White Paper:

Ending the Entropy Drought

February 2018

All questions and enquiries regarding this white paper should be directed to:

John Lister
Director of Cyber Security
jlister@cognitiocorp.com

Table of Contents

Ending the Entropy Drought

Executive Summary

Federal agencies are modernizing with cloud and mobile technologies, yet both will present new security challenges. One such challenge is the ability to handle the magnitude of requests for good random numbers, which are required for most security services to function correctly. And a high amount of entropy – the degree of randomness in a system – is needed for strong, truly random numbers. Over the past 18 months, several low-entropy vulnerabilities were discovered and, in some cases, exploited on mobile devices, virtual machines (VMs) and cloud providers. Organizations need to consider adding a standard service that provides a true random number generator with an endless supply of entropy.

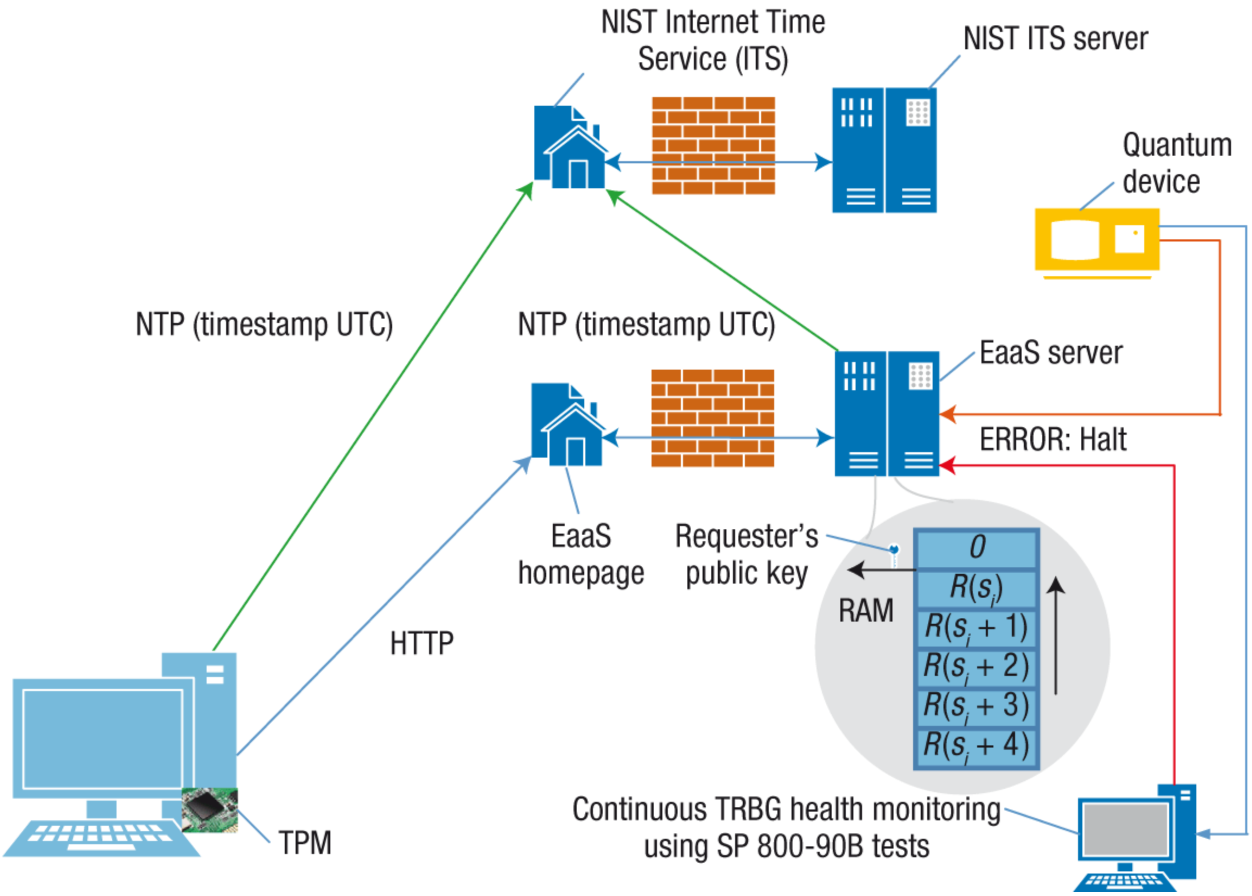

Figure 1: NIST Entropy as a Service Architecture

Challenges of Low Entropy

Is it time to consider adding another critical service for the security of your organization’s infrastructure, applications and services?

