Proskauer

When Smart Contracts are Outsmarted: The Parity Wallet “Freeze” and Software Liability in the Internet of Value

Blockchain and the Law Blog on **December 22, 2017**

The recent Parity wallet “freeze” provides yet another example of a coding vulnerability in a smart contract (rather than a flaw in the underlying blockchain or cryptography) resulting in an exploit that compromises cryptocurrency worth millions. It again highlights some of the pitfalls of insecure code in the context of digital assets and raises questions regarding the extent to which software developers can be held liable to its users for losses suffered due to those oversights. As blockchain-related software that serve as storage vaults for digital assets continue to proliferate, it will be interesting to see how industry standards and the existing software liability regime in the U.S. and other jurisdictions evolve to reflect the critical role of secure software in the “Internet of Value.”

The Parity Wallet “Freeze” Explained

Parity Technologies made available, on an open source basis, multi-signature software “wallets” that users could use to store the keys to Ether cryptocurrency, which are necessary to use Ether. Those multi-sig wallets were smart contracts built to run on the Ethereum blockchain and, unlike standard Parity “accounts” or other cryptocurrency wallets, required more than one digital signature (private key) before Ether associated with them are approved to be transferred.

On November 8, Parity Technologies announced that “devops199”, a user of the prominent web-based software development platform Github, had exploited a software vulnerability in Parity’s multi-sig wallets, resulting in Ether tied to over 500 multi-sig wallets, then valued at over $150 million, becoming completely unusable. Among impacted users were many high-profile blockchain startups that used Parity’s wallet platform to raise funds through initial coin offerings (ICOs). This marked the second time this year that Parity’s wallet software has been compromised, with the prior time being July 19, when hackers exploited another software bug to steal over $30 million in Ether.

On the Parity platform, every multi-sig wallet created after July 20, 2017 was made up of two components: (1) a “light-weight ‘stub’ smart contract,” which is uniquely deployed for each multi-sig wallet instance, and (2) a single “much heavier ‘library’ smart contract” deployed by Parity on July 20 to fix the original vulnerability that made the earlier July 19 hack possible. Only one instance of the library smart contract existed on the Ethereum blockchain, and each and every stub smart contract was inextricably linked to and dependent on it for core functionalities, such as the ability to transact the cryptocurrency “stored” in the wallet.

Parity deployed the library smart contract on July 20 but did not “initialize” it, meaning that technical “ownership” of it was still up for grabs, even though, in its uninitialized form, the library was already powering the Parity multi-sig wallets. In its post-mortem report published one week after the incident, Parity explained that devops199 was the first to call the initialization function of the library smart contract, thereby establishing devops199 as its owner. Devops199 was therefore able to exercise its self-destruct function, which effectively deleted it, rendering the linked multi-sig wallets incapable of transferring any Ether they stored the keys to.

The source code for Parity’s software (and for many other blockchain technologies) is publicly available on Github. In the aftermath of the incident, it was discovered that a Github user had previously pointed out the vulnerability in early August and recommended that Parity run the initialization function to prevent precisely what ended up occurring in November, but Parity had failed to take heed.

The frozen multi-sig wallets were designed without giving their owners the ability to link to another library contract in the event that the originally-linked library contract became inactivated or deleted. At this point, it seems the only way to wholly remedy the situation and re-enable access to the impacted cryptocurrency is to bring back the deleted library contract, which would require altering the Ethereum blockchain through a “hard fork”—a drastic measure. This has been done in the past in limited circumstances for high-profile security breaches, such as last year’s hack of the DAO. In a hard fork, the Ethereum network has to agree, through its consensus mechanism, to deviate from the existing blockchain ledger and implement the intended change. However, in a decentralized platform like Ethereum, it is difficult to reach the high threshold of consensus required to adopt such a fork. In the case of the DAO, the hard fork resulted in two separate Ethereum blockchains, with those who refused to adopt the fork establishing a separate blockchain dubbed “Ethereum Classic.”

Security failures like Parity’s arise from flawed smart contracts written on top of the Ethereum blockchain, not a problem with the Ethereum blockchain itself. Already, many in the Ethereum community have voiced concerns about the impact of repeated hard forks in the Ethereum blockchain to rectify software incidents, citing its potential to fracture the blockchain community and undermine confidence in the immutability of blockchain as a technology infrastructure.

Software Liability in the Internet of Value

Is Parity Technologies liable to its multi-sig wallet users for the inaccessibility of their frozen Ether?

Parity’s wallet software is made available free of charge on an open source basis under the commonly-used GNU General Public License v.3.0. Parity users are required to affirmatively click to accept the license terms before installing the executable form of the Parity wallet software, a process that U.S. courts have generally deemed to create an enforceable contract. Parity users also have the option of building a Parity wallet from the source code made available by Parity on its Gitbhub page, which clearly states that use of the code is subject to the GPL.

As with many “clickwrap” license agreements, the GPL states that the licensed software is provided “as is” and broadly disclaims all warranties, whether express or implied, as well as any and all liability of the developer to the user relating to the licensed software.

In the U.S., courts have been quick to uphold clear and unambiguous warranty disclaimers and limitations of liability in the software context (regardless of whether the software is deemed a transacted “good” that is subject to Article 2 of the Uniform Commercial Code), except to the extent that gross negligence, willful misconduct, fraud or strict product liability are established, or the disclaimers or limitations are deemed unconscionable, violative of a statute or against public policy. Any claim under each of those exceptions, however, might be difficult, especially in the case of unintentional software vulnerabilities in free software that are exploited by third parties (as apparently in Parity’s case).

Notwithstanding the GPL’s sweeping exculpatory language, one possible attempt for redress from Parity may be a negligence tort claim. Unlike Parity’s Terms of Website Use “browsewrap,” the broad disclaimers of liability in the GPL do not specifically name “negligence,” and a plaintiff could potentially ask a court to strictly construe the failure to do so to permit negligence claims. But even if a court were to determine that the GPL does not exclude negligence claims, current U.S. case law suggests it might be difficult for a user to prevail on such a claim. An essential element of a negligence claim is establishing that the defendant owed a legal duty of care to the plaintiff, and then violated that duty. In the case of Parity, a plaintiff might argue that, because Parity distributes software designed to control access to and safeguard the keys to valuable digital assets, Parity owed a duty to its multi-sig wallet users to act as a reasonable software developer would to fix a significant known vulnerability that could jeopardize that security, and that Parity’s failure to initialize its library smart contract even after being notified of the vulnerability was a breach of that duty. However, courts may not find that software licensors owe such a duty to its users with respect to software defects, especially when the software is made available free of charge and is accompanied by broad disclaimers of responsibility.

Aside from the challenges associated with claiming damages from Parity, from a practical perspective even a successful claim may be futile, as Parity’s capitalization and insurance coverage may be insufficient for those damages to actually be recoverable.

The Future of Software Liability

Software has increasingly become an integral part of physical products (e.g., the Internet of Things). Now, with the proliferation of cryptocurrencies and tokenized assets, software has also taken on the function of storing and controlling access to highly valuable digital assets, in many cases with no technical remedy for errors. Citing this metamorphosis in the sensitive role software plays, some commentators have called for more avenues to holding software vendors liable for the security and integrity of their code.

As software continues to evolve in the establishment of a widespread “Internet of Value,” it will be interesting to watch the progression, throughout the world, of industry standards, laws and the approach that courts take to assessing the legal liability of software developers for vulnerabilities in their code that result in its users’ loss of value—especially those that provide, in exchange for fees, software designed to store or safeguard cryptocurrencies and other digital assets.

View Original

Related Professionals

- **Wai L. Choy**
  Partner