InterFi Network

@interfinetwork

SMART CONTRACT SECURITY AUDIT

Horny Dog XXX

AUDITED ON SEPTEMBER 04, 2021

USING INTERFI AUDITING ARCHITECTURE

Summary

Audit:

Auditing Firm	InterFi Network
Architecture	InterFi Auditing Architecture
Smart Contract Audit Approved By	Chris | Blockchain Specialist at InterFi
Project Overview Approved BY	Albert | Project Specialist at InterFi
Platform	Solidity
Audit Check (Mandatory)	Vulnerability Check, Source Code Review, Functional Test
Project Check (Optional)	Website Review, Socials Review, Token Review (Not Applicable)
Consultation Request Date	September 02, 2021
Report Date	September 04, 2021

Risk profile:

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks. According to the smart contract audit, **Horny Dog XXX's smart contract source code has Low Risk Severity.**

For the detailed understanding of risk severity, source code vulnerability, and functional test, kindly refer to the audit. At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before blockchain deployment. Please proceed with caution.

Table of contents

Project Overview

InterFi was consulted by Horny Dog XXX on September 02, 2021 to conduct a smart contract security audit of their solidity source code.

Public information

Horny Dog is the first crypto adult magazine. The erotic market is growing exponentially, everyday. To fulfill this market even more, Horny Dog will provide a monthly based magazine with news about porn stars, adult token projects, blockchain updates and many more.

Information	Horny Dog
Blockchain	No deployment info at the time of audit
Language	Solidity
Contract	https://github.com/interfinetwork/audited-codes/blob/main/HornyDog.sol
Website	https://hornydog.xxx/
Twitter	https://twitter.com/HornyDogBSC
Telegram	https://t.me/HornyDogXXX
E-mail	contact@hornydog.xxx

Public logo

Solidity Source Code

https://github.com/interfinetwork/audited-codes/blob/main/HornyDog.sol

GitHub Commits

Solidity source code committed at: c8297f0546b7977b2740e17165081af6948ec3e3

Audit Scope & Methodology

The scope of this report is to audit the smart contract source code of Horny Dog XXX. The source code can be viewed in its entirety on

https://github.com/interfinetwork/audited-codes/blob/main/HornyDog.sol

InterFi has scanned the contract and reviewed the project for common vulnerabilities, exploits, hacks, and back-doors. Below is the list of commonly known smart contract vulnerabilities, exploits, and hacks:

Category	
Smart Contract Vulnerabilities	❖ Re-entrancy (RE) ❖ Unhandled Exceptions (UE) ❖ Transaction Order Dependency (TO) ❖ Integer Overflow (IO) ❖ Unrestricted Action (UA)
Source Code Review	❖ Ownership Takeover ❖ Gas Limit and Loops ❖ Deployment Consistency ❖ Repository Consistency ❖ Data Consistency ❖ Code Typo Error ❖ Token Supply Manipulation
Functional Assessment	❖ Access Control and Authorization ❖ Operations Trail and Event Generation ❖ Assets Manipulation ❖ Liquidity Access

ECHELON-1 Analysis

The aim of "InterFi's ECHELON-1 Analysis" is to analyze the smart contract and identify the vulnerabilities and the hacks in the smart contract. Mentioned are the steps used by ECHELON-1 to assess the smart contract:

1. Code review that includes the following
   - ❖ Review of the specifications, sources, and instructions provided to InterFi to make sure we understand the size, scope, and functionality of the smart contract.
   - ❖ Manual review of code, which is the process of reading source code line-byline to identify potential vulnerabilities.
2. Testing and automated analysis that includes the following
   - ❖ Test coverage analysis, which is the process of determining whether the test cases are covering the code and how much code is exercised when we run those test cases.
   - ❖ Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts

Automated 3P frameworks used to assess the smart contract vulnerabilities

- ❖ Slither
- ❖ MythX
- ❖ Consensys Mythril
- ❖ Open Zeppelin
- ❖ Solidity Code Complier

InterFi's Risk Classification

Smart contracts are generally designed to manipulate and hold funds denominated in Ether. This makes them very tempting attack targets, as a successful attack may allow the attacker to directly steal funds from the contract. Below are the typical risk levels of a smart contract:

Vulnerable: A contract is vulnerable if it has been flagged by a static analysis tool as such. As we will see later, this means that some contracts may be vulnerable because of a false-positive.

