Scientific Journal Impact Factor (SJIF) : 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

A REVIEW ON ANDROID AUTHENTICATION SYSTEM VULNERABILITIES

Bh . Padma[1] and GVS Raj Kumar[2]
[1]*Dept of Computer Applications,GayatriVidyaParishadCollege for Degree and PG Courses,Yendada, Rushikonda, Visakhapatnam-45.*
[2]*Department of Information Technology,GITAM University,Yendada,Rushikonda, Visakhapatnam-45.*

Abstract—Mobile security has become a crucial aspect of mobile computing. People are maintaining their confidential and valuable information on smart phones. Most of the users and businesses use smart phones as message tools, and means of scheduling and establishing their work and private life. Smart phones contain increasing amount of exposed information to which access must be prohibited. But security is not easy, and security with mobile devices, smart phones is no exemption. Android is an operating system based on the Linux kernel, developed by Android Inc. which Google acquired in 2005. Initially, Android was launched to support touch screen devices like smart phones. These devices support different types of screen locks, like swipe lock, PIN lock, pattern lock, gesture lock, facial lock, etc. These passwords are susceptible to many attacks such as bruteforce, dictionary attacks. But we need to follow preventing strategiesto meaningfully improve our mobile security. This paper presents a study on various attacks on mobile passwords and educates the user about mobile forensics.
Keywords—security,bruteforce,dictionaryattacks, Android, mobile forensics.

I. INTRODUCTION

There are three types of pass codes Android devices currently support. The Android OS uses a variant of the Pass-Go scheme[2]i.e. pattern locking system to increase usability and to adapt for the small screens found on typical devices running Android. Pattern lock uses 9 points arranged in a 3x 3 grid, typically on a touch screen. The user should select at least four points, no point can be used twice, only straight lines are allowed.Fig 1 shows the grid and Fig 2 shows the picture of how android user locks his mobile by using a pattern. There is no easy way to directly calculate the number of possible patterns that follow these rules.But one can easily count all possible patternsand there are 2^{19} probable patterns, which would be sufficient for users to choose their patterns uniformly from this set. Mathematically, it's not a very big deal having all combinations between 1234 and 987654321.

The earlier versions of Android such as KitKatpassword systems are not using salted hashes.Salted hash has an advantage that even though the hash is cracked you cannot get the password. A salted hash has an advantage that even though the hash is cracked, you cannot get the password. Android pattern locks are not salted hashes.Generating a SHA-1[7] hash for the pattern is not more secure, but it is at least a second level of protection. First level is that the file is in a location only readable by the system process.

Android KitKat pattern locks and Android Lollipop earlier versions are not using salted hashes.But the latest versions of Android like Marshmallow authentication systems used salted

hashes using Scrypt algorithm and are somehow strong against dictionaries and rainbow tables but still stored salts are susceptible to bruteforce.

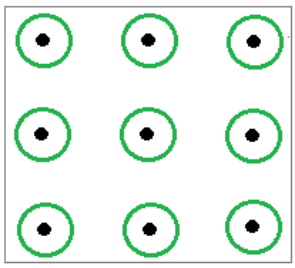

Fig 1

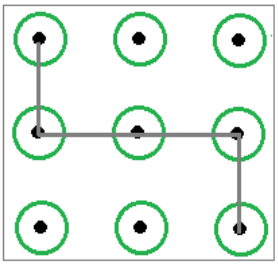

Fig 2

1.1 PIN Locking:

As shown in Fig 3 the second type of password system is the simple personal identification number (PIN) which is commonly found on other mobile devices. Numeric locker are not that much constructive to use it infront of unknown people because if someone finds what is the password then he can get access to the device, either his friend or an opponent , who wants his data from the device.

