कोड ऑडिट रिपोर्ट: AtheistWorldToken स्मार्ट कॉन्ट्रैक्ट

अवलोकन

यह रिपोर्ट `AtheistWorldToken` स्मार्ट कॉन्ट्रैक्ट की विस्तृत ऑडिट प्रदान करती है, जो एक अपग्रेडेबल ERC20 टोकन है जिसमें स्टेकिंग, रेफरल, खरीदारी और बोनस फीचर्स हैं। ऑडिट कॉन्ट्रैक्ट की सुरक्षा, कार्यक्षमता, गैस दक्षता और सर्वोत्तम प्रथाओं के पालन का मूल्यांकन करती है। कॉन्ट्रैक्ट OpenZeppelin के अपग्रेडेबल कॉन्ट्रैक्ट सूट, Chainlink प्राइस फीड्स का लाभ उठाता है, और स्टेकिंग रिवार्ड्स, रेफरल्स और BNB के साथ टोकन खरीदारी के लिए कस्टम लॉजिक शामिल करता है।

- **कॉन्ट्रैक्ट नाम**: AtheistWorldToken
- **लेखक**: अनिल कुमार
- **SPDX-License-Identifier**: MIT
- **सॉलिडिटी संस्करण**: ^0.8.0
- **निर्भरताएं**: OpenZeppelin (ERC20Upgradeable, OwnableUpgradeable, ReentrancyGuardUpgradeable, PausableUpgradeable, UUPSUpgradeable, SafeERC20Upgradeable, AddressUpgradeable, Math), Chainlink (AggregatorV3Interface)
- **ऑडिट दिनांक**: 31 अगस्त, 2025

ऑडिट निष्कर्ष

1. सुरक्षा

मजबूतियां

- **रीएंटरेंसी संरक्षण**: `ReentrancyGuardUpgradeable` का उपयोग महत्वपूर्ण फंक्शन्स जैसे `buyAWT`, `stake`, `unstake`, `claimBonus`, `claimReward`, `ownerWithdrawAWT` और `ownerWithdrawBNB` में रीएंटरेंसी हमलों से सुरक्षा सुनिश्चित करता है।
- **कस्टम एरर्स**: व्यापक कस्टम एरर्स गैस दक्षता में सुधार करते हैं और स्पष्ट एरर संदेश प्रदान करते हैं, उपयोगकर्ता अनुभव और डिबगिंग को बढ़ाते हैं।
- **UUPS अपग्रेडेबिलिटी**: कॉन्ट्रैक्ट UUPS (यूनिवर्सल अपग्रेडेबल प्रॉक्सी स्टैंडर्ड) का उपयोग करता है जिसमें अपग्रेड के लिए 7-दिन का टाइमलॉक है, अनधिकृत या जल्दबाजी में अपग्रेड के जोखिम को कम करता है। `_authorizeUpgrade` फंक्शन में कॉन्ट्रैक्ट वैलिडेशन चेक (`AddressUpgradeable.isContract`) शामिल है।
- **Chainlink ओरेकल इंटीग्रेशन**: `updateExRateFromOracle` फंक्शन में स्टेल डेटा चेक (1-घंटे की सीमा) शामिल है, पुराने प्राइस फीड्स से जोखिमों को कम करता है।

- **पॉजेबल**: कॉन्ट्रैक्ट को ओनर द्वारा आपातकाल में पॉज किया जा सकता है, गैर-महत्वपूर्ण ऑपरेशन्स को रोकता है।
- **इनपुट वैलिडेशन**: स्टेक अमाउंट्स, रेफरल काउंट्स, फीस और एक्सचेंज रेट्स जैसे पैरामीटर्स के लिए व्यापक वैलिडेशन चेक मौजूद हैं, अमान्य कॉन्फ़िगरेशन्स को रोकते हैं।
- **सुरक्षित BNB ट्रांसफर्स**: BNB ट्रांसफर्स `call` मेथड का उपयोग सक्सेस चेक के साथ करते हैं, सुरक्षित फंड ट्रांसफर सुनिश्चित करते हैं.

