Malvertiser “ScamClub” Bypasses Iframe Sandboxing With postMessage() Shenanigans [CVE-2021–1801]

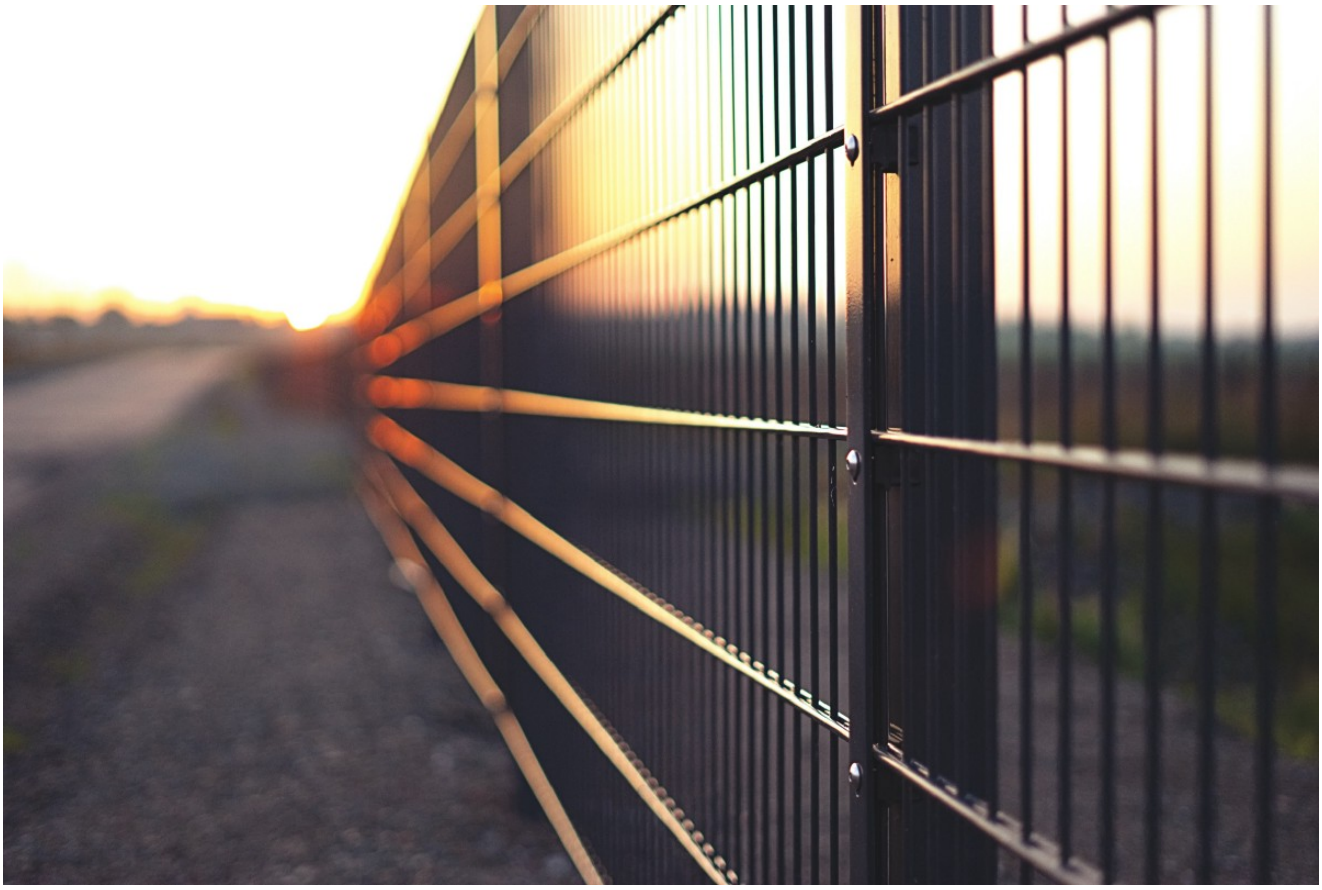
blog.confiant.com/malvertiser-scamclub-bypasses-iframe-sandboxing-with-postmessage-shenanigans-cve-2021-1801-1c998378bfba

Eliya Stein February 16, 2021

Eliya Stein
Follow
Feb 16, 2021

.

