How We Found These Vulnerabilities in Android Applications

Bin Ma(✉)

State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, People's Republic of China
mabin@iie.ac.cn

Abstract. With the rapid growth of application markets, many developers now spend their time and money to develop new smartphone applications, bringing ever more intelligent applications to smartphone users. However, the rapid development process of applications without full testing made them neglect the security of the applications. In this paper, I took UC Browser and Mobile QQ as examples to show some vulnerabilities.

Keywords: Vulnerabilities · UC browser · Mobile QQ · Smartphone security

1 Introduction

The technology of mobile phones has developed dramatically over the last decade. Many developers now spend their time and money to invent new smartphone applications, bringing ever more intelligent applications (apps for short) to smartphone users. People can finish most of the demands for daily through these diversiform apps such as reading, chatting, consuming, etc. Apps on mobile platform have brought great convenience to ordinary people and improved their standard of living [1].

To grab chances in these fast growing smartphone application markets, most of the developers usually promote their products online when finishing the basic function demand, while ignoring the potential security problem existed in their apps. These security issues are usually caused by some logic problems or check mechanisms, which can't be protected by the operating system. Once these vulnerabilities are exploited by hackers, the app may be under great threat, the accounts or information of users may be in danger.

In this paper, we seek to show the risk of these vulnerabilities in some common apps on Android platform. Section 2 describes a high risk information leakage problem existed in UC Browser, which looks like a Google Hacking [2] problem but appears on mobile platform. Section 3 shows a high risk logic flaw existed in Mobile QQ, which can be exploited to spread malicious information in the QQ group. In addition, we revealed the process of finding these vulnerabilities and demonstrated the seriousness of them. After reporting them to the corresponding company, the level of vulnerability is evaluated as high risk, and we also received a big reward for the report.

J. Tian et al. (Eds.): SecureComm 2014, Part II, LNICST 153, pp. 399–406, 2015.
DOI: 10.1007/978-3-319-23802-9_32

2 Information Leakage in UC Browser

2.1 Background

UC Browser [3] is a leading mobile internet browser with more than 500 million users across more than 150 countries and regions, which derives from benefit of its large user base in China (34.83 %) and their rapidly growing Indian market. The majority of smartphone users access the web through UC Browser, which means that once the vulnerability occurred, the influence is enormous.

The vulnerability existed in the search engine of UC called Shenma (sm.cn), which is developed by UCWeb Inc and Alibaba Inc in July 2013 [4]. Shortly after the release, we found the vulnerability and reported it to UCWeb Inc.

Mobile search is an evolving branch of information retrieval services that is centered on the convergence of mobile platforms and mobile phones. With the rapid development of mobile internet, a search engine for mobile platform is urgent for the great demand of smartphone users. Web search engine ability in a mobile form allows users to find mobile contents on websites which are available to mobile devices on mobile networks [5].

Web search engines use Web crawling or spider software to update their web content or indexes of others sites' web content [6]. Web crawlers can copy all the pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them much more quickly.

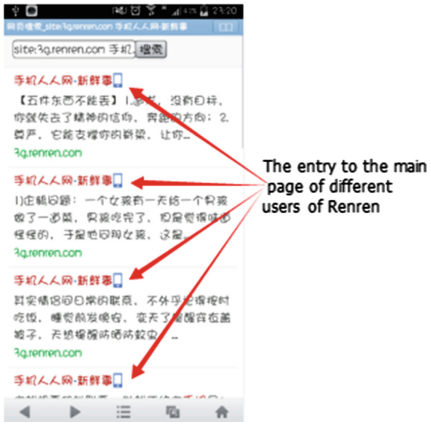
However, the crawler on mobile platform is very sensitive. Because the input method through mobile is difficult for its limited screen and small keyboard, programmers adopt some easy methods to remember the users' login data, such as a unique string called SID (Security Identifiers) which is invisible to users and attached in the link of the login page. When users open the links, the remote server will make a check automatically to finish a login process. But once these links are grabbed by the mobile search engine especially for a new mobile search engine without any filtration mechanism, users' accounts will be in great danger.

As we know, Google hacking is a computer hacking technique that uses Google Search and other Google applications to find security holes in the configuration and computer code that websites use [7]. While that can happen on mobile platform with mobile search engine. The following shows the risk of a new search engine.

2.2 A Description of Finding the Vulnerability

UC Browser takes Shenma as its default search engine. The crawler of Shenma grabbed sensitive links (for example, private links to maintain the login-in state) and saved them and the corresponding contents in its database without any information filtering. For this, we only need to search some related keywords of the login page, the search engine will fetch the corresponding contents from its database, which means that anyone can get these sensitive links just by some simple search to enter a certain user's main page. In addition, the browser supports advanced search like "site:[website][keywords]", which reduces the difficulty of search and increases the risk of user's personal account.

