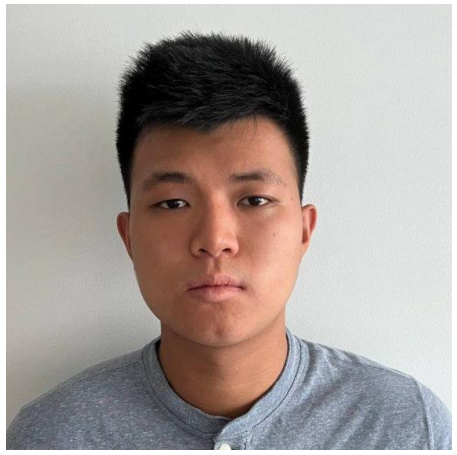


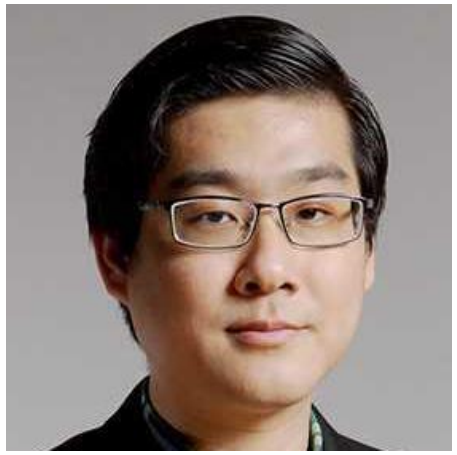


Countering Forensics Software by Baiting Them

Dr. Weihan Goh | Joseph Lim | Isaac Soon

DEF CON 33 Creator Stage, on behalf of the Adversary Village





New Team, Who Dis?

Is This Talk For You?

- Law enforcement / mobile forensics examiner
- Interested in anti-forensic techniques
- Dabble in Android logical extraction
- Everyone!



