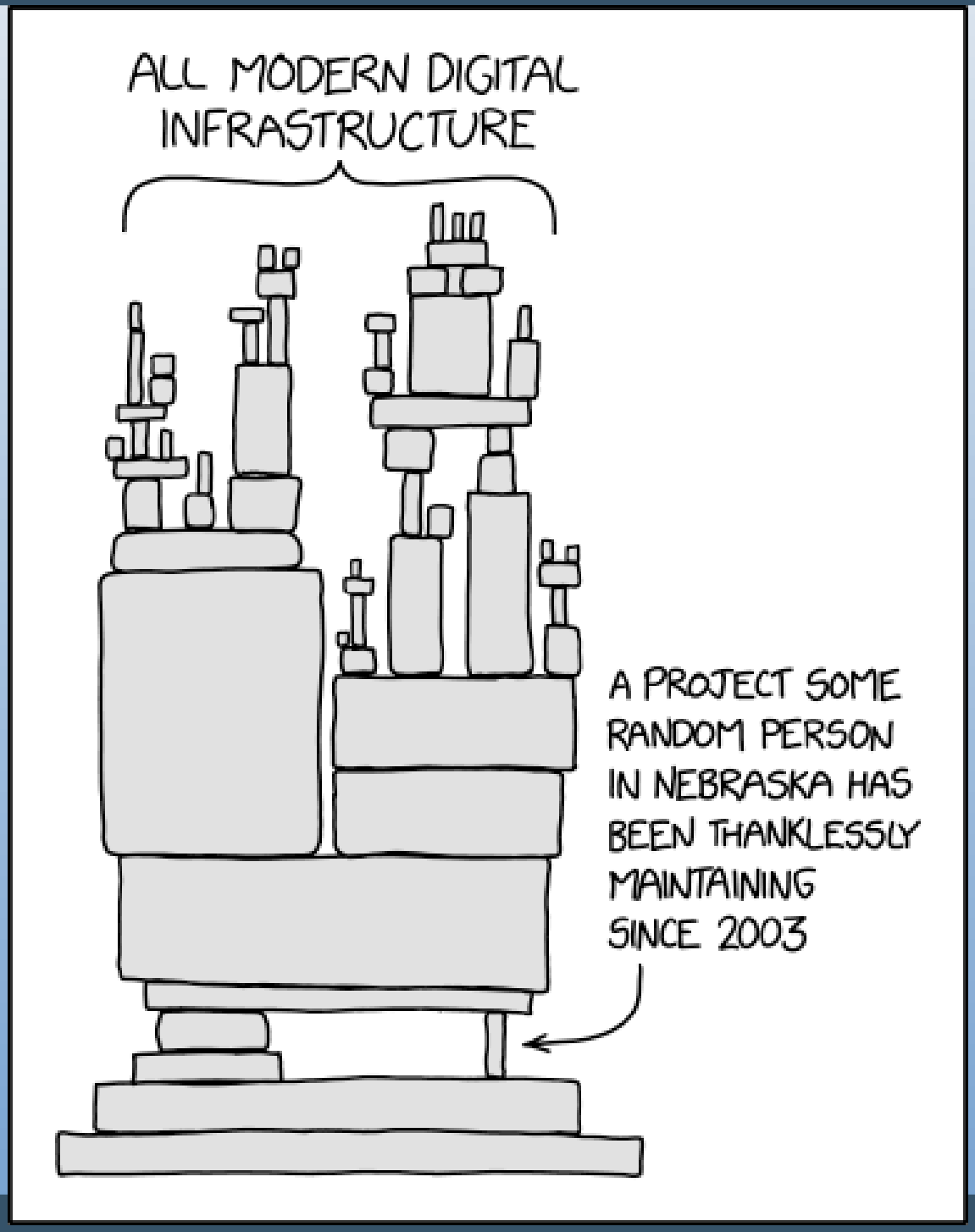
The XZ Utils Backdoor
(CVE-2024-3094)

Xavier Belanger
June 2024



tl;dr

https://xkcd.com/2347/

What are the XZ Utils?

- It’s an open-source library for data compression, using the Lempel-Ziv-Markov chain algorithm (LZMA).
- xz is the main command line tool, used to compress and decompress files; liblzma is the code library.
- This project is mostly managed by Lasse Collin.

Timeline

- **2021**: the JiaT75 (Jia Tan) account is created on GitHub.
- **2022**: Jia Tan starts submitting patches to the xz project. Later this year, he gets more involved with the project and gains the role of maintainer.
- **2023**: The first pieces of code related to the exploit are introduced in xz by Jia Tan (in order to dissimulate the actual backdoor later on).
- **2024**: Changes are made to have control over a subdomain hosting XZ files. Then the backdoor code is added.
- Andres Freund discovers the backdoor on March 29, and alerts the community.

Additional facts and context

- Jia Tan was added as a member to the XZ project after other people asked Lasse Collin to do so.
- After investigation, it appears that those individuals are not involved with the open-source community.
- Once the backdoor was in place, Jia Tan reached out to some Fedora and Ubuntu maintainers to make sure that the “new” version of XZ Utils gets added to the future releases of those distributions.

Technical review

- The backdoor code *is* sophisticated.
- The backdoor starts by affecting the lzma library. This is included in systemd-notify.
- A patch is used by various distributions to modify OpenSSH to work with systemd-notify.
- Once OpenSSH is infected, the backdoor is active and in place.
- This is a very crude and rudimentary summary; more detailled articles (with code) are available online.

Who is behind this?

- Nobody knows for sure.
- The most certain point is that is the result of a planned attack, with people having resources at their disposal (time and technical knowledge).

Is this a new type of attack?

- No, this is a form of supply-chain attack, where the attack targets a component used to build another one and then include the vulnerability.
- Most well-knowns examples are the Target breach, Stuxnet or the Solarwinds attack.

This is actually pretty old

- Read Ken Thompson's article "Reflections on Trusting Trust" (1983).
- "*The actual bug I planted in the compiler would match code in the UNIX "login" command. The replacement code would miscompile the login command so that it would accept either the intended encrypted password or a particular known password. Thus if this code were installed in binary and the binary were used to compile the login command, I could log into that system as any user.*"

Conclusion, questions and discussion.

References

- https://tukaani.org/xz-backdoor/
- https://www.schneier.com/blog/archives/2024/04/xz-utils-backdoor.html
- https://www.schneier.com/blog/archives/2024/04/backdoor-in-xz-utils-that-almost-happened.html
- https://boehs.org/node/everything-i-know-about-the-xz-backdoor
- https://www.wired.com/story/jia-tan-xz-backdoor/
- https://www.theregister.com/2024/04/01/xz_backdoor_open_source/
- https://www.bleepingcomputer.com/news/security/new-xz-backdoor-scanner-detects-implant-in-any-linux-binary/
- https://infosec.exchange/@fr0gger/112189232773640259
- https://lwn.net/ml/oss-security/20240329155126.kjjfduxw2yrlxgzm@awork3.anarazel.de/
- https://gynvael.coldwind.pl/?id=782
- https://gist.github.com/thesamesam/223949d5a074ebc3dce9ee78baad9e27
- https://www.cs.cmu.edu/~rdriley/487/papers/Thompson_1984_ReflectionsonTrustingTrust.pdf