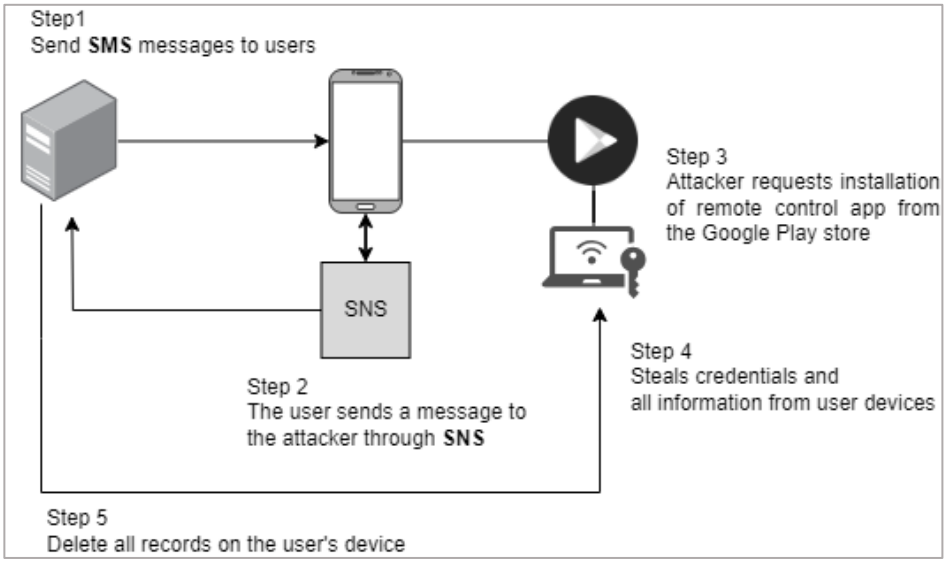

Figure 7 Remotely controlled smishing using applications released on the Google Play Store

Step 1 is a preliminary step for an attacker to select a target. At this point, the attacker sends messages to many random, externally harvested mobile contacts. At this time, the messages are mainly “Mom, my cell phone is broken. Contact me again via social media.” sent in a similar format. In this message, it is emphasized that it is an emergency, and SMS to SNS At this time, the attacker usually impersonates a family member and requests to send a message to SNS.

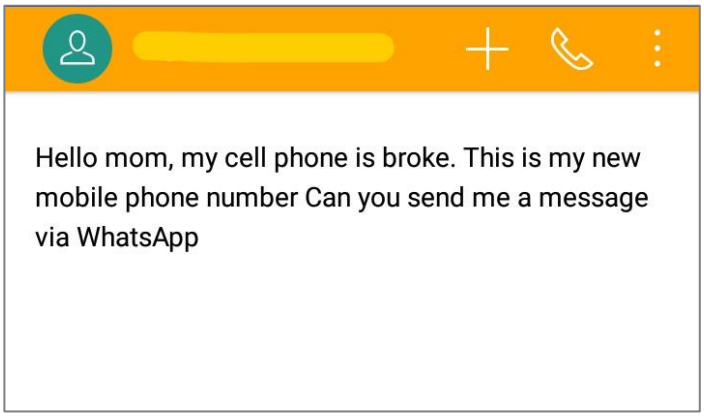

Figure 8 Remote control App smishing approach message example

Step 2 begins when the victim contacts the attacker via SNS. What is unusual is that before the conversation goes to SNS, "Why do I need to contact you via SNS?" If the user replies to the question, the smishing does not proceed any further and ends. However, "What is happening?" Questions like "Where can I contact you?" were linked to smishing.

Step 3 requires installing the remote-control application specified by the attacker. At this time, there are two types. Type A is a remote-control application that includes APK malware, and type B is an application that supports remote functions officially released on the Google Play Store. The case found in this study was type B and required TeamViewer Quick Support to be installed.

