Analysis of Spam Activity over Instant Messaging App

Xinhao Luo, Qihang Zeng, Xi Cai, Bowei Wang
University of California, San Diego
San Diego, USA
{x4luo,qizeng,x9cai,bow007}@ucsd.edu

Abstract

Instant messaging has became an important part of people's daily life. Unlike text messaging and phone calls, instant messaging App has a lot more features than these traditional services. At the same time, spammers are also into this growing field. In this paper, we go through various kinds of spam activities identified in a specific, popular instant messaging app, telegram, and discuss the motivations behind them, and potential defense mechanisms over these spam activities.

Keywords: Spam, Instant Message, SPIM (Spam Over Instant Messaging), Botnet, Telegram, Defense

1 Introduction

Spam filtering has been a well known problem thanks to wide attention to email spams. However, more and more people begin to use instant messaging applications on their mobile devices to communicate with peers while spams on instant messaging platforms remain unnoticed most of the time. In this paper, we try to understand the spam model on instant messaging platforms and we propose some defense practices to address the challenges in this new setting.

Spam messages, in our context, are defined as unsolicited and unwanted junk messages sent to individuals or groups. Usually they are sent in bulks via some automatic process, motivated by commercial reasons. We collect data using the "unwanted" principle since users and administrators ban account mostly because of the message spammers sent is unwanted for themselves or the communities they managed, and this is the standard we use in this paper.

1.1 Related Spam

1.1.1 Email Spam. Email spam is sent in bulk to large number of people. Most messages are advertisements in nature, some include phishing url linked to a phishing or malicious website, other spams include email spoofing, money scams and etc. Organizations usually use spam filter system to classify such messages. Some countermeasures include DNS-based blackhole lists(DNSBL), spamtraps, DKIM, SPF, spam detection model etc. These measures are usually effective, however, due to the different implementations of email transmission protocols and web interface, attackers can bypass email security protocol's verification. Kaiwen etc.[5] identified new attacks for email sender spoofing capable of bypassing SPF, DKIM protections. Providing URL is also important part of the spamming. Multiple techniques including cybersquatting, typosqauttings, and homographs are usually used.

1.1.2 SMS Spam. SMS spams are messages sent by adversaries which try to get you to click on a link or divulge personal details which has been mentioned above. This type of spam includes unsolicited advertising, inappropriate or adult-themed content, premium rate fraud, smashing, mobile malware, etc. But what makes the situation worse is that people are more likely to open the SMS spam than the email spam and will open the spam much faster than the email spam. Since the SMS spams affect so many people, both the mobile network operators and the government keep warning people that they should be careful about the spams.

There are several ways to help the users to protect themselves from SMS spams like asking for the help from telecommunication companies. At the same time, people now also use machine learning ways to identify spams. Tiago et. al.[1] used the SVM(Support Vector Machine) and several other established machine learning methods to identify the spam.

Instant Messaging Spams have included most of the characteristics mentioned in the above spam types. For example, most of them would take advantage of link, media to trick users or promote certain brand/ideas. However, as IM also has more surfaces, there are many more ways to penetrate. We discuss it in more details in the later sections.

1.2 Related Works

There hasn't been a lot of research work of spam detection in the context of instant messaging. A fraction of reports and papers touched on the topic with only very limited scope and most of them have only done some exploratory analysis.

Nobari et. al. [2] try to classify spam messages in Telegram dataset. In their work, they mainly focused on detecting spam messages using features of the message itself. This include number of mentions in a message, timestamp of a message and whether or not the message is a forwarded one. Then a couple of different ML models are applied to classify spam messages. The work focused on a subset of features of messages and does not take into account other user behaviors in Telegram. The types of spams and account activities have complicated the issue today. The landscape of spam activity becomes more intractable, and content-based detection methods become less effective.

Liu et. al. [3] take a different perspective on spam detection. They proposed a framework for instant messaging

spam detection. In their work, they summarized different spam detection methods and try to integrate them into a single application. In addition to traditional ML methods to classify spam messages, they also identified components of a messaging application such as sending rate control and challenge-response control where rules could be applied to limit the impact of spam messages. For ML methods, they mention an interesting P2P based approach which allows spam knowledge to be shared in the community.

Some other research focused on chatbot detection. This is one popular form of sending spam messages nowadays; however, as we will show below, the landscape of spam models on instant messaging platforms is much bigger than just chatbot. Examples of detecting chatbot includes McIntire's work [4] in which messaging patterns of chatbots are analyzed. In addition, they proposed a series of active defense strategies including using CAPTCHA, response challenge (interrogator) etc.

