1. Flaws identification

Recall that along every system there are several breaches which would need to be disposed off whenever its components are to succeed amongst the whole. Therefore, according to the environment in which *the system* may develop in we will hint any possible security breach we might find, nonetheless we would encourage the user to be aware of these every time before exposing themselves to danger itself.

Warning: there is a serious real risk of information loss.

1.1. Input vector

Once the system has migrated at the cloud: 1) Information is being shared along an intermediate communication channel throughout any data is certainly not at least invisible (or encrypted). Therefore, catching and tracking user's data traffic out of a VPN is a real threat which we must consider whenever it allows a third person to break in between the tradeoff user-server.

1.2. Assets

Amongst priorities there are Confidentiality, Integrity and Accessibility. Moreover:

1) User's financial information is *critical*. 2) User's identification is *critical*. 3) User's preferences are *useful*.

1.3. Suspect's attribution

It is certainly a matter of data gathering. Whom owns user's data is able to perform censor, track and follow, disable user access or even phising. Unsuccessful businesses may raise a malicious adversary. Unhappy costumers may want to attempt to fix their choices. Thief and impersonation may often take place. At last but not less important, any system is exposed to being beaten directly (or indirectly) as a collateral damage within the clash of powerful sides or forces above them.

2. Actions (Before attack)

It is worth considering strategies in order to guarantee the CIA (Confidentiality, Integrity, Accessibility) triad, besides earlier assets.

2.1. Digital site identification

All users should be educated so they are able to identify suspicious sites before they can transfer any data towards the server. Remind all cloud-based systems are build up from data storage to applications which remotely perform on a server all kind of operations based on the information received from the user through communication channels (such as internet). So it is important to make sure no unauthorized individual is collecting nor operating any data fraudulently as convincing us of being something else of what they really are. Moreover, this kind of practice may be considered *phising* itself.

Advice: Enable an intuitive user friendly interface in which users can learn more about the site permissions and its certification issuers, furthermore, read more about the legal agreements and conditions. Also let them know if it is the case their data is travelling unsafety unencrypted towards the server, just as without the *https* communication protocol or outside a *VPN*, so that they might understand what they are exposed to and act accordingly.

2.2. Multi-factor authentication

As expected it is impossible to guarantee all users are willing to apply (2.1) amongst other things due to the increasing scalability growth rate, exposing an uncontrollable security breach between users' behavior and an actual cybernetic response mechanism. Despite this, there are several ways of having users to acquire safety habits furthermore without they even get to know. So it is possible to strength the weakest barrier between internal private data and external malicious agents. Some of these is the so called multiple factor authentication paradigm, as its name would suggest, this is a traditional authentication manager implemented at a deeper level; which basically approaches the id validation problem by considering there are three things accurate enough to detect someone's identity, these are: valuable possessions someone may have, secret information someone may know and characteristic features someone may personate. This way users will be safe in case they lose some (not all) of their authentication keys, making it safer whenever they will know or be notified if they are being impersonated.

Advice: In case users were unreachable physical token authentication may be inefficient, so implement a double password authentication mechanism which would require all users to login twice. So user and password would be needed the first time, once it's done, users might need to authenticate a random generated code received on their smart phones or email.

2.3. Least privilege

In case (2.2) isn´t enough systems can have a least privilege access. So attackers might hit and damage only a specific part of the system's structure. Furthermore, if the application is implemented well enough it would allow programmers to completely absorb the nocive effects derived from the attack fair enough.

Advice: Build you site so that all customers can read and comment, likewise virtual assisting admins can answer users requests and enable them to add actual articles to their shopping record, registering the purchase only occurs once the user has saved that article in its history. Users are not allowed to do any further than reading, commenting and administrating their own shopping records. Admins are not allowed to do any further than enabling certain users to modify their own shopping records.

3. Reactions (After attack)

To be continued…

Presented by: *Esteban Hernandez Ramirez*