Security Analysis of Vancouver Pay-By-Phone Parking System

Chris Lee, James Wang, Benjamin Wai, Leo Wong

The security of the Vancouver Pay-by-Phone Parking System of Verrus is based on the authentication of something that the user knows. To enforce this authentication, users are required to input their phone number and a 4 to 6-digit pin. If the input information is correct, the user is allowed to access their personal profile. Users are permitted to input their PINs an unlimited amount of times. Base on this condition, we analyze that it is possible to write an automated program to simulate a user login attempt. The program will attempt to retrieve the PIN of a Verrus account by attempting all possible combinations of four to six digits. We concluded that the authentication system of the Verrus is vulnerable to an exhaustive brute force attack.

***Index Terms*—Pay-By-Phone Parking System, Verrus**

I. INTRODUCTION

This project investigates the security vulnerabilities of the City of Vancouver Pay-by-Phone Parking System. Our objectives in this project are to find the potential security flaws in the system, to show how the flaws can be exploited by users with malicious intends, and to suggest improvement to eliminate those system flaws.

Due to its many conveniences and advantages over the traditional metered parking payment method, Vancouver Pay-by-Phone Parking System has more than 130,000 users in Vancouver and the lower Mainland with 3,500 new users being added to the system each month [4]. Although Verrus Company advertises that the pay-by phone system is secured with authentication policies [5], we were able to locate a number of vulnerabilities in this system; most of these vulnerabilities will result in financial losses and personal information losses to the user. To address and provide suggestions to these flaws, we decided to conduct an analysis of the Vancouver Pay-By-Phone Parking System.

In order to achieve our objectives, several steps were taken. First, a thorough research on the system was done to find out how the system works and what possible attack methods were available. Second, automate our attack strategy with a program. Third, analyze the results and suggest solutions to improve the current authentication policies.

This report is divided into the following primary sections: system overview, vulnerabilities of the system, attacking the system, recommended solutions, and conclusion.

II. SYSTEM OVERVIEW

A. SYSTEM INTRODUCTION

Vancouver Pay-by-Phone Parking System is powered by a mobile payment company called “VERRUS”. This system was proposed to Vancouver in order to provide more conveniences to metered parking customers. There are several benefits of using this system compared to the traditional metered parking payment method [3]. Since the customers make payment using their cell phones, they are no long required to carry spare change, and they can extend their parking sessions without running back to their cars to top up the time left on the parking meters. Also, customers will have the option of receiving a text message reminder 10 minutes prior to the expiry of their parking session so that they do not have to cumbersomely remember the expiry time.

B. HOW THE SYSTEM WORKS

In order to use this system, a customer needs to sign up an account in the Verrus website [6]. When creating an account, a customer needs to provide some personal information such as their mobile phone number, credit card number, and the license plate(s) of his or her car(s). Once the account is created, the customer can log in to the server by using the mobile phone number as the account name. The customer can make a payment by calling 604-909-PARK. If the registered mobile phone is used for the transaction, the system will ask for the PIN, parking meter location number, and the number of minutes. If any other phone is used instead, then the system will ask for the mobile phone number that is registered for creating the account, the PIN, the parking meter location number, and the number of minutes. Fig. 1 below illustrates how to use the system.

Fig. 1. How to Use the Vancouver Pay-by-Phone Parking System

III. VULNERABILITIES OF THE SYSTEM

Most websites that provided private accounts or blogs have password-based or similar protection schemes to ensure the authenticity of users. The Pay-by-Phone system is no exception. As previously mentioned, Verrus asks for the private information of the user which it later uses for authentication purposes [6]. Although the process of creating authentication information and the process of that user during login are relatively convenient and simple, they produce enough vulnerabilities that eventually backfire on the user. With a target audience of drivers, designers of Verrus had to implement an access system that is user-friendly and intuitive for all users across a wide age group. Unfortunately, the designers’ vision of having a user-friendly system resulted in the sacrifice of the security of the system; as a result, the sacrifices make it convenient for attackers to penetrate the

defences of the pay-by-phone system. The following outlines the vulnerabilities of this system.

A. SHORT PINS

The login password is unreasonably short for an account that deals with finances. At sign up, the system only requests a password that is 4 to 6 digits long. Such short passwords are extremely vulnerable to a brute force attack. During the demo in the mini presentation, we had proven that an attack using an automated program to simulate a user login attempt was more than possible. Without the knowledge of any programming, an amateur attacker can also effortlessly attack an account; a simple repetitive manual guessing process could also retrieve a password within four to seven hours.