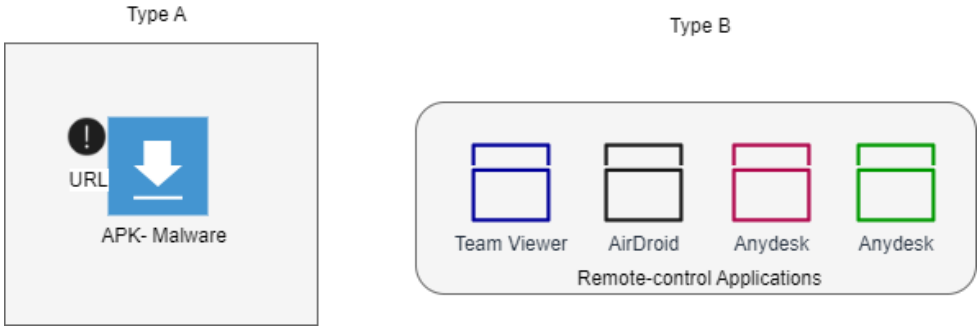

Figure 9 Remote Control Smishing Types

In step 4, if the victim installs the remote application, typically, the attacker steals the information in the mobile device. In particular, in the case of TeamViewer Quick Support, the attacker can easily control the user's mobile device with a computer if an ID is automatically assigned to the victim. Additionally, to seamlessly use the remote application, the attacker continuously tries to talk to the victim. "Almost done." "Show me a photo of your ID to claim the device's insurance." "Please keep it a secret from the rest of the family."

In Step 5, after all, information is transmitted to the attacker, the attacker deletes all records in the mobile device. Most victims were often unaware that their information had been leaked even at this stage, and they often found out that it was smishing after receiving the damage of completely withdrawing money from their account.

Through the case analysis above, we were able to discover that the reason why attackers prefer SNS in recent smishing is that it is easy to conceal the attacker's information. In particular, when smishing text messages with malicious remote codes are generally embedded in SMS in the Android environment, it is possible to block web addresses primarily through user settings. In particular, in the Android version 13 environment and earlier versions, when a message requesting to download a remote application comes in, a service that can block the function of connecting an external link to a web address is provided. Therefore, attackers prefer SNS channels that are easy to join and leave to avoid these distractions and attack flexibly.

In the case of remote smishing in the form of malicious code, it is possible to prevent it primarily by downloading and running an antivirus detection and deletion application released on the Google Play Store in advance. However, B-type applications are difficult to delete and detect with antivirus programs. Applications that include remote control functions released on the Google Play Store include TeamViewer, TeamViewer Host, TeamViewer Quick Support, AirDroid, Anydesk, and VNC. In particular, in the case of TeamViewer Quick Support, the user ID is given immediately upon downloading the application without any sign-up process.

6 Evaluation

In Chapter 6, we provide an evaluation of previous implementation steps. To prevent remote smishing, we implemented smishing measures in three directions: keyword detection through text mining, Naive Bayes Classifier model, and application vulnerability prediction and analysis through a remote-control smishing case study. Moreover, we evaluate keyword-based prevention implementation, algorithm model and remote smishing case analysis.

6.1 Keyword-Based Prevention

We tested how much it could be prevented by setting the 39 keywords found in the implementation stage of this study to an Android device. In this study, we set the top smishing keywords on two Android devices, randomly extracted 300 messages from each generation of Android devices out of 803 spam test sets, and sent them as bulk text messages.

When sending Bulk SMS in this study, we followed the process of sending text messages to devices using Twilio. Device A used Android version 8, and the Android version of the other two devices was set to Android 13, which was released in 2022. It showed an average blocking rate of 90%. As a result, the lower the Android version, the lower the block rate. There was also little difference between Android 12 and 13. However, this has a limitation that results may vary depending on random spam messages. This study found that the most important thing is that the higher the keyword setting, the higher the blocking probability. Therefore, it is essential to continuously identify trends in smishing data and periodically update keywords to prevent smishing.

Model	Android Versions	Keyword Setting	Spam Message	Block Message	Block rate (%)
LG	8	30	300	261	87.0

Table 2 Android device A

Model	Android Versions	Keyword Setting	Spam Message	Block Message	Block rate (%)
Samsung	12	30	300	273	91.0

Table 3 Android device B

Model	Android Versions	Keyword Setting	Spam Message	Block Message	Block rate (%)
Samsung	13	30	300	276	92.0

Table 4 Android device C

6.2 Machine learning algorithm evaluation

To evaluate machine learning algorithms, we analysed public smishing data, mainly used in existing studies and the data we investigated. The test data set was split between 0.7 and 0.3. Table 5 shows the model scores when dividing the existing public dataset by 0.7 for training and 0.3 for testing. The Count for the training data set of 0.7 is 3,900, of which Unique has 3,670. Moreover, the Train Model Score was 0.975.

On the other hand, there was a slight difference when using the combined data of all the collected datasets. According to the Naive Bayes algorithm, the training set (0.7) has a Count of 3939 and a Unique of 3717 words excluding overlapping sentences. Finally, the Count in the test set was 886, and the Train Model Score was 0.975.

As a result, when using the remote smishing spam data we additionally investigated, the detection rate decreased by about 1% to 0.975. However, we still found that the naive Bayes model showed a high detection rate for smishing characters in a simple yet powerful way through this result.

