Student Seminar: Exploiting Two Factor Authentication of Android and IOS

Robin Solignac 235020

March 10, 2017

1 Introduction: The 2 Factors Authentication

Two factor authentication (2FA) is a combination of 2 access control in order to make the authentication more robust to attacker. The general model is: to authenticate yourself to an online service you both need to provide an information you know (your password) and use a physical object you have. One of the most most used "physical object" is a phone or a smartphone as large majority of people in today's world has one. More specifically, it's the possibility, through this phone, to send a message who will not transit by the (maybe compromised) PC who is used. Your smartphone is used as an out of band channel.

While it exist other 2FA schemes we will only study here the ones using smartphones as they are the most frequent in practice. Moreover the attack will focus on SMS based scheme. It make sense as it will discussed later that most of the other smartphones 2FA proposed by services an be passed by asking for SMS 2FA instead.

The precise scheme assumed here is the following: assuming we want to authentication to a service on a PC, after a successful password authentication, the user is required to enter (on the PC) a One-Time Passwords (OTP) send by SMS to its phone in order to fully authenticate to the service.

This paper will present 2 attacks to break this authentication assuming a compromise PC and a sane smartphone. The former is to be used with android, the later on Ios.

This paper is essentially based on "How Anywhere Computing Just Killed Your Phone-Based Two-Factor Authentication" by Konoth, Radhesh Krishnan, Victor van der Veen, and Herbert Bos. [?]

2 Key concepts

Before describing the attacks, this section will explain and summarize the Key concept they use. It will first describe what is synchronization and why it's a threat to the 2FA concept. Then the setup in which we assume the type of attack performed: a Man in the Browser attack

2.1 Synchronization

Our use of informatics is mostly divided between our personal computers and smartphones. For this reason software manufacturer have sometime decided to implement synchronization processes between the 2 devices in order to make transition and general use smoother, thus blurring the line between the two. This mean that the smartphone is less and less out of band with respect to the PC, which break the main assumption done by 2FA scheme using smartphone.

3 examples of such synchronization who will be used in our attack are the following:

Google play remote install it's possible from a PC via the Google play website to remotely install an application on an Android smartphone if both are logged on the same Google account. The only thing appearing on the phone afterward is the application icon in the app tray and an notification saying "<app_name> as been successfully installed".

Apple Continuity On recent version of Ios and macOs you can enable this setting to synchronize in clear, read and send your SMS from your mac.

Browser synchronization Almost all of today's most popular Internet browser (including Firefox, Chrome and safari) propose synchronization between the mobile and desktop versions logged under same user account. it cans synchronize history, bookmark and sometime currently open tabs.

2.2 MitB: Man in the browser attack

MitB is a type of attack who assume that the attacker has an entire control and view on the PC browser of the victim. Like a man in the middle attack, the attacker can see all data exchanged by the browser and server and can modify them (on the fly). it also can send and receive data in the name of the user. But unlike the former it has access to these data before they are encrypted (and after they are decrypted) And has also modify browser related setting like bookmark and current open tabs URLs. In short powerful man in the browser attack can remotely perform the same actions has someone getting physical access to the browser as well as modify content of request and responses messages.

there are different way to do a MitB attack, using malware infecting the whole system, by API hooking or via malicious plugin.

3 2FA Attack

This section describe attacks to defeat 2FA under the following model. We assume that the PC of the customer is compromised and can perform man in the browser attack but the smartphone is still sane. The attacker try to authenticate on a service from the PC, It has get the password from the PC infection but the service request a second factor authentication by sending a one time password (OTP) by SMS to the victim smartphone who run is either under Android or Ios. This choice of model is motivated by the fact that 2FA is specially design to prevent authentication to services because of only an infected PC.

3.1 Android

The principle to the attack is: via the MitB, Hijack an google session via password, cookie stealing or else. Then Install from the web play-store an application with authorization to read SMS (authorization confirmed from the browser too). Then when an SMS is received it is forward via Internet to the attacker, who can successfully authenticate.

In order to succeed the attacker need to pass two defense setup by Google.

Bypassing Google boncer Google remote install only allow to install application published on the Google play store. So the attacker need to publish an SMS stealing app on the store. In order to do this it must bypass Google Boncer, the automated malware analysis tool deployed by Google on its store. Whenever an app in upload to the store, it's analyzed by Boncer, who perform static analysis

(code inspection, ...) as well as dynamic analysis (sending request and action to the app and analyze the resulting behavior)

Recent work CITE show that this defense is easy to bypass in various way. CITE. [?] is using an another clever way by opening a poorly protected of web window outside of screen from which it's possible to remotely execute malicious (Javascript) code. As this code is invisible to Google boncer the app will not be detected as malicious.

