EXHIBIT 19

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: Mike Vernal </O=THEFACEBOOK/OU=FIRST ADMINISTRATIVE GROUP/CN=RECIPIENTS/CN=MVERNAL>
Sent: Monday, June 27, 2011 12:43 PM
To: Will Cathcart; Carl Sjogreen; Dan Rubinstein; Arturo Bejar
Cc: Wayne Kao; George Lee
Subject: Re: [Platform Integrity] I know there's a constant tension between...

The real issue here is this — we only have one tool today (a shotgun) and we give you know warning when we're going to shoot you in the head. This was initially fine because we saved the shotgun for people who were really bad, but over time we've extended this to more well-intentioned actors that are spamming Facebook because they see results (more users) but they don't have the negative feedback cycle. One example is Path — at the end of the day, we should assume they're a well-intentioned actor but they were spamming Facebook because it was giving them viral growth.

All the apps we disabled on Thursday night / Friday were somewhat spammy. But I think the general consensus is that disabling them was an overreaction. With User Karma, we don't permanently ban users who send too many messages. We gently warn them to cut it out, first, and then we restrict their ability to send more messages for a while. That's the model we need to evolve to for developers. We need the punishment to fit the crime.

So in the very short-term (this week), we're focused on a few things:

- Getting Insights launched, even if it's rougher than we'd like
- Moving towards per-channel moratoria rather than disabling for channel-specific negative feedback (blocking stream if you're spamming stream)
- Switching from disable to "forced sandbox" mode, where you can still see your application but users cannot

Over the medium-term (before + at f8), we're moving to a model of conditional distribution, where we show more of your stuff if you have net positive feedback and less of your stuff if you have net negative feedback, with the option to be completely hidden if your negative feedback is very high. We're hoping that makes this less of a punishment model and more of an incentive model.

Given that plan, is there something you think we should be doing differently?

-mike

From: Will Cathcart <wcathcart@fb.com>
Date: Sat, 25 Jun 2011 20:45:13 -0700
To: Microsoft Office User <vernal@fb.com>, Carl Sjogreen <carlsjogreen@fb.com>, Dan Rubinstein <ydanrubinstein@fb.com>, Arturo Bejar <arturo.bejar@fb.com>
Subject: FW: [Platform Integrity] I know there's a constant tension between...

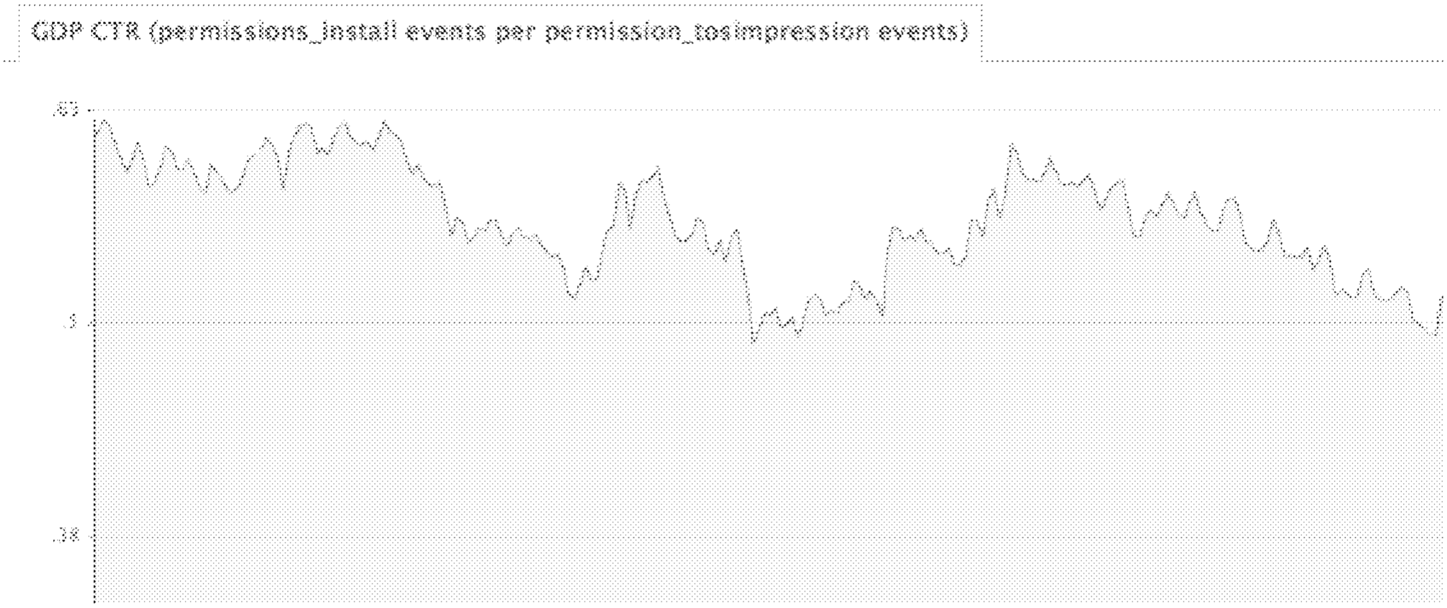
I understand the sentiment, but this really worries me.

