All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

The Facebook debacle and what it says about the other providers

Anybody that didn't imagine that Facebook would do something generally distasteful with whatever power was given to them, has been living in another world. This is only the next in a series of such things that Facebook has done throughout its history. But that's not the story here. The story is that every other major provider allowed them to do it, favoring the corporate connection over the personal privacy of their user base. Shame on them, and a pox on all of their houses.

What am I talking about?

Facebook forced its hundreds of millions of users to change their email addresses from whatever they were into a user ID followed by @facebook.com.[1] "Facebook users say contacts' e-mail addresses on phones and personal devices have been altered without their consent -- and their e-mail communication is being redirected elsewhere, and lost." This includes users at major corporations, in governments, in small and medium sized businesses, and individuals from all over the World. The net effect includes – so far (1) misdirecting emails from the normal account to the @facebook account – but not necessarily the same user ID as the original recipient's new Facebook email address, (2) altering and removing existing email addresses of many/most Facebook users on other systems, without warning or notice – so that their previous email addresses are lost and the mobile devices of perhaps tens of millions of users are no longer correct – and no longer retain the previous correct email addresses, (3) global denial of services to users seeking to send email to previously valid accounts, and of course, (4) a massive loss of trust in all of the affected providers – which is to say – all of the major providers around the globe.

Where can users go to be safe? Nowhere!

It is to be expected that any – indeed many major providers of service will act maliciously from time to time. Whether by ignorance, the race to the bottom, a falsely placed trust, the desire to serve their shareholders, malicious attack, insider abuse, or simply accidents. That we have known for years. But the massive effect of Facebook with its applications on devices all over the World and of many different sorts, is a clear example of lack of adequate controls by Apple, Microsoft, Android, corporate systems, access control and related security providers, and just about anyone else you can identify with the mobile and desktop providers that allowed an application – any application – to change email addresses without notice, consent, and/or other adequate controls. It demonstrates just how poorly our protection works, and how poorly thought out the protection schemes of all of these providers are.

The story is simple enough. Too much trust is put in applications. They are granted unfettered access to the devices they operate on – to a lesser extent on desktops – but still to enough. No application really has any legitimate need to change or see anything it doesn't create, write, read, and control without explicit permission of the user. But the desire for integration...

1 Violet Blue,Cnet news "Facebook e-mail mess: Address books altered; e-mail lost", June 30, 2012.

The desire for integration

Isn't it cool that my calendar can integrate with my mapping application, my phone book and calling mechanisms, my email, contacts, and my Web browser? Sure. And that is what drives our systems into interdependencies that create the havoc we have only just begun to see.

Consider this. Suppose a trusted application is hacked – an exploit to LinkedIn, Angry Birds, a server at an online advertising agency, Google, Yahoo, you name it. That's bad enough, of course, but with this sort of integration, your calendar, contacts, emails, calling, Web services, and any and every other service you have can all lose integrity, availability, confidentiality, use control, and accountability. Now, one hack, accident, act of malice by an insider, gets it all – on a global basis – for an enormous number of users – across all of their applications.

Escalation

The Facebook accident (assuming it was that) of today is the strategic attack of tomorrow. You can bet that every country and serious threat to information protection has noticed the impacts and started their crash program to create more exploits across this same spectrum of systems. And no quick fix will do, because the integration is exactly what the devices are used for. Get rid of the integration and you get rid of the major revolution in functionality. This class of attack on this scale is likely to escalate, out of control, and soon. They may not be as spectacular seeming, but they will be broad, deep, and have major effects. Even the simplest effect – consumption of battery and communications bandwidth – will potentially make mobile communications far less reliable and more costly, while driving the usage time for the devices down. Replace emails in subtle ways, read messages between parties and send copies to the opposition, intercept private communications on a massive scale, disrupt business operations, all within reach. You bet it is a disaster waiting to happen – more likely already in motion.

What to do about it?

The role of government – at least the US government – legitimately includes “provide for the common defense”. If we are to depend on these systems for our national infrastructures, if it is indeed to continue to be critical, then there can be no doubt that there is an urgent need for government to govern more. Regulations? Fines? Liability for all parties whose systems were in the causal chain? Piercing the corporate veil for massive scale effects? Personal liability for officers and board members? Class action law suits? The loss of trust has not yet hit home, and perhaps it won't for some time. But you can bet that automatic updates, vendors downloading software changes underneath of operating environments, change in libraries, and all of the other find and fix approaches are readily exploitable today to similar ends and on similar scales. The company store approach didn't save anybody this time, and the controls surrounding the massive numbers of applications distributed through so-called trusted channels still lack the basis for any trust at all.

The technical issue – change and inventory control

The basic technical issue comes down to change control and inventory control. Change control and inventory control have been around since the early days of computers and the middle ages respectively. Well understood principles and operational methods have long been in practical use. It's not that hard. Here are some basic principles for end-user systems, including mobile and desktop devices.

- **User permission for changes:** No changes to user information should be done without explicit user permission for the individual change of a datum (i.e., submit commit cycles change by change for datum, fields, records, etc.). For example:
  - You cannot change or delete an individual or set of {picture / calendar entry / contact / todo / memo / etc.} or their characteristics without explicit permission of the user granted through the operating system and independent of any program running within it. The interface should clearly allow the user to tell the difference.
- **Reversion:** The user should be able to go back in time allow, step by step, to any prior point in time, restoring any value for any field in any record in any dataset.
  - If I allow a picture to be deleted, I should be able to go back in time and recover it. Disk is cheap these days, so an 8Gig mobile device can be fully backed up 120 times for $100, and over its entire life cycle, it is unlikely to actually consume a whole disk in reversions.
- **Inventory control:** The user should be able to see and control what their data is and where it is at any and all times. They should also be in control of what programs are able to change and see what programs have changed and seen which data, fields, records, and datasets.
  - The permission systems in most modern devices grant things like “disk access” or “network access”. These are overly broad. When a program (e.g., the Facebook application) tries to access or modify a record in any dataset other than the one it creates and maintains (each application should have a private area for its own content), the user should be notified and given the option of using false data so that the application can continue under false assumptions, denying access so that the application will fail if the data is required, or granting access to the real data, all either temporarily or permanently with user revocation at their sole discretion. The specific data requested should be shown to the user.
  - All accesses of all programs to all data should be tracked and made readily visible to the user, who should also be given the option of revoking access to the content retrospectively and at any time, with the program provider forced by law to remove it both for itself and recursively for anyone else it has been given to.

Conclusions:

The path of the industry that has led to the fast and loose alteration and leakage of user data and controls has continued to escalate and worsen over time. It's almost impossible to find a platform you can trust anymore, and even the most knowledgeable and skilled user or defender simply cannot know what they need to know to protect themselves or those they are responsible to defend in today's environment. Complexity is the enemy of security, and the enemy of all of us. And it will get far worse before it gets better.

The technical solution is to impose a regimen of controls – change control and inventory control – over data at all levels of detail for all time and for all uses. The only viable path to invoking this solution or any alternative is through governmental control. Industry will not do it on their own. At least that's how I see it.