5 min read



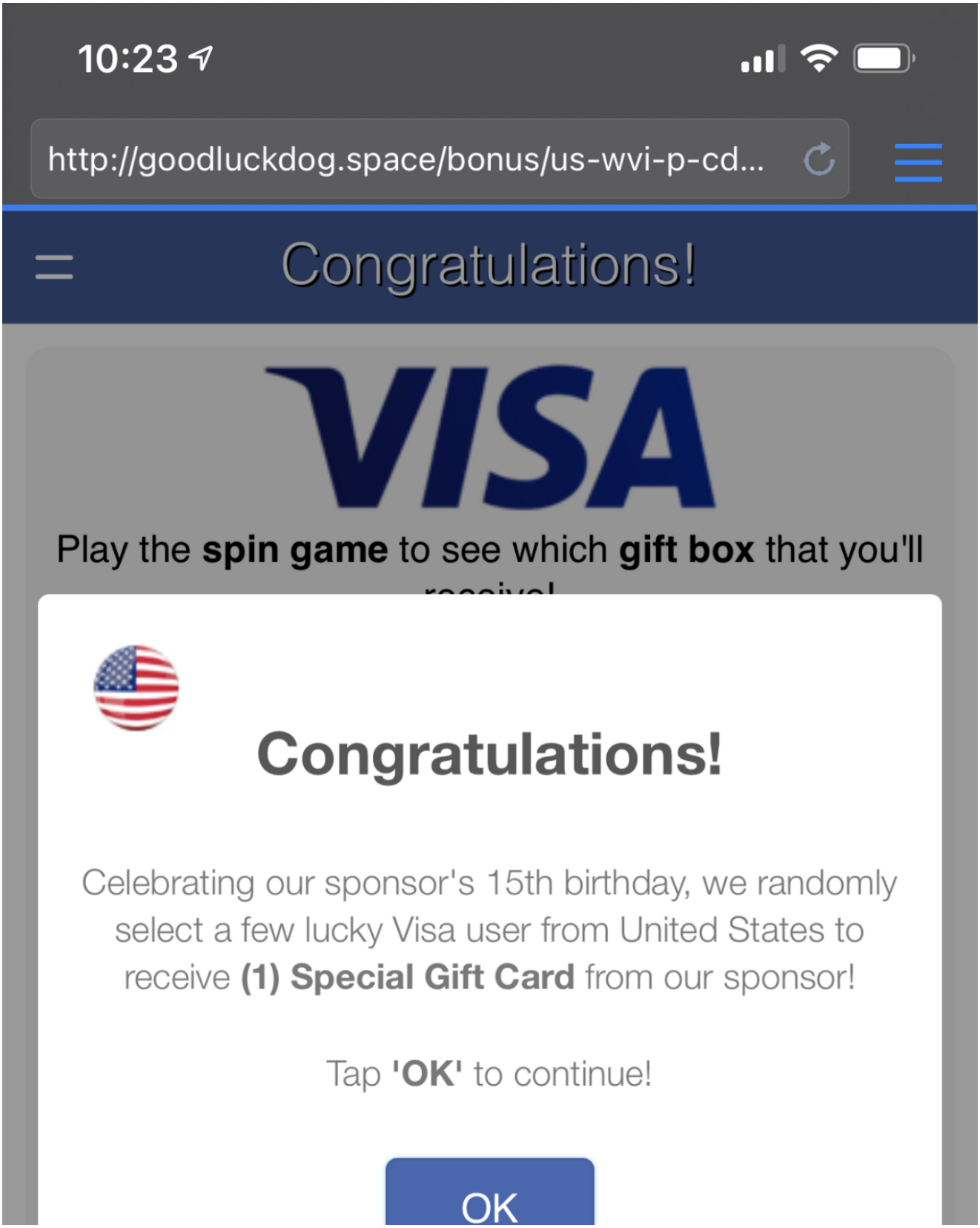
Stock Photo Via Unsplash.com

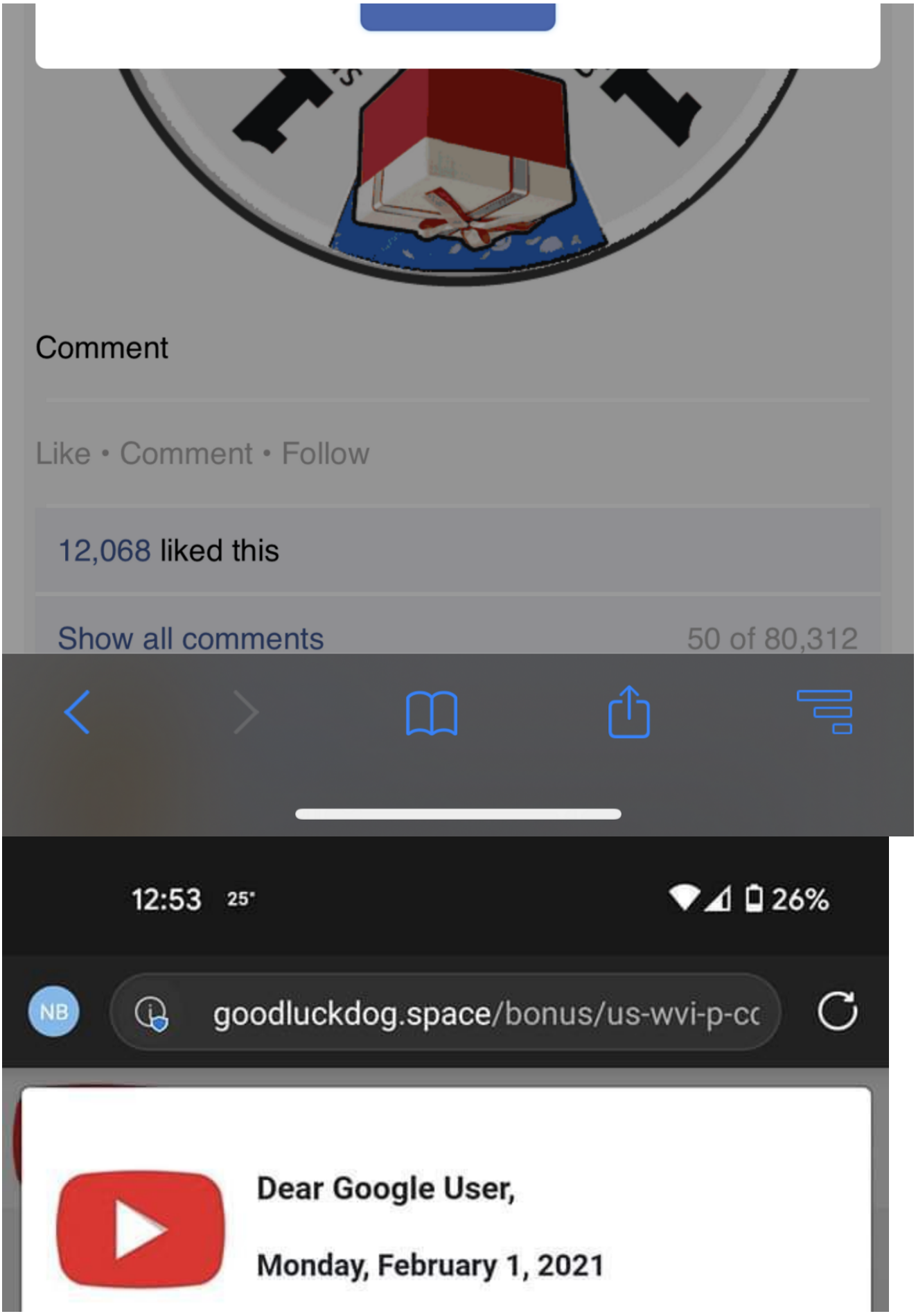
This blog post is about the mechanics of a long tail iframe sandbox bypass found in a payload belonging to the persistent malvertising attacker that we call ScamClub.

