https://web.archive.org/web/20211023221110/https://blog.trendmicro.com/trendlabs-security-intelligence/ios-url-scheme-susceptible-to-hijacking/

iOS URL Scheme Susceptible to Hijacking - TrendLabs Security Intelligence Blog

By Trend Micro

Published: 2019-07-12 · Archived: 2026-04-05 14:36:15 UTC

by Lilang Wu, Yuchen Zhou, Moony Li

Apple manages application security and privacy concerns by using a sandbox mechanism for iOS that constrains the reachable resources for each application. This was created to contain damage if an app was compromised, and all apps distributed through the App Store adopt it. However, because of this access control, communication between apps becomes more difficult.

Fortunately, Apple provides several carefully designed methods to assist app communication, the most common of which is the URL Scheme. In essence, this is a feature that allows developers to launch apps on an iOS device through URLs. It is a method of conveying information from one app to another. For example, when a URL with facetime:// is opened, FaceTime places a call — this is the URL Scheme coming into play. It is a very convenient shortcut; but the URL Scheme is designed for communication, not security.

Below, we discuss how abuse of the URL Scheme can potentially result in the loss of privacy, bill fraud, exposure to pop-up ads, and more. We illustrate the misuse using several official iOS apps from the Chinese market, notably the messaging and mobile payment app WeChat and the retail app Suning. There are many other apps with the same features and capabilities as WeChat and Suning that are also vulnerable to the attacks outlined below. We notified and coordinated with vendors, particularly WeChat, and reported these attacks to the Zero Day Initiative.

How does it work?

iOS allows one single URL Scheme to be claimed by multiple apps. For instance, Sample:// can be used by two completely separate apps in their implementation of URL Schemes. This is how some malicious apps can take advantage of the URL Scheme and compromise users.

Apple addressed the issue in later iOS versions (iOS 11), where the first-come-first-served principle applies, and only the prior installed app using the URL Scheme will be launched. However, the vulnerability can still be exploited in different ways.

URL Schemes affect account privacy

The URL Schemes function as portals for apps to receive information from other apps. As mentioned above, since Apple allows different apps to declare the same URL Scheme, malicious apps can hijack sensitive data of certain apps. This vulnerability is particularly critical if the login process of app A is associated with app B.

For instance, the Suning app (a retail application) allows a victim to log in using their WeChat account. The normal authentication process (seen below) is that the Suning app generates a URL Scheme query and sends it to WeChat App. When the WeChat app receives the query from the Suning app, it requests a Login-Token from the WeChat server and sends it back to the Suning app for authentication.

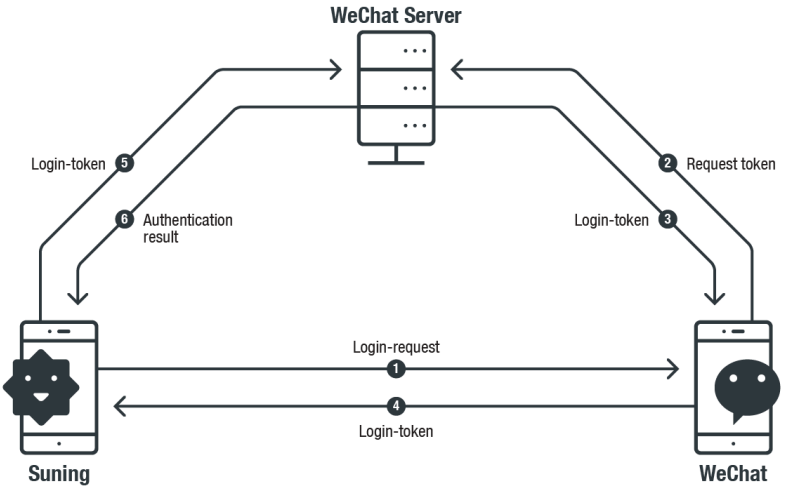

Figure 1. Suning app login with WeChat account associated with WeChat app

Our research found that Suning always uses the same Login-Request URL Scheme query to request the Login-Token, but WeChat does not authenticate the source of the login request. This allows an attacker to use an app's Login-Request URL Scheme for malicious purposes.

