On the Exploitation of Online SMS Receiving Services to Forge ID Verification

Md. Hajian Berenjestanaki
mhajianb@ut.ac.ir

Mauro Conti
conti@math.unipd.it

Ankit Gangwal*
ankit.gangwal@phd.unipd.it

ABSTRACT

Communication service providers (e.g., *Whatsapp*) enable users to connect with people around the world. These services have been widely adopted and used by millions of users, and such services have emerged as a replacement of the transitional calling and messaging. Unfortunately, these communication services have also been used to commit illegal activities and serious crimes. Therefore, service providers ask for user's phone/mobile number to verify the user's identity and to prevent misuses.

The Internet is full of freebie services. Short Message Service (SMS) receiving services/websites are one of them. These message receiving websites provide users with real phone numbers and allow them to receive messages. In this paper, we investigate whether these message receiving website have been used as a tool to forge identity verification - typically done using One Time Passwords (OTP) - required for account creation. In our initial investigation, we created and successfully verified accounts for several messaging/calling apps as well as for social networking sites/apps using these message receiving services. Motivated from these findings, we collected and analyzed over 900K unique SMS messages received (upon request of other users) on 18 SMS receiving websites. Our analysis of these messages shows that 82.34% received messages included an OTP. This situation is very alarming that demonstrates the tendency of people to evade identity verification to create online accounts. We also found that the majority (52.47%) of verification code were six-characters long while nine-characters long verification codes were the least used.

CCS CONCEPTS

• **Security and privacy → Authentication**; *Social aspects of security and privacy*; *Pseudonymity, anonymity and untraceability*;

KEYWORDS

Fake accounts, Identity, KYC, SMS, Verification.

1 INTRODUCTION

Nowadays, the Internet has become an integral and important part of our daily lives. With continuous advancement in technology, the Internet has revolutionized almost every industry, e.g., advertisement, entertainment, and most importantly, communication. Today, communication service providers, such as *Whatsapp*, have replaced conventional way of making phone calls and messaging. These services are continuously evolving, offering a richer set of features to the users. On another side, the users are also quickly adopting such services at large scale. Internet-based calling is expected to increase to $44.78 billion with 116.5 million subscribers by the end of 2019 [9].

Unfortunately, these communications services have also been substantially exploited to commit criminal and other illegal activities [1–3, 7]. As the first step to counter misuse, these services enforce user's identity verification (generally, during sign up) via the user's phone/mobile number. The main reason to use phone numbers for identity verification is that phone numbers are traceable and have eventually become our identity. Verification codes - typically in the form of OTP - have been extensively used for identity verification step. OTPs are generally sent via SMS message or narrated over a voice call on the number specified (claimed to be in possession) by the user. Usually, messages are preferred over voice calls to communicate OTP as messages are less intrusive and do not require an interactive engagement of the user.

Motivation: There are several SMS receiving websites (hereinafter referred to as "SRW") on the Internet that provide real phone numbers to the users and allow them to receive SMS messages for free. As part of our initial investigation, we considered 25 mobile messaging/voice calling apps [4, 5, 8] as well as 15 social networking sites and apps [6]. We created a new account for each of the app/site that allowed message-based verification, and then tried to verify these new accounts using the services of SRW. Surprisingly, we could verify each account given that a fresh number (newly listed on SRW) was used for verification. It is noteworthy that even a single fake account may cause several unpleasant consequences while SRW add new numbers frequently.

Previous works: There are several works that discuss fake online account detection [14, 16] and commonly use machine learning techniques [19, 20], graph analysis [11], user activity based approaches [10], etc. But, none of the existing research works discuss how these fake accounts are successfully created when the identity verification via phone numbers has become a standard practice.

Our work: To the best of our knowledge, our research work is the first study that focuses on the most fundamental and essential element of today's online account creation process, i.e., the identity verification step. We seek to find whether SRW - that are usually free to use for a user - are being exploited to trick the verification step, and if so, then to what extent? To this end, our primary task is to analyze whether the SMS messages received via SRW contain verification codes or not. It might appear a simple task, but the worldwide accessibility of the Internet/websites and the linguistic, as well as the demographic diversities, pose difficulty to our task.

