lanmaster53.com

lanmaster53.com - Tim Tomes, lanmaster53.com.

- Published: date not stated
- Original: <https://www.lanmaster53.com/2018/03/15/report-spam-get-owned/>
- Preserved from: https://www.lanmaster53.com/2018/03/15/report-spam-get-owned/ (live) on 2026-08-10
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

lanmaster53.com

- [[image: image]](https://www.lanmaster53.com/)
- [[image: image]](https://bitbucket.org/lanmaster53)
- [[image: image]](https://github.com/lanmaster53)
- [[image: image]](https://twitter.com/lanmaster53)
- [[image: image]](https://www.linkedin.com/in/lanmaster53)
- [[image: image]](https://www.youtube.com/user/lanmaster53)

Report Spam. Get Owned.

Thursday, March 15, 2018

So, a couple weeks ago Matt Svensson (@TechNerdings) dropped me a DM in Twitter:

>

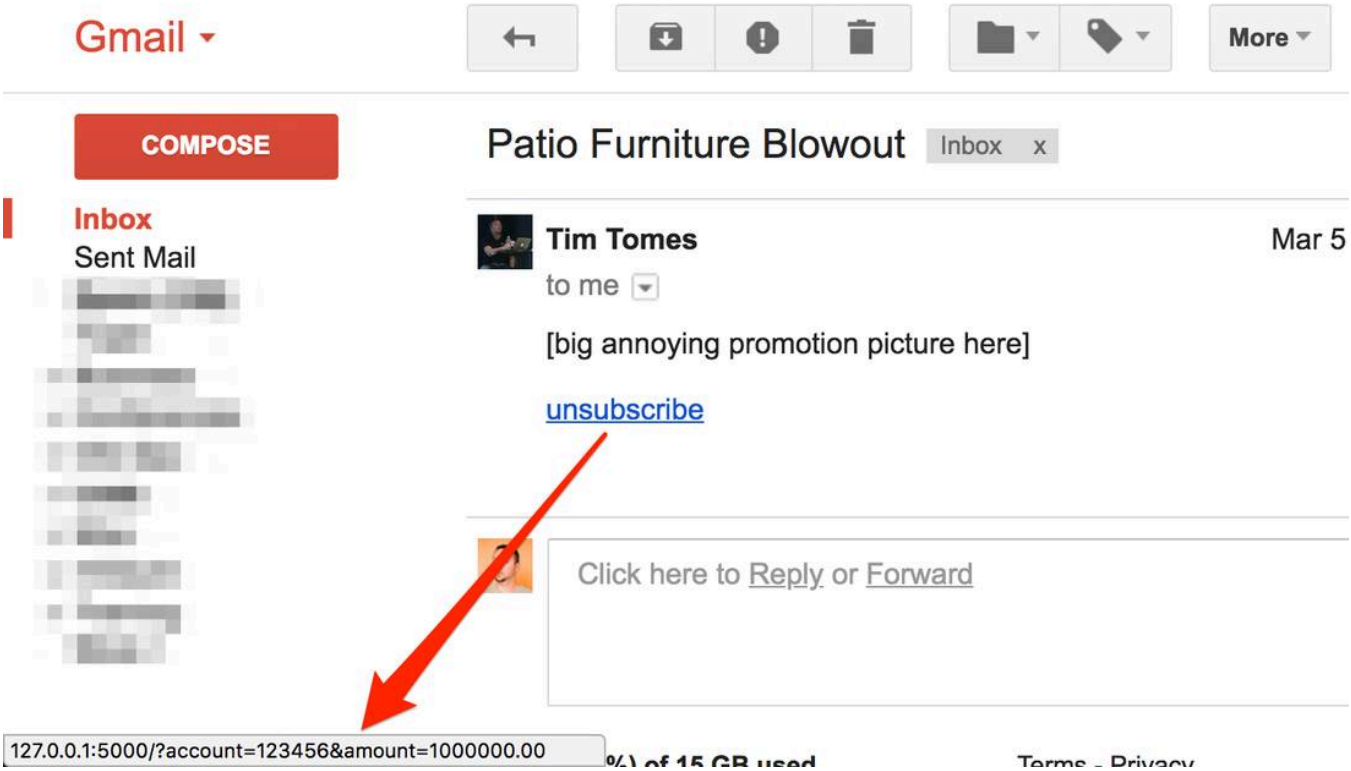
Random other thing that I am curious if you guys have seen anything on... I just got an email from the local eye clinic. I hit the "spam" button on Gmail to report spam and unsubscribe. What I didn't realize is that it actually opens the unsubscribe link in the browser. Good news, easy unsubscribe. Maybe.....if you properly craft the spam...you could use the unsubscribe button to open a malicious web page?

