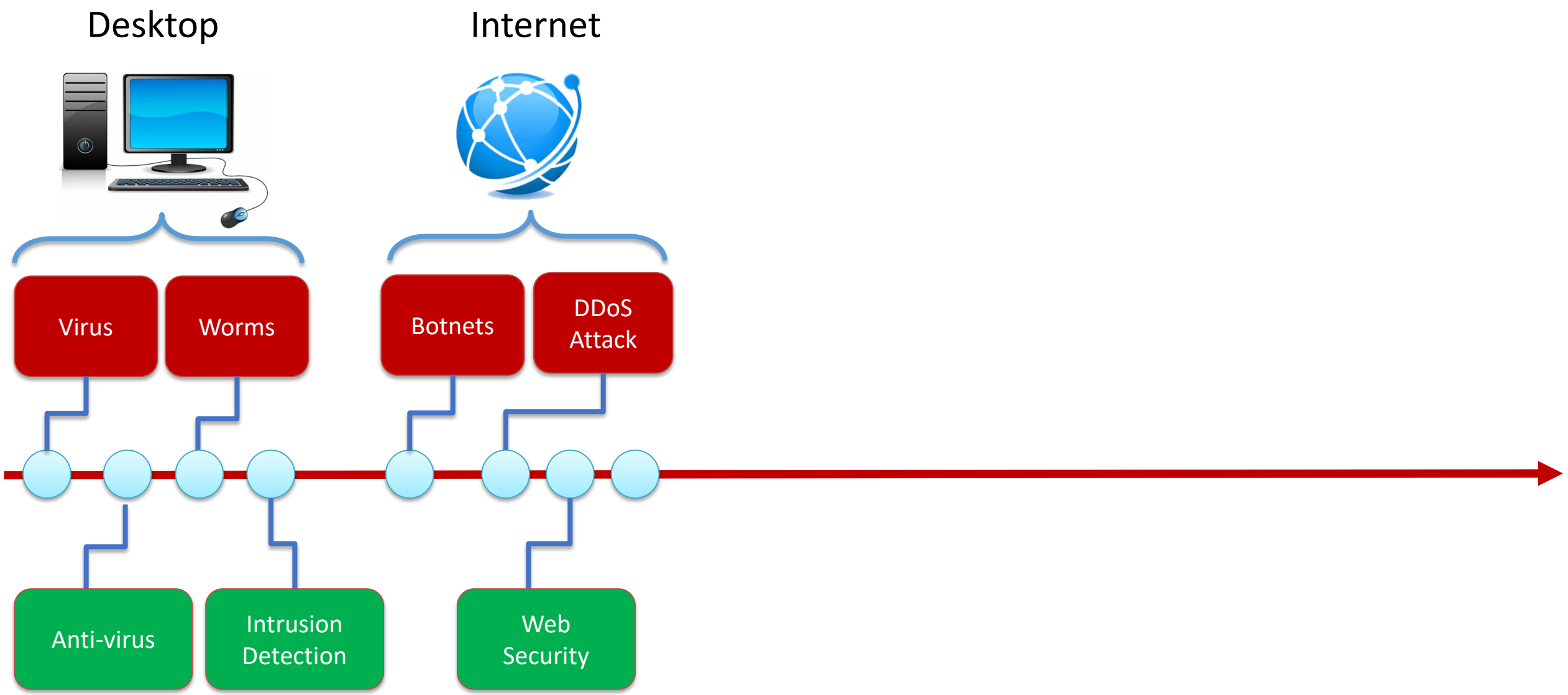
Unveiling Insecure and Privacy-Risky Practice in Mobile Apps with Automated Program Analysis

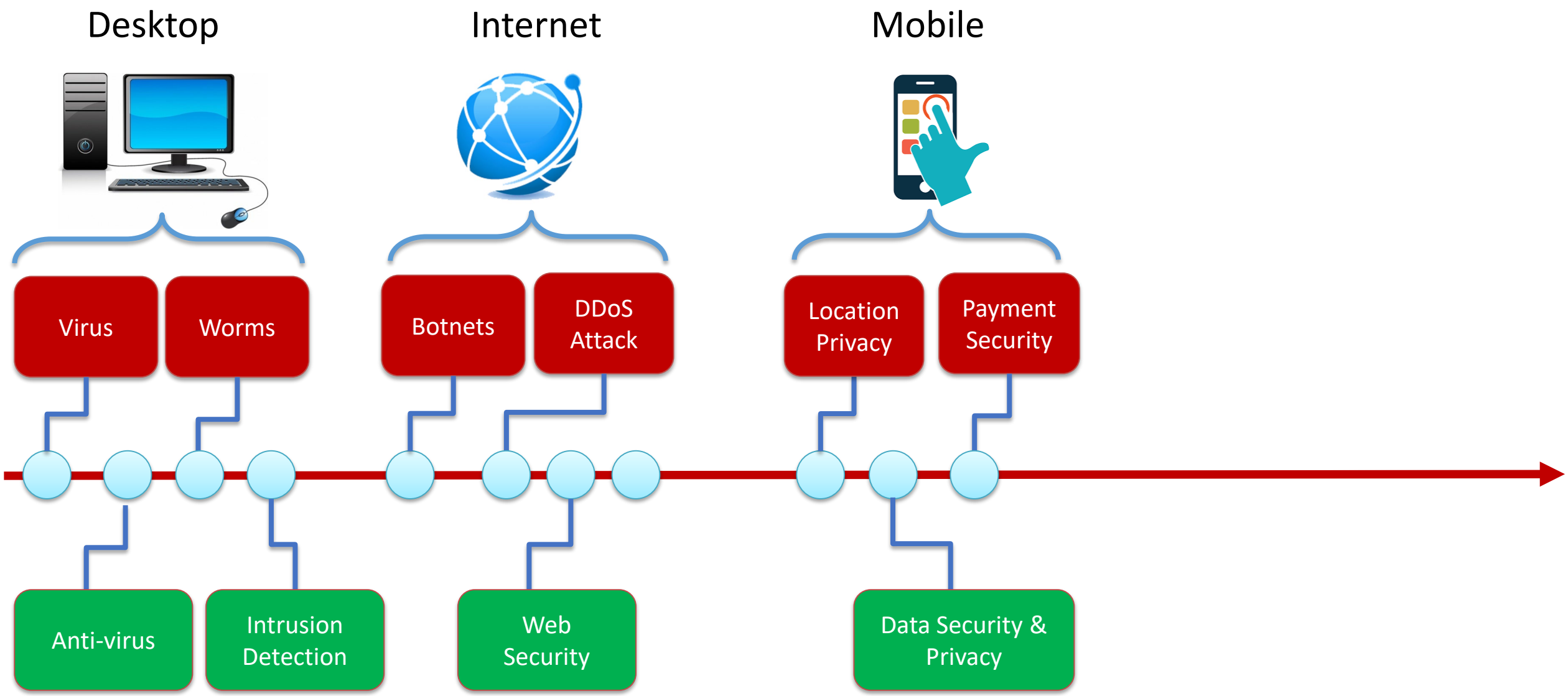
Zhiqiang Lin

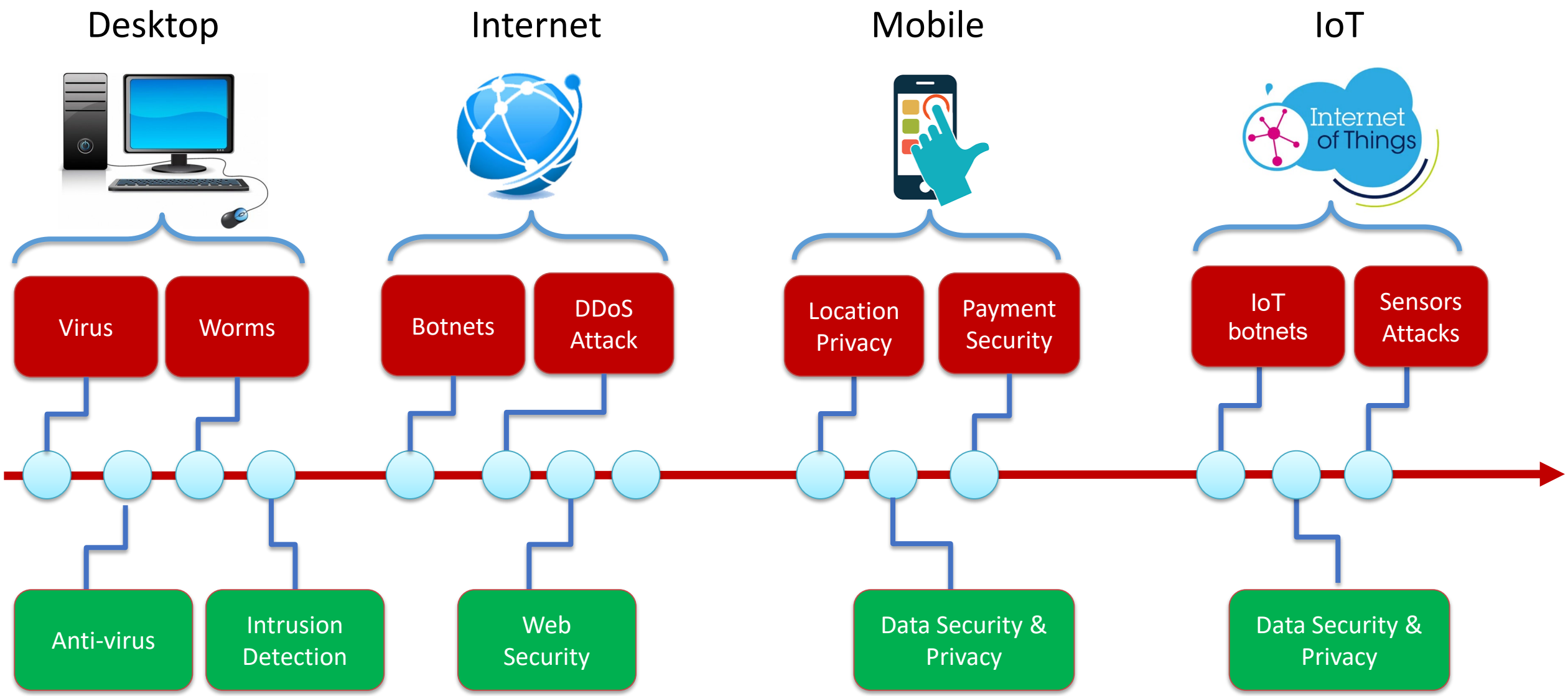
zlin@cse.ohio-state.edu

August 3rd, 2021

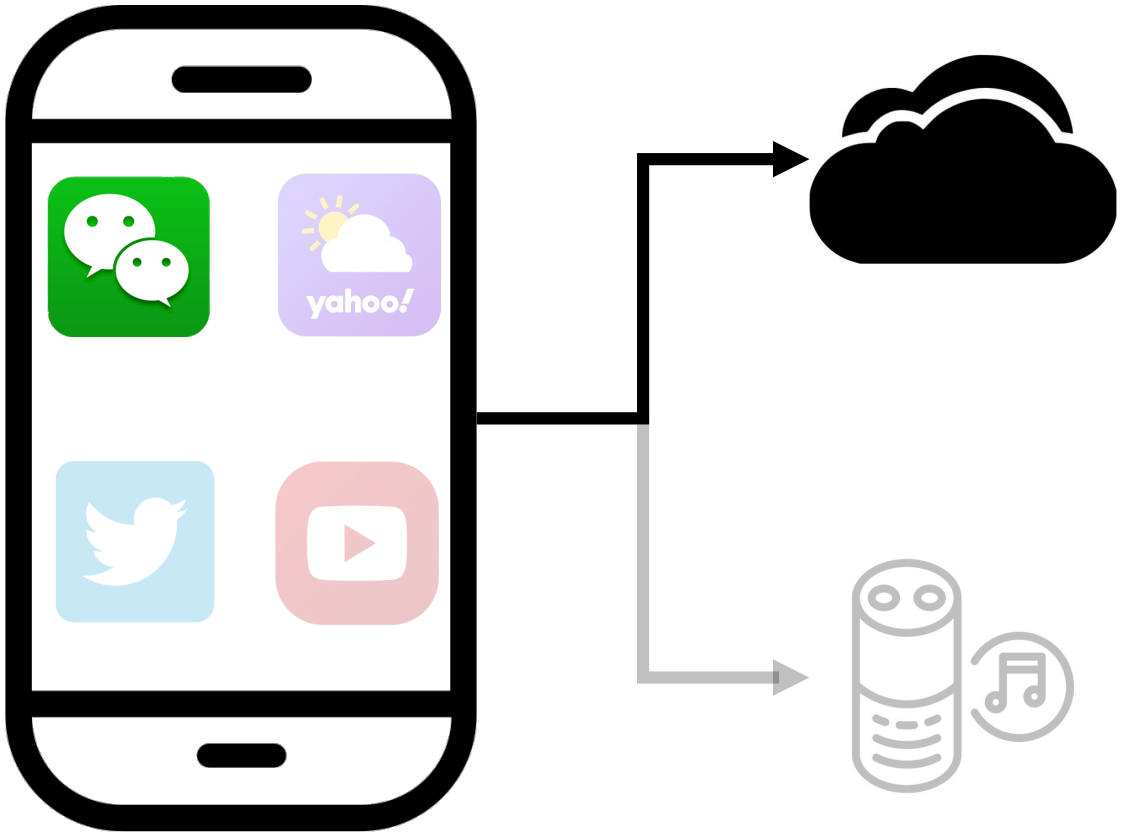
Desktop

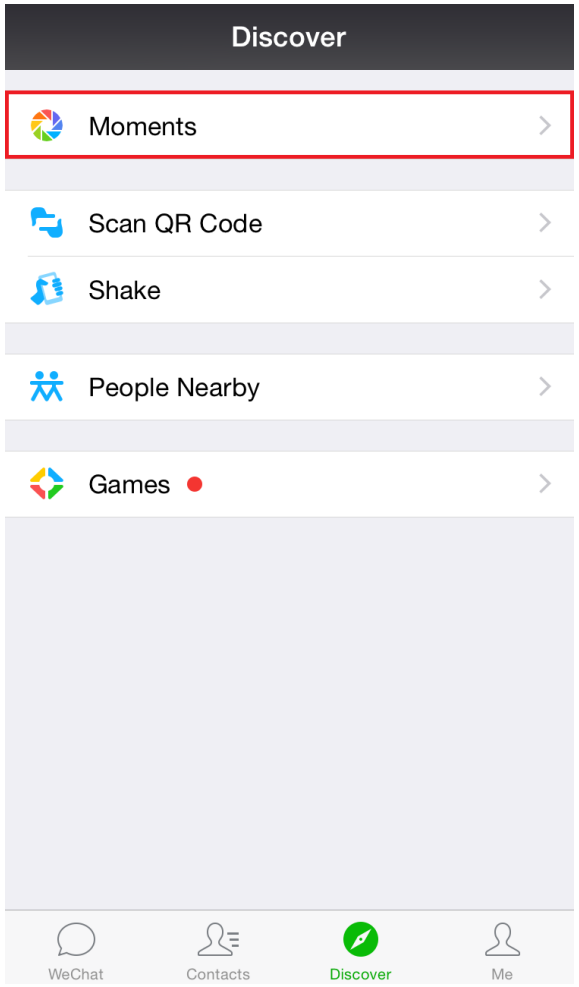

Desktop
Internet
Virus
Worms
Botnets
DDoS Attack
Anti-virus
Intrusion Detection
Web Security

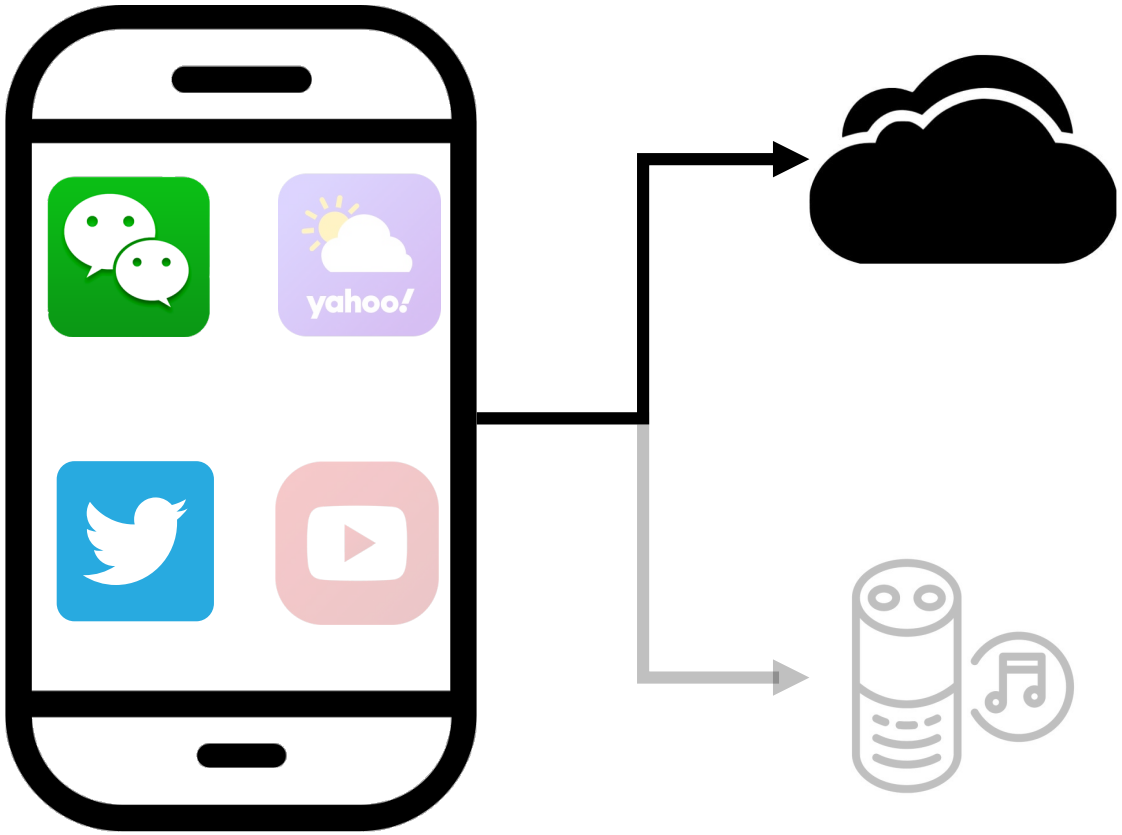

Desktop
Internet
Mobile
Virus
Worms
Botnets
DDoS Attack
Location Privacy
Payment Security
Anti-virus
Intrusion Detection
Web Security
Data Security & Privacy


Desktop
Internet
Mobile
IoT
Internet of Things
Virus
Worms
Botnets
DDoS Attack
Location Privacy
Payment Security
IoT botnets
Sensors Attacks
Anti-virus
Intrusion Detection
Web Security
Data Security & Privacy
Data Security & Privacy

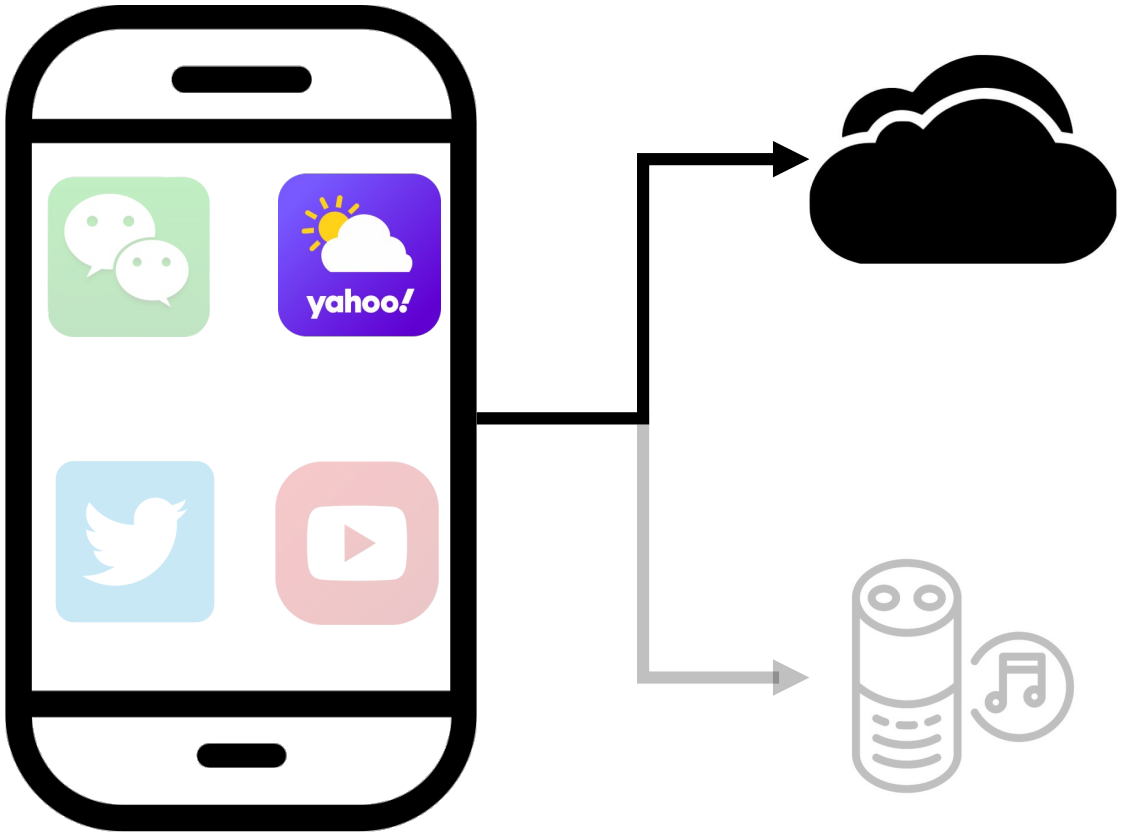
Source: cloudxtension.com


yahoo!

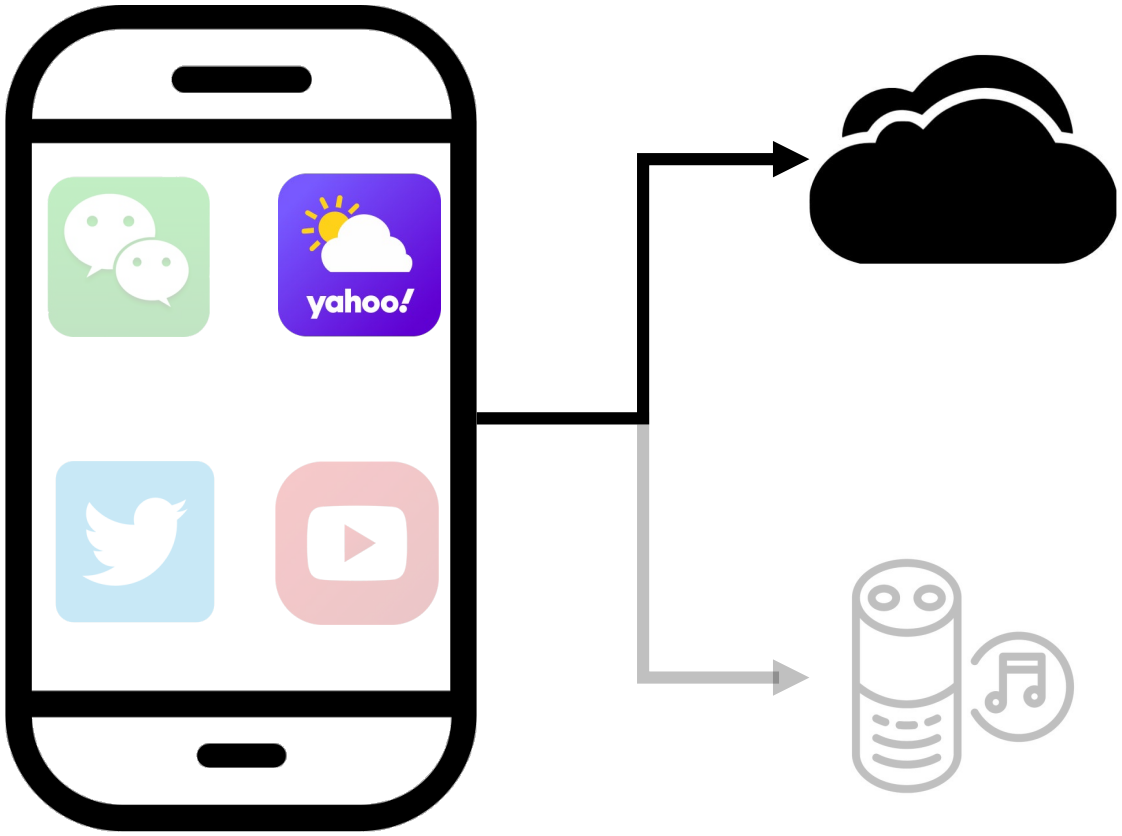

Discover
Moments
Scan QR Code
Shake
People Nearby
Games
WeChat
Contacts
Discover
Me


yahoo!

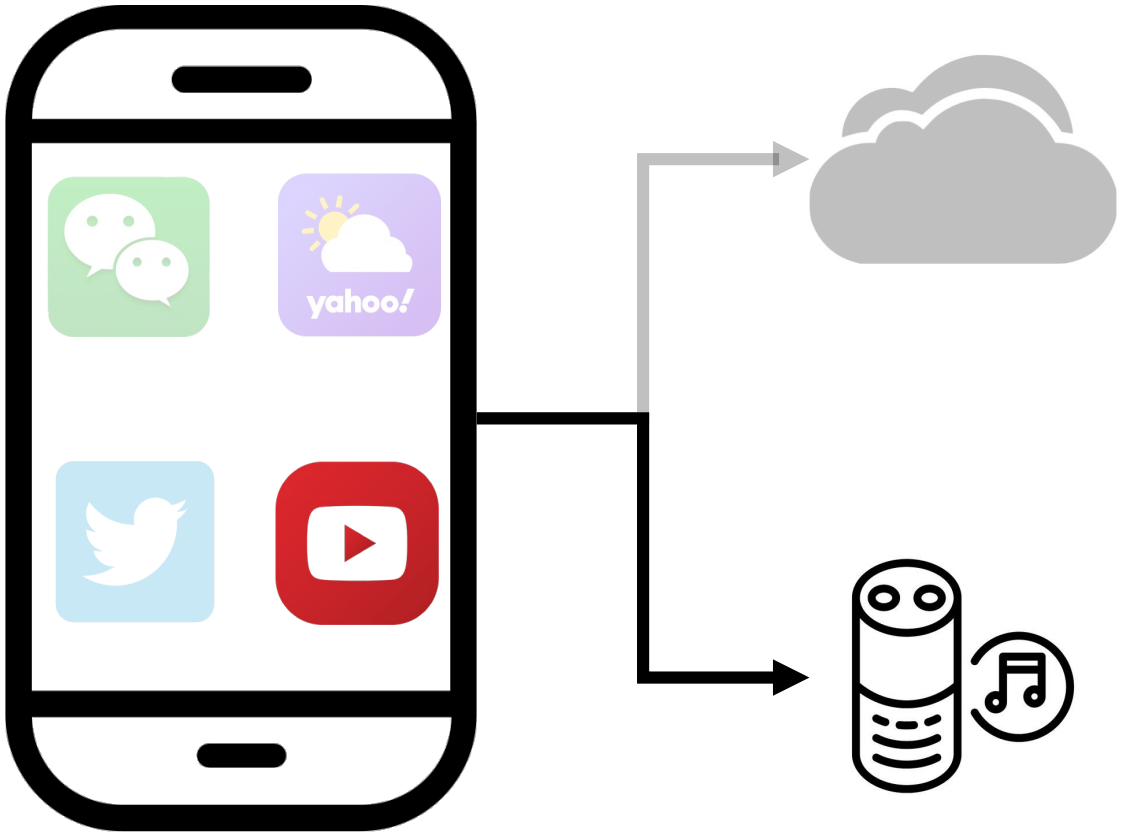

Olympics
15K Tweets
JULY 23 – AUGUST 8
2021
TOKYO 2020
Follow
Olympics
@Olympics
#Olympics - Inspiring people through the Olympic values of friendship, respect, and excellence.
olympics.com
Joined June 2009
5,013 Following
6.3M Followers
Tweets
Tweets & replies
Media
Likes
Pinned Tweet
Olympics @Olympics · 7h
Oh this? Just Olympic champ @TomDaley1994 knitting in the stands while watching the diving.


yahoo!

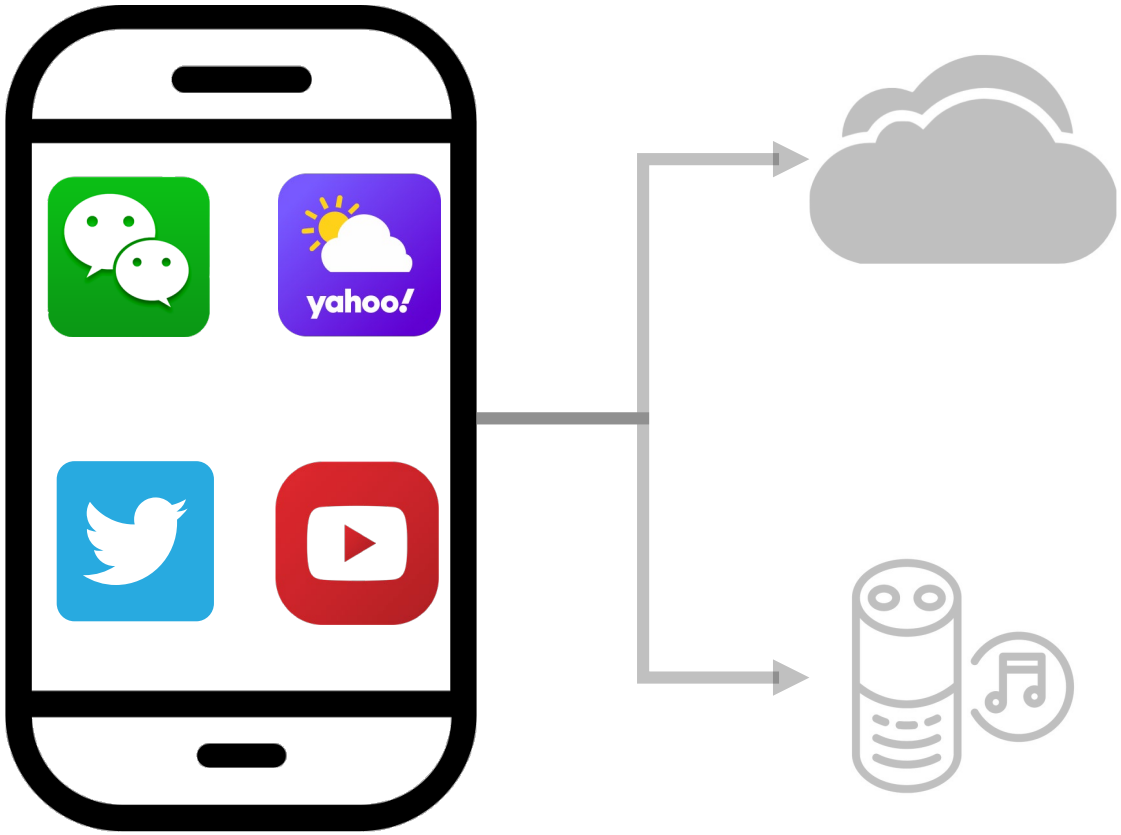

Columbus, OH
16:34 EDT
Cloudy
↑13° ↓-4°
13°
07:44
19:39
12°
17:00
11°
18:00
9°
19:00
7°
20:00
5°
21:00
4°
22:00


yahoo!

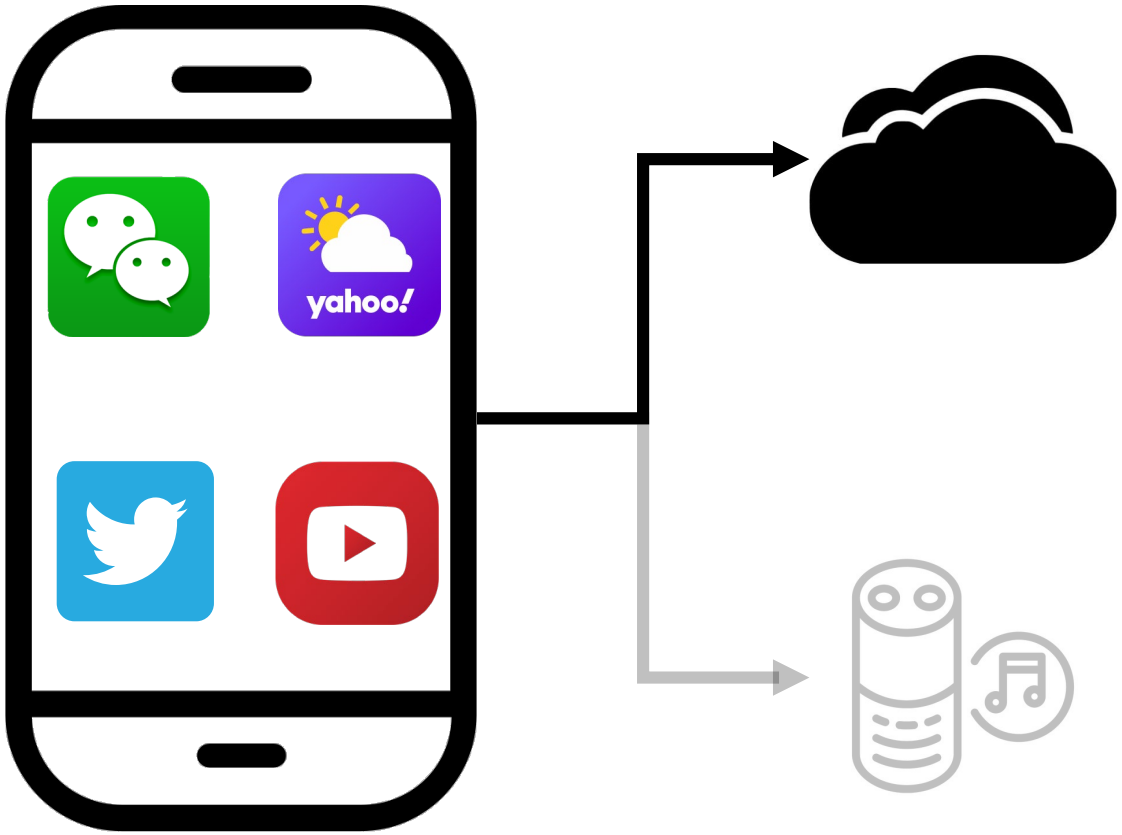

Columbus, OH
16:34 EDT
Cloudy
↑13° ↓-4°
13°
07:44
19:39
12°
17:00
11°
18:00
9°
19:00
7°
20:00
5°
21:00
4°
22:00


yahoo!

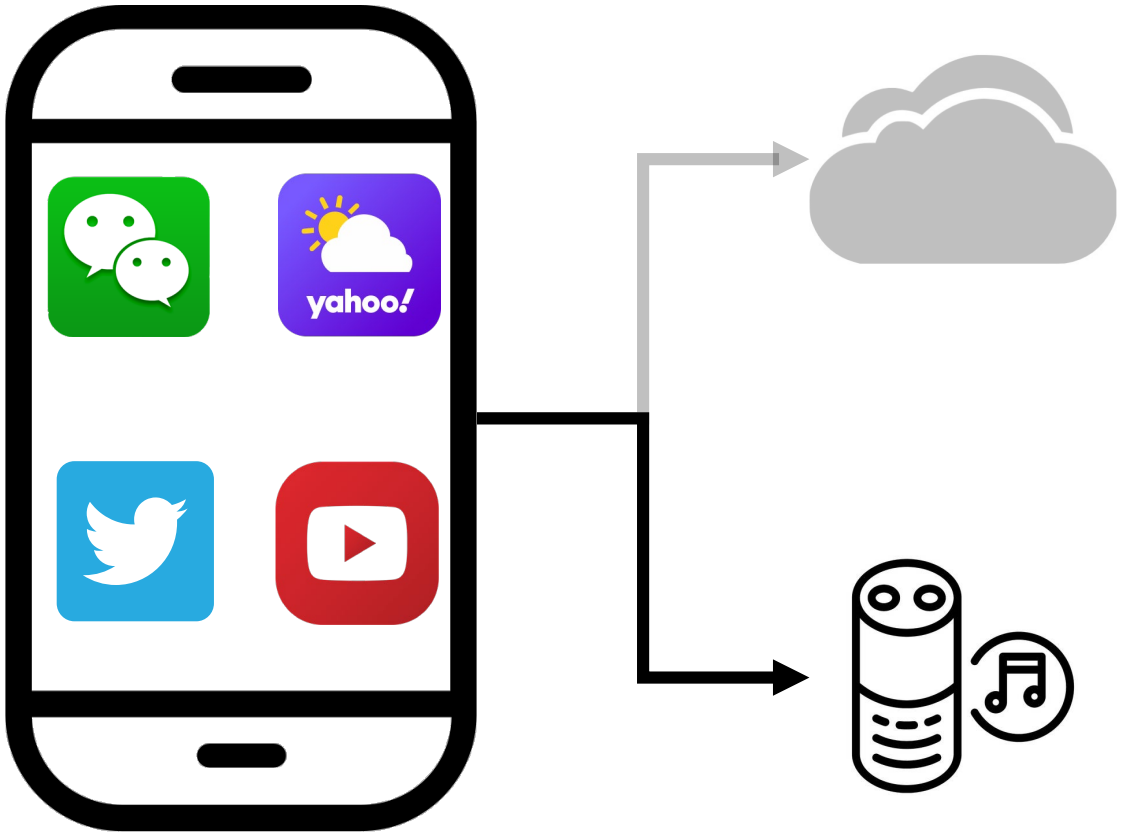

Bloomberg
newsy
npr
CNN
FOX NEWS channel
CNBC
amazon



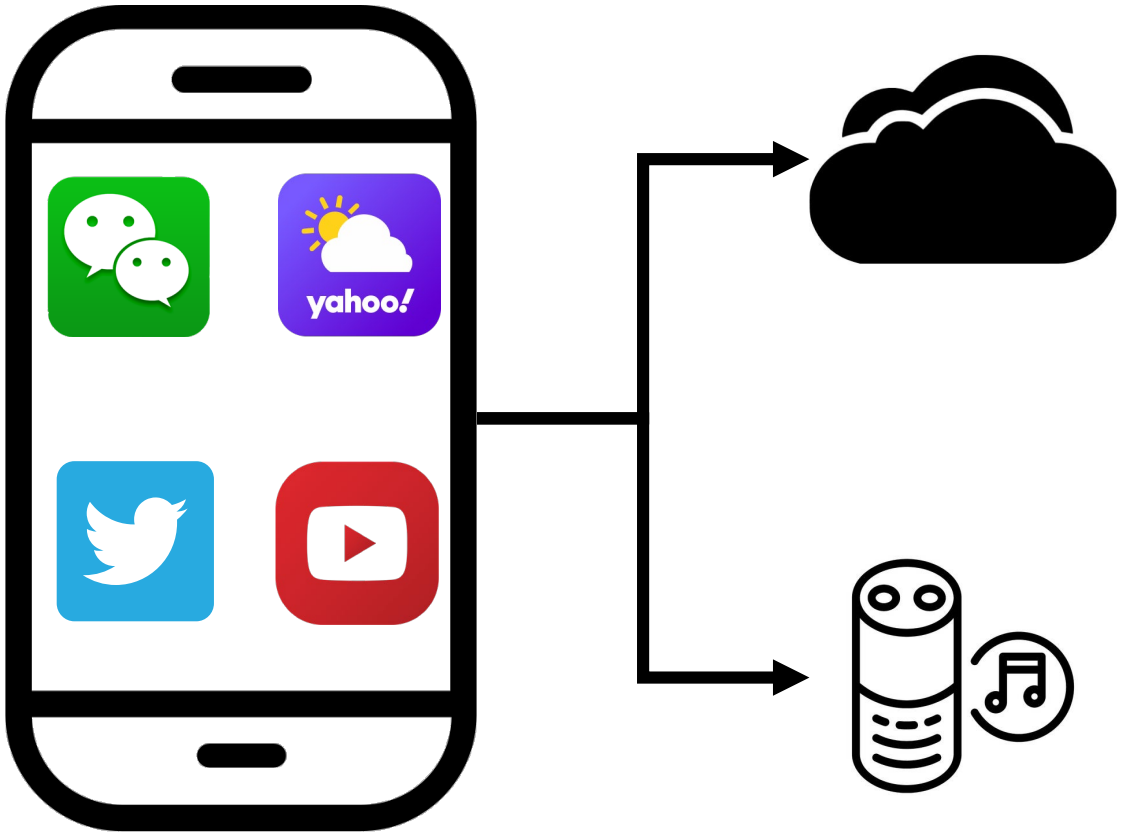
Good Manner?



