Vaccines, Privacy, Software Updates, and Trust

Daniel Kahn Gillmor dkg@fifthhorseman.net[*], ACLU

October 2019

Contents

Abstract

Participation in healthy modern society requires trust, and trust is an increasingly rare commodity. Social systems that rely on collective, prosocial action by individual people to keep each other safe — like vaccinations, contact tracing, and software updates — depend on public confidence that participation in these actions is unlikely to put themselves at risk of harm.

The COVID-19 pandemic has demonstrated that risks of surveillance and control are seen by many as legitimate harm, commensurate with a potentially deadly illness. Protocol designers working on systems to try to mitigate the pandemic

[*]mailto:dkg@fifthhorseman.net

have made decisions that rightly foreground these concerns about privacy while also supporting the public health goals.

But support for privacy is missing in comparable systems in the digital realm. In particular, while privacy-preserving software update mechanisms are known, there are major gaps the most popular software update platforms, and in the IETF's specification work around Software Update for Internet of Things (SUIT). This paper calls for improvements in this space.

Public Health Paranoia

A popular piece of misinformation suggests that a future coronavirus vaccination will contain a tracking microchip[1]. While objectively untrue, the misinformation was catchy and attractive because of people's uncertainty about technology (in this case, medical technology) and fears about overreaching surveillance.

This is also not entirely without precedent or basis in reality. In 2014, the CIA acknowledged its use of a vaccination program[2] as cover for surveillance in Afghanistan. It seems likely that the ensuing public mistrust of vaccination programs due to CIA abuse damaged the fight against polio. And bogus threats of mandatory vaccination[3] have been used this year to attempt to suppress voter turnout.

In the COVID-19 pandemic, fear of overreach or privacy concerns has contributed to failure of traditional contact-tracing in New York[4]:

> . . . only 42 percent of infected people provided the tracers with the name of even a single contact they might have exposed, a level that epidemiologists consider too low for the program to be broadly effective.

Some universities report student views of contact-tracing as "snitching"[5], rather than socially responsible behavior, and respected software projects call out the potentially disastrous risks of automated public health surveillance[6]:

> These apps can be made in a way to fully respect privacy, and to build trust with its users. Unfortunately the majority of the ones introduced are failing to live up to this promise.

Public health professionals are increasingly taken to task to explain why the systems that they put in place are justified, and that they do not create new

[1] https://www.reuters.com/article/uk-factcheck-video-microchip-coronavirus-idUSKBN22R2GS

[2] https://www.nytimes.com/2012/07/10/health/cia-vaccine-ruse-in-pakistan-may-have-harmed-polio-fight.html

[3] https://www.npr.org/2020/10/01/919309649/far-right-activists-charged-over-robocalls-that-allegedly-targeted-minority-vote

[4] https://www.nytimes.com/2020/07/29/nyregion/new-york-contact-tracing.html

[5] https://www.miamistudent.net/article/2020/09/to-snitch-or-not-to-snitch

[6] https://guardianproject.info/2020/04/09/the-promise-and-hazards-of-covid-contact-tracing-apps/

harms. Techniques like contact tracing or vaccines that were met with minimal resistance (or even enthusiastic adoption) in past epidemiological crises are facing pressure. This legitimate resistance puts the public health at risk.

Defending Public Health Through Privacy

Protocol designers have risen to this public health challenge, rapidly designing remarkable systems that mitigate many of the privacy risks to ensure that the protocol is acceptable to a wary public.

For example, Google and Apple's COVID-19 Exposure Notification system[7] (or `G/AEN`) has an explicit top-level goal to defend against surveillance by the providers of the system itself:

> All of the Exposure Notification matching happens on your device. The system does not share your identity with other users, Apple, or Google. Public health authorities may ask you for additional information, such as a phone number, to contact you with additional guidance.

Similarly, the CrowdNotifier[8] proposal for tracing contacts at potential super-spreader events deliberately states (among several other privacy, confidentiality, and security goals):

> No central collection of personal data. The presence-tracing system should not require the central collection of personal data (e.g., name, IP-address, e-mail address, telephone number, locations visited) of people that visit a location. Nor should the system be able to infer location or co-location data of either visitors or notified visitors.

Both of these systems aim to increase public health effectiveness by reducing privacy concerns. They do this by ensuring that each framework has technical limits on how it can be deployed adversarially against its users.