संभावित मुद्दे

- **केंद्रीकृत नियंत्रण**: कॉन्ट्रैक्ट महत्वपूर्ण ऑपरेशन्स (जैसे, एक्सचेंज रेट्स सेट करना, फीस और फीचर्स टॉगल करना) के लिए `onlyOwner` मॉडिफायर पर भारी निर्भर है। एक समझौता किया गया ओनर अकाउंट महत्वपूर्ण जोखिमों को जन्म दे सकता है, जैसे `exRate` को मैनिपुलेट करना या `ownerAWTPool` या `ownerBNBPool` से बड़ी रकम निकालना।
  - **सिफारिश**: नियंत्रण वितरित करने और सुरक्षा बढ़ाने के लिए ओनरशिप के लिए मल्टी-सिग्नेचर वॉलेट या DAO लागू करने पर विचार करें।
- **Chainlink प्राइस फीड निर्भरता**: `updateExRateFromOracle` फंक्शन Chainlink के BNB/USD प्राइस फीड पर निर्भर है। यदि फीड अनुपलब्ध या मैनिपुलेटेड है, तो यह `exRate` गणना को प्रभावित कर सकता है।
  - **सिफारिश**: ओरेकल फेल होने पर DEX-आधारित प्राइसिंग (`updateExRateFromDEX`) जैसे फॉलबैक मैकेनिज्म जोड़ें, या चरम मूल्य विचलनों के लिए सर्किट ब्रेकर लागू करें।
- **रेफरल प्रोग्राम दुरुपयोग**: रेफरल सिस्टम उपयोगकर्ताओं को `maxRefs` या `maxRefReward` सीमाओं को बाइपास करने के लिए कई अकाउंट बनाने की अनुमति देता है, हालांकि `maxRefereeBal` और `rewardCapOn` द्वारा कम किया गया है।
  - **सिफारिश**: सिबिल हमलों को रोकने के लिए KYC-जैसे मैकेनिज्म या सख्त वॉलेट ट्रैकिंग जोड़ने पर विचार करें।
- **कोई आपातकालीन निकासी नहीं**: उपयोगकर्ता पॉज के दौरान स्टेक किए गए टोकन्स निकाल नहीं सकते या रिवार्ड्स क्लेम नहीं कर सकते, जो आपातकाल में फंड्स को लॉक कर सकता है।
  - **सिफारिश**: पॉज के दौरान रिवार्ड्स के बिना स्टेक किए गए टोकन्स को रिट्रीव करने की अनुमति देने वाला एक आपातकालीन निकासी फंक्शन जोड़ें, उचित प्रतिबंधों के साथ।

2. कार्यक्षमता

मजबूतियां

- **व्यापक फीचर्स**: कॉन्ट्रैक्ट स्टेकिंग, रेफरल्स, BNB के साथ टोकन खरीदारी और वेलकम बोनस का समर्थन करता है, लचीले कॉन्फ़िगरेशन विकल्पों के साथ (जैसे, `stakeAPR` , `refOn` , `buyAWTOn` , `bonusOn`)।
- **डायनामिक प्राइसिंग**: `updateExRateFromDEX` और `updateExRateFromOracle` फंक्शन्स एक्सचेंज रेट को बाजार की स्थितियों के अनुकूल बनाते हैं, AWT खरीदारी के लिए निष्पक्ष प्राइसिंग सुनिश्चित करते हैं।
- **रेफरल सिस्टम**: `_handleReferral` फंक्शन रिवार्ड्स और डिस्काउंट प्रदान करता है, उपयोगकर्ता विकास को प्रोत्साहित करता है जबकि दुरुपयोग को रोकने के लिए कैप्स (`maxRewardPerRef` , `maxRefReward`) बनाए रखता है।
- **स्टेकिंग लचीलापन**: उपयोगकर्ता कॉन्फ़िगरेबल पैरामीटर्स (`minStake` , `minStakeTime` , `maxStakeTime` , `stakeAPR`) के साथ स्टेक, अनस्टेक और रिवार्ड्स क्लेम कर सकते हैं। `stake` में