Exploitable: A contract is exploitable if it is vulnerable and the vulnerability could be exploited by an external attacker. For example, if the "vulnerability" flagged by a tool is in a function which requires to own the contract, it would be vulnerable but not exploitable.

Exploited: A contract is exploited if it received a transaction on the main network which triggered one of its vulnerabilities. Therefore, a contract can be vulnerable or even exploitable without having been exploited.

Risk severity	Meaning
! Critical	This level vulnerabilities could be exploited easily, and can lead to asset loss, data loss, asset manipulation, or data manipulation. They should be fixed right away.
! High	This level vulnerabilities are hard to exploit but very important to fix, they carry an elevated risk of smart contract manipulation, which can lead to critical risk severity
! Medium	This level vulnerabilities are should be fixed, as they carry an inherent risk of future exploits, and hacks which may or may not impact the smart contract execution.
! Low	This level vulnerabilities can be ignored. They are code style violations, and informational statements in the code. They may not affect the smart contract execution

Smart Contract Overview

Knick-knacks in the smart contract

Query	Result
marketingWallet	0xF4dCD0920A7224AaD0f564e902e859A879D8F40b
developmentWallet	0x95CFEa77697df18D4392fEe292DbD419faeeDdFb
airdropWallet	0x79C9F4115fB296D9aE33c0Ef768Cf3a432A8Ecf0
marketingFee	4
developmentFee	4
liquidityFee	2
_maxTxAmount	1000 * 10*6 * 10**9
_maxBuyAmount	1000 * 10**6 * 10**9
_maxSellAmount	1000 * 10**6 * 10**9
_maxWhaleAmount	1000 * 10**6 * 10**9
limitTheWhales	True
numTokensSellToAddToLiquidity	5 * 10**6 * 10**9 (0.5%)
numTokensMinimumSellToAddToLiquidity	5 * 10**6 * 10**9 (5%)
swapAndLiquifyEnabled	False
symbol	HDXXX
Name	HornyDogXXX
uniswapRouterV2	0x10ED43C718714eb63d5aA57B78B54704E256024E

Verifying token functions

Function	Description	Tested	Verdict
TotalSupply	provides information about the total token supply	**Yes**	**Passed**
BalanceOf	provides account balance of the owner's account	**Yes**	**Passed**
Transfer	executes transfers of a specified number of tokens to a specified address	**Yes**	**Passed**
TransferFrom	executes transfers of a specified number of tokens from a specified address	**Yes**	**Passed**
Approve	allow a spender to withdraw a set number of tokens from a specified account	**Yes**	**Passed**
Allowance	returns a set number of tokens from a spender to the owner	**Yes**	**Passed**

Verified

- Owner can mint tokens at token launch
- Owner can not burn/lock users' assets
- Owner can not pause the contract
- Max sell is 50% only if user hold more than 0.1% or more of total supply for 30 min

Note

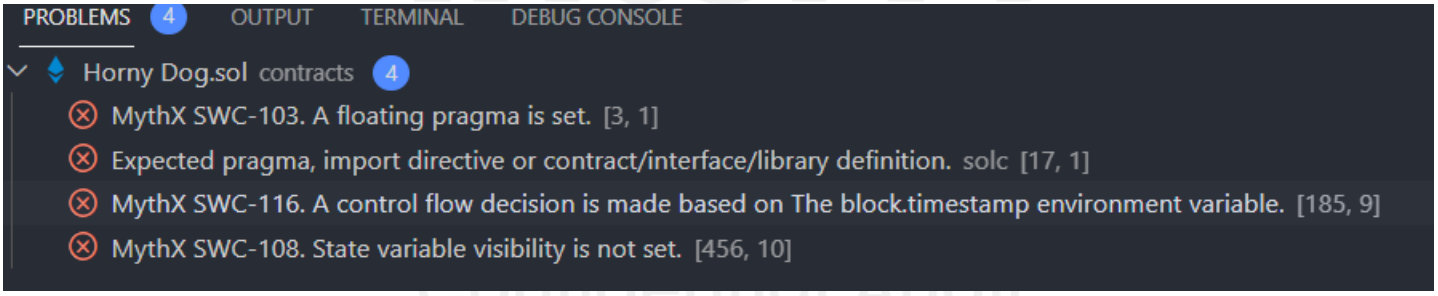
- At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before blockchain deployment.