1.3 Challenges

This study may involve real human activities as we are looking into an active, popular messaging app. However, we are mostly observing behaviors from groups and we could also setup our own groups trying to track these behavior.

Telegram has provided rich APIs and has open sourced their client app so that developers could easily build their own app. However, all members in our group still need to get familiar with the Telegram platform and the SDK language we are about to use – Python, Go and some database knowledge on how to efficiently gather and analyze data.

1.4 Evaluation

We expect that we have a through understanding of the spam activities in general on Telegram, with solid data to back up our theory. Getting some effective ways to fight against these behaviors would be a plus, but it is not strictly required to be "successful".

2 IM Components

2.1 Message

Message is the basic element in Telegram. It could be text, picture with captions, a group of pictures and videos, files, voice messages, etc. Message usually has its unique ID and could be forwarded, downloaded, or edited. When forwarded, the original sender's name will be attached. Message being viewed will also be marked with a two ticks and for the channel (mentioned below), it will have a unique viewer count and optional signature from the administrator.

2.2 Chat

A chat could be either secret or standard. Usually, this is a peer to peer communication, where only two individuals are involved in this element. Secret chats destruct historic chat records automatically after some preset time. While standard chat messages are not restricted, secret chat messages could be limited e.g. cannot be forwarded to other chats. Secret chat also exercise end-to-end encryption.

2.3 Group

A group is a place where two or more people gather in one place. group members could broadcast their messages to other people in the group. A group could be either private or public. The public group would have a globally unique name that could be easily referenced, and messages could be viewed even if the user is not within the group. Messages inside could be reference using generic link (t.me/${group unique name}/${Message ID}) and public searchable on the Internet. However, private group users could only read messages or interact with it when they have joined the group. In addition, chat bots could be added to a group to provide fancy features.

2.4 Channel

Users could subscribe to channels to his/her interest. Unlike groups where users might send and receive messages, channel is a form of the publish-subscribe model. The organizer of a channel creates new content from time to time and subscribers would receive the content. Some common genres include news feeds and stock market updates.

Any individual could create a new public or private channel, and the channel owner could assign other users or bots as administrators. The procedure is mostly the same as signing up a new user.

2.5 Profile

Upon registration, a user provides a phone number, which is secret by default. Each user would have a profile shown to others when either clicking from a common group or searching by username if they have set it up. The profile would show the first name and the last name user specified when registered. The user could upload a profile picture and pick a username as a unique identity globally searchable within the IM. At last, the profile page allows users to present a short biography about themselves. All this information including username is changeable for users.

3 Threat Model

3.1 Motivation

There are different groups that use spam for their interests. Here we categorize them as the following.

1. Advertisements. Attackers try to blend in a group. Then they broadcast advertising messages in the group. Some of the popular messages are related to cryptocurrency community, gambling, porn, money laundering, shopping(food deliveries etc.), immigration services, malicious part-time jobs, etc.

2. Phishing. Phishing is more popular through direct messaging. An attacker could use various techniques to masquerade itself as being some official agent (with the help of profile and hyperlinks that are indistinguishable from the real ones). Phishing could result in stealing passwords, personal information, cryptocurrency, etc. Other spam messages could include text or images containing violence or pornography content. The spammers may also fabricate stories asking for money or personal information to collect a prize.
3. Disruption. Some Spam Activities involve competition between different organizations/groups of people. By disrupting or dis-form their competitors, they will go to existing groups or their competitors' groups and do unwanted behavior so that people will leave or move to different communities.

There are many attack models we have observed during the normal usage of the IM. There are some platform specific problems, while most of them we believe would be common issues for other competitors: for instance WhatsApp.

3.2 Group Message Spam

Most of the time, spammers tried to spread their malicious business to more potential groups of users, and group chat where most people would receive and be involved would be an ideal place to get started. There are many variants instead of simple message spamming during this process.

3.2.1 Service Message. There is a special type of message named Service Message being broadcast in the group, during some special events. For example, when a user joins the group, there will be a message like "$name has joined the group" (Figure 1). This feature is being used for spammers who name their accounts with spam content and join groups in batches.

Figure 1. Spam Activities using Service Message

3.2.2 Edit Message/Name/Avatar. Some groups may have already implemented some bots with mechanisms like Keyword detection when it is the first time user joined the group and sent their first message. However, spammers make use of the edited message functionality so that they could first send a legit message and change it into some spam content after joining the group. This method also applies to service message spam if the user joined with a legit name or avatar and changes it later.

