Dedaub
Security Technology for Smart Contracts

SAP Token Audit for Yesbit/Krawlcat

Date: 23rd October 2020

Dedaub was commissioned to perform a security audit on the SAP token, currently deployed to the Ethereum mainnet at address `0xf6ed276a69270a895d6e419d99dcb5aaa2f3cb4a`, together with its state. The SAP token is a standard ERC20 token, minted by a centralized entity but with a capped supply.

The SAP Token is implemented around recent versions of the standard OpenZeppelin libraries, so it should be considered secure. The implementation is also composed in a very standard fashion. In particular, most of the logic is derived from OpenZeppelin's `ERC20.sol,` last updated at commit `1229c28`. Capping of total supply is achieved by inheriting `ERC20Capped.sol` at commit `0408e51`. There is a low risk of issues given that these libraries have been reused in several other projects. Nevertheless, we also performed thorough code inspection independently of the provenance of the code, as if it had been written from scratch.

Since the technical security of this token hinges almost exclusively on faithfully replicating the intended contract composition, additional checks were performed by decompiling the deployed contract and manually verifying this. In all cases, the method resolution order matches the intended design of the library.

Trust Model

[*This section is included for context, although its contents should already be known to the commissioner of an audit.*]

Users of the token need to be comfortable with the following centralization elements:

- Minting of the token is performed by a single administrator/owner. A fixed cap of 21 million SAP tokens can be minted. At the time of writing ~4.8% of all tokens have already been minted.
- The current administrator is an end-user address, and thus minting of the token is completely centralized.
- No facility for burning/destroying tokens exists.

Critical Severity

No critical severity vulnerabilities were identified

High Severity

No high severity vulnerabilities were identified

Medium Severity

No medium severity vulnerabilities were identified

Low Severity

No low severity vulnerabilities were identified

Lowest/Code/Style/Info

The contract was compiled with the Solidity compiler `v0.6.6` which has some known minor issues. Later iterations within the `0.6.x` major version, such as `0.6.9`, have fewer issues.

One of the issues (`EmptyByteArrayCopy`) was identified just a week ago and is only fixed in the latest Solidity version. The contract is not affected: the issue requires copying zero-length memory arrays to storage, and the contract only does this when setting the name and symbol during construction. Since only the owner sets these fields the compiler bug is not applicable to SAPToken.

Disclaimer

The audited contracts have been analyzed using automated techniques and extensive human inspection in accordance with state-of-the-art practices as of the date of this report. The audit makes no statements or warranties on the security of the code. On its own, it cannot be considered a sufficient assessment of the correctness status of the contract. While we have conducted an analysis to the best of our ability, it is our recommendation for high-value contracts to commission several independent audits, as well as a public bug bounty program.

About Dedaub

Dedaub offers technology and auditing services for smart contract security. The founders, Neville Grech and Yannis Smaragdakis, are top researchers in program analysis. Dedaub's smart contract technology is demonstrated in the contract-library.com service, which decompiles and performs security analyses on the full Ethereum blockchain.