Contribution: The major contributions of our work are as follows:

(1) We present the first study on the exploitation of online SRW to forge ID verification.
(2) We propose an approach to determine whether the full text of a message contains a verification code. Our approach can work with messages composed in any language used in daily

* *Corresponding author.*

Md. Hajian Berenjestanaki is with School of Electrical and Computer Engineering, University of Tehran, Iran. Mauro Conti and Ankit Gangwal are with the Department of Mathematics, University of Padua, Italy.

life. Moreover, it can be tuned according to the desired levels of type-I (false positive) and type-II (false negative) errors.
(3) Finally, we also present our analysis of the linguistic distribution of SMS messages and the length of verification codes.

Organization: The rest of the paper is organized as follows: Section 2 presents a brief introduction to OTP. We elaborate our data collection technique, collected dataset, and our approach to identify messages containing OTP in Section 3. We present and discuss our results in Section 4. Finally, Section 5 concludes the paper.

2 ONE-TIME PASSWORD

In this section, we briefly explain the essential concept relevant to our work, i.e., OTP. It is also known as one-time passcode/pin. OTP is a time-bound and single-use code commonly used for authentication of users [17]. OTP verification is done on the server-side. Hence, the user must insert the OTP in the designated field to send it to the server for verification. OTP generation algorithms typically use randomness, or at least pseudo-randomness, to make the prediction of successor OTPs difficult. One can implement OTP generation in several ways, as long as the codes are unpredictable. Typically, an OTP is either generated on the client-side (e.g., using physical security tokens) or generated by the server and sent to the user over email, in-app (push-based) notifications, SMS, and/or voice calls. Therefore, we can classify the state-of-the-art OTP generation approaches [12, 13, 15, 18] broadly into two categories:

(1) *Client-side:* As mentioned above, such OTP generation techniques use security tokens and rely on time-synchronization between the authentication server and the user providing the password. The security tokens use the time as the seed, and the generated OTPs are valid only for a very short period of time. Naturally, this approach requires the user to physically possess the token or a third-party token emulator, e.g., Google Authenticator. Such techniques are not viable for various reasons including cost/time to distribute the tokens and trust on the third-party token emulators.
(2) *Server-side:* Such techniques typically use one-way hashing functions to generate OTPs and utilize user's password to generate (in a chained-fashion) a new OTP, user's response to a challenge (e.g., transaction details), or an arbitrary counter. The generated OTP is communicated to the user via a user-specified communication channel, e.g., via SMS. Since OTP generation on the server or OTP transmission may suffer delays, server-generated OTPs has a longer validity as compared to the token-based OTPs.

Assuming a simple case where English is the working language, each character of OTP can have 62 possibilities (26 lowercase letters + 26 uppercase letters + 10 decimal digits). In such situation, a four character long OTP can have 62^4 combinations. Web-based services are intended for international users, which forces their proprietors to use a common/popular set of characters that is available on most of the devices. Furthermore, the short lifetime of OTP and the limited number of permissible attempts during the verification process make digit-only OTP suitable for most scenarios.

3 OUR METHODOLOGY

The fundamental requirement of our study was to collect real messages from online SRW that allow a user to receive SMS using the phone numbers listed on their website. To begin with, we manually searched the web and found eighteen such SRW (listed in Table 1). It is worth mentioning that apart from these eighteen SRW others blocked us from fetching the messages programmatically, limited our scrapers to fetch only a very few numbers of messages, or required premium subscription. Hence, we discarded such SRW. Next, we organized our experiment in two phases: (i) scraping phase and (ii) classification phase.