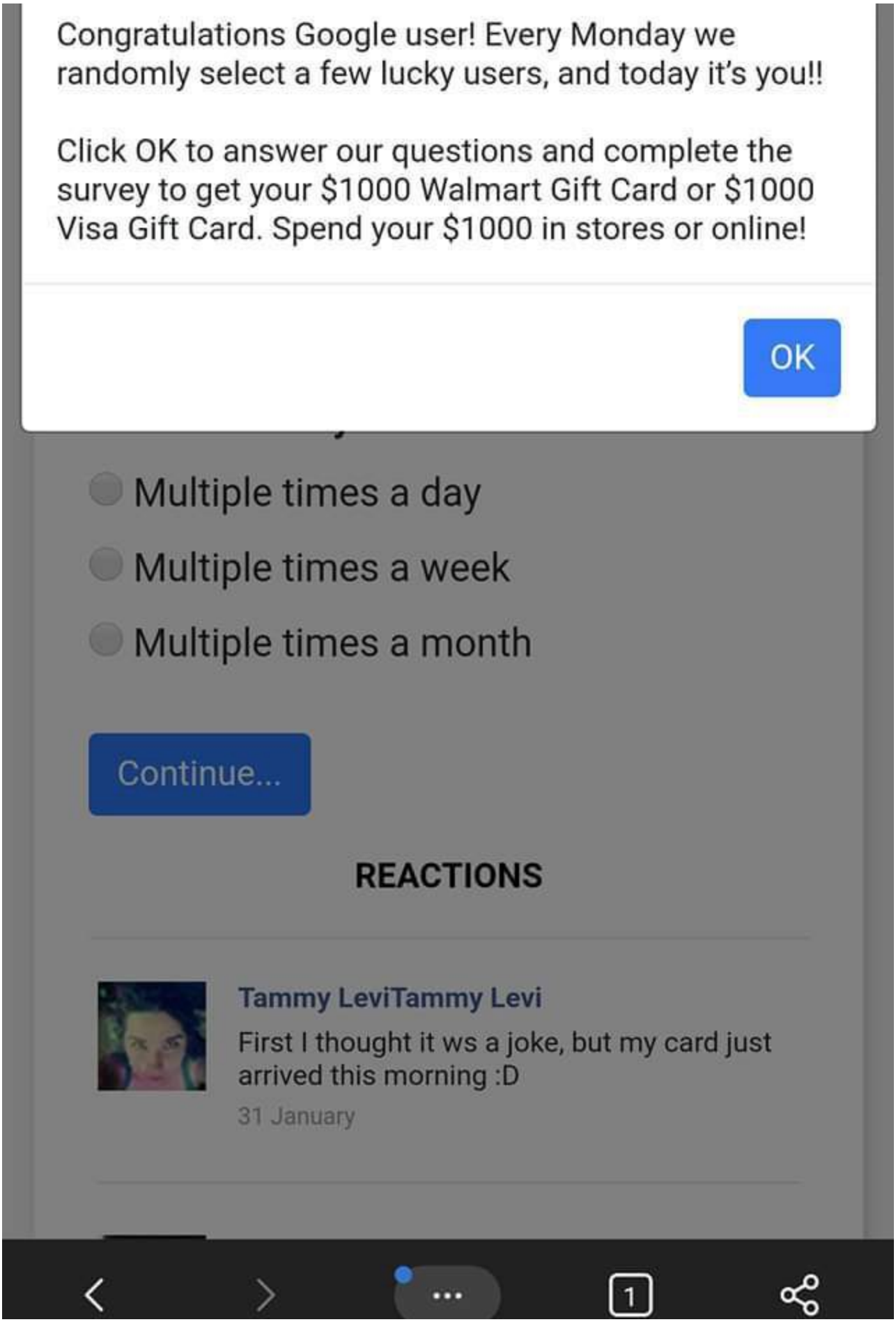
A Word About ScamClub

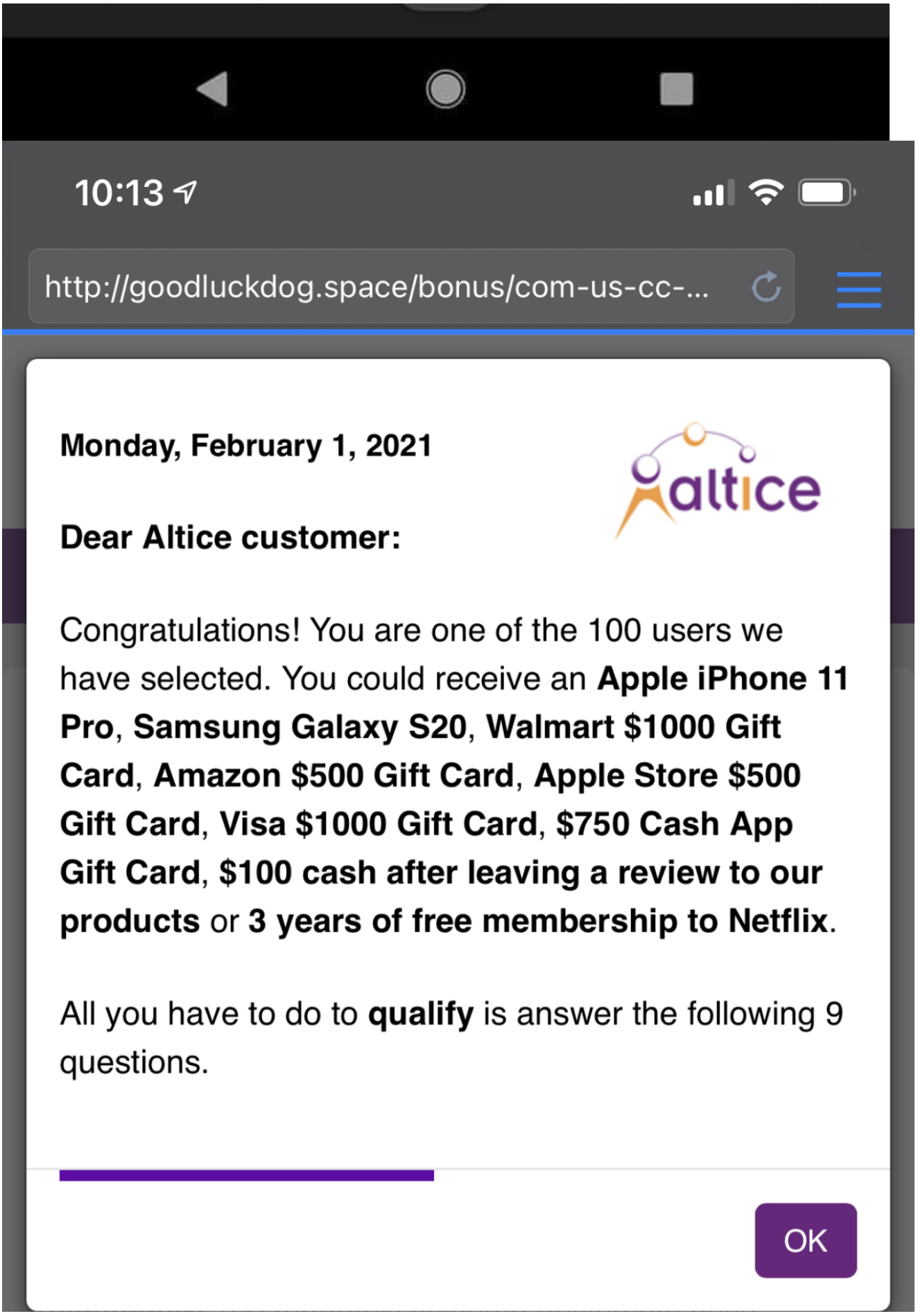
Active for at least several years now, ScamClub malvertisements are defined mainly by forced redirections to scams that offer prizes to “lucky” users, like the all too ubiquitous “You’ve won a Walmart giftcard!” or “You’ve won an iPhone!” landing pages.

