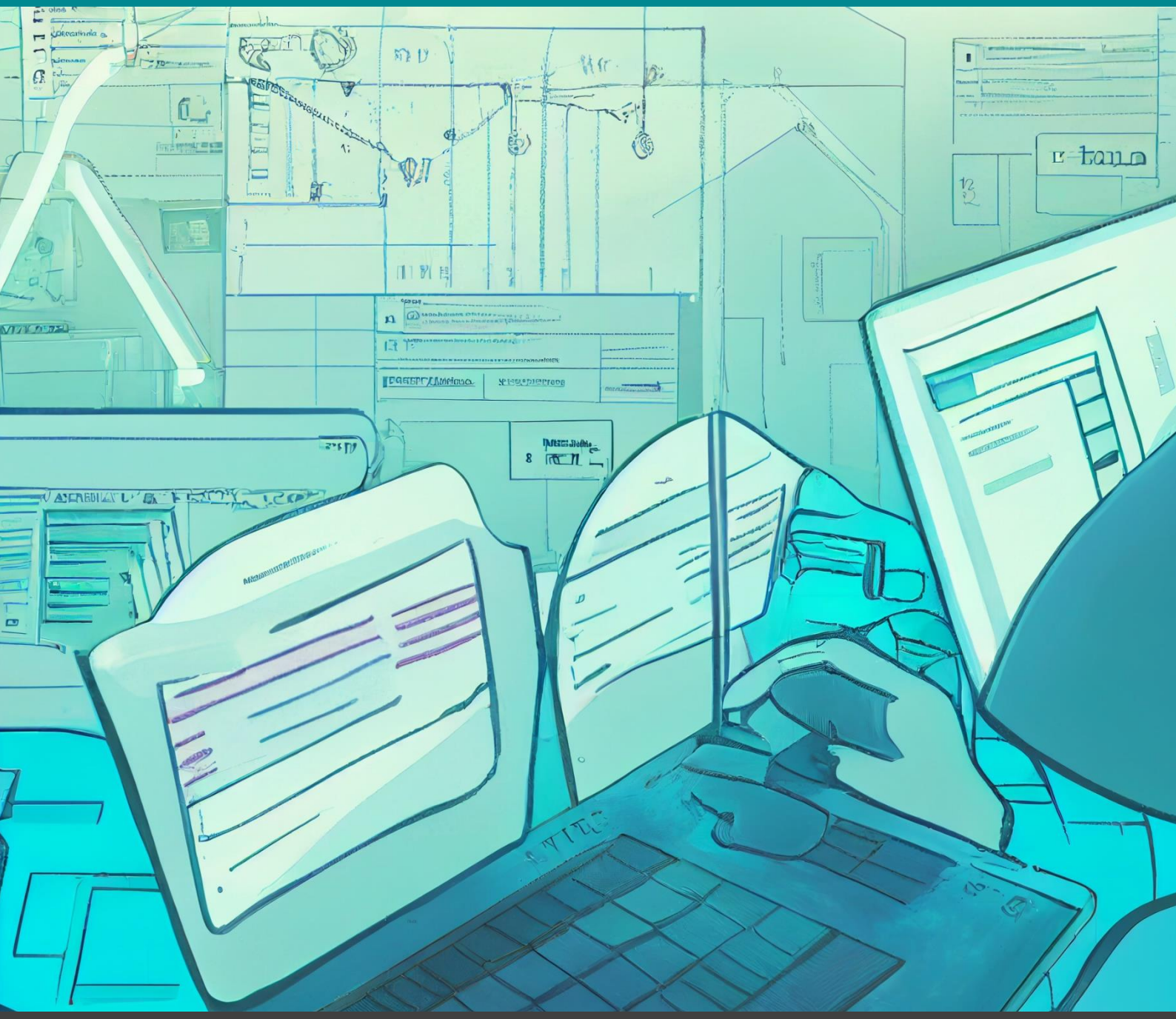
RomyRom



Audit report

Ermine Virtual Miners (EVM) token ERC-1155

Contract address: 0xF07fF1a72E1F71B38F07eA278FB95Dc8eA029889

February 07, 2023

Content

Disclaimer

RomyRom audits various projects where smart contracts are involved. We do not guarantee that the purpose of the project may be fraud, “rug pull”, withdrawing liquidity, selling and transferring the project and its team.

An audit of a smart contract is not an offer, investment advice, or a recommendation to buy tokens.

RomyRom is not responsible for any losses and speculative investments. Use the information obtained from this audit for informational purposes.

Do your own research on the project, documentation, and its social media.

Object under study

Ermine Virtual Miners (ECR1155)

Ermine is developing an ecosystem consisting of virtual mining using virtual miners, a liquidity pool with an exchange, a return system, a referral program, a treasury and interaction with yield smart contracts. EVM tokens are NFTs of the ERC1155 standard. In total, there are 12 categories of NFTs with a pre-limited release. EVM tokens are virtual miners. In the ecosystem, Ermine suggests using the EVM as an NFT to calculate the expected return depending on which smart contract pool it will be locked in. Each NFT category will have a different profit margin, called IPS performance.