One of my growing fears is that we're routinely erring on the side of avoiding pain for developers, and in the process, causing widespread user pain. In the long-run, widespread user pain will severely undermine platform for all developers of all shades and stripes, much more so than the erosion of trust for developers you describe below. And yet, it seems to me that we've been over-optimizing for reducing that developer pain, even before last week's mistakes.

Mostly this sense comes from watching some of the spam attacks over the last several months and our reaction to them. I thought the one Clement thoughtfully tee'd up at Q&A in March was a good example, but also comment-jacking, interactions with POPS/policy on apps with incredibly high rates of negative user feedback, etc.

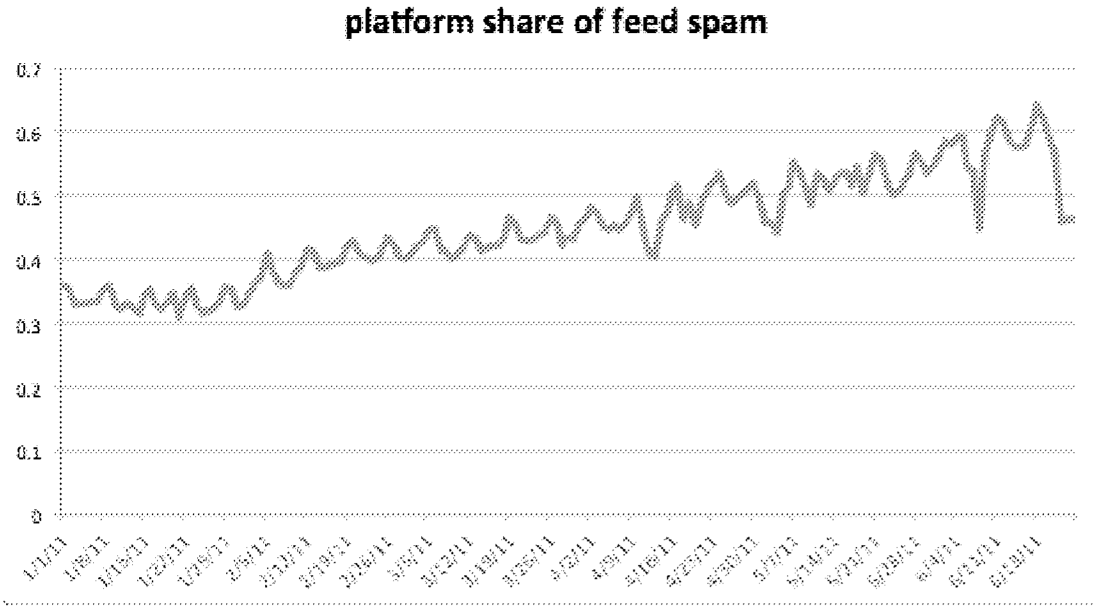
But, to try and express my fear with data:

1. Users don't trust apps to do the right thing. My understanding is that 56% of the time when a user sees a platform permission dialog, they don't grant them. This has been steadily getting worse – it's up from only 39% a year ago. Anecdotally, I've watched many friends and family members encounter a permissions dialog, hesitate, and – when I queried them – describe anxiety over what would happen to their account based on past negative experiences.


