All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

Let’s all go to 10-factor

In case you didn’t guess by now, this is about authentication – proof that it’s me – or you. I should start by noting that my wife and I share credit card numbers as co-holders of the same credit card accounts. So whatever you do to prove it’s me, in these cases, better allow her to prove it’s her as well. Now that I have broken (read denial of services) 99% or more of all multi-factor authentication systems in actual operation, let’s move on to some other issues.

We already have 2-factor built in – maybe 3 or 4… or more...

Current systems tells me when I use a new device to login (or when anyone else does). That means I can do a timely response to undo most of the harm, but it is less than ideal. So the two factors are the password and device in use. So when we hear the cry for 2-factor – we already have it.

Actually, the device alone is not all that is used. Typically you have to authenticate to the device as well. So the device authentication PLUS the device PLUS your password for the service makes this 3-factor. Some folks will claim that two passwords are not 2 factors.

But the device also uses encryption or cookies – so the cookies on the device are actually another factor. As long as the device isn’t stolen and the content for authentication is controlled by the device, I have a third (or is it fourth) factor by the stored cookies, etc. in the device.

The fifth factor is location. Dorothy Denning wrote a paper years ago called “location-based security” (or some such thing). Most current mobile devices have location information and with the minimum travel time results from research done in the DARPA ADAMS program, location becomes a reliable, albeit imperfect, fifth factor.

The sixth factor is service provider. My cell phone uses a limited set of service providers and my computers are connected through my service providers and they provide traceability information (IP address as well as routing details if desired), so access path becomes the sixth factor, add IP address, and we are up to 7 factors.

Usage patterns are also examined by many systems, and unauthorized access attempts lead to additional proof requirements, so that’s 8 factors. Add typing behaviors and we have 9 factors. Then we have call-backs for submit commit cycles, perhaps using independent paths and devices, and we are up over 10 factors.

That’s 10-factor authentication WITHOUT adding biometrics!

Never enough

So I already have 10-factor authentication, I still have never had a password guessed on any system that I care about, and the only cases I am aware of where someone got one of my passwords was when a 3rd party let someone break in and the break-in resulted in leaking my password. That’s why I use the password ****** in almost every system I use that I don’t care about. I redacted it to force you to look it up in a hacked database and email it to me.

A reasonable question to ask is “how much is enough”? A seemingly reasonable answer is “enough so nobody breaks in”, but of course we all know that is not going to happen. There is now and never will be a perfect system. Give it up.

- The answer does not lie in perfection, it relies in risk management. And that means understanding risks.

Risk management and authentication

- Understanding risks means understanding that one size does not fit all.

You don’t have to be an expert in risk management to see that

- My needs for authentication for my doodle account are not the same as the President’s needs for authentication to fire a nuclear weapon.
- You cannot reasonably make my risk management decisions for me.

Managing risks effectively requires that the entity whose risk it is:

1. Understand the nature of the risks they face (sense, communicate, cogitate)
2. Decide whether and how to act on those risks (decide)
3. Address those risks per their decision (communicate, act)

Any effective scheme for authentication should have these same properties. It should allow the person whose risk it is (i.e., the user or entity that “owns” the user) to understand the nature of the risks they face, decide how to act on it, and provide the means to act that way.

An example

You will likely find few who are more aware of, better able to understand, or more able to act on cyber-related risks than your author. So you would think that my decisions might be a guidepost for yours. That would be a mistake. My needs are different from your needs, and as a result, my decisions are likely to be different from yours. I also play a risk management role for several entities, and each has different needs, and different decisions get made for them.

Submit-commit cycles and related methods?

This example applies to my everyday use of online services. I have two cell phones, several laptop computers, and infrastructure computers that support remote Internet services. I live in a place with pathetic cellular service, most of the time, and I have redundant Internet connections and carriers so that, in total, I can make or get a cell call using any of 4 different telephone carriers and 2 different ISPs – that’s 6 different ways to get or make calls. And yet, I cannot reliably get text messages sent to me from Internet service providers, like Google, at least for a period of several hours, and sometimes a day or more. So much for the “Call back” or “text message” authentication (submit commit cycle) process.

Notification systems?

I should also note that I have something like 20 different email addresses I send from and 100 or more I receive to, all associated with different purposes for different businesses I own, operate, work with, invest in, etc. So when I am told I only have to use a separate authentication once a month, that translates into 120 authentication processes a month, usually taking more than a minute each, usually all at the same time, usually while I am in a real-time Internet-based meeting with important clients. That happened to me once and I decided that the bad multi-factor authentication process had to go away.

I get notified every time I login to a new account from a new computer. Actually, new in this context means not yet used for this account from this place before, or after a reset of the computer or a reload of the software or a software upgrade, or whatever. This happens in a series of typically 2-4 messages sent to each of 2 accounts (primary and backup). So with 2 cell phones, 4 computers in daily use, 20 outbound accounts, and updates weekly in some cases, you would expect about 720 email messages a week just to inform me of a use from a device that had not previously been used (or meeting their criteria for used). So that is not going to cut it.