3.1 Scraping phase

To scrape messages, we created two different Python scripts. The first script uses Application Programming Interface (API), if available, provided by SRW while the second script is designed to work in the absence of the APIs. The latter type uses regular expressions customized according to the structure of each individual website (mainly to the SMS rendering page) to fetch the desired information. Essentially, both the scripts collect the following information for a number listed on a website:

- Receiving Number (RN) - the phone number listed by SRW - with its Country Code (CC).
- Full Text (FT) of the message sent to a given RN.
- The contact information of the Message Sender (MS).
- Name of the SRW.
- Date and time of the message.

We store the information collected from each received SMS as a row in a MySQL database. We responsibly scraped these websites from December 1st, 2018 to February 28th, 2019 to collect messages. We kept only one unique copy of each message and discarded any duplicates fetched during the scraping phase. We collected a total of 905932 unique SMS-messages. These messages were sent by 73802 unique MS to 1010 unique RN of 36 unique countries[1]. Table 1 provides the details of our dataset. It is important to mention that the contact information of a MS can consist of only digits (e.g., 123456), only text (e.g., ABCDEF), a combination of digits and text (e.g., ABC123), and may also have some part morphed/hidden (e.g., 123XXX). Unlike the first three possibilities, the morphed MS cannot be uniquely identified. For such MS, we assumed two MS to be the same if they match on the length and on both morphed as well as visible part. Hence, 73802 MS in our dataset depicts the lower bound of the message senders.

3.2 Classification phase

There are no standards or guidelines for crafting/formatting an OTP message. Hence, creating an automatic classifier that can achieve perfect results is a very challenging task. We used some reasonable assumptions and created a quasi-automated process in Python to determine whether the FT of an SMS contains verification code. Our goal is to keep false positives as low as possible. But, we can

[1] Australia, Austria, Belgium, Brazil, Canada, Croatia, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hong Kong, Indonesia, Israel, Italy, Latvia, Lithuania, Malaysia, Mexico, Netherlands, Nigeria, Norway, Pakistan, Poland, Portugal, Puerto Rico, Romania, Russia, South Africa, Spain, Sweden, Switzerland, Ukraine, UK, USA. It is worth mentioning that we considered the UK (+44) as one country.

Table 1: Our dataset

SRW	Total n. of CC	Total n. of RN	Total n. of distinct MS	Total n. of collected SMS
7sim.net	29	94	2281	8616
freephonenum.com	2	25	2926	41517
getfreesmsnumber.com	8	59	8928	45318
miracletele.com/sms	9	9	1889	9197
receive-sms.com	1	28	6630	130423
receive-smss.com	13	48	3348	34960
sms-online.co/receive-free-sms	8	10	3191	15204
sms-receive.net	8	25	3141	34700
smsreceiving.com	7	39	3973	16904
smstools.online	18	468	15270	157885
www.freeonlinephone.org	5	10	1011	4887
www.mytrashmobile.com/numbers	6	6	11855	91232
www.receive-a-sms.com	15	17	3481	9151
www.receive-sms-online.info	14	32	8382	120316
www.receivesms.co	14	54	5068	30080
www.receivesms.org	7	21	2483	13623
www.receivesms.xyz	4	5	427	1507
www.smsreceivefree.com	2	60	6687	140412
Total (unique)	**36**	**1010**	**73802**	**905932**

afford false negatives because our results, in the presence of false negatives, would still represent the lower-bound of the ground-truth. We use the following assumptions about the characteristics of an OTP:

(1) *Length of an OTP:* Although less likely, an OTP can be as short as one character. Based on our observations, we assume the minimum length of an OTP to be four characters. A four-digit long OTP (1 correct combination in 10^4 combinations) is adequately robust and has been used commonly. OTPs with three or fewer characters will contribute to false negatives. On another side, we manually verified long (nine[2] or more characters) OTPs to prevent false positives.

(2) *An SMS with OTP includes special keywords:* Of course, a service provider may choose to send merely the verification code in a message. Albeit, as standard practice the user must be given proper instructions and information. Therefore, we composed a list of keywords, i.e., *otp, passcode, password, pin, verification* that we noticed in most of the messages with verification codes. Again, this list is kept short to keep false positives low.

