**EMCS2430: Human Factors: People and Software**

6. Convenience Versus Security: TripIt and App of Your Choice

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

*Part I - TripIt*

*TripIt is a popular travel organizer app (there are others like it that probably also behave similarly, so this analysis is not specific to TripIt). If you don’t know what TripIt does, please take a look at it first.  
TripIt has a convenient feature: if you give it your GMail credentials, it will scan your inbox and automatically extract all travel-related mail without you having to forward your messages to TripIt.  
Comment on this feature. Should one use it? Discuss any convenience versus security trade-offs. Your answer should try to use the concepts introduced in this module.*

“Giving an application your credentials,” that sounds pretty scary by itself. Maybe it doesn’t sound scary until we delineate exactly that means, because there is a difference between “giving an application your credentials” and “authorizing an application to access your inbox.” Let’s analyze the fine details, then make some determinations on what is what.

**“Giving an application your credentials”**

Actually giving an application your credentials, or giving them your username and password means:

1. Your username and password are stored in a database ( maybe even more than one if they do regular backups ) for use by an application, that would log in as you. You have no guarantees that these credentials are encrypted in the database or what type of encryption is used to secure the credentials. Furthermore, even if they are stored in the database with strong encryption, who has the key? The developers ( all of them )? The owner of the application? Anyone who steals access to the application? Does giving your credentials to an application seem safe now?
2. Once the application has your credentials, it can act like you and on your behalf. Now, the application has said that it will just be scanning the inbox for emails related to the purposed of TripIt’s functions. *But, did they pinky promise?* The fact is that the application can do whatever it wants like it was you and on your behalf ( send emails, delete emails, copy and store emails ) because that is what having credentials means. Credentials are meant for the authorized user, not for an application. Unless you are giving your credentials to your administrative assistant or your spouse. I would think twice about giving this try of access to an application.
3. Password Stuffing / Credentials stuffing is a very common attack strategy for people who commit crimes and malicious state actors.

**“Authorizing an application to access your inbox”**

There is a possible different meaning to the question posed in the exercise. One that is more common in the world of modern application development. With the help of a library called OAuth, an application can get a token which would allow it to access another application or service based on the validity of the token. The token is usually issued over a SSL/TLS connection so others can’t intercept it and has to be re-authenticated or reissued based on parameters defined by the application developers. Unlike giving your “credentials to an application” the application can have a defined scope of access, one that is usually defined by the application that is being accessed and communicated to the user when access is given. In this way, the user can revoke the access when she wants, and even revoke parts of the access if she wishes. This more advanced system does not guarantee more safety and privacy, however. Sloppy implementation of OAuth has many of the same dangers as giving away your credentials. If the application that is granting the token doesn’t limit the scope of the authorized application the user is open to abuse. However, I would argue that OAuth provides a well thought out balance between ease of use and security if used correctly.

*Part II - App of Your Choice  
Provide at least one other example of an app (ideally one that you use) that poses significant usability versus security tradeoffs, and how these are manifest through the permissions.*

Facebook, used by almost 2 billion people, and probably the most readily easy to abuse application on the planet. Recently I [listened to a podcast](https://thecyberwire.com/podcasts/cw-podcasts-daily-2019-02-05.html) that said Facebook doesn’t just save a hash of your real password, but they also allow you to log in with 4 different versions of the same password in case you “type it slightly wrong” or have cap locks on. My jaw hit the floor. I can only imagine the heated arguments that happened internally over that. Personally, as an engineer and as a security professional, that pretty much guarantees that I will never work for Facebook. A company that bakes the disregard for one of the most vulnerable pieces of our security into their app has truly lost touch with where we are at now in a time where security and privacy are under fire. I get it, people mistype password on a mobile device because typing on the little screen is hard for …. Someone, some people. But, really? We are going to make the job of hackers easier because people are too lazy to remember what probably isn’t a very strong password in the first place? I am disgusted by this report and with FB for implementing such a feature in the face of their ongoing controversies.