Um... yeah! I immediately thought of how great a CSRF-via-email attack vector this was. Think about it. Users are trained not to click links, but in the case of Gmail, they're taught to click the handy-dandy "Report Spam" button to report it to the spam filter. But wait a second. The handy-

dandy "Report Spam" button will go the extra step and unsubscribe the user from future attacks as well if the user so desires... and they do.

Scenario

An attacker crafts a spam message with an embedded "unsubscribe" link containing the CSRF attack payload like so:



The attacker then sends the email to their victims. In the process of reporting suspicious links and unsubscribing from future messages, because it's the "safe" thing the victims were trained to do, Google clicks the link for the victims, and the CSRF attack payload is triggered from the victim's browser.

Further Study

Being a user of both Gmail and G Suite, I did some additional testing and noticed some other interesting behavior regarding the effectiveness of this attack across these platforms.

- Gmail to G Suite worked as noted above.
- G Suite to Gmail resulted in a different message that did not have the option to mark as spam and unsubscribe and warned of possible danger.
- Gmail to Gmail worked as well.

Interesting. You'd think the stuff coming from Gmail would be the most suspicious because anyone could create a free Gmail account and use it for spam. But Google trusts the Gmail stuff where it warns of the G Suite stuff.

While in theory I love this idea, it wasn't nearly as awesome in practice. After a little bit of fooling around, I couldn't get it to trigger in any of my accounts anymore. Gmail learned something about

my attempts to replicate the attack and stopped asking about the spam when clicking the "Report Spam" button. Even after going into the spam folder and marking the message as "Not Spam." I suspect when you report something as spam once, Gmail remembers and doesn't ask whether you want to unsubscribe or just filter the next time you click the "Report Spam" button. It just filters it.

Verifying Targets

Before this information is at all useful, an attacker must validate whether or not their target is using one of Google's email services. Detecting Gmail is easy. Just look for the `@gmail.com` domain in the email address. Detecting G Suite isn't much harder. Do a MX record lookup for the email addresses domain (hostname actually) via DNS and examine the mail server addresses. Below is an example of using dig to conduct such a lookup for the `[[email protected]](https://www.lanmaster53.com/cdn-cgi/l/email-protection)` email address:

```
$ dig -t MX practisec.com +short
1 aspmx.l.google.com.
5 alt1.aspmx.l.google.com.
5 alt2.aspmx.l.google.com.
10 alt3.aspmx.l.google.com.
10 alt4.aspmx.l.google.com.
```

As you can see, it quickly becomes obvious who the target is using for a mail provider. Any domain other than `gmail.com` using Google's mail servers is a G Suite user.

Responsible Disclosure

With Matt's permission, I went ahead and submitted the issue to Google as a security issue, knowing full well that it was a long shot. I mean, technically, the onus is on the user to understand their technology, but Google definitely makes it easier to exploit users through their platform, albeit to attack someone else's vulnerability. Google's response?

>

Status: Won't Fix (Intended Behavior)

Thanks Google.

If the bug is ever made public, it will be available here.