(3) *OTPs are random words:* As discussed in Section 2, randomness is the key property of OTPs. Moreover, the standard OTP generation techniques [12, 13, 15, 18] focus on the randomness of each character in OTP rather than the randomness of whole OTP as a word. Hence, it is least likely that a generated OTP either will be a valid word in the dictionary of the message's original language or match with the message receive date (explained in the next paragraph). Nevertheless, a random word can be the name of a website and consequently can raise a false positive. To eliminate such a possibility, we check such potential words in the list of *Alexa*[3] Top 1-Million websites.

Using these assumptions, we built our system. Algorithm 1 explains the flow of our process. Since a message can be composed in any language, we begin with identifying the language of the message. To do so, we used *Polyglot*[4] - a natural language pipeline that supports massive multilingual applications. If the message was not written in English, we translate it to English. We search (case-insensitive) for the keywords in the message and terminate the process in case none of the keywords is found. Next, we tokenize FT using *WordPunctTokenizer*[5] except on hyphens because OTP might be hyphenated, e.g., six digits OTP from *Whatsapp* are hyphenated in the middle. Therefore, we simply delete any hyphen encountered during the tokenization process. We discard all the tokens shorter than four characters and keep tokens composed with letters-only/digits-only/combination of letters and digits. We also discard tokens that match with the message's (receive) date because messages, e.g., for credit card transactions, indeed contain dates. To do so, we use the format "*yyyymmdd*", "*yyyyddmm*", "*mmddyyyy*" and "*ddmmyyyy*", where *yyyy* is a two or four digit number while *mm* and *dd* are one or two digits numbers. We also checked manually when an OTP was found to be 2018 or 2019 (the year when we collected our data). Finally, we classify the message as an SMS containing verification code if at least one of the token is an invalid word in the dictionary of message's original language and the same token is not the name of a website in *Alexa's* top-site list.

[2]We empirically chose this criterion mainly to filter mobile/account numbers.

[3]https://www.alexa.com

[4]https://polyglot.readthedocs.io/en/latest/index.html

[5]https://tedboy.github.io/nlps/generated/nltk.tokenize.html

Algorithm 1 To check whether FT contains a verification code.

```
keywords := ["otp", "passcode", "password", "pin",
             "verification"]
FT_has_verification_code := 0
temp_FT := FT
ml := Polyglot_Language_Detector(FT)
if ml! = English then
    temp_FT := Translate_To_English(FT)
end if
if none of the keywords found in temp_FT then
    exit()
end if
words := Tokenize(FT)
for w in words do
    if (length(w) > 3) &&
     (w is not a valid word in ml dictionary) &&
     (w does not match with message receive date) &&
     (w is not listed in Alexa Top 1-Million websites)  then
        if length(w) > 8 then
            FT_has_verification_code := −1   ▷ Manual check
            exit()
        else
            FT_has_verification_code := 1
            exit()
        end if
    end if
end for
return FT_has_verification_code
```

4 RESULTS AND DISCUSSION

We analyzed our data on various dimensions. In our dataset of 905932 messages, we found a total of 154 distinct languages that were used to compose these messages. 62.59% messages were written in English, it shows the prevalence of English as the messaging-language. Additionally, 40.58% messages were off-language[6]. Refer Table 2 for website-wise results.

Now, we discuss our primary research question, which is, whether SRW are being used to trick the identity verification step, and if so, then to what extent? As shown in Table 2, 82.34% received messages included an OTP. It is important to note that these results are obtained when our system is lenient with false negatives. Nevertheless, it is tough to comment on whether the account creation was successful or not. But, considering the fact that even a single fake account may cause unpleasant consequences, this situation is very alarming that shows the tendency of people to evade identity verification to create online accounts. On another side, one of the possibility is that these verification codes were not intended at all for sign-ups instead were used for other purposes, e.g., for logging in to an existing account. As mentioned in Section 1, SRW add new numbers and remove old ones frequently. Therefore, it is less likely that these websites are being exploited for use-cases such as logging in.