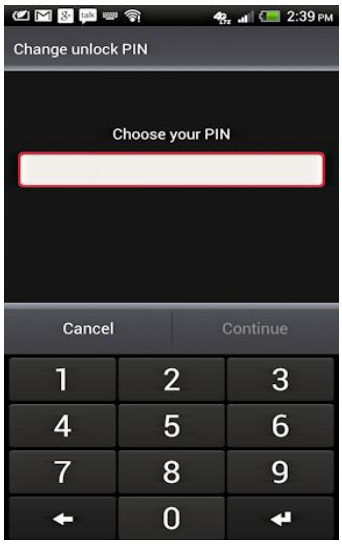

Fig 3

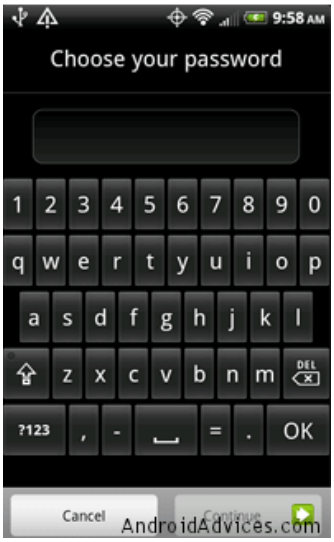

Fig 4

1.2 PASSWORD Locking:

The final type of pass code currently created on Android devices is a full, alphanumeric code, as shown in Fig.4. This is the next upgrading of locking is alphanumeric locks. The device settings will allow you to enter in a new password of your choosing. Here you'll have the full keyboard to select from. It means you've got letters, numbers and symbols to choose from. Some password schemes add special characters so if you want to use a letter with an accent over the top you can do it. Our password will need to fall somewhere in between 5-12 characters. It can be a combination of numbers, letters and symbols allowing us to make the most complicated and therefore most safe combinations.

II. ATTACKS ON ANDROID PASSWORD LOCKS

Any securitymethod comes with some restraint or the other. It is prerequisite to figure out which technique would help you depending on the situation. We will take a few use cases and see how you can people guess the passwords in each case.

2.1 Shoulder Surfing

Passwords can be gained by making watching on the user. Congested circuit hidden cameras can be placed for tracking the entry made into the system. This attack is attainable on all available graphical schemes as well as on text based passwords. Mobile terminals are normally locked when put in a pocket or bag. The occurrence of the authentication will increase whenever a user uses the mobile terminals. Therefore, it is significant to consider the usability of an authentication system. Most of the authentications algorithms suffer from shoulder surfing. Shoulder surfing is the process in which authentication information is intentionally obtained by a person peeking over the shoulder of the user while he completes the authentication process and types the password. Therefore, the research and development of an authentication method that is resistant to shoulder surfing is essential.

2.2 Smudge Attacks:

This was not particular to any android device but used commonly by forensic analysts where they can deduce the password of a touch screen mobile. The attack depend on the fact that smudges[1] are left behind by the user's fingers because they make repeated swiping across same locations. The pattern lock or any lock system of the mobile is something that the user will have to swipe every time that he wants to use his mobile. So we can infer that the smudges would be more from corner to corner the same places and hence under proper lighting and high resolution pictures we can assume the code. So while examining any device, forensic analysts make research and usually take care to avoid hand contact with the screen so as avoid smudge attacks. Figure 5 shows the smudge on the mobile.

Fig 5

2.3 Social Engineering:

Social engineering is the skill of manipulating people so they give up secret information. The criminals are usually trying to trick the people into giving them our passwords or credit card information, or access to our computer to secretly install malicious software, that will give them admission to your passwords and bank information as well as giving them the whole control over your computer.Attackers use this type attacks because it is regularly easier to develop your natural inclination to trust than realizing your software. For example, it is much easier to fool someone into giving you their password than it is for you to try attackingtheir password.

2.4 Brute Force Attack:

Text based passwords have limited password space. In this attack, the attacker tries every feasible value for a password until they achieve something. A brute force attack, if practicable computationally, will always be winning because it will primarily go through all possible passwords

given the alphabet used (lower case letters, upper case letters, numbers, symbols, etc.) and the maximum length of the password.

A system will be particularly vulnerable to this type of an attack if it does not have a correct enforcement instrument in place to guarantee that passwords chosen by users are robust passwords that comply with a suitable password policy. In practice a pure brute force attack on passwords is rarely used, unless the password is supposed to be weak. It is not much effective compared to password cracking methods like dictionary attacks, rainbow tables. If we know the half-finished password,you can employ pattern-based Brute-force crack to make the password retrieval faster. If the password length is 12 and ends with "123", then the command given below will reduce the time to crack the password.

HashCrackerControl.exe –b –c "abyz" -1 12 –p "secret???123" <put_hash_text>

III. DICTIONARY AND RAINBOW TABLE ATTACKS

We can think of a Rainbow Table as a huge dictionary with pre-calculated hashes and the passwords from which they were calculated. The variation between Rainbow Tables and other dictionaries is simply in the method how the entries are stored. The Rainbow table is optimized for hashes and passwords, and thus achieves excessive space optimization while still maintaining good look-up speed. But in essence, it's just a dictionary.

