Android Reverse Engineering

Nicholas Rocksvold • Dr. Michael Soltys • COMP 499

Introduction and Hexdump Part 1

The purpose of this project is to examine the contents of an android phone only given the binary file. The goal of this project is to obtain the lock-screen password of the phone so the data can be accessed.

I began by performing a hexdump of the file.

The command I used to perform the hexdump is:

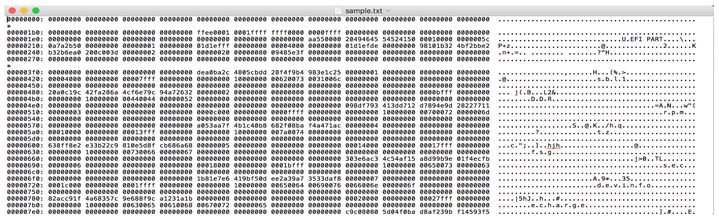
xxd -e -a -c 48 EMMC_ROM1_QE13MB_3007443627_00000000_3A3E00000.bin hexdump.txt

The result of that command was a file that was 43GB. I could not open it in a standard text editor so I had to split the file with this command:

split -b 1000m hexdump.txt hexpart

This split the file into 1GB pieces named:

hexpartaa,hexpartab,hexpartac,hexpartad,hexpartae,hexpartaf,hexpartag,hexpartah,hexpartai,hexpartaj,hexpartak,hexpartal,hexpartam,hexpartan,hexpartao,hexpartap,hexpartaq,hexpartar,hexpartas,hexpartat,hexpartau,hexpartav,hexpartaw,hexparttax,hexpartay,hexpartaz,hexpartba,hexpartbb,hexpartbc,hexpartbd,hexpartbe,hexpartbf,hexpartbg,hexpartbh,hexpartbi,hexpartbj,hexpartbk,hexpartbl,hexpartbm,hexpartbn,hexpartbo,hexpartbp



Looking for Strings

To see all the strings in the binary file I used the command

strings -a -t x EMMC_ROM1_QE13MB_3007443627_00000000_3A3E00000.bin > allstrings.txt

This command found all of the strings in the binary file of at least length 4

This command found all of the strings in the file of at least length 64

strings -a -64 -t x EMMC_ROM1_QE13MB_3007443627_00000000_3A3E00000.bin > strings64.txt

Here is another command for strings I used

strings -a EMMC_ROM1_QE13MB_3007443627_00000000_3A3E00000.bin > stringsnonum.txt

This is just all of the strings without the line numbers, this was to make what I did next easier.

Hexdump Part 2

Using the previous strategy the file was very difficult to read, there were not enough ascii characters on the screen at once. I used a new command for the hexdump which is:

xxd -a -c 160 EMMC_ROM1_QE13MB_3007443627_00000000_3A3E00000.bin hexdump.txt

Since I set the column number to 160 the ascii characters alone cover my screen. I wrote a few lines of code to get rid of the hex numbers.

```
f = open('hexdump.txt')
lines = f.readlines()
f.close()
f = open('hexdump.txt', 'w+')
for line in lines:
    f.write(line[:9])
    f.write(line[411:])
f.close()
```

This code goes through the hexdump and writes to a file the line number and the ascii text. The result was a 13GB file which I split into 500m pieces with the command:

split -b 500m hexdump.txt hexpart

Now my files look like this
hexpartaa,hexpartab,hexpartac,hexpartad,hexpartae,hexpartaf,hexpartag,hexpartah,hexpartai,hexpartaj,hexpartak,hexpartal,hexpartam,hexpartan,hexpartao,hexpartap,hexpartaq,hexpartar,hexpartas,hexpartat,hexpartau,hexpartav,hexpartaw,hexpartax,hexpartay,hexpartaz

Hexadecimal Strings

This is another few lines of code I used to get hexadecimal strings out of my stringsnonum file

```
import re

f = open('stringsnonum.txt')
lines = f.readlines()
f.close()
f = open('hexstrings.txt', 'w+')
for line in lines:
    if re.match('^[A-Fa-f0-9]+$',line):
        f.write(line)
f.close()
```

I did this because I'm assuming the phone is hashed with SHA which would be some length of hexadecimal characters

Findings

I think the overall structure of the phone's data is given at the beginning of the hexdump, which is as follows:

Sbl1,DDR,Rpm,Tz,Fsg,Sec,Devinfo,Echarge,Splash,Aboot,Modem,Boot,Recovery,Pad,Pad1,Ztelk,Ssd,Ztecfg,Fsc,Modemst1,Modemst2,Misc,Keystore,Config,Oem,Pad3,Persist,Carrier,Cache,System,Userdata

I've made a couple assumptions. One, I thought initially the lock screen might be in keystore but from what I researched the lock screen doesn't use the keystore feature. My second assumption is that from hexpartah onward is all userdata because it is unreadable.

The very last part of the hexdump(hexpartaz) lists the structure of the phone's data again and also this line above it

aes-xts.essiv:sha256

I'm assuming this is what the phone is encrypted with

I found some lines with the phrase Standard Security Handler followed by what looks like passwords, but it appears that Standard Security Handler is used for PDF documents. Examples of those lines are:

93d4cdc7 _ZN28CPDF_StandardSecurityHandler20AES256_CheckPasswordEPKhjiPh
93d4ce07 _ZN28CPDF_StandardSecurityHandler17CheckUserPasswordEPKhjiPhi
93d4ce45 _ZN28CPDF_StandardSecurityHandler15GetUserPasswordEPKhji

Hashcat Commands

I haven't found the phone's lock screen but I tried to run some of the hex strings I found through a password cracking software called hashcat. Here is the command I used.

./hashcat -a 3 -m 1400 --status --force -o android-password.txt -i --increment-min 1 android-hash.txt ?a?a?a?a?a?a?a?a

This command performs a brute force attack on SHA 256 hashes. The increment is there so I can test password lengths of 1 through 8. The reason I stopped at 8 characters is because at that length it would take 3 years to brute force and longer with more characters

If I had found the lock screen hash I would perform these commands on the hexdump.

./hashcat -a 0 -m 1400 --status --force -o android-password.txt android-hash.txt wordlists/wordlist.txt

I would start with this one since wordlists are very quick. If the wordlist fails I can attempt to brute force again with:

./hashcat -a 3 -m 1400 --status --force -o android-password.txt -i --increment-min 1 android-hash.txt ?a?a?a?a?a?a?a?a

Conclusions

I haven't found the lock screen password for the phone. What I've learned is that reverse engineering is very time consuming. There is not a lot of information out there on reverse engineering which would make sense since phone companies don't release their software structures. The little information I could find was a blog post. https://forensics.spreitzenbarth.de/2012/02/28/cracking-pin-and-password-locks-on-android/

According to this the password should have been next to a string " lockscreen.password_salt" but there is nothing next to this string. This doesn't mean the password is impossible to find it just means that more time will have to be spent looking at the data. There is a lot of it to go through and I could have missed something.