Technische Universität Berlin

Forschungsberichte der Fakultät IV
Elektrotechnik und Informatik

SMS-based One-Time Passwords: Attacks and Defense

Collin Mulliner

Northeastern University

Ravishankar Borgaonkar
Patrick Stewin
Jean-Pierre Seifert

Technische Universität Berlin

Technical Report

Bericht-Nummer: 2014-02
ISSN: 1436-9915

September 2014

SMS-based One-Time Passwords: Attacks and Defense*

Collin Mulliner[1,2], Ravishankar Borgaonkar[2], Patrick Stewin[2], Jean-Pierre Seifert[2]

[1] Northeastern University crm@ccs.neu.edu
[2] Technische Universität Berlin
{ravii,patrickx,jpseifert}@sec.t-labs.tu-berlin.de

Abstract. *SMS-based One-Time Passwords* (SMS OTP) were introduced to counter phishing and other attacks against Internet services such as online banking. Today, SMS OTPs are commonly used for authentication and authorization for many different applications. Recently, SMS OTPs have come under heavy attack, especially by smartphone trojans. In this paper, we analyze the security architecture of SMS OTP systems and study attacks that pose a threat to Internet-based authentication and authorization services. We determined that the two foundations SMS OTP is built on, cellular networks and mobile handsets, were completely different at the time when SMS OTP was designed and introduced. Throughout this work, we show why SMS OTP systems cannot be considered secure anymore. Based on our findings, we propose mechanisms to secure SMS OTPs against common attacks and specifically against smartphone trojans.

Keywords: mobile phone, smartphone, banking, OTP, SMS, mTAN, malware, multi-factor

1 Introduction

Short Message Service (SMS) [1] based *One-Time Passwords* (OTP) were introduced to counter phishing and other attacks against authentication and authorization of Internet services. In these scenarios, SMS OTPs are mostly used as an additional factor in a multi-factor authentication system. Users are required to enter an OTP after logging in with a user name and password, or the OTP is required to authorize a transaction [12, 32, 40, 38, 36, 19, 20, 7]. The prime example of SMS OTP is the *mobile Transaction Authorization Number* (mobile TAN or mTAN) that is used to authorize transactions for online banking services.

The basic idea behind SMS OTP is that every account in a system is bound to a mobile phone and that the mobile phone is in the possession of the owner of that account. Thus, the owner of that account is the only person who is

* This is the extended version of the paper with the same title published at DIMVA 2013.

able to receive SMS messages sent to the phone number that is linked to the account. This turns the mobile phone into an access token. The addition of physically owning the specific device (the mobile phone) that is linked to a specific account of an online system makes SMS OTP a strong part of a multi-factor authentication/authorization system.

Unfortunately, today SMS OTP cannot be considered secure. Two reasons contribute to this fact. First, the security of SMS OTP relies on the privacy of SMS messages that in turn heavily relies on the security of cellular networks. Lately, several attacks against GSM and even 3G networks have shown that confidentiality for SMS messages cannot necessarily be provided. Furthermore, criminals have adjusted and created specialized mobile phone trojans [3, 26, 13, 23], since many service providers adapted SMS OTP to secure online transactions.

To the best of our knowledge, so far nobody has studied the weaknesses of SMS OTPs in-depth, nor offered any solution that protects against specialized trojans.

In this work, we seek to improve the security of SMS-based one-time passwords. We investigate attacks against SMS-based one-time passwords in general and analyze attacks that are currently used in the real world. Through this analysis, we show that the perception of SMS messages as secure is probably false. In today's world, one would expect that OTPs are transported using end-to-end security. Our work shows that this is not true anymore. Our argument is based on facts and observations in two areas, cellular network infrastructure and the design of mobile phone as well as smartphone hardware and software.

Based on the results of our analysis, we investigate security enhancements for SMS OTPs. We design two solutions, and implement and evaluate the most promising one. Our primary solution, a virtual dedicated OTP channel, only requires minimal modification of the mobile phone operating system to secure SMS-based OTPs against common attacks. Our solution is completely backwards compatible since it does not require modification of the SMS or OTP message. The solution is implemented entirely as software modifications to the mobile phone. We created a demo video of our OTP channel solution running on a real Android phone. We uploaded the demo video anonymously to YouTube, it can be found at: `http://www.youtube.com/watch?v=SF2HoK0D3_4`

The contributions of this paper are:

- **Attacks against SMS OTP:** Recently, more and more attacks against SMS-based one-time passwords have surfaced. We provide an overview of currently known attacks that are performed in the wild against SMS confidentiality and, thus, against SMS-based OTPs.
- **Analysis of the Attacks:** We analyze the various attacks and weaknesses of SMS OTPs. Through our analysis, we identify the root causes for the insecurity of SMS OTP today. The analysis provides the basis for the design of countermeasures.