Attackers can use Suning's Login-Request URL Scheme query to request the Login-Token of a victim's WeChat account. He can then use that token to log into the Suning app with the victim's WeChat account (illustrated below). This process allows an attacker to collect personal information or abuse access to both accounts.

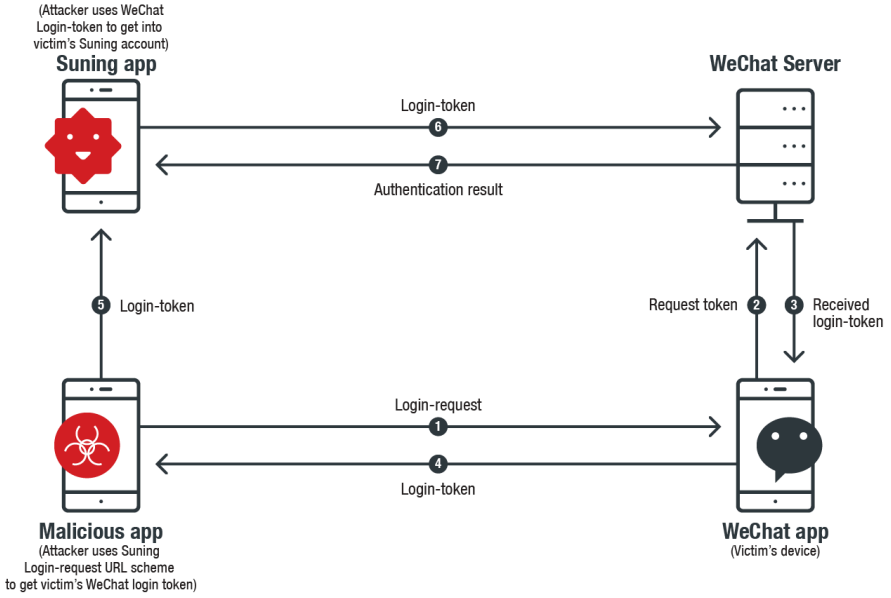

Figure 2. Malicious actor logging into the Suning app with victim's WeChat account

For the abovementioned attack to work, an attacker must first get Suning's Login-Request URL Scheme. Capturing it from the Suning app requires creating a whole separate app with WeChat's URL Scheme (WeChat's URL Scheme could be found in the field of *LSApplicationQueriesSchemes* in *Info.plist* of Suning). With the legitimate WeChat URL Scheme, a fake-WeChat can be crafted and Suning will query the fake one for Login-Token.

If the Suning app sends the query, then the fake app can capture its Login-Request URL Scheme . Our analysis found that this Login-Request includes the constant parameter with the constant value in multiple querying rounds, which gives attackers the chance to replay the request.

```
url=====weixin://app/wxe386966df7b712ca/auth/?scope=snsapi_userinfo&state=xxx
```

Figure 3. Captured Login-Request from Suning to WeChat

```
url=====wxe386966df7b712ca://oauth?code=02108kz80sbirH1HBcz80rmvz8008kzb&state=xxx
```

Figure 4. Captured Login-Token from WeChat to Suning

As illustrated in figures 3 and 4, Suning crafts its query by inserting a unique and complex URL Scheme (*wxe386966df7b712ca*) for WeChat to respond to. That particular URL is registered on WeChat as a Suning login. WeChat recognizes it but it will not authenticate the source of the Login-Request. Instead, it will directly respond with a Login-Token to the source of the request.

Unfortunately, the source could be a malicious app that is abusing the Suning URL scheme.