https://www.linkedin.com/posts/abrignoni_dfir-activity-6928451100442304512-QabW

Agenda

The Big Picture

Honey Tokens 101

Video Demo

Defensive Strategies

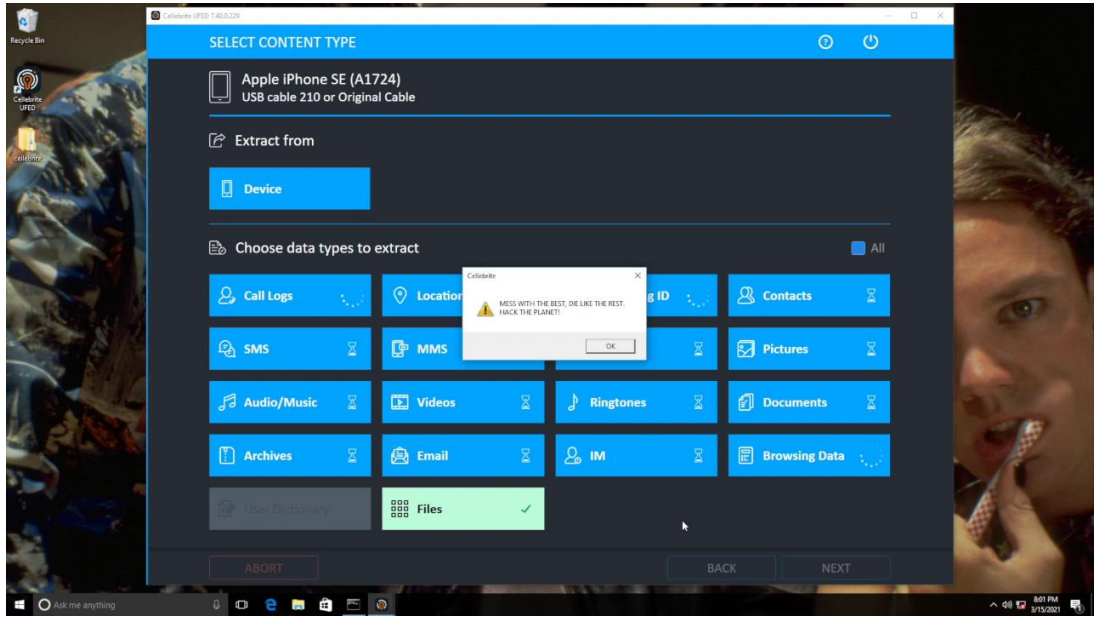
Conclusion

Why "First Stage" Access Isn't Enough

- **Logical extraction workflow**
  1. Gain access to the device (Unlock device / bypass security)
  2. Extract data
  3. ???
  4. PROFIT!