– **Propose of Defensive Mechanisms:** Our solution, the virtual dedicated channel, protects against mobile phone trojans and requires only a minor modification of the mobile phone operating system. Our solution is completely backwards compatible to currently deployed SMS OTP systems.

The rest of this paper is organized in the following way. Section 2 provides an introduction of SMS-based one-time passwords. In Section 3, we present the threats and attacks against SMS OTPs. In Section 4, we analyze and discuss the attacks and identify the root problems of SMS OTP (in)security. In Section 5, we investigate solutions for improving the security of SMS OTPs. In Section 6, we present our solution, called the virtual dedicated channel, that prevents trojans from accessing SMS OTP messages on the mobile phone. Section 7, presents the implementation and evaluation of our virtual dedicated channel. Section 8 discusses related work and in Section 9 we briefly conclude.

2 One-Time Passwords via SMS

One-Time Passwords (OTP) are utilized as an additional factor in multi-factor authorization/authentication applications. They are only valid for exactly one authorization or authentication request. To avoid password lists, a convenient way to provide the user with an OTP is to send it via SMS. The phone number of the user must be registered for the service that provides SMS OTPs for authentication or authorization.

OTPs are quite popular as an additional authorization or authentication factor in web-based services. These passwords can be utilized to *authenticate* a user, i.e., the user needs a valid OTP to prove his identity to log into a web application or to access the company's private network [12, 32, 40, 38, 36]. SMS OTPs are also used for account verification, e.g., *Google App Engine* [19] and *Google Mail* [20]. Recently, the online storage service Dropbox added SMS-based two factor authentication[3] after facing some security issues. Online games such as Blizzard's Battle.net [7] have also started using SMS for account unlocking.

Another application for OTPs is *authorization*. Here, the OTP is bound to a certain request or transaction in order to confirm it. Additionally, the OTP can be restricted to a very short time window. In online banking web applications for example, the user has to authenticate himself via a valid username and password to initiate a transaction. Directly after this transaction request, the user gets an SMS message containing the OTP that must be additionally entered to authorize the transaction. In this application area the OTP is called a *mobile Transaction Authorization Number* (mobile TAN or mTAN). An example of an mTAN bound to a certain transaction is presented in Figure 1.

The basic principle for SMS-based OTPs is always the same, no matter what application is considered. The online service sends the OTP to the user's mobile phone via the cellular network, and the user enters the OTP to authenticate or authorize a transaction. Figure 2 summarizes this basic principle.

[3] https://www.dropbox.com/help/363/en

```
The mobileTAN for the Transfer of 1337 bitcoins
to the Account 123456789 is: 73KXCM
```

Fig. 1. SMS OTP example in form of a Mobile TAN used in an Online Banking Application. The depicted SMS links the OTP to a certain transaction request.

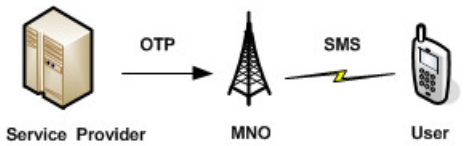


Fig. 2. SMS OTP Principle: The OTP is generated by the service provider and sent to the mobile network operator (MNO) that delivers the OTP via SMS to the user.

2.1 SMS Access Control

A user must usually pass a multi-factor authentication mechanism to access an SMS. The *Subscriber Identity Module* (SIM) card/mobile phone[4] is one factor, i.e., something the user has. It can be seen as a beneficial substitute of the classical two-factor authentication tokens (USB token, smartcard, password generator). The SIM card *Personal Identification Number* (PIN) or screen lock mechanism is the other factor, i.e., something the user knows. Only the person who knows the PIN and is in possession of the SIM card/mobile phone is able to access the SMS OTP.

So far, it has been assumed that the OTP for the actual application (e.g., web based online banking) is well protected when using this multi-factor authentication mechanism. In Section 3, we argue why the security of SMS OTPs cannot only rely on the SMS access control mechanism.

2.2 Parties Involved in SMS OTP

There are a number of parties involved in SMS OTP systems. The first one is the service provider that seeks authentication or authorization. The second party is the mobile network operator as the entity that delivers SMS messages to the mobile device. The end-user and his mobile phone or smartphone is the third party. These three parties are shown in Figure 2.

In some cases, a fourth party is involved. The fourth party is an SMS provider that connects the actual service provider to the mobile networks. SMS providers abstract the mobile network for the service providers by offering a simple interface to deliver a text message to a mobile phone.

[4] Note, certain CDMA phones sold by Verizon do not use SIM cards [25].

3 SMS OTP Threat Model

In this section, we present a threat model for SMS OTP. The underlying threat is breaking SMS confidentiality, i.e., eavesdropping, intercepting, and forwarding of SMS messages anywhere along the path between the sender and receiver.