Similar to the trend seen in the our raw dataset (905932 messages), the majority (63.51%) of messages that included a verification code were written in English. Figure 1 depicts the language of messages containing verification code.

[6]The language of the message does not match with the national language of the country (based on CC) of RN.

Table 2: Our findings

SRW	N. of distinct languages found	Dominating language (%)	Off-languages messages (%)	SMS containing OTP (%)
7sim.net	101	English (59.59)	57.67	78.16
freephonenum.com	100	English (87.39)	12.34	85.68
getfreesmsnumber.com	114	English (59.48)	55.84	82.42
miracletele.com/sms	73	English (73.35)	60.14	83.79
receive-sms.com	133	English (68.84)	31.16	92.25
receive-smss.com	82	English (82.45)	47.26	76.7
sms-online.co/receive-free-sms	85	English (55.86)	40.94	85.14
sms-receive.net	111	English (57.49)	69.01	81.21
smsreceiving.com	70	English (61.36)	48.56	83.22
smstools.online	134	Russian (47.63)	48.16	78.17
www.freeonlinephone.org	58	English (67.10)	45.30	90.24
www.mytrashmobile.com/numbers	133	English (75.62)	31.94	75.17
www.receive-a-sms.com	97	English (59.00)	47.21	79.88
www.receive-sms-online.info	139	English (60.55)	61.27	80.24
www.receivesms.co	100	English (69.35)	43.86	82.62
www.receivesms.org	75	English (72.92)	42.63	85.67
www.receivesms.xyz	43	English (57.47)	54.74	73.92
www.smsreceivefree.com	121	English (80.91)	18.47	84.29
Total (unique)	**154**	**English (62.59 %)**	**40.58 %**	**82.34 %**

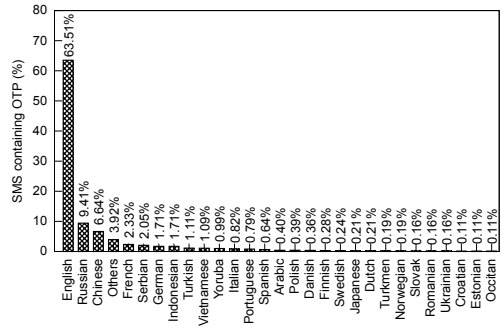

Figure 1: The language of messages containing verification code (*others* include all languages with contribution less than 0.1%)

Figure 2 shows the length of the verification codes identified by our system. The majority (52.47%) of messages with verification code had six-character verification code while the nine-character verification codes were the least used.

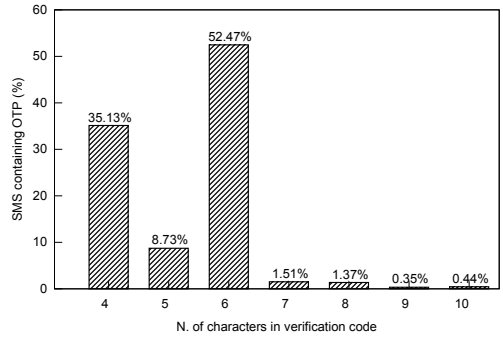

Figure 2: Length of the verification codes

Mitigation: A simple but effective mitigation to such forgery could be that an online service provider scrutinizes for the existence of the phone number - provided during the verification step - on various SRW. Implementing such checks is a trivial task, and it is one of our future goals. Given the nature of the problem and the seriousness of the consequences, such checks must be mandatorily used by online service providers, especially those offering communication and other critical services.

5 CONCLUSION AND FUTURE WORKS

OTP-based identity verification has become a standard practice to prevent any misuse of online services. Generally, this critical step is carried out using a unique OTP sent in an SMS message on the user's phone/mobile number. Online SRW allows users to receive messages. In our unique study, we found that message receiving websites have significantly been exploited, or are at least used as an option, to circumvent identity verification step necessary for sign up. This is a severe matter of concern that depicts the tendency of users to forge identity verification to create online accounts. We also proposed a simple but effective mitigation to this problem, which is of our future goals, i.e., we will create an open online portal that can check the existence of a phone number on SRW.

