EXHIBIT 106

UNREDACTED VERSION OF DOCUMENT SOUGHT TO BE LODGED UNDER SEAL

From: TR Vishwanath </O=THEFACEBOOK/OU=EXTERNAL (FYDIBOHF25SPDLT)/CN=RECIPIENTS/CN=3B7968ED4055433DAA8C67F062C5FFDE>
Sent: Wednesday, October 09, 2013 10:57 PM
To: George Lee; Ilya Sukhar; Vladimir Fedorov
Cc: Vishu Gupta; Eddie O'Neil; Namita Gupta; Sean Ryan; Douglas Purdy; Harshdeep Singh; Amir Naor; Gareth Morris
Subject: RE: Invites & PS12N

Maybe this problem is already solved but here's what I was getting at - in general I understand we want to make it hard for a developer to grow a new app by cross promoting from existing apps. The one exception is canvas - we want developers to be able to grow a new canvas game by cross-promoting from an existing canvas game, and this is what the lookup API would solve.

Is my understanding correct? If so my question is about how we would enforce that the developer does not abuse the lookup API to grow games off canvas (for example zynga.com). If we prevent this by policy, how do we enforce such a policy?

Thanks
Vish

From: George Lee
Sent: Wednesday, October 09, 2013 10:51 PM
To: TR Vishwanath; Ilya Sukhar; Vladimir Fedorov
Cc: Vishu Gupta; Eddie O'Neil; Namita Gupta; Sean Ryan; Douglas Purdy; Harshdeep Singh; Amir Naor; Gareth Morris
Subject: Re: Invites & PS12N

Redirecting off canvas is against policy and devs can actually do it today. I'm not sure how this lookup API helps with that abuse case...@Vish
- can you explain?

On 10/9/13 10:48 PM, "TR Vishwanath" <trvish@fb.com> wrote:

>I think the lookup API is fine although I'm worried about the developer
>exploiting this for cross-promotion off canvas (ex Zynga uses this to
>take users from canvas to new zynga.com games). How do we protect
>against this?
>
>Thanks
>Vish
>_____
>From: Ilya Sukhar
>Sent: Wednesday, October 09, 2013 10:35 PM
>To: Vladimir Fedorov; George Lee
>Cc: Vishu Gupta; Eddie O'Neil; TR Vishwanath; Namita Gupta; Sean Ryan;
>Douglas Purdy; Harshdeep Singh; Amir Naor; Gareth Morris
>Subject: Re: Invites & PS12N
>
>Would we give this lookup API to everyone?

>
>On 10/9/13 10:33 PM, "Vladimir Fedorov" <vladf@fb.com> wrote:
>
>>Good point. It does create an incentive in the world where they don't
>>always get email. I apologize I didn't read your original email
>>thoroughly enough - I blame the small phone screen :)
>>
>>Lets get together and chat through the lookup API and how it would be
>>scoped
>>
>>Vladimir
>>
>>On Oct 9, 2013, at 10:06 PM, "George Lee" <george@fb.com> wrote:
>>
>>> I actually don't disagreeŠhowever, I think creating this kind of
>>>implicit dependency isn't the right approach. We're basically
>>>forcing devs to ask for email so they can tie user-app-pair IDs
>>>together which either could hurt GDP conversion or just hurt user
>>>trust ("why does this game need my email?"). If we're ok with them
>>>tying user-app-pair IDs together, then we should just give it to
>>>them instead of allowing them to have it unofficially through asking
>>>for email.
>>>
>>> On 10/9/13 6:16 PM, "Vladimir Fedorov" <vladf@fb.com> wrote:
>>>
>>>> I don't think cross game promotions breaks with option 2. As long
>>>>as the user TOSes the application - the developer has access to
>>>>unique identifiers (such as email) that they can use to stitch
>>>>together profiles across application boundaries. So they will be
>>>>easily able to cross promote people who use app A if their friend
>>>>uses both apps A and B
>>>>
>>>> Thanks,
>>>>
>>>> Vladimir
>>>>
>>>> On Oct 9, 2013, at 6:03 PM, "Vishu Gupta" <v@fb.com> wrote:
>>>>
>>>>> I like the lookup based solution for cross-app intelligence building.
>>>>> To be fair, it does make the graph a little bit leakier, but
>>>>> solves a big part of the cross-promo problem.
>>>>>
>>>>> Regarding game friends: I would not say its completely orthogonal.
>>>>> If we do not do lookup based approach, game friends would be the
>>>>> way to help developers build their custom invites MFS. We have
>>>>> also talked about this in terms of "suggested friends API" before.
>>>>>
>>>>> From: George Lee <george@fb.com<mailto:george@fb.com>>
>>>>> Date: Wednesday, October 9, 2013 4:38 PM
>>>>> To: Eddie O'Neil <ekoneil@fb.com<mailto:ekoneil@fb.com>>, TR
>>>>>Vishwanath <trvish@fb.com<mailto:trvish@fb.com>>, Vishu Gupta
>>>>><v@fb.com<mailto:v@fb.com>>, Namita Gupta