- This assumption underpins the workflow of a number of mainstream forensic software

- Reality check: **Second-layer countermeasures!**



https://signal.org/blog/cellebrite-vulnerabilities/

Rethinking Traditional Anti-Forensics

Conventional anti-forensic tactics

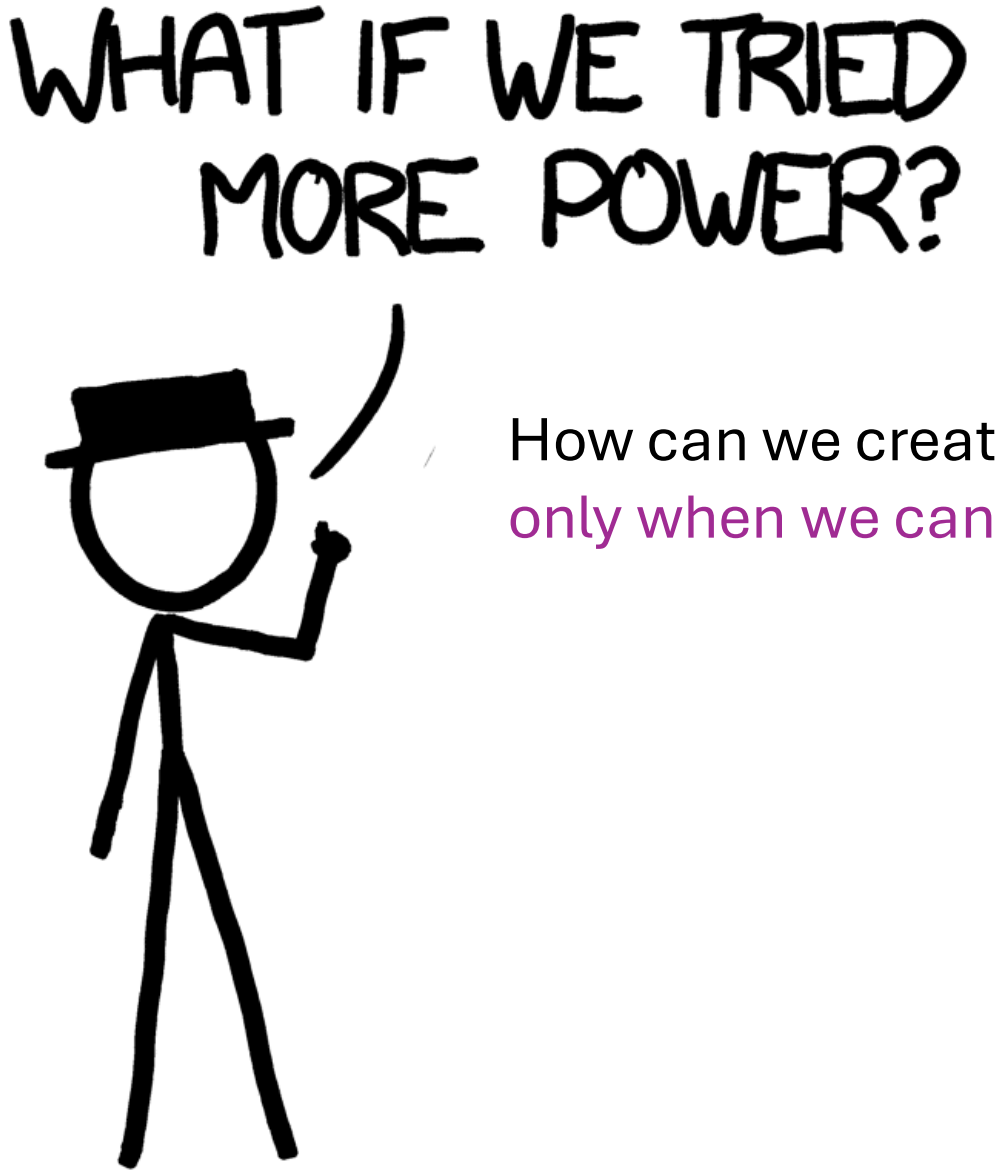
- Disabling USB debugging entirely
- Monitor for forensic software installation, then block ADB connections
- Kill the forer