If we consider the case of the mTAN, the basic attack works as follows. The criminal needs to be in possession of the victim's online banking login credentials. With these credentials the criminal logs into the banking website and initiates a transfer of funds to his own account. Before finalizing the transfer the bank sends an mTAN to the victim. Now the criminal needs to acquire the mTAN in order to complete the transfer.

The attacker's goal is the acquisition of the OTP, and for this he has several options that we present below. Note that as the attacks target SMS interception in general, they can be used against all SMS OTP systems.

3.1 Physical Access to Phone

With physical access to the phone that receives the SMS messages, the attacker can easily extract the OTP. Of course, gaining physical access is hard, time consuming, and easily detected. While there are ready-made toolkits to extract data from mobile phones of most manufactures, this kind of attack is unlikely for fraud since it cannot be performed on a large scale.

3.2 SIM Swap Attack

The SIM Swap Attack [21] is a *social engineering attack* with the goal of acquiring a replacement SIM card for the victim's mobile phone number. The replacement SIM is linked to the victim's mobile phone number. Hence, the attacker will receive all SMS messages that are supposed to be read by the victim. Therefore, the attacker will receive the OTP once he initiates an online banking transfer, for example. The SIM Swap Attack is mostly carried out in Africa [28].

3.3 Wireless Interception

As mentioned earlier, SMS OTP authentication systems completely rely on the security provided by the cellular network. In this subsection, we discuss the security of cellular networks and vulnerabilities that allow SMS messages to be intercepted over-the-air.

Cellular operators use the GSM, 3G, and CDMA technologies to provide mobile services such as SMS messages. However, GSM is insecure due to several vulnerabilities such as a lack of mutual authentication and weak encryption algorithms. In particular, there is no mutual authentication between mobile phones and base stations in GSM networks, hence fake base station attacks are possible. These are generally used to intercept mobile traffic (including SMS) of the end users. GSM uses different algorithms such as A5/1, A5/2, and A5/3 to encrypt

wireless communication between mobile phones and base stations (the A5/0 algorithm means there is no encryption). The algorithm A5/2 is weak and can be broken in a few seconds [5].

Recent advances in GSM research show that there is no end-to-end security. It is possible to capture GSM traffic using low cost devices and decrypt the traffic due to weak algorithms [31]. Nohl et al. show that the communication between mobile phones and base stations can be eavesdropped and decrypted using protocol weaknesses [30, 4, 6]. They were able to decrypt 64-bit A5/1 encryption. The attacking framework presented in that work can be used to intercept GSM traffic of a dedicated end user, including SMS messages [30].

Lately, is has been shown that femtocells – small 3G base stations [9] that are deployed in user homes – can be abused to intercept 3G communication. The attack works by installing a modified firmware on the femtocell that contains sniffing and interception capabilities. The capability to intercept SMS messages through a femtocell have been demonstrated [17]. The attack utilizes a femtocell to intercept the SMS message containing a password. Since 3G femtocells have already outnumbered the traditional 3G base stations [14] and their number is increasing rapidly, we believe that online criminals might misuse them to sniff OTPs. Furthermore, the report [29] suggests that such devices can be used to mount malicious attacks against mobile devices by online criminals.

3.4 Mobile Phone Trojans

Mobile phone malware, and especially trojans that are specifically designed to intercept SMS messages containing OTPs, are a rising threat. This kind of malware is created by criminals directly for the purpose of making money. In the following, we provide an overview of the different kinds of SMS OTP-stealing trojans.

The ZITMO (Zeus In The MObile) [3] trojan for Symbian OS is the first known piece of malware that was specifically created for intercepting mTANs. The ZITMO binary is delivered as a normal signed Symbian application. It possesses the required capabilities in order to register itself with the Symbian OS to receive SMS messages when they arrive from the mobile network. Upon reception it can forward SMS messages to a predefined mobile number. Besides the capability to forward SMS messages, ZITMO can also delete SMS messages. This capability can be used to completely hide the fact that an SMS message containing an mTAN ever arrived at the infected phone. Further, the ZITMO trojan can be remotely reconfigured via SMS. Through this the attacker can, for example, change the destination number for forwarded SMS messages.

In February 2011, a ZeuS version for Windows Mobile was detected and named Trojan-Spy.WinCE.Zbot.a [26]. The trojan contained the same basic functionality as ZITMO. Similar trojans also exist for Android [13] and RIM's Black Berry [15].

More recently, a new variant of Android malware was discovered. It targets mobile banking users in Germany, the Netherlands, Portugal, and Spain [23].

This malware is not part of the ZeuS family, but is also designed to capture banking OTPs sent via SMS.