Low entropy is a problem that can affect an organization’s mobile devices, endpoints, and network gear, including infrastructure in the cloud. In a presentation at the 2017 Black Hat cybersecurity conference in Las Vegas, iSec Partners researchers Alex Stamos, Andres Becherer and Nathan Wilcox discussed using many VMs needed for random numbers and their lack of entropy, leading to the VMs having much weaker encryption for file storage and communications.

Security professionals know the importance of random numbers in the creation of encryption keys. Random numbers are a cornerstone of protection in many common security and encryption services, such as website communications, Wi-Fi connections, or Skype calls. They're also important in some not-so-common services such as address space randomization, which helps prevent the exploitation of memory corruption vulnerabilities in endpoints and mobile devices. Random numbers are also used in several modern applications for statistical modeling, big data analysis, and gaming.

How do most IT products and services get random numbers? The answer is a bit complicated, and it depends, though most devices and applications get random numbers from their internal random number generators. In these cases, the services rely on a pseudo random number generator (PRNG) or cryptographically secure PRNG (CSPRNG). PRNG needs a high entropy seed key to properly generate random numbers, and if a particularly high number of requests come in, they often need high entropy seed keys to reseed the PRNG.

This poses problems. For example, Linux servers retrieve chunks of bits from an entropy pool; hitches occur because the machines are not generating enough unique raw information to create entropy for the pool, and the pools run out of entropy for creating seed keys. A PRNG that uses AES encryption seeded with a single full-entropy 256-bit key has a cryptographic strength of 256 bits. This means that on average, 2^{128} guesses would be required to “crack” the PRNG’s output. However, if the initial seed’s entropy was less than 256 bits – say, only 10 – then theoretically the output could be cracked in 2^{9} guesses. The problem with low-entropy random number generators is that it’s easier to guess the numbers’ states, leading to attacks on your organization and potentially gaining access.

Benefits to High Entropy

Start taking proactive steps to mitigate a low-entropy security vulnerability. The problem of low entropy will continue to grow as more applications and security services request random numbers. Two considerations need to be made when solving this problem: the way an organization gets high-quality entropy, and how it’s deployed.

A true random number will have maximum entropy and be perfectly unpredictable, which is ideal for any organization. Several products on the market today offer a high-speed true random source.

A High Entropy Service

An organization should consider a service model to deploy entropy. Not unlike a network time service architecture achieved through Network Time Protocol (NTP) services, a single- or multiple-distributed high-speed trusted entropy service should be deployed across an organization to provide an endless pool of true random numbers. Rigorously tested products like “qStream” from Quintessence Labs (QLabs) can enable an entropy service by delivering 100% entropy at 1Gb/second, enough to meet the needs of most organizations.

qStream uses quantum physics to generate an unlimited stream of entropy that provides true random numbers for a PRNG to use as seeds. QLabs also has an effective open key management server that’s compliant with KMIP and FIPS 140-2 level 3. Their robust APIs permit users to manage keys from web interfaces, elsewhere in the cloud, and other parts of the organization. And smart client applications make deployment of entropy and key management simple.

As a bonus, entropy can also be used by non-security applications when true random is needed for data analytics and statistical modeling.

Examples of Poor Entropy Implementations, Vulnerabilities and Security Breaches

Qualys was able to brute-force ASLR (Address Space Layout Randomization) because it used only eight bits of entropy:

https://www.csoonline.com/article/3202031/security/6-points-you-need-to-know-to-keep-stack-clash-from-compromising-your-shared-linux-environment.html

Mobile apps can be brute-forced due to low entropy. There are numerous ways for an attacker to determine if a user exists in the system; a brute force attack is a method to determine an unknown value via an automated process that tries many possible values. The attack takes advantage of the fact that the entropy of the values is smaller than perceived:

https://sdtimes.com/application-development/top-10-vulnerabilities-mobile-applications/

A key Windows 10 defense is 'worthless' and a bug that dates to Windows 8. A system-wide mandatory ASLR (enabled via EMET) has zero entropy, essentially making it worthless. Windows Defender Exploit Guard for Windows 10 is in the same boat:

http://www.zdnet.com/article/key-windows-10-defense-is-worthless-and-bug-dates-back-to-windows-8/

Entropy drought hits Raspberry Pi harvests, weakens SSH security:

https://www.theregister.co.uk/2015/12/02/raspberry_pi_weak_ssh_keys/