Public Health and Software Updates

Vaccines, contact tracing, and exposure notification have a clear parallel in the digital realm: all modern computer systems that connect to the Internet rely on some form of software updates to ensure system security, to patch against newly-discovered vulnerabilities, and to ensure ongoing functionality. A software update protects a system by inoculating it against a known vulnerability, just as a vaccine builds the body's defenses against a known threat.

Like vaccinations, software updates also protect the inoculated party's neighbors in addition to the inoculated party. If Alice is no longer vulnerable, she can't spread an infection to Bob. Similar public health measures like contact tracing or exposure notification can help not only individuals that may have been exposed,

[7] https://www.google.com/covid19/exposurenotifications
[8] https://github.com/CrowdNotifier/documents

but also their neighbors, if the at-risk individuals can be supported to get tested and quarantine.

Software updates, like regular vaccinations, offer a touchpoint where the regular person depends on a trusted party (their software vendor, or their public health system) to help them – and their neighbors – to avoid a bad situation. But they also offer a point where users can be tracked by their provider, or even attacked by their provider, as with the CIA in Afghanistan.

How do we ensure that responsible public health-like mechanisms for software updates do not face the same level of pushback that we're seeing for public health measures like vaccines and contact tracing?

Software update providers need to proactively take concerns about privacy more seriously so users don't start opting out.

Privacy from Whom?

When considering privacy concerns for users of a software update channel over the Internet, there are (at least) two distinct attackers:

- The software update provider
- A network observer

While protecting privacy against a network observer is likely to be simpler, the real action (and much of the legitimate risk) comes from the update provider themselves.

Regardless of whether they learn the formal legal identity of the user, a software vendor that can reliably link visits from the same user over time might be able to track the user as they move around the network. In addition to identity and location information, the software update provider often gets some level of intimate detail about what is going on on the user's device. For example, the provider might know whether the user has a dating app (queer or straight), a fertility app, or an app associated with a specific religious denomination (calls to prayer, daily bible verses, etc).

The information available to a privacy-invasive software update provider could be used to selectively target specific religious, sexual, or ethnic minorities, or even to ship malicious software updates to certain targets.

Privacy-preserving Software Update Mechanisms Exist

We know that it's possible to provide privacy-preserving software updates, even if they are not used everywhere. The ACLU (including this author) has identified software updates[9] as a critical component of security and privacy in today's information ecosystem.

[9] https://www.aclu.org/issues/privacy-technology/consumer-privacy/how-malicious-software-updates-endanger-everyone

Some examples of useful techniques include:

Mirrored Distribution (e.g., GNU/Linux distros)

Debian[10], openSUSE[11] and other GNU/Linux operating systems have used mirrors for decades when shipping software updates to users. While mirrors were initially implemented for efficiency reasons, they also provide users with a way of retrieving specific software updates that don't reveal any trackable metadata directly to the software update provider. A user who doesn't trust one mirror operator can manually choose a different mirror operator, for example, and the original software provider doesn't see the change at all.

Fetching updates over Tor

The Tor Browser[12] and Tails[13] both encourage users to fetch software updates over the Tor anonymizing network. This provides network-level anonymity to most users, so that the software provider can't track users by their IP address.

Avoiding Standard HTTP Tracking Mechanisms

In addition to fetching updates over Tor, the F-Droid security model[14] notes specific steps taken in the software client updater to ensure that cookie-like HTTP mechanism (including "etags") cannot be abused by the update mirror to track users.

Many Popular Software Update Mechanisms are not Privacy-Friendly

Notwithstanding the examples above, many of the most popular software update systems do not employ robust privacy protection for their users against the update provider.

Microsoft's software update system appears to be tied to the OS license (which is often tied to a specific customer), and the Google Play Store (for Android) and Apple's App Store for iOS both identify the user (or at least the user's device) to the provider when checking for updates. MacOS's software update mechanism is tied to the user's iTunes account as well.

While it's possible that the providers behind these systems (Microsoft, Apple, and Google) will avoid using use these troves of information against some subset of their users, we have no technical way to hold them to these promises. The mechanisms involved offer no strong guarantees.