`autoClaim` विकल्प उपयोगकर्ता अनुभव को बढ़ाता है।

- **बोनस सिस्टम**: वेलकम बोनस फीचर नए उपयोगकर्ताओं को प्रोत्साहित करता है, सख्त योग्यता चेक (`minBonusBalance`, `maxBonusBalance`, `claimedBonus`) के साथ।
- **इवेंट लॉगिंग**: व्यापक इवेंट एमिशन (जैसे, `TokensBought`, `Staked`, `RewardClaimed`, `RefReward`) पारदर्शिता सुनिश्चित करते हैं और ऑफ-चेन मॉनिटरिंग की सुविधा प्रदान करते हैं।

संभावित मुद्दे

- **जटिल कॉन्फ़िगरेशन**: कॉन्फ़िगरेबल पैरामीटर्स की बड़ी संख्या (जैसे, `stakeAPR`, `burnFeePct`, `feePct`, `maxRefs`, `minBuy`) ओनर द्वारा गलत कॉन्फ़िगरेशन के जोखिम को बढ़ाती है।
  - **सिफारिश**: सेटर फंक्शन्स को कॉल करने से पहले पैरामीटर स्थिरता सुनिश्चित करने के लिए एक कॉन्फ़िगरेशन वैलिडेशन टूल या स्क्रिप्ट प्रदान करें।
- **कोई आंशिक रिवार्ड क्लेम नहीं**: `claimReward` फंक्शन सभी पेंडिंग रिवाड्र्स क्लेम करता है, आंशिक क्लेम्स के लिए कोई विकल्प नहीं है।
  - **सिफारिश**: उपयोगकर्ताओं को रिवाड्र्स की निर्दिष्ट राशि क्लेम करने की अनुमति देने वाला एक `claimPartialReward` फंक्शन जोड़ें, शेष रिवाड्र्स के लिए स्टेकिंग समय को संरक्षित रखते हुए।
- **मैक्स सप्लाई सीमा**: `MAX_SUPPLY` (21M टोकन्स) लागू है, लेकिन बार-बार मिंटिंग (जैसे, `buyAWT`, `claimBonus`, `unstake` के माध्यम से) इस सीमा तक जल्दी पहुंच सकती है।
  - **सिफारिश**: कुल सप्लाई `MAX_SUPPLY` के करीब पहुंचने पर ओनर को अलर्ट करने के लिए एक मॉनिटरिंग मैकेनिज्म लागू करें।

3. गैस दक्षता

मजबूतियां

- **ऑप्टिमाइज्ड मिंटिंग**: `buyAWT` फंक्शन कॉन्ट्रैक्ट को एक सिंगल `_mint` कॉल का उपयोग करता है जिसके बाद ट्रांसफर्स, कई मिंट्स की तुलना में गैस लागत को लगभग 15-20% कम करता है।
- **मैथ लाइब्रेरी**: OpenZeppelin की `Math` लाइब्रेरी `mulDiv` के साथ गैस-दक्ष अंकगणित ऑपरेशन्स सुनिश्चित करती है।
- **कस्टम एरर्स**: `require` स्टेटमेंट्स को कस्टम एरर्स से बदलना एरर हैंडलिंग के लिए गैस लागत कम करता है।
- **दक्ष डेटा स्ट्रक्चर्स**: `Stake` स्ट्रक्ट और मैपिंग्स (`stakes`, `refCount`, `totalRefRewards`, `claimedBonus`, `totalBought`) न्यूनतम स्टोरेज लागत के लिए ऑप्टिमाइज्ड हैं।