Here we take Renren as an example, a Chinese social networking site which has been called the Facebook of China [8]. We entered a search string like "site:3g.renren.com [keywords]", ("3g.renren.com" is a website designed for the smartphone users of Renren, and the keywords are some symbolic words when users finish a login and back to the main page, using these keywords, we found many other information leakages problems in UC Browser) the search results were so amazing (Fig. 1 shows the results). We got plenty of entries to different Renren users' main pages. When clicking these results, we can enter the main pages of different users without any login process or password, we also got the permissions like a normal user, and we can update or delete the user data, send a message, upload or download a picture etc. In a word, we can operate a Renren account at will and without any password just like the actual user.



(a) The search results when seraching the keywords in UC Browser

(b) Entering the main page of different users when clicking the results

Fig. 1. Screenshots of the example of Renren.

After further studying, we found that the crawler of UC Browser grabbed the links with a SID (Security Identifier) which is a unique string to mark the user who has finished a login process. For example, when a Renren user finished a login process, a new SID will be generated (e.g., the SID is "CEKfLCa0n8obv_QmTVh7am"), which means that the user can enter his or her main page without inputting the password and only by accessing "3g.renren.com/home.do?&sid=CEKfLCa0n8obv_QmTVh7am" next time. Unfortunately, the links with SID was grabbed by Shenma and stored in its database. When searching some keywords, these entry web addresses with SID will be shown in the browser. For other mobile websites, they take a similar approach to keep users in the login-in state. According to this, hackers can control someone's account only by searching the corresponding SID.

2.3 Threats of This Kind of Vulnerability

This kind of information leakage may cause great damage. When we first find the vulnerability of Renren, we thought it may be the problem of Renren. After further researching, we located the problem of UC Browser, and a series of information leakage occurred except for Renren. Users who have this kind of information leakage are generally because they used UC Browser to finish a login process before, and their entry web addresses were captured by its search engine.

We found that most of the famous Chinese mobile websites like Tecent, Sina Weibo all have this problem. For example, we can search the keywords like "site:ish.z.qq.com [keywords in the main page of Qzone]" to find a great information leakage in Qzone. In a similar format, we can find the problem of Sina Weibo by search "Site:weibo.cn [keywords]". All of the search results can be exploited to control the corresponding accounts. Figure 2 shows the results.

We also researched the SID of several large mobile websites of China. (Figure 3). Based on the table, we can see that the SID is widely used in most mobile websites, once a smartphone user finished a login process through UC Browser, his SID may be grabbed

(a) Qzone (b) Sina Weibo

Fig. 2. Information leakage of Qzone and Sina Weibo.

	Tencent Weibo	QQ Zone	Netease Weibo	Renren Mobile	Sina Weibo
Keep user logging in with SID	√	√	×	√	√
Log in with old SID	×	√	×	√	√
Dead time of SID	Session Time	Session Time	×	30 days	30 days

Fig. 3. The SID of several large mobile websites of China.

by the crawler, and his entry web address of main page may be searched by others easily, his personal information may be exposed in public. That's really dangerous.

2.4 Bug Reporting and Fixing

After finding the vulnerability, we reported it to the UCWeb Inc and shared the details and our analyses with them. They realized the seriousness of the problem and deleted the sensitive links in a short time. And they gave us a rank of 20 about the vulnerability (the highest rank) and sent us a reward [9].

3 Logic Flaw in Mobile QQ

3.1 Background

For most online apps, there are some web interfaces for apps to communicate with the remote servers to update data. These interfaces are usually invisible to the app for the decompilation protection mechanism, and the remote server are generally designed for the specific app, which means we can't access the interfaces through a browser or other methods. Because of this, finding the vulnerabilities in apps becomes difficult.

Some people are trying to analyze the source code or decompilation code of Android apk file to mine vulnerabilities [10]. In theory, this mining techniques can find more bugs and have higher efficiency. But the method may not work sometime for the decompilation protection mechanism in apps. In addition, a decompilation of apk file may take a lot of effort to find a logic flaw.

In this section, we use an agent tool called Burp Suite [11] to capture the packets between app and remote server to get the interface. In this way, we can get the request or post data sent to the server, by changing the post parameters and replaying the packet, we can find some vulnerabilities easily, especially for some logic flaws. According to this, we can also achieve a common tool for automatic detection, which may have higher efficiency and lower cost.

3.2 A Description of Finding the Vulnerability

As mentioned above, we decide to capture the packets between app and remote server to find vulnerabilities. We use a tool called Burp Suite to act as an internet agent between the local app and remote server, which means that all the http packets between them will be captured by the agent. In addition, Burp Suite can modify the parameters of the packets and drop certain packets or replay certain packets. So we can find some vulnerabilities by this way.

There is a new functionality called QQ Notice in the mobile client of QQ, which means that users can add a reminder and edit the contents of reminder in a QQ group at a future moment. When the time is up, group users on mobile platform will receive a notice message, and group users of PC (personal computer for short) will receive a notice at the bottom right corner of the screen. Figure 4 shows the notice in mobile client and PC client.