Pattern lock data is kept in a file named gesture.key[8] and stored in the /data/system folder. Lock sequence is encrypted with a SHA1 hashing algorithm. Since SHA1 is a one-way algorithm there is no inverse function to convert hash to original sequence. To restore the code the attacker will need to create a table of sequences with hash strings. The best way here could be to have a dictionary to recover the pattern. You can download this dictionary and then easily discover hash that will recover the original pattern. If we select a pattern 1478,this pattern is stored with a 20-byte SHA-1 hash. So the SHA-1 hash for 1478 is:"06CF96F30A7283FF7258FCEF5CF587ED51156C37", which is stored in a file called gesture. key in /data/system folder in Android's internal memory of the device.

In the case of PIN/alphanumeric passwords, If USB debugging[4] in android is enabled, then bypassing the lock code can be done in matter of seconds. The attacker can use the following method to take advantage of the USB debugging to bypass the screen lock. He taps the gesture.key file by connecting the device to a machine where Android SDK is installed. He runs the command "rm/data/system/gesture.key" to remove the key file.

But if the attacker wanted to be acquainted with the actual PIN so that he can lock/unlock at any time,in android, the PIN that user enters is stored in "/data/system/password.key" file. As we mightexpect, this key is not stored in plain text. It's first salted using a random string, then the SHA1-hashsum and MD5-hashsum are calculated and concatenated and then the final result is stored. This seems very complicated but not to the modern computing power. Here the hash is salted. It is not possible to employ a regular dictionary or rainbow table attack, but people can follow steps given below to crack the PIN value:

1) Get out the salt using adb. The value of the Salt is stored in the 'secure' table from , "/data/data/com.android.providers.settings/databases/settings.db".
2) Get the password : sha1+md5: (40+32) This is stored at " /data/system/password.key".
Ex: 0C4C24508F0D29CF54FFC4DBC5520C3C10496F43313B4D3ADDFF8ACDD5C8DC3CA69CE740.
3) Once we have the md5 and the salt we can use brute force using the available tools available in the market .

4)The prerequisites for the above attacks is the mobile should be rooted and USB enabled.

IV. VULNERABILITIES OF LATEST ANDROID PASSWORD SCHEMES SUCH AS MARSHMALLOW

Android Marshmallow introduces a new password system service called gatekeeper, which is accountable for converting plain text passwords to password handles which can be safely stored on disk. The Android M uses scryptalgorithm to hash PINs or passwords, unlock patterns, and also provides more better protection against brute-forcing than the formerly used MD5 and SHA-1 hashes. But still they use salted hashes, and salted hashes are vulnerable to bruteforcing. A rainbow table is built by an attacker. By using a salt, you cannot construct a rainbow table for a particular algorithm prior to an attack. A salt is not meant to be undisclosed, you store it alongwith thepassword in the database.x = hash(salt+password) . We will store it in the database in the format of salt+x. This renders rainbow tables and hash tables useless. But once the salt is attacked, because gaining 100% security for any authentication system is not possible, if the attacker finds someway to gain the salt value, he can still experiment the bruteforce attack, to gain the password.

V. ANDRILLER(A FORENSIC TOOL TO CRACK ANDROID PASSWORDS)

Andriller is a software utility with a collection of forensic tools for smart phones. It performs read-only, forensically sound, constructive achievement from Android devices. It has other features, such as dominant Lock screen cracking for Pattern, PIN code or Password. Extraction and decoders[6] produce reports in HTML and Excel (.xlsx) formats. Figure.6 and Figure.7 show the how this tool can be used to crack android pattern and PIN passwords.