Let's note that this possibility to load web content and let it execute some code on the app is an another intended feature made by Google called Web-View

Activate the app When installed, an Android app can't be triggered by external event (such as RECEIVE_SMS) until it has been explicitly open for the first time. So we need to trick the user to open the app from the phone.

The first way is to give a clickbait title to the app so that the user will be tempted to open it when he sees the installation notification.

The second is to trigger its opening from the mobile browser by clicking a link. And with browser sync is activated (by default on chrome) we can modify from MitB all open tabs, bookmarks, and history URL to be this particular link, making it eventually clicked on. It will then redirect the user to the normal URL after the opening of the app, making the move difficult to notice.

In both case just after this opening the can make itself disappear for homescreen and app tray, leaving little trace of it existence of the smartphone, only in the application of the parameters menu. While still permanently running.

3.2 Ios

Since 2015 Ios has forbid the application to read all notification or SMS without explicit authorization (or they will be rejected from entering the appstore). Since then, the previous attack does not work.

However if the infected browser is on a Mac and Continuity is activated on the Iphone. the browser can still has access to to SMS in clear on the mac as soon as both are on the same LAN, which is likely to eventually occurs as they belong to the same person.

So under these pretty likely to appears conditions, the 2FA authentication can be very easily bypassed on 2FA by an MitB attack.

4 Discutions

This section will discuss practical feasibility of these attack and potential solution to defeat them on both platform.

4.1 It's not a exploit, it's a feature

Before talking about feasibility we must emphases a particularity of these 2 attacks: After the initial infection of the PC, no exploit, bug or hack are used. Every tool we used as been purposely designed for the use we makes of them. Services provider (Google, Apple, Mozzila, ...) propose way to reduce the air gap beetween the PC and the smartphone while a the same time security engineer rely on it too implement secure 2FA.

And that's why defeat these attacks are complicated, there's nothing fix. Ether developers should modify the offered sync possibilities offer, most of the time it would mean reducing them. Or security engineers should modify the assumption they make, which mean find new ones.

4.2 Android

This attack is largely feasible because its nearly impossible to use a Android phone without

Google account and if you have one it's extremely likely that you will also use it on your PC in order to access to use on of the many Google service.

But the truth is that a the solution against to this attack already exist but can still be bypassed. Since Android Marshmallow, deployed in October 2015 the authorization model has change, all authorization are now granted individually at run time when needed. With this new model, authorization to read SMS can't be given from a remote browser. The only problem is that if the app is purposely compiled for older version of android the authorization scheme will be the old one while still running on newer versions. But its really likely that in the future it will be forbidden to apps to be at least compile for version older than at least marshmallow.

It's also likely that the pure engineering challenge that Google bouncer is will be more and more powerfull in the future.

4.3 Ios

The Ios version of the attack is a even bigger example of the problem describe previously, the attack only use one feature made by apple itself to defeat the whole 2FA concept, and the only solution to this is a modification of ether the apple continuity feature of apple or the 2FA scheme used.

On the feasibility side, this attack need more Prerequisites: user need to use a Mac, have Continuity enable (off by default) and the 2 devices must be on the same LAN. This do not reduce the feasibility of the attack as the condition are still likely to happened. But it reduce scalability of a such attack compare to the Android one.

A solution to this attack on Apple side with light impact of user experience would be to only sync SMS whose sender is in contact list of the user.

4.4 Other 2FA authentication on smartphone

SMS OTP are not the only existing scheme of 2FA on smartphone. Majority of services also propose 2FA dedicate app (such as Google authenticator or Azure authenticator). As on both Ios and Android application can't access each other data's (Application sandboxing technique) both attacks are defeated. But most of the service who implement 2FA this way also include SMS 2FA as a backup solution. If for some reason the user can't use or access the app (i.e lack of mobile network or buggy phone) it can still authenticate through SMS. So assuming the user has provide its phone number backup to the service. 2FA using dedicated app can still be defeat by these attack by asking, from the PC, to use the backup solution to authenticate

5 Conclusion

The very soul of 2FA using smartphone is the assumption that there's an air barrier between your smartphone and your channel and is has been true for many year. But except for 2FA this is perceive more as a default than a desirable feature, and so it's been reduced over the year. This need to be taken in account while implementing 2FA on smartphone. services must make sure that the channel they are using is really out of band with the user PC. What's sure is that today 2FA through SMS is not secure anymore on Android and Ios and should be avoided, even as a backup solution.