Figure 2. Link Preview

3.2.3 Link Preview. Telegram also has a nice feature of fetching links content and showing it within the chat, so users could have an abstract of what's on the link before clicking on it. However, this feature is also being used by spammers as a short link could explode to a pretty long message with a preview shown on users' UI, and the web preview content is not visible to the bot — what they can only see is the link itself. The link preview is shown in Figure 2 in the red box.

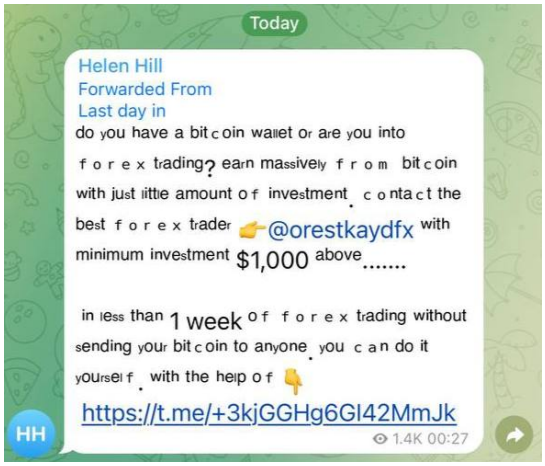


Figure 3. Variant Words

3.2.4 Variant Words. While some spam messages can be detected through keyword searching, a way to bypass is using variant words. For example, the spammers can use "@" to

substitute "a". Also, they can mix English with other language like German or Spanish that include some similar characters to English. In this way, using the variant words, they can bypass some keyword searching. The Figure 3 shows a variant words example.

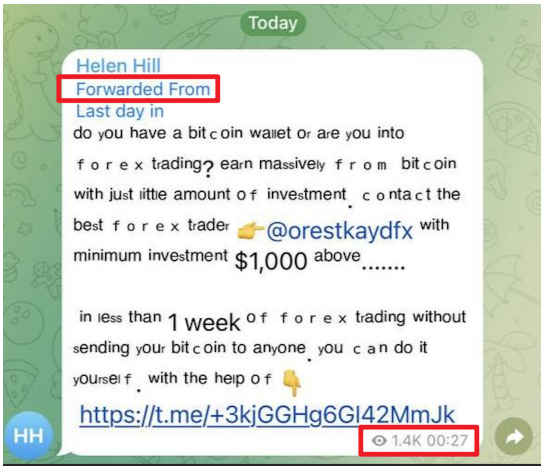


Figure 4. Forward Message

3.2.5 Metric Collection. One issue where spammers have to deal with is the number of viewer of their messages. It is possible that they send a link and see how many people clicked on it, but it is hard to tell if they have successfully reach enough people, so they make use of a feature provided in Channel: every message send in the channel will have a view count, and the counts also applied when the message is forward to other places, and it counts the number of unique viewer, which is a good source for judging how effective a spam activities would go.

3.3 Point to Point Spam

On Telegram, Chat is also an important element where users need to talk with each other privately sometimes. However, spammers could also make use of it to send spam messages. Unlike group message spam, which may not be able to grab users' attention even though being viewed as new messages are coming in constantly, or it may be deleted before actually viewed by a real human, a private chat is always guaranteed to be viewed by human and has special notification. Telegram right now does not have an allowlist mechanism for private chat, though there are some limitations imposed. For example, users have to be within the same group before sending messages. Still, there are many ways to work around it.

3.3.1 Group Indexing. As we have mentioned before, there are many projects and resources collecting various popular group chats and channels. Spammers first create the account and join these groups, and then iterate all users within the same group. It could be done either by going through chat history or directly from the group member list. Sometimes they go from history to avoid sending messages to inactive users. From here, they could send messages to users without limitations. Usually, they will implant accounts batch by batch so that it would regularly have many users hiding in these groups and collecting data.

3.3.2 Username. With a username, such limitation is no longer imposed. A username is a unique id and once it is acquired, spammers could regularly send spam messages as long as it is still pointed to a user. This means that a user who creates an alias opens a permanent door for spam. However, it is intended behavior as the username is publicly available to all users for quick identification.

3.3.3 Forced Group Creation. Another way to send the spam message is a "forced" creation of a group. There are two ways to "invite" other people to join a group. One is via other groups, the other is via username. After "inviting" (Figure 5), the user will be in the group automatically, even without asking for the user's agreement. Therefore, the spammers can send spam messages in the new group and the users will see them. Therefore, in this aspect, the users are "forced" to join the new group and suffer from spam messages.

