Today's 'Unbreakable' Encryption Is Tomorrow's Security Weakness

We hear a lot about encryption as a magic silver bullet solution that will secure and protect our data. Businesses and consumers are directed to employ encryption wherever possible, and in the wake of any data breach one of the first questions that comes up—often from encryption vendors with a marketing agenda—is whether the compromised data was protected with encryption.

Encryption evolves over time, though. A complex encryption algorithm may seem unbreakable when it's first introduced, but cracking tools and decryption methods will adapt. Over time the computational power available to the average user increases as well, rendering outdated encryption tools virtually useless.

The Roman emperor Julius Caesar is credited with coming up with the Caesar Cipher to encode military communications. The Caesar Cipher offsets the letters of the message by a set number. For example, the name "tony bradley" with an offset of 4 would end up being "pkju xnwzhau". A message converted using the Caesar Cipher would appear to be gibberish. It's not very secure, however, because the maximum offset would be 25 and it would only take a minute or two to run through every possible permutation of the Caesar Cipher to reveal the "encrypted" text.

Now we have significantly more complex encryption algorithms using 128-bit or 256-bit keys. A 256-bit encryption key is theoretically unbreakable—today. A post on Reddit works through the math of the processing power required and claims that it would take longer than the entire universe has existed up until now to brute-force all possible combinations of a 256-bit key. However, a few years ago researchers at Fujitsu were able to crack a 923-bit key in just under 150 days using a combined total of just 252 processor cores.

There are two other important considerations aside from the length of the key itself. First, many algorithms are poorly implemented. Key generation for encryption typically uses a pseudorandom number generator. The computer can't actually be random so the pseudorandom number generator has to be initialized using some known or finite value. If an attacker can find the weakness in the PRNG used to generate the encryption key, the length of the key itself becomes somewhat irrelevant.

Second, theoretical exercises that calculate the amount of time it would take to brute force every possible solution for a 256-bit encryption key seem to assume that the correct encryption key will be the last one that is tried. It is equally possible that a cracking tool will get the encryption key right after a week, or a day, or even on the first try.

Those two points aside, though, the reality is that the complexity or “unbreakability” of an encryption algorithm is a function of the computing power used to crack it. As computers get exponentially faster and more powerful, cracking smaller encryption keys becomes more trivial. When attackers can harness botnets of tens or hundreds of thousands of computers to compress the time required to run through the possible keys, it means that data that is secure today may be easily compromised tomorrow.

Does that mean you shouldn’t use encryption? Absolutely not. Just because encryption can be cracked doesn’t mean it’s trivial to do so, and encryption ensures that your data isn’t the “low-hanging fruit”. Most attackers won’t invest the effort and will move on to other targets. It does, however, mean that you still have to be vigilant about data security, and that you need to periodically assess the encryption used to protect archived data.

Data you encrypted and stored a few years ago might not be so secure any more.

by Tony Bradley
Editor-in-Chief, TechSpective.net
on January 8, 2016