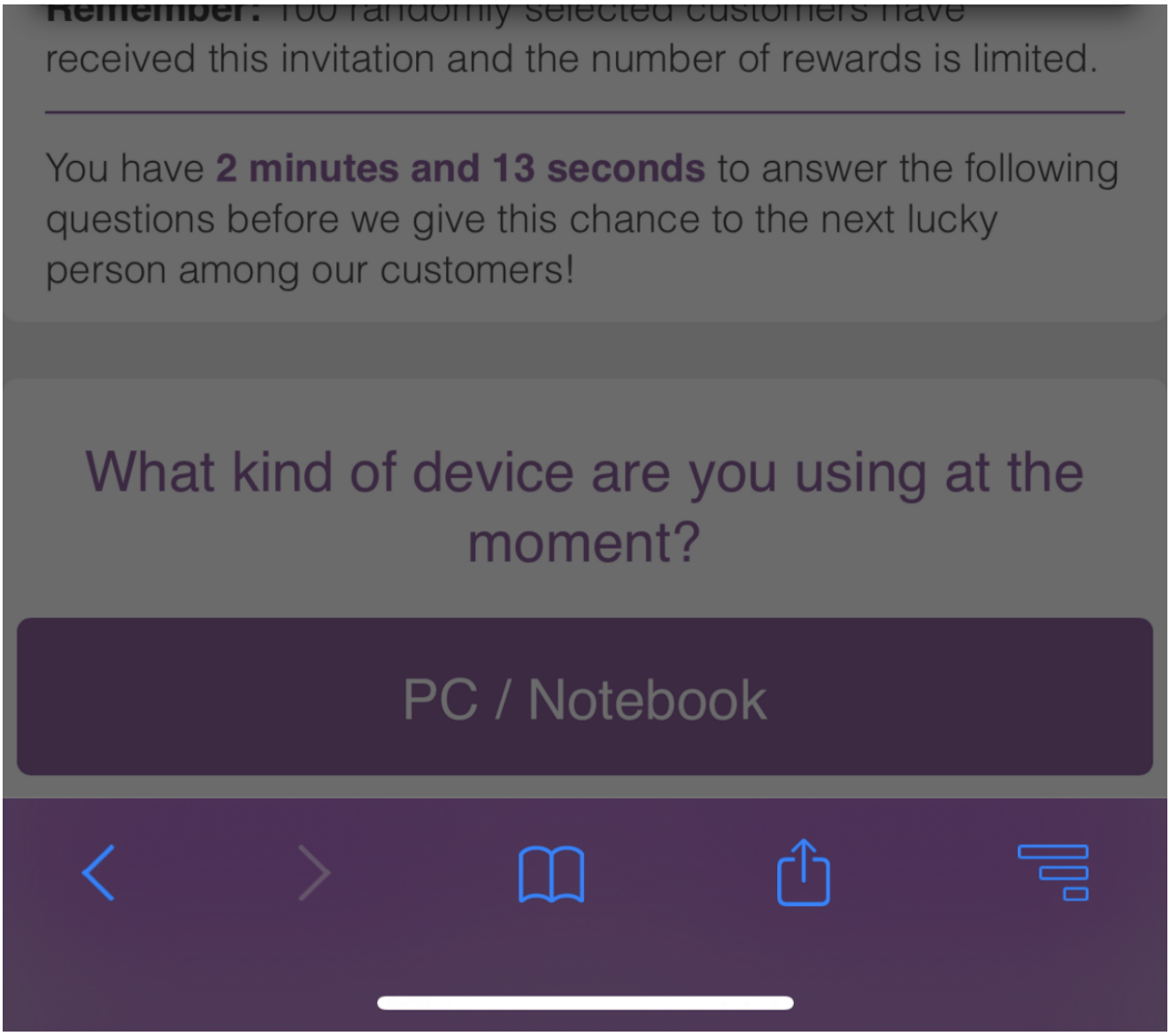
For example:




Comment
Like • Comment • Follow
12,068 liked this
Show all comments
50 of 80,312
12:53
25°
26%
NB
goodluckdog.space/bonus/us-wvi-p-cc
Dear Google User,
Monday, February 1, 2021


Congratulations Google user! Every Monday we randomly select a few lucky users, and today it's you!!
Click OK to answer our questions and complete the survey to get your $1000 Walmart Gift Card or $1000 Visa Gift Card. Spend your $1000 in stores or online!
OK
Multiple times a day
Multiple times a week
Multiple times a month
Continue...
REACTIONS
Tammy LeviTammy Levi
First I thought it ws a joke, but my card just arrived this morning :D
31 January


10:13
http://goodluckdog.space/bonus/com-us-cc-...
Monday, February 1, 2021
altice
Dear Altice customer:
Congratulations! You are one of the 100 users we have selected. You could receive an **Apple iPhone 11 Pro**, **Samsung Galaxy S20**, **Walmart $1000 Gift Card**, **Amazon $500 Gift Card**, **Apple Store $500 Gift Card**, **Visa $1000 Gift Card**, **$750 Cash App Gift Card**, **$100 cash after leaving a review to our products** or **3 years of free membership to Netflix**.
All you have to do to **qualify** is answer the following 9 questions.
OK



ScamClub campaigns have been covered by us in depth over the years, but a great reference if you need a refresher is this post from 2018:

Malvertising Attack Hijacks 300 Million Sessions Over 48 Hours

Nov 12th Malvertising Attack Serves Adult Content and Gift Card Scams

blog.confiant.com

On the tactics side, this attacker historically favors what we refer to as a “bombardment” strategy. Instead of trying to fly under the radar, they flood the ad tech ecosystem with tons of horrendous demand well aware that the majority of it will be blocked by some kind of gatekeeping, but they do this at incredibly high volumes in the hopes that the small percentage that slips through will do significant damage.

The Payload

A typical ScamClub payload has a few layers to it, starting with an ad tag that loads a malicious CDN hosted dependency . This of course is usually obfuscated in absurd ways in attempt to evade url blocklists.

Here's a recent example:

As the payload unfurls to pull in additional layers, we are met with a mess of obfuscated nonsense that usually expands to thousands of lines of code — mostly decoy of course. A full walkthrough of a ScamClub payload is beyond the scope of this blog post, but rather we will focus on four lines of code that piqued our curiosity.

Here they are recreated:

```
function receiveMessage(event) {                          top.location.href =
"http://[malicious giftcard scam]";
}window.addEventListener("message", receiveMessage);
```

The thing is, it's very normal for malvertisers to spray a bunch of redirect attempts in a single payload that try to do the redirect in different ways. It's not uncommon to see a multilayered try catch statement try multiple top level redirects, pop-ups, etc.

The reason some attackers chose to do this is because they have partial control at best when it comes to what device or platform the payload is running on, and they want to maximize their monetization opportunity as best they can.

For example, one browser version might block redirect attempts from cross-origin frames, while the prior version lets them through, so they try all of the things including known bypasses that might have since been patched.

However, none of this explains the event listener… or does it?

We investigate by eliminating the noise and stage a simple html file that implements a cross-origin sandboxed frame and a button that dispatches our event. (payload.html is our event listener).

```
<iframe src="" id='targ' sandbox="allow-forms allow-pointer-lock allow-popups-to-
escape-sandbox allow-popups allow-same-origin allow-scripts allow-top-navigation-by-
user-activation"></iframe>

<script>

  function do_it(){        var wn = document.getElementsByTagName('iframe')
[0].contentWindow;                wn.postMessage('Hello!', '*');    }do_it();
</script><input id="clickMe" type="button" value="clickme" onclick="do_it();" />
```

The `allow-top-navigation-by-user-activation` sandbox attribute, which is often lauded as one of the most vital tools in an anti-malvertising strategy should in theory prevent any redirection unless a proper activation takes place. Activation in this context typically means a tap or a

click *inside* the frame.