>>>>><namita@fb.com<mailto:namita@fb.com>>, Sean Ryan
>>>>><seandryan@fb.com<mailto:seandryan@fb.com>>, Vladimir Fedorov
>>>>><vladf@fb.com<mailto:vladf@fb.com>>
>>>>> Cc: Douglas Purdy <dmp@fb.com<mailto:dmp@fb.com>>, Ilya Sukhar
>>>>><is@fb.com<mailto:is@fb.com>>, Harshdeep Singh
>>>>><harshdeeps@fb.com<mailto:harshdeeps@fb.com>>, Amir Naor
>>>>><amirn@fb.com<mailto:amirn@fb.com>>, Gareth Morris
>>>>><gjm@fb.com<mailto:gjm@fb.com>>
>>>>> Subject: Re: Invites & PS12N
>>>>>
>>>>> + ilya, harsheep, amir, gareth
>>>>>
>>>>> From: George Lee <george@fb.com<mailto:george@fb.com>>
>>>>> Date: Wednesday, October 9, 2013 4:31 PM
>>>>> To: Eddie O'Neil <ekoneil@fb.com<mailto:ekoneil@fb.com>>, TR
>>>>>Vishwanath <trvish@fb.com<mailto:trvish@fb.com>>, Vishu Gupta
>>>>><v@fb.com<mailto:v@fb.com>>, Namita Gupta
>>>>><namita@fb.com<mailto:namita@fb.com>>, Sean Ryan
>>>>><seandryan@fb.com<mailto:seandryan@fb.com>>, Eddie O'Neil
>>>>><ekoneil@fb.com<mailto:ekoneil@fb.com>>, Vladimir Fedorov
>>>>><vladf@fb.com<mailto:vladf@fb.com>>
>>>>> Cc: Douglas Purdy <dmp@fb.com<mailto:dmp@fb.com>>
>>>>> Subject: Invites & PS12N
>>>>>
>>>>> Sorry of the long email, but there are a few threads and I'm
>>>>>attempting to consolidate and offer some options.
>>>>>
>>>>> G
>>>>>
>>>>> Key concern
>>>>> We're leaking the social graph to platform developers and that
>>>>>needs to stop.
>>>>>
>>>>> Proposed solutions to date
>>>>> 1/ We remove all non-app friends from the API and force all
>>>>>developers to use a FB-hosted invite dialog to reach non-app
>>>>>friends. We also obscure the FB UID sufficiently to prevent
>>>>>reconstruction of the social graph. The proposal is to move to a
>>>>>user-app-pair hashed ID and only share first/last name and profile
>>>>>pic for non-app friends.
>>>>>
>>>>> *   Pros
>>>>>   *   Completely prevents developers from building out the social
>>>>> graph as they will only have the graph for users who have TOS'd
>>>>>their app
>>>>> *   Cons
>>>>>   *   All custom multi-friend selectors will break and developers
>>>>> will need to integrate a new FB-hosted invite dialog
>>>>>   *   Developers with multiple applications will no longer be able
>>>>>to
>>>>> cross-promote intelligently b/c the user-app-pair hashed ID means
>>>>>that the same users looks like two different users across apps by