ACKNOWLEDGMENTS

Ankit Gangwal is pursuing his Ph.D. with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). This work was supported in part by EU LOCARD Project under Grant H2020-SU-SEC-2018-832735, and in part by Huawei Project "Secure Remote OTA Updates for In-Vehicle Software Systems" under Grant HIRPO 2018040400359-2018.

REFERENCES

[1] 2015. "Organised Gangs using Technology to Evade Police". https://tinyurl.com/yy7rn57b.
[2] 2017. "Forget About Backdoors, This Is The Data WhatsApp Actually Hands To Cops". https://tinyurl.com/y5f5flmu.
[3] 2018. "How WhatsApp has Helped Heroin become Mozambique's Second Biggest Export". https://tinyurl.com/yyt5gwff.
[4] 2018. "The 12 Best Team Chat Apps for Your Company". https://tinyurl.com/m6fa6rc.
[5] 2018. "The Best (and Most Secure) Chat Apps". https://tinyurl.com/yyxx3jbf.
[6] 2018. "Top 15 Most Popular Social Networking Sites and Apps". https://tinyurl.com/jd4ao6r.
[7] 2019. "Police Warn Against WhatsApp Scams". https://tinyurl.com/yyjphfoc.
[8] 2019. "The 10 Best Mobile Messaging Apps". https://tinyurl.com/ke8xnq4.
[9] 2019. "VoIP Services Market: Global Industry Analysis and Opportunity Assessment 2015-2025". https://tinyurl.com/yxgvx7ha.
[10] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, Konstantin Beznosov, and Hassan Halawa. 2016. Íntegro: Leveraging Victim Prediction For Robust Fake Account Detection in Large Scale OSNs. *Elsevier Computers & Security* 61 (2016), 142–168.
[11] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the Detection of Fake Accounts in Large Scale Social Online Services. In *9th USENIX Conference on Networked Systems Design and Implementation*. 1–14.
[12] Neil Haller. 1995. The S/KEY One-Time Password System. https://tools.ietf.org/html/rfc1760. (1995).
[13] Frank Hoornaert, David Naccache, Mihir Bellare, and Ohad Ranen. 2005. HOTP: An HMAC-Based One-Time Password Algorithm. https://tools.ietf.org/html/rfc4226. (2005).
[14] Boyeon Jang, Sihyun Jeong, and Chong-kwon Kim. 2019. Distance-based Customer Detection in Fake Follower Markets. *Elsevier Information Systems* 81 (2019), 104–116.
[15] Craig Metz, Neil Haller, and Mike Straw. 1998. A One-Time Password System. https://tools.ietf.org/html/rfc2289. (1998).
[16] Devakunchari Ramalingam and Valliyammai Chinnaiah. 2018. Fake Profile Detection Techniques in Large-scale Online Social Networks: A Comprehensive Review. *Elsevier Computers & Electrical Engineering* 65 (2018), 165–177.
[17] Aviel D Rubin. 1996. Independent One-Time Passwords. *Computing Systems* 9, 1 (1996), 15–27.
[18] Johan Rydell, Mingliang Pei, and Salah Machani. 2011. TOTP: Time-Based One-Time Password Algorithm. https://tools.ietf.org/html/rfc6238. (2011).
[19] Estée Van Der Walt and Jan Eloff. 2018. Using Machine Learning to Detect Fake Identities: Bots vs Humans. *IEEE Access* 6 (2018), 6540–6549.
[20] Cao Xiao, David Mandell Freeman, and Theodore Hwa. 2015. Detecting Clusters of Fake Accounts in Online Social Networks. In *8th ACM Workshop on Artificial Intelligence and Security*. 91–101.