PHP-TUF

Problem Statement

Many PHP projects have automatic updating as a goal, but scaling a trustworthy infrastructure for builds, distribution, and verification is hard. One of the hardest parts is having a broadly compatible, modern, and secure signing model.

Why TUF?

Python has pioneered work in The Update Framework (TUF), which is a recent graduate into the CNCF. Their security and specification work is rigorous and linguistically portable (JSON, SHA-2, and EdDSA). Because TUF has a well-tested reference implementation in Python, much of the work is done for us: creating signatures, managing repositories, and having test fixtures for our implementation of verification logic.

We may only need to implement a PHP verifier, at least initially.

MVP Proposal

To make the first milestone both useful and easily reachable, scope would initially be:

- Only support verification, no signing or repository management.
    - Signing and repository management is easy with the existing Python implementation.
- No support for in-band root key rotation.
    - The in-band root key rotation is only designed to respond to partial compromise, like one compromised key when “two of four” may be the validation threshold.
        - At least for Drupal, this scenario is less important because our root keys are maintained on HSMs. We are happy to share procedure around this for implementation with YubiHSM 2 and connect people with Yubico.
        - If we did encounter such a partial compromise, we could implement and deploy root key rotation at that time.
    - Excluding this for now has no effect on the rest of the implementation. Likewise, adding in support later will not have cascading effects.
    - This remains desirable to implement in the medium-term, ideally before we need it.
- No support for RSA signatures.
    - Both libsodium (and Paragon.ie’s polyfill for it) support EdDSA. We don’t have a clear path to supporting RSA as broadly.

    - RSA is worse. The only upside to TUF's RSA option is the existence of cloud HSMs like GCP's that support TUF's RSA signature scheme but not EdDSA.
    - TUF's design supports offline roots of trust and rotation of intermediate secrets so rigorously that the importance of a cloud HSM is much lower.
- Support consistent snapshots? Only support consistent snapshots?
    - What is right for PHP projects and controlling complexity?
- Should pass all upstream tests for verification.
    - Except: Fail securely for anything we don't support.

Code Sprint

- 2-5 days of effort toward MVP
- Participation virtually and in-person

Interested People

Name	**Project**	**Location**	**Contact**	**email (based on Slack)**
Benni Mack	TYPO3	DE	bmack on Drupal and CMS Sec Summit Slack	benni@typo3.org
David Strauss	Drupal	SF, US	dts on Drupal and CMS Sec Summit Slack	david@davidstrauss.net
Tobias Zulauf	Joomla!	Cologne, DE	"Tobias Zulauf" on CMS Sec Summit Slack	tobias.zulauf@community.joomla.org
David Jardin	Joomla!	Cologne, DE	"David Jardin" on CMS Sec Summit Slack	david.jardin@community.joomla.org
Oliver Hader	TYPO3	Hof, DE	"Oliver Hader" on CMS Sec Summit Slack	oliver.hader@typo3.org
Michael Hess	Drupal	Ann Arbor, US	mlhess on Drupal Slack and CMS Sec Summit Slack	mlhess@umich.edu

Logistics

- Early June in NYC? Sooner?
- Will probably reach out to Google for space unless there are other proposals
    - Maybe reach out to NYU, given their role in TUF
    - Also have contacts at Princeton
- Several people have access to funding to travel
    - Probably possible to get sponsorship for others

Meeting Agenda and Notes:

https://docs.google.com/document/d/1Kbe_0wtvD6DN2KbPB3sXmfN6KAXen6MJJtvuDfDRAYs/edit#heading=h.29ccxp82o0qz