GDP CTR (permissions_install events per permission_tosimpression events)

2. Users don't trust us enough to handle bad apps. We recently did some research on the effects of different types of negative engagement on users perceptions of Facebook. While not the worst of all negative interactions, reporting an application had fairly bad effects: one of the highest hits to overall likelihood to recommend Facebook, the least likely to feel neutral/positive about the experience of having reported it of any of the major feedback categories, and the greatest detriment to how likely users were to describe Facebook as fun. A lot of the anecdotal feedback was along the lines of: *"I really feel that report an application will have no result."*

3. Platform spam is getting worse. Until last week's changes, the share[*] of feed spam complaints (itself a rapidly rising number over the last year) coming from platform apps has been on the rise.



A lot of the measures you describe below are about Pareto efficient improvements, which are obviously great (#1/#2/#3), but for any given level of tools we have available, we're implicitly making tradeoffs between user and developer pain. I was already worried that the natural reaction from the PI team to last week's mistakes and the corresponding coverage was going to push us farther away from balance on the user/developer tradeoff, and I'm terrified that things like #5 will exacerbate this.

~Will

```
* SELECT ds, SUM(IF(app_id>0,1,0)) as app_num, count(1) as
total_num FROM nectar_si_feedback_r WHERE
appeventtype='feed_spam' AND ds>="2011-01-01" GROUP BY ds. This probably includes things like iphone,
 etc., but I bet the change in share would be worse if we figured out how to exclude them.
```

From: Mike Vernal <notification+f_z4sjy9@facebookmail.com>
Reply-To: Reply to Comment <g+413ui3r000000056i900026bzii0c81001rl7slsybs1tc46@groups.facebook.com>
Date: Sat, 25 Jun 2011 18:29:21 -0700
To: Platform Integrity <platformintegrity@groups.facebook.com>
Subject: [Platform Integrity] I know there's a constant tension between...

Mike Vernal posted in Platform Integrity.

Mike Vernal 6:29pm Jun 25

I know there's a constant tension between protecting users and respecting our developer community. And I know that before Thursday of this past week, there were a lot of people complaining internally that we weren't being aggressive enough about protecting users and we were letting too much spam through.

On the flip side, it's very, very bad when we disable a legitimate application. It erodes trust in the platform, because it makes developers think that their entire business could disappear at any second. I've seen a little bit of cheerleading about the aggressiveness of the action on Thursday/Friday -- it's really not appropriate. Some of these apps were malicious, but a lot of them were developers trying to build apps within the rules we set out.

We really need to get to a new world, ASAP, where a few things happen:

1/ We have a graduated set of enforcements, and are more conservative with them. If you are marked as spammy for wall-to-wall posts, you lose the ability to make wall-to-wall posts, but your app still works. If you are marked as spammy in general for stream, you lose the ability to post to stream, but your app still works.

2/ If you violate multiple channels or are otherwise flagged as being more malicious, we move your app into sandbox mode. To do this, we need to make some improvements to sandbox mode:
* You can't get out of it without our approval
* You can't make API calls for people who previously TOSed but are not developers/testers
* You can still access Insights, etc. to understand what went wrong.

3/ We need to launch Insights. We've been talking about this forever. We need to get this out and give a feedback loop to developers.

4/ We probably need to send email + notification based warnings before you get placed on moratoria or bans.

5/ We need to pull together a list of false positives every week and start sending it to me, Carl, Don, Dan, Bret, and others to make sure we're being as careful as possible.

This is tricky stuff; one week everyone is yelling that we're not protecting users enough. The next week everyone is swooping in and saying we're being too aggressive. It's a delicate balance, but both groups are right. We need to soften the punishment ASAP so that we can protect users without screwing developers. We can't kill apps over minor or accidental infractions.

View Post on Facebook · Edit Email Settings · Reply to this email to add a comment.