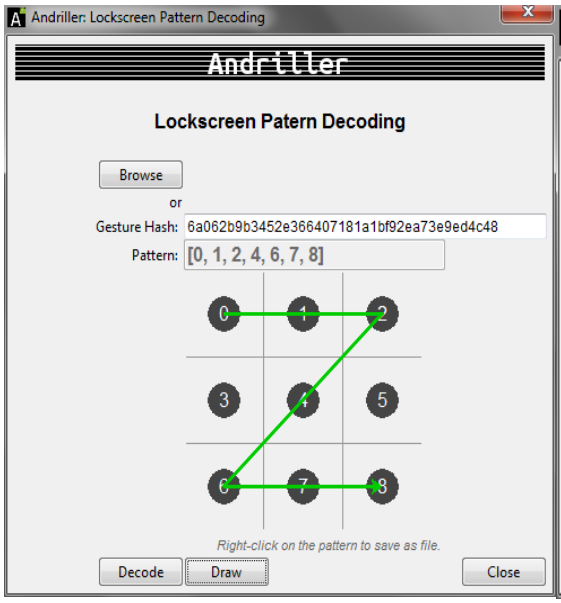

Fig 6

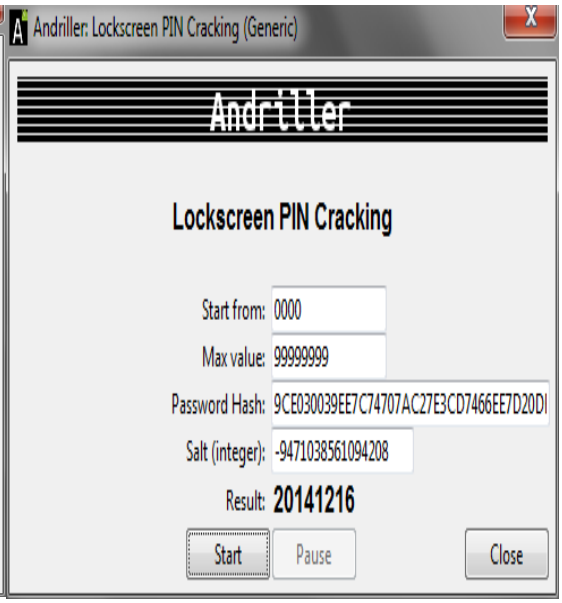

Fig 7

VI. DEFENSE MECHANISMS AGAINST MOBILE FORENSICS

To overcome shoulder surfing puzzle-based methods and graphical passwords[3] can be experimented. It is difficult to do brute-force attack on graphical passwords. We believe it is harder for this attack to succeed for graphical passwords. Draw-A-Secret is resistant to this attack all other authentication techniques may be vulnerable to this attack.

The social engineering attacks also almost impossible in graphical passwords[5] as keyboard input is not involved words in dictionary, so they cannot crack the password. We must delete any request for financial information or passwords. If you get asked to reply to a message with

personal information, it's a scam.Pairing based and Hybrid based authentication can be used to protect the alphanumeric passwords. A salted pattern password mechanism relatively decrease the security threats. For PIN password locking systems any mechanism that generates salts from a secret value rather than storing it in the device, is more secure than existing schemes. The widgets of the cell phone should not allow people to access potentially critical functions like our Gmail. We should explore few ways to defense against brute force attacks on salts also.

VII. CONCLUSION

Security and usability are measured as the most vital factors of the system design that increase the user responsiveness of the system. Android mobile is one such system that requires high user friendliness. Because of several attacks on android mobiles have augmented in present times, highly protected authentication scheme has become essential.

There are no difficulties hacking or bypassing the above kinds of defense schemes for Android mobile users , the only real obstruction is that we make it impractical to directly access the /data/system/ folder and gesture.key file except when we are dealing with a rooted device and USB enabled device. We need new security approaches that avoid undesired taps on the mobiles and offers better authorization scheme than the existing one with regard to rainbow and dictionary attacks. Further security improvement of these security schemes is very much essential to withstand to the above security threats. When hashing passwords, the two most significant considerations are the computational cost, and the salt. The more computationally expensive the hashing algorithm, the longer it will take to brute force its value. We need to make use of salted hashes for pattern password schemes and for PIN passwords we need alternative mechanisms so that to avoid storing the salt values that are subjects to attacks.

REFERENCES

1. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge Attacks on Smartphone Touch Screens. In USENIX Workshop on O_ensive Technologies (WOOT), 2010.
2. Passfaces Corporation. The Science Behind Passfaces.White paper, available at http://www.passfaces.com/enterprise/resources/white_papers.htm.
3. Guidelines on Cellphone Forensics: http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf.
4. R. Morris and K. Thompson. Password Security: A Case History. Communications of the ACM,22(11):594:597, 1979.
5. E. Dirik, N. Memon, and J.-C.Birget.Modeling User Choice in the PassPointsGraphical PasswordScheme. In Symposiumon Usable Privacy and Security (SOUPS), 2007.
6. "Cellphone Forensic Tools: An overview and Analysis" available at http://csrc.nist.gov/publications/nistir/nistir-7250.pdf
7. Android Forensics:How To Bypass The Android Phone Pattern Lockhttp://niiconsulting.com/checkmate.2014/04/how-to-bypass-the-android-phone-pattern-lock/
8. PHP Password Hashing Manual available at php.net/manual/en/function.password-hash.php.