Count (0.7)	**Unique**	**Train Model Score**
3900	3670	0.984

Table5 Original public data set counts and model scores

Count (0.7)	**Unique**	**Train Model Score**
3939	3717	0.975

Table 6 Remote-control smishing training set model score

6.3 Remote control application smishing case analysis evaluation

As a result of evaluating the results of the analysis of remote smishing cases, if a user attempts to download an application installation that includes remote control possibility and high-risk functions in the Android environment, a more detailed and high-risk message should be used in the remote warning alarm notification. In the case of the current notification, it is a short message asking if the user wants to allow device A to support the device remotely.

A countermeasure for preventing smishing can be to make the user accurately recognize the possibility and risk of an attack. In addition, users need to sign up for a Google account to download the application. Compared to IOS, Google's account in the Android environment did not require any authentication and found a vulnerability where people can quickly create an account with fake information within 1 minute.

In the actual smishing case, there was a case in which attackers stole credentials, created a Google account remotely in the Android environment, and attempted payment by linking card information. Therefore, it is necessary to strengthen Google account creation in the Android environment.

6.4 Discussion

As mentioned in the Chapter 2 literature review, there have been various detection methods for smishing. According to the naive Bayes algorithm we used in this study, the smishing character classification method is one of the methods already discovered by other researchers. However, we tried to find essential keywords from data with a high possibility of remote smishing independently of previous studies. In this study, we discovered vulnerabilities by adding new smishing data and analysing text contents and processes used for remote smishing. In addition, most studies are meaningful in that only public data sets were studied, and recent and remote smishing data were not analysed.

In addition, we need to research ways to detect and prevent smishing without setting keywords on the user side and without downloading applications used for smishing detection. Our study focused on analysing whether keywords are meaningful in recent smishing and testing them. However, research needs to be improved to apply this to technology in practice.

7 Conclusion and Future Work

In Chapter 7, we will discuss the results of this study and future work. So far, in this study, we have discussed how to prevent remote control smishing in the Android environment. The study's goal of preventing remote smishing could be presented in three aspects. In particular, this study is meaningful because it can approach the Naive Bayes Classifier machine learning algorithm and analyse the frequency of words in smishing data in terms of data mining and discovering features. It is also a meaningful study that collected and analysed the latest smishing data and found 30 top keywords.

However, as a limitation of this study, the word remote control smishing tends to be recognized as the same attack as URL smishing, and it is difficult to distinguish clearly because remote control smishing is also combined with various cyber-attack methods. In addition, as can be seen in research cases, attack methods such as remote smishing malware and smishing methods using released applications with remote functions continue to diversify and evolve. Therefore, since there are security risks and limitations in discussing how to prevent remote control smishing with only one method, it is a problem that should be continuously discussed in various aspects, such as social engineering techniques, technical aspects, and remote application vulnerability investigations.

In order to prevent remote smishing, message detection and analysis are more important than anything else, and continuous research is needed due to continuously changing smishing patterns. In order to improve the results of future research, research on algorithms such as artificial neural networks and naive Bayes will be needed.

In addition, since it relies on smishing data, there is a limit that the analysis method that relies on keywords and machine learning can be dangerous in the vast amount of data that is continuously upgraded and changed. Also, since this study is to prevent remote-controlled smishing, it mainly focuses on the stage before smishing occurs. As a follow-up project of this study, it will be necessary to study how to solve and protect it after an actual remote control smishing occurs. And, Future research is needed to prevent smishing and improve the accuracy of message filtering in a situation where attack methods are constantly changing.

References

Alabdan, R., 2020. Phishing attacks survey: Types, vectors, and technical approaches. Future internet, 12(10), p.168.

Aleroud, A., Abu-Shanab, E., Al-Aiad, A. and Alshboul, Y., (2020). An examination of susceptibility to spear phishing cyber attacks in non-English speaking communities. *Journal of Information Security and Applications*, *55*, p.102614.

Almeida,Tiago and Hidalgo,Jos. (2012). SMS Spam Collection. UCI Machine Learning Repository. https://doi.org/10.24432/C5CC84.

Ansari, M.F., Sharma, P.K. and Dash, B., 2022. Prevention of phishing attacks using AI-based Cybersecurity Awareness Training. Prevention.
Azeez, N.A., Misra, S., Margaret, I.A. and Fernandez-Sanz, L., (2021). Adopting automated whitelist approach for detecting phishing attacks. Computers & Security, 108, p.102328.

Breda, F., Barbosa, H. and Morais, T., (2017). Social engineering and cyber security. In INTED2017 Proceedings (pp. 4204-4211). IATED.