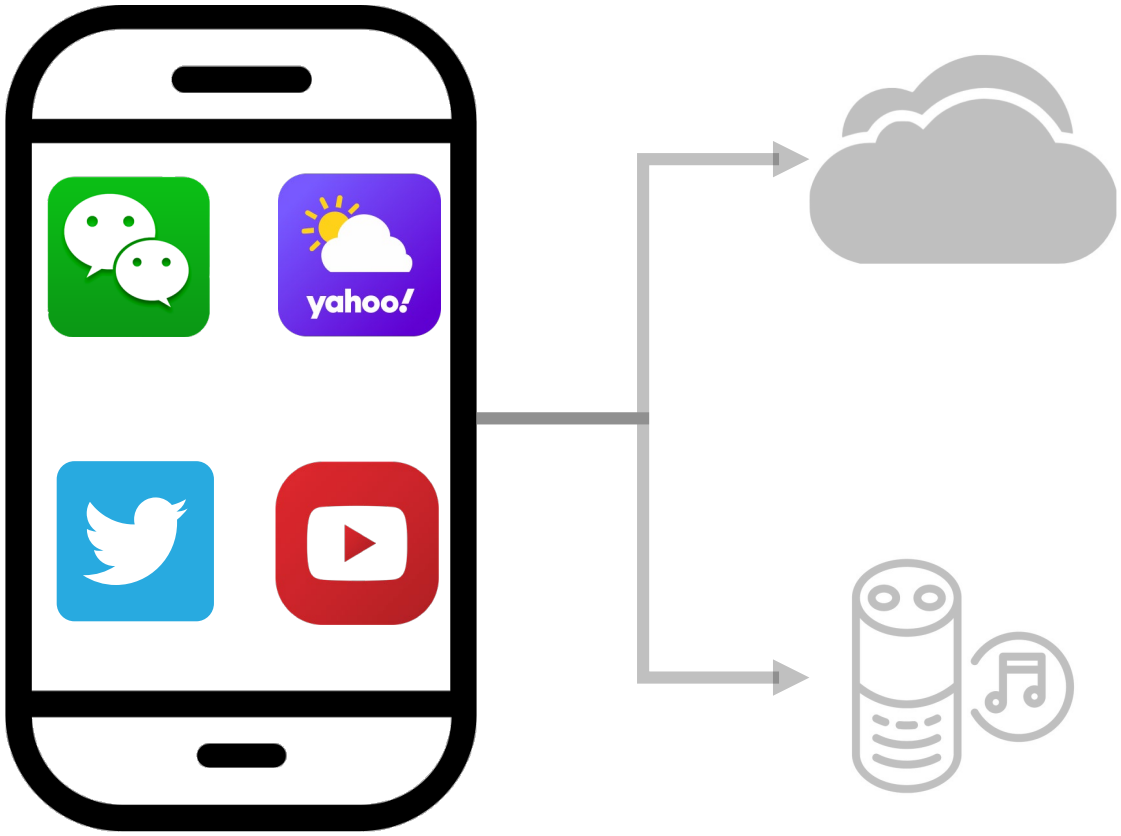
Well Protected?



Secure?



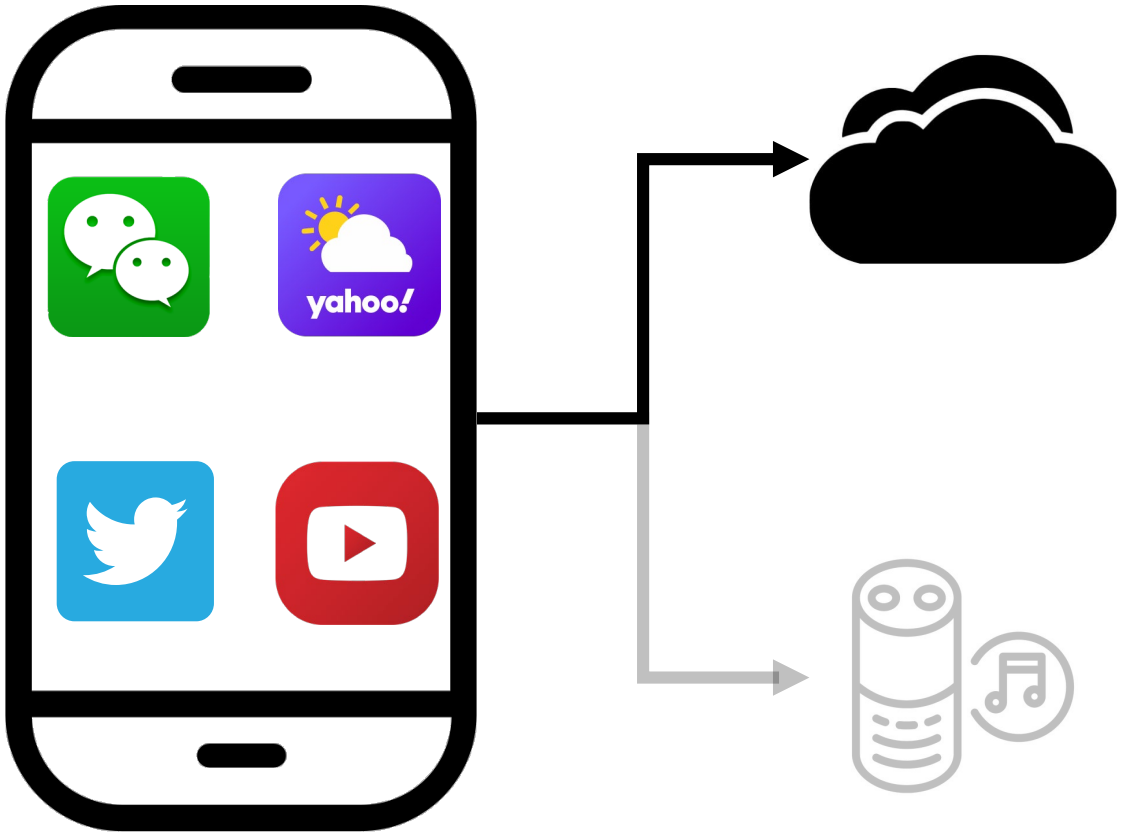
Vulnerable



This powerful Android malware stayed hidden for years, infecting tens of thousands of smartphones

Mandrake spyware hoovers up information raging from account credentials, screen records, GPS and more -- and has been for years. All while those behind it carefully cover their tracks.

Mobile Apps could be **Malware**



This powerful Android malware stayed hidden for years, infecting tens of thousands of smartphones

Mandrak
been for

Help Net Security
June 3, 2020

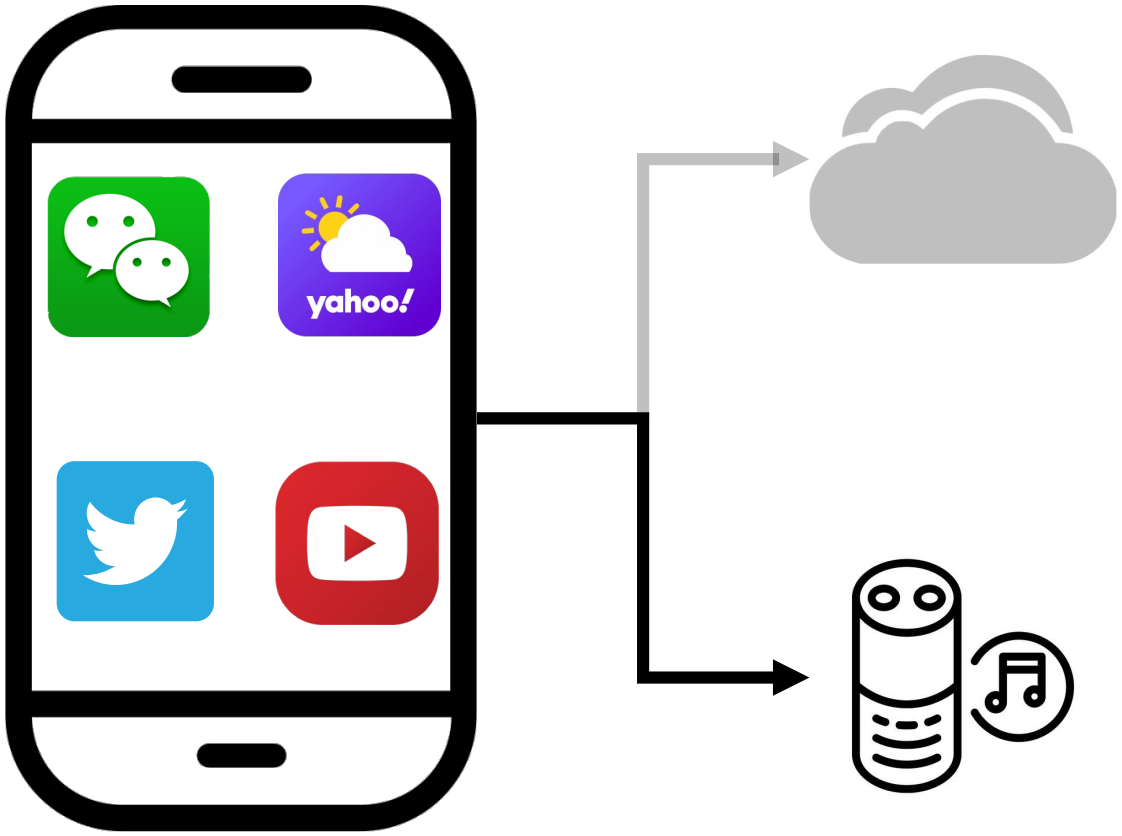
Share

and has

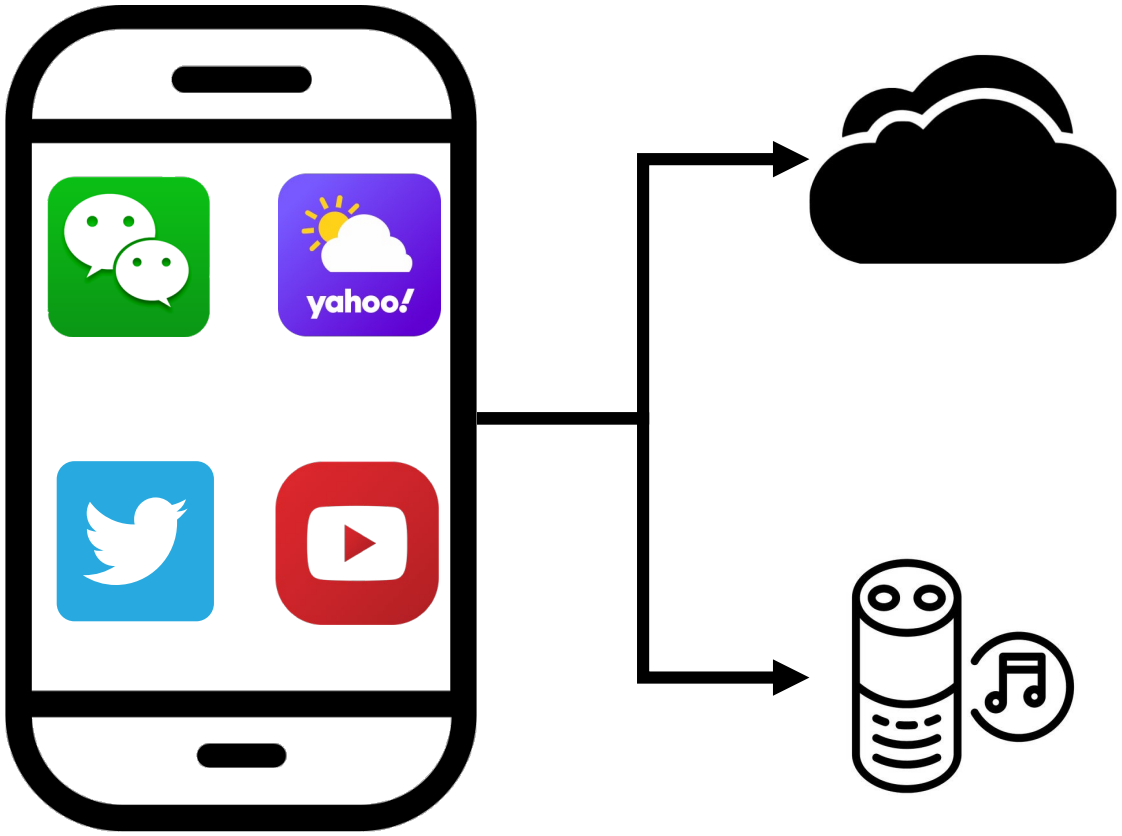
Most companies suffered a cloud data breach in the past 18 months

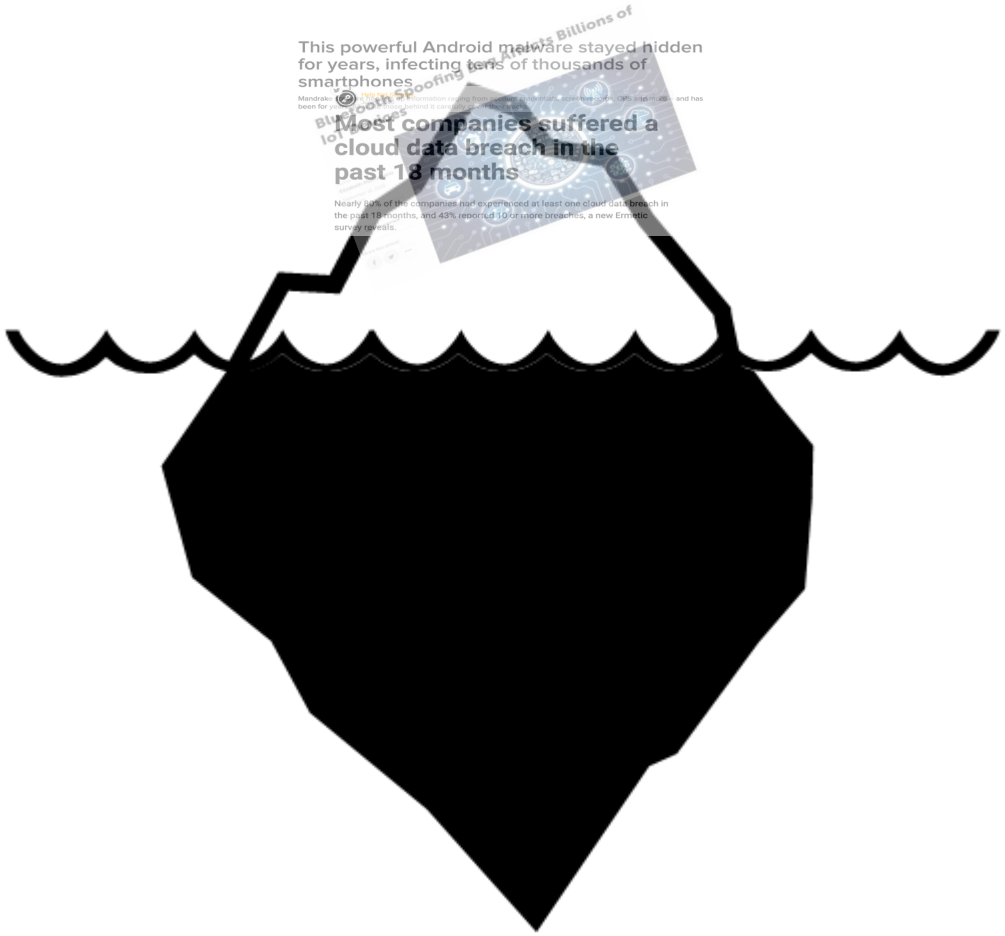
Nearly 80% of the companies had experienced at least one cloud data breach in the past 18 months, and 43% reported 10 or more breaches, a new Ermetic survey reveals.