Points To Note

The HornyDog.sol smart contract utilizes the “SafeMath” to prevent Integer Overflow.

```
UnitTest stub | dependencies | uml | draw.io
28  library SafeMath {
29
        ftrace | funcSig
30      function add(uint256 a↑, uint256 b↑) internal pure returns (uint256) {
31          uint256 c = a↑ + b↑;
32          require(c >= a↑, "SafeMath: addition overflow");
33
34          return c;
35      }
36
        ftrace | funcSig
37      function sub(uint256 a↑, uint256 b↑) internal pure returns (uint256) {
38          return sub(a↑, b↑, "SafeMath: subtraction overflow");
39      }
40
```

The HornyDog.sol smart contract has 3 low severity issues which may not create any functional vulnerability.



REPORT 613305F39DD4E80018A0A37F

Created	Sat Sep 04 2021 05:36:51 GMT+0000 (Coordinated Universal Time)
Number of analyses	1
User	61247cb78bfa120cb0f29add

REPORT SUMMARY

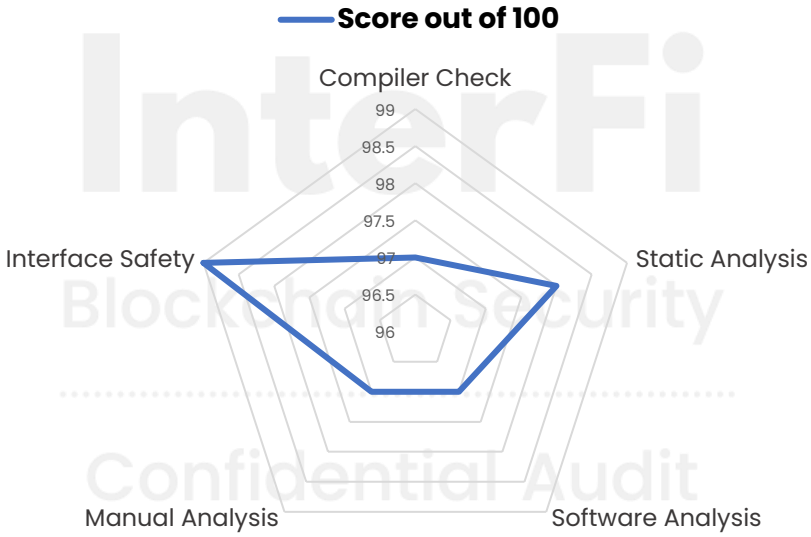
Analyses ID	Main source file	Detected vulnerabilities
e6499559-49b7-4f91-b3e5-7e0ef0e2184d	/contracts/horny dog.sol	3

Vulnerability	Status
Compiler errors	! Low
Re-entrancy. Race conditions and cross function race conditions (RE)	Passed
Possible delays in data delivery	Passed
Gas optimization	! Low
Integer Underflow and overflow	Passed
Oracle Calls	Passed
Call stack depth attack	Passed
Parity Multisig Bug	Passed
Tx ordering dependency (TO)	Passed
DOS with revert and block gas limit	Passed
Private user data leaks	Passed
Malicious event log	Passed
Safe open zeppelin contract implementation and usage	Passed
The impact of exchange rate on the logic	Passed
Functions that are not used (dead-code)	! Low
Typographical Errors	Passed
Signature Malleability	Passed
Floating Pragma	! Low
Scoping and declarations	Passed