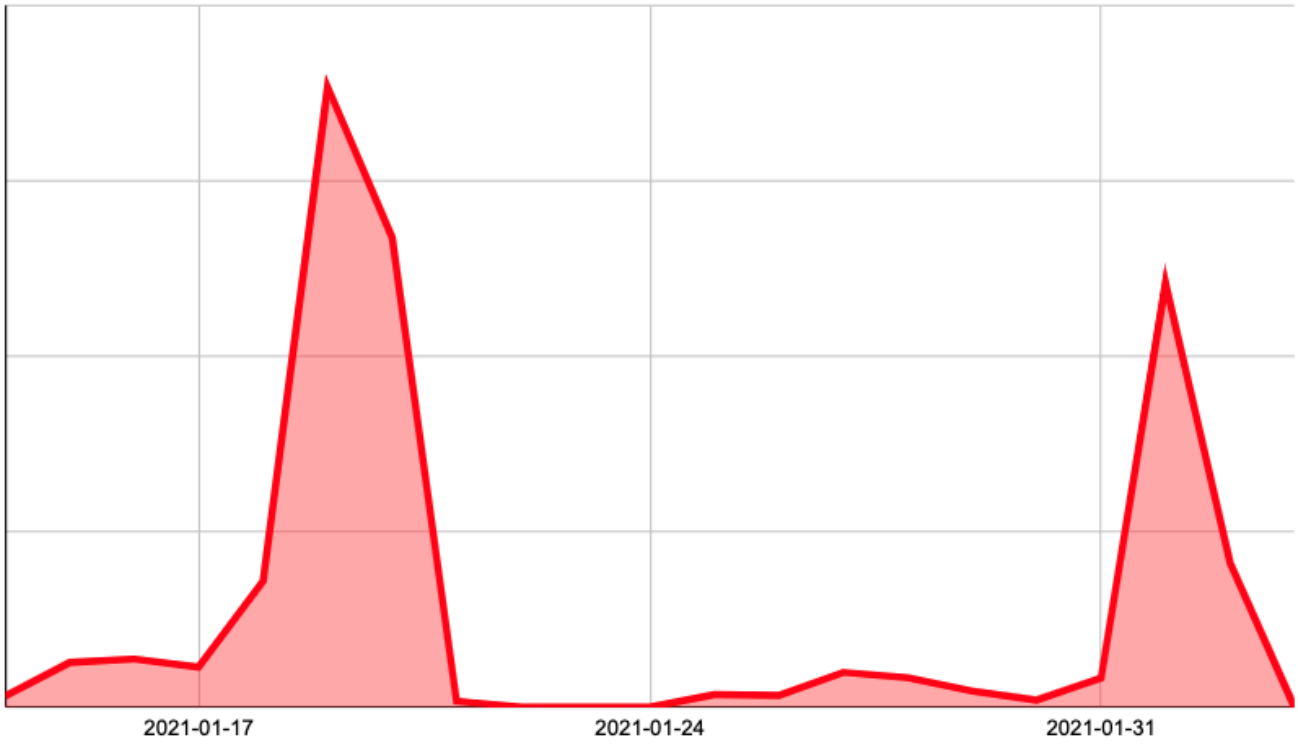
This means our proof of concept shouldn't work under any circumstances. The clickMe button is outside of the sandboxed frame after all. However, if it *does* redirect, that means we have a browser security bug on our hands, which turned out to be the case when tested on WebKit based browsers, namely Safari on desktop and iOS.

The Long Tail

If you're following along, this is probably where you might shrug your shoulders and say "But so what? It's not like these guys can place a clickable message dispatcher on the publisher site outside of their ad frame!"

This is true, but that doesn't mean the ScamClub event listener is a complete shot in the dark. In modern web applications, messages are flying around all the time, usually with wildcard destinations, often on user interaction.

Combined with ScamClub's large volumes and broad targeting that hits dozens of different websites, it's all about the increased efficacy of spawning a successful redirect — even if we're talking about a single digit percentage increase, that can mean tens of thousands of impacted impressions over the duration of a single campaign.



ScamClub was busy in January 2021

> Over the last 90 days, ScamClub has delivered over 50MM malicious impressions, maintaining a low baseline of activity augmented by frequent manic bursts — with as many as 16MM impacted ads being served in a single day.

Disclosure Timeline

June 22, 2020 — Confiant Security Team observes event listeners in ScamClub redirect payload.

June 23, 2020 — Bug report submitted to Apple Security. Google Chrome team also notified (Chrome on iOS uses WebKit).