Clouds could be **Breached**

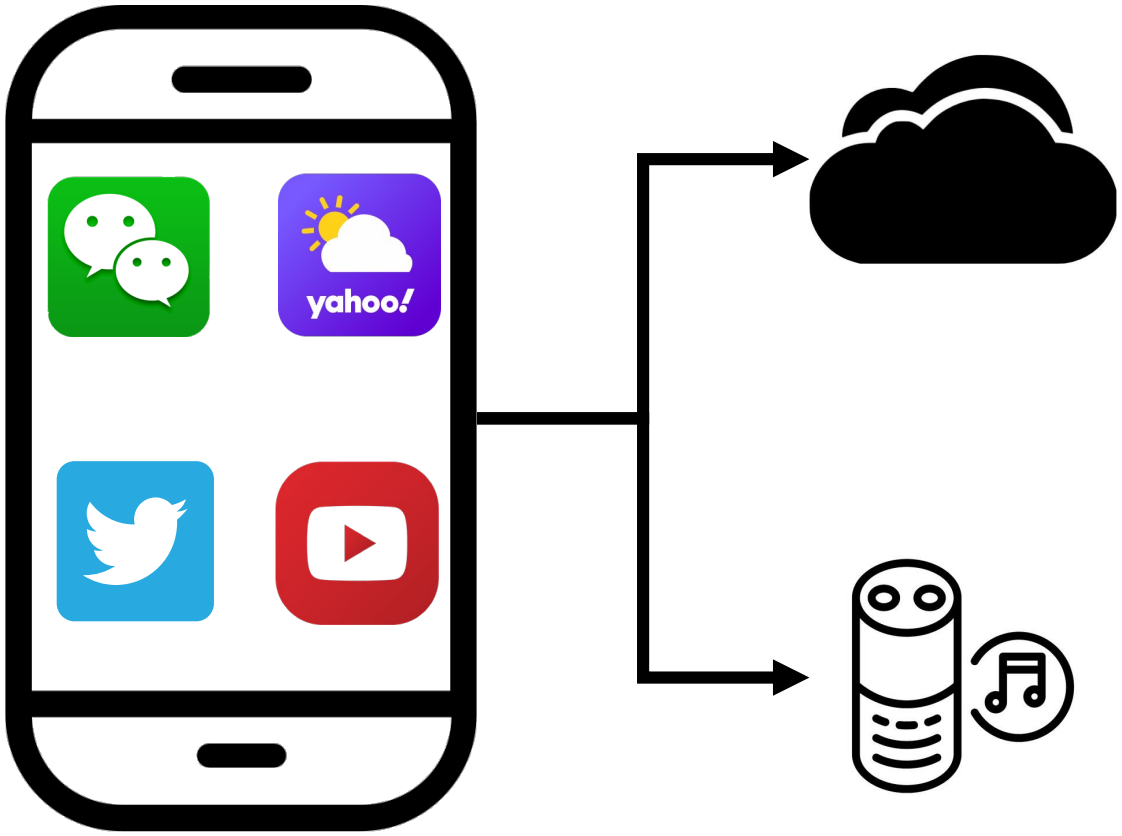


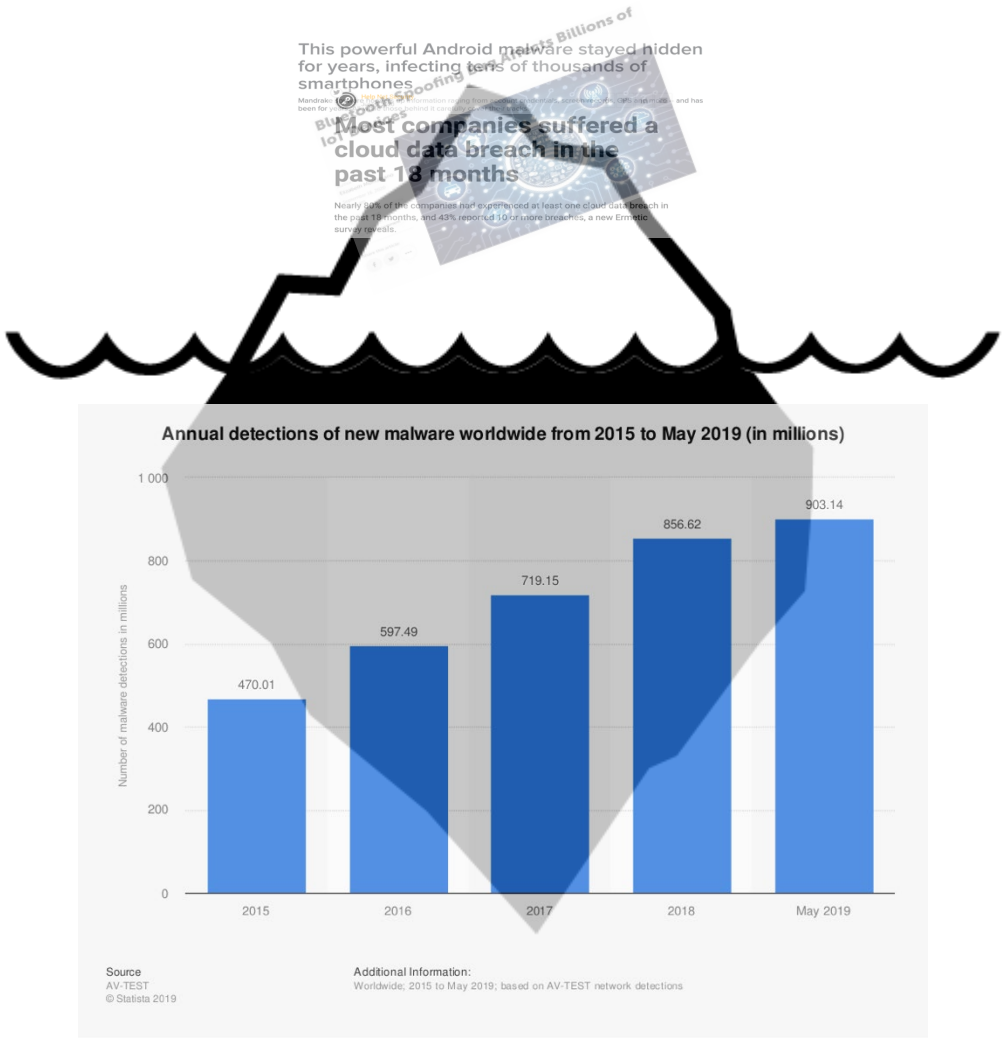
Peripherals could be **Insecure**



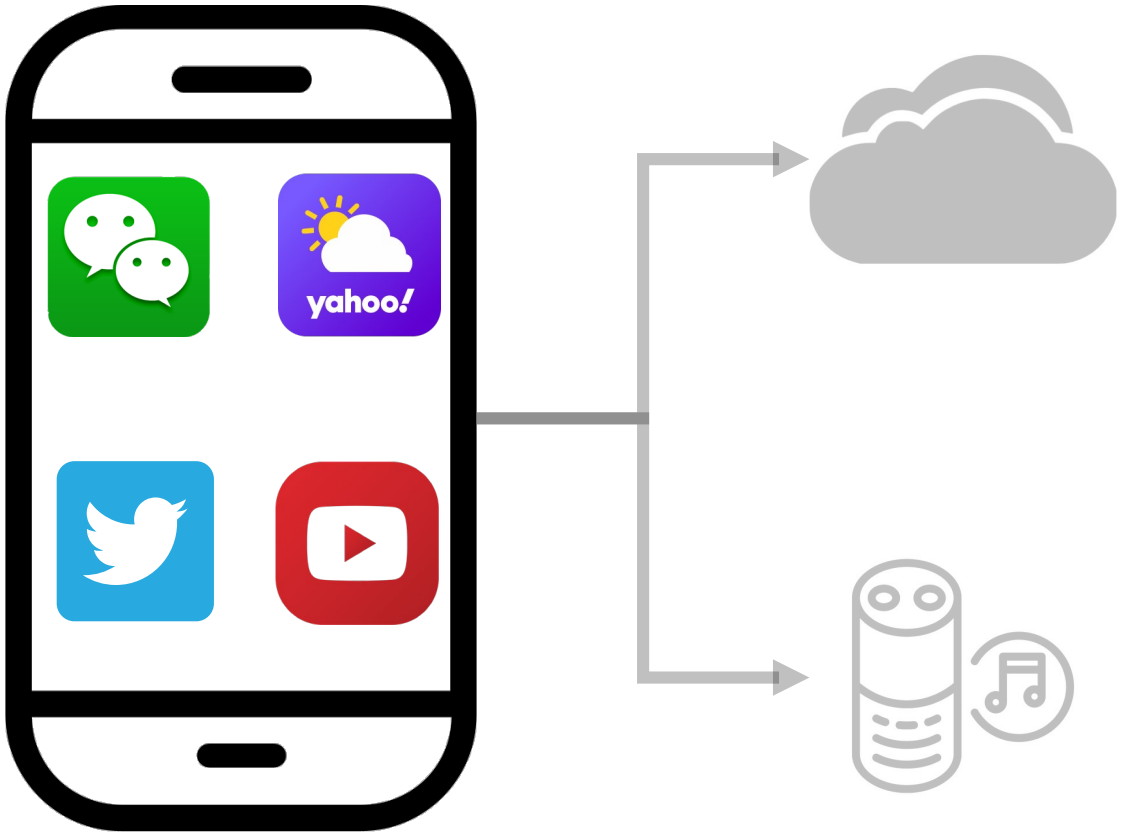


A ***Tip*** of The ***Iceberg***





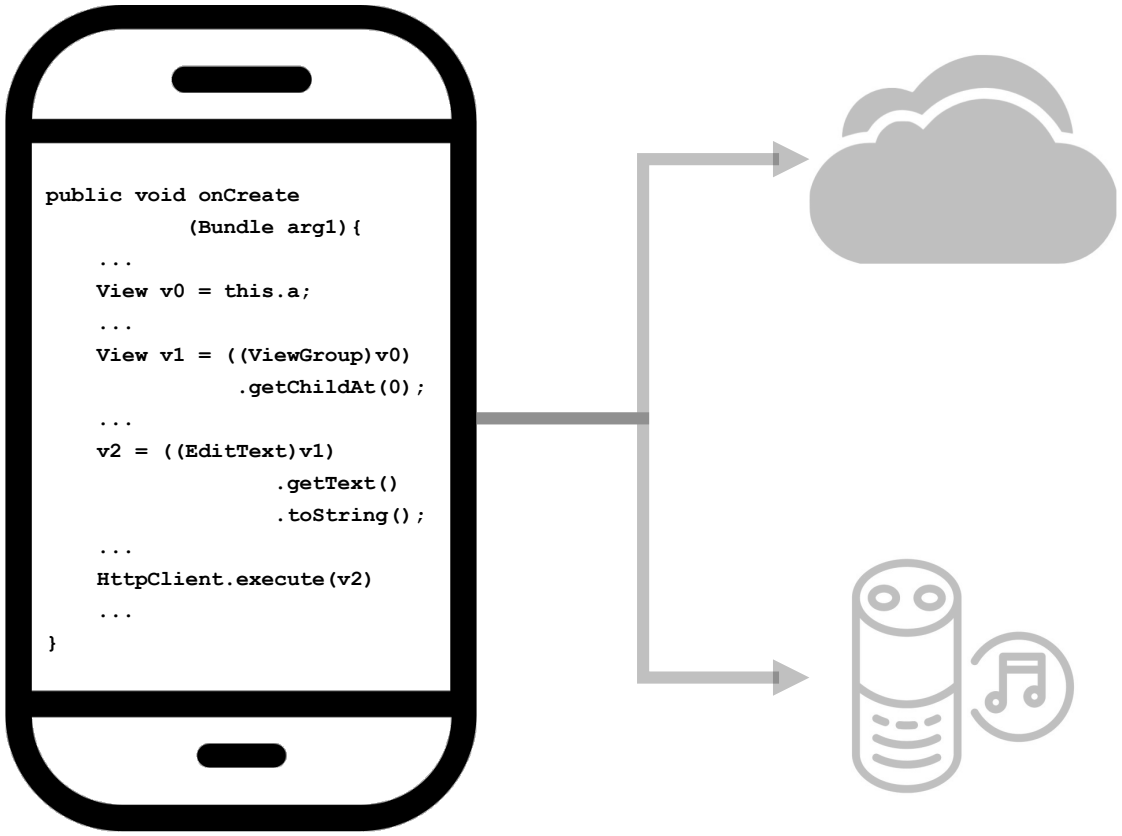
Severe Security and Privacy Concerns



Primary Target

- Mobile Apps
- Android

Our Research



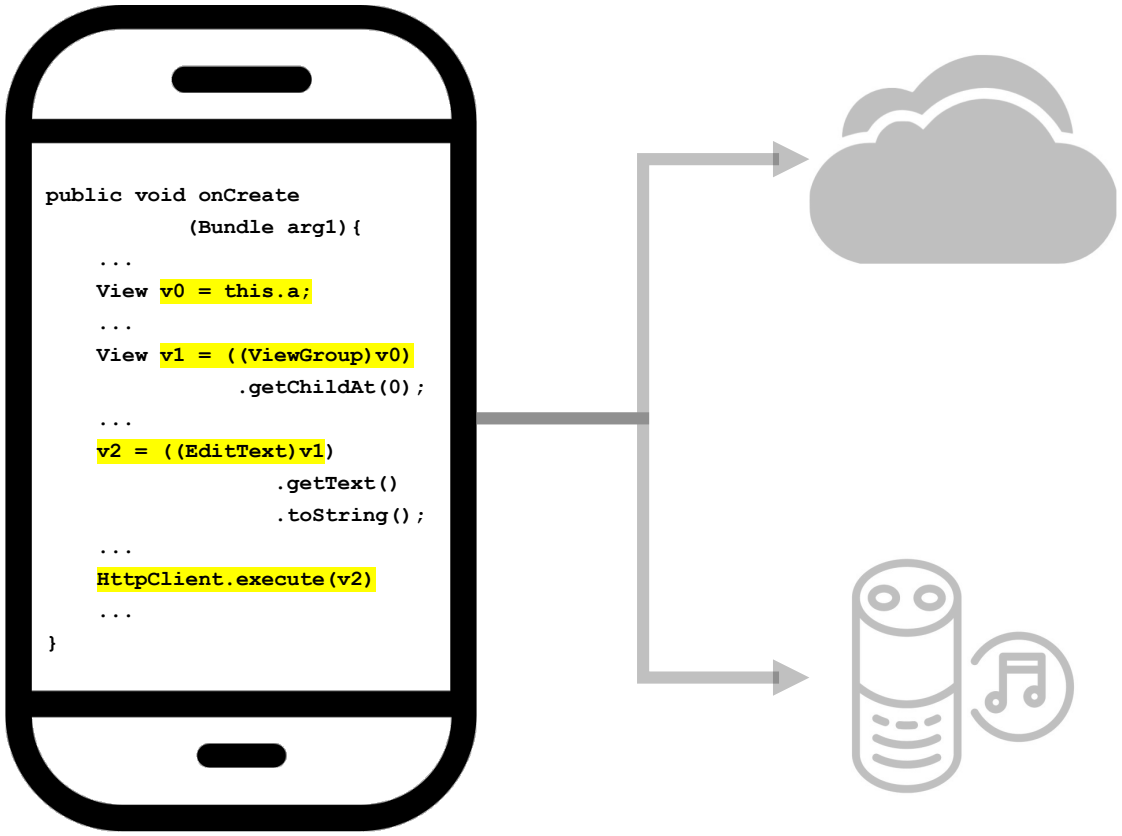
Primary Target

- Mobile Apps
- Android

Techniques

- Reverse Engineering

Our Research



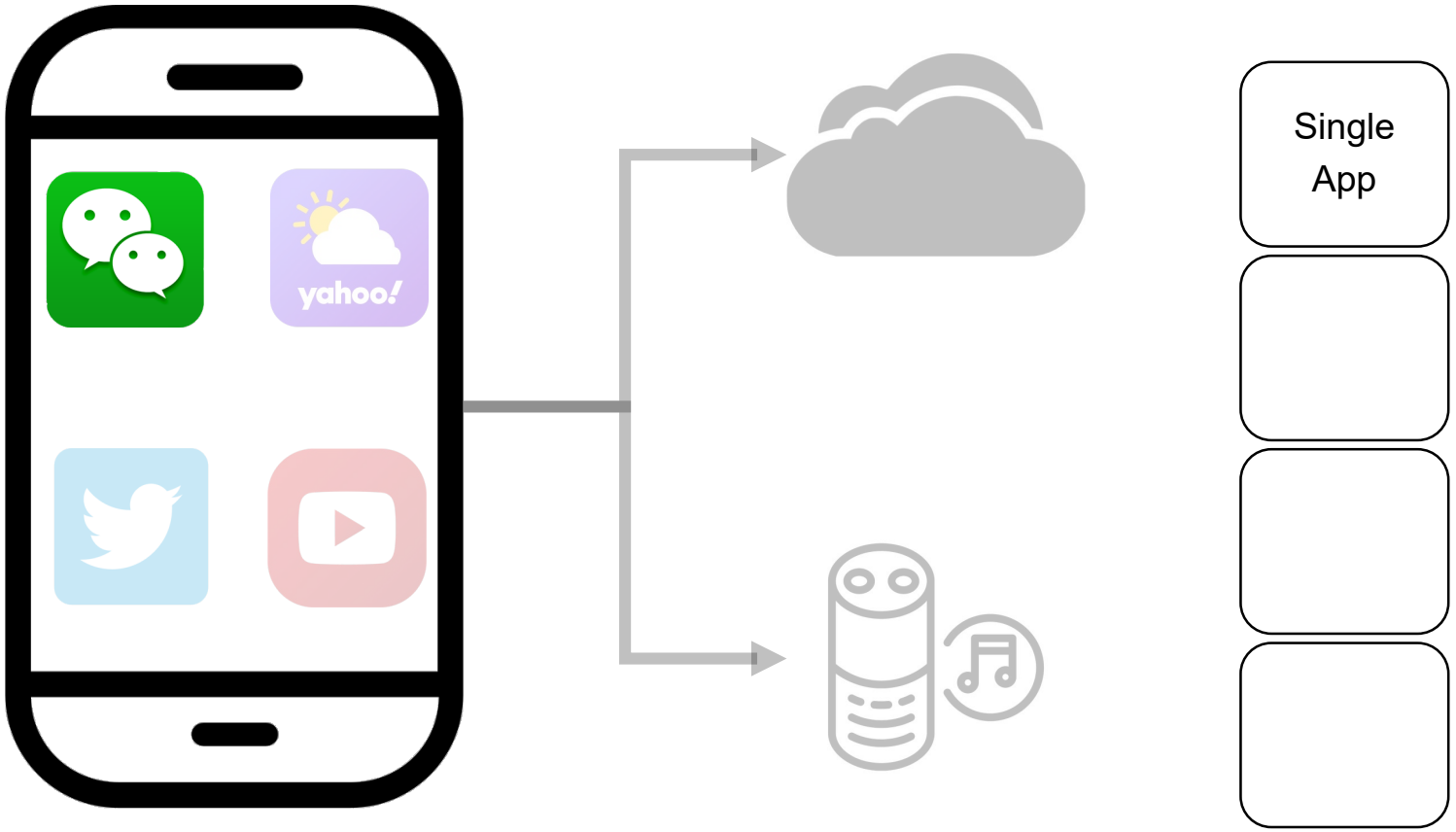
Primary Target

- Mobile Apps
- Android

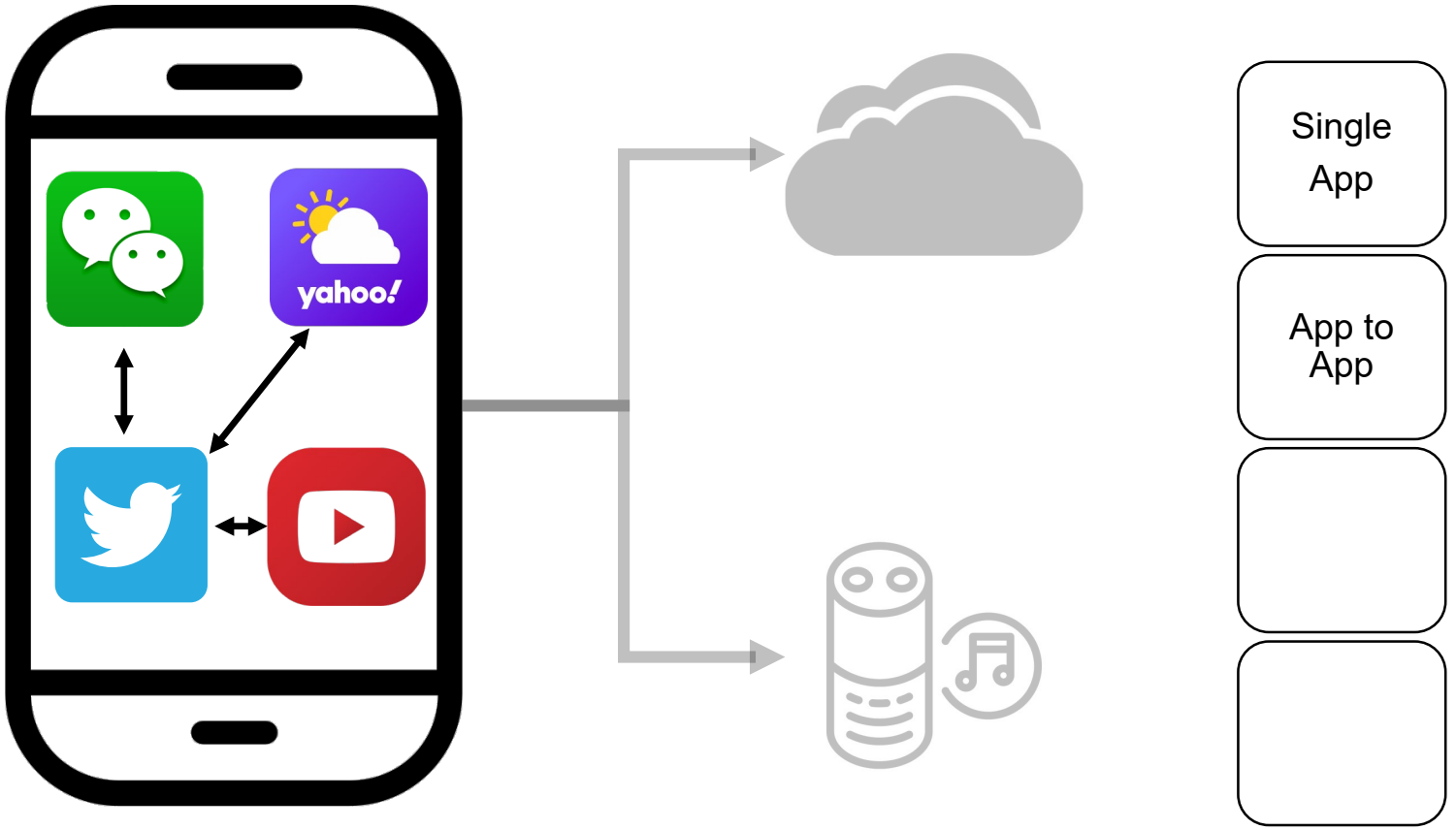
Techniques

- Reverse Engineering
- Automated Program Analysis

Our Research



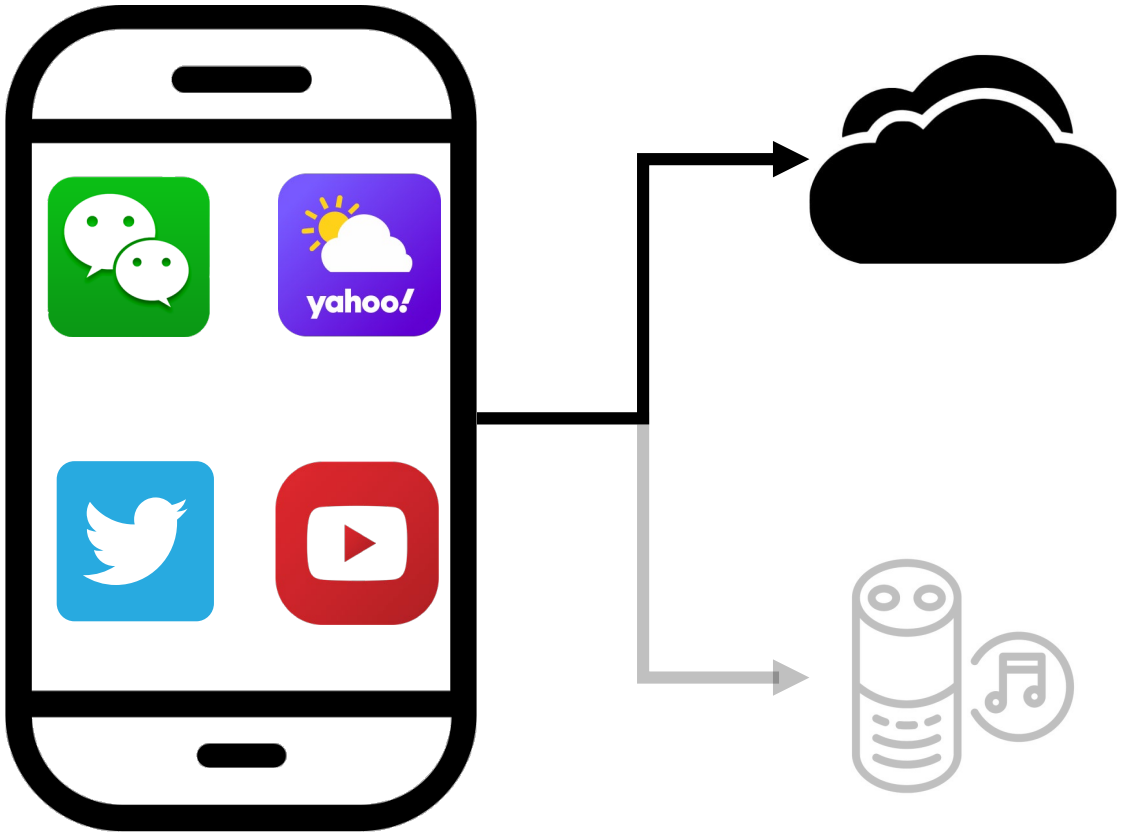
Our Research



Single App

App to App

Our Research

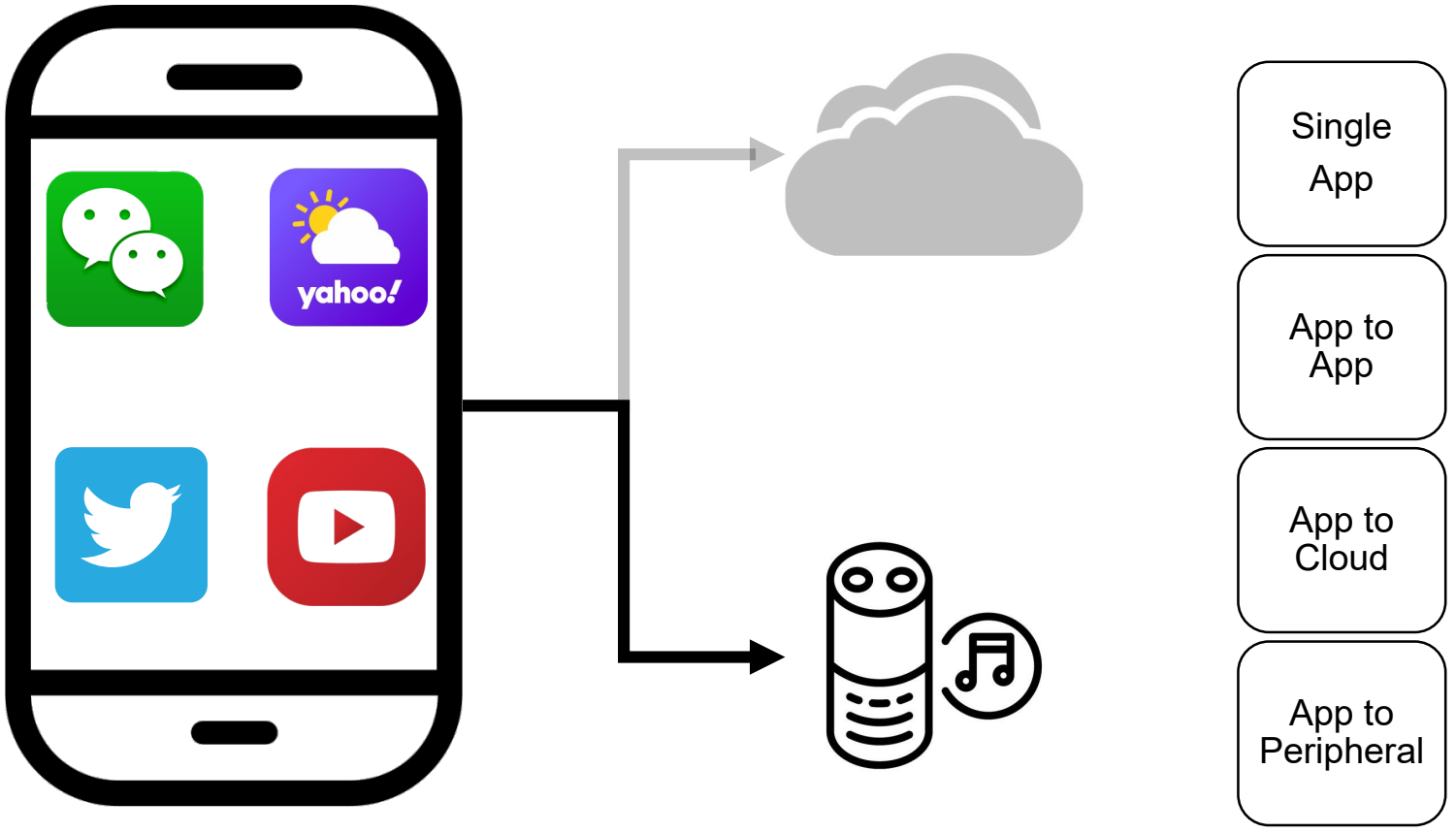


Single App

App to App

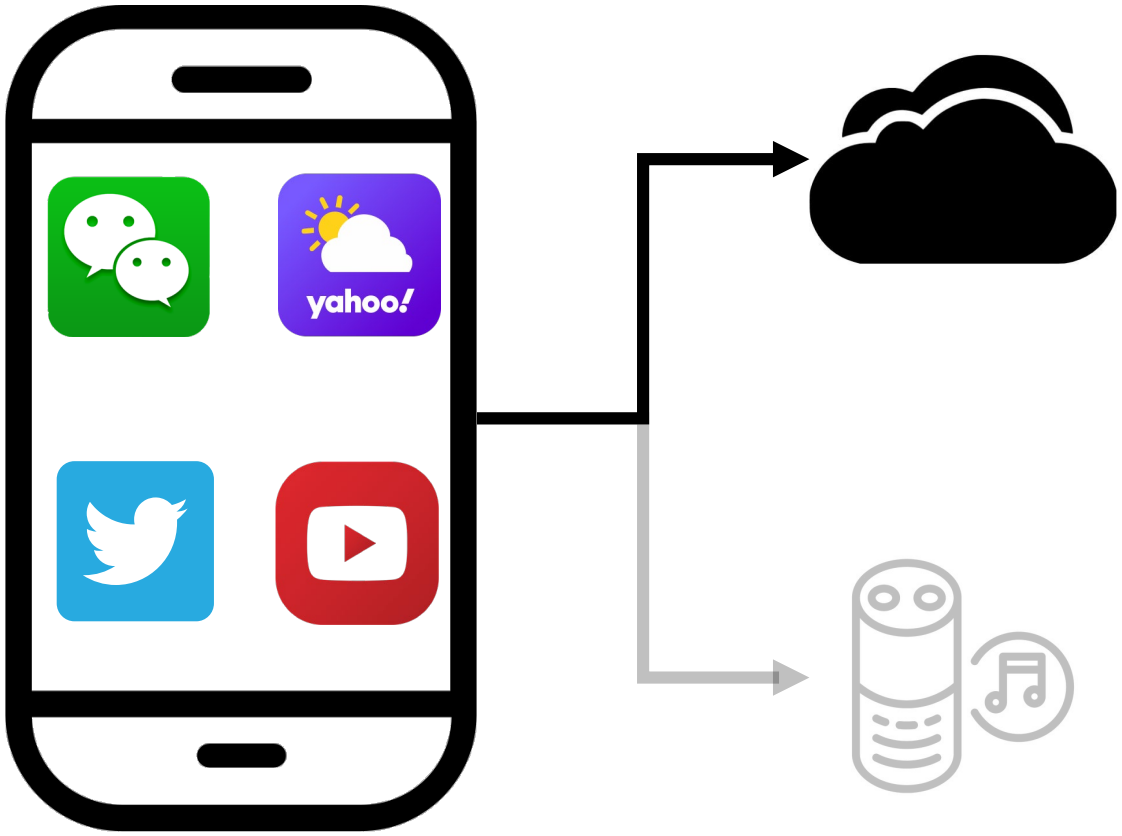
App to Cloud

Our Research



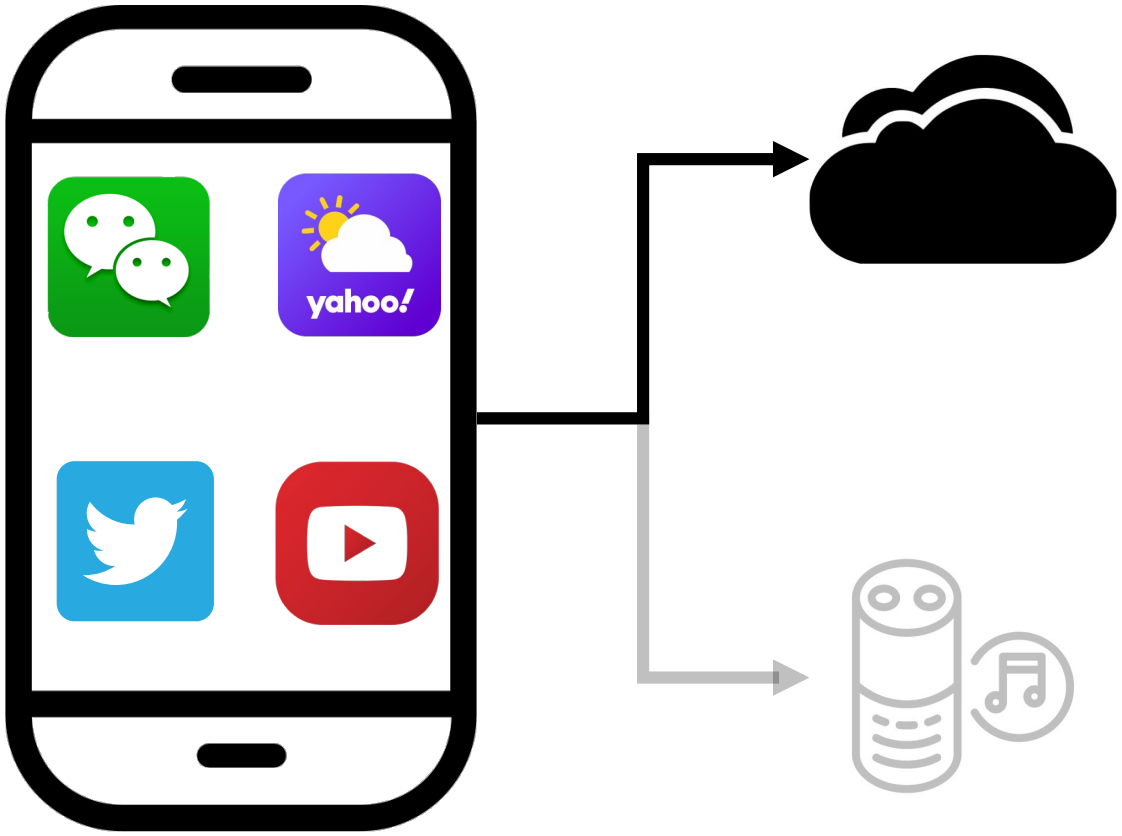
- Single App
- App to App
- App to Cloud
- App to Peripheral

Our Research



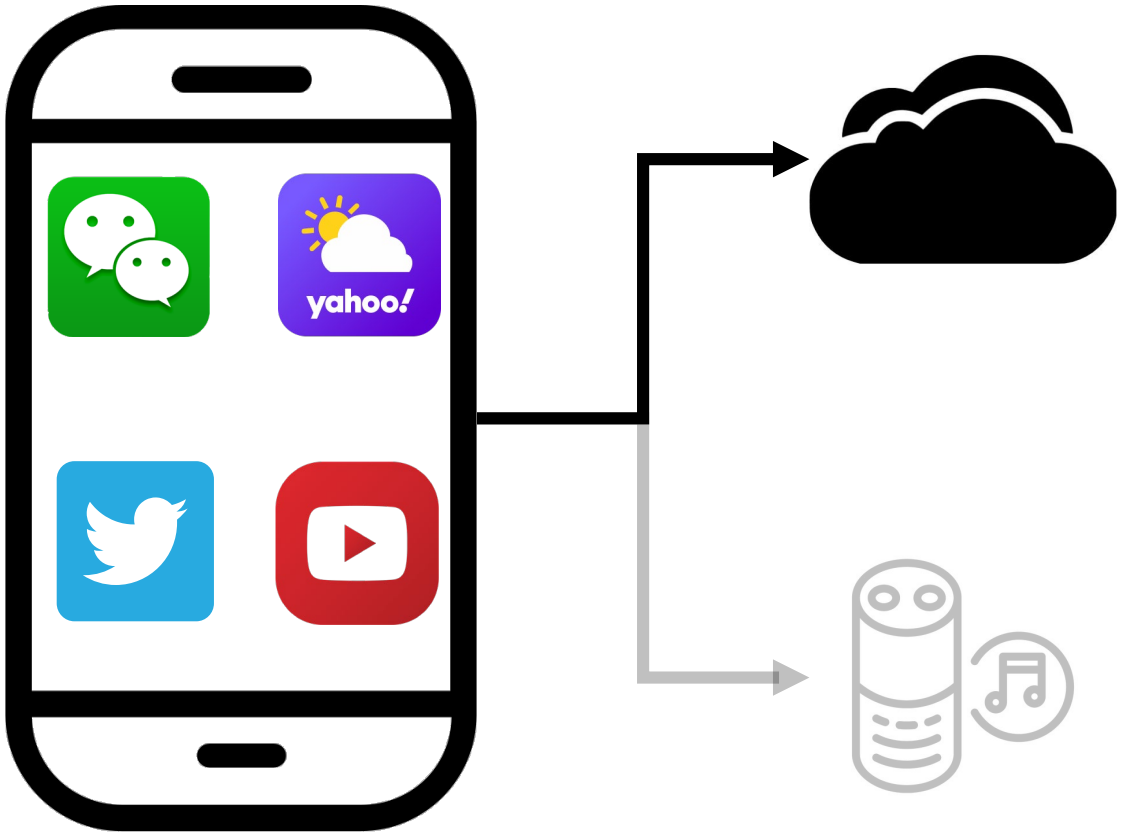
	Attack	Analysis	Detection
Single App	• NDSS'14		
App to App			
App to Cloud	• NDSS'19 • NDSS'16		
App to Peripheral	• USENIX Security'20		

1. "Geo-locating Drivers: A Study of Sensitive Data Leakagein Ride-Hailing Services". Qingchuan Zhao, Chaoshun Zuo, Giancarlo Pellegrino, and Zhiqiang Lin. In Proceedings of the 26th ISOC Network and Distributed System Security Symposium, San Diego, CA, February 2019.
2. "Automatic Forgery of Cryptographically Consistent Messages to Identify Security Vulnerabilities in Mobile Services". Chaoshun Zuo, Wubing Wang, Rui Wang, and Zhiqiang Lin. in *Proceedings of the 23rd ISOC Network and Distributed System Security Symposium*, San Diego, CA, February 2016
3. "Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks", Yue Zhang, Jian Weng, Rajib Dey, Yier Jin, Zhiqiang Lin, and Xinwen Fu. In *Proceedings of the 29th USENIX Security Symposium*, Boston, MA. August 2020.
4. "SMV-Hunter: Large Scale, Automated Detection of SSL/TLS Man-in-the-Middle Vulnerabilities in Android Apps". David Sounthiraraj, Justin Sahs, Garrett Greenwood, Zhiqiang Lin, and Latifur Khan. In *Proceedings of the 21st ISOC Network and Distributed System Security Symposium*, San Diego, CA, February 2014



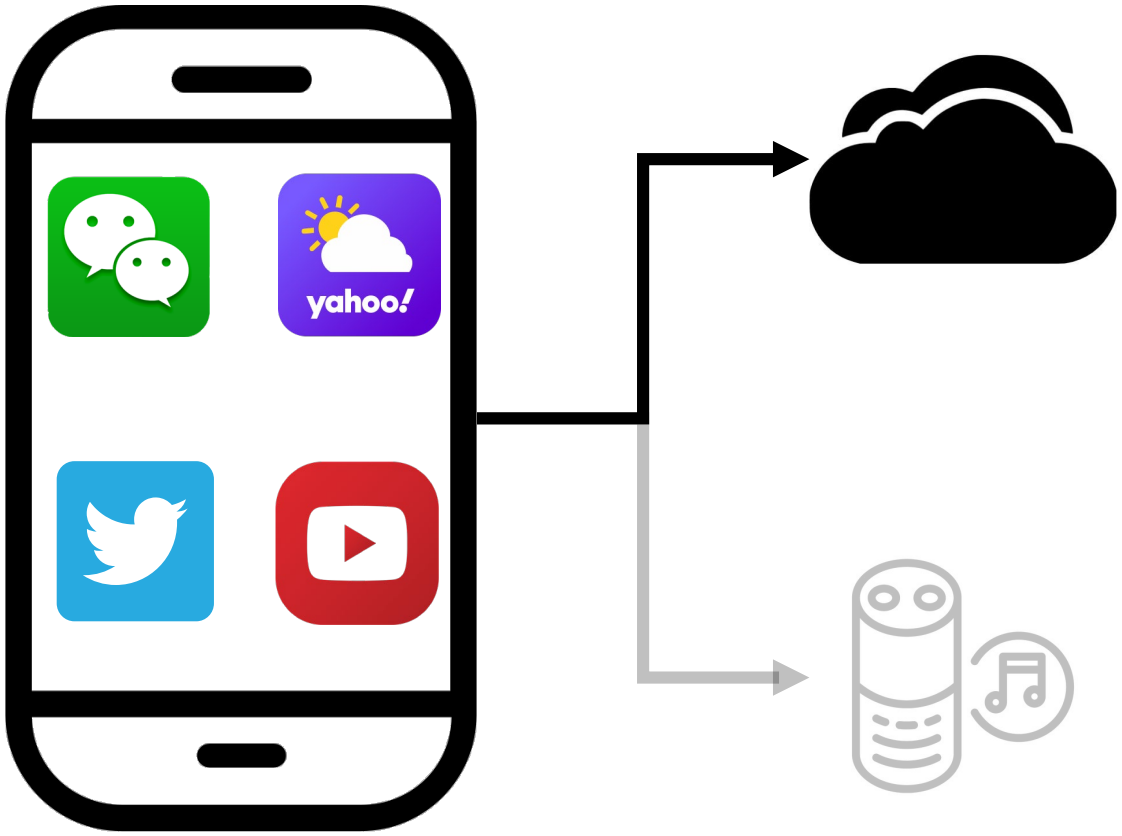
	Attack	Analysis	Detection
Single App	• NDSS'14		
App to App			
App to Cloud	• NDSS'19 • NDSS'16	• SP'19 • USENIX Security'19	
App to Peripheral	• USENIX Security'20		

1. "The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends", Omar Alrawi*, Chaoshun Zuo*, Ruian Duan, Ranjita Kasturi, Zhiqiang Lin, Brendan Saltaformaggio. (*authors contributed equally) In *Proceedings of the 2019 USENIX Security Symposium*, Santa Clara. August 2019.
2. "Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps". Chaoshun Zuo, Zhiqiang Lin, and Yinqian Zhang. In *Proceedings of the 40th IEEE Symposium on Security and Privacy*, San Fransisco, CA, May 2019.



	Attack	Analysis	Detection
Single App	• NDSS'14		• SP'20
App to App			• USENIX Security'20
App to Cloud	• NDSS'19 • NDSS'16	• SP'19 • USENIX Security'19	• CCS'17
App to Peripheral	• USENIX Security'20		• NDSS'20 • NDSS'18

1. "Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps". Qingchuan Zhao, Chaoshun Zuo, Dolan-Gavitt Brendan, Giancarlo Pellegrino, and Zhiqiang Lin. In Proceedings of the 41st IEEE Symposium on Security and Privacy, San Francisco, CA, May 2020.
2. "FirmScope: Automatic Uncovering of Privilege-Escalation Vulnerabilities in Pre-Installed Apps in Android Firmware". Mohamed Elsabagh, Ryan Johnson, Angelos Stavrou, Chaoshun Zuo, Qingchuan Zhao, and Zhiqiang Lin. In Proceedings of the 29th USENIX Security Symposium, Boston, MA. August 2020.
3. "Automated Cross-Platform Reverse Engineering of CAN Bus Commands From Mobile Apps". Haohuang Wen, Qingchuan Zhao, Qi Alfred Chen, and Zhiqiang Lin. In Proceedings of the 27th ISOC Network and Distributed System Security Symposium, San Diego, CA, February 2020.
4. "IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing". Jiongyi Chen, Wenrui Diao, Qingchuan Zhao, Chaoshun Zuo, Zhiqiang Lin, XiaoFeng Wang, Wing Cheong Lau, Menghan Sun, Ronghai Yang, Kehuan Zhang. In Proceedings of the 25th ISOC Network and Distributed System Security Symposium, San Diego, CA, February 2018.
5. "AuthScope: Towards Automatic Discovery of Vulnerable Authorizations in Online Services". Chaoshun Zuo, Qingchuan Zhao, and Zhiqiang Lin. In Proceedings of the 24th ACM Conference Computer and Communications Security, Dallas, Texas. November 2017.

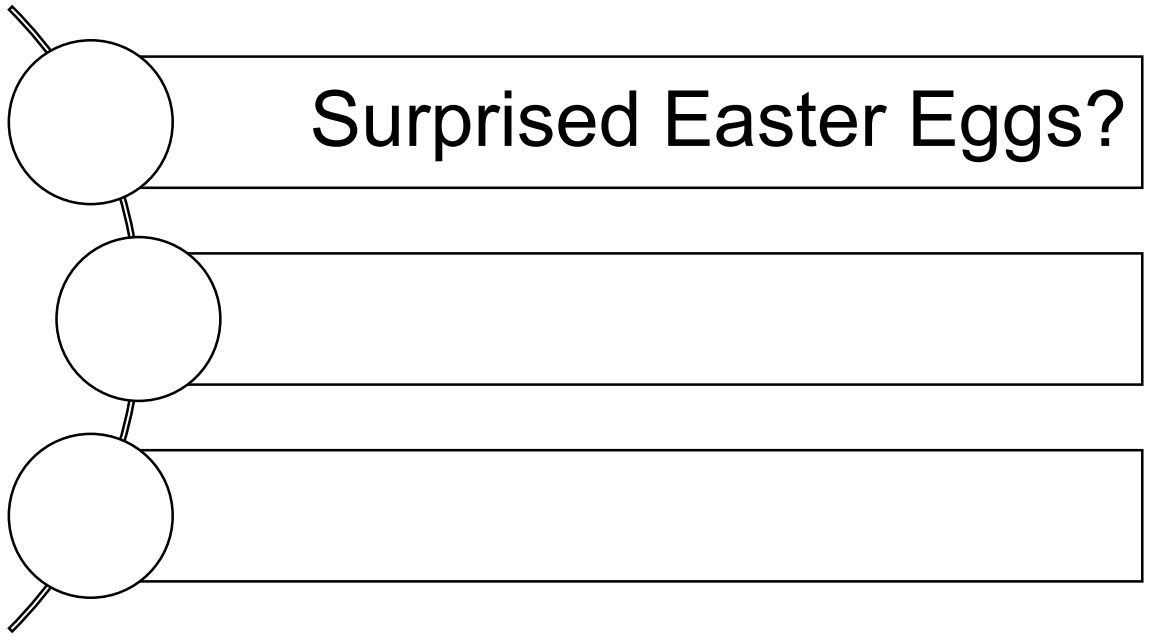


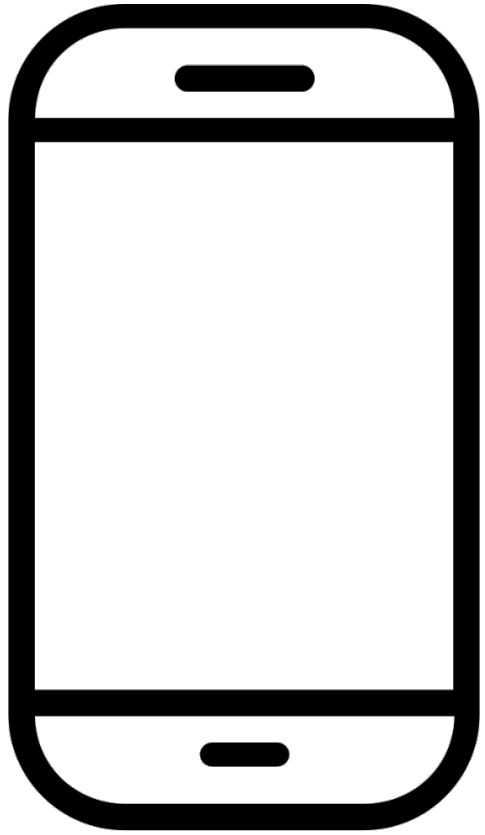
	Attack	Analysis	Detection
Single App	• NDSS'14		• **SP'20**
App to App			• USENIX Security'20
App to Cloud	• **NDSS'19** • NDSS'16	• SP'19 • USENIX Security'19	• CCS'17
App to Peripheral	• USENIX Security'20		• NDSS'20 • NDSS'18

1. "Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps". Qingchuan Zhao, Chaoshun Zuo, Dolan-Gavitt Brendan, Giancarlo Pellegrino, and Zhiqiang Lin. In Proceedings of the 41st IEEE Symposium on Security and Privacy, San Francisco, CA, May 2020.
2. "Geo-locating Drivers: A Study of Sensitive Data Leakage in Ride-Hailing Services". Qingchuan Zhao, Chaoshun Zuo, Giancarlo Pellegrino, and Zhiqiang Lin. In Proceedings of the 26th ISOC Network and Distributed System Security Symposium, San Diego, CA, February 2019.

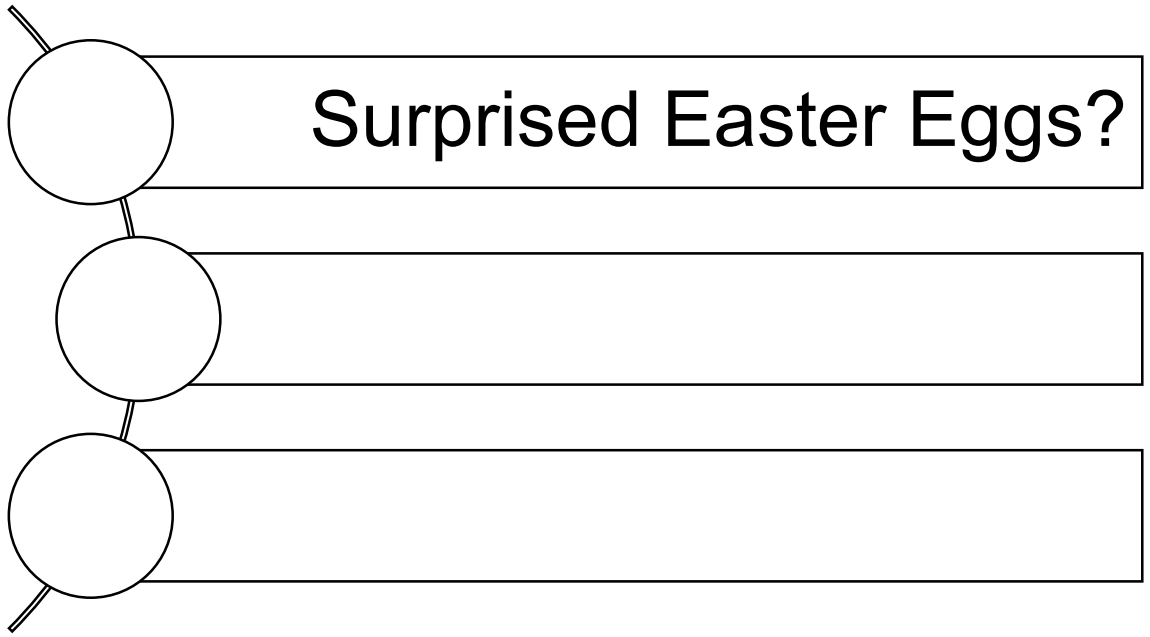
Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps

In Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P), 2020



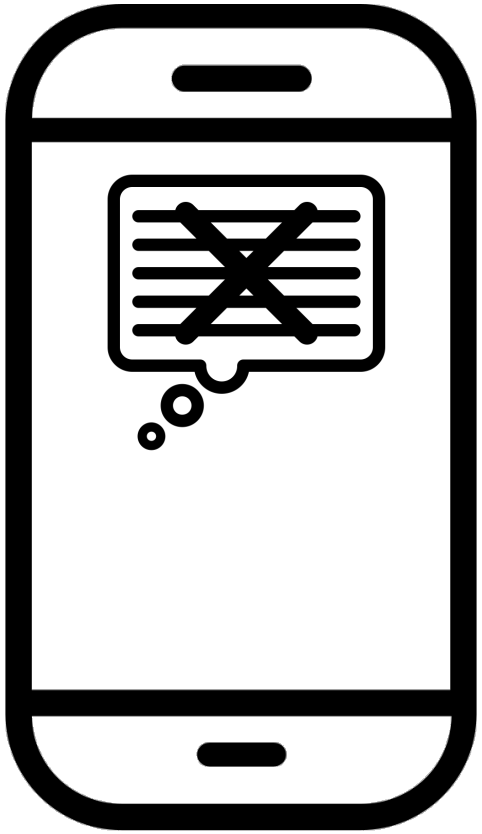


Hypothetical Questions?



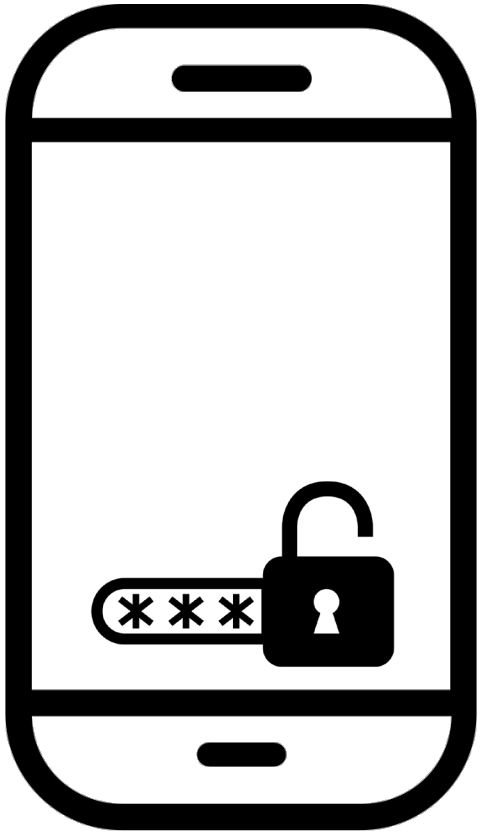
Hypothetical Questions?

- Surprised Easter Eggs?
- Disallowed Expressions?
-



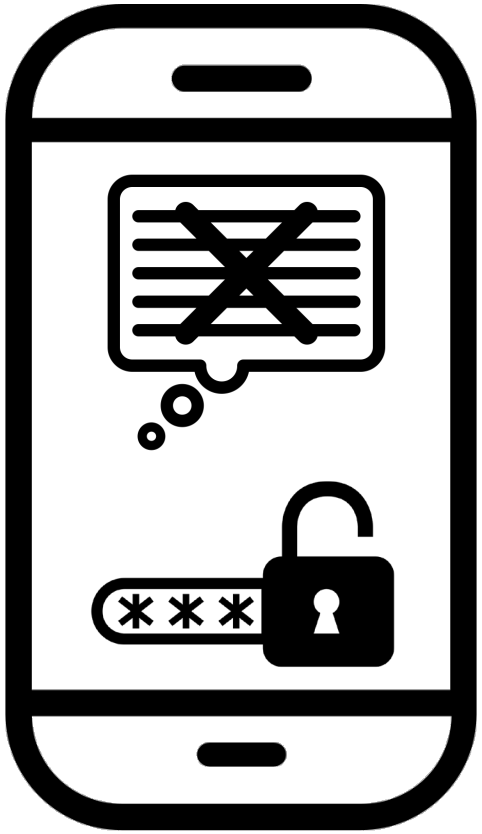
Hypothetical Questions?

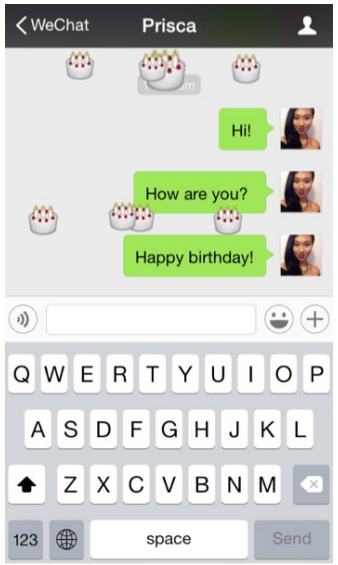
- Surprised Easter Eggs?
- Disallowed Expressions?
- Backdoors?



Hypothetical Questions?

- Surprised Easter Eggs?
- Disallowed Expressions?
- Backdoors?





Hidden Behaviors

iFeng News

```
 1 private void validate_nickname(String arg3, Dialog arg4) {
 2     if(!TextUtils.isEmpty(((CharSequence)arg3))) {
 3         String v0 = this.a.getText().toString();
 4         if(StringUtil.isInterceptedNickName(this.e, v0)) {
 5             String v1 = "Nickname contains illegal
                           characters!";
 6             ann.a(this.e).a(v4, v1);
 7         } else ...
```

```
 8 public static boolean isInterceptedNickName
                                (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Filtering Unwanted Content With A Blacklist

iFeng News

- Reading user's nickname

```
 1 private void validate_nickname(String arg3, Dialog arg4) {
 2     if(!TextUtils.isEmpty(((CharSequence)arg3))) {
 3         String v0 = this.a.getText().toString();
 4         if(StringUtil.isInterceptedNickName(this.e, v0)) {
 5             String v1 = "Nickname contains illegal
                           characters!";
 6             ann.a(this.e).a(v4, v1);
 7         } else ...
```

```
 8 public static boolean isInterceptedNickName
                                (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Filtering Unwanted Content With A Blacklist

iFeng News

- Reading user's nickname
- Comparing nickname with a list of words
- Checking for equivalence

```
 1 private void validate_nickname(String arg3, Dialog arg4) {
 2     if(!TextUtils.isEmpty(((CharSequence)arg3))) {
 3         String v0 = this.a.getText().toString();
 4         if(StringUtil.isInterceptedNickName(this.e, v0)) {
 5             String v1 = "Nickname contains illegal
                           characters!";
 6             ann.a(this.e).a(v4, v1);
 7         } else ...
```

```
 8 public static boolean isInterceptedNickName
                                (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Filtering Unwanted Content With A Blacklist

iFeng News

- Reading user's nickname
- Comparing nickname with a list of words
- Checking for equivalence
- Forbidden words

```
 1 private void validate_nickname(String arg3, Dialog arg4) {
 2     if(!TextUtils.isEmpty(((CharSequence)arg3))) {
 3         String v0 = this.a.getText().toString();
 4         if(StringUtil.isInterceptedNickName(this.e, v0)) {
 5             String v1 = "Nickname contains illegal
                           characters!";
 6             ann.a(this.e).a(v4, v1);
 7         } else ...
```

```
 8 public static boolean isInterceptedNickName
                                (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

File Location: */assets/intercepted_word.txt*

Filtering Unwanted Content With A Blacklist

iFeng News

- Reading user's nickname
- Comparing nickname with a list of words
- Checking for equivalence
- Forbidden words
- Nickname is not allowed to use

```
 1 private void validate_nickname(String arg3, Dialog arg4) {
 2     if(!TextUtils.isEmpty(((CharSequence)arg3))) {
 3         String v0 = this.a.getText().toString();
 4         if(StringUtil.isInterceptedNickName(this.e, v0)) {
 5             String v1 = "Nickname contains illegal
                           characters!";
 6             ann.a(this.e).a(v4, v1);
 7         } else ...
```

```
 8 public static boolean isInterceptedNickName
                                (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

File Location: */assets/intercepted_word.txt*

Filtering Unwanted Content With A Blacklist

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Master Password

- Reading user input

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Master Password

- Reading user input
- Comparing user input twice
- Checking equivalence

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Master Password

- Reading user input
- Comparing user input twice
- Checking equivalence
- Hardcoded string
- Value from database

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Master Password

- Reading user input
- Comparing user input twice
- Checking equivalence
- Hardcoded string
- Value from database
- Unlock files

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Master Password

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                    .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                    (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                  (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence
- Pre-defined value

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                  (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                  (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

- Different input checking

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                  (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty((CharSequence)v1)) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

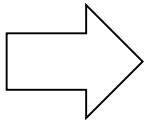
- Different input checking
- Difference comparison content

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
 8 public static boolean isInterceptedNickName
                                  (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty((CharSequence)v1)) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

Triggering

- Checking equivalence
- Pre-defined value



Different Behaviors

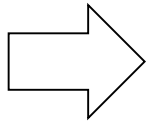
- Different input checking
- Difference comparison content

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

- Different input checking
- Difference comparison content



Triggering

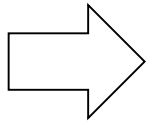
- Checking equivalence

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

- Different input checking
- Difference comparison content



Triggering

- Checking equivalence

Comparison Context

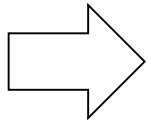
- Source of comparison content

Triggering

- Checking equivalence
- Pre-defined value

Different Behaviors

- Different input checking
- Difference comparison content

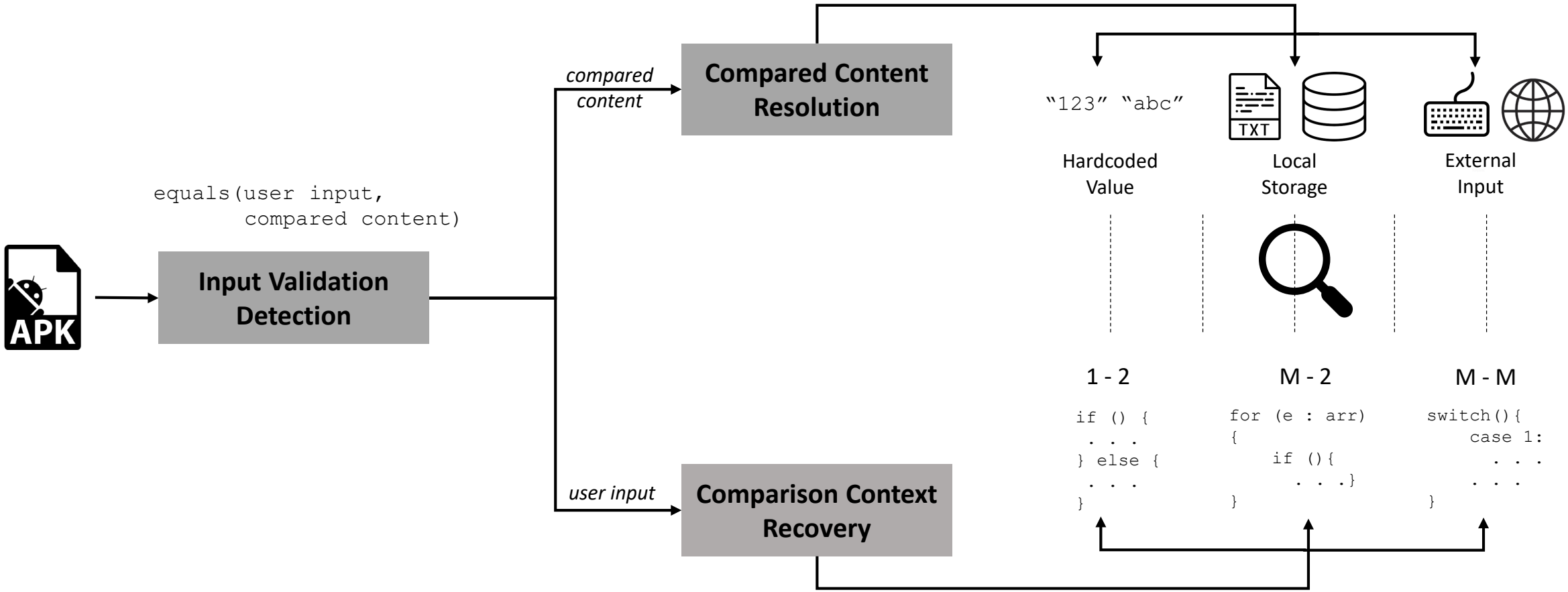


Triggering

- Checking equivalence

Comparison Context

- Source of comparison content
- Code dispatch for input checking


equals(user input,
compared content)
APK
Input Validation Detection
compared content
user input
Compared Content Resolution
Comparison Context Recovery
"123" "abc"
TXT
Hardcoded Value
Local Storage
External Input
1 - 2
M - 2
M - M
if () {
. . .
} else {
. . .
}
for (e : arr)
{
if (){
. . .}
}
switch(){
case 1:
. . .
. . .
}