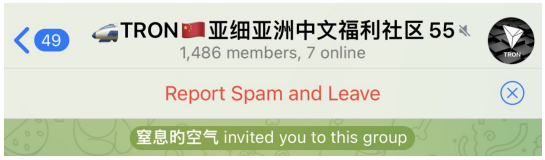


Figure 5. Group Invitation without consent

3.3.4 Account Credential Enumeration Attack. When someone uses his or her phone number to register the Telegram, the Telegram will store it for contact matching purposes so that if other people have the same phone number in their contacts, when logging into the Telegram, the Telegram will notify both sides that there someone on their contact is now using Telegram. This will cause a problem that malicious users can do the account credential enumeration attack using this feature. The spammers can brute force the phone numbers, and add them to their contacts. If the Telegram shows that this phone number has been registered, spammers can send spam messages. Also, it can be used to match the people and the phone number, which can be stored for later regular spamming activities — unlike username, one could hardly change their phone number or their user id. Figure 6 shows an example that Telegram shows the phone number of an account who sends us a spam message. This account does not have common groups with the account we owned while Telegram considers this account as our contacts.

Figure 6. Account Credential Enumeration Attack

3.4 Fake Identity

On Telegram, it is really easy to fake the identity. A user's profile is public. As we mentioned above, the profile is one of the IM components and will show the user's profile picture, username and biography. In all parts, the only unique thing is the username. There will be a unique ID per user but now the official Telegram client will not show the ID. So, if we want to fake someone's identity, we can copy their profile picture and the biography. As for the "unique" username, we can find other letters to confuse others. The naming convention of the Telegram is using a-z, 0-9 and underscore and it is not case sensitive. Therefore, we can use capital "i" to substitute the lower case "l" and vice versa. For example, in Figure 7, it shows that although the two channel name looks like the same, they are different. In this situation, it is very easy to fake someone's identity and do things like phishing, scamming etc.

3.5 Malicious Clients

3.5.1 Send malicious message. We have also found that some crypto Apps make use of the Telegram and embed it into their Apps so that it gets free chat functions directly in their Apps. At the same time they also make some custom functions e.g. Red envelope, Airdrop functions that allows user to make transactions directly via Chat. However, the way they implemented this kind of feature is by sending text message with special format (Figure 8). This kind of message is generated by those special Apps and can only interpret by the corresponded Apps, while at the same time other users on Telegram is unable to interact with it and this is considered as Spam by most groups that originated on Telegram. At the same time, users who are on those Apps may not aware of this and have no idea why they are banned.

Figure 9. Malicious client detect and change text message when crypto address is detected.

3.5.2 Detect and Swap. This kind of attack may be a bit out of scope but are still active on the platform, so it would

Figure 7. The top two username are fake accounts that tried to phish people with "pay to remove from blacklist" and gain profit from it. The original username listed below is a volunteer powered community blacklist service on Telegram

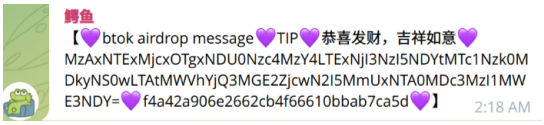


Figure 8. Malicious client send special formatted message in the group.

be great to cover this issue in our paper. For now, there is no reliable way to detect such behavior, especially Telegram has open-sourced its database, TDlib[1]. This allows developer to quickly create a client without knowing how to interact with network, message retrieval, storing, etc. At the same time, these malicious on the clients can do anything on users' behalf, like modify user's message, or hide/censor result. One of the example (Figure 9) show that some third-party build websites and clients with localization to lure user to download and use, turn out to detect any crypto wallet address and swap with third-party address during the chat.

3.6 Banned on purpose

Sometimes, instead of making profit, one would just want to destroy a community by getting it banned by the official. The reason to making something get banned on Telegram would be the following:

1. Spam
2. Violence
3. Child Abuse
4. Personal Details
5. Illegal Drugs

An interesting fact to be mentioned would be that generic porn will get "banned" by being tagged with a special mark so that clients published on Apple App Store, by default, cannot be able to view them. However, if users log in with a different client, e.g. Web Client, media will be shown without issue, and they could even change their settings to allow Apps on another platform to view these messages. This setting must be changed on different platforms. For all the other kinds listed, those messages, if found in the group chat, will be banned by the Telegram. Group owner will be notified that their group will be inaccessible (Figure 10) until those messages are deleted. If it happened multiple times, Telegram will permanently ban the group. Thus, one tactic used would be some people, creating a bunch of accounts and joining the group, interacting like normal group members, and then some accounts send child porn pics and videos to the group chat and the others report them. At the same time, this account will also try to flood the group with messages, faking some conversations so that the admin could not identify messages of this kind at the first time, and ends up warning or banned by the official. Sometimes, even though a group may not be able to get a permanent ban, these disgusting materials will scare people and cause people leaving groups.