There are other Android trojans that leverage access to SMS OTPs such as the MMarketPay.A [37] trojan. This trojan buys items from online stores and intercepts the SMS messages containing a verification code that is needed to complete the payment process.

Further studies [33, 41] show that authentication credential stealing mobile malware exist in the wild.

All known SMS OTP trojans are user-installed malware. This means they do not leverage any security vulnerability of the affected platform. Instead, they use social engineering to trick the user into installing the binary. Further, the trojans are executed as normal applications without special privileges.

4 Analysis of Weaknesses and Attacks

In this section, we analyze and discuss the security issues and attacks presented in Section 3. We identify and present the general reasons why certain weaknesses exist and why attacks are possible.

4.1 Cellular Network Insecurities

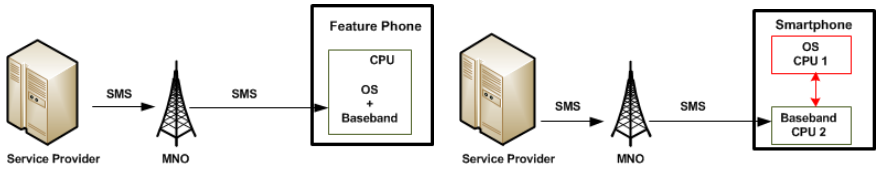
One major issue of SMS OTPs is that authentication service providers blindly rely on security provided by the mobile network operator. However as described in Section 3.3, numerous vulnerabilities in cellular network technologies suggest that it is possible to intercept cellular network traffic (in case of GSM). In addition, in some countries such as India, cellular network traffic is not encrypted by default. Furthermore, mobile network operators disable wireless encryption of SMS and call traffic. This can happen to decrease network load. Sometimes it occurs because of technical difficulties or because of a disaster such as an earthquake [22, 10]. In these cases, an attacker equipped with suitable tools can intercept traffic to capture authentication codes transmitted over-the-air. However, one could argue that such personalized attacks against the authentication systems are less likely to happen and difficult to achieve in practice. Our goal is to stress that such new attacks prove that the fundamental assumption of considering cellular networks as a secure element and transmitting authentication codes in plain text cannot provide end-to-end security.

Furthermore, while accessing OTP-based services during travelling abroad, SMS messages are delivered by the roaming cellular network operator. If this roaming operator is rogue, then SMS messages containing OTP can be intercepted. There was a case of a rogue cellular network operator in 2005 [35]. Thus, a roaming cellular network operator cannot be trusted when requiring end-to-end security.

4.2 Mobile Phone Design Issues

In Section 3.4 we presented a number of mobile phone trojans that are specifically designed to intercept SMS OTPs. This is possible due to the way mobile phone and smartphone operating systems are designed today.

Most mobile OSes provide an API to access received SMS messages from the SMS inbox. An OS can alternatively provide an API that allows an application to actively participate in the delivery process of SMS messages on the phone. If the latter is possible, a trojan can receive, alter, delete, and forward SMS messages without user interaction without leaving a trace of its malicious behavior.



(a) The restricted OS of feature phones protects SMS messages.

(b) SMS messages are usually less protected once they left the separated baseband environment.

Fig. 3. Revealing End-to-End Security Deficiencies of Modern Smartphones

By examining the hardware design of modern smartphones, we get a clearer picture of what has happened to the basic assumptions of the security of SMS messages. In the past mobile phones only consisted of one system, as shown in Figure 3(a), where one CPU executes both the mobile operating system and the baseband (the cellular interface). Smartphones today consist of two dedicated systems (two CPUs), as shown in Figure 3(b), one for the mobile operating system (e.g. Android) and one for the baseband.

To protect the security-critical baseband, feature phone OSes were very restricted compared to smartphone OSes. This restriction helps to protect SMS messages on feature phones. Due to the described separation, baseband security is not the concern of the smartphone OS. As a result, smartphone OSes became very open. This means manufacturers are able to provide, among other things, very sophisticated APIs to the cellular subsystems such as SMS messaging.

The main issue we identified is that SMS OTP was designed at a time where a mobile phone was a simple and dedicated system. This system was the endpoint for SMS messages. Legitimate applications could not access SMS messages on those phones, neither could trojans. On smartphones, end-to-end security, as present on feature phones, does not exist anymore.

Some smartphone OSes protect SMS messages through their permission system. Unfortunately, most users grant any permission to any application [34]. In Section 5.2, we present a protection mechanism to protect SMS messages while they are transported within the smartphone OS.

5 Defending SMS OTP

In this section, we present possible countermeasures that mitigate attacks against SMS OTP systems. We investigate approaches that require support of service providers, cellular network operators, and mobile OS manufacturers.