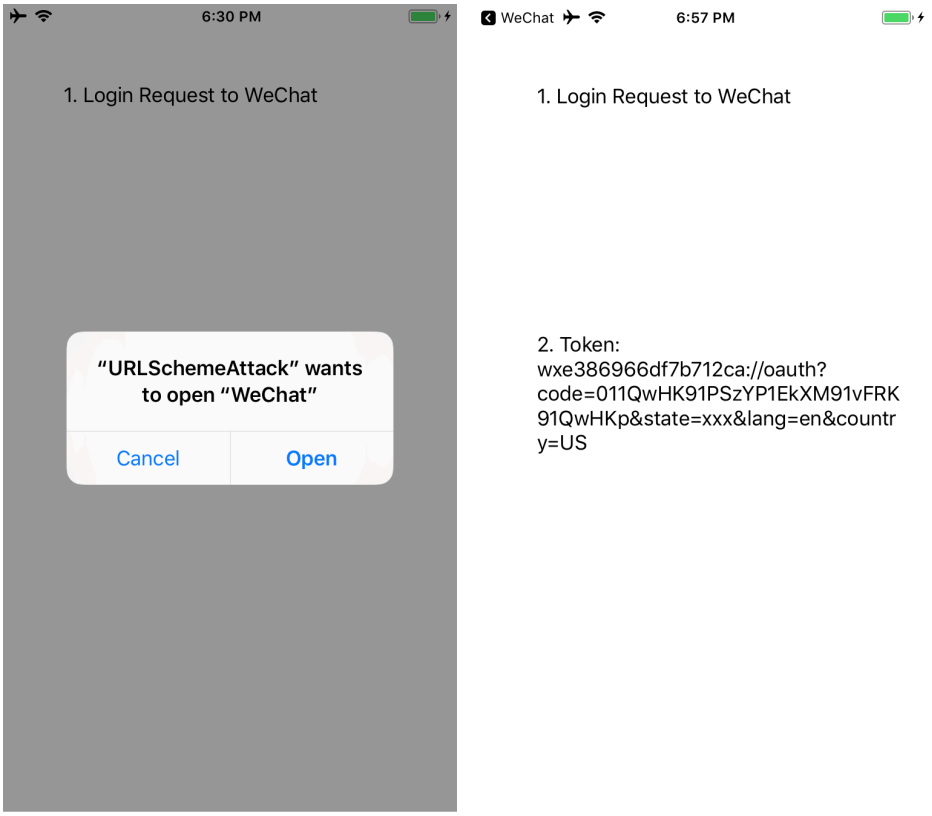

Figure 5. Stolen Login-Token by the Malicious app named URLSchemeAttack

This Login-Token can allow the malicious actor to access the victim's Suning account, exposing personal information. The compromised account can also be used for malicious purposes.

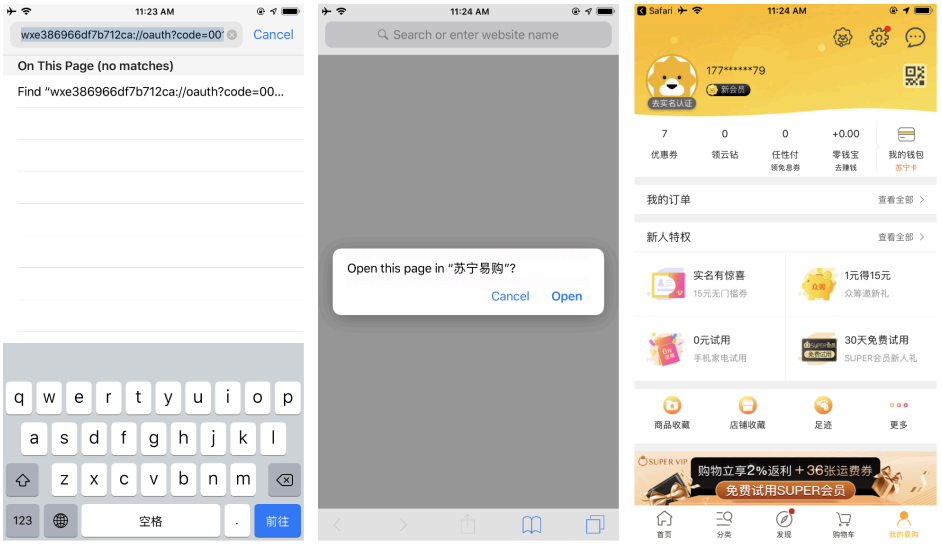

Figure 6. Attacker's device logs into victim's Suning account using stolen Login-Token

This issue is not exclusive to these two apps, the vulnerability exists in many apps that have this particular login feature.

URL Schemes abused for bill phishing

Counterfeit URL Schemes can be used in multiple attack scenarios. One other example is bill replay phishing, which tricks victims into paying others' bills. This attack uses both social engineering and the vulnerability of URL Schemes.

In general, bill replay phishing is done by sending the URL Scheme of a billing request to certain apps with a payment feature. This attack can be illustrated by a specific feature of DiDi or MeiTuanDaChe (both transportation apps) associated with any app with a payment function.