But…

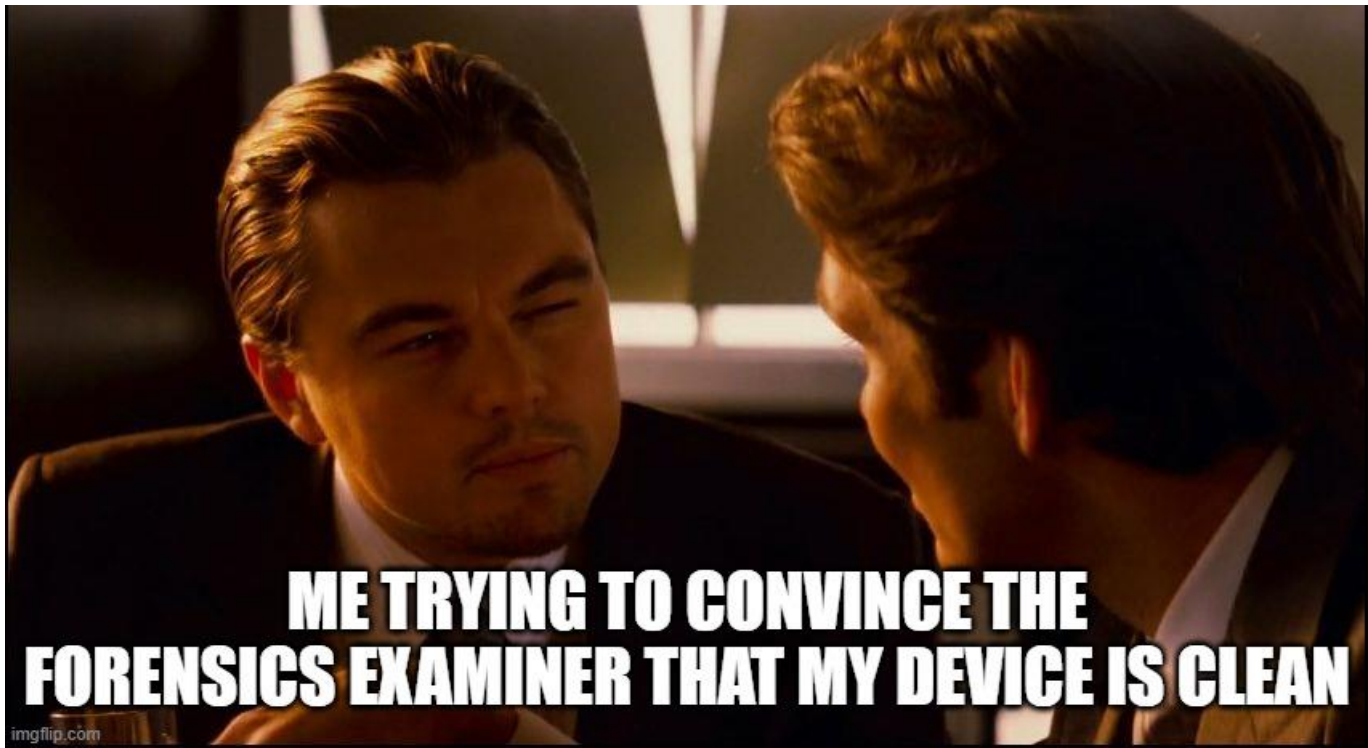
- 
- Once alert iners switch to a or more s, leading my that can l subterfuge



```
if (packageName.contains("belkasoft") || packageName.contain
    //String commandDenyPackage= "su -c pm disable " + packageN
    String commandDenyPackage= "su -c do pkill -f " + packageName;
    Process denyPackage = Runtime.getRuntime().exec(commandDenyPackage);
    denyPackage.waitFor();
}
```