```
equals(user input,
       compared content)
```

```
equals(user input,
       compared content)
```


```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
equals(user input,
       compared content)
```


```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

```
equals(user input,
       compared content)
```


```
1  public void onClick(DialogInterface arg7, int arg8) {
2      String v2 = "";
3      View v0 = this.a;
4      int v3 = 0;
5      while(v3 < ((ViewGroup)v0).getChildCount()) {
6          View v1 = ((ViewGroup)v0).getChildAt(v3);
7          if(v1 != null && v1.getId() == 2131624072)
8              v2 = ((EditText)v1).getText().toString();
9          ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                    .show();
15     }
16 }
```

Type	System APIs
Taint Sources	EditText.getText()
	EditText.getEditableText()
	Editable.toString()

```
equals(user input,
       compared content)
```

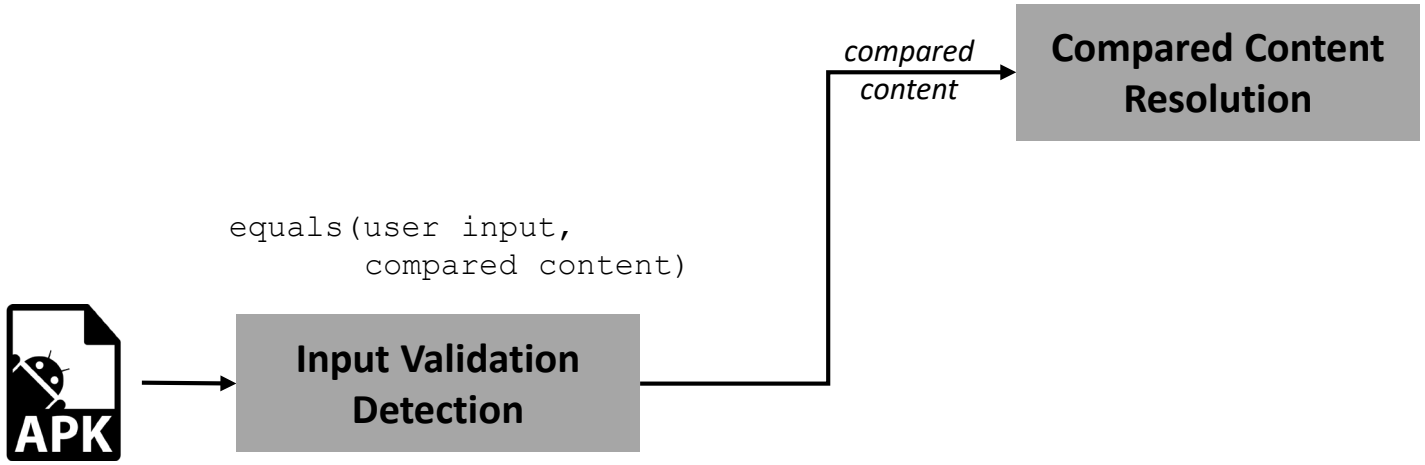

```
1  public void onClick(DialogInterface arg7, int arg8) {
2      String v2 = "";
3      View v0 = this.a;
4      int v3 = 0;
5      while(v3 < ((ViewGroup)v0).getChildCount()) {
6          View v1 = ((ViewGroup)v0).getChildAt(v3);
7          if(v1 != null && v1.getId() == 2131624072)
8              v2 = ((EditText)v1).getText().toString();
9          ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                    .show();
15     }
16 }
```

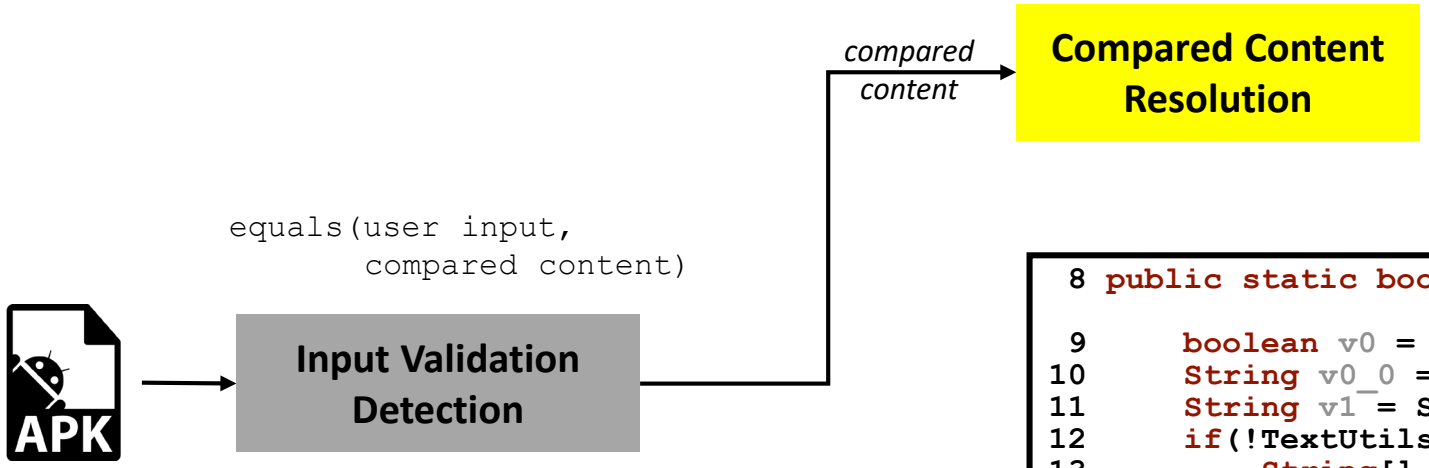
Type	System APIs
Taint Sources	EditText.getText()
	EditText.getEditableText()
	Editable.toString()
Taint Sinks	Object.equals()
	String.equals()
	TextUtils.equals()

```
equals(user input,
       compared content)
```

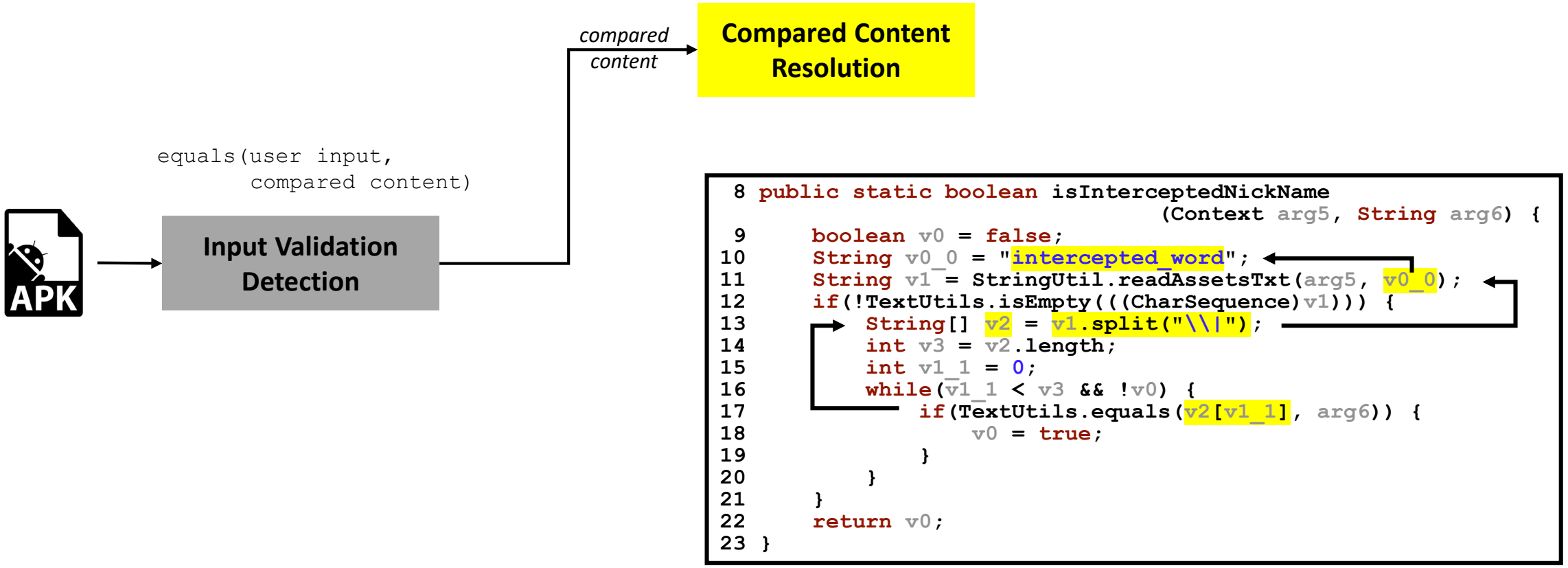

```
1  public void onClick(DialogInterface arg7, int arg8) {
2      String v2 = "";
3      View v0 = this.a;
4      int v3 = 0;
5      while(v3 < ((ViewGroup)v0).getChildCount()) {
6          View v1 = ((ViewGroup)v0).getChildAt(v3);
7          if(v1 != null && v1.getId() == 2131624072)
8              v2 = ((EditText)v1).getText().toString();
9          ++v3;
10     }
11     if(v2.equals(this.b) |
           v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Type	System APIs
Taint Sources	EditText.getText()
	EditText.getEditableText()
	Editable.toString()
Taint Sinks	Object.equals()
	String.equals()
	TextUtils.equals()
	StringBuffer.indexOf()
	HashMap.containsKey()
	Map.get()

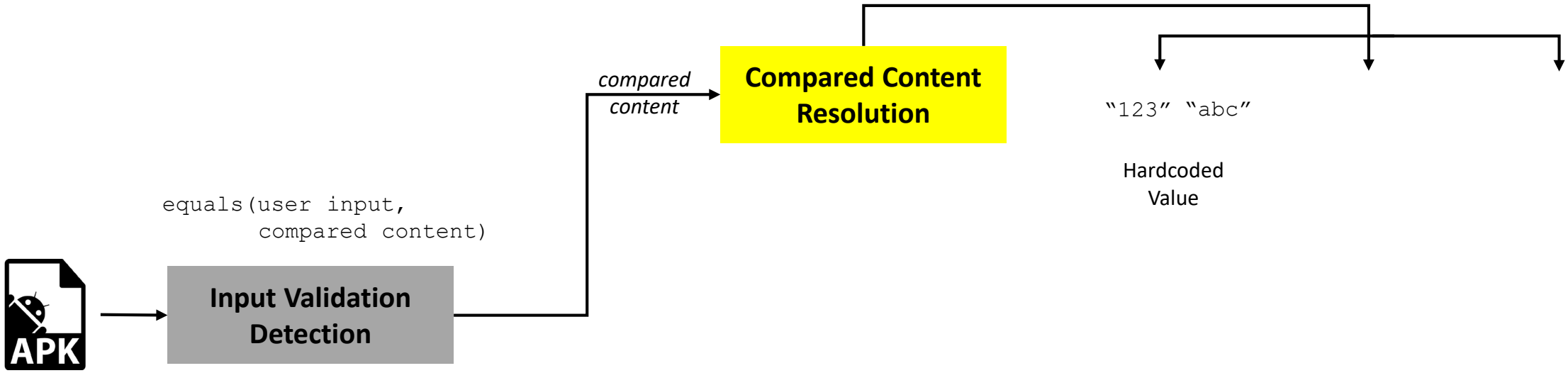

compared
content
Compared Content Resolution
equals(user input,
compared content)
APK
Input Validation Detection

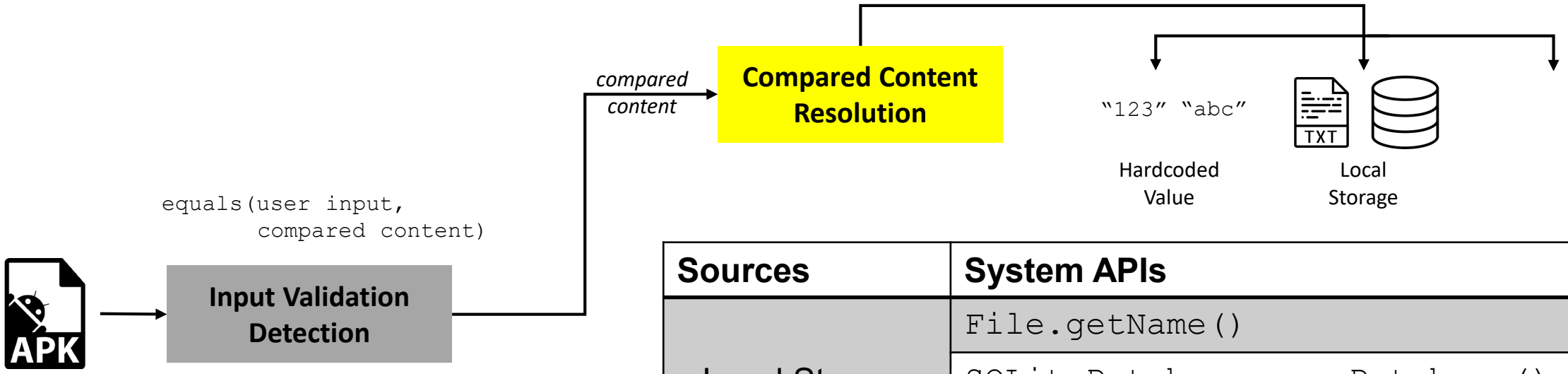



```
 8 public static boolean isInterceptedNickName
                                    (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

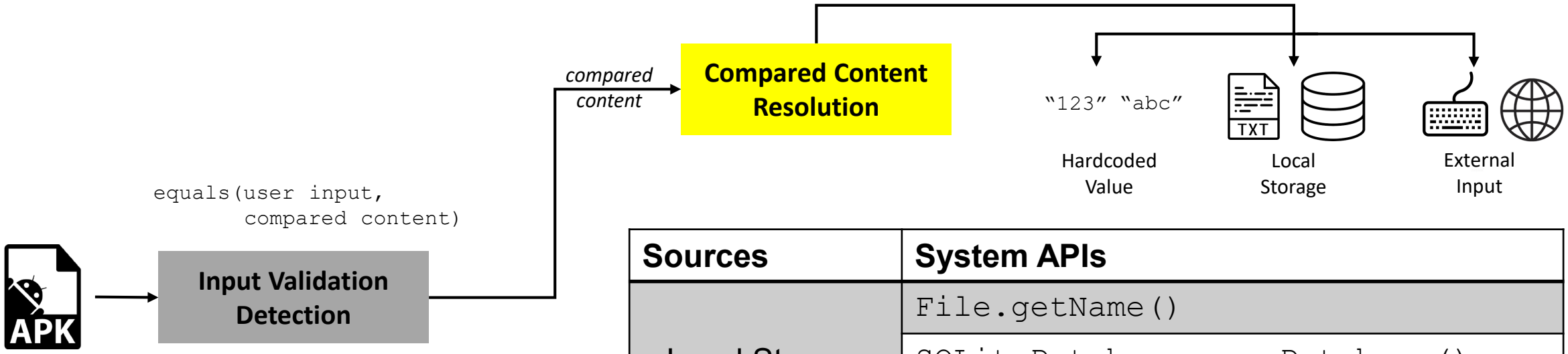



```
 8 public static boolean isInterceptedNickName
                                    (Context arg5, String arg6) {
 9     boolean v0 = false;
10     String v0_0 = "intercepted_word";
11     String v1 = StringUtil.readAssetsTxt(arg5, v0_0);
12     if(!TextUtils.isEmpty(((CharSequence)v1))) {
13         String[] v2 = v1.split("\\|");
14         int v3 = v2.length;
15         int v1_1 = 0;
16         while(v1_1 < v3 && !v0) {
17             if(TextUtils.equals(v2[v1_1], arg6)) {
18                 v0 = true;
19             }
20         }
21     }
22     return v0;
23 }
```

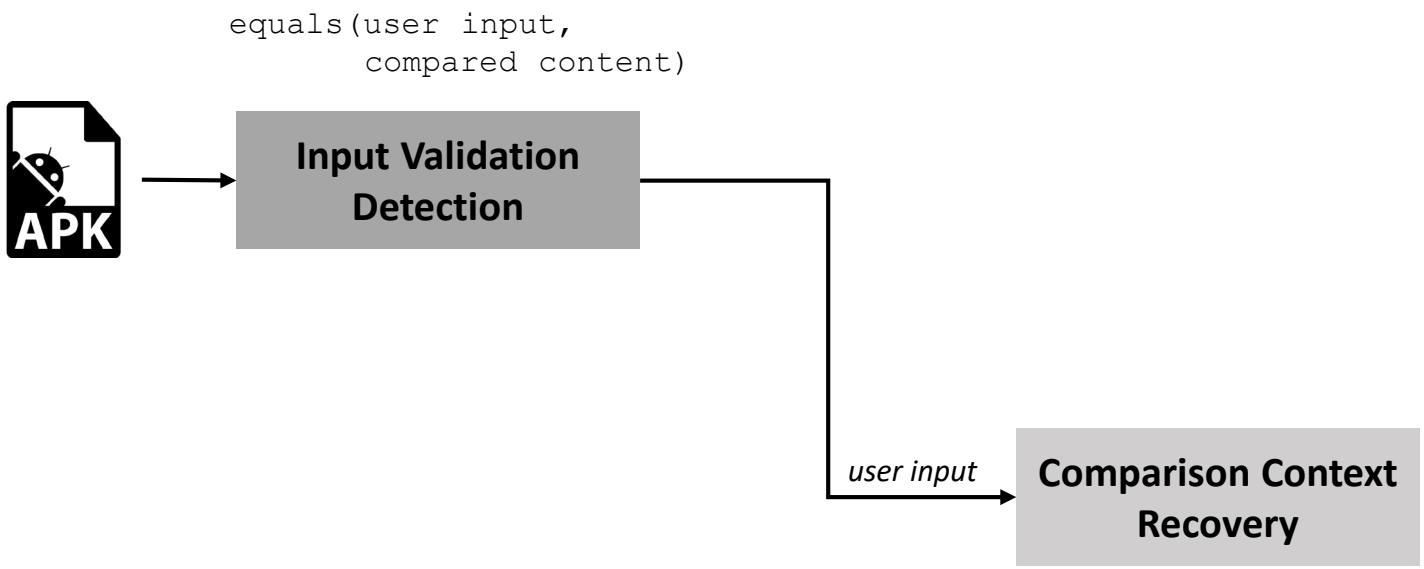

compared content
Compared Content Resolution
"123" "abc"
Hardcoded Value
equals(user input, compared content)
APK
Input Validation Detection



Sources	System APIs
Local Storage	`File.getName()`
	`SQLiteDatabase.openDatabase()`
	`SharedPreferences.getString()`



Sources	System APIs
Local Storage	`File.getName()`
	`SQLiteDatabase.openDatabase()`
	`SharedPreferences.getString()`
External Values	`Intent.getStringExtra()`
	`EditText.getText()`
	`Socket.getInputStream()`


equals(user input,
compared content)
APK
Input Validation Detection
user input
Comparison Context Recovery

```
equals(user input,
       compared content)
```

user input

Comparison Context Recovery

```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                  .show();
15     }
16 }
```

Dispatch	*Times of Validations*	Number of Branches

```
equals(user input,
       compared content)
```

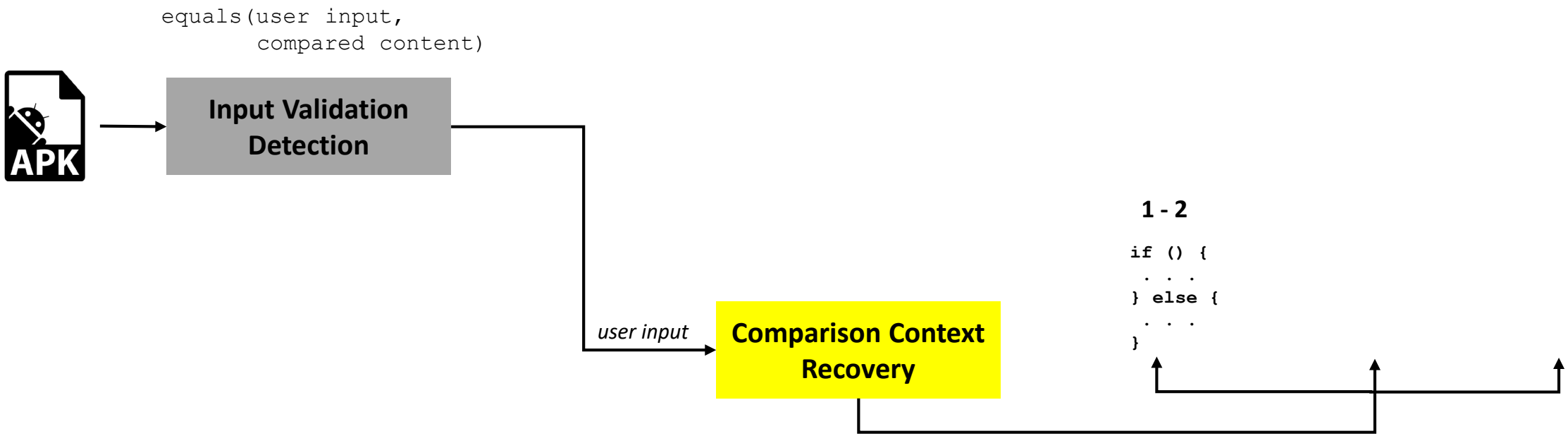
user input

Comparison Context Recovery

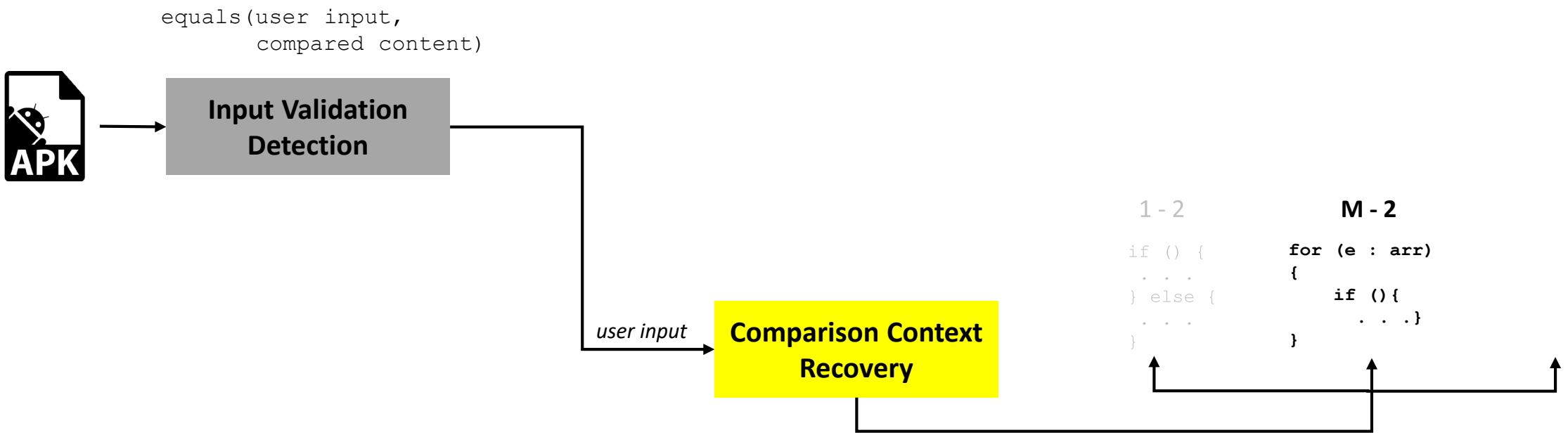
```
 1 public void onClick(DialogInterface arg7, int arg8) {
 2     String v2 = "";
 3     View v0 = this.a;
 4     int v3 = 0;
 5     while(v3 < ((ViewGroup)v0).getChildCount()) {
 6         View v1 = ((ViewGroup)v0).getChildAt(v3);
 7         if(v1 != null && v1.getId() == 2131624072)
 8             v2 = ((EditText)v1).getText().toString();
 9         ++v3;
10     }
11     if(v2.equals(this.b) |
          v2.equals("bypassPassword9977251")) {
12         ... // viewing files
13     } else {
14         Toast.makeText(this,"Incorrect password", 1)
                                                .show();
15     }
16 }
```

Dispatch	Times of Validations	***Number of Branches***

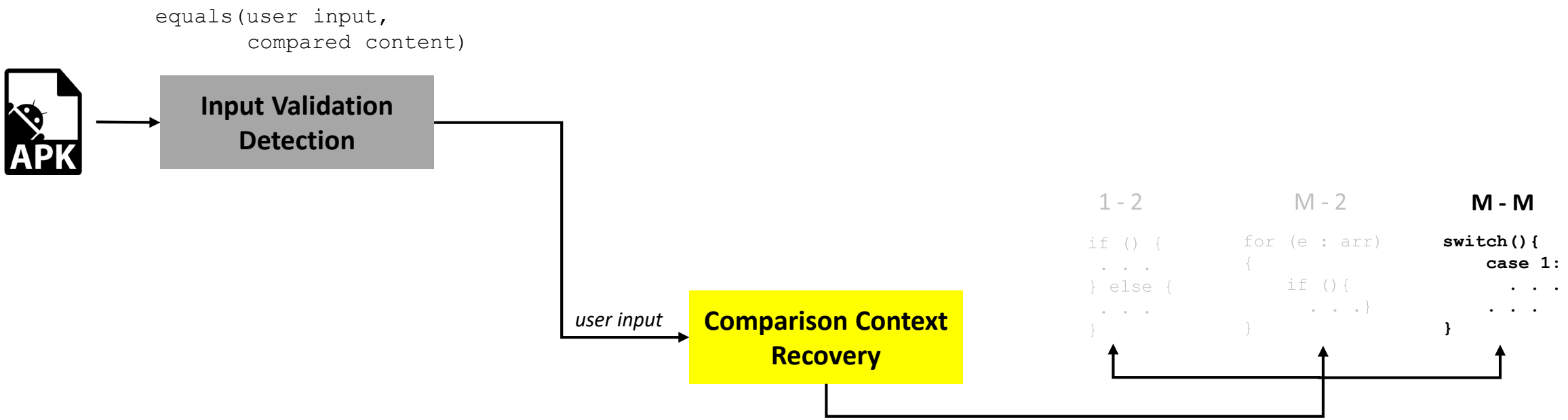
Dispatch	Times of Validations	Number of Branches
One-to-Two	**1**	**2**

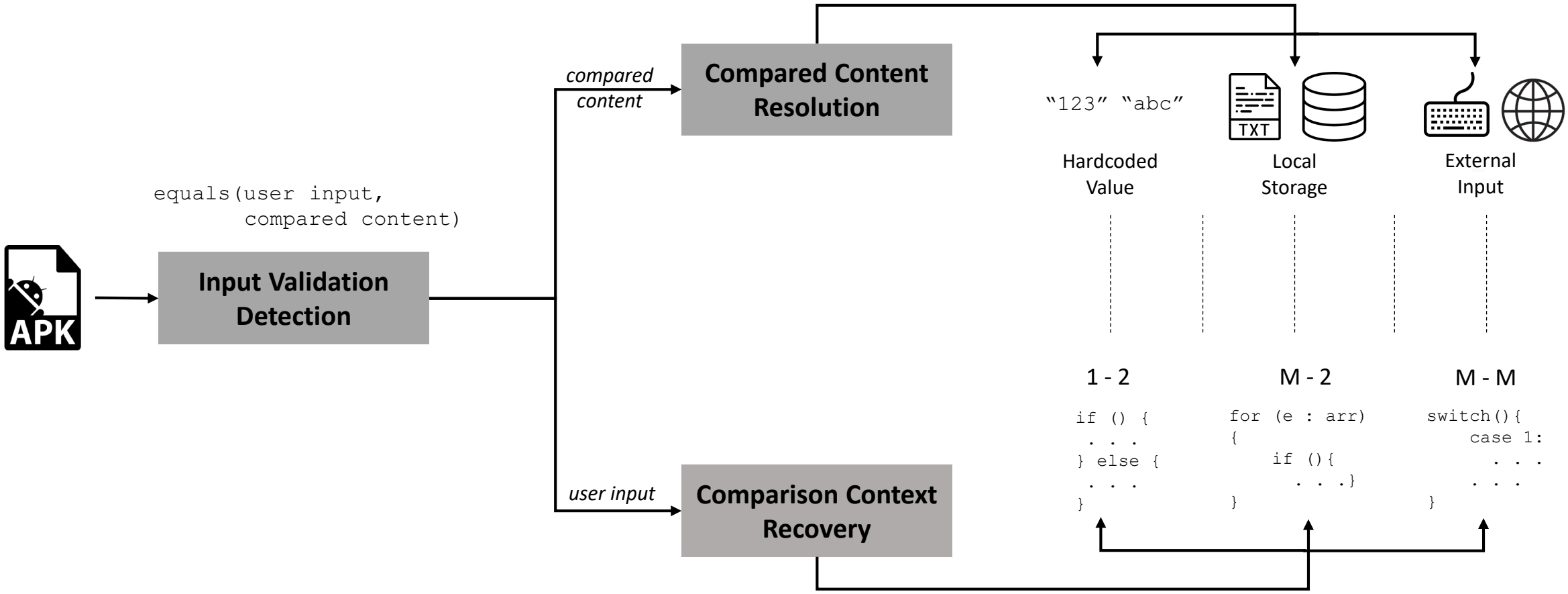


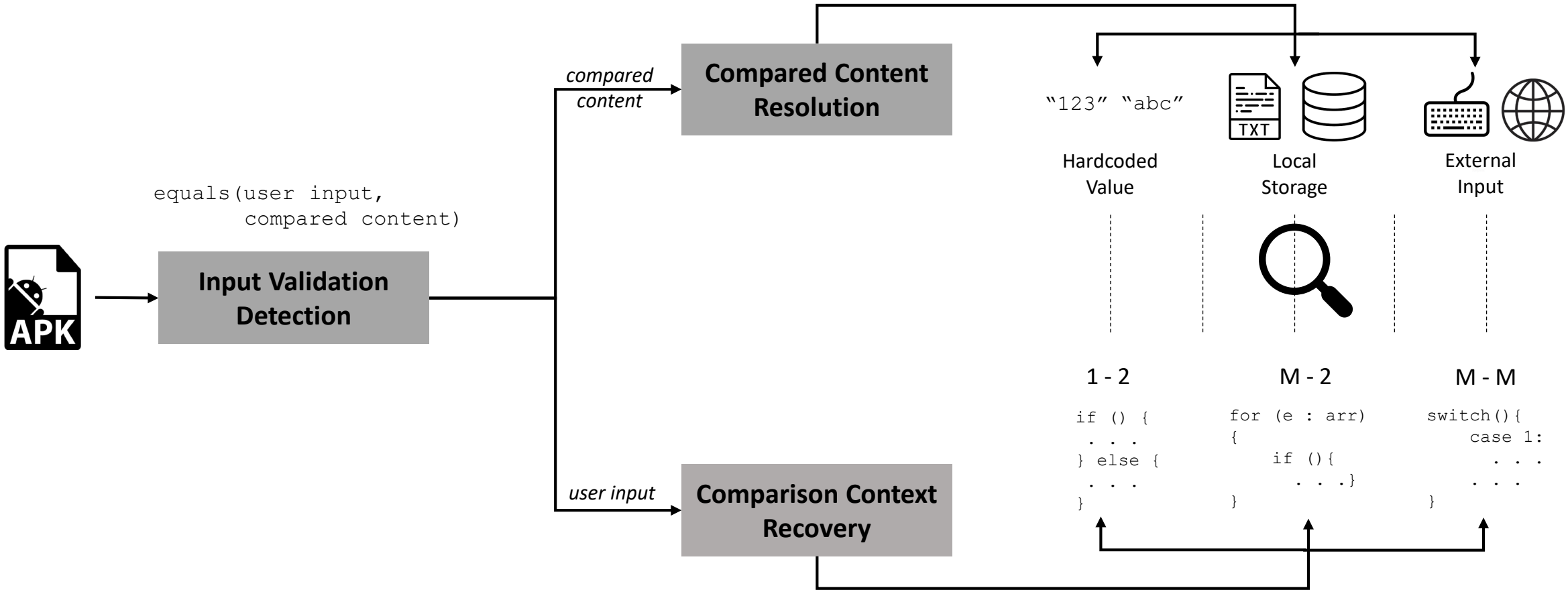
Dispatch	Times of Validations	Number of Branches
One-to-Two	**1**	**2**
Many-to-Two	**Many**	**2**



Dispatch	Times of Validations	Number of Branches
One-to-Two	**1**	**2**
Many-to-Two	**Many**	**2**
Many-to-Many	**Many**	**Many**




equals(user input,
compared content)
APK
Input Validation Detection
compared content
Compared Content Resolution
user input
Comparison Context Recovery
"123" "abc"
TXT
Hardcoded Value
Local Storage
External Input
1 - 2
M - 2
M - M
if () {
. . .
} else {
. . .
}
for (e : arr)
{
if (){
. . .}
}
switch(){
case 1:
. . .
. . .
}


equals(user input,
compared content)
APK
Input Validation Detection
compared content
user input
Compared Content Resolution
Comparison Context Recovery
"123" "abc"
TXT
Hardcoded Value
Local Storage
External Input
1 - 2
M - 2
M - M
if () {
. . .
} else {
. . .
}
for (e : arr)
{
if (){
. . .}
}
switch(){
case 1:
. . .
. . .
}

Access Key	• 1 -2 Dispatch • Hardcoded String

Comparison Content Types

"123" "abc"

Hardcoded Value | Local Storage | External Input

```
if(this.d.getText().toString()
    .equals("qpmzad") ) {
     . . . // removing ads
     return 0;
}
```

1 - 2

```
if () {
 . . .
} else {
 . . .
}
```

M - 2

```
for (e : arr)
{
    if (){
      . . .}
}
```

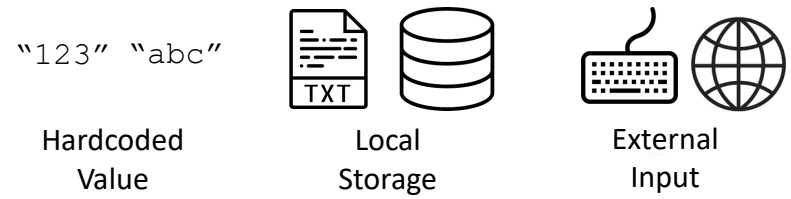
M - M

```
switch(){
    case 1:
      . . .
    . . .
}
```

Code Dispatch Behaviors

Access Key	• 1 -2 Dispatch • Hardcoded String
Master Password	• M – 2 Dispatch • Comparison values from different sources • One hardcoded string value

Comparison Content Types




```
if( v0.equals(v4_1) |
    v0.equals(
     "bypassPassword9977251"))
{
    . . . // viewing files

}
```

1 - 2

```
if () {
 . . .
} else {
 . . .
}
```

M - 2

```
for (e : arr)
{
    if (){
      . . .}
}
```

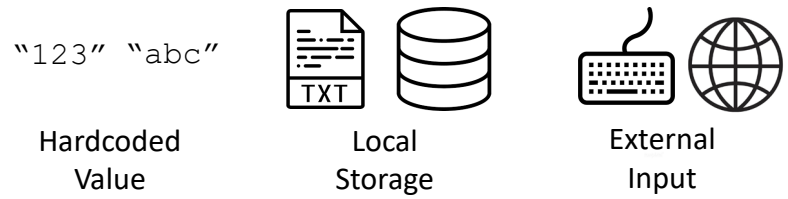
M - M

```
switch(){
    case 1:
      . . .
    . . .
}
```

Code Dispatch Behaviors

Access Key	• 1 -2 Dispatch • Hardcoded String
Master Password	• M – 2 Dispatch • Comparison values from different sources • One hardcoded string value
Blacklist	• M -2 Dispatch • Comparison value from the same source

Comparison Content Types




```
while( v1_1 < v3 ){
    if (TextUtil.equals(
            v2[v1_1], arg_6)) {
        v0 == true;
}
```

1 - 2

```
if () {
 . . .
} else {
 . . .
}
```

M - 2

```
for (e : arr)
{
    if (){
      . . .}
}
```

M - M

```
switch(){
    case 1:
      . . .
    . . .
}
```

Code Dispatch Behaviors

Access Key	• 1 -2 Dispatch • Hardcoded String
Master Password	• M – 2 Dispatch • Comparison values from different sources • One hardcoded string value
Blacklist	• M -2 Dispatch • Comparison value from the same source
Secret Command	• M- M Dispatch • At least one hardcoded string value

Comparison Content Types

Hardcoded Value | Local Storage | External Input

```
switch(a.getText()){
    case "bonusall":
        . . . // upgrading
    case "debug":
        . . . // debugging mode
    . . .
}
```

1 - 2

```
if () {
 . . .
} else {
 . . .
}
```

M - 2

```
for (e : arr)
{
    if (){
      . . .}
}
```

M - M

```
switch(){
    case 1:
      . . .
    . . .
}
```

Code Dispatch Behaviors

Item	Value
Total number of Apps Tested	150,000

Overall Statistics of The Evaluation Results

Item	Value
Total number of Apps Tested	150,000
Number of Apps from Google Play	100,000
Number of Apps from Pre-installs	30,000
Number of Apps from Baidu Market	20,000

Overall Statistics of The Evaluation Results

Item	Value
Total number of Apps Tested	150,000
Number of Apps from Google Play	100,000
Number of Apps from Pre-installs	30,000
Number of Apps from Baidu Market	20,000
Number of Apps W/ Access Keys	7,584

Overall Statistics of The Evaluation Results

Usage Description	Installs	App Name	Access Kyes
Hidden Admin Interface Login	5,000,000+	NBC Sports	UUDDLRLRBASS
	5,000,000+	1km	ilovemalang
	1,000,000+	Gaydar	y@dm1n
	1,000,000+	Jetcost	Jenny2018!#
	1,000,000+	Washington Post	w@5Hp0$t&P#
Arbitrary User Password Recovery	10,000,000+	Period Tracker	8424488
	5,000,000+	kpop lock screen	060890
	1,000,000+	War of Colony	qwerasdf123
	1,000,000+	Cute Note	hpasscode
	1,000,000+	Pattern lock screen	060890
Advanced Service Payment Bypassing	1,000,000+	Annotations	#remove.ads.withpwd:1570+
	1,000,000+	Arabic Dictionary	qpmzad
	1,000,000+	English Persian Dictionary	qpmzad
	1,000,000+	Translate Box	qpmzad
	1,000,000+	Speak to Voice Translator	qpmzad

Results of Top Inspected Secret Access Keys

Usage Description	Installs	App Name	Access Kyes
Hidden Admin Interface Login	5,000,000+	NBC Sports	UUDDLRLRBASS
	5,000,000+	1km	ilovemalang
	1,000,000+	Gaydar	y@dm1n
	1,000,000+	Jetcost	Jenny2018!#
	1,000,000+	Washington Post	w@5Hp0$t&P#
Arbitrary User Password Recovery	10,000,000+	Period Tracker	8424488
	5,000,000+	kpop lock screen	060890
	1,000,000+	War of Colony	qwerasdf123
	1,000,000+	Cute Note	hpasscode
	1,000,000+	Pattern lock screen	060890
Advanced Service Payment Bypassing	1,000,000+	Annotations	#remove.ads.withpwd:1570+
	1,000,000+	Arabic Dictionary	qpmzad
	1,000,000+	English Persian Dictionary	qpmzad
	1,000,000+	Translate Box	qpmzad
	1,000,000+	Speak to Voice Translator	qpmzad

Results of Top Inspected Secret Access Keys

Usage Description	Installs	App Name	Access Kyes
Hidden Admin Interface Login	5,000,000+	NBC Sports	UUDDLRLRBASS
	5,000,000+	1km	ilovemalang
	1,000,000+	Gaydar	y@dm1n
	1,000,000+	Jetcost	Jenny2018!#
	1,000,000+	Washington Post	w@5Hp0$t&P#
Arbitrary User Password Recovery	10,000,000+	Period Tracker	8424488
	5,000,000+	kpop lock screen	060890
	1,000,000+	War of Colony	qwerasdf123
	1,000,000+	Cute Note	hpasscode
	1,000,000+	Pattern lock screen	060890
Advanced Service Payment Bypassing	1,000,000+	Annotations	#remove.ads.withpwd:1570+
	1,000,000+	Arabic Dictionary	qpmzad
	1,000,000+	English Persian Dictionary	qpmzad
	1,000,000+	Translate Box	qpmzad
	1,000,000+	Speak to Voice Translator	qpmzad

Results of Top Inspected Secret Access Keys

Usage Description	Installs	App Name	Access Kyes
Hidden Admin Interface Login	5,000,000+	NBC Sports	UUDDLRLRBASS
	5,000,000+	1km	ilovemalang
	1,000,000+	Gaydar	y@dm1n
	1,000,000+	Jetcost	Jenny2018!#
	1,000,000+	Washington Post	w@5Hp0$t&P#
Arbitrary User Password Recovery	10,000,000+	Period Tracker	8424488
	5,000,000+	kpop lock screen	060890
	1,000,000+	War of Colony	qwerasdf123
	1,000,000+	Cute Note	hpasscode
	1,000,000+	Pattern lock screen	060890
Advanced Service Payment Bypassing	1,000,000+	Annotations	#remove.ads.withpwd:1570+
	1,000,000+	Arabic Dictionary	qpmzad
	1,000,000+	English Persian Dictionary	qpmzad
	1,000,000+	Translate Box	qpmzad
	1,000,000+	Speak to Voice Translator	qpmzad

Results of Top Inspected Secret Access Keys

Item	Value
Total number of Apps Tested	150,000
Number of Apps from Google Play	100,000
Number of Apps from Pre-installs	30,000
Number of Apps from Baidu Market	20,000
Number of Apps W/ Access Keys	7,584
Number of Apps W/ Master Passwords	501

Overall Statistics of The Evaluation Results

Installs	App Name	Master Passwords
10,000,000+	Lost Android	9382153981325298
5,000,000+	Photo,Video Locker	17621762
5,000,000+	Maya	19780902
1,000,000+	Anti Theft Screen Lock	42424242
1,000,000+	Kids Place	5493
1,000,000+	Journal costs	oj9qbnrsfld0x3as
1,000,000+	Secret Diary	1vulne43
500,000+	MOMS	0001
500,000+	Quick Note	1349100416
500,000+	Hide and Lock	bypassPassword9977251

Results of Top Inspected Master Passwords

Installs	App Name	Master Passwords
10,000,000+	Lost Android	9382153981325298
5,000,000+	Photo,Video Locker	17621762
5,000,000+	Maya	19780902
1,000,000+	Anti Theft Screen Lock	42424242
1,000,000+	Kids Place	5493
1,000,000+	Journal costs	oj9qbnrsfld0x3as
1,000,000+	Secret Diary	1vulne43
500,000+	MOMS	0001
500,000+	Quick Note	1349100416
500,000+	Hide and Lock	bypassPassword9977251

Results of Top Inspected Master Passwords

Item	Value
Total number of Apps Tested	150,000
Number of Apps from Google Play	100,000
Number of Apps from Pre-installs	30,000
Number of Apps from Baidu Market	20,000
Number of Apps W/ Access Keys	7,584
Number of Apps W/ Master Passwords	501
Number of Apps W/ Secret Commands	6,013

Overall Statistics of The Evaluation Results

Installs	App Name	Secret Commands
10,000,000+	Wheres My Droid	wmdscdfnyl, debugcon, Brad.Degelau, wmdlock ...
10,000,000+	MP3 Cutter	enableartistalbum, disableartistalbum ...
10,000,000+	Shooting club	823141744, 820196108, 694666759 ...
5,000,000+	Driver Test	testmode on, bonusall, free, bonus1, debug ...
1,000,000+	Pocket Auctions for eBay	proxyoff, debughttp, connoff, proxyon, ...
1,000,000+	BrainPOP	STOPPD#, MEM, GIMMEUS, GDI, DMP, CLR ...
1,000,000+	Forgotten Tales	crash, export ...
1,000,000+	Connect Dialer Speed	*#03#, *#0101#, *#05*3#, *#05*5#, *#0102# ...
1,000,000+	Euclidea	unlock packs, lock all, lock packs, unlock all ...
1,000,000+	Character Story Planner	(maroonAuth), (amberAuth), (darkCyanAuth) ...

