Subject: User-Trusted Computing Definition
Posted by connie on Tue, 13 Feb 2018 19:28:36 GMT
View Forum Message <> Reply to Message

UTC Definition

For the purposes of this definition, an anti-feature is a programmed behavior which is detrimental or outright malicious to a computer system and/or its user, and which can be used by hostile parties to exploit the user and/or the computer system. Anti-features can be distinguished from exploitable side-effects because the latter are challenging or impossible to mitigate, regardless of circumstances, while anti-features can be easily removed by at least some parties other than the user.

A computing system or component is UTC when it is designed to act in full agreement with the user's will, to the extent that is technically possible. A UTC system

1. Should not contain any anti-features.

2. Should not endorse or suggest any computing components with known anti-features.

3. Should not make it difficult to detect and/or remove an anti-feature when one creeps in by accident or is introduced by a hostile party.

UTC Implies Free Software

A software component must be free in the sense used by the Free Software Foundation in order to be UTC.

Freedom 0 is required because any behavior which arbitrarily prevents a user from running a program is an anti-feature.

Freedom 1 is necessary to detect anti-features.

Freedom 2 is folded into Freedom 3.

Freedom 3 is required to remove anti-features. Software development is a collaborative process by its very nature, and the only way a typical user can afford to modify software is by joining a large community of users and developers. Within that community, there should be no barriers to distributing copies of the software, either modified or unmodified, either in source or in binary form. The reason for that is any form of censorship at the development stage can and will be used by censors to insert and/or protect an anti-feature.

Not Just For Software

When we talk of computer systems, we are not talking merely of the software component, or even the soft+hardware components. These and other components of the process we call computing can be evaluated with respect to UTC either individually or as a whole. In general, we can and often should take a step back and look at the entire computing ecosystem in question. If we

evaluate a software package, for example, we can consider a wide variety of factors, included but not limited to licensing, documentation, intended/actual platform, intended/actual audience, and everything up to and including the political process within the community of users and developers. We are forced to take this very holistic approach because none of these factors can guarantee a relief from anti-features when used in isolation.

Specifics by Examples

Anti-features Versus Side-effects

If a hand-held device wants to use a wired or a short-range wireless network, then it will leak its physical location to the network infrastructure. This behavior can be exploited by third parties, starting with the network provider, but it is unclear how to mitigate it: the physical location does not have to be reported, since the network equipment knows its own location, and knows that the connected device is very close by. This is what we would call an unpleasant side-effect.

On the other hand, a programmed feature which reports the exact location periodically over the network to a third party can be seen as an anti-feature, as long as significant difficulties arise when a user attempts to get rid of it without losing any other functions provided by the computer system.

Suggesting and Endorsing Anti-features

When a user opens a Web browser and uses a third-party, general-purpose search engine, the user assumes full responsibility for avoiding anti-features. Nothing in these search results should be interpreted as a suggestion by a UTC system to obtain an anti-feature.

On the other hand, when a Web browser (like Firefox) suggests non-UTC plugins via its own configuration interface, it makes a definite endorsement, which makes it non-UTC. A trivial solution would be to supply the same plugins via a regular Web page, along with an explicit un-endorsement of this software compilation, or at least the part of it which is known to be non-UTC. Of course, an even better solution would be to purge the non-UTC plugins.

Holistic Approach

AdBlock Plus

Licensing alone cannot stem the anti-feature creep. As the famous example of AdBlock Plus have shown, a permissively licensed piece of software can in fact introduce an anti-feature and start exploiting users, even though in theory this kind of behavior is unlikely, due to the possibility of forking, which is guaranteed by the license. Yet as it stands, AdBlock Plus is free software, but not UTC, having included an advertising-related anti-feature. A major role in this evaluation is played by the facts which have nothing to do with licensing.

What matters in this case is the ongoing presense of an anti-feature, which enables the exploitation of users by advertisers, and therefore also by the lead developer of ABP, who is getting paid by advertisers. And even though this anti-feature has been removed in multiple forks of ABP, there is no practical way for ABP users to fix ABP itself. This is in part because the

development is controlled by a single person, who benefits from the anti-feature directly, and the history of alleged corruption erodes the hope that frequent software updates will not re-introduce the same anti-feature, or perhaps a different one. Besides, ABP is also afoul of UTC(3), since this software is designed to make the detection and the removal of the anti-feature harder than it should be, by making the malicious setting the default.

Signal

Another example of a permissively licensed software which is not UTC is Signal app produced by Open Whisper Systems. Signal is free software but it comes with an easy-to-miss anti-feature: it fails to protect the privacy of conversations due to specific design choices made by its developers. According to their own documentation, in order for Signal to function, the encryption credentials must be generated and set up on a system known to be fully compromised by a user-hostile party (users get to pick between Google and Apple), so users are goaded towards non-UTC systems, which violates UTC(2). As a direct result, all encryption keys are made available by design to untrusted parties, and OWS is definitely aware of it, but is very careful not to mention it to its users. Instead, they claim falsely that no one can decrypt the private conversations, which is a definite effort to conceal the anti-feature, violating UTC(3).