The World Is Not Quantum Ready Yet

(BUT IS THAT A CATASTROPHE?)

Disclaimer:

All views in the article are strictly personal. All errors are of course entirely mine.

Context:

There is a risk that advances in Quantum Computers may enable sophisticated actors to compromise the encryption we rely on in our daily lives and businesses. The term "Quantum Resistant" refers to encryption that has a reasonable chance of withstanding attacks from Quantum Computers.

Objective:

This article aims to assess the readiness of various components of the "Technology Stack," considering the ongoing developments worldwide. I have chosen to analyse the Linux-based stack that I use in my private laboratory. However, if you conduct similar research on a Windows-based stack, your findings are likely to be comparable.

Analytical Approach:

The individual components of the stack can be classified into three categories:

- **"Not Ready"**: Very little progress has been made to make these components Quantum Resistant.
- **"Experimental"**: Some code or proof-of-concept solutions are now available, allowing us to begin experimentation.
- **"Workaround"**: This category includes practical decisions to avoid waiting for solutions to be developed.

For each component, we need to think like sophisticated criminals, also known as Advanced Persistent Threat actors (APTs). We will assess the Tactics, Techniques, and Procedures (TTPs) that these APTs would need to exploit traditional encryption using Quantum Computers.

Based on this assessment, I will classify the dependency of the stack on Quantum Resistant algorithms into two categories:

- **"Critical"**: In the worst-case scenario, we may need to disconnect services.
- **"Low"**: We can likely manage this risk for another five years. This is my personal view, and your level of concern may be significantly higher.

What Can We Do?:

Professionals need to stay updated on the latest developments in Quantum Resistant solutions. We should also pay attention to the final choices being published by NIST and whether other authorities, such as ETSI in the EU or Japan, will adopt them without modifications.

I coined the term "Cryptoagility" a few years ago to describe an organisation's ability to replace its cryptographic algorithms. The time has come to start practising Cryptoagility. For this, we must understand the use of cryptography in our technology stacks to create a proper inventory.

We should also begin experimenting with various proofs-of-concept and hybrid algorithms in a test environment so that we can eventually update the technology stack in the production environment.

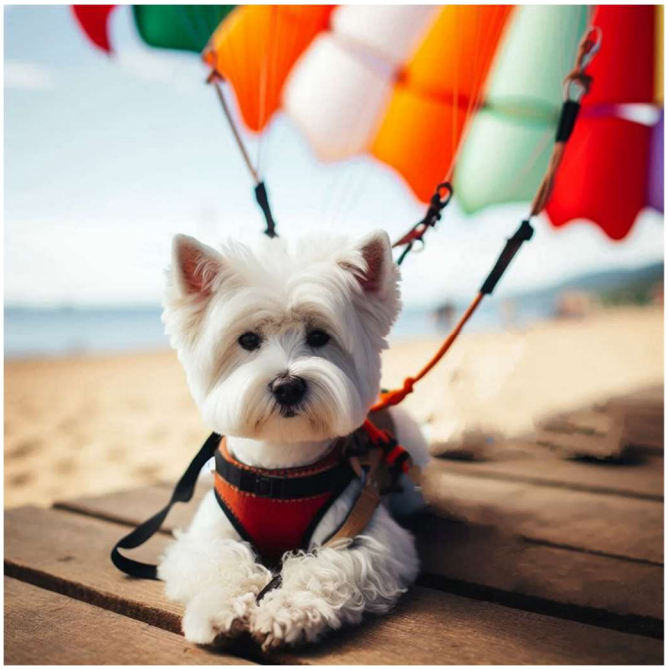
Conclusions:

The Linux technology stack will require significant changes at almost every layer, including the kernel, core libraries, and commonly used applications. This is relevant to both internet-facing infrastructure and back-end systems (e.g. backup systems).

While this may seem doom-and-gloom, I believe there is hope. Where dependencies are classified as "**critical**," I observe some progress in developing Quantum Resistant solutions that have reached the "**experimental**" stage, or we have identified viable "**workarounds**."