Our first idea provides a general solution that protects against most threats such as wireless interception and phone trojans. The idea is based on end-to-end encryption and requires support from the service providers or cellular network operators. Our second approach directly targets mobile phone trojans, as this is the current attack observed in the wild. It only requires support from the operating system manufacturer.

5.1 SMS End-to-End Encryption

Our first idea is to use end-to-end encryption to protect OTP messages when the SMS message gets intercepted or eavesdropped on. The idea relies on a concept called *application private storage* that is found on almost all mobile platforms today. This is a permanent storage area that is private to each application. Only the application that stored a piece of data is able to access it. This kind of private storage is available on most of the common smartphone platforms such as Apple iOS, Google Android, Symbian OS, Windows Phone 7, and Java 2 Platform, Micro Edition (J2ME). The Android Data Storage description [18] states *"You can save files directly on the device's internal storage. By default, files saved to the internal storage are private to your application and other applications cannot access them (nor can the user). When the user uninstalls your application, these files are removed."* Windows Phone 7 and iOS have a similar model [27, 2].

The concept is as follows. The OTP service generates the OTP message. For this it can keep its existing setup. In the second step the OTP message is encrypted with a customer-specific key. Each of the service's customers has a unique secret key. The encrypted OTP message is sent to the customer's mobile phone via SMS. This uses the existing OTP infrastructure operated by the service. On the user's phone, a dedicated application decrypts and displays the OTP message to the user.

While an SMS OTP trojan can still access the SMS message it cannot access the key that is required to decrypt the OTP message. The downside of this approach is the key distribution. Key distribution can be solved in many ways. We decided to not solve key distribution and rather investigate other solutions.

5.2 Virtual Dedicated Channel on the Handset

We identified mobile phone trojans as the major threat to SMS OTP since the trojan attack can be easily performed on a large scale. Hence, we present the following solution to protect against trojan attacks that requires minimal support from operating system manufacturers and minimal-to-no support from the service provider and cellular network operators. Our solution is therefore very easy to deploy.

Our main idea is to protect *certain* SMS messages against local interception by delivering them only to a specific application on the phone. Normally, any SMS capable application can read any SMS message that is received by the phone, as we discussed in Section 4.2.

We create a *virtual dedicated channel inside the mobile phone OS* by removing *certain* SMS messages from the general delivery process on the phone and redirecting them to a special OTP application. Messages sent via this dedicated channel are secure against local interception.

The endpoint of the virtual dedicated channel is an application with similar functionality to the default SMS application. It receives and stores SMS messages. The only difference is that it will only receive OTP messages, and that its message store cannot be read by other applications. The protection is ensured by the use of application private storage. From now on, we refer to this as the *OtpMessages* application.

Our dedicated channel is based on a minor modification of the mobile operating system. The modification is small since all mobile phones already implement specialized local routing of SMS messages to implement the various features present in the SMS standard, e.g., WAP push. In the following section, we will discuss the dedicated channel in detail.

6 Dedicated SMS OTP Channel

We investigated several methods for designing our dedicated channel on the mobile phone. Throughout our investigation, we considered multiple aspects such as implementation effort, having a clean design, backward compatibility, and ease of deployment.

In the following, we present two design approaches. The first approach is based on SMS ports that represents a low effort and a clean design approach. The second approach is based on a message filter that is easy to implement and offers backward compatibility and thus is easy to deploy.

6.1 SMS Port-based Channel

The SMS standard supports directing messages to specific application via the use of SMS ports implemented using the *User Data Header* (UDH) [1]. An SMS port works exactly like a TCP/IP port where an application can bind itself to a specific port to receive traffic that is sent to that port.

The idea is to pick a port that is going to be used for OTP messages. The *OtpMessages* application will listen on this port to receive all OTP messages. To make sure that trojans cannot bind to this port, operating system assistance is required. In particular, the OS only allows an application with a specific cryptographic signature to bind to this port. Almost all mobile operating systems support both required components: signed applications and SMS message routing based on ports.

There are two minor challenges for this approach. First, the mobile operating system would need to be modified to add support for the SMS port-application signature combination. Second, the services that send SMS OTP messages need to know if a specific phone supports the dedicated OTP channel, since messages sent to an unused port are simply discarded. Due to these issues, we decided to explore a different path that we present in the next section.

6.2 Message Filter-based Channel

The previous idea had the goal to have small implementation effort and a clean design. To fullfil the other two goals, backward compatibility and deployability, we refined our idea and designed a channel based on message filtering.

The idea is to have a phone only solution that neither involves the service provider nor the cellular operator. Furthermore, we want to keep the solution backwards compatible with phones that do not implement our protection mechanism. This is achieved through the fact that we do not require the SMS OTP messages to be changed.