To reproduce this attack, we can look at issues with WeChat's capabilities as a payment-app. This has not been seen in the wild, but is possible with the given parameters. Attackers can use the strategy mentioned above: create a fake WeChat with the same URL Scheme of the legitimate one, and it will allow attackers to capture the URL Scheme billing request from DiDi or MeiTuanDaChe.

With the billing URL Scheme, an attacker can replay a billing request to any legitimate WeChat app and it will automatically call up its payment interface. Because the attacker is using the stolen (but legitimate) URL scheme request of DiDi or MeiTuanDaChe, the victim's legitimate WeChat app will accept the payment request.

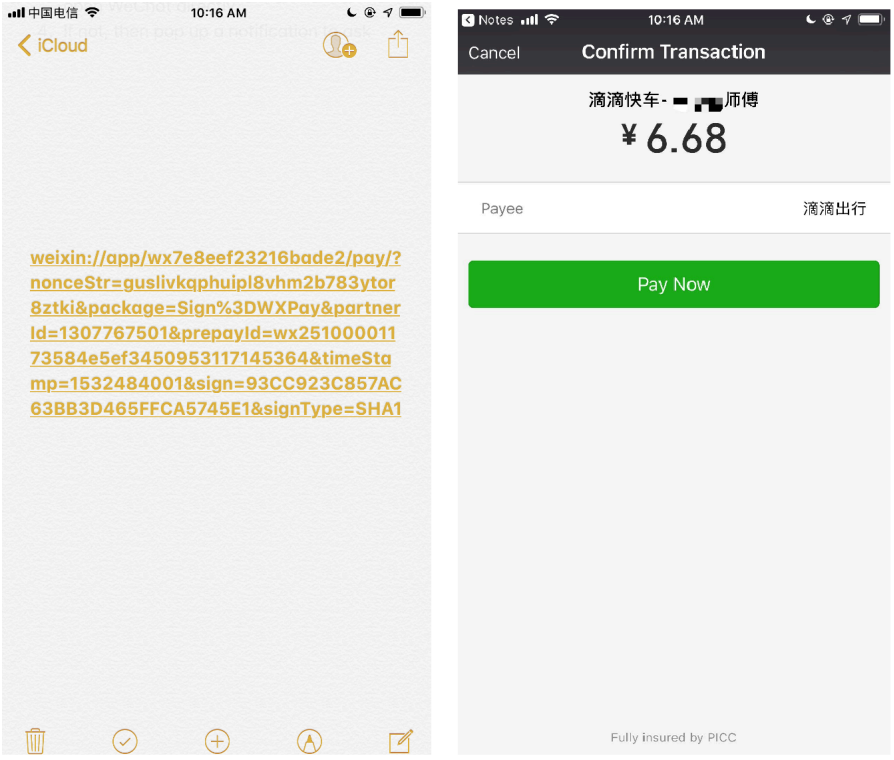

Figure 7. Proof that the URL Scheme request can be replayed

Admittedly, normal users might not be deceived, since they are unlikely to pay for a random billing request popping up in their WeChat payment interface. Nevertheless, this feature tremendously increases the probability of fraud. A user can absentmindedly click the pay button, or may think that this is a legitimate payment request.

Another attack scenario is possible as well, specifically using SMS social engineering and the URL scheme. Using the same apps as an example, the attack starts by taking advantage of the billing process.

Those who use DiDi or MeiTuanDaChe apps regularly get SMS messages to remind them of their unpaid bills. An attacker could generate the same SMS message to victims with a link containing the URL Scheme of a billing request — this billing request can be any request from the attacker. This link will be redirected to WeChat payment interface, and will ask victims for payment. This is an easy way to manipulate victims and trick them into paying bills that do not belong to them.