B. MULTIPLE PINS

Allowing multiple PINs for a single account is a security loop hole of the Verrus system. The intention of having multiple PINs is that when a user forgets the PIN that he or she had provided when creating an account, he or she can still login with the last four digits of their credit card number; in other words, every account has two PINs; the PIN that the user made, and the last four digit of the credit card. This multiple PIN system is built-in to the system; users do not have the option to deny this multiple PIN setting. Furthermore, the PIN that the user provides can be changed whenever the user wishes, however, the last 4-digits of the credit card number remains unless another credit card is used for the account.

C. MULTIPLE LOGIN ATTEMPTS

The Verrus login page allows multiple login attempts without any penalties. Base on this security vulnerability, the attacker can repeatedly attempt to login to a user's account. An attacker can write an automated program to guess the PINs of an account without any delay. This condition allows us to demonstrate in the mini conference that PINs can be obtained without any interference from the Verrus servers [8].

D. EASY CONFIGURATION

In the account profile [6], a user can edit the vehicle(s) that they wish to use for the pay-by-phone parking system. Users have the option of adding the licence plate numbers to the account through the website, or adding them through their mobile phone. If an attacker is able to break in to the account of a user, he can add up to 100 licence plate numbers to the account anywhere, anytime. The victim of the account will have absolutely no awareness to the changes to their account. Lastly, the licence plate does not have to be a local licence; the attacker can input a licence plate number that is in the United States or in Australia.

E. DENIAL OF SERVICE

Other than being able to add additional licence plate numbers to the user account, an attacker is also able to erase the original licence plate of the user to cause a denial of service. Although this course of action is highly aggressive and will certainly cause inconveniences to the victim, we analyze that this will be an unlikely course of action for an attacker to take because such action will draw the attention of the victim, who in turn will alert Verrus. An attacker will likely not attempt to disrupt the account from proper functionality to avoid being caught by authorities.

F. ACCESS WITHOUT REGISTERED MOBILE

For the convenience of its users, Verrus designed the system to allow unregistered mobile phones to access and configure a user account when the registered mobile is unavailable. This freedom in design focuses generally on the scenarios when the registered mobile is out of battery or is unreachable. Unfortunately, it also provides a gateway for people with malicious intents to access real user accounts without using a registered mobile. As mention before, without a registered mobile, an attacker is just required to input a registered mobile and PIN to gain access to the account.

G. PHONE NUMBER ASSUMED PRIVATE

The pay-by-phone system has two assumptions on the information used for authentication: only the user knows the PINs, and only the user knows the mobile phone number used for the pay-by-phone parking system. During an interview with the Verrus Q/A department of Vancouver [7], it is revealed that Verrus assumes consumer is expected to keep their mobile phone number (aka. account number) private. Such assumption, as we analyze, is absolutely infeasible because mobile numbers nowadays are easy to gain access to. For example, asking to borrow a mobile phone, business cards with mobile phone number(s), internet searches, or even interception of mobile waves when a user is accessing the pay-by-phone system are ways in which an attacker can retrieve this number. Even without actual knowledge of which mobile numbers are using the pay-by-phone system, we can always speculate which groups of people are using it. The Vancouver city council and its staff, for example, have been using the system since it was introduced to Vancouver. The Verrus staffs, as revealed during the interview [7], are also a pay-by-phone user.

IV. ATTACKING THE SYSTEM

A. OVERVIEW

One of the weaknesses of the system is the web-based access where users can login to change their account information. Due to the relatively weak security in this area of the system, we chose to focus our exploit attempts here. The major weakness, as aforementioned, is that a user can attempt an unlimited amount of logins to the server without the server intervening. The server itself does not implement any sort of prevention mechanism, such as image verification, or port traffic monitoring to prevent these multiple successive login attempts.

Theoretically, an attacker can physically sit in front of a computer with a known account number (a cell phone number) and try all possible combinations of a 4-digit PIN, starting at 0000, successively, until 9999. This way, an attacker is guaranteed to come across at least one of the possible PINs (the last 4-digits of the account owner's credit card). To put this into practice, it was important to

demonstrate that this process can be automated and, more importantly, a PIN can be recovered in a reasonable amount of time.

B. TOOLS USED

To help analyze what protocol the website used for authentication, a packet-sniffing program was used. Wireshark, WebScarab, or TamperData (Firefox add-on) were all suitable for our purpose, and WebScarab was used. By navigating to the website and sending a login request, the packet was bounced through the WebScarab proxy, revealing its contents to us, before hitting the actual server over at Verrus.

