Topic

Sensitive Software Update Proposal to reduce the “time to exploit” potential vulnerabilities in the network.

Context

Sensitive Software Update Proposal will allow you to create, vote and discuss a network update without disclosing it to the general public through the Proposal, which reduces the opportunity to take advantage of an error that has not yet been fixed on the network by researching diff.

We continue a good policy to avoid Disclosure Critical bug or security issue.

A significant example is Juno which was attacked while the vote on the adoption of the update was taking place

Attackers used a known @CosmWasm

bug to halt @JunoNetwork

just a few hours before they deployed a patched version (c)@assafmo

Global research on time to exploit show that a significant part of vulnerabilities were exploited after

the patch was made public, thus, taking into account the Juno incident, it is clearly unsafe for the vulnerabities patch code as well as vulnerability description be public.

For 2019:

[

699×592 67.4 KB

](https://global.discourse-cdn.com/standard17/uploads/enigma1/original/2X/e/ecfdf3bc4e4f955fd6fa222637424ba557162470.png)

For these non-zero-day vulnerabilities, there was a very small window (often only hours or a few days) between when the patch was released and the first observed instance of attacker exploitation.

Summary

Sensitive Software Update Proposal’s code will be represented as a git patch file (diff with git commit history) in a IPFS, encrypted via the symmetric key, that is then redistributed among specific wallets via secret smart contract, ensuring only specific wallets (and their trusted personnel) as well as proposal’s authors will be able to inspect the contents of the code that is to be applied to the network on successfully passing the proposal.

Current capability is trivially implemented as a part of Cosmovisor

or a different utility for inspecting encrypted diffs

.

[

image

1322×1060 82.8 KB

](https://global.discourse-cdn.com/standard17/uploads/enigma1/original/2X/1/19de82e0439628f0cd47a735fe2ba2c092dacd9f.png)

Variants of realisation

Choosing the best option may require discussion among the community

We see the need for 2 artifacts:

for utility

- make new utility for working with encrypted diff
- fork Cosmovisor to extend functionality to handle Sensitive Software Update

for network

- build smartcontract to publish update to a limited number of people
- write readme guide “how to make Sensitive Software Update” with a description about creating a new Sensitive Software Update massage
- make new type of proposal which will manage access to the encryption keys

The best solution we see is make new utility for working with encrypted diff and smartcontract to publish update to a limited number of people

In the future, it is possible to introduce this idea as part of the work of Cosmovizor to bring automatization and a new type of proposal.

Motivation

It is necessary to introduce privacy into the network update process. We believe that not all data should be exposed by default, and Sensitive Software Update is among them. It is a global security practice to postpone the publication of information about a vulnerability until it is fixed.

We start with secret because there are already the necessary tools. In the future, this solution can be integrated into the cosmos sdk and distributed to all Cosmos blockchain networks, this will help make web3 a safer place.

Motivation behind IPFS as a diff storage

Potentially, vulnerability fix may reach a significant size, which will make it quite cumbersome to store on a ledger, thus usage of a distributed storage protocol with additional encryption is necessary.

Risks

- Low-trust towards the software update from stakers with voting power, but not the access towards the update’s content.
- Remaining risks of validators exposing sensitive update information

Benefits

- Significantly decreasing the lifetime of zero-day vulnerabilities