(a) Mobile client (b) PC Client

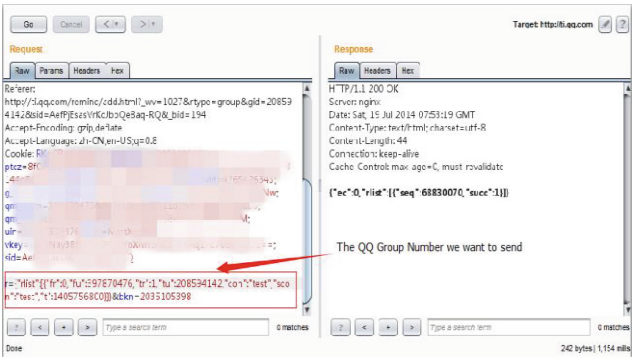
Fig. 4. The notice in Mobile client and PC client.

To check if there are any bugs, we captured the post data when creating a reminder. Figure 5a shows the packet and parameters. We can see that there is a parameter called "tu", which is short for "to user" and represents the QQ Group number we want to send a notice. We tried to modify the parameter to any other Group number, and replay the post packet, finally we received a 200 OK response.

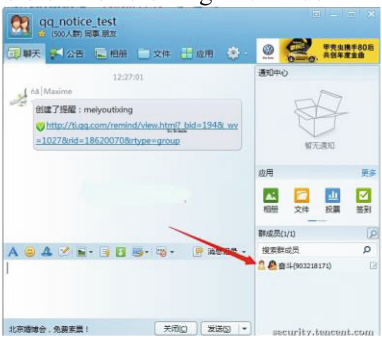
To confirm the findings, we conducted a simple experiment. We registered two users named A and B, user A created a QQ group which only have one member namely A itself, user B replay a packet whose group number was modified to the group that user A created just like the way we mentioned above. Finally we received the notice sent from user B in user A's group, but B is not the member of the group. Figure 5b shows our results.

3.3 Threats of the Vulnerability

The vulnerability can be described as sending a notice to any QQ Group without becoming a member of the group. Because the content of the notice can be edit casually,



(a) The packet and parameters when creating a reminder



(b) There is only one member in the group, namely user A, but A received the notice sent from B who are not in the group.

Fig. 5. Screenshots of our experiment.

hackers can exploit the bug to send any message to any QQ Group, such as advertisements, malicious links, illegal information etc. Thus a vulnerability of notice become a huge bug of spreading all kinds of spam information.

For instance, we can use a crawler to get all of the existed QQ Group numbers and names, and simulate a mobile login process on the computer, then we can send the notice packet to specific QQ Group. In this way, hackers can carry out a precision message sending.

3.4 Bug Reporting and Fixing

We reported the vulnerability to TSRC (Tencent Security Response Center) [12] and shared the exploit methods with them. After a short while, they realized the seriousness and froze the notice functionality. Three days later, they fixed the vulnerability and set the risk level as 7 (The highest is 10) for the vulnerability [13]. To express their gratitude, they gave us a reward which is worth more than 800 Yuan.

4 Conclusion

In this paper, we seek to show two types of vulnerabilities in Android applications, both of which are high-risk. We first described a high risk information leakage problem existed in UC Browser, which looks like a Google Hacking problem but appears on mobile platform. Then we showed a high risk logic flaw existed in Mobile QQ, which can be exploited to spread malicious information in the QQ group. In addition, we revealed the process of findings these vulnerabilities and demonstrated the seriousness of them.

In future, we will devote to finding an automatic method to detect these types of vulnerabilities, and we will study continuously for finding out the other vulnerabilities of generic apps on smartphone platform.

References

1. Young, J.R.: Top smartphone apps to improve teaching, research, and your life. Educ. Dig. Essent. Read. Condens. Quick Rev. **76**, 12–15 (2011)
2. Google hacking. http://en.wikipedia.org/wiki/Google_hacking
3. UC Browser. http://en.wikipedia.org/wiki/UC_Browser
4. Search engine of Shenma. http://baike.baidu.com/view/13036750.htm
5. Lagerspetz, E., Tarkoma, S.: Mobile search and the cloud: the benefits of offloading. In: 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 117–122. IEEE (2011)
6. Olston, C., Najork, M.: Web crawling. Found. Trends Inf. Retrieval **4**, 175–246 (2010)
7. Billig, J., Danilchenko, Y., Frank, C.E.: Evaluation of google hacking. In: Proceedings of the 5th Annual Conference on Information Security Curriculum Development. ACM (2008)
8. Renren Inc. http://en.wikipedia.org/wiki/Renren
9. Report of vulnerability in UC Browser (2013). http://wooyun.org/bugs/wooyun-2014-060257
10. Zhang, W, Cao, C., Liu, W., et al.: Vulnerability mining techniques in android platform (2013)
11. Burp suite. http://en.wikipedia.org/wiki/Burp_suite
12. Tencent security response center. http://security.tencent.com/
13. Report of Mobile QQ (2014). http://security.tencent.com/index.php/report/detail/11857