C. ANALYSIS

Upon analysis, we saw that the login process utilizes a simple HTTP POST protocol for authentication and only retrieves the fields the user entered: an account number and PIN. The login form itself will also generate two more pieces of data required for the login process to complete: a __VIEWSTATE field and a __EVENTVALIDATION field, which are hidden from the user. Lastly, to show that the login button on the html form had been clicked, a btnLogin field was necessary too.

An attacker with little HTML experience may have some trouble figuring out all the pieces of information required to get authenticated. Unfortunately, a simple packet sniffing program like the ones used will reveal to the attacker exactly what fields are necessary to generate a proper POST message for this server and what values these fields need to be. An attacker can also look into the HTML source code to find out these values. During our hacking attempts, it can be noted that the hidden values appear relatively static and seem to only change on a daily basis at the very least. What this means to an attacker is the ability to retrieve multiple PINs from a known list of accounts relatively trouble free. With all the necessary pieces of information at our disposal, the next task was designing an automated program to simulate a user login attempt.

D. IMPLEMENTATION

At the design stages, several options were available to us to automate the login. The first was scripting the process using AppleScript (for Macintosh) or iMacro (Firefox add-on). A second similar option was accessing the Windows API directly to simulate keyboard input. Lastly, socket programming to bypass the GUI interface altogether was the last alternative.

Socket programming through Java ended up being used. The program makes use of the Java HttpsURLConnection class to establish a connection to the Verrus web server. It then generates a String query formatted according to the POST method in the HTTP protocol and sends it to the server. This query will become the equivalent of a user logging in through the web portal. The PIN is stored as a String variable starting at value “0000” and is incremented by one after each iteration. The values for the hidden fields are retrieved manually from the login form on the website and hardcoded into the code, although simply parsing the webpage to automatically retrieve these values was also possible. In response to the POST, the Verrus web server will return a HTTP 200 OK and an HTML page. If a correct account and PIN combination was not discovered, the program will wait “x” amount of seconds and iterate this entire process again. Even though the Verrus website did not seem to monitor network traffic, it was important not to arouse suspicion by repeatedly hammering the server with login attempts, which is why a delay was implemented.

E. RESULTS

We realized that regardless of whether the account number and PIN combination used was correct or not, an HTML page would be returned. Upon analysis, it was discovered that the HTML pages returned from an incorrect login attempt versus a valid one differed only by the line “Invalid Phone number or password” or a lack thereof respectively. By this principle, the program was designed to parse the response from the server in search of this difference and if the aforementioned line was not found, it can be concluded that an actual account number and PIN combination was successfully discovered.

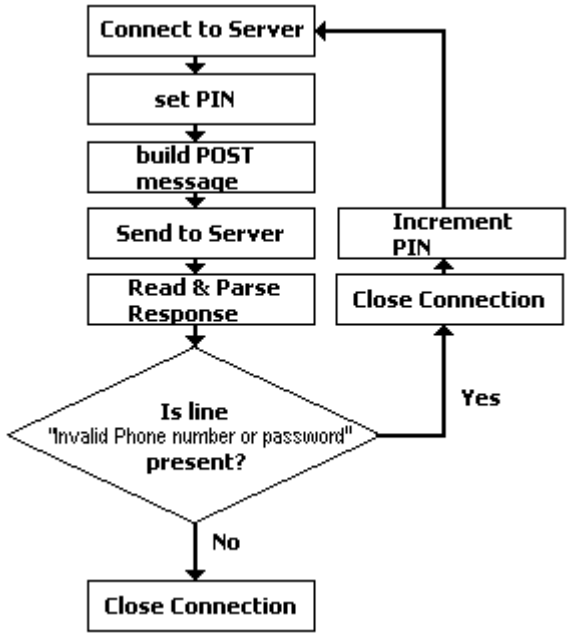

Fig. 2. – Software Flow Chart

With a key space of only $10^4 = 10000$, which is approximately 2^{13}, an attacker only needs to make 2^{12} attempts (according to the birthday theorem) before recovering a valid pin. This key space drops another power if the account has another unique 4-digit pin in addition to the last 4-digits of the registered credit card. With two possible 4-digit pins, an attacker can retrieve either one of these within 7 minutes. If optimization was applied to the program, such as trying possible PIN combinations in random blocks (i.e. 1000-1400, 0750-0920..) and leaving the unlikely PINs last (i.e. 0000, 9999), a PIN can potentially be recovered in an even smaller amount of time.

Lastly, the program can also be modified to randomly generate an account number as well. The area code is already known – either 604 or 778 – which leaves seven digits to be randomly generated. Although this process will take significantly longer, if left running long enough, a valid account and PIN combination will be recovered eventually.