How can we create a system that deploys anti-forensic measures only when we can determine that forensic acquisition is occurring?



https://what-if.xkcd.com/13/

Let Them Fight

Forensic Examiner / Software

1. Gain access to the device (Unlock device / bypass security)
2. Extract data
3. ???
4. WTF?

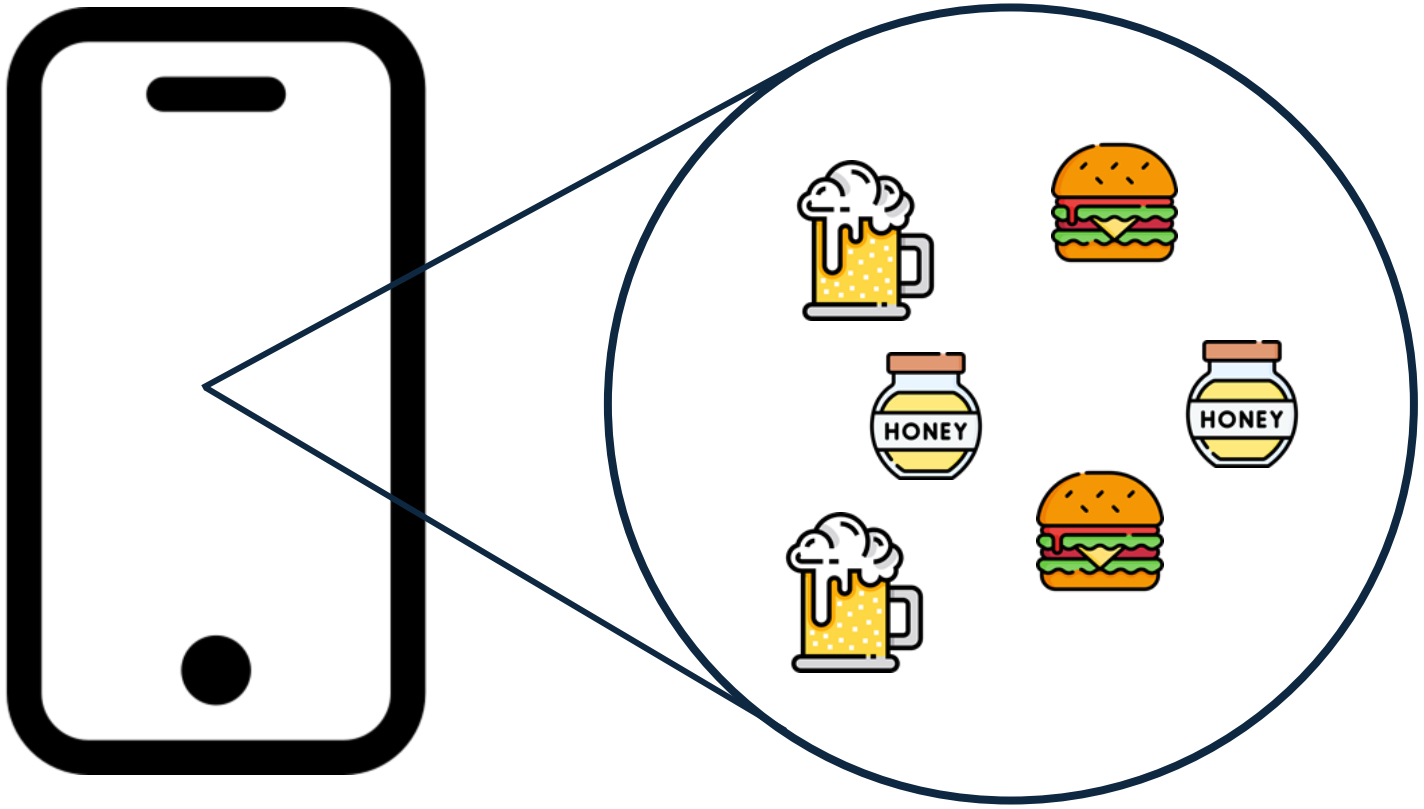


Adversary

1. Let it happen (Show nothing, but do something 🧘)
2. Let it happen
3. ???
4. PROFIT!!