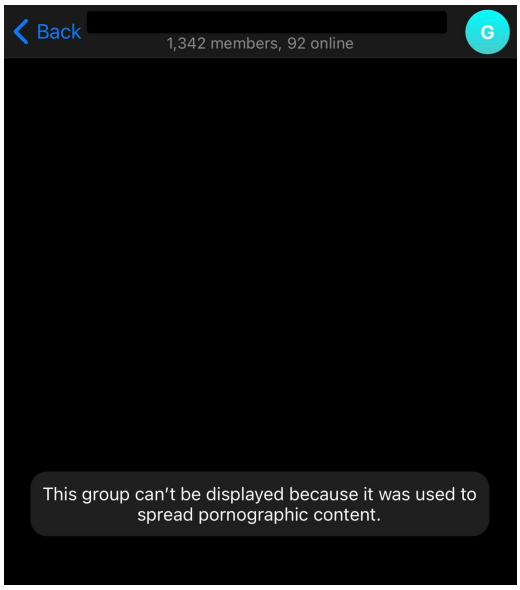


Figure 10. A Group being banned because of porn content, although this level means Telegram still allow recovery if the message are removed later

4 Defense Discussions

In this section, we would like to show some data analysis (Collection details in Section A), as well as some defense mechanisms based on the analysis. To be more specific, we would like to propose a Pro-Active way that is not formally nor widely discuss in various studies. Traditionally, most of the spam detection will only happen after the message was delivered. However, in the world of IM, before the first message is delivered, there is a gap between joining groups and the harm or say, the spam message is actually sent out and spread. We may make use of this gap more effectively to prevent further damage. Pro-active could also save resources than traditional spam filters since once spam/unwelcome activities is detected, a permanent ban could be issued and there is no need to process the rest of the chat messages. Also, in this paper, we may be more talking about automatic and batch activities. Spam operated by a real person behind each account in our opinion is pretty hard and impossible without huge funds.

4.1 Dimensions

When a user is joining the group, as we discussed before, a service message will be delivered. At this time, there is no spam message being sent yet. However, the user profile at this time would expose some useful information for risk analysis. Those information includes:

[1] https://github.com/tdlib/td

1. First and Last name
2. Bio
3. Avatar
4. Username
5. User ID

Except the traditional Service message spam, we would like to discuss some special characteristic happened in IM.

4.1.1 Information completeness. After registration, by default, Username, Bio, and avatar will be left empty. Parts of the spam accounts, in order to operate more efficiently, would not fill these optional fields. It is also observed that since some groups use this simple trait to target spam activities, more and more spam accounts would spend more time filling these information, including but not limiting using the fake data library to generate a name that seems to be real (e.g. python faker library).

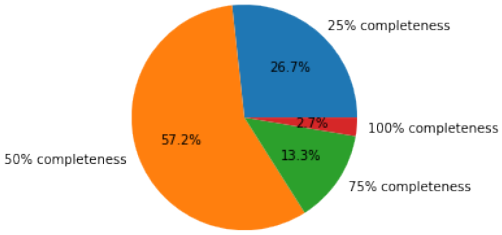


Figure 11. Profile Completeness of Spam Accounts, We consider non-empty string in those field as completed.

Figure 11 shows the profile completeness of some accounts we sampled from spam detection logs. We considered four profile fields: first name, last name, biography, and username. It is not surprising that only 2.7% accounts filled up all the fields we considered. More than half of the accounts only have 50% completeness. Notice that due to the lack of fields when fetching data from the API, we were unable to take into account other useful information such as avatar and avatar histories.

There is a special trait we would like to mention for Telegram, which we have observed that haven't been actively monitored, is the number of history avatar. Some user, as time goes by, would update their avatar, and Telegram by default will keep all history avatars. This would be a plus to identify an account that is operated by a real person.

4.1.2 Account Creation Date. Due to the fact that spam accounts may inevitably interact with real people and be identified and reported to Telegram, some limitations would be imposed once verified. In order to remove such limitations, spammers could choose to completely delete the account and create a new account by batch. However, each phone number, after deletion, will not be able to register again for 14 days. Spams often demand, on the other hand, not to have such gaps. If we know that an account is created only recently and a spam is reported or detected, we could be more confident that this is likely to be a spam account.

However, Telegram itself does not provide such information on its profile. On the other hand, we have the User ID, which is an int64 and unique to each user, generated by Telegram when a new account is created. It is also believed to be self-incremental, so if we could have some anchor e.g. some user ID with their creation date, interpolate the data between, we could roughly have some idea of when an account is created. Here we would like to show some data about the time between creation and the time they were caught sending spam:

We can see from figure 12 that most of the spam happened during the first year of the account creation. To be more specific, most of them are in the first six month from the creation date. The largest portion is happened within the first month, so it is a good sign of risk if the account is too new since spam activities are usually around new accounts.

