All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

How to steal a billion IDs and Passwords and how to prevent it – Trust Architecture

I'm not sure the exact numbers, but browser providers these days are providing the very useful, ability to share your browser-stored IDs and passwords between browser instances. Do you have them on your phone, desktop, pad, and laptop computers? Fantastic..

Except for one thing

If I should gain access to the storage area they use to synchronize them I gain access to not only all of yours but all of everyone's associated with the browser.

But encryption, multi-factor, and …

I've heard it all before. Just don't believe it. It's ridiculous to imagine that with the volume of capabilities and monies available by breaking into these mechanisms, there can conceivably be enough protection to make them safe. Just consider these little things:

- An insider puts a Trojan into the browsers, or crypto mechanisms, or storage, or ...
- The encryption turns out to be breakable with quantum, cloud based, or mathematical breakthroughs, or perhaps already is…

I could come up with more, but I will leave that up to your LLM to give you lists.

So what's the problem?

It's all about risk aggregation, or in other words, trust architecture.

- Who and what do you trust to what extent for what purpose over what time frame?
  - You individually – that's one thing – but society writ larger, that's quite another.

If you put that much trust on any technology, eventually your trust will not be justified.

What do we do about it?

Disaggregate the risk. Or in other words, architect the trust to that the amount of value gained by attacking is lower than the cost of attack. An example?

- Let the user provide their own storage selections (provider, account, storage name):
  - Google cloud? AWS, Microsoft? Your ISP? There are lots of providers out there, and more will pop up if enough business is out there.
  - Then, the attacker who wants my passwords will have to: (1) Figure out which provider and account I am using, (2) how to bypass authentication on their system, (3) the name or mechanism I chose to store it under
  - And that will only get them my password store. Which I could also split across multiple storage areas… and setup different authentications and identities, and ...

Conclusions

The point here is that we have lots of these kinds of things, and we have not applied what we know, as a field of endeavor. How do you architect and manage trust? More next month.

