All.Net Analyst Report and Newsletter

Welcome to our Analyst Report and Newsletter

The now all too common MFA exploitation by MFA – confusing the issue

Multi-Factor Authentication (MFA^1) has often been praised as the authentication solution to authentication pollution. Multi-Factor Attack (MFA_2) is the dilution of the the MFA solution.[1]

What is MFA^1?

Most folks by now have a notion that MFA^1 uses more than one seemingly independent method to authenticate an identity. The identity represents, ultimately, the authority to perform acts. So a properly authenticated identity is able to send nuclear weapons across the World, and a properly authenticated identity is able to transfer funds from and to accounts. Redundancy is used to decrease the odds of accidental or malicious acts. But of course, redundancy costs. We can have redundant identities, authentications, and authorizations required to perform acts, and that redundancy can be at any reasonable level of quantity or quantity and sequential and/or combinational in nature.

It we use a probabilistic model, we can make a claim that if each method has an 1 in X chance of being defeated and is somehow independent, the more methods we use, the lower the chances of defeat (1 in X^N for N methods), and thus we can achieve any desired level of probability of defeat by paying the associated level of cost in adding redundancy. Of course this is pure baloney because the methods we use are not unlimited and not independent of each other.

What is MFA_2?

Most folks don't get it that MFA_2 is in fact feasible, and in fact used every day to defeat MFA^1 schemes. Here is a scenario I have seen used in multiple incidents for theft of funds via wire transfer. It typically involves something simple, like having the victim transfer an amount of money to the wrong place. That involves changing precisely 2 numbers on a single form. Specifically, an invoice typically used to produce a payment through a reasonably controlled process, will have payment instructions. The instructions typically include directions on how to make payment, including an electronic transfer to an identified recipient account (e.g., a routing and account number). Note there are also bank name, account name, and type, and banks are supposed to verify transfers with that other information, but often do not verify it.

Somehow, the right numbers have to get from me to you so you can pay me. All other things being equal, if the numbers I send to you are not the same numbers you get, then you will send the funds to the "not the same numbers", and someone else will get it. Trying to get it back is problematic because such transfers are permanent and non-rescind-able by nature.

The largest such fraudulent transfer I am aware of was for more than a billion dollars, when a billion dollars was a lot of money… The largest fraudulent transfer I was ever able to set up (even though it was not executed) for a client to demonstrate such a weakness was for about $40B, but one of my former students and associates worked at the Federal Reserve, and obviously was able to do this for far larger numbers. These days we see lots of high valued thefts exploiting weaknesses in crypto-currency systems and electronic funds transfers.

1 There is an old, and no longer agreed to, statement that "the solution to pollution is dilution".

Examples of MFA$_2$ methods?

Here are the most common schemes I seem to encounter today:

- Access endpoints and alter the numbers in a PDF
  - At either the sending or receiving end of the transmission of routing and account (or similar) information, the numbers are changed, and then acted upon normally.
- Access email servers and alter the numbers in an email (or a PDF attachment to it)
  - In an email (typically) server, a filter reroutes select emails to the attacker, content is altered to change the account numbers, sent along, and acted upon normally.
- Access a cell phone or its infrastructure and intercept (and/or alter) communications
  - A cell phone is used for authentication via text message. The infrastructure is fooled into rerouting the text by a subscriber identity module (SIM) card substitution and the attacker authorizes without the owner ever knowing it happened.

There is a common theme here. It is the alteration of data values between source and destination. A variation on the classic Malicious[2] In The Middle (MITM) attack. All of these involve improperly authorized access. They are all authorized accesses, just not properly so.

How do we fix MFA^1?

We don't. Improper authorization has always been and will likely long continue to be problematic. The problem, in this case, is really that, as a community, we are trying to solve an integrity problem with an authentication solution. It is the alteration of content that is creating the losses here, and this is not really solvable by making sure that the people and mechanisms with access to the end process are who and what they claim to be. The overall system is a composite of interdependent components creating a process. These complex components and processes create large numbers of possible paths to corruption. The more parts and paths, the more opportunities for attack. This is increasingly recognized as a supply chain problem. And not unlike the supply chain issues associated with software, the content supply chain has to be addressed in order to assure the integrity of the use of systems.

That is not to say that we should give up on assuring proper authorization for access. It's just that solving problems with the wrong tools makes everything look like a nail because we have a hammer. The more we pull on a nail, the more likely it will come out, because it's designed to prevent horizontal motion, not removal. The more we lean on authentication to address corruption, the more likely it is to fail. That's because its not designed to assure integrity.

Conclusions

We see more and more successful attacks on systems using MFA^1 because we use MFA^1 to solve problems it is not designed to solve. Ultimately this only reduces a potentially unlimited number of exploitation by corruption paths by a finite number. It's like trying to build a hill to the moon. Every shovel full of dirt we add gets us closer to the moon, but we will never reach it that way. We need to look at end-to-end integrity more closely and reduce dependency on increasingly complex, interdependent, and difficult to use MFA^1 to solve corruption problems.

2 Used to be Man In The Middle, but apparently women and machines are now also in the middle, so...