The device intended user only eats burgers and drink beer…

But the forensic examiner consumes all…



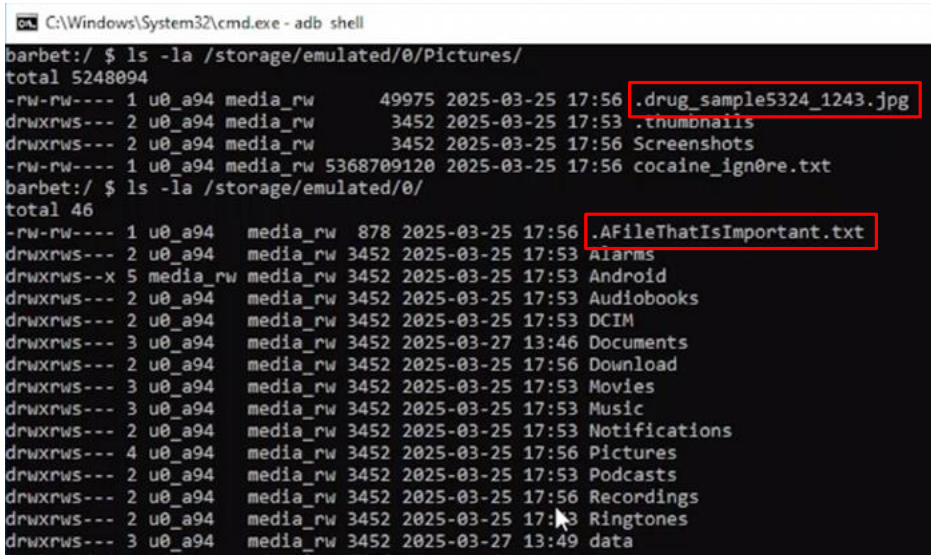
Honey Tokens 101

- We can **litter bait files** containing rubbish information
  1. Deep in within filesystem directories
  2. In top-level directories (e.g., Documents) but files lead with "." suffix

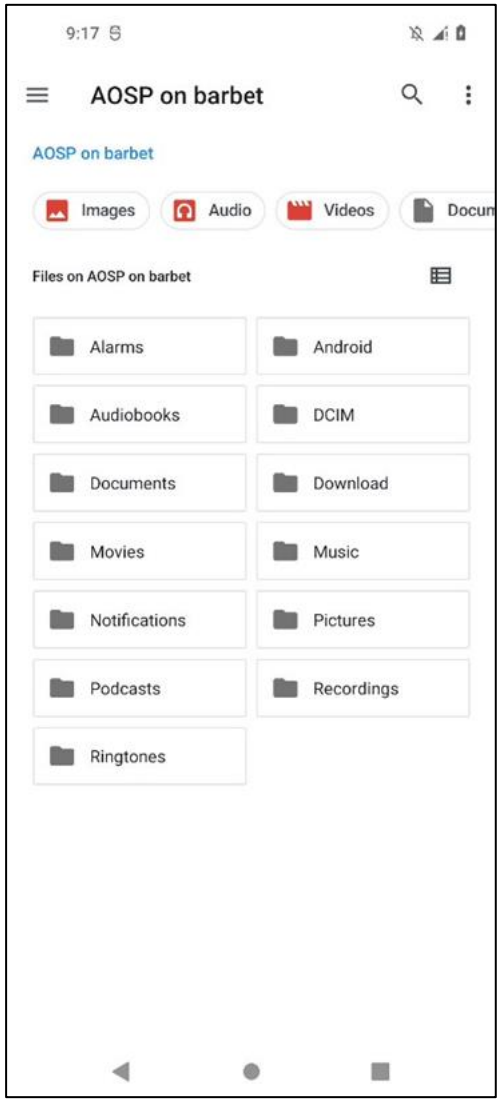
- Can be done programmatically too! When a new app is installed.