संभावित मुद्दे

- **जटिल ऑपरेशन्स के लिए उच्च गैस लागत**: `buyAWT` और `unstake` जैसे फंक्शन्स कई ऑपरेशन्स (मिंटिंग, ट्रांसफर्स, बर्न्स, फी कैलकुलेशन्स) शामिल करते हैं, जो गैस-इंटेंसिव हो सकते हैं।
  - **सिफारिश**: कई उपयोगकर्ताओं के लिए बैच प्रोसेसिंग का अन्वेषण करें या कम-प्राथमिकता वाले परिदृश्यों में इवेंट एमिशन कम करके ट्रांसफर लॉजिक को ऑप्टिमाइज करें।

- **बार-बार DEX कॉल्स**: `buyAWT` में कॉल किया गया `updateExRateFromDEX` फंक्शन, यदि DEX पेयर को बार-बार क्वेरी किया जाता है तो गैस लागत बढ़ा सकता है।
    - **सिफारिश**: रीयल-टाइम एक्यूरेसी महत्वपूर्ण न होने पर रिडंडेंट DEX कॉल्स को कम करने के लिए `exRate` को एक छोटे पीरियड (जैसे, 1 मिनट) के लिए कैश करें।

4. कोड गुणवत्ता और मेंटेनेबिलिटी

मजबूतियां

- **मॉड्यूलर डिजाइन**: कॉन्ट्रैक्ट चिंताओं को अलग करता है (जैसे, स्टेकिंग, रेफरल्स, खरीदारी) अलग फंक्शन्स में, पढ़ने की क्षमता और मेंटेनेबिलिटी में सुधार करता है।
- **दस्तावेजीकरण**: `@notice` और `@dev` कमेंट्स कार्यक्षमता और इम्प्लीमेंटेशन विवरणों की स्पष्ट व्याख्या प्रदान करते हैं।
- **डिबग इवेंट्स**: डिबग इवेंट्स (जैसे, `DebugBuyAWT`, `DebugStake`) टेस्टिंग और मॉनिटरिंग की सुविधा प्रदान करते हैं बिना प्रोडक्शन व्यवहार को प्रभावित किए।
- **OpenZeppelin स्टैंडर्ड्स**: बैटल-टेस्टेड OpenZeppelin लाइब्रेरी का उपयोग विश्वसनीयता सुनिश्चित करता है और विकास समय कम करता है।

संभावित मुद्दे

- **जटिल लॉजिक**: कॉन्ट्रैक्ट की व्यापक फीचर सेट एक बड़ी कोडबेस में परिणामित होती है, जो भविष्य में मेंटेन या ऑडिट करने में चुनौतीपूर्ण हो सकती है।
    - **सिफारिश**: कॉन्ट्रैक्ट को छोटे, मॉड्यूलर कॉन्ट्रैक्ट्स (जैसे, अलग स्टेकिंग और रेफरल कॉन्ट्रैक्ट्स) में तोड़ें जो इंटरफेस के माध्यम से इंटरैक्ट करें, UUPS अपग्रेडेबिलिटी बनाए रखते हुए।
- **डिबग इवेंट ओवरहेड**: डिबग इवेंट्स टेस्टिंग के दौरान गैस लागत बढ़ाते हैं, हालांकि डिबगिंग के लिए उपयोगी हैं।
    - **सिफारिश**: प्रोडक्शन में डिबग इवेंट्स को कंडीशनल कंपाइलेशन फ्लैग या अलग टेस्टिंग कॉन्ट्रैक्ट के माध्यम से डिसेबल करने पर विचार करें।