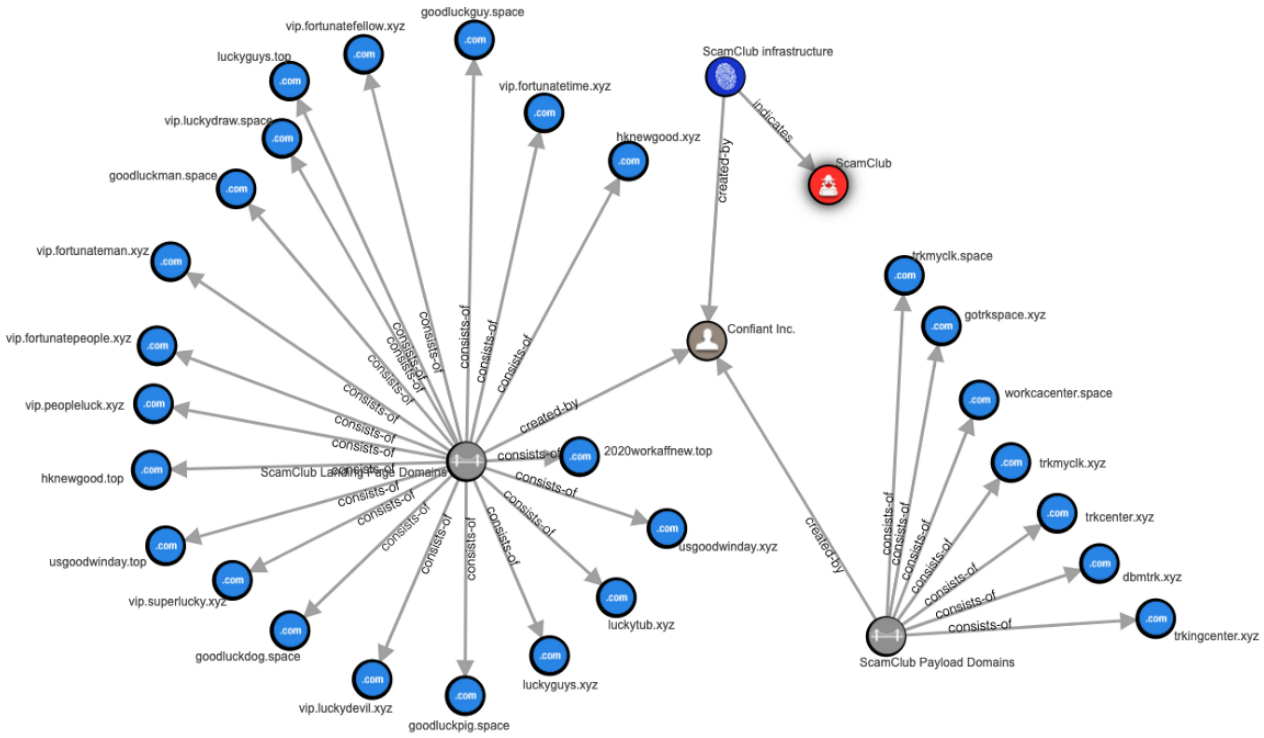
Dec 2, 2020 — Patched https://trac.webkit.org/changeset/270373/webkit

Feb 1, 2021 — CVE assigned and published as part of Apple's security update: https://support.apple.com/en-us/HT212147

IOCs

Recent IOCs are all available as STIX here:

> https://github.com/WeAreConfiant/security/blob/master/stix-feeds/scamclub.stix.json



S3 hosted JavaScript payloads:

xmou.s3.us-east-2.amazonaws.com/mou.jsimpve.s3.amazonaws.com/create.jsdgoi.s3.us-east-2.amazonaws.com/goi.jsyflx.s3.us-east-2.amazonaws.com/flx.jsmiil.s3.us-east-2.amazonaws.com/iia.jsdjian.s3.amazonaws.com/jia.jsaimppv.s3.amazonaws.com/jiy.jsayleieast-2.amazonaws.com/pan.jsdkjieg.s3.amazonaws.com/jieg.jsadlya.s3.amazonaws.com/lya.jsyddceast-2.amazonaws.com/xop.jsaqkol.s3.amazonaws.com/kol.jsimpvv.s3.us-east-2.amazonaws.com/dsd.jsmqyuj.s3.amazonaws.com/yuj.jswpbgm.s3.amazonaws.com/bgm.jspzhufmap-southeast-1.amazonaws.com/lr.jskiyy.s3-ap-southeast-1.amazonaws.com/ki.jsoummm.s3.amazonaws.com/ou.jsgsyyd.s3.amazonaws.com/gs.jsqqpm.s3.aap-southeast-1.amazonaws.com/nx.jszpdk.s3.amazonaws.com/zp.jsmrptm.s3.amazonaws.com/mr.jsktzmy.s3-ap-southeast-1.amazonaws.com/kt.jsnzdpy.s3-ap-southeast-1.amazonaws.com/nz.jsvpydy.s3-ap-southeast-1.amazonaws.com/vp.j

Domains:

goodluckpig.spacegoodluckman.spacegoodluckguy.spacegoodluckdog.spaceluckytub.xyzluckyg