Honey Token Deployment

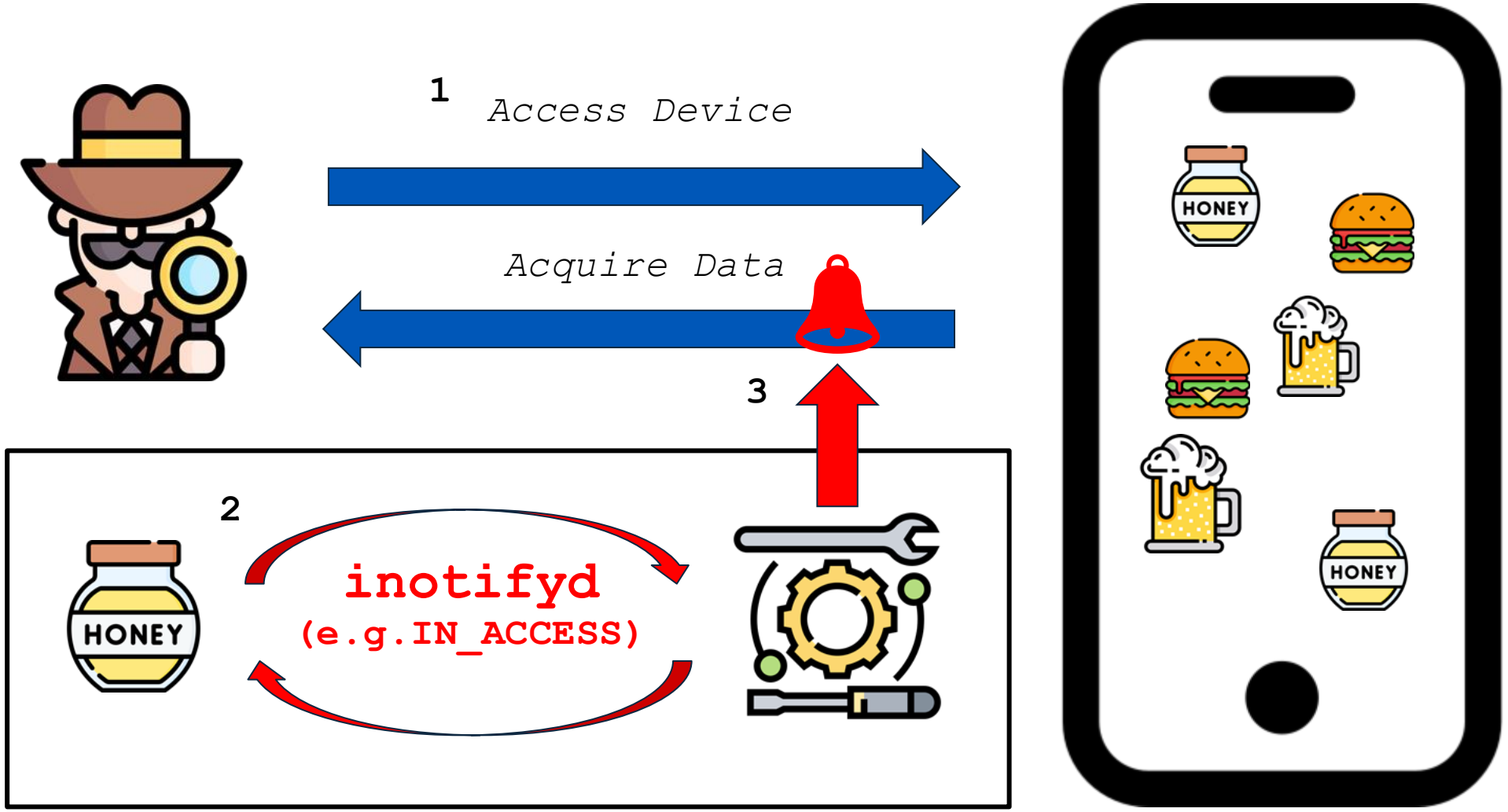



```
barbet:/ # ls -la /storage/emulated/0/Android/data/org.telegram.messenger.web/files/
total 13
drwxrws--- 3 u0_a102 ext_data_rw 3452 2025-07-07 21:18 .
drwxrws--- 4 u0_a102 ext_data_rw 3452 2025-07-07 21:18 ..
drwxrws--- 6 u0_a102 ext_data_rw 3452 2025-07-07 21:18 Telegram
-rw-rw---- 1 u0_a102 ext_data_rw 1326 2025-07-07 21:18 action_history_20250707_211850.log
barbet:/ # ls -la /data/data/org.telegram.messenger.web/files/
total 1048
drwxrwx--x 4 u0_a102 u0_a102          3452 2025-07-07 21:18 .
drwx------ 8 u0_a102 u0_a102          3452 2025-07-07 21:18 ..
drwx------ 2 u0_a102 u0_a102          3452 2025-07-07 21:18 account1
drwx------ 2 u0_a102 u0_a102          3452 2025-07-07 21:18 account2
-rw-r--r-- 1 u0_a102 u0_a102_cache    1772 2025-07-07 21:18 action_history_20250707_211850.log
-rw------- 1 u0_a102 u0_a102          5929 2025-07-07 21:18 bluebubbles.attheme
-rw------- 1 u0_a102 u0_a102          4096 2025-07-07 21:18 cache4.db
-rw------- 1 u0_a102 u0_a102         32768 2025-07-07 21:18 cache4.db-shm
-rw------- 1 u0_a102 u0_a102       1005312 2025-07-07 21:18 cache4.db-wal
-rw-rw---- 1 u0_a102 u0_a102           912 2025-07-07 21:18 tgnet.dat
barbet:/ #
```