Results of Top Inspected Secret Commands

Installs	App Name	Secret Commands
10,000,000+	Wheres My Droid	wmdscdfnyl, debugcon, Brad.Degelau, wmdlock ...
10,000,000+	MP3 Cutter	enableartistalbum, disableartistalbum ...
10,000,000+	Shooting club	823141744, 820196108, 694666759 ...
5,000,000+	Driver Test	testmode on, bonusall, free, bonus1, debug ...
1,000,000+	Pocket Auctions for eBay	proxyoff, debughttp, connoff, proxyon, ...
1,000,000+	BrainPOP	STOPPD#, MEM, GIMMEUS, GDI, DMP, CLR ...
1,000,000+	Forgotten Tales	crash, export ...
1,000,000+	Connect Dialer Speed	*#03#, *#0101#, *#05*3#, *#05*5#, *#0102# ...
1,000,000+	Euclidea	unlock packs, lock all, lock packs, unlock all ...
1,000,000+	Character Story Planner	(maroonAuth), (amberAuth), (darkCyanAuth) ...

Results of Top Inspected Secret Commands

Item	Value
Total number of Apps Tested	150,000
Number of Apps from Google Play	100,000
Number of Apps from Pre-installs	30,000
Number of Apps from Baidu Market	20,000
Number of Apps W/ Access Keys	7,584
Number of Apps W/ Master Passwords	501
Number of Apps W/ Secret Commands	6,013
Number of Apps W/ Blacklist Secrets	4,028

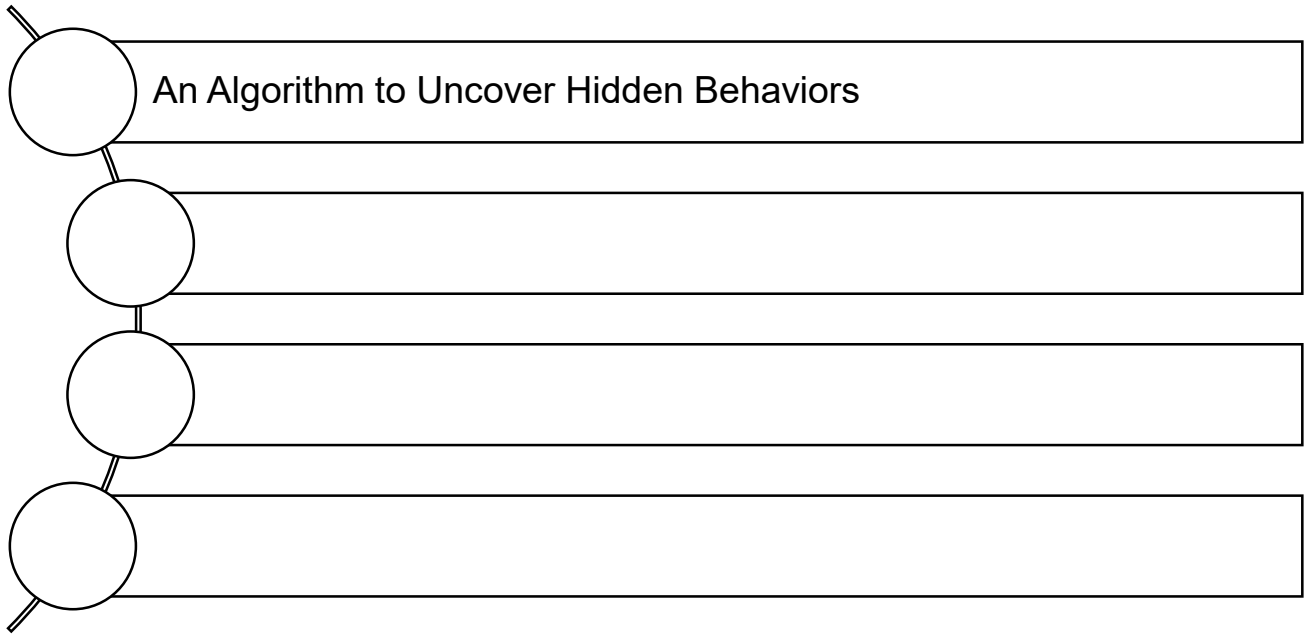
Overall Statistics of The Evaluation Results

Category	Detailed Blacklist Type
Drug	01-Addictive Drug, 02-Aphrodisiac, 03-Hallucinogen
Cult	04-Cults Name, 05-Malignant Event
Fraud	06-Fake Certificates, 07-MLM
Gamble	08-Chess&Card, 09-Lottery, 10-Jockey
Insult	11-Bullying, 12-Racial Discrimination, 13-Obscenity,
Password	14-Weak Password
Politics	15-Leaders Name, 16-Mass Incident, 17-Rebel, 18-Parade, 19-Separatist
Pornography	20-Adult Video, 21-Escort Service
Website	22-Anti—government, 23-Fake News, 24-Pornography, 25-Criminal

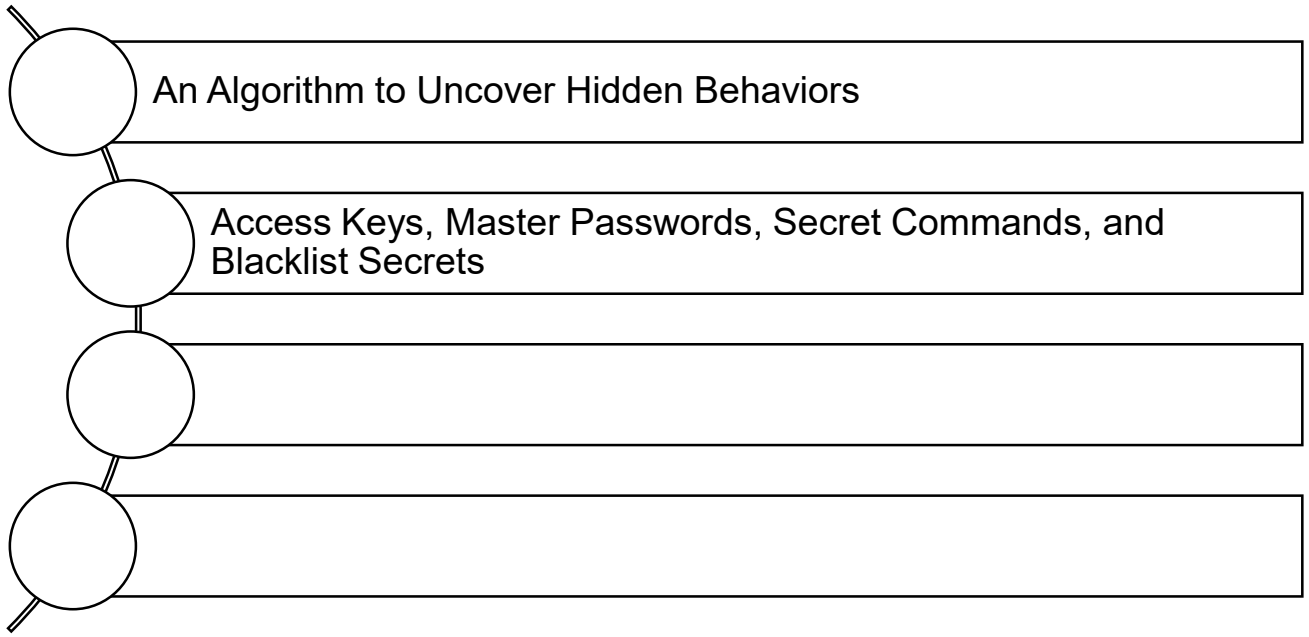
Results of Top Inspected Blacklist Types

		Drug			Cult		Fraud			Gamble			Insult		PW			Politics			Porn			Website		
Installs	Package Name	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
10,000,000+	pokerdeluxe	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	○	○	○
5,000,000+	f3	○	○	○	○	○	○	○	○	○	○	●	●	●	●	○	○	○	○	○	○	○	○	○	○	○
5,000,000+	snakeoff	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	●	●
500,000+	joyride	○	○	○	○	○	○	○	○	○	○	●	●	●	○	○	○	○	○	○	○	○	○	●	○	○
100,000+	partyline	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○
5,000,000+	mafia42	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○
1,000,000+	quackquack	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○
100,000+	lovecompat	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	○	○	○
100,000+	videochat	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	●	○	○
50,000+	doodletoss	○	○	○	○	○	○	○	○	○	○	●	○	●	○	○	○	○	○	○	○	○	○	●	○	○
50,000,000+	lft	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	●	●
50,000,000+	yuantiku	●	●	●	●	●	●	○	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	○	●	○
10,000,000+	ikuaiyue	●	●	●	●	○	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	○	●	●
1,000,000+	yiyuan	●	●	●	●	●	●	●	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	●	●	●
1,000,000+	modernsky	●	●	●	●	●	●	○	●	●	●	●	●	●	○	●	●	●	●	●	●	●	●	○	●	●
10,000,000+	zhaopin	○	○	○	●	●	●	●	○	○	○	○	○	●	○	●	●	○	●	●	○	●	○	●	○	○
10,000,000+	qingka	○	○	○	●	●	●	○	○	●	○	●	○	●	○	●	●	●	○	●	●	●	●	○	●	○
5,000,000+	attention	○	●	○	●	○	●	○	○	○	○	●	○	●	○	●	●	●	○	●	●	○	○	○	○	●
1,000,000+	yaya	○	○	●	●	○	●	●	○	○	○	●	○	●	○	●	●	○	●	●	●	●	○	○	○	○
1,000,000+	yilan	●	○	●	●	●	●	○	●	●	○	●	○	●	○	●	●	○	○	●	●	●	●	●	●	●

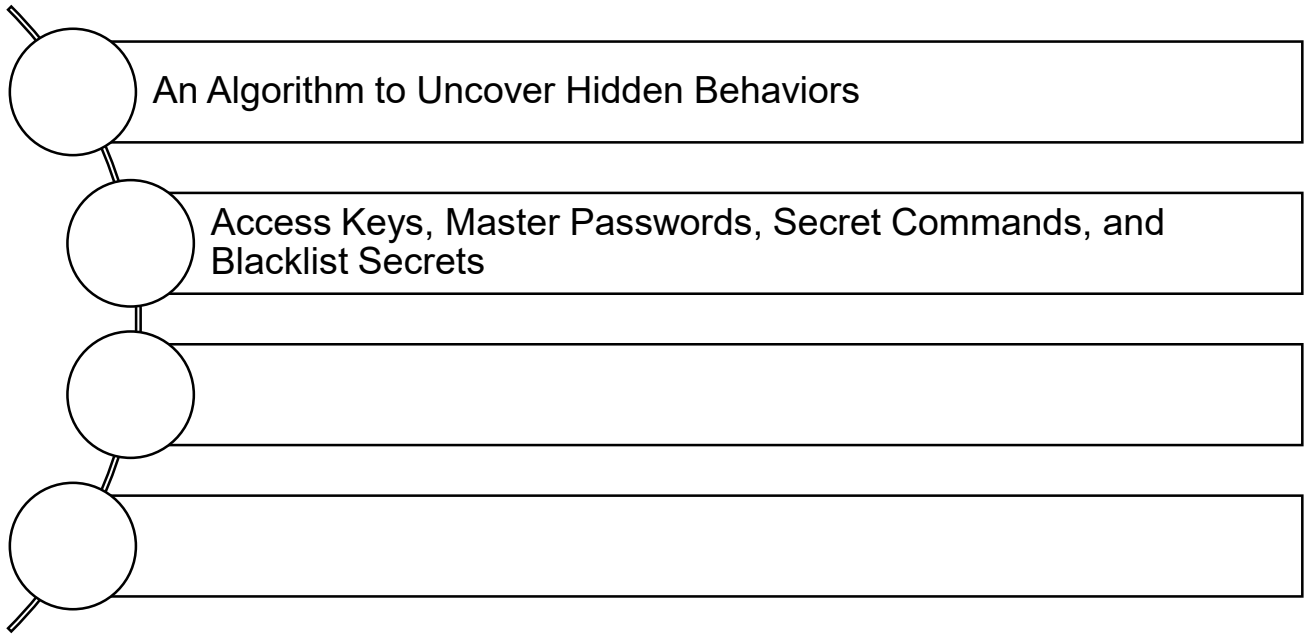
Results of Top Inspected Blacklists



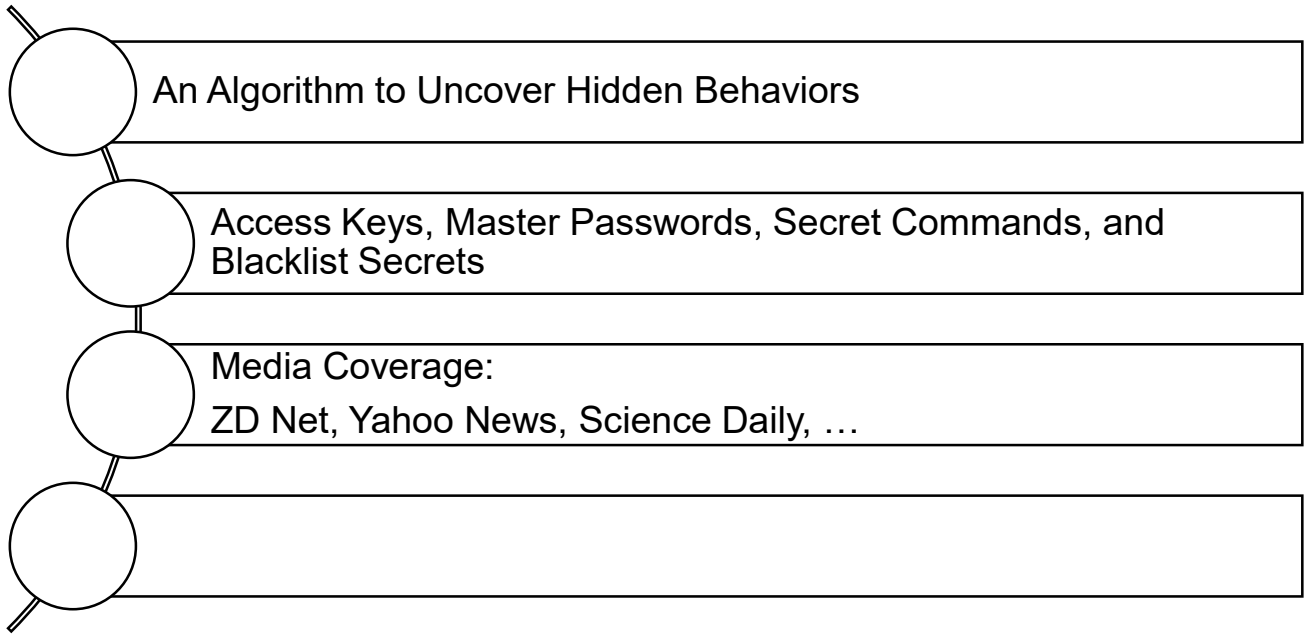
Summary


An Algorithm to Uncover Hidden Behaviors
Access Keys, Master Passwords, Secret Commands, and Blacklist Secrets

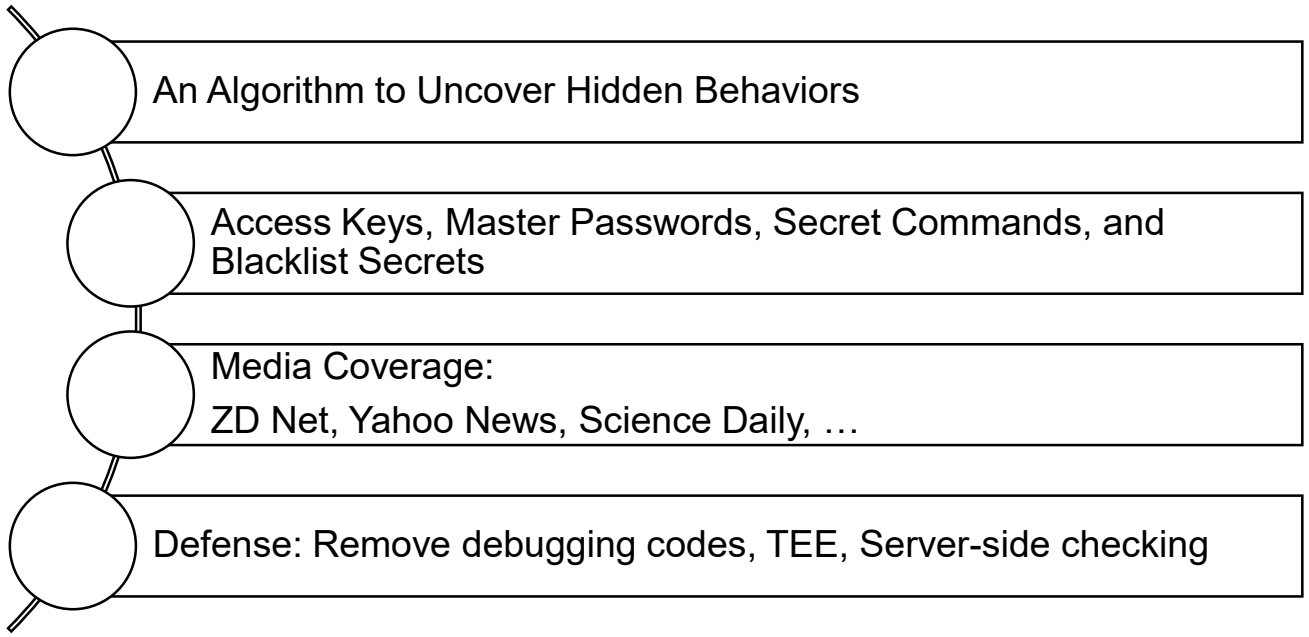
Summary


An Algorithm to Uncover Hidden Behaviors
Access Keys, Master Passwords, Secret Commands, and Blacklist Secrets

Summary


An Algorithm to Uncover Hidden Behaviors
Access Keys, Master Passwords, Secret Commands, and Blacklist Secrets
Media Coverage:
ZD Net, Yahoo News, Science Daily, …

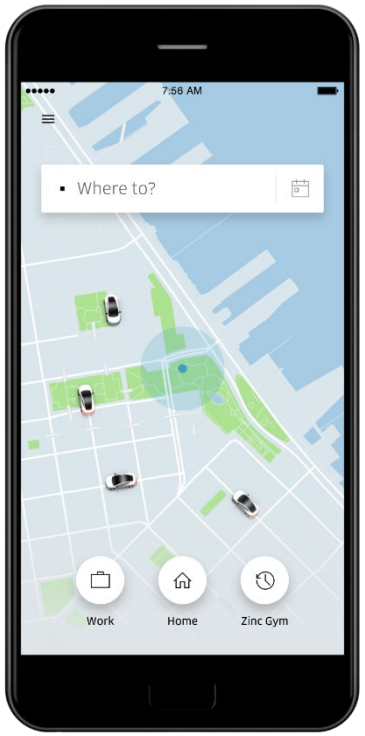
Summary


An Algorithm to Uncover Hidden Behaviors
Access Keys, Master Passwords, Secret Commands, and Blacklist Secrets
Media Coverage:
ZD Net, Yahoo News, Science Daily, …
Defense: Remove debugging codes, TEE, Server-side checking

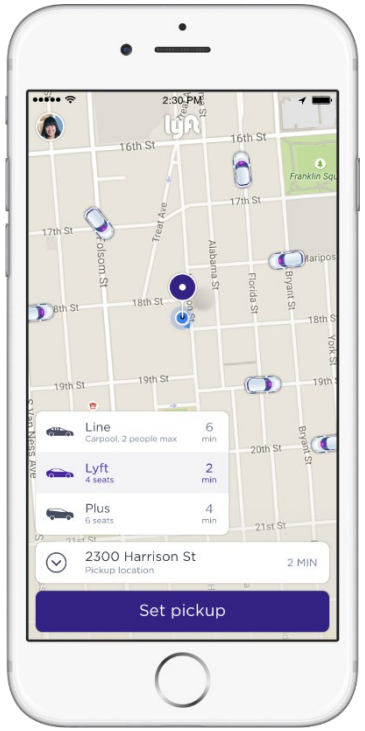
Summary

Geo-locating Drivers: A Study of Sensitive Data Leakage in Ride-Hailing Services

In Proceedings of the 26th ISOC Network and Distributed System Security (NDSS) Symposium, 2019.



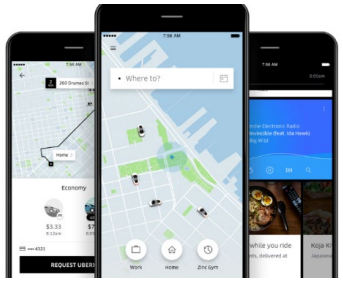
Uber



lyft

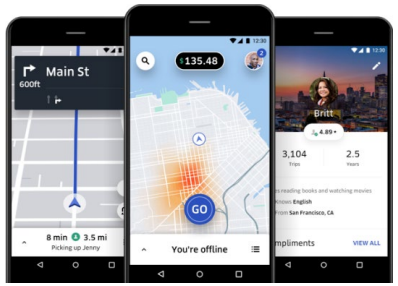
滴滴出行

Popular Ride-hailing Services

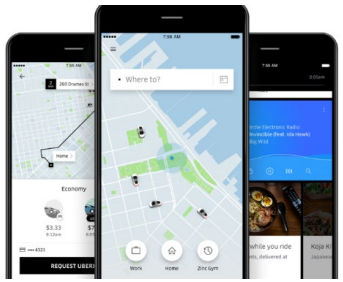


Rider App

Backend Servers



Driver App

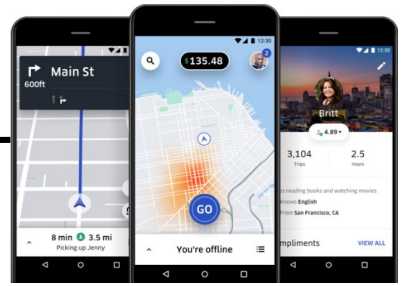


Rider App

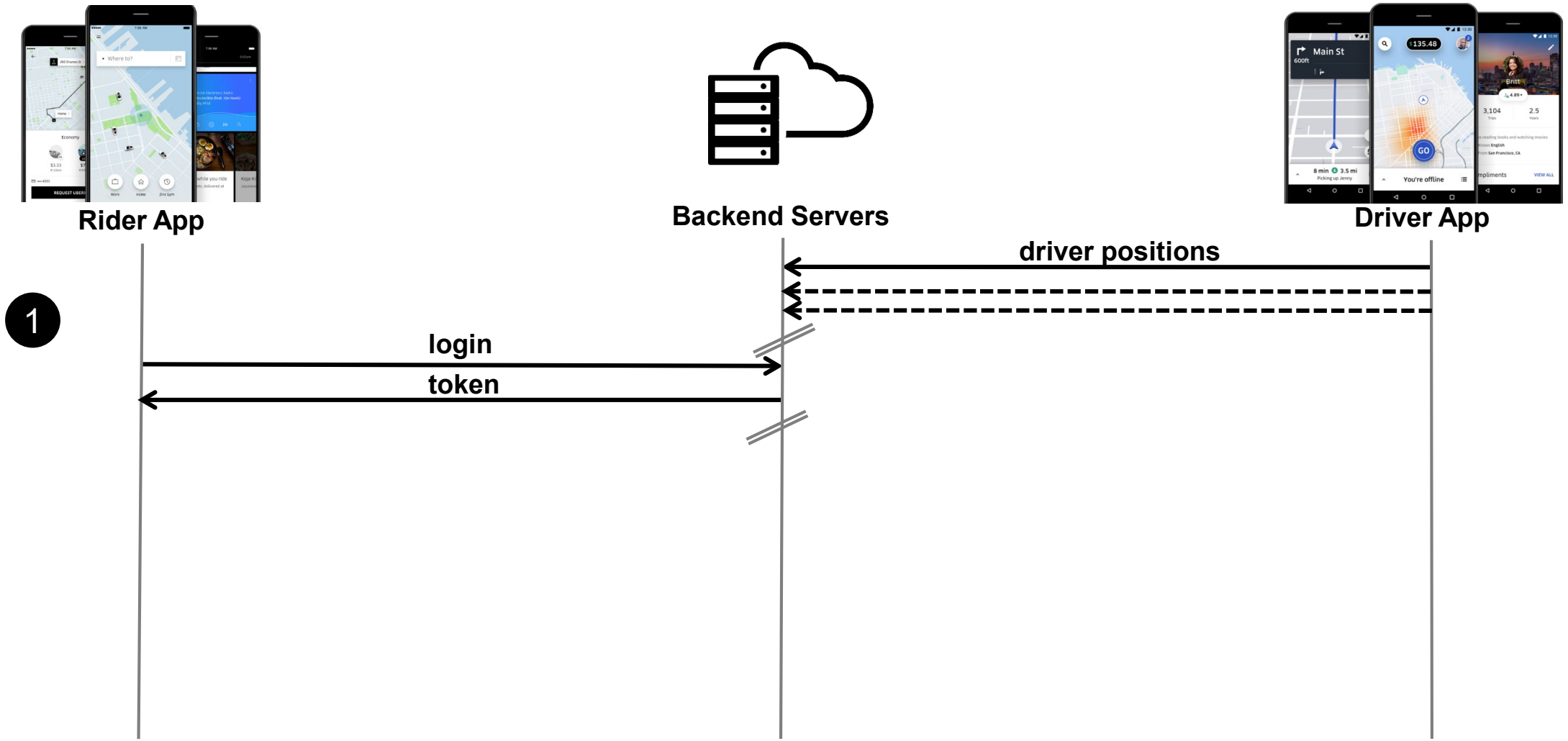
Rider
GPS,PII

Backend Servers

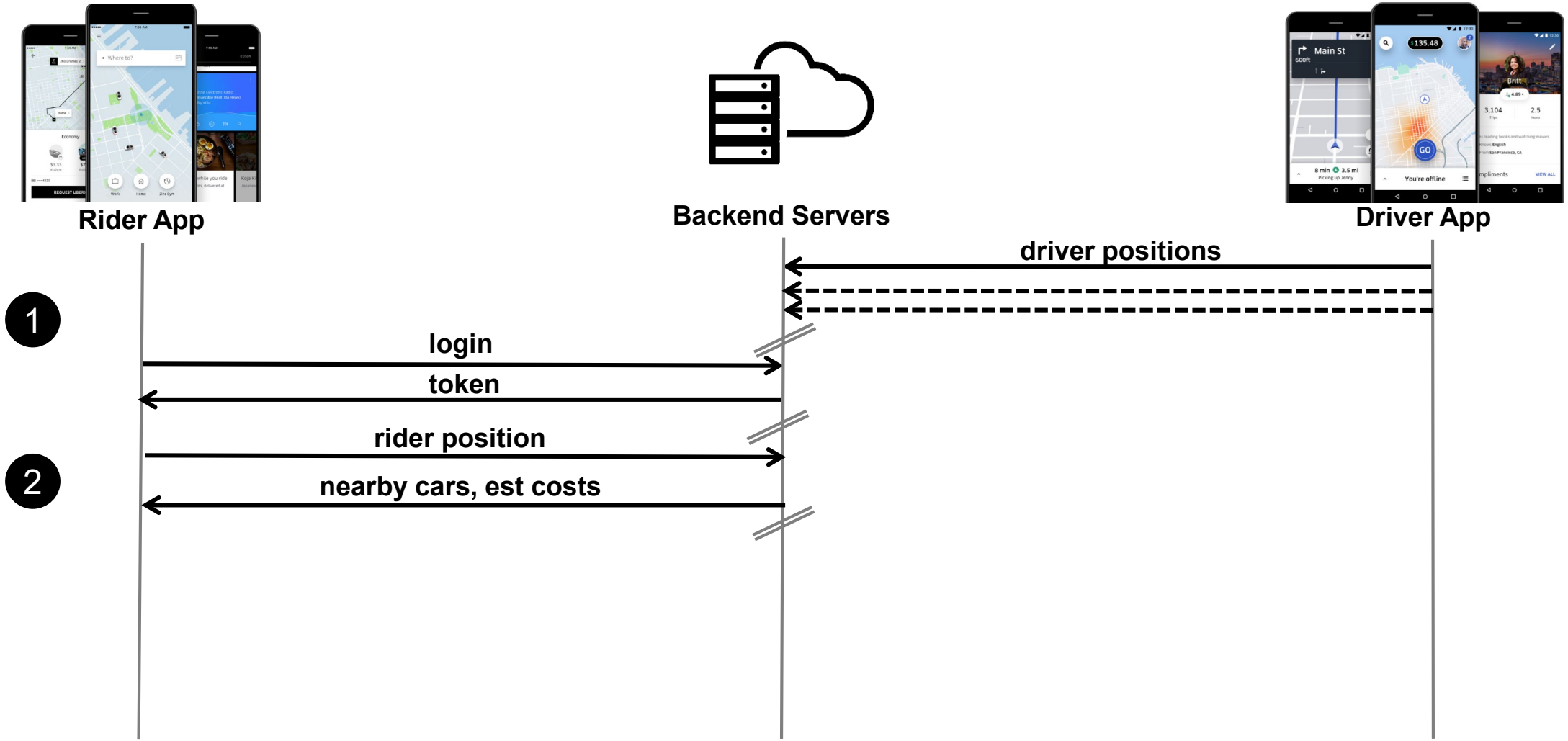
Driver
GPS, PII



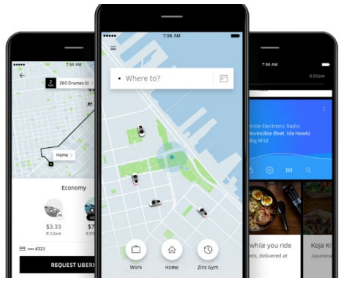
Driver App



A Simplified Protocol of Ride-hailing Services

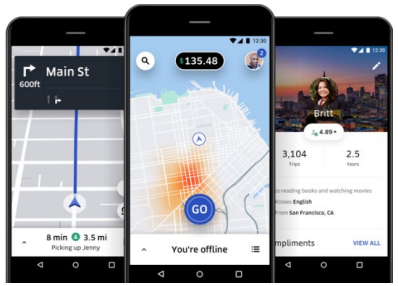


A Simplified Protocol of Ride-hailing Services

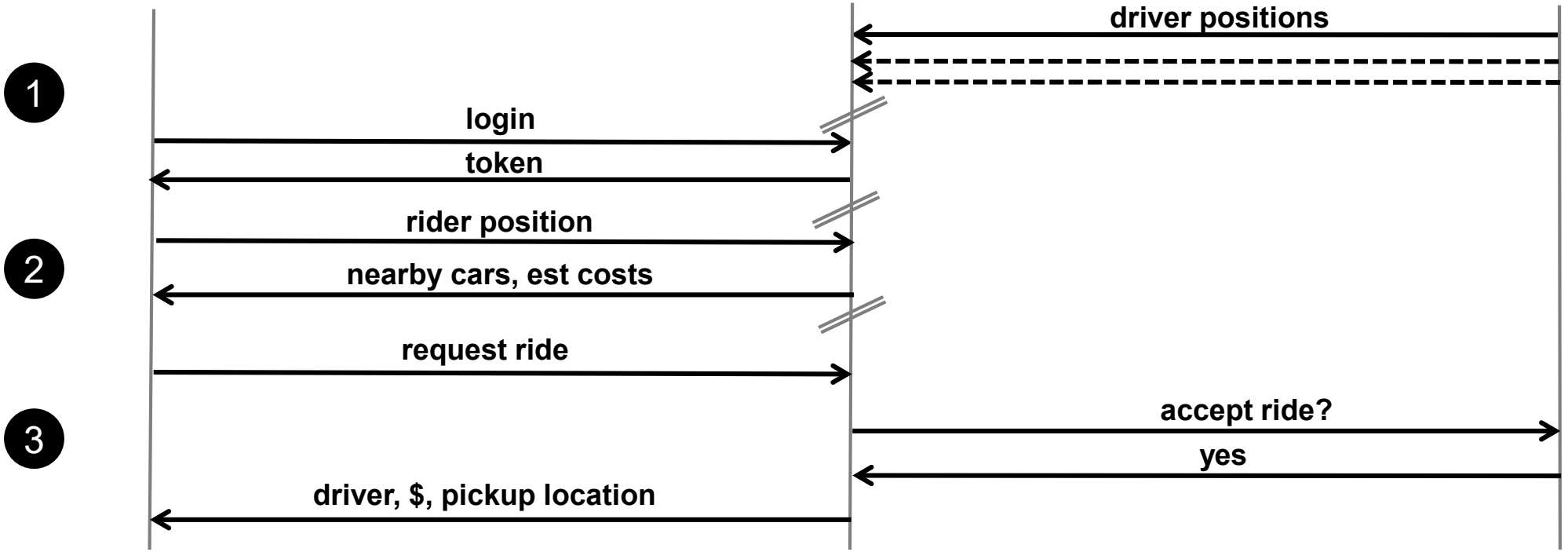


Rider App

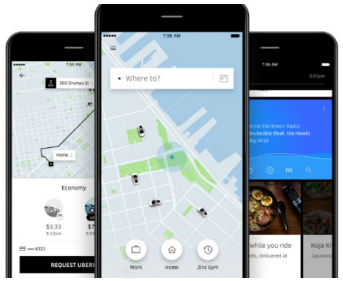
Backend Servers



Driver App

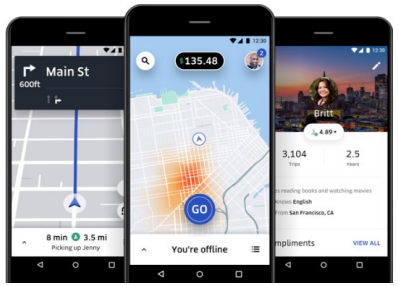


A Simplified Protocol of Ride-hailing Services



Rider App

Backend Servers



Driver App

Uber taxi driver held on rape charge is serial sex offender, Indian media claim

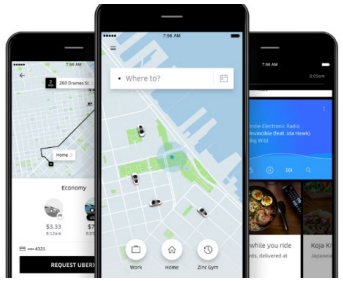
According to Hindustan Times Shiv Kumar Yadav was arrested and held for six months last year on another rape charge

Woman raped by Uber driver in India sues company for privacy breaches

- **Woman alleges executives obtained medical records to cast doubt on claims**
- **Lawsuit comes at time of considerable turmoil for scandal-hit company**

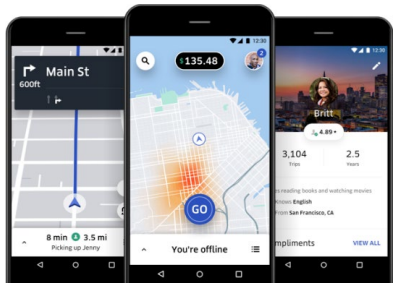
Uber revealed that more than 3,000 sexual assaults occurred during rides on its service in the U.S. in 2018. It shared these statistics in a comprehensive U.S. Safety Report on Thursday.

Security Concerns



Rider App

Backend Servers



Driver App

Uber taxi driver held on rape charge i… serial sex offender, Indian media

According to Hindustan Times Shiv Kumar Yada… held for six months last year on another rape…

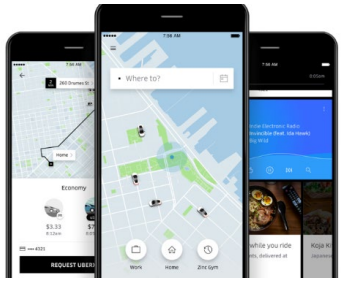
Woman raped by Ub… sues company f…

- Woman alleges … doubt on cl… …s to cast
- Lawsu… …noil for scandal-hit co…

The ride-hailing company and peers, including Lyft, have been criticized over these personal safety issues for years. Uber has seen an uptick in lawsuits over sexual assaults that allegedly occurred between riders and drivers in recent years.

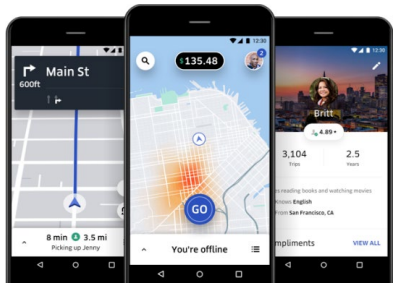
…nan 3,000 sexual assaults occurred during rides on …s. in 2018. It shared these statistics in a comprehensive …eport on Thursday.

Security Concerns



Rider App

Backend Servers



Driver App

Uber taxi driver held on rape charge i… serial sex offender, Indian media

According to Hindustan Times Shiv Kumar Yadav… held for six months last year on another rape…

Woman raped by Ub… sues company f…

- Woman alleges… doubt on cl… …s to cast
- Lawsu… …noil for scandal-hit co…

The ride-hailing company and peers, including Lyft, have been criticized over these personal safety issues for years. Uber has seen an uptick in lawsuits over sexual assaults that allegedly occurred between riders and drivers in recent years.

…nan 3,000 sexual assaults occurred during rides on …. in 2018. It shared these statistics in a comprehensive …eport on Thursday.

Uber under assault around the world as taxi drivers fight back

Gregg Zoroya and Angela Waters, USA TODAY Published 3:44 p.m. ET July 7, 2015 | Updated 6:59 p.m. ET July 7, 2015

(Photo11: Michel Euler,

While conceding France is … stories.