Our method acts as a filter inside the mobile operating system's SMS receiving code. Therefore, this solution can be easily added into the existing infrastructure present in the mobile phone OS. Our filter inspects every incoming SMS message to decide if the message has to be forwarded to the dedicated channel receiver, the *OtpMessages* app, or if the message is routed through the OS's default SMS path. We developed two kinds of filters that can be used for our purpose:

1. Keyword-based filter that matches a keyword or a set of keywords against the message body or the start of a message.

2. Sender-based filter that matches against the originator address of an SMS message. This could also match against all *short codes*. Short codes refer to 4 to 6 digit phone numbers, short codes are mostly used to interact with paid services.

7 Implementation and Evaluation

In this section, we present our implementation and evaluation. We only implemented the message filter-based dedicated channel approach, since it is the most promising solution. We leveraged the Android platform for development and evaluation. We created a demo video that shows our implementation and evaluation. The video can be viewed at: `http://www.youtube.com/watch?v=SF2HoK0D3_4`

7.1 Implementation

The implementation extends the `dispatchPdus(..)` method in `SMSDispatcher.java` at `com/android/internal/telephony` of the Android 4.0 sources. We

added lines 3 to 6 as depicted in Figure 4 to implement the SMS routing. The filtering decision is implemented as a string match in `channel_filter(..)`. If a message matches, it is routed to the OTP application. On Android, this is accomplished by adding a *Component* to the *Intent*[5]. The Component directly addresses the receiving class by its fully qualified name, in this case `com.example.OtpMessages.Receive`. Therefore, it is only delivered to the specific application.

```
1 protected void dispatchPdus(byte[][] pdus) {
2  Intent intent = new Intent(Intents.SMS_RECEIVED_ACTION);
3  if (channel_filter(pdus)) {
4       ComponentName cn = new ComponentName("com.example.OtpMessages",
          "com.example.OtpMessages.Receive");
5       intent.setComponent(cn);
6  }
7  intent.putExtra("pdus", pdus);
8  intent.putExtra("format", getFormat());
9  dispatch(intent, RECEIVE_SMS_PERMISSION);
10 }
```

Fig. 4. SMS OTP Routing Implementation.

7.2 Evaluation

To evaluate our approach, we reconstructed the SMS sniffing trojan scenario. We implemented a simple SMS sniffing trojan by registering for `android.provider.Telephony.SMS_RECEIVED` events. This is the way SMS messages are received by any application, including malware [41]. Our trojan grabs SMS messages as soon as they arrive and pops up a message box to show "SMS intercepted" and the message text, thus providing immediate feedback when the message is intercepted.

In a second step, we implemented the *OtpMessages* application. The application registers to receive incoming SMS messages using the same method as our trojan. Every time *OtpMessages* receives an SMS message, it will display a pop-up containing the message and the string "OTP Message Received". This way, we can easily distinguish our two applications.

For the actual evaluation we crafted a number of SMS messages based on the message shown in Figure 1. We configured the keyword filter to match on the string "The mobileTAN" at the beginning of a message. Then, we simply sent the messages from another mobile phone to our test device. All messages that contained the string "The mobileTAN" were only received by the *OtpMessages* application. The appendix includes a screenshot of a received OTP message. To

[5] http://developer.android.com/reference/android/content/Intent.html

verify that our trojan still works, we sent a few messages to the phone that do not contain the filter string. Those messages were received by the trojan.

Possible Attacks against the Dedicated Channel

An attacker with root privileges on the mobile phone has access to all data stored on the device. This includes SMS messages received via the dedicated channel.

We further investigated the possibility of Denial-of-Service attacks against the dedicated channel and the *OtpMessages* application. An attacker can flood a device with useless messages that contain the keyword used for filtering in order to annoy the victim. This attack is possible already without our solution in place. In any case it will not reveal the OTP message.

8 Related Work

No prior work has examined SMS-based one-time passwords in detail. We are the first to investigate the root issues that threaten the security and privacy of SMS messages, and specifically the security of OTP messages sent via SMS.

Koot [24] provides a simple risk analysis of mTAN security for iOS as well as Android smartphones. The work fails to provide an in-depth study of the root causes of mTAN insecurity. They do not aim to secure mTAN, but rather try to link the mobile phone to the computer used for online banking.

Many tools exist to encrypt short messages sent between mobile phones [16, 39]. Such tools cannot be used for multi-factor authentication/authorization applications. The tools enable a mobile phone user to confidentially send an SMS to another mobile phone user, but the tools do not aim to protect SMS OTPs.

Several studies conducted on mobile malware [33, 41] show that authentication credential stealing mobile malware exists in the wild. Furhter research such as [11] shows that there are even more possibilities for criminals. In this work, we present countermeasures that specifically protect against mobile malware that is built to intercept and exfiltrate authentication credentials sent via SMS.