>>>>>the same developer.
>>>>>
>>>>> 2/ We only obscure the FB UID sufficiently to prevent
>>>>>reconstruction of the social graph. The proposal is to move to a
>>>>>user-app-pair hashed ID and only share first/last name and profile
>>>>>pic for non-app friends.
>>>>>
>>>>> * Pros
>>>>> * Custom multi-friend selectors still work and will not require
>>>>> integration to a new FB-hosted invite dialog
>>>>> * Cons
>>>>> * Does not completely prevent developers from building out the
>>>>> social graph as they may still have enough information (email,
>>>>>first name, last name, profile pic) to stitch together some
>>>>>semblance of the graph.
>>>>> * Developers with multiple applications will no longer be able
>>>>>to
>>>>> cross-promote intelligently b/c the user-app-pair hashed ID means
>>>>>that the same users looks like two different users across apps by
>>>>>the same developer.
>>>>>
>>>>> Dealing with the cons
>>>>>
>>>>> * Breaking multi-friend selectors - this is basically a
>>>>>non-starter.
>>>>> Developers (especially games) have invested heavily into
>>>>>multi-friend selectors over time and they are frankly better than
>>>>>we are at optimizing this right now. It would also require game
>>>>>mechanics to change b/c it would require a strict difference
>>>>>between requests to existing players and invites to non-app players.
>>>>> * Killing cross-promotion - as mentioned in previous emails, this
>>>>>is
>>>>> also a non-starter in the sense that it is an important game
>>>>>installs channel for our developers today (see below). The big
>>>>>humps are game launches and the steady state is just the
>>>>>cross-promotion strip at the top of games that has now become
>>>>>common. At ~1M installs a day it is close to 10% of our daily
>>>>>installs. All of that said, this doesn't inherently kill
>>>>>cross-promotions in full. What it means is that the first time a
>>>>>user shows up in a game, they actually don't know if you already
>>>>>play any of the other games by that dev. They know this today b/c
>>>>>of the clear-text FB ID. So, showing me Farm Heroes Saga could be
>>>>>either an ad for a new game I should play or just a bookmark to an
>>>>>existing game that I already play. This is the loss of control
>>>>>that creates a problem for developers. Few possible options here:
>>>>> * We tell developers that getting user email via permission is
>>>>> the way that they stitch things together.
>>>>> * We tell developers that there are ways for them to stitch
>>>>>users
>>>>> together outside of our own systemŠflash cookies, passing user
>>>>>params on the intra-canvas clicks.
>>>>> * We implement one of three hashed ID solutionsŠall of which

>>>>> require a solid developer/app ontology
>>>>> * We implement a dev-user hashed ID
>>>>> * We do a lookup on our end and pass along proactively all
>>>>> user-app hashed IDs for any other apps that user uses from that
>>>>>developer
>>>>> * We implement a new lookup API where a developer can get all
>>>>> other user-app hashed IDs for any other apps that are managed by
>>>>>that developer
>>>>>
>>>>> I actually think the first two options are reasonable, but I
>>>>>conceptually like the idea of implementing a lookup API as well.
>>>>>This
>>>>> would suggest that we go ahead with option 2/ above and supplement
>>>>>with the lookup API. Thoughts?
>>>>>
>>>>> [cid:052F5FE1-1176-4B83-8771-61CBEEC656C0]
>>>>>
>>>>> [cid:01CF6B74-A2C2-4494-AC96-7CC74FCDBA80]
>>>>>
>>>>>
>>>>> Other considerations
>>>>>
>>>>> * Game friends - I think this is orthogonal. We originally
>>>>> considered this an option if we were taking non-app friends away
>>>>>b/c leaking the gamer friends graph is less of an issue. Now that
>>>>>the hashed ID has traction, I think this is moot.
>>>>> * Migration - seems reasonable that if we support a lookup ID, that
>>>>> the lookup could support the original FB ID as well for some
>>>>>period of time. Would easy the migration concerns a little bit.
>>>>>Are there other concerns here?
>>>>>
>>>>> Invites as a separate effort
>>>>> As you all know, we've been trying to make progress on requests.
>>>>>As part of that, we've created a new FB-hosted MFS that we're
>>>>>starting to roll out (it includes both requests to existing users
>>>>>and requests to non-app usersŠaka invites). Separately, Connie is
>>>>>working through some ideas on new receives-side experiences
>>>>>(beyond notifications, home reminders on web and app center on
>>>>>web). We're still iterating here.
>>>>> Separately, Eddie is working on whether we should use messenger as
>>>>>a way to deliver an OG object to a friend through an explicit,
>>>>>FB-hosted messenger interface. That also feels orthogonal to the
>>>>>main invites issue above and we should really talk about that
>>>>>separately.
>>>>>
>>>>> Make sense? Feedback?
>>>>>
>>>>> G
>>>>>
>>>>>
>>>>>
>>>>>

>>>>>
>>>>>
>>>>> <Screen Shot 2013-10-09 at 2.16.17 PM.png>
>>>>> <052F5FE1-1176-4B83-8771-61CBEEC656C0.png>
>>>
>