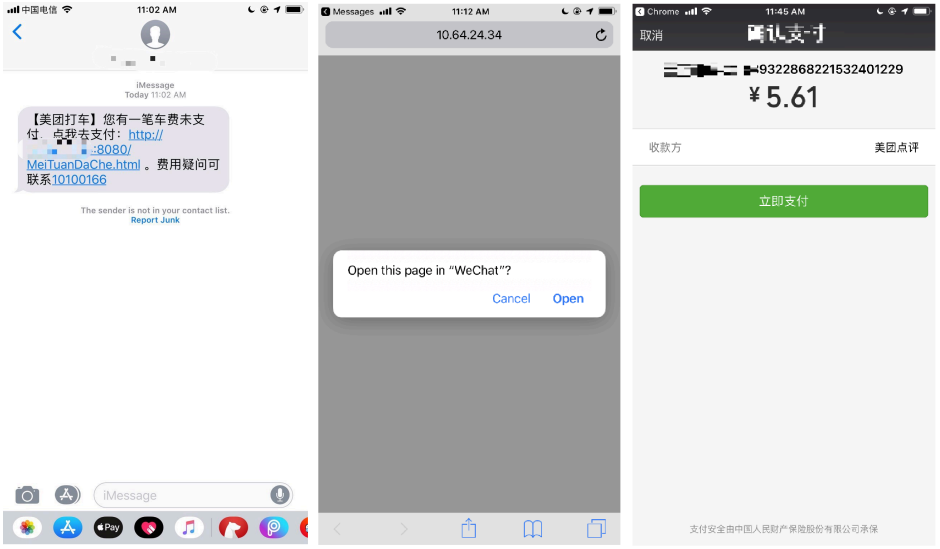

Figure 8. Bill Replay Phishing from SMS message

Fortunately, the latest version of WeChat has a new security strategy to prevent this billing replay attack: WeChat will not accept a billing request if it was conveyed by MobileSafari. However, billing replay is still feasible if the billing request is initiated by apps such as Chrome, Message, Gmail, and others.

URL Scheme used in pop-up ads

One more issue with the URL Scheme is that it can be used to launch applications. This means that once a certain URL Scheme is registered by a malicious app, there is a chance that the malicious app will be launched when the URL Scheme is called. In our research, plenty of apps that our system audited were found taking advantage of this feature to show ads to victims.

Potentially malicious apps would intentionally claim the URL Scheme associated with popular apps: *wechat://*, *line://*, *fb://*, *fb-messenger://*, etc. We identified some of these malicious apps:

SHA256	**Bundle ID**	**Number of declared URL Schemes per app**
1377742266f2a56f0424c1884c9c1fab8daa113b51c7f4d2ac4b2f88b770a1f5	co.Mu.Mu	490
c329fc984a2d75e4f1f15288cb5d9383c2f439dbae67bf808759bc5bc8336aee	co.medaistream.pokemap2	430
93326edbbd8863fbb0d2e602e1d8a7348349053ae144912286e78fb17a685c0f	co.musical.Musical	491
fe8fe4226ba17924cd31505a07b19691b5eecf2e9ae677bc9765130020662b89	com.musictechnology.Musia	491

Mitigation and solutions

The URL Scheme can be dangerous and is not recommended for the transfer of sensitive data. Attackers can take advantage of the non-authentication feature since communication and data is transferred regardless of the source or destination.

Affected vendors were notified July and August of 2018 when the vulnerability was first found, and we noted many of the vulnerable apps were actually popular downloads from the Chinese App Store. Apple is aware of this threat; they have already notified developers about the dangers of URL Schemes and offered recommendations. They specifically state: "URL schemes offer a potential attack vector into your app, so make sure to validate all URL parameters and discard any malformed URLs. In addition, limit the available actions to those that do not risk the user's data."

For developers, universal links are generally recommended as a best practice for deep linking. Setting up a universal link (HTTP or HTTPS) login interface, and musing a random identifier to authenticate the received login token locally, prevents hijacking and malicious login token replaying.

Furthermore, Trend Micro detects the potentially malicious apps that may take advantage of this feature. Our vulnerability scan system will be able to recognize the malicious apps that potentially misuse the URL Scheme. Trend Micro™ XGen™ security provides high-fidelity machine learning that can secure the gateway and endpoints, and protect physical, virtual, and cloud workloads. With technologies that employ web/URL filtering, behavioral analysis, and custom sandboxing, XGen security offers protection against evolving threats that bypass traditional controls and exploit known and unknown vulnerabilities. XGen security also powers Trend Micro's suite of security solutions: Hybrid Cloud Security, User Protection, and Network Defense.

Source: https://web.archive.org/web/20211023221110/https://blog.trendmicro.com/trendlabs-security-intelligence/ios-url-scheme-susceptible-to-hijacking/