A large scale study [8] evaluated authentication schemes in general using three main characteristics: usability, deployability, and security. Their security characteristics basically attest SMS OTP with maximum points besides two issues. These issues are: not *Resilient-to-Internal-Observation* and not *Resilient-to-Theft*. Our virtual dedicated channel makes SMS OTPs *Resilient-to-Internal-Observation* and thus increases the security of SMS OTP significantly.

9 Conclusions

With increasing demand of stronger authentication mechanisms, online services adopted SMS-based one-time passwords to mitigate phishing and other attacks.

Services adopting SMS OTPs are not limited to banking and other financial services, but include email providers and popular online games.

Lately, SMS OTP have come under heavy attack, especially by mobile phone trojans that are specifically designed to intercept and forward OTP authentication and authorization credentials to criminals.

We presented the virtual dedicated channel, a solution that secures SMS-based OTPs against SMS stealing mobile phone trojans. Our solution is completely backwards compatible and only requires minimal changes on the mobile phone side. Thus, our solution is easy to deploy since it leaves the infrastructure at the service provider and the OTP message format unchanged.

SMS-based OTP is one of the most user friendly multi-factor authentication mechanisms today that does not require an additional device. We believe our solution provides the means to secure SMS OTPs against attacks and thus helps to prevent online account theft and fraud.

Acknowledgements. This work was partially-supported by DARPA grant no: KK1243 (DarkDroid).

References

1. 3rd Generation Partnership Project. 3GPP TS 23.040 - Technical realization of the Short Message Service (SMS). http://www.3gpp.org/ftp/Specs/html-info/23040.htm, September 2004.
2. Apple Inc. IOS Developer Library: Cryptographic Services. http://developer.apple.com/library/ios/#documentation/Security/Conceptual/Security_Overview/CryptographicServices/CryptographicServices.html#//apple_ref/doc/uid/TP30000976-CH3-SW6, July 2012.
3. A. Apvrille. Zeus In The Mobile (Zitmo): Online Banking's Two Factor Authentication Defeated. http://blog.fortinet.com/zeus-in-the-mobile-zitmo-online-bankings-two-factor-authentication-defeated/, September 2010.
4. E. Barkan and E. Biham. Conditional estimators: an effective attack on a5/1. In *Proceedings of the 12th international conference on Selected Areas in Cryptography*, SAC'05, pages 1–19, Berlin, Heidelberg, 2006. Springer-Verlag.
5. E. Barkan, E. Biham, and N. Keller. Instant ciphertext-only cryptanalysis of gsm encrypted communication. pages 600–616. Springer-Verlag, 2003.
6. A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of a5/1 on a pc. In *Proceedings of the 7th International Workshop on Fast Software Encryption*, FSE '00, pages 1–18, London, UK, UK, 2001. Springer-Verlag.
7. Blizzard Inc. Battle.net SMS Protect FAQ. https://us.battle.net/support/en/article/battlenet-sms-protect, September 2012.
8. J. Bonneau, C. Herley, P. C. von Oorschot, and F. Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2012.
9. H. Claussen, L. T. W. Ho, and L. G. Samuel. An overview of the femtocell concept. *Bell Labs Technical Journal*, 13(1):221–245, 2008.
10. G. Cryptophone. Questions about the Interception of GSM Calls. http://www.cryptophone.de/en/support/faq/questions-about-the-interception-of-gsm-calls/, 2012.