4.2 Client Interactions

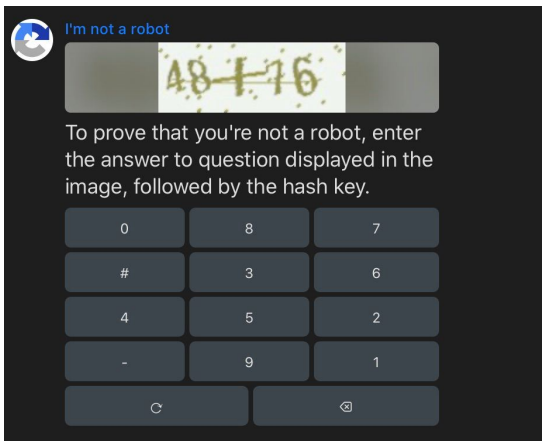
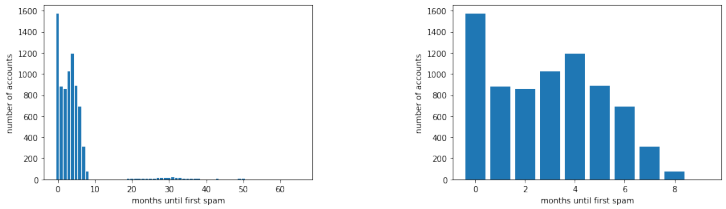


Figure 13. A CAPTCHA Bot using Telegram inline-keyboard feature

To identify automatic activities, there are solutions like re-Captcha on the web. Of course, IM could also make delegate this problem to open a website using the browser and get results accordingly. However, we would like to introduce some alternative ways that would be less intrusive and more user-friendly in the world of IM.

Since Telegram has open-sourced its implementation of clients, as well as SDKs, it is never been easier to create automation over this platform. However, implementations also need to be handled properly by developers — there are various functionalities that Telegram could do about chat messages: inline-keyboard, voice message, stickers, etc. We observed that some of the accounts do not handle it correctly.



(a) Full Spam Analysis **(b)** Zoom in for the first year

Figure 12. This graph show the number of account collected in our dataset, aggregated by the date between our estimate creation date from user id, to the date recorded being identified as spam.

What they could do would simply join the group and send the designated messages, while no response to other events. In this case, we could make use of this flaw and identify these clients. For example, asking the user to select the answer to a simple question using the inline-keyboard feature (shown in Figure 13), or establishing a voice call request (shown in Figure 14). These methods would be able to fingerprint clients by testing their capability and calculate risk accordingly.

4.3 First Message Characteristics

Message characteristics in different groups could vary a lot, while it is worth noting a more specific stage happening in the IM platform: the first message in the group chat. Usually, spammers only expected that their first message would be successfully sent out, and it is likely to be identified and cause a permanent ban, which effectively makes this account invalid. To make the most out of the first impression, they usually try to make this message take as much screen space as possible by adding new lines, media, links, special characters, editing the message, and forwarding messages (Section 3.2.5), etc. (Figure 15)[2]

However, in normal conversation, this is not happening very often. To make use of these characteristics, we could effectively detect the first message traits and guide normal conversations more effectively (Figure 16).

Figure 16. Group bot that guides new members send their first message to introduce themselves

This method also could be helpful to identify inactive accounts that only join groups for collecting group member information: these accounts once joined, will avoid any activity that will cause a ban or removal. No message will send out once joined hence hard to detect at the same time.

4.4 Sudden Death/Flood

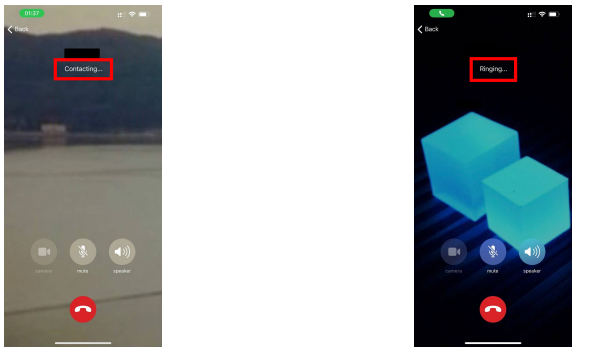
For this type of attack, it could be easily handled by limiting the interval an account could send messages, and detecting duplicate messages in the group for flood situations. Right now Telegram has implemented the "slow mode" into their Group management system while flood could be detected by filtering similar messages. As it already has mature solutions, we won't discuss too much about this type of attack in our paper here.