Technical characteristics of the ERM token

Ecosystem – Ethereum

Token standard – ERC1155

Name – Ermine Virtual Miners

Ticker – EVM

Contract address - 0xF07fF1a72E1F71B38F07eA278FB95Dc8eA029889

Total supply and number of EVMs minted

Id NFT EVM	MAX Total supply, EVM	Total minted, EVM
0	1,000,000	100
1	500,000	0
2	300,000	0
3	100,000	0
4	50,000	0

5	20,000	0
6	10,000	0
7	5,000	0
8	50,000	0
9	20,000	0
10	3,000	0
11	1,000	0

Verify of smart contracts

Verified contract source code in browser:

https://etherscan.io/address/0xf07ff1a72e1f71b38f07ea278fb95dc8ea029889#code

Contract Name: Ermine Virtual Miners

Compiler Version v0.8.17+commit.8df45f5f

Optimization Enabled: Yes with 200 runs

Other Settings: default evmVersion

Contract Source Code Verified (Exact Match)

Checking hashes (SHA256) of OpenZeppelin libraries

File name	SHA256 Checksum	Conclusion
IERC165.sol	92762629f91532d937e795ceee7391d5e4e9db0ca8eba233da3dd1e95ce9d792	verified
ERC165.sol	d57a87486b2a47da36c0e094aa079d83bd6660ef2d17daa41647b229156461c6	verified
Strings.sol	6d56683b5d732e414339aa2d43d363efba20e87882b22088472af2cc19b6a515	verified
Context.sol	6de5302543723d32c8eaf17becc4525936e16d9c4551455c93d306b9b72c0799	verified
Address.sol	3d33fb7ef4e347f5e24362ea6ae7c9a002de118bfee827123c9741b138a7e0b9	verified
IERC1155MetadataURI.sol	69ee9280731ed78862e8390ead5dafb390b2fe3b79663eed72dda29167fa473b	verified
ERC1155Supply.sol	8b8b46208f0e976416d2dc562127b78f44e381ba07a2b2355050192c18968794	verified
ERC1155Burnable.sol	8d868d02fded94c021d73b4863a7ac1d024c7e62c6937fcd0b13802cbb0e25f6	verified
IERC1155Receiver.sol	7738b943b504a88f433ebe5bd4bbdf00f0f8233f045a01e1afcf27e0c9ab96b6	verified
IERC1155.sol	0dbe712a2532edfe2f8ede273002b1a95b10b45deeab2024ab41e7f4a52e5910	verified
ERC1155.sol	f113cda1133d8ecfe9778b4fee4bb2a3e80a051c3566f6076d6878013bb9901e	verified
Ownable.sol	96a3b09372173d7174fcb0080a97c0cd9abb51cd31e71ecd597d62e0942cb7c4	verified

Owner: 0x18b5d766C8714BA3280403A1fD5e8e1183B707b8

Control modifiers: none

EVM tokens can only be minted by the owner to an external address (assumed Ermine store smart contract) by calling the mintEVMforStore() function. The mint EVM with id 8-11 is available to everyone after 15638400 seconds from the moment the mintEVMforStore() function was called. The minting of EVM with id 8-11 occurs to an external address

(assumed Ermine store smart contract) for you need to call the mintEVMLEforStore() function.

The Ermine Store smart contract has not yet been accepted for audit. The EVM to mint address is not specified in the smart contract.

This place needs special attention.

An unused import was found (not critical)

```
14  import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol";
```

and

```
16  import "@openzeppelin/contracts/token/ERC1155/extensions/ERC1155Burnable.sol";
```

Functions created by the developer

Below is a study of the functions created by the developer

The ErmineVM.sol file includes 7 functions

Function name	Permissions	Description
mintEVMforStore()	available to onlyOwner	The function is executed once when the conditions are met. An external address for EVM mint must be set and the first batch of EVM tokens has not yet been minted before. Upon successful execution of the function, EVM tokens will be mint to the address specified in the smart contract and the timeAddLE variable will set the time from the moment of which it will be possible to mint EVM tokens with id 7-11.
mintEVMLEforStore()	available for everyone	The function is executed once when the conditions are met. It's block time, i.e. block time >= timeAddLE, first batch of EVM with id 0-7 has already been minted, EVM with id 7-11 has not been minted before. Upon successful execution of the function, EVM tokens will be mint to the address specified in the smart contract
burnMyEVM(uint evm_id, uint256 amount)	available for everyone	Burns the number of EVMs with the given ID, if there are enough EVMs with that ID in its balance (>=number). The burnEVM[] array stores the number of burnt EVMs of each ID.
setErmineStore(address _ErmineStore)	available to onlyOwner	The function prescribes the address to which EVM can be mint. You can only enter an address once. The registered address is set as an administrator with limited rights.
setURI(string memory newuri)	available to admins	The function sets the base URI for linking the content with the EVM token ids.
RemoveAdmin()	available to admins	The function deprives the caller of administration rights
uri(uint256 _id)	Available for everyone to read	Read function to view the address of the bound content of a token with a specific id

Warning! Tokens EVM after burning are not subject to recovery and are not sent back for mint. After burning, the total number of EVM tokens will be reduced by the amount in their category id (evm_id).

Risks for the user may be associated with an accidental call to the burn function, which may lead to the irretrievable loss of their EVM tokens.

Don't call any functions, in any contracts, unless you understand what might happen next.

A huge responsibility lies with the owner of the contract when writing the address of the Ermine store when calling the **setErmineStore** function. It is very important not to make mistakes when writing the address for EVM mint. Before EVM is minted, the Ermine Store contract must be checked for errors.

Conclusion

During the study of the Ermine Virtual Miners contract, no errors were found in the source code. The code audit of the libraries used did not reveal any changes made to them. Library file hashes correspond to stock values. Having the burn function available to everyone can create a risk of losing user EVM tokens if it is accidentally invoked.

The function to set the address for the Ermine shop and the function to call the coinage of the first batch of EVM with id 0-7 will be subject to the responsibility of the contract owner.

No vulnerabilities were found. The Ermine smart contract has successfully passed security tests and does not pose a threat when used.

Contacts

https://romyrom.com/en/

https://twitter.com/RRWeb3