11. A. Dmitrienko, A. Sadeghi, C. Liebchen, and L. Davi. Over-the-Air Cross-platform Infection for Breaking mTAN-based Online Banking Authentication. https://media.blackhat.com/ad-12/Dmitrienko/bh-ad-12-over-the-air-dmitrienko-slides.pdf, December 2012.
12. Duo Security. Modern Two-Factor Authentication. http://duosecurity.com.
13. F-Secure. Threat Description: Trojan:Android/Crusewind.A. http://www.f-secure.com/v-descs/trojan_android_crusewind_a.shtml, 2011.
14. FemtoForum. 3G femtocells now outnumber conventional 3G basestations globally. http://femtoforum.org/fem2/pressreleases.php?id=277, June 2011.
15. D. Fisher. Zeus Comes to the BlackBerry. http://threatpost.com/en_us/blogs/zeus-comes-blackberry-080712, August 2012.
16. FSU Mobile Solutions. CryptoSms. Google Play: https://play.google.com/store/apps/details?id=edu.fsu.cs.cryptosms&hl=de, 2012.
17. N. Gold, K. Redon, and R. Borgaonkar. Weaponizing femtocells: The effect of rogue devices on mobile telecommunication. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS)*, Feb. 2012.
18. Google Inc. Data Storage — Android Developers. http://developer.android.com/guide/topics/data/data-storage.html#filesInternal.
19. Google Inc. SMS Verification for App Creation. https://developers.google.com/appengine/kb/sms.
20. Google Inc. Verifying your account via SMS or Voice Call. http://support.google.com/mail/bin/answer.py?hl=en&answer=114129.
21. icici Bank. What is SIM-Swap fraud? http://www.icicibank.com/online-safe-banking/simswap.html.
22. Infosecurity-Magazine. Indian company hacks GSM and usurps IMSI. http://www.infosecurity-magazine.com/view/24680/indian-company-hacks-gsm-and-usurps-imsi/, 2012.
23. A. Klein. The Song Remains the Same: Man in the Mobile Attacks Single out Android. http://www.trusteer.com/blog/song-remains-same-man-mobile-attacks-single-out-android, July 2012.
24. L. Koot. Security of mobile TAN an smartphones. Master's thesis, Radboud University Nijmegen, Feb. 2012.
25. L. Martin. Can Verizon Be Used on a SIM? eHow tech: http://www.ehow.com/info_12182861_can-verizon-used-sim.html, 2011.
26. D. Maslennikov. ZeuS in the Mobile is back. http://www.securelist.com/en/blog/11169/Zeus_in_the_Mobile_is_back, February 2011.
27. Microsoft Coperation. Windows Phone 7 Security Model. http://download.microsoft.com/download/9/3/5/93565816-AD4E-4448-B49B-457D07ABB991/WindowsPhone7SecurityModel_FINAL_122010.pdf, December 2010.
28. Mobile Banking Blog. Another SIM swap fraud. http://mbanking.blogspot.com/2007/12/another-sim-swap-fraud.html, December 2007.
29. I. Muttik. Securing Mobile Devices:Present and Future. http://www.mcafee.com/us/resources/reports/rp-securing-mobile-devices.pdf, December 2011.
30. K. Nohl and C. Pudget. GSM: SRSLY? http://events.ccc.de/congress/2009/Fahrplan/events/3654.en.html, 2009.
31. OsmocomSDR. inexpensive SDR (Software Defined Radio) project. http://sdr.osmocom.org/trac/wiki/rtl-sdr, 2011.
32. PhoneFactor, Inc. Comparing PhoneFactor to Other SMS Authentication Solutions. http://www.phonefactor.com/sms-authentication.

33. A. Porter Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A Survey of Mobile Malware in the Wild. In *Proceedings of the ACM Workshop on Security and Privacy in Mobile Devices (SPSM)*, 2011.

34. A. Porter Felt, K. Greenwood, and D. Wagner. The Effectiveness of Application Permissions. In *USENIX Conference on Web Application Development*, 2011.

35. V. Prevelakis and D. Spinellis. The athens affair. *IEEE Spectr.*, July 2007.

36. SMS PASSCODE A/S. Two-factor Authentication. `http://www.smspasscode.com/twofactorauthentication`.

37. TrustGo Mobile Inc. MMarketPay.A. `http://blog.trustgo.com/mmarketpay-a-new-android-malware-found-in-the-wild-2/`, 2012.

38. VISUALtron Software Corporation. 2-Factor Authentication - What is MobileKey? `http://www.visualtron.com/products_mobilekey.htm`.

39. WhisperSystems Inc. TextSecure. `http://www.whispersys.com/`, July 2012.

40. R. Yang. SMS Text Message Based Authentication. Citrix Developer Network: `http://community.citrix.com/display/xa/SMS+Text+Message+Based+Authentication`.

41. Y. Zhou and X. Jiang. Dissecting Android Malware: Characterization and Evolution. In *33rd IEEE Symposium on Security and Privacy*, May 2012.

Appendix

```
public boolean channel_filter(byte pdus[][])
{
  String keywords[] = {"The mobileTAN", "The mTan",  "OTP!", "mTAN"};
  SmsMessage s = SmsMessage.createFromPdu(pdus[0]);
  String body = s.getMessageBody();

  for (int i = 0; i < keywords.length; i++) {
    if (body.startsWith(keywords[i]))
      return true;
  }
  return false;
}
```

Fig. 5. Our `channel_filter(..)` Implementation used for our Evaluation.

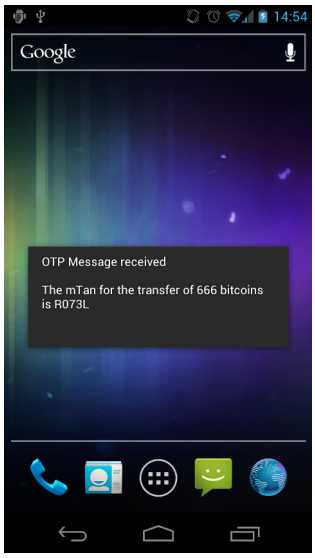


Fig. 6. The OtpMessages Application receives an SMS containing an mTAN OTP.