

Customer Service and Security: Social engineering without even trying

By Mikhael Felker

I suspected that my own SSN would not be sufficient to gain account access. However, stubbornness prevailed and I offered my digits.

Recently, I dialed my wireless carrier's customer service line from my mobile phone to get assistance in upgrading that phone. I was asked to verify my identity by providing my name and the last four digits of my Social Security number (SSN).

I happen to be on a shared plan and am not the primary account holder, so I suspected that my own SSN would not be sufficient to gain account access. However, stubbornness prevailed and I offered my digits. The representative kindly told me they were incorrect, and waited patiently for me to remember the proper numbers.

Here's where it gets scary. After I had made several tries at guessing the primary account holder's last four digits – all the while being pathetically apologetic for not knowing them – the representative interjected and kindly *provided* the primary's SSN digits for me.

"Sir, do you mean '8352'?"

"Yes, of course that's what it was!" I replied.

I had just engaged in successful social engineering without even trying – absolutely unintentional, but unbelievably effective.

From an attacker's perspective, this is fabulous news. Knowing the weaknesses in both the technology and the process, an attacker could use stolen phones and attempt a similar tactic to gain account access. After all, part of the authentication process involves calling from the mobile phone itself, and the name of the owner is commonly stored in the phone, such as in contacts or calendar appointments. The last piece of required information – the last four digits of the primary account holder's Social Security number – can apparently be procured via the method described above.

In the aftermath of my successful transaction, armed with a shiny new phone, I questioned what had just happened. The entire process startled me. As a security practitioner my first thought was: The representatives can see all of this information. Why?

The representatives shouldn't necessarily see the last four digits of the account holder's SSN. If the system were designed properly there

might be, for example, a box into which the representative could enter the SSN digits provided by the customer, so that the system could determine if the digits were valid or not. An alternate design might allow the customer to input the last four digits of his or her SSN via telephone keypad. The application then would signal to the representative whether or not the verification process had been successful. Both of these methods would obviate the need for the representative to view the customer's entire Social Security number, while still fulfilling the same functional need. The representative would never see the correct digits if the customer could not provide them.

Other questions also warrant further analysis: Is there *any* legitimate reason for the representative to see the last four digits of my Social Security number displayed on his or her screen? How many other companies have a similar verification procedure? How many millions of customer service representatives have access to my name, address and SSN – all nine digits?

The potentially troubling answers to these questions result from incomplete system design and rationale. The problem is that the application places complete trust in the representative instead of applying the "need to know" principle. The application should help the representative discern no more than that *I am who I claim to be* – not what my credentials are in themselves.

It is important to note that representatives shouldn't be made scapegoats simply because they may be able to access too much information. This is a problem of needs development. Strong application requirements and solid design are needed to address it.

About the Author

Mikhael Felker, CISSP-ISSEP, works for SEI (www.sei.cmu.edu) and is a graduate student at Carnegie Mellon University. He has worked for several FFRDCs and is experienced in developing, evaluating and building security systems. Felker received his BS degree in Computer Science from UCLA. He can be reached at mikhael@ieee.org.