For components that are "**not ready**," I find that the TTPs involved would be extremely complex, and the potential rewards for attackers are relatively low; therefore, I would classify these dependencies as "**low**."

So hopefully what looks like a jump off the cliff, the parachute of Cryptoagility and Quantum Resistant algorithms would help us land safely.



Stay Quantum Resistant!

DMs are always open.

Santosh Pandit

London, 5 November 2024

Table A: Post Quantum Readiness of the Linux Stack

Stocktake on Post Quantum Risk to the Linux Technology Stack				
Component Type	**My Preference(s)**	**Status**	**Dependency**	**Comment**
Kernel	Linux Kernel 6.12 (Mainline) Linux Kernel 6.11 (Stable) Linux Kernel 6.6 (LTS)	**Not Ready**	**Low**	See OS and Cryptographic libraries. Apps that require kernel embedded cryptographic support (e.g. Ipsec) may not be quantum safe.
Operating Systems (OS)	Debian Ubuntu Archlinux Fedora	**Experimental**	**Low**	Watch: QUBIP, FEDORA, and RHEL.
Core Libraries and Public Key Infrastructure	OpenSSL 3.3.2 BoringSSL GnuTLS 3.8.4 Libgcrypt 2.4.6 MbedTLS 3.6.2	**Experimental**	**Critical**	Watch: OQS (liboqs, oqs provider and OQS-BoringSSL PQ OpenSSL and PQCrypto-Lib OpenSSL's "providers" Cloudflare about connection from the edge to your server.
Data-at-Rest Encryption	LUKS / dm-crypt (cryptsetup 2.7.5) eCryptfs (111)	**Not Ready**	**Low**	Rely on other security layers including physical security. OR use Veracrypt (see below).
Backup and Storage Encryption	Veracrypt (1.26.15)	**Workround**	**Low**	I already use long, complex, random key and THREE ROUNDS (Serpent, Twofish, AES) and feel reasonably safe.
File and Object Storage	Ceph 19.2	**Not Ready**	**Low**	Rely on other security layers including physical security.
Database Encryption	MySQL PostgreSQL MariaDB	**Not Ready**	**Low**	I would use Mutual TLS for now and restrict access path. Watch: PQC Keyfactor Lab and other similar experiments.
Package Signatures	APT repository verification	**Not Ready**	**Low**	I would try to secure the path to the repository. Therefore even if a quantum computer is used to calculate the private key from the public key; where is the criminal going to upload their malicious package?
Key Exchange Algorithms	Diffie-Hellman, RSA	**Experimental**	**Critical**	Experimental PQ KEMs (Kyber, NTRU) in hybrid solutions for key exchanges.
SSH (Secure Shell) and SSH based transfers e.g. WinSCP.	OpenSSH 9.8p1	**Experimental**	**Critical**	PQ algorithms (e.g., NTRUEncrypt, Kyber) are available in experimental SSH branches via Open Quantum Safe project.
IPsec	StrongSwan Libreswan	**Experimental**	**Critical**	Watch: Juniper and other IPSEC based VPN providers.
TLS (Transport Layer Security)	Apache NGINX 1.27.2 cURL wget	**Experimental**	**Critical**	Watch: some PoCs for nginx OQS.
VPN Software	OpenVPN WireGuard	**Experimental**	**Critical**	Watch: Some vendors seem have developed PQ Wireguard. Research is required. I use key rotation and psk.
DNS Security (DNSSEC)	BIND 9 Unbound 1.17	**Not Ready**	**Low**	I do not think the world will be ready for another 5 years, however I am less worried as the TTP required is extremely complex and the value to be gained seems minimal. I could be wrong!
Email Protocols (SMTP, IMAP, POP3)	Postfix Dovecot	**Not Ready**	**Low**	For now, I am happy with the hardening based on TLS1.3 / AE256. Watch: Limited experimentation; some TLS-based PQ KEMs available.

Analysis as at 4 Nov. 2024