V. RECOMMENDED SOLUTIONS

As the results of our team's hacking attempts illustrate, the Verrus Pay-By-Phone parking system is indeed vulnerable from attacks. The weakness of the PIN used for logging into the system is a serious security risk that may be exploited by hackers easily. Since the system gives the user the option to log in by using the simple four-digit PIN combination, it gives hackers the opportunity to try out all 10000 possible combinations from 0000 to 9999 and brute-search for the correct PIN. The tool described earlier that our team uses allows us to break into the system within a few minutes. In addition, creating the hacking tool itself does not require a lot of work. The time and resources needed to compose similar tools are trivial for an experienced programmer. Thus, taking advantage of the pay-by-phone parking system's weak password strength is a practical and easy way to hack into it.

To fix the issue of weak password and prevent attacker from getting user information in the database, there are many solutions for the Pay-By-Phone parking system. In terms of the ten principles of designing secure systems, there are two areas which the system can work on to improve its security level.

A. DEFENCE IN DEPTH

The first is the "Defence in depth" principle [1]. Since the main security issue of the system is its weak password, the most straightforward way to increase the security level is to strengthen the password. The website can achieve this by forcing the user to create passwords of more digits. To make the passwords even stronger, the website can make the requirement that the PINs must be combinations of digits and letters or even of digits, letters and symbols that are available in a mobile phone. Although such security measure is less convenient than before, it strongly discourages any malicious intent. Longer passwords or passwords with different compositions add significant difficulties to the method of brute-force searching, by making the passwords at least eight digits in length, the difficulty to brute-force search the correct password is increased by a power of two; by further specifying that passwords must contain both digits and letters, the difficulty is increased by a factor of $2.8*10^8$.

In cases where the level of security is still insufficient even after making the changes in password length, the website can add another defence layer in its identity verifying system. One method of such is to add another field in the login page which requires the user to enter his/her email address or other personal information. By doing so, the brute-force search method is useless unless the attacker happens to know the email address or the required information. Furthermore, instead of asking the user to enter personal information, the website can even implement the "CAPTCHA" challenge-response test system (Fig.3) [2] which makes the brute-force search impractical to use.

Fig. 3. The "CAPTCHA" challenge-response test.

B. FAIL-SAFE DEFAULTS

The second design principle that can be improved is the "Fail-Safe Defaults" principle [1]. The Verrus website can prohibit the attempts to brute-force search for the correct password by forcing the user to enter the correct password for a limited number of trials. If the number of incorrectly entered password is beyond the limit, the website should identify the user's activity as malicious and stop him from continuing his attempts for a period of time. Although this may cause problems of usability and create some inconvenience for the users, it can effectively disable the brute-force search method.

VI. CONCLUSION

In conclusion, there are many methods the Pay-By-Phone parking system can implement to improve the security of its website. The lacks of the most fundamental elements in designing secure system and the absence of the most basic anti-brute-force technique in the system are the main reasons the website can be broken into without too much effort. As the analysts of the security of the Vancouver pay-by-phone system, we suggested that the system implement additional security policies before expanding its services.

ACKNOWLEDGMENT

We would like to thank Professor Konstantin (Kosta) Beznosov for providing this great analysis topic for us, and his courageous generosity for allowing us to use his mobile number for testing during the demo [8].

REFERENCES

[1] Beznosov, Konstantin K. "Principles of Designing." EECE 412. UBC, Vancouver. 2 Oct. 2008.
[2] Digital image. Sign up for Windows Live. 30 Nov. 2008 <https://signup.live.com/signup>.
[3] "How It Works." Verrus For Consumers. 15 Dec. 2008 <https://verrus.com/verrus/how.aspx>.
[4] Macdonald, RG (Bob). Review of Pay by Phone Parking Program. Rep.No. 5423. General Manager of Engineering Services, City of Vancouver. 2008.
[5] "Privacy Policy of Verrus (UK) Limited and Verrus Mobile Technologies Inc." Verrus for Consumers. 1 Dec. 2008 <https://verrus.com/verrus/privacypolicy.aspx>.
[6] "Verrus Sign Up." Verrus for Consumers. <https://verrus.com/verrus/signup.aspx>.
[7] Wai, Benjamin. "Quetions of Verrus Security System." Telephone interview. 1 Dec. 2008.
[8] Wang, James, Benjamin Wai, Leo Wong, and Chris Lee. "Security Analysis of Vancouver Pay-By-Phone Parking System." UBC, Mini-conference, 18 Nov. 2008.