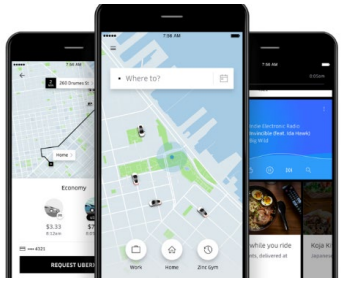
ANGRY TAXI DRIVERS ON STRIKE ATTACK UBER TAXIS IN DOWNTOWN ATHENS (VIDEOS)

March 6, 2018 Social 684 Views

Like 0 Save Share 1

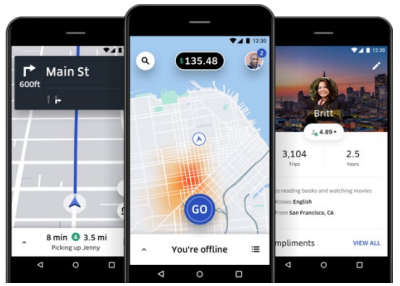
Angry taxi drivers on work stoppage attacked Uber drivers but also their colleagues who had refused to join the 9-hour work stoppage in Athens and Attica on Tuesday. strike. It was mostly Uber drivers who

Security Concerns

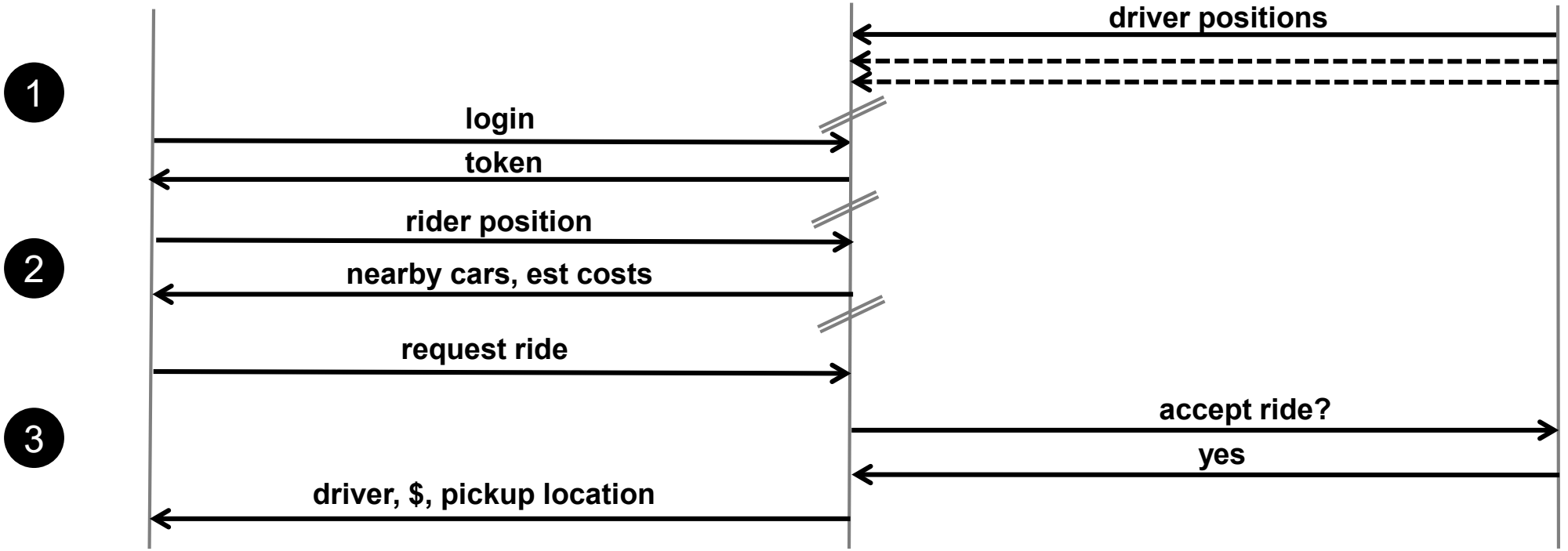


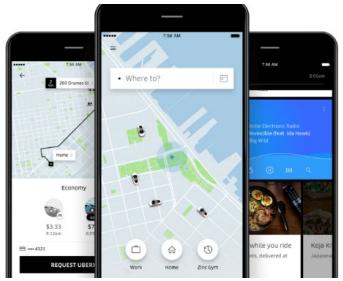
Rider App

Backend Servers



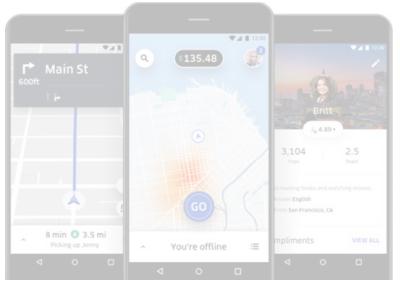
Driver App



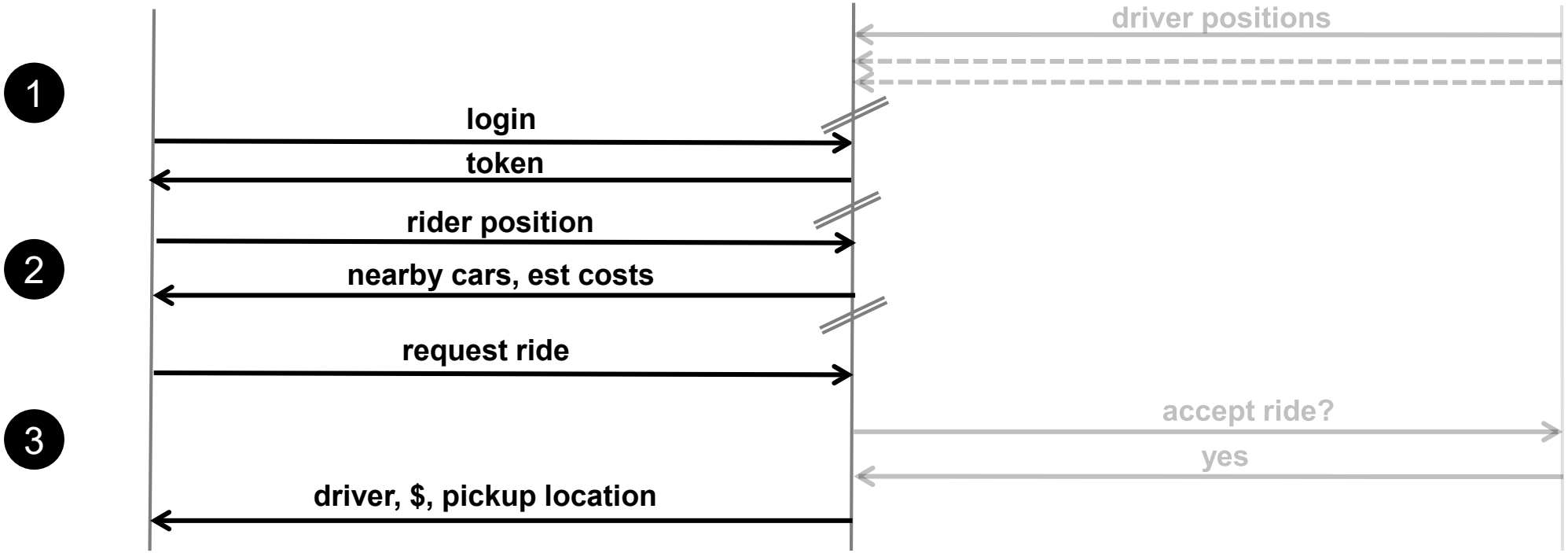


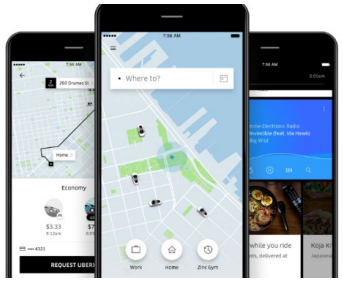
Rider App

Backend Servers



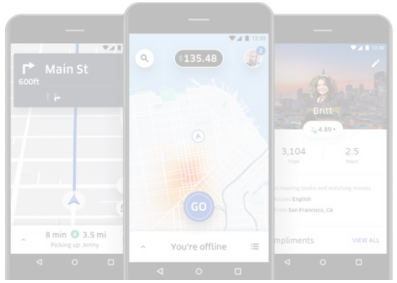
Driver App



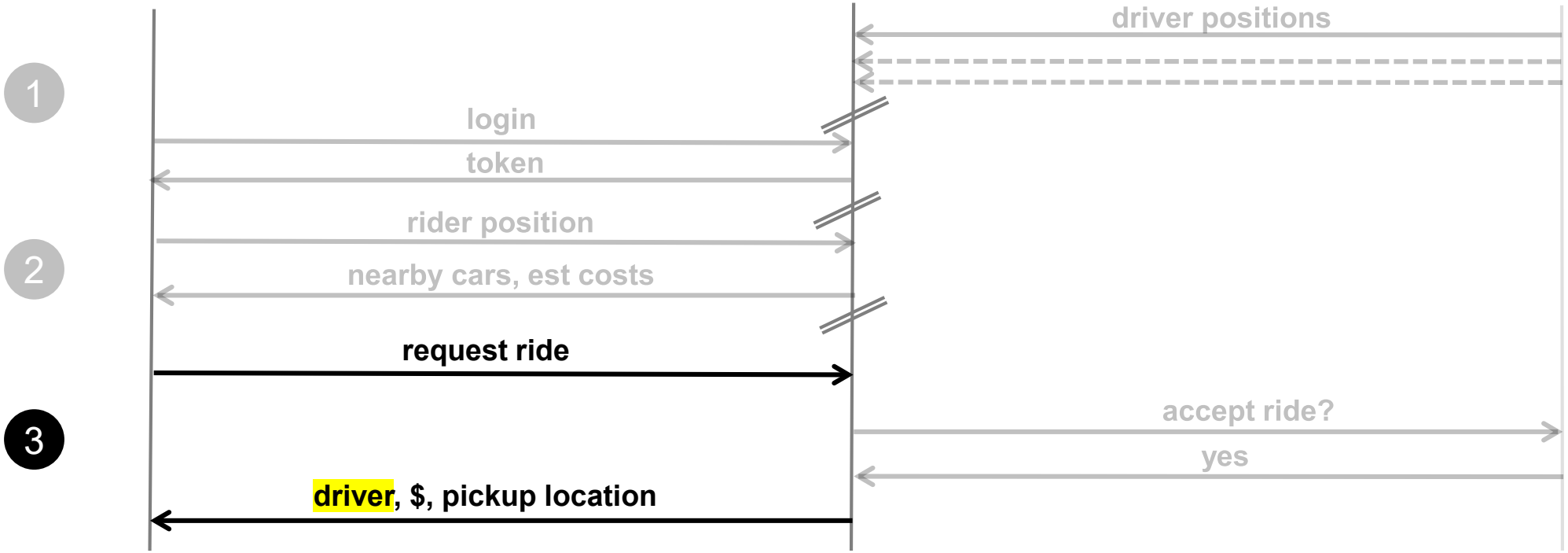


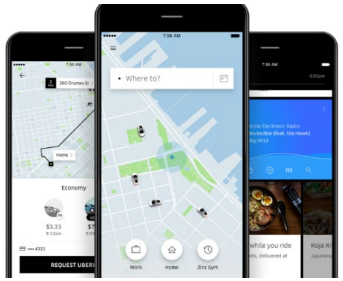
Rider App

Backend Servers



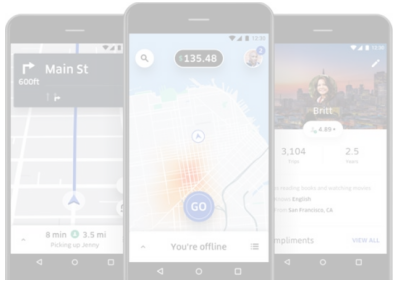
Driver App



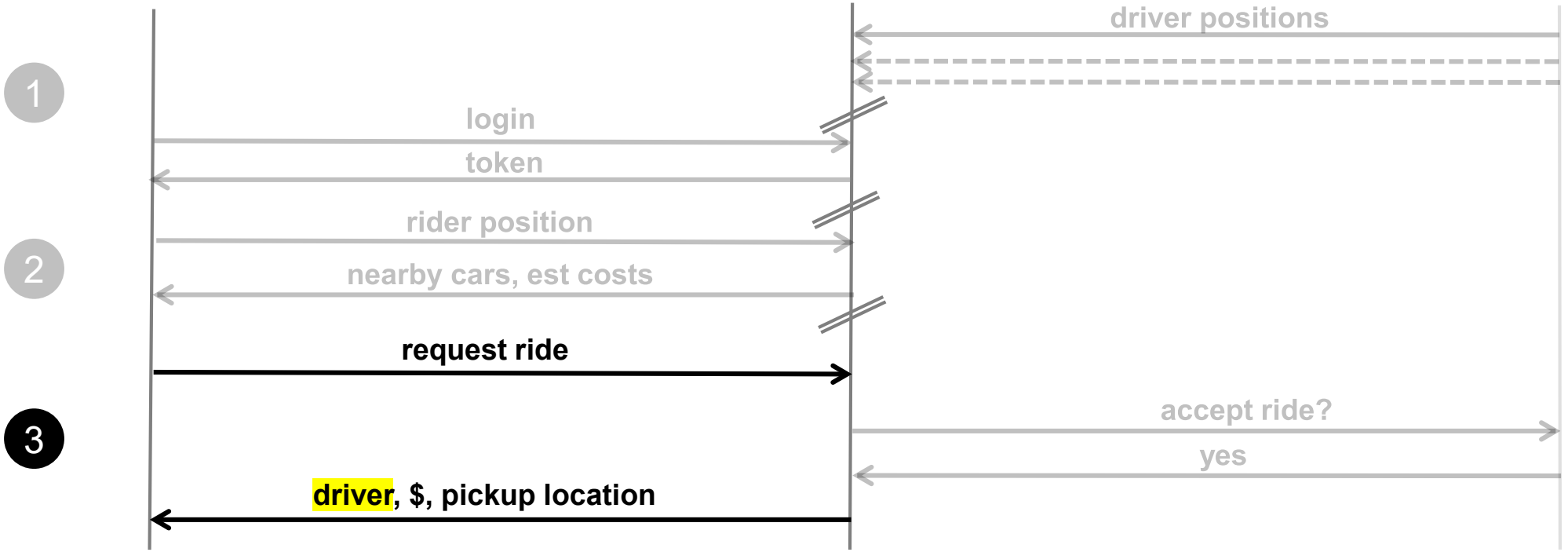


Rider App

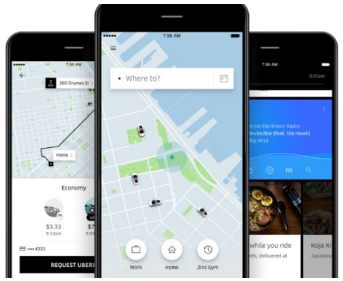
Backend Servers



Driver App

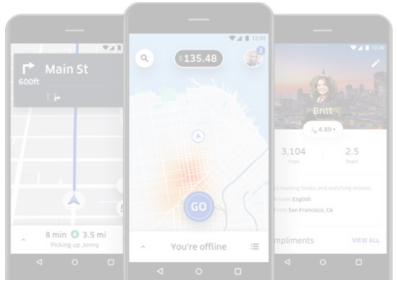


Pham et al. 2017, PoPETs

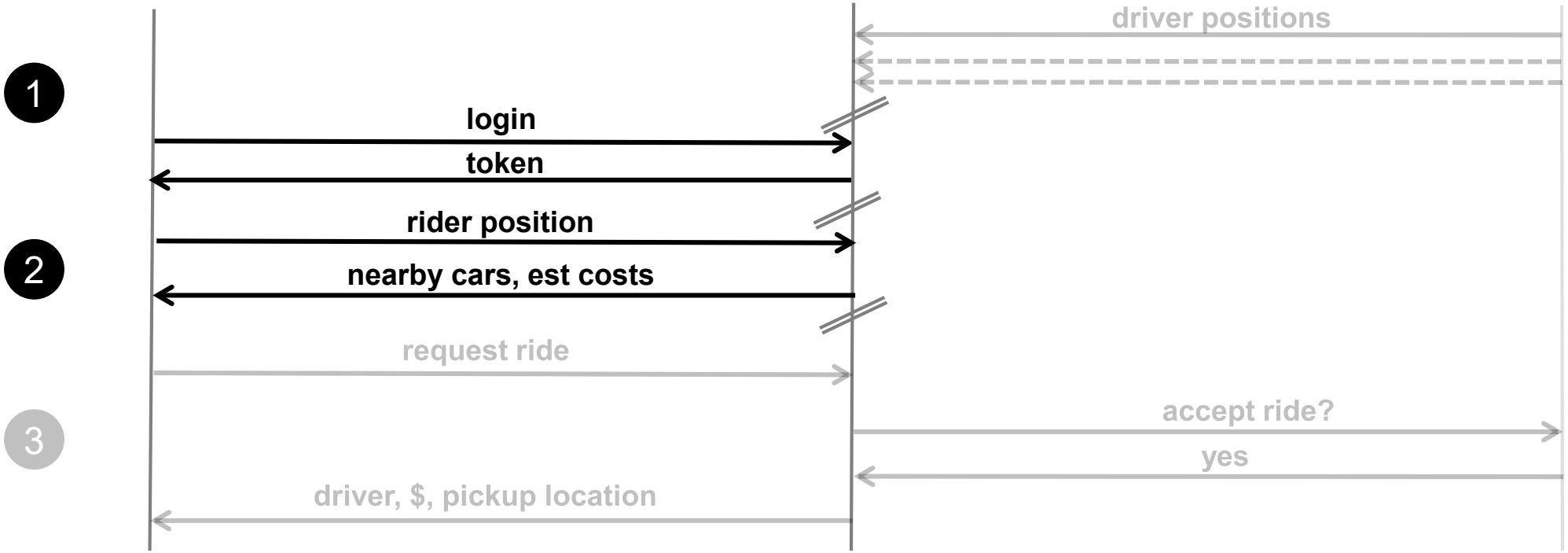


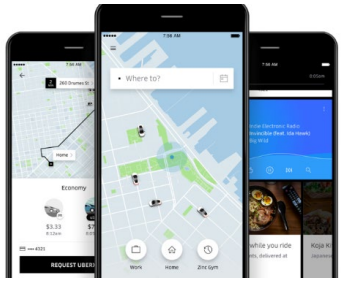
Rider App

Backend Servers



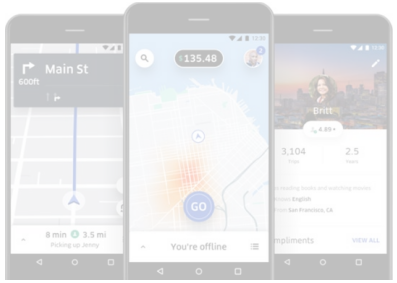
Driver App



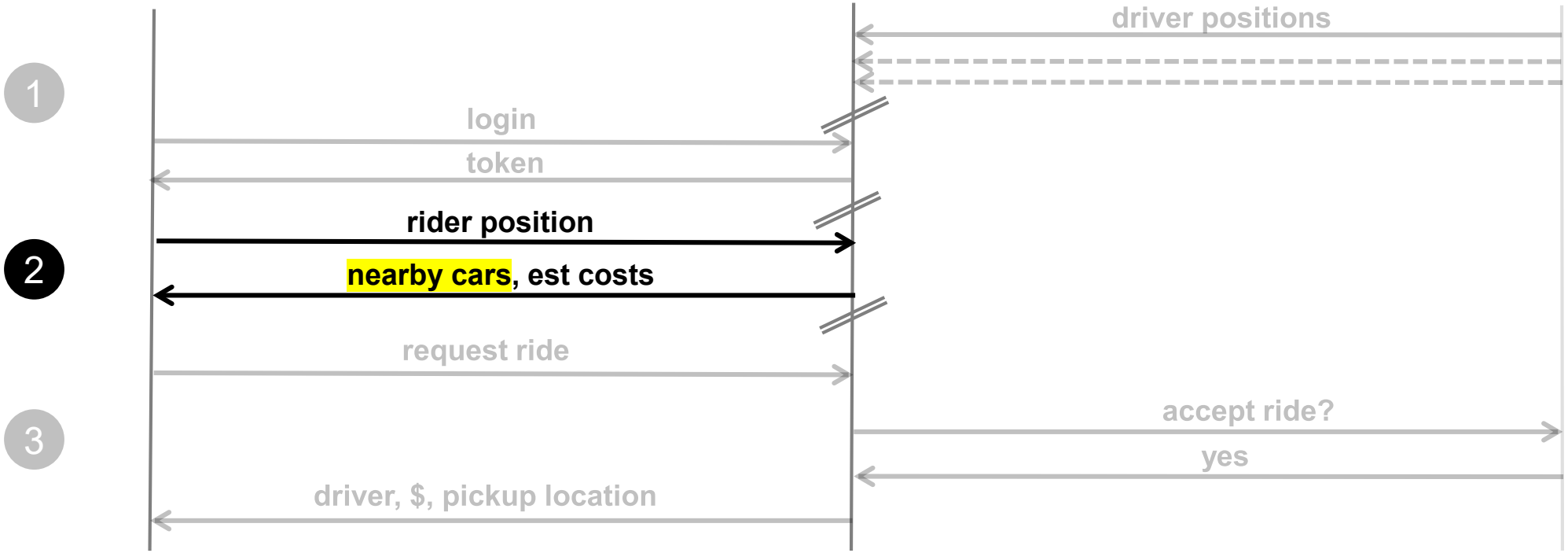


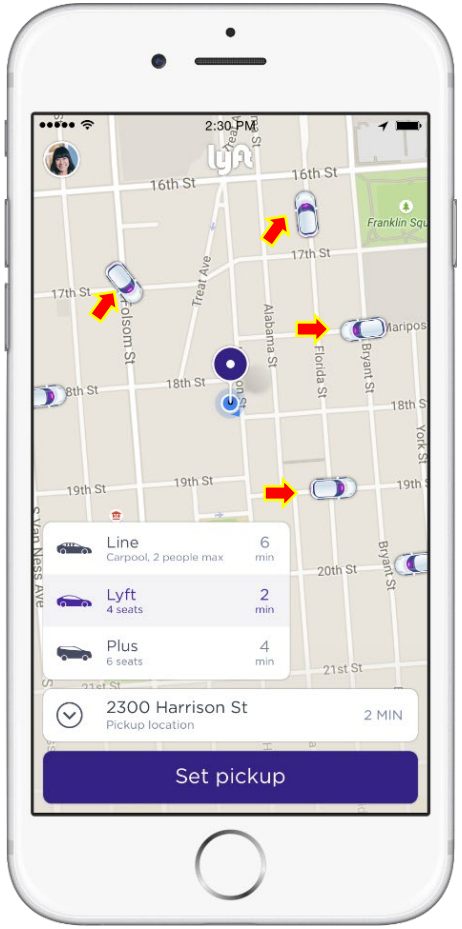
Rider App

Backend Servers

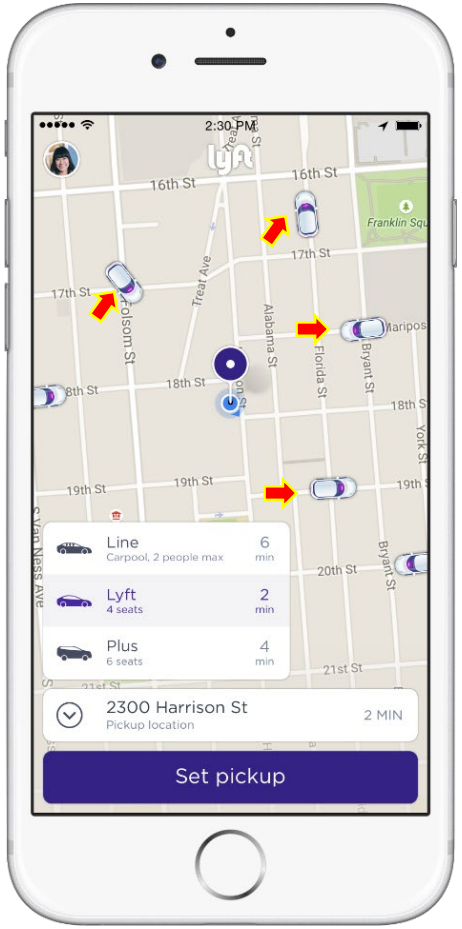


Driver App



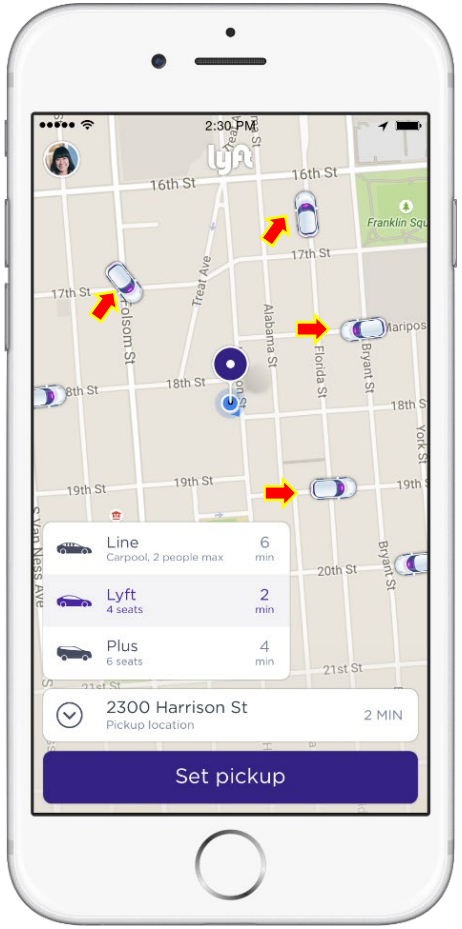


Nearby Cars Functionality



Why Moving Smooth?

Suspicious ***Nearby Cars*** Functionality



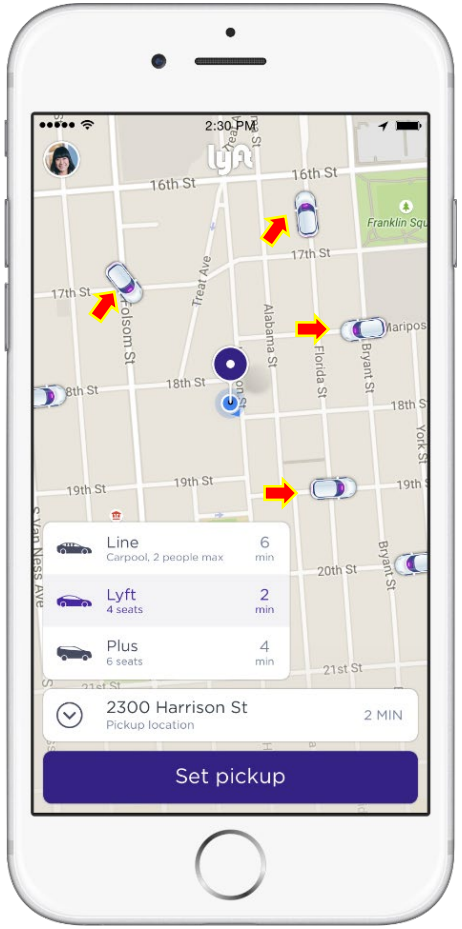
Why Moving Smooth?

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```

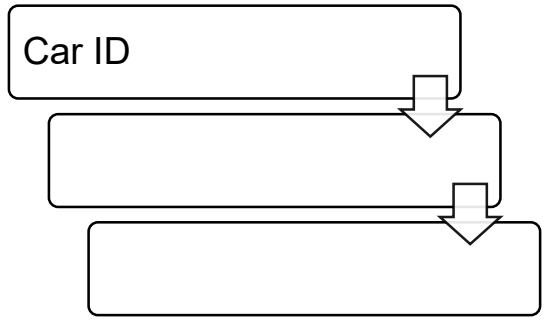
```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Suspicious ***Nearby Cars*** Functionality



Why Moving Smooth?

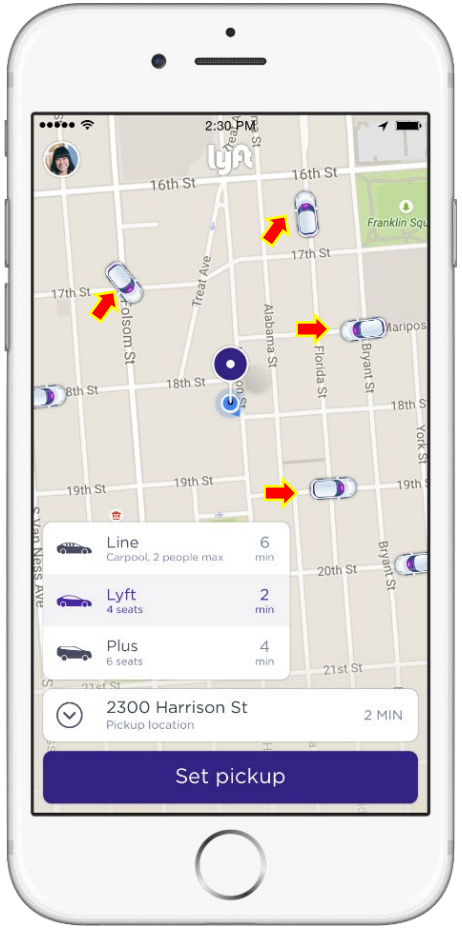



```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

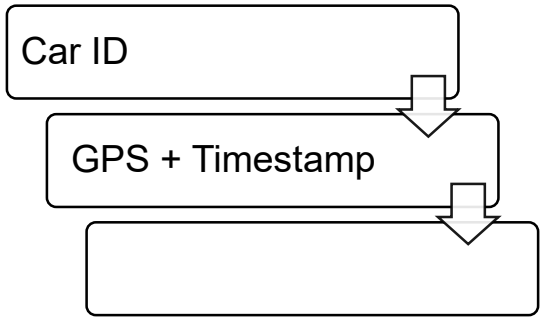
```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Suspicious ***Nearby Cars*** Functionality



Why Moving Smooth?

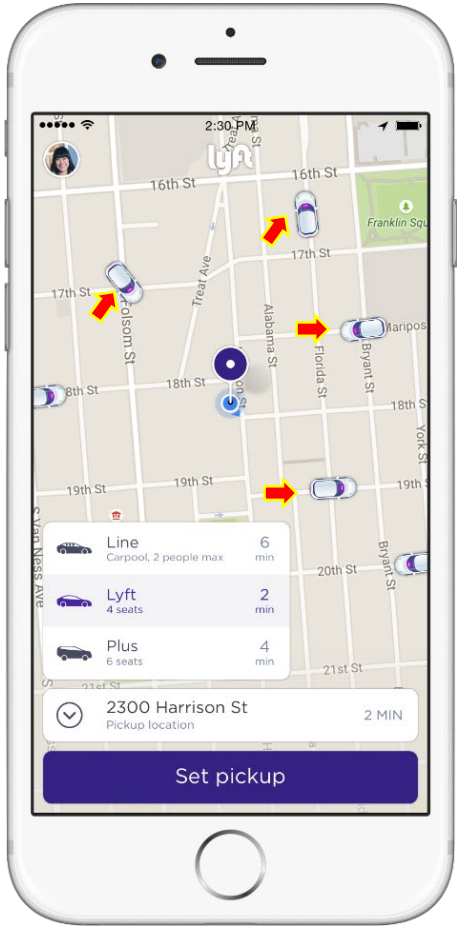



```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Suspicious ***Nearby Cars*** Functionality



Why Moving Smooth?

- Car ID
- GPS + Timestamp
- ***Tracking Drivers***

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Suspicious ***Nearby Cars*** Functionality

Tracking Drivers

- Reusing Nearby Cars feature

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"   : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"   : "6F09E2AA",
            ...
        },
        ...
}
```

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

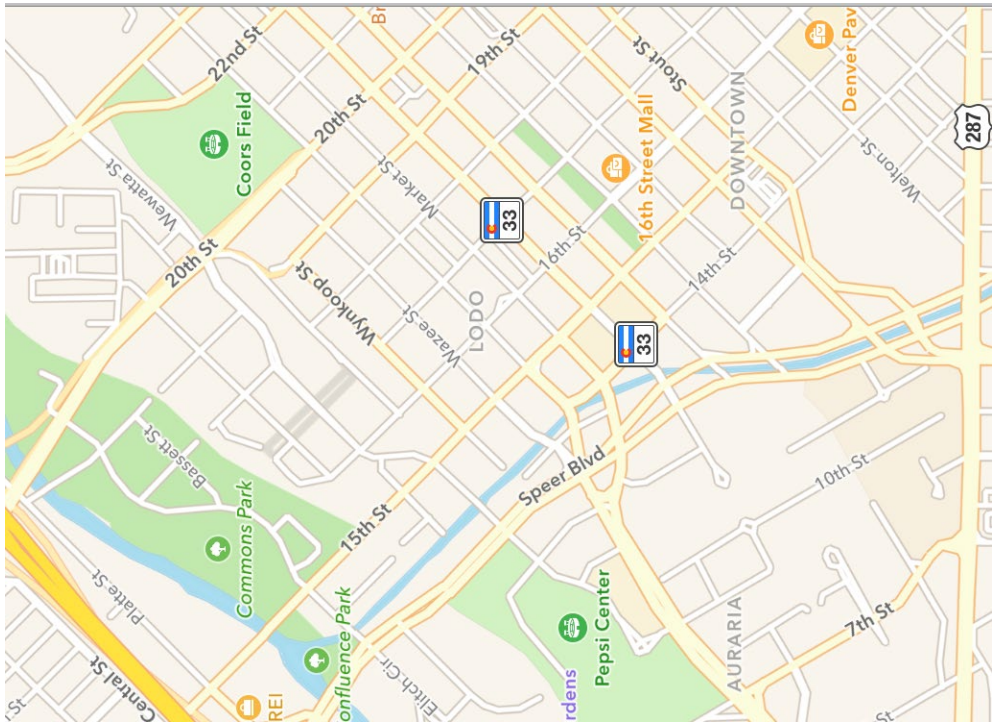
Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors

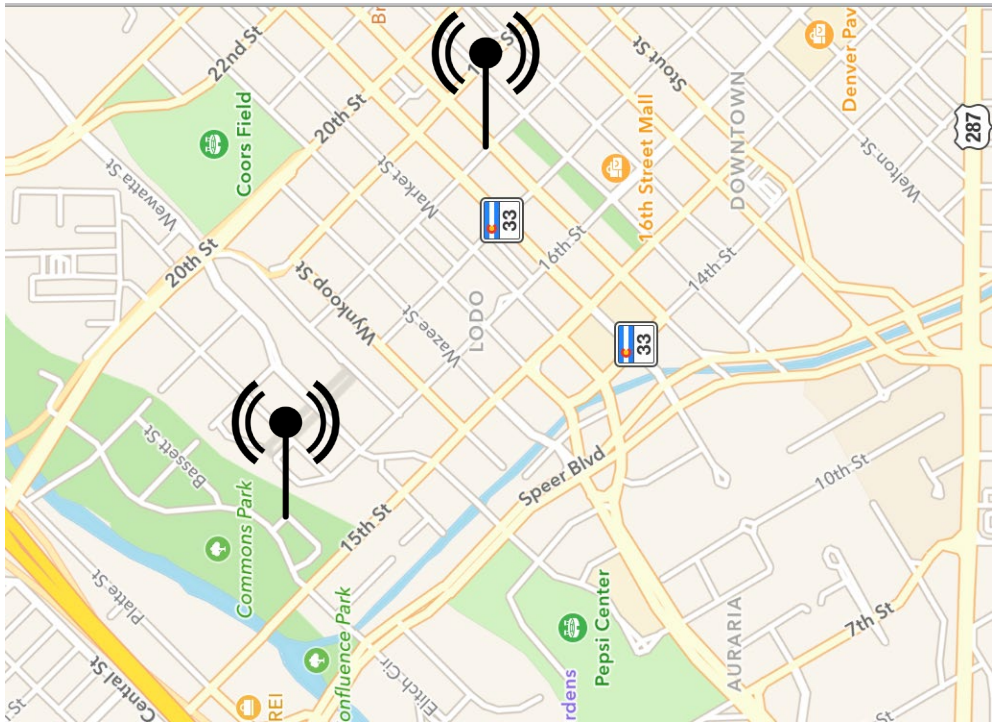
```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```



Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors

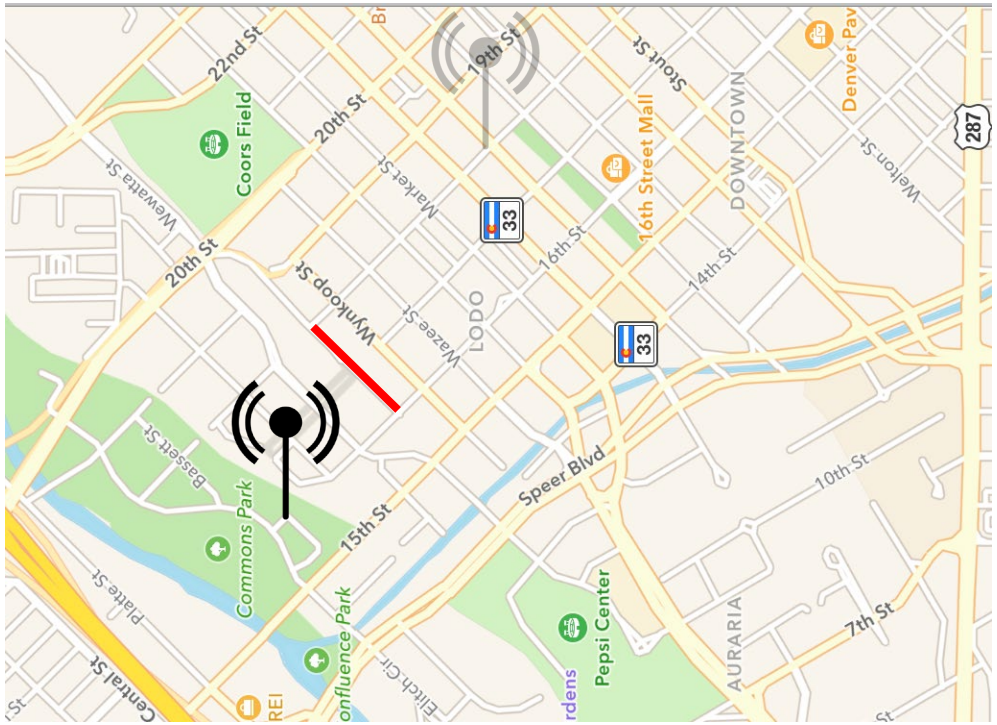
```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```



Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors

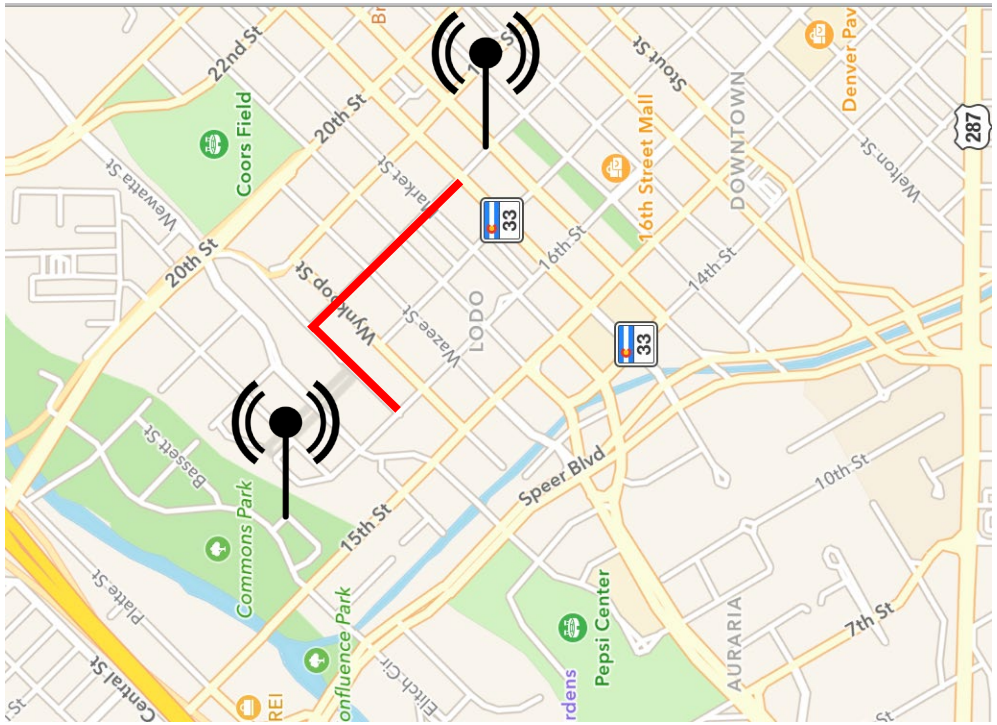
```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```



Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
...
```



Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"   : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                },  {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"   : "6F09E2AA",
            ...
        },
        ...
}
```

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period
  - Using ID to recognize each driver

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period
  - Using ID to recognize each driver
  - Uncovering driver movements by linking GPS with timestamp

```
GET /nearby-cars?lat=33.7114&lng=151.1321
HTTP/1.1
…
```

```
HTTP/1.1 200 OK
Content-type: application/json
...

{
    "cars": [
        {
            "id"    : "509AE827",
            "positions": [
                {
                    "GPS": "-33.7100 / 151.1342",
                    "t"  : "15259620050000"
                }, {
                    "GPS": "-33.7300 / 151.1200",
                    "t"  : "15259620060000"
                },
                ...
        }, {
            "id"    : "6F09E2AA",
            ...
        },
        ...
}
```

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period
  - Using ID to recognize each driver
  - Uncovering driver movements by linking GPS with timestamp

- Reverse engineering of Nearby Cars request

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period
  - Using ID to recognize each driver
  - Uncovering driver movements by linking GPS with timestamp

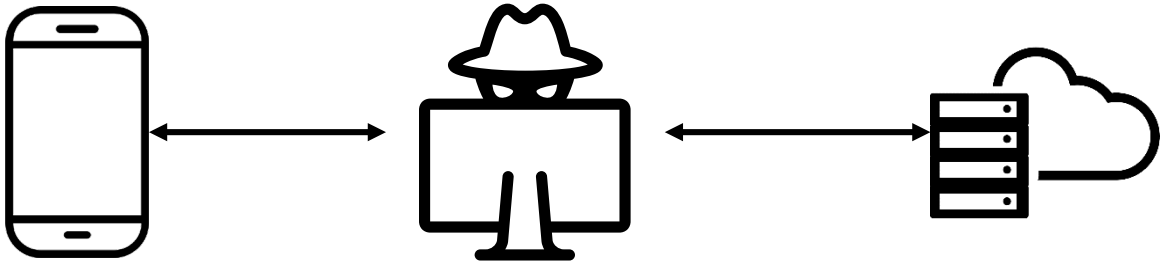
- Reverse engineering of Nearby Cars request
- Long period of data harvesting

Tracking Drivers

- Reusing Nearby Cars feature
  - Changing GPS to place monitors
  - Harvesting data over a long period
  - Using ID to recognize each driver
  - Uncovering driver movements by linking GPS with timestamp

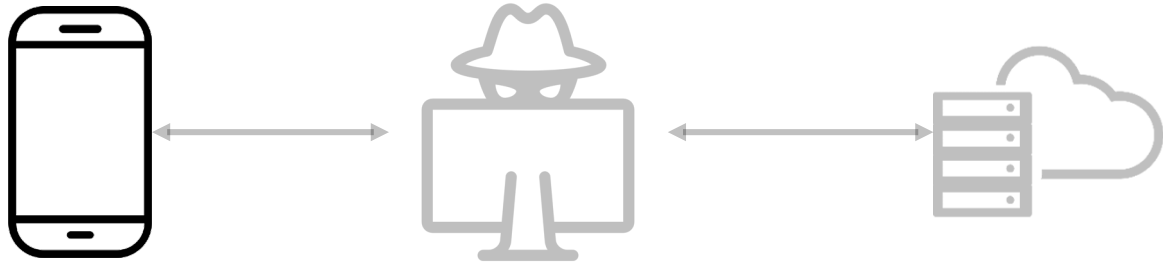
- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID



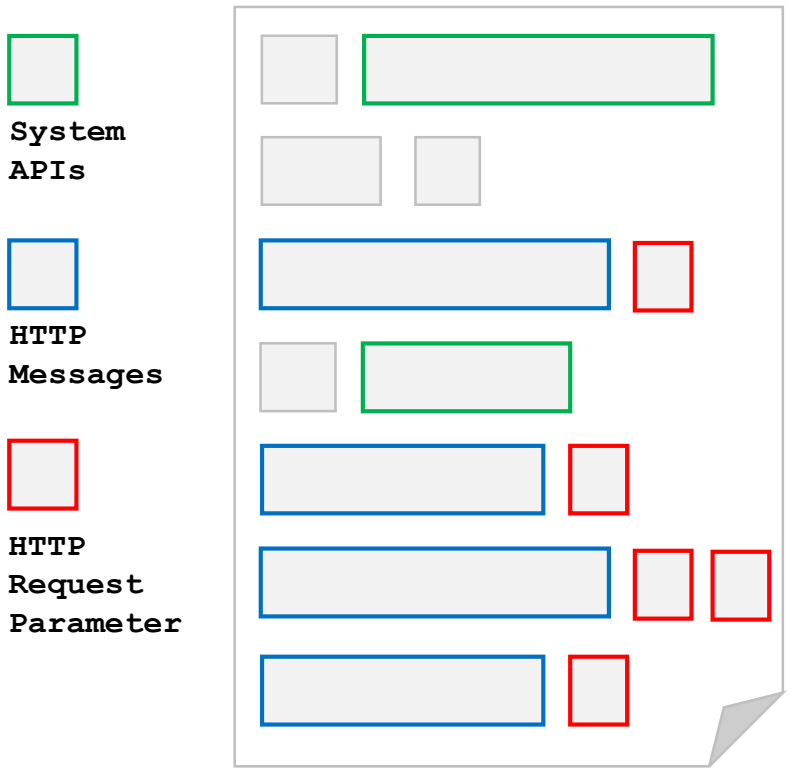
- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Setting up a proxy?



- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs



- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs

```
GET /v1/nearby-drivers-pickup-etas?
lat=10.10&lng=-10.10 HTTP/1.1
Authorization: Bearer dmGtpMx1qCKeA
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "nearby_drivers":[
        {
            ...
            "driver":{
               ...
            },
            "locations":[
                {
                   "lat":10.10,
                   "lng":-10.10,
                   "recorded_at_ms":1234
                },
                ...
            ]
        },
        {
            ...
            "driver":{
               ...
            },
        ...
}
```

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs

```
GET /v1/nearby-drivers-pickup-etas?
lat=10.10&lng=-10.10 HTTP/1.1
Authorization: Bearer dmGtpMx1qCKeA
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "nearby_drivers":[
        {
            ...
            "driver":{
               ...
            },
            "locations":[
                {
                    "lat":10.10,
                    "lng":-10.10,
                    "recorded_at_ms":1234
                },
                ...
            ]
        },
        {
            ...
            "driver":{
               ...
            },
        ...
}
```

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs

```
GET /v1/nearby-drivers-pickup-etas?
lat=10.10&lng=-10.10 HTTP/1.1
Authorization: Bearer dmGtpMx1qCKeA
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "nearby_drivers":[
        {
            ...
            "driver":{
               ...
            },
            "locations":[
                {
                   "lat":10.10,
                   "lng":-10.10,
                   "recorded_at_ms":1234
                },
                ...
            ]
        },
        {
            ...
            "driver":{
               ...
            },
        ...
}
```

```
POST /oauth2/access_token  HTTP/1.1
grant_type = ***Aphone &
phone_number = 123 & phone_code = 111
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "access_token": "eHdNsgsNvREH1",
    "expires_in": 86400,
    "refresh_token": "bEwazc0wcI",
}
```

```
POST /oauth2/access_token  HTTP/1.1
grant_type=refresh_token &
refresh_token=bEwazc0wcI
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "access_token": "dmGtpMx1qCKeA",
    "expires_in": 86400,
    "refresh_token": "3Rva2VuIiw",
}
```

```
GET /v1/nearby-drivers-pickup-etas?
lat=10.10&lng=-10.10 HTTP/1.1
Authorization: Bearer dmGtpMx1qCKeA
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "nearby_drivers":[
        {
            ...
            "driver":{
               ...
            },
            "locations":[
                {
                   "lat":10.10,
                   "lng":-10.10,
                   "recorded_at_ms":1234
                },
                ...
            ]
        },
        {
            ...
            "driver":{
               ...
            },
        ...
}
```

```
POST /oauth2/access_token  HTTP/1.1
grant_type = ***Aphone &
phone_number = 123 & phone_code = 111
```

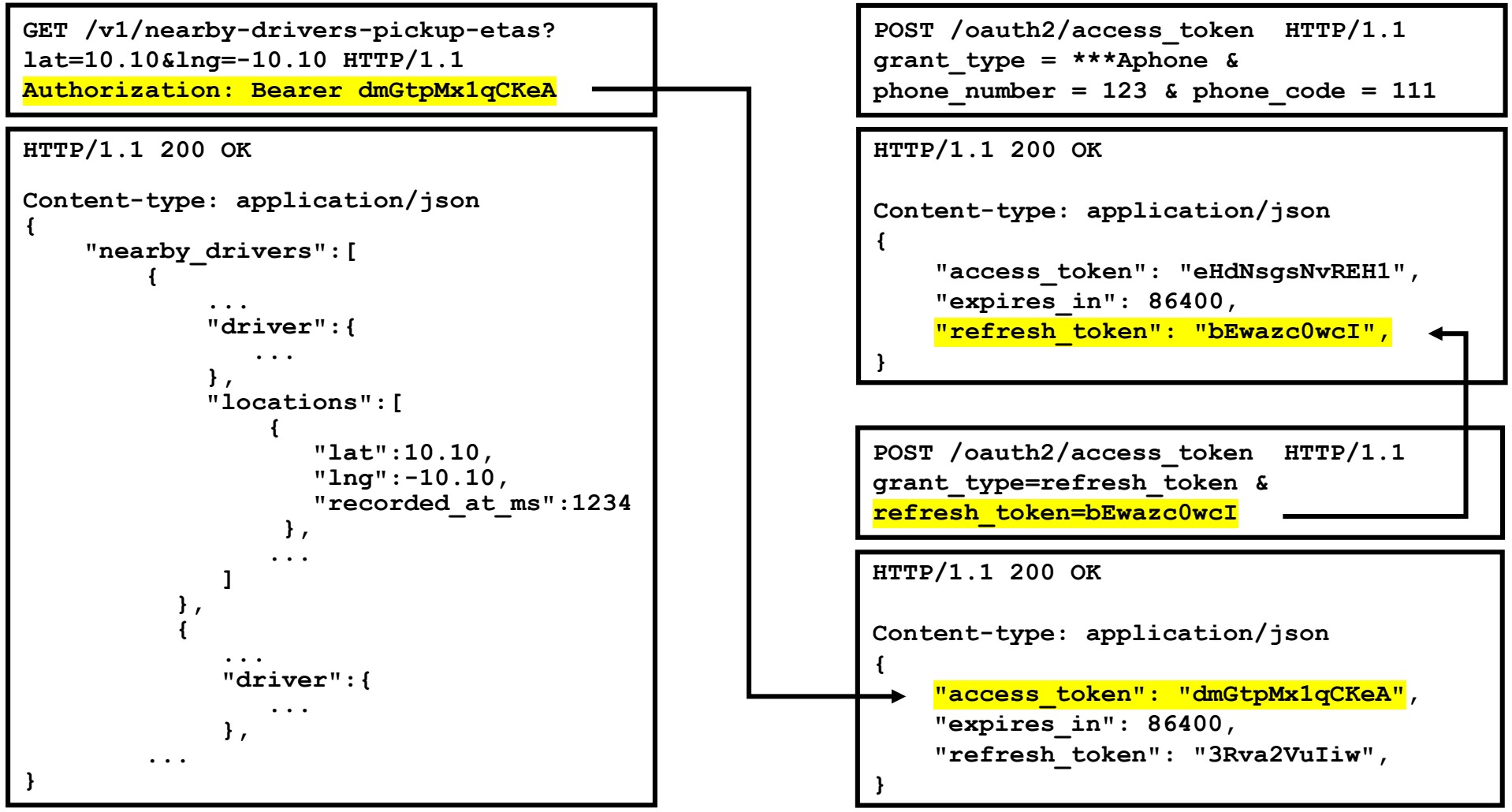
```
HTTP/1.1 200 OK

Content-type: application/json
{
    "access_token": "eHdNsgsNvREH1",
    "expires_in": 86400,
    "refresh_token": "bEwazc0wcI",
}
```

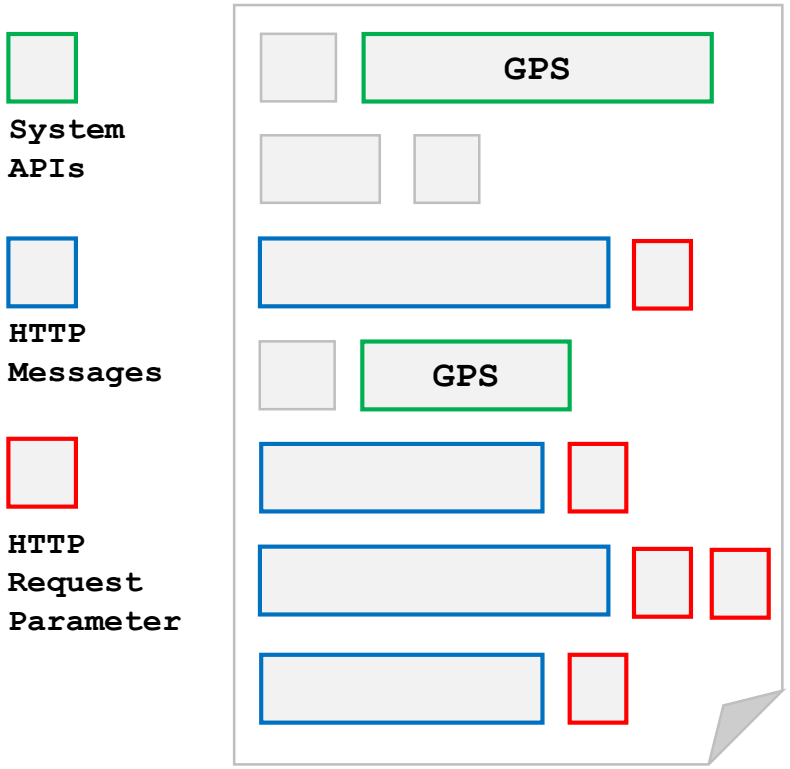
```
POST /oauth2/access_token  HTTP/1.1
grant_type=refresh_token &
refresh_token=bEwazc0wcI
```

```
HTTP/1.1 200 OK

Content-type: application/json
{
    "access_token": "dmGtpMx1qCKeA",
    "expires_in": 86400,
    "refresh_token": "3Rva2VuIiw",
}
```

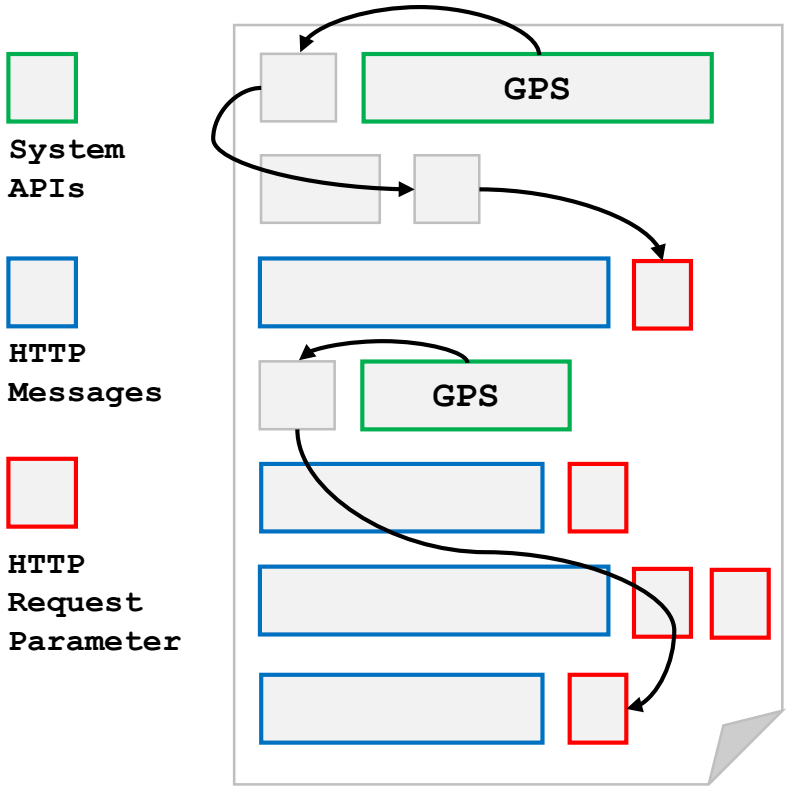


Dependencies



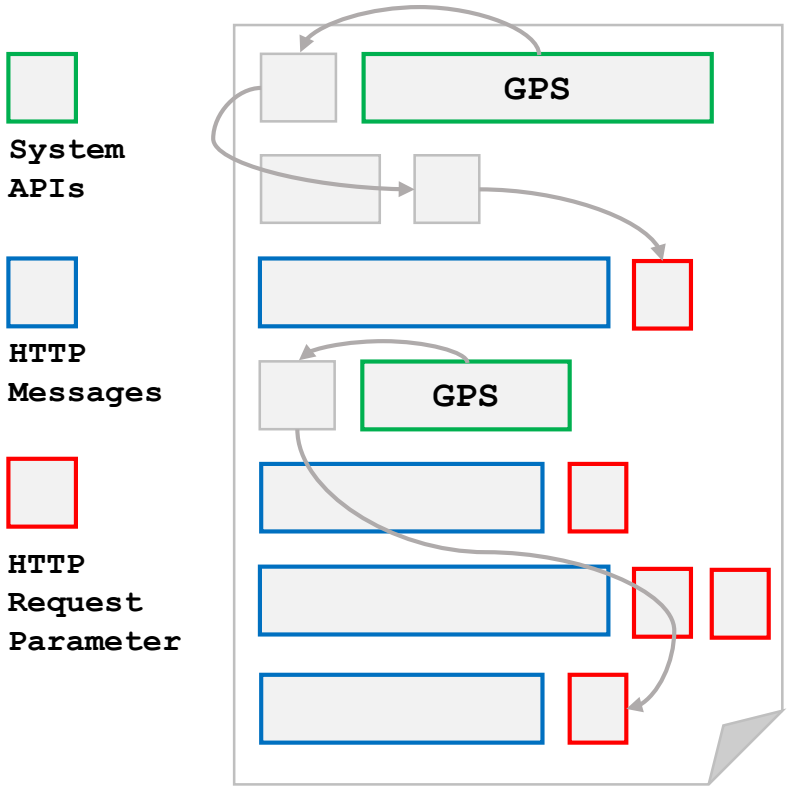
- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs



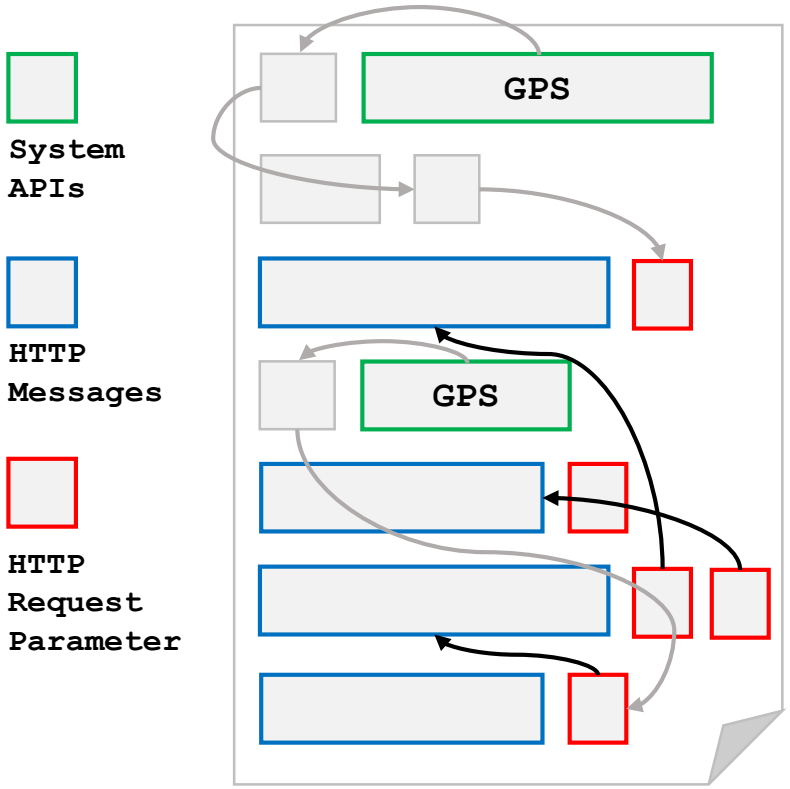
- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Dynamic hooking system APIs



- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Pinpoint Nearby Cars Request



- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Identifying Dependencies

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Rate Limiting

- Req / s
- # IP W/ Same Session

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Rate Limiting	• Req / s • # IP W/ Same Session
Session Management	• Authentication • Session Lifespan

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Rate Limiting	• Req / s • # IP W/ Same Session
Session Management	• Authentication • Session Lifespan
Anti-GPS Spoofing	• Distant Places

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Rate Limiting	• Req / s • # IP W/ Same Session
Session Management	• Authentication • Session Lifespan
Anti-GPS Spoofing	• Distant Places
Anonymization	• Identifier Lifespan • Personal Identifiable Information

- Reverse engineering of Nearby Cars request
- Long period of data harvesting
- Permanent linkability of ID

Installs	Service
100,000,000+	Uber
10,000,000+	Easy
10,000,000+	Gett
10,000,000+	Lyft
5,000,000+	My Taxi
5,000,000+	Taxify
1,000,000+	BiTaksi
1,000,000+	Heetch
500,000+	Jeeny
100,000+	Flywheel
100,000+	GoCatch
100,000+	miCab
100,000+	RideAustin
100,000+	Ztrip
50,000+	eCab
10,000+	GroundLink
10,000+	HelloCabs
10,000+	Ride LA
10,000+	Bounce
5,000+	DC Taxi Rider

Installs	Service	Reqs/s	Diff IPs	Authen	Sn Lifespan	Anti-GPS	ID Lifespan	PII
100,000,000+	Uber	●	○	●	∞	○	∞	●
10,000,000+	Easy	-	○	○	∞	○	∞	●
10,000,000+	Gett	-	○	●	∞	○	∞	●
10,000,000+	Lyft	●	○	●	24h	○	∞	○
5,000,000+	My Taxi	-	○	○	∞	○	20m	●
5,000,000+	Taxify	●	○	●	∞	○	∞	●
1,000,000+	BiTaksi	-	○	●	∞	○	∞	●
1,000,000+	Heetch	-	○	●	∞	○	∞	●
500,000+	Jeeny	-	○	○	∞	○	20m	●
100,000+	Flywheel	-	○	●	20m	○	10m	●
100,000+	GoCatch	-	○	●	∞	○	∞	●
100,000+	miCab	-	○	●	∞	○	∞	○
100,000+	RideAustin	-	○	●	∞	○	∞	●
100,000+	Ztrip	-	○	●	30m	○	∞	●
50,000+	eCab	●	○	○	∞	○	∞	●
10,000+	GroundLink	-	○	○	∞	○	∞	●
10,000+	HelloCabs	-	○	●	∞	○	∞	○
10,000+	Ride LA	-	○	○	∞	○	∞	○
10,000+	Bounce	-	○	●	∞	○	∞	○
5,000+	DC Taxi Rider	-	○	●	∞	○	∞	○

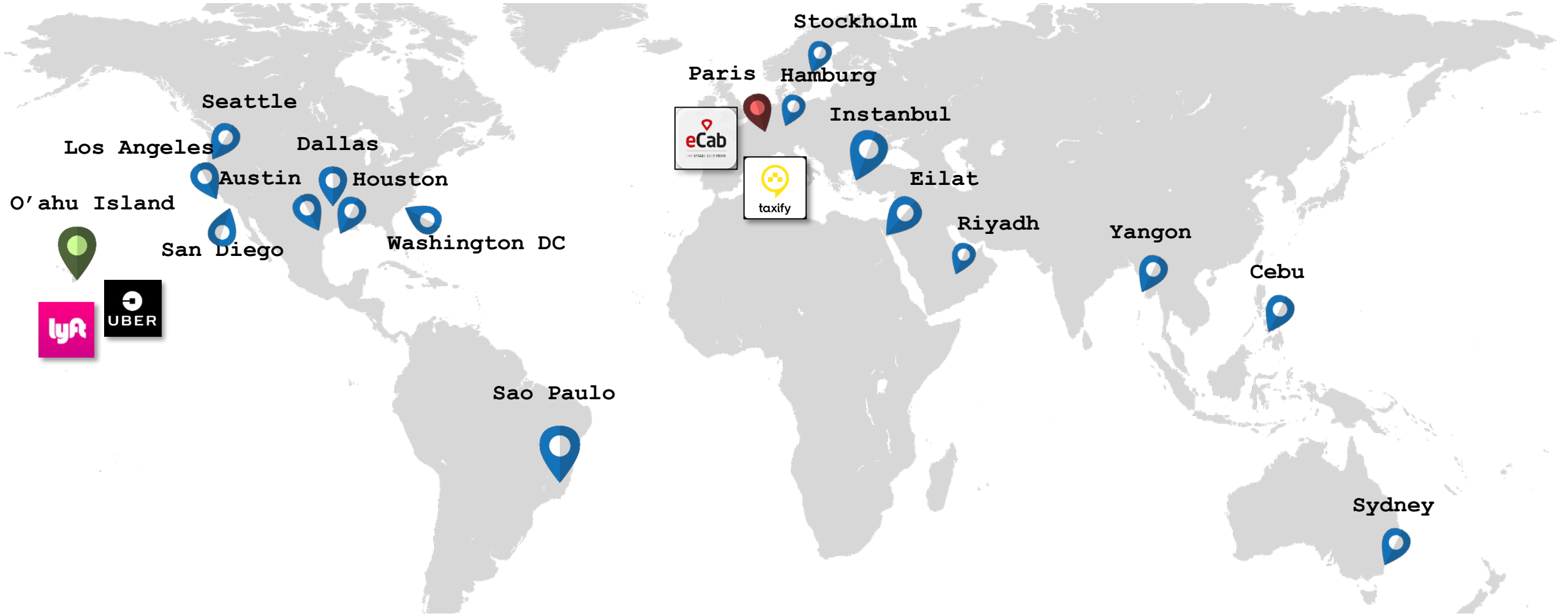
Installs	Service	Reqs/s	Diff IPs	Authen	Sn Lifespan	Anti-GPS	ID Lifespan	PII
100,000,000+	Uber	●	○	●	∞	○	∞	●
10,000,000+	Easy	-	○	○	∞	○	∞	●
10,000,000+	Gett	-	○	●	∞	○	∞	●
10,000,000+	Lyft	●	○	●	24h	○	∞	○
5,000,000+	My Taxi	-	○	○	∞	○	20m	●
5,000,000+	Taxify	●	○	●	∞	○	∞	●
1,000,000+	BiTaksi	-	○	●	∞	○	∞	●
1,000,000+	Heetch	-	○	●	∞	○	∞	●
500,000+	Jeeny	-	○	○	∞	○	20m	●
100,000+	Flywheel	-	○	●	20m	○	10m	●
100,000+	GoCatch	-	○	●	∞	○	∞	●
100,000+	miCab	-	○	●	∞	○	∞	○
100,000+	RideAustin	-	○	●	∞	○	∞	●
100,000+	Ztrip	-	○	●	30m	○	∞	●
50,000+	eCab	●	○	○	∞	○	∞	●
10,000+	GroundLink	-	○	○	∞	○	∞	●
10,000+	HelloCabs	-	○	●	∞	○	∞	○
10,000+	Ride LA	-	○	○	∞	○	∞	○
10,000+	Bounce	-	○	●	∞	○	∞	○
5,000+	DC Taxi Rider	-	○	●	∞	○	∞	○

Installs	Service	Reqs/s	Diff IPs	Authen	Sn Lifespan	Anti-GPS	ID Lifespan	PII
100,000,000+	Uber	●	○	●	∞	○	∞	●
10,000,000+	Easy	-	○	○	∞	○	∞	●
10,000,000+	Gett	-	○	●	∞	○	∞	●
10,000,000+	Lyft	●	○	●	24h	○	∞	○
5,000,000+	My Taxi	-	○	○	∞	○	20m	●
5,000,000+	Taxify	●	○	●	∞	○	∞	●
1,000,000+	BiTaksi	-	○	●	∞	○	∞	●
1,000,000+	Heetch	-	○	●	∞	○	∞	●
500,000+	Jeeny	-	○	○	∞	○	20m	●
100,000+	Flywheel	-	○	●	20m	○	10m	●
100,000+	GoCatch	-	○	●	∞	○	∞	●
100,000+	miCab	-	○	●	∞	○	∞	○
100,000+	RideAustin	-	○	●	∞	○	∞	●
100,000+	Ztrip	-	○	●	30m	○	∞	●
50,000+	eCab	●	○	○	∞	○	∞	●
10,000+	GroundLink	-	○	○	∞	○	∞	●
10,000+	HelloCabs	-	○	●	∞	○	∞	○
10,000+	Ride LA	-	○	○	∞	○	∞	○
10,000+	Bounce	-	○	●	∞	○	∞	○
5,000+	DC Taxi Rider	-	○	●	∞	○	∞	○

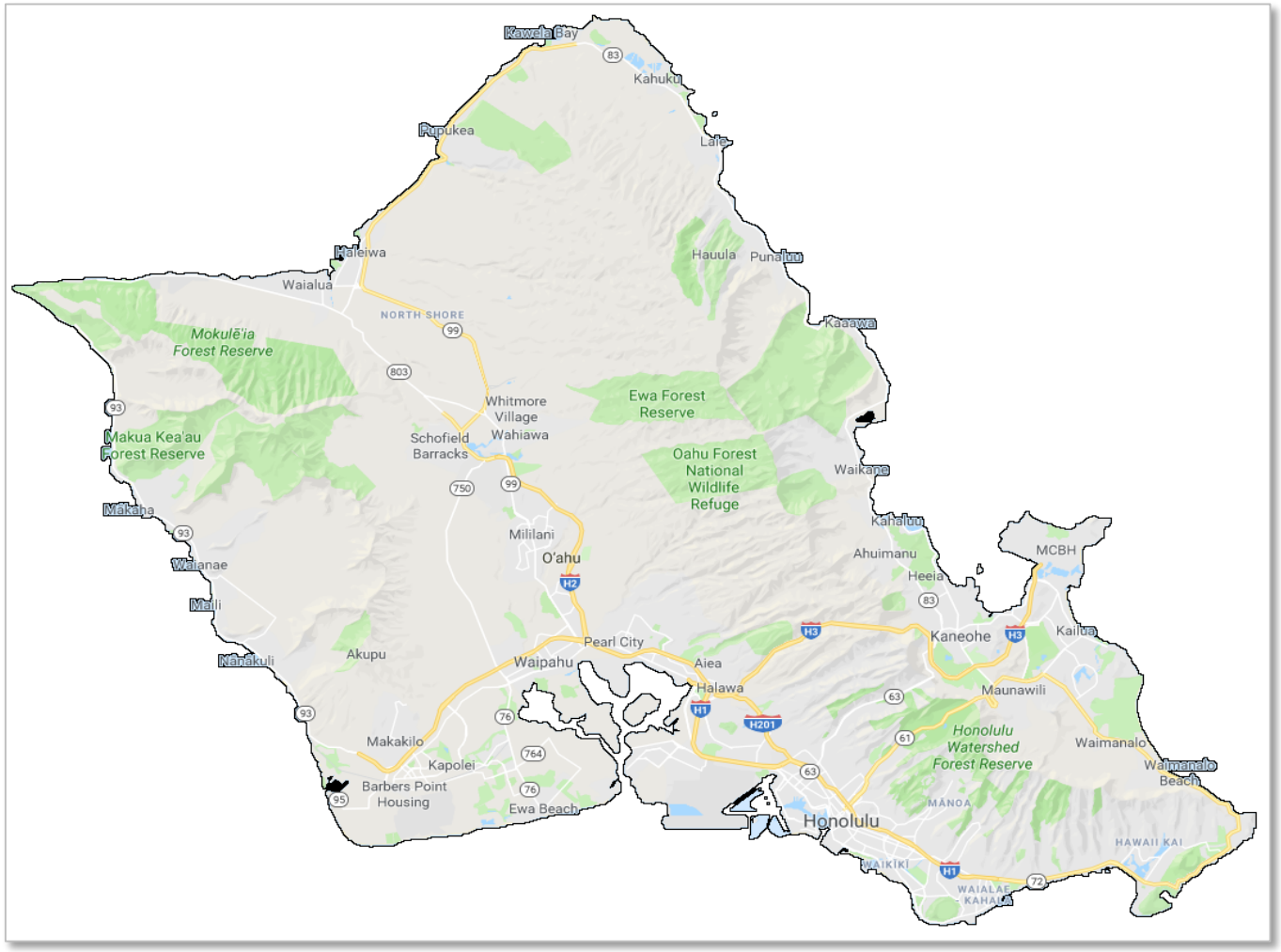
Installs	Service	Reqs/s	Diff IPs	Authen	Sn Lifespan	Anti-GPS	ID Lifespan	PII
100,000,000+	Uber	●	○	●	∞	○	∞	●
10,000,000+	Easy	-	○	○	∞	○	∞	●
10,000,000+	Gett	-	○	●	∞	○	∞	●
10,000,000+	Lyft	●	○	●	24h	○	∞	○
5,000,000+	My Taxi	-	○	○	∞	○	20m	●
5,000,000+	Taxify	●	○	●	∞	○	∞	●
1,000,000+	BiTaksi	-	○	●	∞	○	∞	●
1,000,000+	Heetch	-	○	●	∞	○	∞	●
500,000+	Jeeny	-	○	○	∞	○	20m	●
100,000+	Flywheel	-	○	●	20m	○	10m	●
100,000+	GoCatch	-	○	●	∞	○	∞	●
100,000+	miCab	-	○	●	∞	○	∞	○
100,000+	RideAustin	-	○	●	∞	○	∞	●
100,000+	Ztrip	-	○	●	30m	○	∞	●
50,000+	eCab	●	○	○	∞	○	∞	●
10,000+	GroundLink	-	○	○	∞	○	∞	●
10,000+	HelloCabs	-	○	●	∞	○	∞	○
10,000+	Ride LA	-	○	○	∞	○	∞	○
10,000+	Bounce	-	○	●	∞	○	∞	○
5,000+	DC Taxi Rider	-	○	●	∞	○	∞	○

Selected Cities

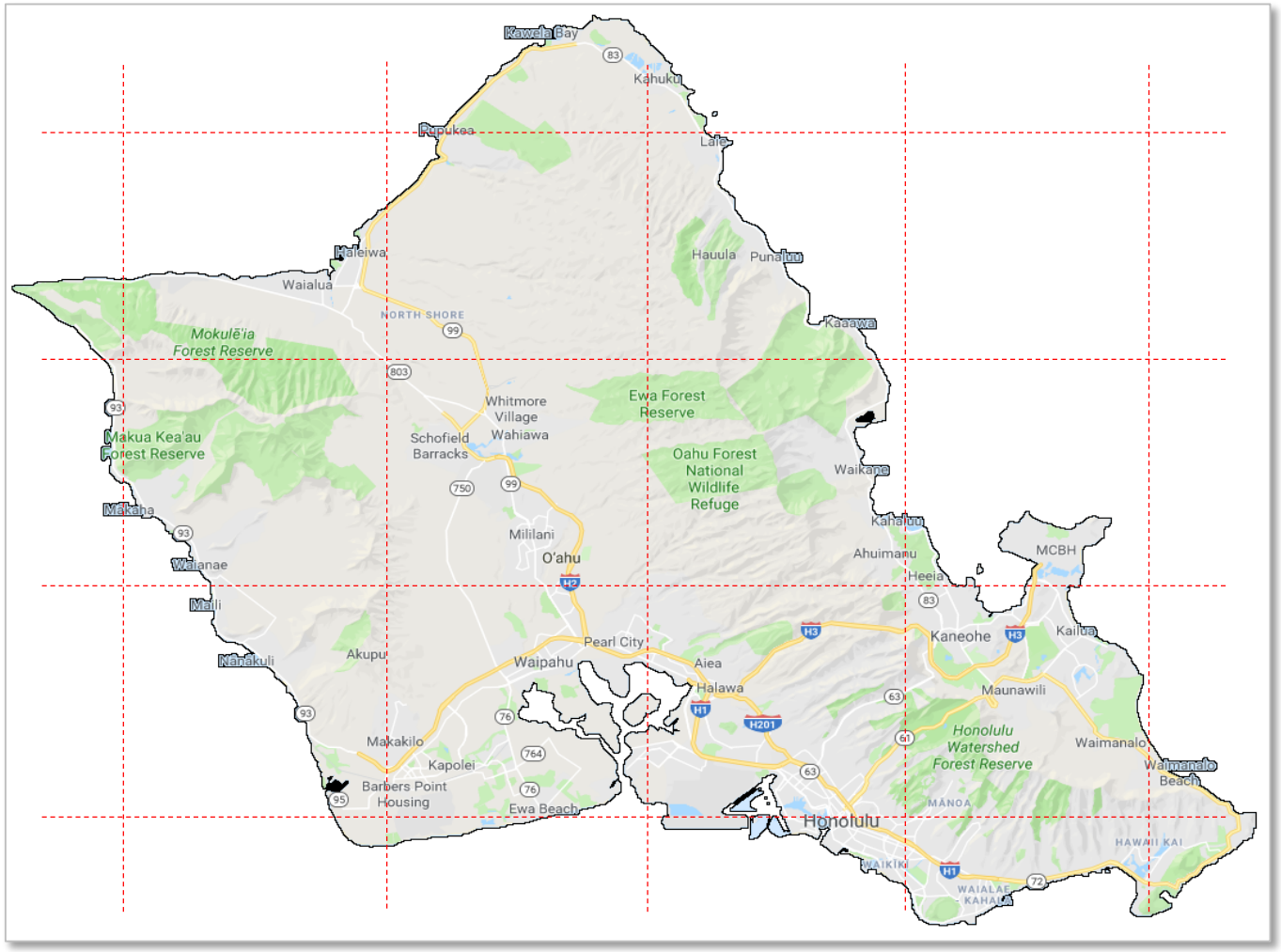
Selected Cities



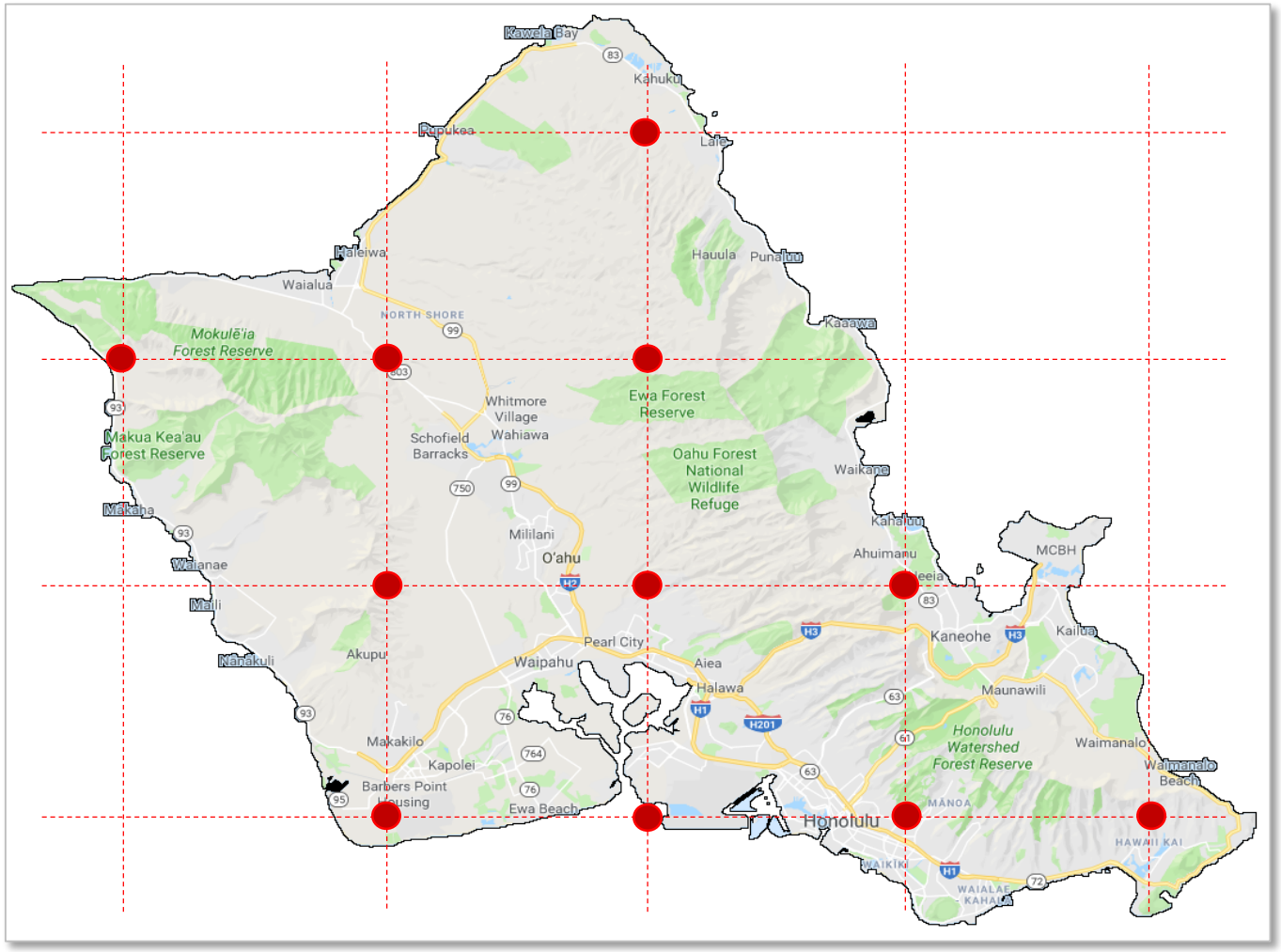
Selected Cities



Placing Monitors



Placing Monitors



Placing Monitors

Private Information

- Movements

Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Private Information

- ***PII***
- Movements
- Working Patterns
- Appeared Locations

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Service	Direct PII
Lyft	Driver avatar
HelloCabs	Name, Phone number
Ride LA	Name, Phone number
DC Taxi Rider	Name, Phone number, Email address
miCab	Account creating time, Account last update time, Device number, Hiring status
Bounce	Name, Date of birth, Driver Avatar, Phone number, Social Security Number, Driver License Number, Driver License Expiration Data, Home Address, Bank Account Number, Routing Number, Account Balance, Vehicle Insurance Details