Smart Contract Risk Assessment

SWC Errors	Issue	Severity
SWC-103	**A floating pragma is set**	! Low
	The current pragma Solidity directive is ""^0.8.0"". It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.	
SWC-108	**State variable visibility is not set.**	! Low
	It is best practice to set the visibility of state variables explicitly. The default visibility for "inSwapAndLiquify" is internal. Other possible visibility settings are public and private.	
SWC-116	**A control flow decision is made based on The block.timestamp environment variable.**	! Low
	The block.timestamp environment variable is used to determine a control flow decision. Note that the values of variables like coinbase, gaslimit, block number and timestamp are predictable and can be manipulated by a malicious miner. Also keep in mind that attackers know hashes of earlier blocks. Don't use any of those environment variables as sources of randomness and be aware that use of these variables introduces a certain level of trust into miners.	

Risk Severity	Status
! Critical	None critical severity issues identified
! High	None high severity issues identified
! Medium	None medium severity issues identified
! Low	3 Low severity issues identified **(! Low Impact)**
Passed	22 functions and instances verified and passed



Compiler Check	97
Static Analysis	98
Software Analysis	97
Manual Analysis	97
Interface Safety	99

Auditor's Verdict

InterFi team has performed a line-by-line manual analysis and automated review of the smart contract. The smart contract was analyzed mainly for common smart contract vulnerabilities, exploits, and manipulation hacks.

Horny Dog XXX's smart contract source code has LOW RISK SEVERITY.

Horny Dog XXX has PASSED the InterFi's ECHELON-1 standard smart contract audit.

Auditor's Footnote:

- At the time of the audit, the contract is not deployed on any blockchain. Note, the owner/developer can change/modify contract before deployment. Please proceed with caution.
- Project website is not checked due to out of scope. The website hasn't been reviewed for SSL and lighthouse report.
- Project team, and the project's social channels are not checked due to out of scope.

Important Disclaimer

InterFi Network provides contract auditing and project verification services for blockchain projects. The purpose of the audit is to analyse the on-chain smart contract source code, and to provide basic overview of the project. **This report should not be transmitted, disclosed, referred to, or relied upon by any person for any purposes without InterFi's prior written consent.**

InterFi provides the easy-to-understand assessment of the project, and the smart contract (otherwise known as the source code). The audit makes no statements or warranties on the security of the code. It also cannot be considered as an enough assessment regarding the utility and safety of the code, bug-free status, or any other statements of the contract. While we have used all the data at our disposal to provide the transparent analysis, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. **Be aware that smart contracts deployed on a blockchain aren't resistant from external vulnerability, or a hack. Be aware that active smart contract owner privileges constitute an elevated impact to smart contract's safety and security. Therefore, InterFi does not guarantee the explicit security of the audited smart contract.**

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report should not be considered as an endorsement or disapproval of any project or team. The information provided on this report does not constitute investment advice, financial advice, trading advice, or any other sort of advice and you should not treat any of the report's content as such. Do conduct your own due diligence and consult your financial advisor before making any investment decisions.

About InterFi Network

InterFi Network provides intelligent blockchain solutions. InterFi is developing an ecosystem that is seamless and responsive. Some of our services: Blockchain Security, Token Launchpad, NFT Marketplace, etc. **InterFi's mission is to interconnect multiple services like Blockchain Security, DeFi, Gaming, and Marketplace under one ecosystem that is seamless, multi-chain compatible, scalable, secure, fast, responsive, and easy-to-use.**

InterFi is built by a decentralized team of UI experts, contributors, engineers, and enthusiasts from all over the world. Our team currently consists of 6+ core team members, and 10+ casual contributors. **InterFi provides manual, static, and automatic smart contract analysis, to ensure that project is checked against known attacks and potential vulnerabilities.**

For more information, visit https://interfi.network

To book an audit, message https://t.me/interfiaudits