Herding Cats:
Ragebcl Znggref

June 2012

BrandenWilliams
SECURE BUSINESS GROWTH

This month's issue is dedicated to cryptography, and since I am terrible at math, we're going to address advancements (or needs therein) from a slightly different perspective. Over the last several months, we've seen some interesting articles around cryptography, and specifically the issues with key generation for the RSA algorithm. We saw a paper with a snarky title in "Ron was wrong, Whit is right," by Lenstra, Hughes, Augier, Bos, Kleinjung, and Wachter, that analyzed the randomness associated with generating keys for the RSA algorithm[1]. The response seemed to suggest that there was a major flaw in the algorithm—but further analysis showed that the issues were associated with how random numbers were generated, not with the math.

Why am I prattling on about randomness when we are fundamentally talking about a math operation? Because the foundation of any good encryption algorithm includes generating strong keys and protecting them appropriately. This becomes increasingly difficult as we use a variety of low-power processors in devices that are becoming "smarter" by the day. I'm not just talking about smart phones either, I'm talking smart meters, IP-enabled controls for industrial systems, pacemakers or other bionic devices, and household appliances. All of these items may want to establish bi-directional communication at some point, and the information exchanged must be protected and properly authenticated.

Embedded systems pose a very unique threat to this process for two reasons: today many of them are either identically keyed from the factory, or they generate their own keys without sufficient randomness. For example, in the RSA algorithm, if you know the seed values to a particular key, you can derive the key which essentially zeroes out its protection. Pretty scary! In fact, in the Lenstra article, the authors speculate that nearly 13,000 of the keys they tested offered no security at all. The question is, how did we get to a point where nearly 13,000 RSA keys are just window dressing? Many embedded systems simply cannot be random enough to generate truly unique values as an input into generating a key. Either that, or the implementation of the key generation component is intentionally weakened either due to lazy coding or to increase device performance. Is encryption happening? Sure, but it's the equivalent of buying a super sophisticated locking mechanism but weakening it with a master key.

Let's draw a corollary to a major implementation snafu that plagued many of us in the middle of the last decade—WEP encryption. WEP is based on a popular stream cipher called RC4, which has many uses like securing SSL sessions that protect data in-flight. If we trust RC4 for SSL connections, what happened in its implementation in WEP that made it so hackable? Fluhrer, Itsik, and Shamir discovered that WEP's implementation of RC4 leaked information that could be used to derive the entire key[2]. This lead to a number of tools to automate this process and gave yours truly a personal best of three minutes to crack a 128-bit WEP key using an ARP injection technique. WEP is superseded by WPA and WPA2 (802.11i) which can either use legacy RC4 with a key scheduling workaround or substitute the AES algorithm instead.

Imagine for a second that you have a pacemaker. Many of these devices are designed to have bi-directional communication such that any adjustments to the device itself don't require cutting you open again. There are a number of ways that these can be

FOOTNOTES

[1] Arjen K. Lenstra JPH, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, and Christophe Wachter. Ron was wrong, Whit is right2012. Available from: http://eprint.iacr.org/2012/064.pdf.

[2] Fluhrer S, Mantin I, Shamir A. Weaknesses in the Key Scheduling Algorithm of RC4 Selected Areas in Cryptography. In: Vaudenay S, Youssef A, editors.: Springer Berlin / Heidelberg; 2001. p. 1-24.

adjusted, but in this scenario let's say it uses a near field communication technique (i.e., communication is designed to happen within a few inches) to adjust the timing of the pulses delivered to your heart. Without proper entropy, the pacemaker could be adjusted without consent simply due to weak encryption and key generation.

Not every situation is quite this dire, but go back and think about the fun that bad guys could have with some of these embedded systems I described earlier. From a nuisance like remotely turning off appliances to stealing water or power, we must have better control of these systems to prevent misuse. If encryption is one of those controls, security starts with great entropy!

About the Author:

Branden R. Williams, CISSP, CISM, has been making a name for himself in the Information Technology and Security arena since 1994, as a high school Junior. Now, a graduate of the University of Texas, Arlington earning his BBA in 2000 with a concentration in Marketing and the University of Dallas, where he earned an MBA in Supply Chain Management & Market Logistics, in 2004, Williams is sought after as both an Adjunct Professor and Information Technology & Security Strategy Leader in the corporate world.

Williams regularly assists top global retailers, financial institutions, and multinationals with their information security initiatives. Read his blog, buy his book, or reach him directly at http://www.brandenwilliams.com/.

TEL 214 727 8227
FAX 214 432 6174

BLOG brandenwilliams.com

EMAIL brw@brandenwilliams.com

brandenwilliams
SECURE BUSINESS GROWTH