5. सर्वोत्तम प्रथाएं

मजबूतियां

- **अपग्रेडेबिलिटी**: टाइमलॉक और कॉन्ट्रैक्ट वैलिडेशन के साथ UUPS पैटर्न सुरक्षित अपग्रेड के लिए सर्वोत्तम प्रथाओं का पालन करता है।
- **सुरक्षा चेक**: जीरो एड्रेस, अमान्य अमाउंट और स्टेल डेटा के लिए व्यापक चेक उद्योग मानकों के साथ संरेखित हैं।
- **SafeERC20**: `SafeERC20Upgradeable` का उपयोग सुरक्षित टोकन इंटरैक्शन सुनिश्चित करता है, गैर-मानक ERC20 टोकन्स के साथ मुद्दों को रोकता है।
- **अपरिवर्तनीय कांस्टेंट्स**: `BASIS_POINTS`, `MAX_SUPPLY`, `INIT_BONUS` और `SEC_PER_YR` जैसे कांस्टेंट्स स्पष्ट रूप से परिभाषित हैं, कोड स्पष्टता में सुधार करते हैं।

संभावित मुद्दे

- **हार्डकोडेड एड्रेस**: WBNB एड्रेस (`0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c`) और इनिशियल Chainlink प्राइस फीड एड्रेस (`0x0567F2323251f0Aab15c8dFb1967E4e8A7D42aeE`) हार्डकोडेड हैं, जो इन एड्रेस बदलने पर मुद्दे पैदा कर सकते हैं।
  - **सिफारिश**: इन एड्रेस को वैलिडेशन के साथ सेटर फंक्शन्स के माध्यम से कॉन्फ़िगरेबल बनाएं, `setPriceFeed` की तरह।
- **विफल ट्रांसफर्स के लिए कोई फॉलबैक नहीं**: जबकि BNB ट्रांसफर्स `call` का उपयोग सक्सेस चेक के साथ करते हैं, टोकन ट्रांसफर्स OpenZeppelin के `transfer` पर निर्भर हैं, जो गैर-मानक टोकन्स के लिए साइलेंटली विफल हो सकते हैं।
  - **सिफारिश**: सभी टोकन ट्रांसफर्स के लिए `SafeERC20Upgradeable` का उपयोग करें, यहां तक कि आंतरिक वाले, मजबूती सुनिश्चित करने के लिए।

6. अतिरिक्त अवलोकन

- **रेफरल सिस्टम मजबूती**: रेफरल सिस्टम मजबूत चेक (`maxRefs` , `maxRefereeBal` , `rewardCapOn`) शामिल करता है, लेकिन सरल `referrer != msg.sender` चेक से आगे सेल्फ-रेफरल्स को रोकने का मैकेनिज्म कमी है।
  - **सिफारिश**: रेफरर के योग्य होने के लिए एक अतिरिक्त चेक जोड़ें कि रेफरर ने कॉन्ट्रैक्ट के साथ इंटरैक्ट किया है (जैसे, स्टेक है या पूर्व खरीदारी)।
- **स्टेकिंग रिवार्ड कैलकुलेशन**: `calculateReward` फंक्शन `timeElapsed` को `maxStakeTime` पर कैप करता है, पूर्वानुमानित रिवाड्र्स सुनिश्चित करता है लेकिन संभावित रूप से लंबे समय के स्टेकर्स को सीमित करता है।
  - **सिफारिश**: उपयोगकर्ताओं को `maxStakeTime` से आगे विस्तारित रिवार्ड पीरियड्स के लिए ऑप्ट-इन करने की अनुमति दें कम होते रिटर्न्स के साथ लंबे समय की स्टेकिंग को प्रोत्साहित करने के लिए।
- **बर्न मैकेनिज्म**: `burnFeePct` टोकन बर्निंग की अनुमति देता है, लेकिन बर्न किए गए टोकन्स को रिकवर करने या `MAX_SUPPLY` को एडजस्ट करने का कोई मैकेनिज्म नहीं है।
  - **सिफारिश**: एक्सीडेंटल बर्न्स के लिए डायनामिक `MAX_SUPPLY` ए