4.5 Username/public protection

In order to pin point the source of most point to point spam activities, we have done the following experiment:

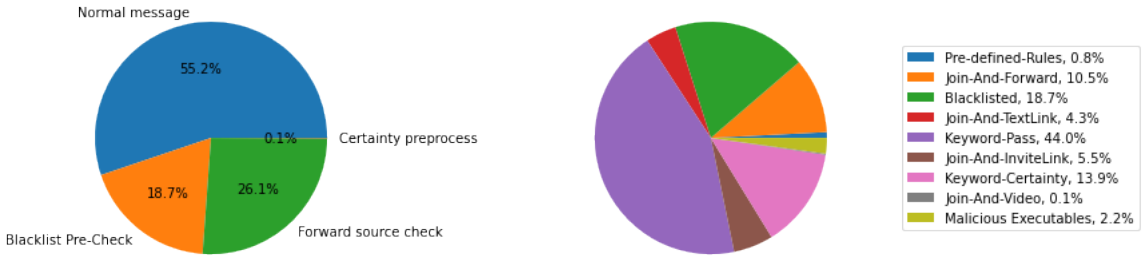
1. We create two clean account, and waited for a week or two, making sure there is no point to point spam activities observed is happening
2. We have one account A, set up with a username X, and make it join into various large public groups
3. We observed that after a week or so, account A has getting various point to point spam messages.

[2]Source/Reason graph explanation: Keyword-Certainty means that the spam word is pretty unique e.g. phone number, links, etc. once found in the message, it would be identified as spam. Keyword-Pass means that the message matches multiple spam rules. E.g. containing sensitive words, making use of links, and too many symbols, etc. which would be identified as spam. Blacklist and blacklist pre-check means that the ID is in the spam database before and found trying to enter groups other than the one it is being originated banned from



(a) Abnormal Call Stage **(b)** Normal Call Stage

Figure 14. This screenshot shows that different behavior on clients handing voice call. The left one is known to be a spam bot and when requesting call, its call state will stuck at contacting, indicating the client is unable to handle this event. However, for normal clients, shortly after contacting state, it will switch to ringing state where the client indicating it is able to establish voice connection and notify user about this event.



(a) Banned Source Graph **(b)** Banned Reason Graph

Figure 15. This graph show the percentage each source and reason for banned spam account. We can see that reason "Join-and-*" action has taken around 20% of the activities. For Source, forwarded spam message take around 30% of the spam activities collected.

4. At this time, we removed username X from account A, and set up the same one to account B, which is another clean account we have created before, and set up username Y for account A
5. It is observed that some point to point spam activities are happening on account B as well
6. However, it is observed that the activities ceased after a day or two. The number and frequency is way less than account A at the same time, which approximately the same as before.

It is also worth mentioning that Telegram has functions that show whether a user has the same group with you, but none of the spam accounts has the same groups during our experiment.

Based on this result, we believe that people who want to have more private space on IMs should in general set up multiple accounts and limited the attack surfaces. One public one could be used to explore various groups and contents while others could stay more in friends and private groups.

4.6 Federated Spam Detection

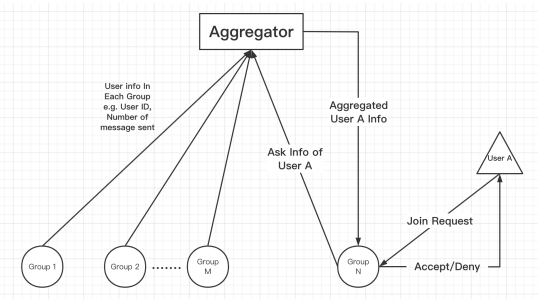


Figure 17. Architecture of the Federated Spam Detection

In our previous section, we provided various angles to tackle spam activities in individual ways. However, the IM platform is not just a single group, and spam activities usually would appear in multiple different communities at the same time. As in Figure 15, around 20% of the event is directly

banned because of the blacklist, which means the account is reported as spam in other groups. However, directly sharing the User ID with other groups may not be a good idea since other group administrators may not be able to know the exact reason why they are banned and different groups would have different spam tolerances. Some spam rules in one group may be too strict and some may be too relaxed. Therefore, we would like to propose a design that is intended to provide group administrators with useful information to make decisions, and at the same time, their decisions could help the whole IM community to identify spam activities.