Single sign on?

Single sign-on (actually reduced sign-on) seems sensible for someone like me, but of course single sign-on (1) does not work across the various services I use and (2) aggregates all my risks in the SSO provider(s) and all their surrogates everywhere. It actually makes my risk of compromise far worse. Add to that the access controls associated with these services exceed my desires (I don't mind if they get a copy of the email addresses, but I don't want them to have the phone numbers and I don;t want them to be able to delete or change any information). Once you get in one place you are in all sorts of other places – not my desired risk profile.

But that's not the only issue. Legally, I need to keep my businesses separated, and I need to keep my client accounts separated. Otherwise, if you get into one you get into all. So that is a definite no no.

One time passwords on paper

I have used one-time pads before for logins, but when you have so many accounts this becomes a paper nightmare. I would have to have 20 labeled (so I can associate them with the different accounts) small font (so they fit), pieces of paper with me all the time, and if any of this gets stolen, I then have to change my passwords (do si do) or otherwise invalidate their use. And of course they could be taken, copied, and replaced so that I don't know they were taken. I could also store them on my phone – but then if someone takes my phone they have the codes to access the accounts the phone could access, which is no different from the situation today without the one-time passwords. One time pads suffer from the key distribution and secure storage problem. Passwords don't. They have other problems of course. But what do I gain for what I give up?

An authentication application

Google has an authentication app that is supposed to generate codes as a function of time to grant additional authentication. Other similar things, like the RSA keys folks used to carry around have similar problems. The RSA keys were broken algorithmically allowing attackers to guess the codes. All such systems have similar potentials for misuse. But in addition, I have to have one such device or mechanism per multi-factor system. Then there is the problem that if they are on my device (e.g., phone) I still need to enter my UID and password to generate the code to augment my password. It depends on the password and the device! If you get my device and password you have the codes so why bother with the codes?

Priorities

My risk profile is such that I need to be able to access reliable, authentic information. Secrecy is far less important. If I am denied access for too long, things start to fall apart big-time. That means that if Google goes away, if the Internet is down, if there is a local or regional power outage, if my ISP fails, I still need to be able to get to reliable and authentic data and use it in order to continue to succeed in my job. To do this requires use control and accountability – not secrecy – control over who can change what, records of who changed what and when, and the history of previous content before it was changed. These are systems of record.

Passwords over encrypted channels with trustworthy endpoints – my actual solution

I am fine with 3-factor - or 10 factor - if it is done well - for example once - or only on access from a new device - or from a master account only instead of all the accounts - or if it gains me something. Note that I have been doing this for more than 40 years and so far I have never had a password used by anyone but me as far as I can tell. And I have had passwords controlling access to about 20 of my systems on average - for scores of accounts on some systems. So that's ~800 user account years of experience with ZERO detected password defeats. I don't change my passwords very often - perhaps every 10 years or so I might do it. What if I don't know it and someone has my password? No harm, no foul! (and I can detect).

You can easily defeat my protections – if you really want to – as I can yours

Like anyone else, I am vulnerable. If you put a gun to my head I will tell you my passwords. You don't even have to go that far, so please don't. My computer accounts are not worth my life or anyone else's. They are less valuable than broken bones or hospital bills or any number of other things that someone who really wanted to get me personally could reasonably think of. Don't cut off my fingers or toes or other body parts – I have nothing that valuable. And you might make me mad. It is important to remember that attackers can also be attacked, and if you get me or anyone else mad enough, they may fight back. I might cringe in fear when you assault me, but I will get a rifle and wait in the tall grass to get even if you go too far. If you are from a military or government entity, you are probably already in if you want to be, and if not, please just let me know and I will arrange for you to have access. No need to become violent.

Conclusion

Security has to be convenient to be effective. And that's what the multi-factor authentication (and most security) people miss. My cell phone has a finger print function, but it requires a password periodically to augment the finger print. Really? Why not just use the password and forget the finger print. For me, a reliable repeatable efficient process outweighs the security benefit. Perhaps they could let ME decide how often I want to be interrupted - or allow ME to decide to authenticate some time in the next day instead of RIGHT NOW because they said so. If security load goes too high, users will avoid the mechanisms. The perceived value of security is less than the perceived punishment of being "secure".

I am not you. If you have one account and once a month you have to enter a code, it's not much of a load. What you gain on average is unclear, unless you are high profile for attack. But at that low a security load, adding a few factors may be a good idea. Add up the pain from the extra steps day after day and compare it to what might happen if your password is taken. Why not a monthly code? But then why not just build all this into the devices we use to access computers, reduce harm from stolen devices, and allow us to block the device if it gets taken?