Honey Token Implementation



inotifyd

- **`inotifyd`** is a command in the userland that is built on top of **`inotify`** Linux kernel subsystem which is a useful mechanism for monitoring filesystems!
- **`inotifyd`** is part of **`toybox`** which is included in **Android 6.0+**!

```
com.example.sweetdeceit  I  File event detected: r  /storage/emulated/0/Download/.drug_shipping_schedule_mapletree_1914.txt
com.example.sweetdeceit  I  File event detected: a  /storage/emulated/0/Download/.drug_shipping_schedule_mapletree_1914.txt
com.example.sweetdeceit  I  File event detected: 0  /storage/emulated/0/Download/.drug_shipping_schedule_mapletree_1914.txt
```

Legend

`r -> opened IN_OPEN`
`a -> accessed IN_ACCESS`
`0 -> closed (unwritable) IN_CLOSE_NOWRITE`

We have a Tripwire... So Now What?

1. Encrypt data
2. Factory reset the device
3. **Sky's the limit**

BEFORE ENCRYPT

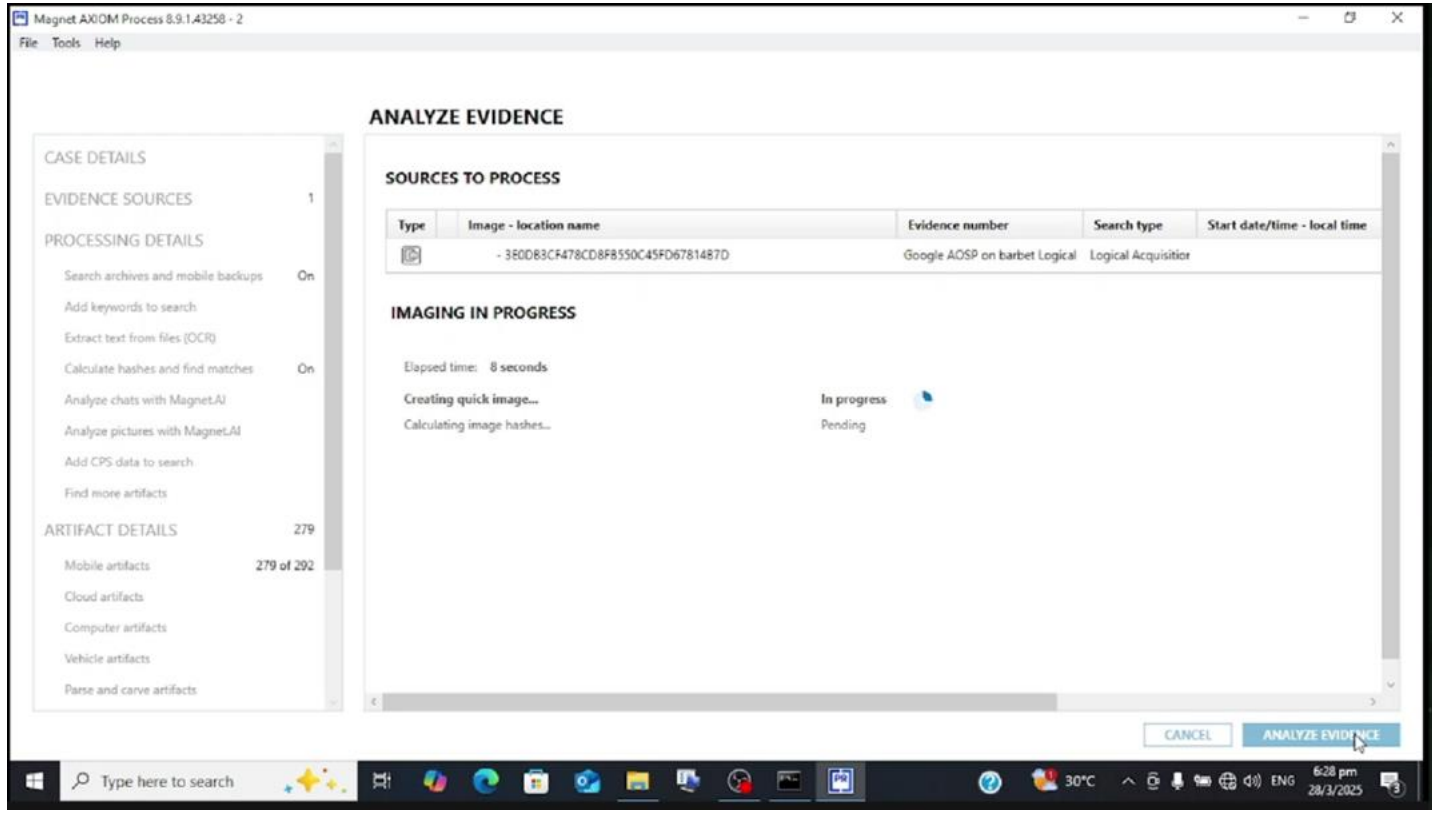
← secret.txt

```
shhhh this is my password: S3cUr3P@ssw0rd
```

AFTER ENCRYPT

← secret.txt

```
E+eN5GGDS2nJmBYS
AuiZXVODWnPHvl+N3j9j5O/uQYllJfXW14YWANzzlFopVymr4aH
mIl6dbQN5YbeM65dw/Bkow==
```









Video demo goes here

Mitigations and Defensive Strategies

- Selective vs Bulk Acquisition
- Exploit Countermeasure Weakness
  - Accelerated Acquisition
- Cold System Forensics



Thank You!