The main idea of the federated spam detection is about information sharing across different groups. Figure 17 shows the architecture of the federated spam detection. The most important thing in architecture is the "Aggregator" part. All information shared across the groups will be processed and stored in Aggregator. Groups will submit user info data to the Aggregator. The user info data includes Group ID, User ID, number of messages that the user had sent in the group, the user's status like whether the user has been banned in the group or not, etc. When a new user A wants to join a new group Group N, Group N could ask for the information about user A. Then the Aggregator will return the aggregated user A info. Those may include the total number of messages sent, dates users join groups, if being banned, the banned reasons, etc., and the Group N Administrators, based on the data from other groups, decide whether it should let the user A join the group. During the whole process, the information shared between groups and the Aggregator doesn't contain any actual chat content that the user has sent. Instead, groups will only share some abstract information in order to protect the users' privacy.

4.7 Limitations

There are several limitations over analysis and mechanism described above.

4.7.1 Malicious Clients. We have also observed that some accounts, that seem to be used by a normal user, will sometimes spell spams. It is mostly believed that they are using malicious clients and these clients will hijack credentials and spam on user's behalf. There are no effective ways to stop this kind of behavior from the methods discussed in this paper. Mostly it has to be stopped by the IM platform. For Telegram, for each third-party apps, there is an appid [3] and its token so Telegram could ban malicious clients. Unfortunately, this token can be easily extracted[4], so that malicious will be the same identity as Telegram Official Web Clients.

4.7.2 Data Accuracy. For this paper, some of our analysis is pretty subject and doesn't have enough data backed up. It is mostly due to the limitation of the time span we worked on this topics, as well as the unpredictability of spam activities. Fortunately we have some history data of spams but in order to observe the full path and active methods, as well as keep track of all related elements information, we could only do this by hands for now. Some kind of information e.g. Service Message is not tracked in our system, and only observed by hands as well. A more effective message data collection should be proposed.

[3] Creating your Telegram Application
[4] Telegram Web Github Link of APPID

Acknowledgments

Many Thanks to Prof. Deian for the discussions and improvements, and spam data provided by the volunteer community.

References

[1] Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the Study of SMS Spam Filtering: New Collection and Results. In *Proceedings of the 11th ACM Symposium on Document Engineering* (Mountain View, California, USA) *(DocEng '11)*. Association for Computing Machinery, New York, NY, USA, 259–262. https://doi.org/10.1145/2034691.2034742

[2] Arash Dargahi Nobari, Negar Reshadatmand, and Mahmood Neshati. 2017. Analysis of Telegram, An Instant Messaging Service. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 2035–2038. https://doi.org/10.1145/3132847.3133132

[3] Zhijun Liu, Weili Lin, Na Li, and D. Lee. 2005. Detecting and filtering instant messaging spam - a global and personalized approach. In *1st IEEE ICNP Workshop on Secure Network Protocols, 2005. (NPSec)*. 19–24. https://doi.org/10.1109/NPSEC.2005.1532048

[4] John P. McIntire, Lindsey K. McIntire, and Paul R. Havig. 2010. Methods for chatbot detection in distributed text-based communications. In *2010 International Symposium on Collaborative Technologies and Systems*. 463–472. https://doi.org/10.1109/CTS.2010.5478478

[5] Kaiwen Shen, Chuhan Wang, Minglei Guo, Xiaofeng Zheng, Chaoyi Lu, Baojun Liu, Yuxuan Zhao, Shuang Hao, Haixin Duan, Qingfeng Pan, et al. 2021. Weak links in authentication chains: a large-scale analysis of email sender spoofing attacks. In *30th USENIX Security Symposium (USENIX Security 21)*. 3201–3217.

A Data Collection

Our current message collection is mostly around ZH-CN community where there are some anti-spam project going on. Via Telegram Bot API, we have successfully gathered some data, mostly in 2019, some spam message and activities in the community. Each entries has the follow structures:

```
    {
        'date': '2019-05-27T23:20:16',
        'forwarded_from': None,
        'text': '低价出pua全面教程',
        'photo': False,
        'detail': {
              'group_title': 'SSR交流',
              'admin_name': '',
              'msg_user': {
                'uid': 726034457,
  'approx_create_datetime': '2019-12-01T23:34:12.678335',
                'first_name': 'liulang',
                'last_name': 'liulang0612',
                'user_name': 'liulang666',
                'description': '我有一首歌足以度
风尘'
              },
            'source': 'Blacklist Pre-Check',
            'reason': 'Blacklisted'
        }
    }
```

We added the field of "approx_create_datetime" using our interpolation method, and some other non-informative fields are ignored. "msg_user" is the profile info about the account and "text" field is the message, if not only media, text content of the message.

This dataset is not publicly available yet and would provided upon request.