Tricky installers and bundled Adware (Fireball)

Rishi Krishna

CS 3604 (CRN 83358)

compiled using typst. it's like LaTeX with fewer backslashes.

1. Description

In around 2017, a piece of adware/malware was discovered independently by several thread security labs to have infected hundreds of millions of users[0]. It was called 'Fireball', and distributed by purportedly legitimate Chinese digital marketing firm Rafotech.

This adware came silently bundled with real software installed by users – unbeknownst to them, once they installed it their computer began showing them injected advertisments, the money for which was going to Rafotech. These advertisments were not obviously caused by some sort of malicious program – they took the place of any other ad one would see when browsing.

The method of transmission, willful installation by the end-user while trying to install some other software, is a big problem that affects the average computer user. Deceptive installer tricks, like adding a bunch of pages for the user to click through then stealthily making one of these pages agreement to installing this adware [1], mean that despite technically agreeing, the user was fooled.

2. Relevance

This situation is specifically relevant to the unit **Behavioral Design**, and the topic **Persuasive and Deceptive** for the very reason discussed above: by the strict letter of the law, these "trick installers" do cause the user to consent to adware being installed on their machine. In reality, though, it's the deceptive design of the process that fools them into doing something they otherwise would not have.

There are broader tie-ins with concepts in Computer Science, like the strict legality of Terms of Service and License Agreements, where they usually aren't read by the user but still might have some legal standing in exempting the company for negative behavior. In this situation, accidentally clicking install is analogous to clicking "I have Read & Agree with the Terms of Service".

3. Crux

On one hand, consumer protection laws for software would probably limit a lot of other prevalent industry practices, like running active telemetry in the background of programs, and later selling the data, or using software clients to make network requests for niche purposes. Enforcing that what the software user sees must encompass all the action that occurs, could set a tricky precedent that hinders modern software development processes.

On the other hand, enforcing such a law would also provide consumers peace of mind when installing software.

4. References

[0] https://blog.checkpoint.com/research/fireball-chinese-malware-250-million-infection/

[1] https://www.researchgate.net/profile/Adrian-Tobias/publication/318732750_Adware_Nuisance_or_Espionage_Agent/links/597ab5714585151e35a2dc87/Adware-Nuisance-or-Espionage-Agent.pdf