Private Information

- ***PII***
- Movements
- Working Patterns
- Appeared Locations

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Service	Direct PII
Lyft	Driver avatar
HelloCabs	Name, Phone number
Ride LA	Name, Phone number
DC Taxi Rider	Name, Phone number, Email address
miCab	Account creating time, Account last update time, Device number, Hiring status
Bounce	Name, Date of birth, Driver Avatar, Phone number, Social Security Number, Driver License Number, Driver License Expiration Data, Home Address, Bank Account Number, Routing Number, Account Balance, Vehicle Insurance Details

Private Information

- PII
- Movements
- Working Patterns
- ***Appeared Locations***

Business Information

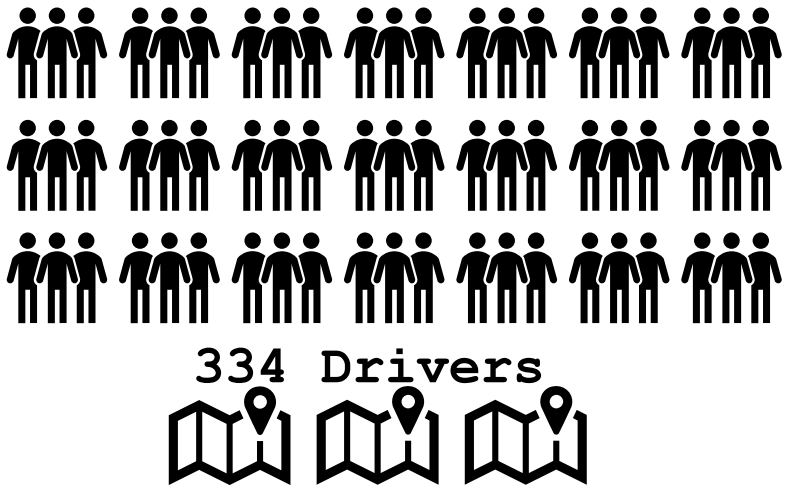
- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Private Information

- PII
- Movements
- Working Patterns
- ***Appeared Locations***

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

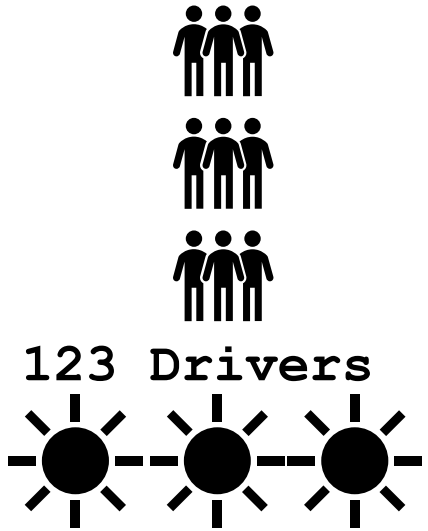


Private Information

- PII
- Movements
- Working Patterns
- ***Appeared Locations***

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

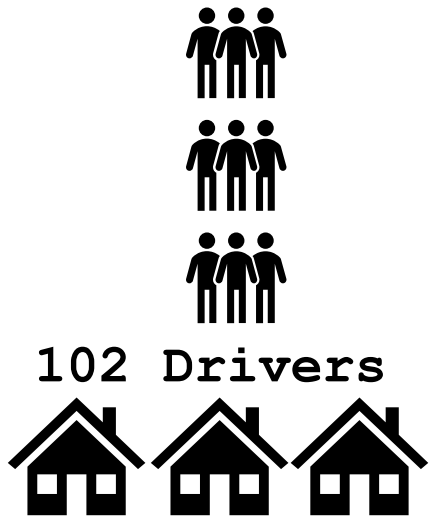


Private Information

- PII
- Movements
- Working Patterns
- ***Appeared Locations***

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

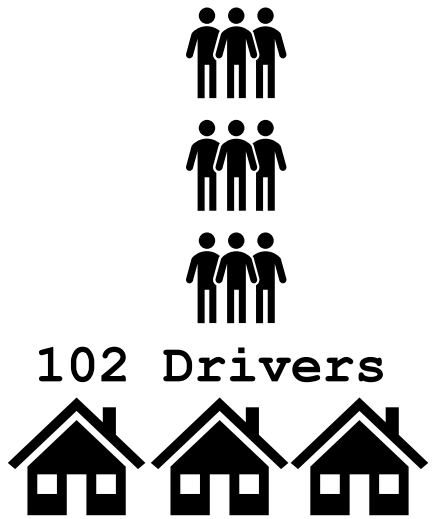


Private Information

- PII
- Movements
- Working Patterns
- ***Appeared Locations***

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance



Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

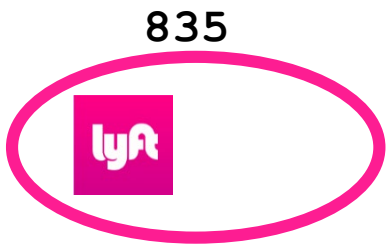
- ***Dual-apping Drivers***
- Driver Preference
- Number of Drivers
- Operation Performance

Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- ***Dual-apping Drivers***
- Driver Preference
- Number of Drivers
- Operation Performance

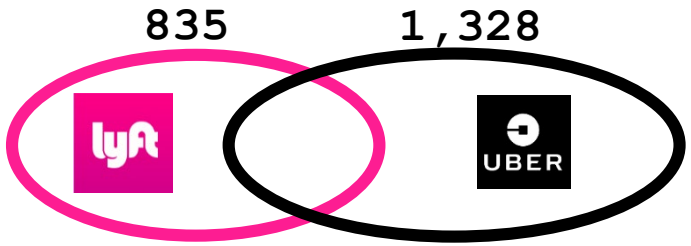


Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- ***Dual-apping Drivers***
- Driver Preference
- Number of Drivers
- Operation Performance

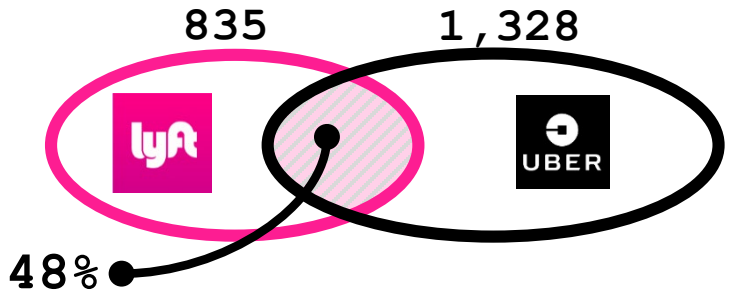


Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- ***Dual-apping Drivers***
- Driver Preference
- Number of Drivers
- Operation Performance

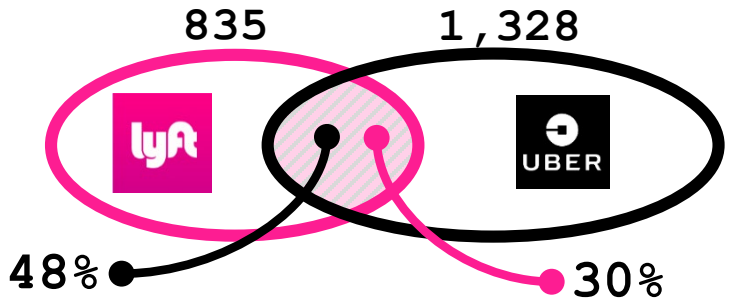


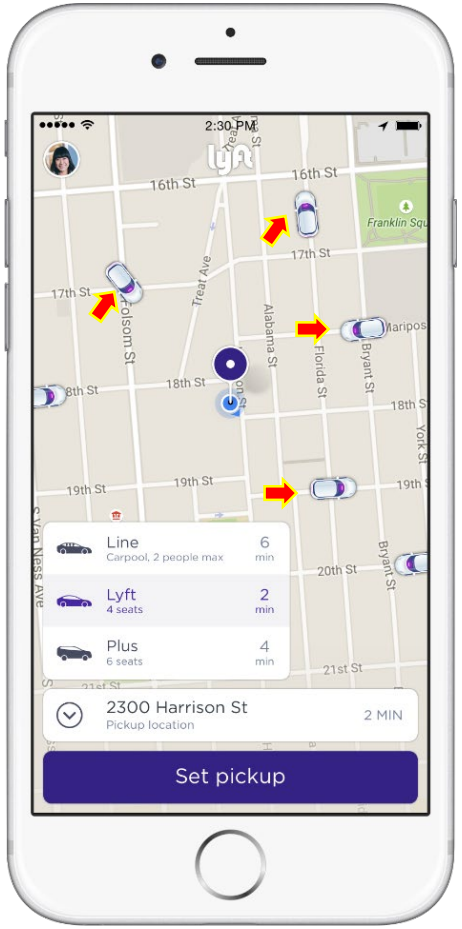
Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

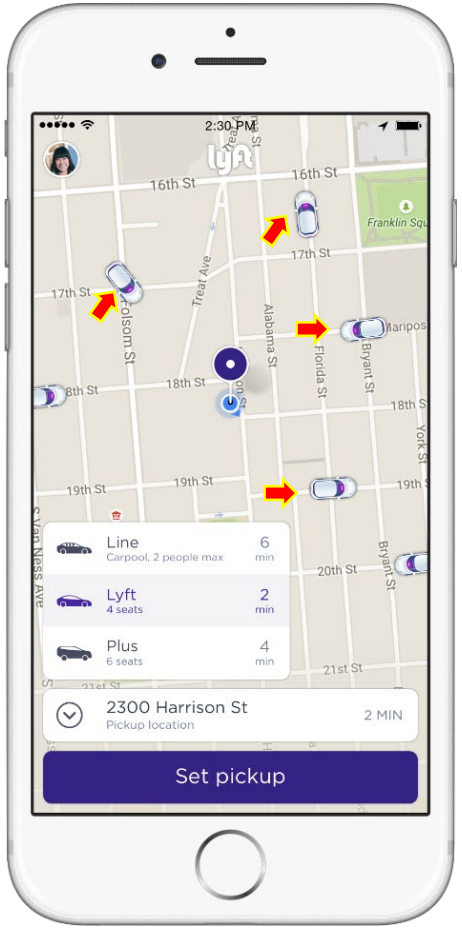
Business Information

- ***Dual-apping Drivers***
- Driver Preference
- Number of Drivers
- Operation Performance




2:30 PM
16th St
17th St
18th St
19th St
20th St
21st St
Folsom St
Treat Ave
Alabama St
Florida St
Bryant St
York St
Franklin Sq
Line
Carpool, 2 people max
6 min
Lyft
4 seats
2 min
Plus
6 seats
4 min
2300 Harrison St
Pickup location
2 MIN
Set pickup

Summary



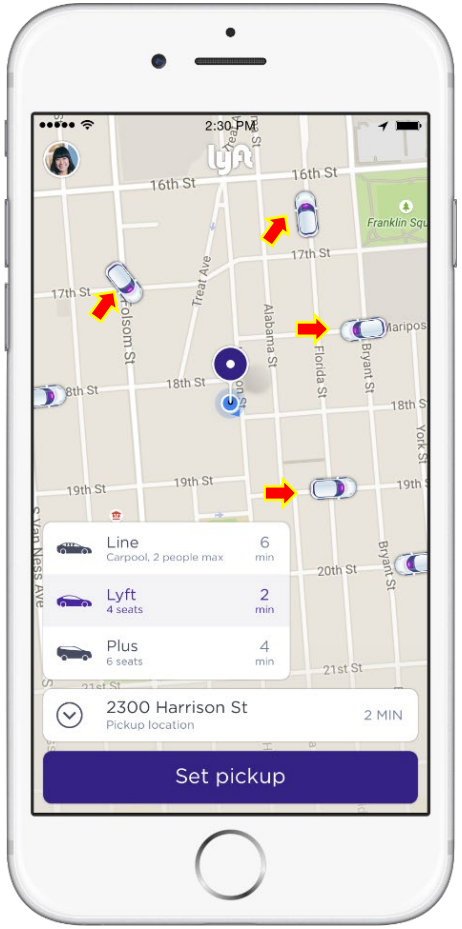
Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Summary



Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

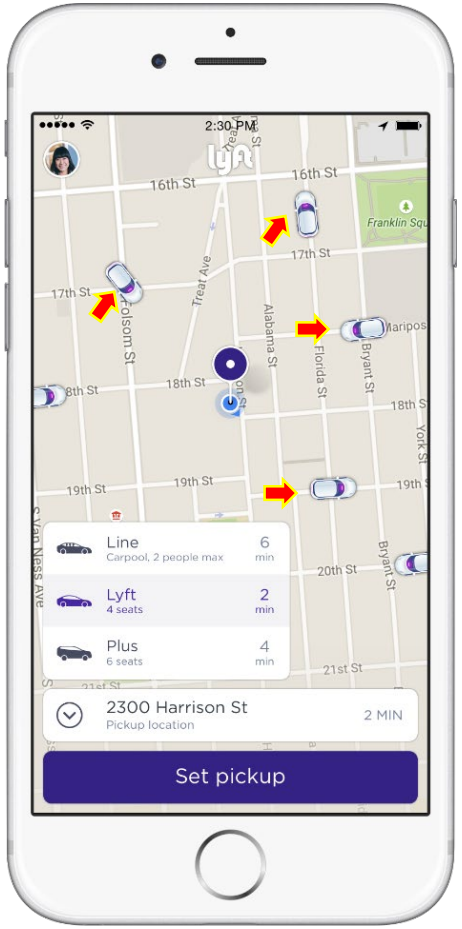
Business Information

- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Defense

- Removing PII
- Short-live Car ID
- Rate limiting & Anti-GPS Spoofing

Summary



Private Information

- PII
- Movements
- Working Patterns
- Appeared Locations

Business Information

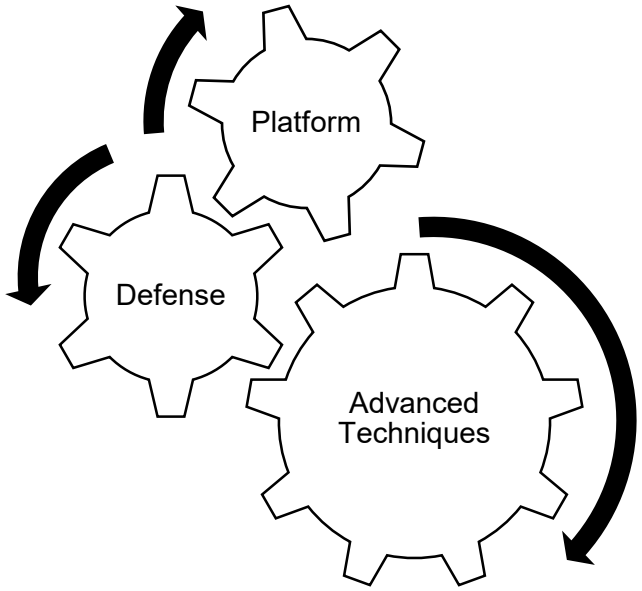
- Dual-apping Drivers
- Driver Preference
- Number of Drivers
- Operation Performance

Defense

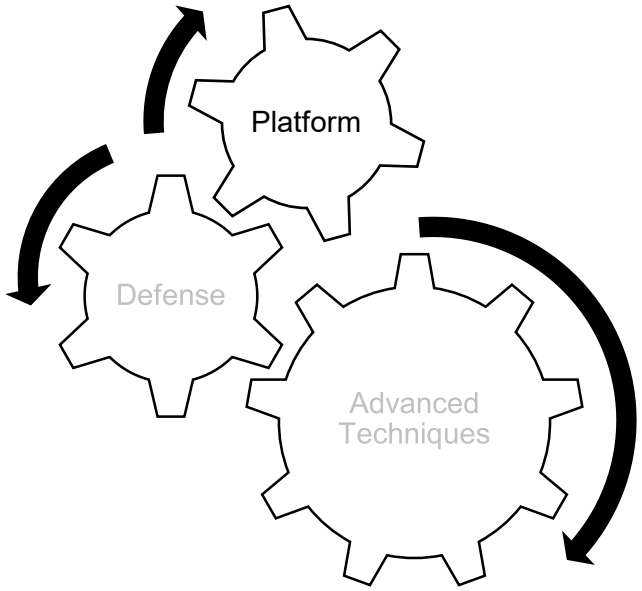
- Removing PII
- Short-live Car ID
- Rate limiting & Anti-GPS Spoofing

Bug Bounty: UBER lyft

Summary

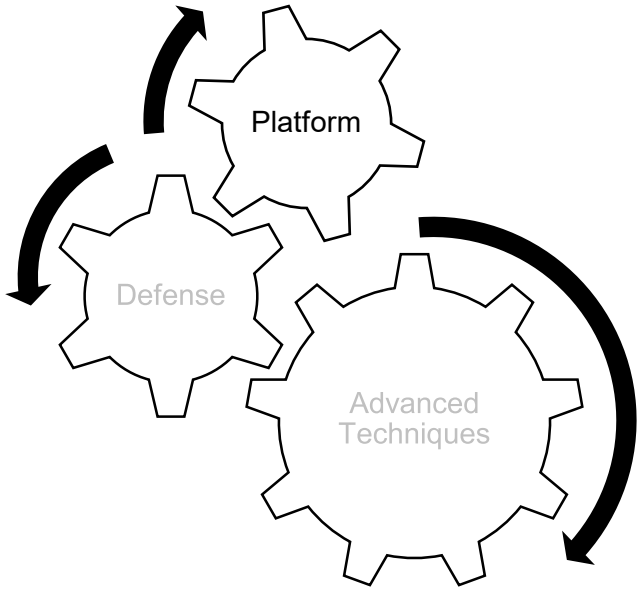


Future Directions



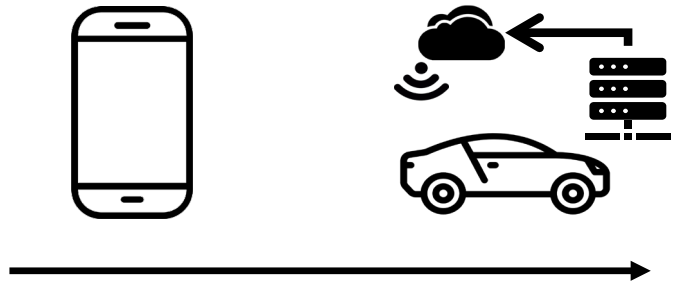
- Mobile platform:

Future Directions

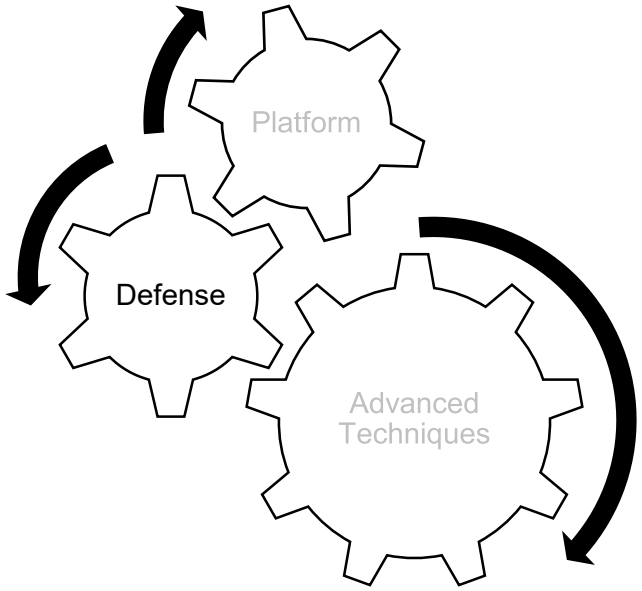


- Mobile platform:

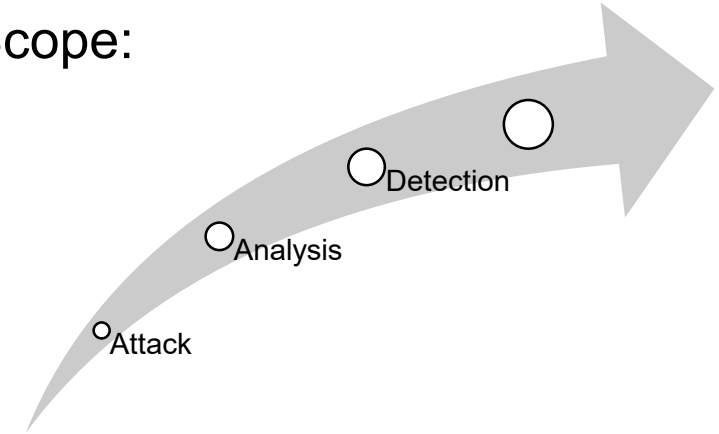
- Emerging Platform:



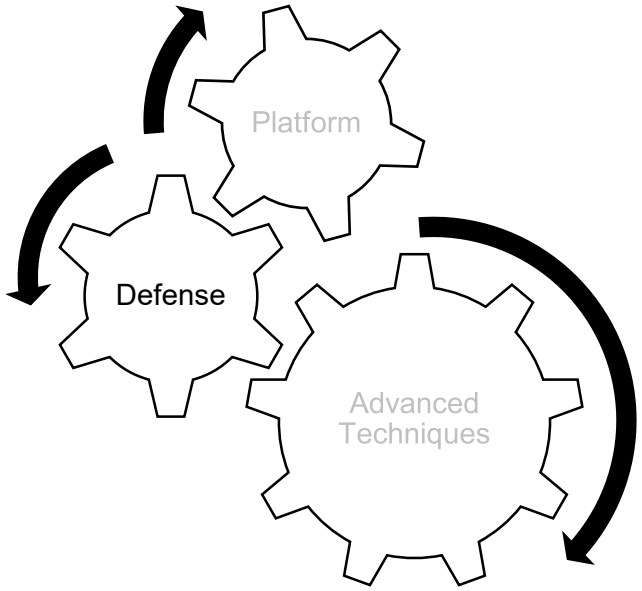
Future Directions



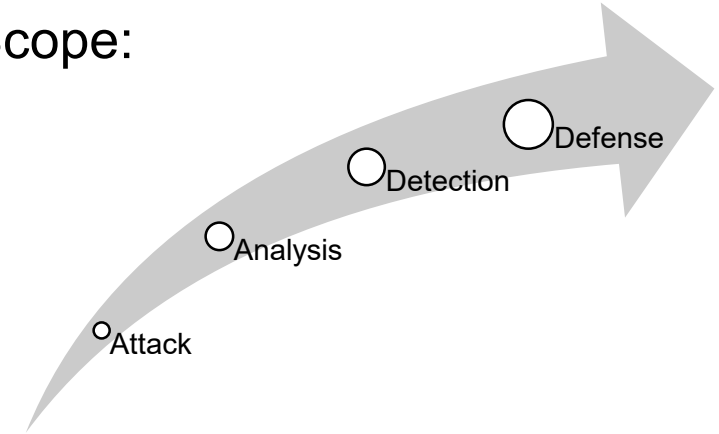
- Research Scope:



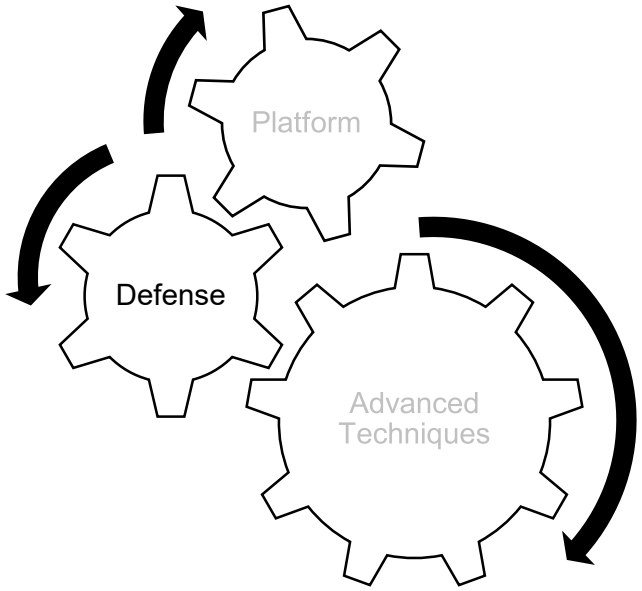
Future Directions



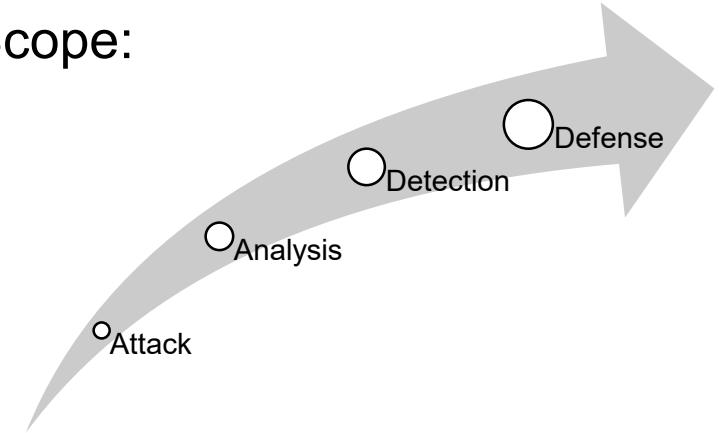
- Research Scope:



Future Directions



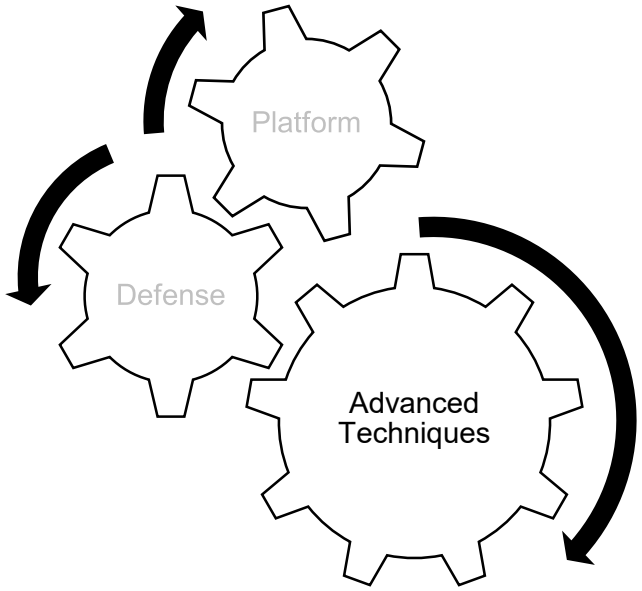
- Research Scope:



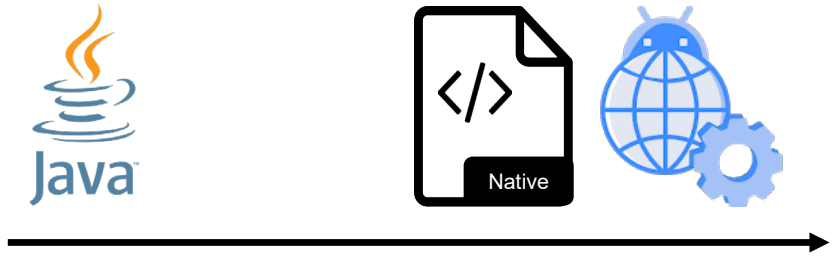
- Defense Mechanism:

arm TRUSTZONE

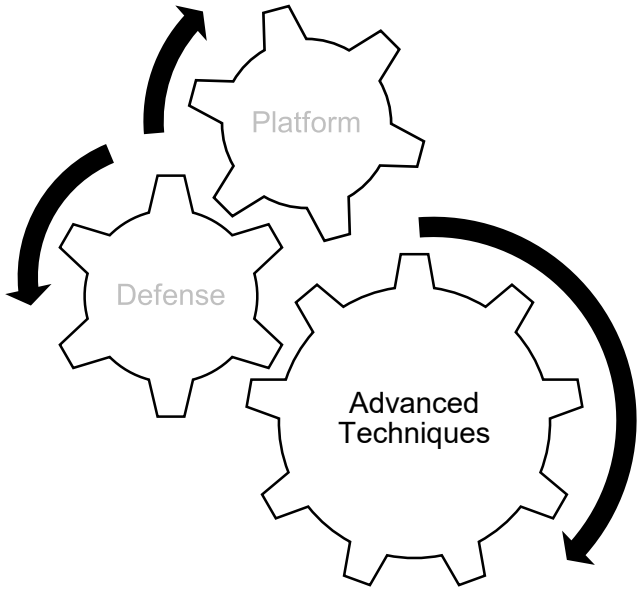
Future Directions



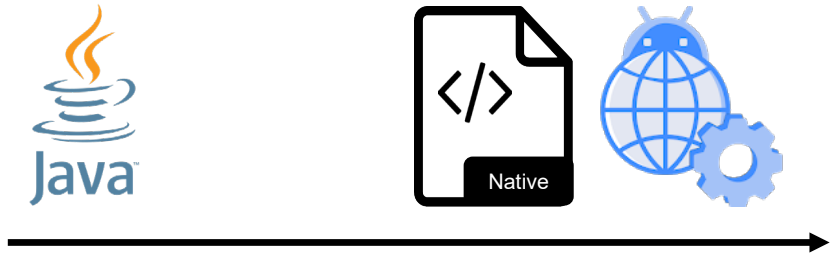
- Code Coverage



Future Directions

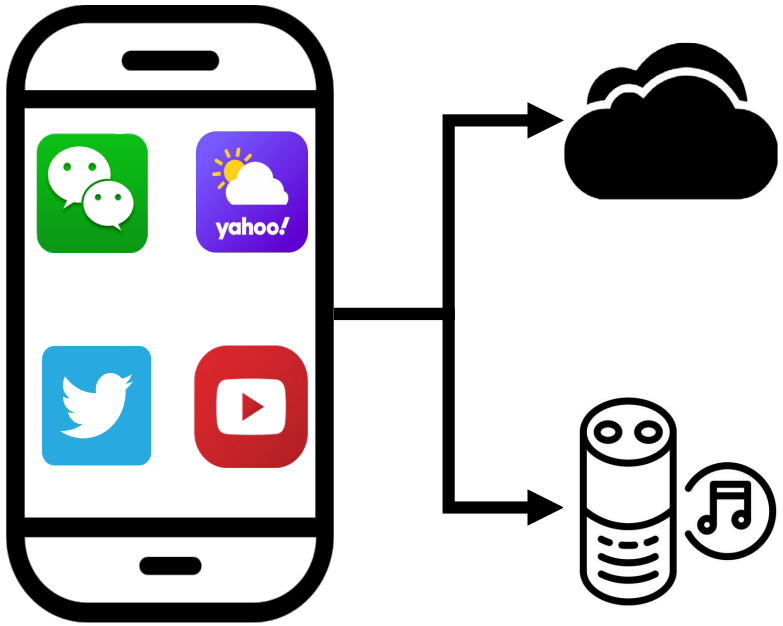


- Code Coverage

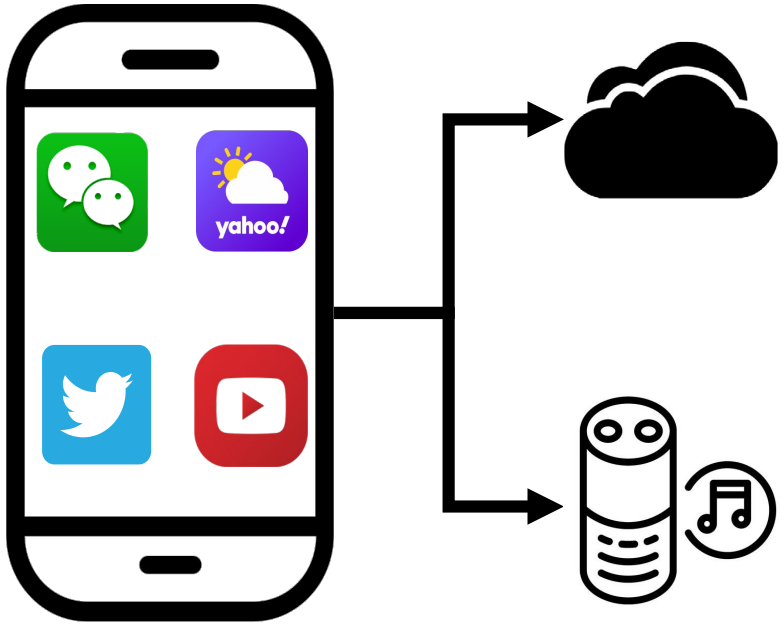


- Semantics Inference:

Future Directions

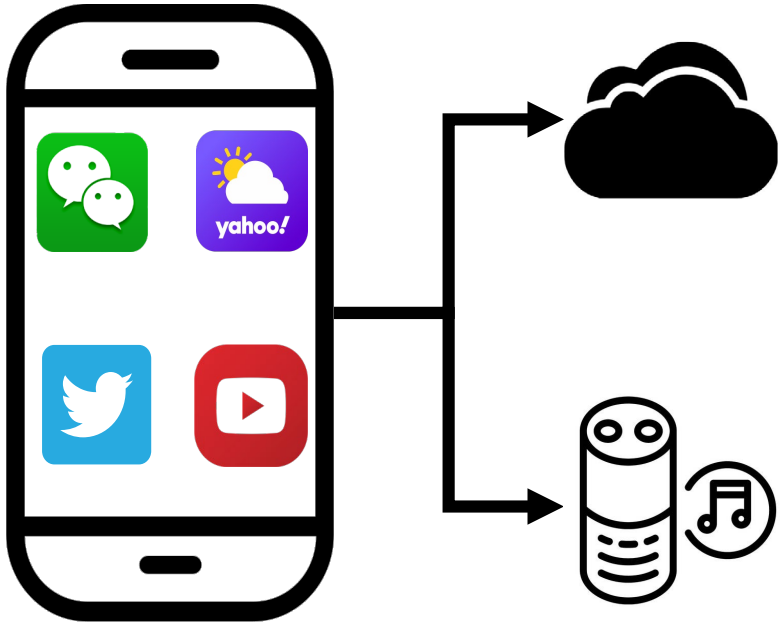


Conclusion

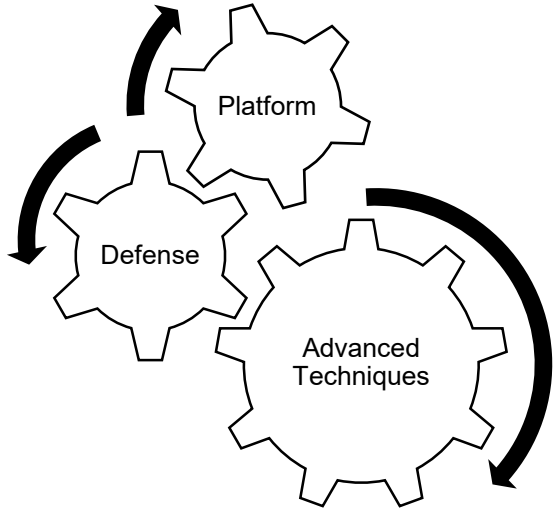


	Attack	Analysis	Detection
Single App	• NDSS'14		• **SP'20**
App to App			• USENIX Security'20
App to Cloud	• **NDSS'19** • NDSS'16	• SP'19 • USENIX Security'19	• CCS'17
App to Peripheral	• USENIX Security'20		• NDSS'20 • NDSS'18

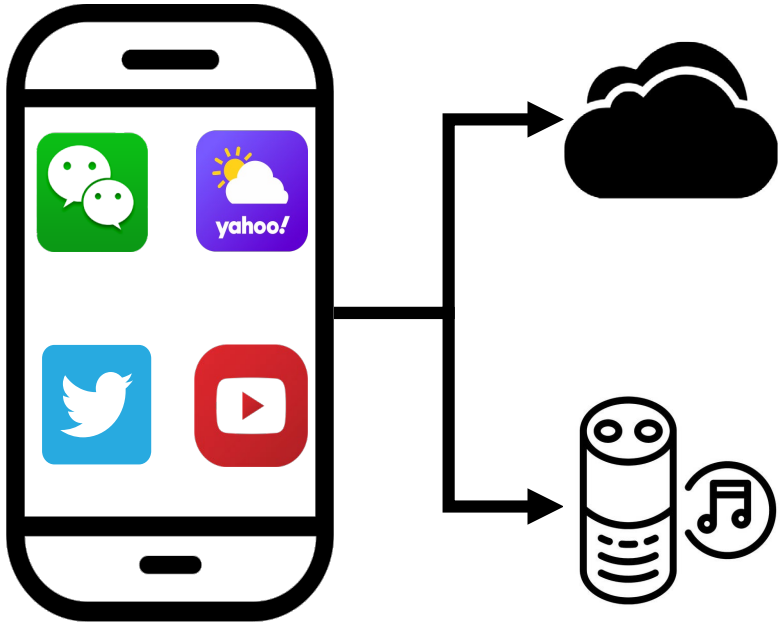
Conclusion



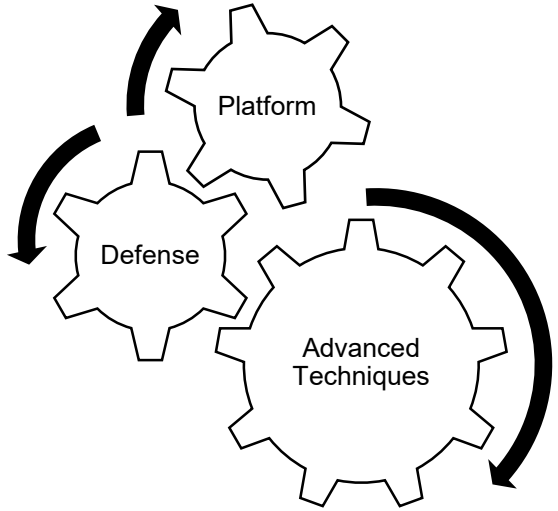
	Attack	Analysis	Detection
Single App	• NDSS'14		• **SP'20**
App to App			• USENIX Security'20
App to Cloud	• **NDSS'19** • NDSS'16	• SP'19 • USENIX Security'19	• CCS'17
App to Peripheral	• USENIX Security'20		• NDSS'20 • NDSS'18



Conclusion



	Attack	Analysis	Detection
Single App	• NDSS'14		• **SP'20**
App to App			• USENIX Security'20
App to Cloud	• **NDSS'19** • NDSS'16	• SP'19 • USENIX Security'19	• CCS'17
App to Peripheral	• USENIX Security'20		• NDSS'20 • NDSS'18



Acknowledgement:

Qingchuan Zhao, Chaoshun Zuo, Dolan-Gavitt Brendan, and Giancarlo Pellegrino

Unveiling Insecure and Privacy-Risky Practice in Mobile Apps with Automated Program Analysis

Zhiqiang Lin

zlin@cse.ohio-state.edu

August 3rd, 2021