[10] https://www.debian.org/mirror/
[11] https://mirrors.opensuse.org/
[12] https://torproject.org
[13] https://tails.boum.org
[14] https://f-droid.org/en/docs/Security_Model/

`G/AEN` Depends on non-Privacy-Friendly Software Update Mechanisms

Despite the claims that the `G/AEN` public health system makes about not collecting user identity or network location information, users of neither major mobile platform can make use of `G/AEN` without shipping deeply invasive details to the platform provider.

The iOS `G/AEN` implementation is tied to the operating system itself, and the operating system includes an automatic software update mechanism tied directly to the device and user identity. So users of `G/AEN` on Apple hardware are by default already sharing privacy-sensitive information with Apple.

The Android `G/AEN` implementation appears to depend strictly on Google Play Services, which itself ships a tremendous amount of user-specific detailed information back to Google multiple times a day, as documented by Leith and Farrell[15].

There are no other widely-available interoperable `G/AEN` implementations, so the laudable privacy-protecting work done for `G/AEN` appears to be shortchanged by its reliance on privacy-unfriendly software update mechanisms.

Planning for Privacy in Software Update Protocols

The COVID-19 pandemic has demonstrated that there is a relationship between the efficacy of public health measures and their impact on end-user privacy. As the general public becomes aware of privacy and surveillance risks from public health infrastructure, some will turn away from it, or try to defeat it. The same fate may await software update mechanisms, which would be a security and privacy disaster for the Internet.

As technologists, designers, and maintainers of infrastructure, we need to demonstrate our commitment to ensuring that these channels cannot and will not be used adversarially against their users.

The IETF Has Not Prioritized Privacy in Software Updates

While the IETF includes active work on software updates, the IETF's Software Update for Internet of Things (SUIT) working group has failed to prioritize privacy. Though the report from the 2016 IoTSU workshop[16] mentions privacy as a concern, it ultimately says:

> The proposal from the group was to introduce a minimal requirement of not sending any new identifiers over an unencrypted channel as part of an update protocol.

The SUIT charter[17] doesn't mention privacy at all, and none of the ongoing SUIT drafts even contemplate the software update provider as a privacy adversary.

[15] https://www.scss.tcd.ie/Doug.Leith/pubs/contact_tracing_app_traffic.pdf
[16] https://tools.ietf.org/html/rfc8240
[17] https://datatracker.ietf.org/doc/charter-ietf-suit/

Identifying Principles of Privacy-Preserving Software Updates

The sections above identify some privacy-preserving mechanisms for software update that are already deployed in some contexts.

More research is needed on the efficacy of these measures, as well as new mechanisms that might protect users against privacy risks from their software providers.

As a technical community, the IETF should design, study, and encourage deployment of these systems where possible.

Identifying Blockers of Privacy-Preserving Software Updates

In addition to the malicious goals of surveillance and targeted attack on users, non-malicious business pressures can also result in non-privacy-friendly software update mechanisms.

For example:

- Payment for software is easiest to implement by tracking users and their payment information.
- Accounting and “leaderboards” for app stores often work by tracking individual installs of an application.
- Telemetry, bug reporting, and other feedback mechanisms can be integrated into software updates, and offer a significant surface for privacy violations.
- Many simple software update schemes can be more efficient if information is cached on both sides of the connection. This sort of state can be used to track the user.
- Some vendors want to be able to shut down systems remotely (“mobile device management” or theft defense) and integrate this functionality into their software update mechanisms.

While the naïve implementation aimed at these goals often has privacy risks, clever protocol design and systems analysis can provide alternate implementations that might arrive at or near the same goals. For example, Prio[18] may offer a privacy-friendly solution to certain forms of telemetry needs, and payment systems like GNU Taler[19] and ZCash[20] may offer privacy-preserving payment schemes, though integration of these schemes with software update platforms is still an open question.

[18] https://crypto.stanford.edu/prio/
[19] https://nlnet.nl/project/GNUTaler/
[20] https://z.cash

Conclusion

Both biological and informatic “public health” systems depend on network effects, where adoption rates are a prime factor in their efficacy. But concerns around privacy and surveillance may limit adoption of these systems, as people become unsure of the tradeoffs that they are being asked to make.

Systems designers need to take these concerns seriously in ways that we have not yet done. In particular, popular software update providers need to adopt more privacy-friendly mechanisms that are already available, and more research needs to be done to identify risks and potential mitigations.