"Count Vectorizer Library" Sklearn feature extraction 3.0, 2022. [Online]. Available: https://scikit-learn.org/ . [Accessed: 22-July-2023].

Chanti, S. and Chithralekha, T., (2022). A literature review on classification of phishing attacks. *International Journal of Advanced Technology and Engineering Exploration*, *9*(89), p.446.

Choi, Y. and LEE, J., (2021). THE CHANGE IN THE METHODS OF SMISHING IN SOUTH-KOREA AFTER THE ONSET OF COVID-19. Journal of Legal, Ethical and Regulatory Issues, 24, pp. 1-12.

Giovannitti, E., Mannella, L., Marcelli, A. and Squillero, G., 2019, June. Evolutionary Antivirus Signature Optimization. In 2019 IEEE Congress on Evolutionary Computation (CEC) (pp. 928-935). IEEE.

Goel, D. and Jain, A.K., (2018). Smishing-classifier: a novel framework for detection of smishing attack in mobile environment. In Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3 (pp. 502-512). Springer Singapore.

Hamandi, K., Chehab, A., Elhajj, I.H. and Kayssi, A., (2013), March. Android SMS malware: Vulnerability and mitigation. 2013. 27th International Conference on Advanced Information Networking and Applications Workshops (pp. 1004-1009).

Harnett, D. and Jones, W.S., (2023). Smishing vs. Phishing: Understanding the Differences– Global Security Mag Online.

Jain, A.K. and Gupta, B.B., (2019). Feature based approach for detection of smishing messages in the mobile environment. Journal of Information Technology Research (JITR), 12(2), pp.17-35.

Joo, J.W., Moon, S.Y., Singh, S. and Park, J.H., (2017). S-Detector: an enhanced security model for detecting Smishing attack for mobile computing. Telecommunication Systems, 66, pp.29-38.

Kamau, J. and Kaburu, D., (2022). A Review of Smishing Attaks Mitigation Strategies. International Journal of Computer and Information Technology (2279-0764), 11(1).

Kazmi, Z., Felguera, T., Vila, J.A. and Marcos, M.M., 2012, May. TASAM-Towards the Smart Devices App-Stores Applications Security Management Related Best Practices. In 2012 5th International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-5). IEEE.

Koski, T., 2021. Increase in remote work: effects on phishing.

Mambina, I.S., Ndibwile, J.D. and Michael, K.F., 2022. Classifying Swahili Smishing Attacks for Mobile Money Users: A Machine-Learning Approach. IEEE Access, 10, pp.83061-83074.

Mohamed, I. and Patel, D., 2015, April. Android vs iOS security: A comparative study. In 2015 12th international conference on information technology-new generations (pp. 725-730). IEEE.

Mishra, S., & Soni, D. (2023). DSmishSMS-A System to Detect Smishing SMS. Neural computing & applications, 35(7), 4975–4992. https://doi.org/10.1007/s00521-021-06305-y

Park, D.W. and Seo, J.M., (2007). A study of information leakage prevention through certified authentication in phishing, vishing, smishing attacks. Journal of the Korea Society of Computer and Information, 12(2), pp.171-180.

Petroc Taylor, (2023). Smartphone subscriptions worldwide 2016-2021, with forecasts from 2022 to 2027

Sandhya, S., Srivastava, H., Kaila, T., Tyagi, A. and Gaikwad, K., (2020). Methods and tools for plant organelle genome sequencing, assembly, and downstream analysis. Legume genomics: Methods and protocols, pp.49-98.

Scofield, M., (2016). Benefiting from the NIST cybersecurity framework. Information Management, 50(2), p.25.

Shahriar, H., Klintic, T. and Clincy, V., (2015). Mobile phishing attacks and mitigation techniques. Journal of Information Security, 6(03), p.206.

Shravasti, S.S. and Chavan, M.,(2021). Smishing Detection: Using Artificial Intelligence.

“Software for text mining” KH Coder Version 3.0, 2022. [Online]. Available: https://khcoder.net/en/. [Accessed: 22-July-2023].

VanderPlas, J., 2016. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".

Verma, A. and Shri, C., 2022. Cyber Security: A Review of Cyber Crimes, Security Challenges and Measures to Control. Vision, p.09722629221074760.

Yatsyk, T.P. and Shkelebei, V.A., 2018. Investigation of new forms of cyber crime (phishing and cybersquatting).

Yeboah-Boateng, E.O. and Amanor, P.M., (2014). Phishing, SMiShing & Vishing: an assessment of threats against mobile devices. Journal of Emerging Trends in Computing and Information Sciences, 5(4), pp.297-307.