Auditoria de Contrato Inteligente - HuskySeal ($Husky)

AUDITORIA VERIFICADA E TESTADA VIA API BSCSCAN
Realizada por: VKINHA IA
Data da Auditoria: 28 de Abril de 2025

Visão Geral do Contrato

O contrato **HuskySeal** é um token BEP-20 implementado na Binance Smart Chain (BSC), com funcionalidades adicionais como taxas dinâmicas (buy/sell taxes), suporte a liquidez, queima de tokens (burn), e carteiras para marketing, buyback e desenvolvimento. O contrato foi submetido para verificação no BscScan em 19 de abril de 2025. A auditoria avalia a segurança do código, funções específicas, carteiras isentas de taxas, e verifica a existência de códigos ocultos.

Análise Detalhada das Funções do Contrato

1. Construtor

- **Descrição**: Inicializa o token com nome "Husky Seal" e símbolo "$Husky", mintando 1 bilhão de tokens para o `msg.sender`. Configura o roteador PancakeSwap (endereço `0x10ED43C718714eb63d5aA57B78B54704E256024E`) e cria um par de liquidez com WETH.
- **Análise**:
  - O endereço do roteador é fixo e corresponde à PancakeSwap, o que é esperado para tokens BEP-20 na BSC.
  - **Carteiras Isentas de Taxas**: O construtor define várias carteiras como isentas de taxas (`exemptFee`):
    - `msg.sender` (deployer do contrato)
    - Endereço do contrato (`address(this)`)
    - `Marketingwallet` (`0x2517f3BAa7357f4dF1A81a86ae21aDdBAB6b7973`)
    - `Buyback` (`0x8646CacbD1D6624Ab85B16e8424AC2e007762dfD`)
    - `deadWallet` (`0x000000000000000000000000000000000000dEaD`)
    - `DevWallet` (`0x54D5daF7ae5E6Af1C39306781E269Aa21e03a39c`)
    - Outros endereços fixos: `0x87D02f93BC0A97c4586ED0df351F7ea1F18a4335`, `0xc9Ef8217cE5ba596aFEe16e4f7130b1c4Cd66A87`, `0x2FEcF6FE979463EAA45619B6eE9C1c460a`
  - **Alerta**: A presença de múltiplas carteiras isentas de taxas, incluindo endereços fixos não identificados, pode ser um ponto de preocupação. Isso permite que essas carteiras realizem transferências sem pagar taxas, o que pode ser explorado para manipulação de mercado ou dumping de tokens.
- **Segurança**: O mint inicial é padrão, mas a isenção de taxas para várias carteiras requer atenção.

2. **Função `setSwapTokens`**

- **Descrição**: Permite ao dono do contrato ajustar o limite de tokens a serem trocados (`swapTokens`) durante a liquidez.
- **Análise**:
  - Restrições razoáveis: o valor deve estar entre 0,1% e 100% do fornecimento total.
  - **Modificador**: `onlyOwner`, garantindo que apenas o dono pode alterar o valor.
  - **Segurança**: Função segura, sem vulnerabilidades aparentes.

3. **Função `enableTrading`**

- **Descrição**: Ativa a negociação (`tradingEnabled`) e a provisão de liquidez (`providingLiquidity`).
- **Análise**:
  - **Modificador**: `onlyOwner`.
  - Inclui uma verificação para evitar ativação redundante.
  - **Segurança**: Função segura, mas a ativação do trading deve ser monitorada, pois permite transferências com taxas.

4. **Função `setProvidingLiquidity`**

- **Descrição**: Ativa/desativa a provisão de liquidez.
- **Análise**:
  - **Modificador**: `onlyOwner`.
  - Emite um evento `ProvidingLiquidityUpdated`.
  - **Segurança**: Função segura, sem riscos aparentes.

5. **Função `excludeFromFee`**

- **Descrição**: Permite ao dono do contrato adicionar endereços à lista de isenção de taxas.
- **Análise**:
  - **Modificador**: `onlyOwner`.
  - **Alerta**: Essa função permite ao dono adicionar carteiras isentas de taxas dinamicamente. Isso pode ser usado para favorecer certos endereços, potencialmente manipulando o mercado.
  - **Segurança**: A função é segura em termos de implementação, mas o uso indiscriminado pode ser prejudicial aos investidores.

6. **Função `forceLiquify`**

- **Descrição**: Força a liquidez manualmente, podendo ser chamada pelo dono ou pelo `liquifyAdmin`.
- **Análise**:
  - **Acesso**: Restrito ao dono ou ao `liquifyAdmin` (`0x18d018e8E3265E471eaf7601E4484Ef89341D894`).

  - **Alerta**: O `liquifyAdmin` é um endereço fixo e imutável, com privilégios para forçar liquidez. Isso cria um ponto de centralização, pois esse endereço pode influenciar o contrato independentemente do dono.
  - **Segurança**: A função é segura, mas o privilégio do `liquifyAdmin` deve ser monitorado.

7. Funções `setBuyTaxes` e `setSellTaxes`

- **Descrição**: Permitem ao dono ajustar as taxas de compra e venda (Marketing, Buyback, Burn, Liquidity, Dev).
- **Análise**:
  - **Modificador**: `onlyOwner`.
  - Limite máximo de 30% para a soma das taxas, o que é razoável para evitar abusos.
  - **Segurança**: Funções seguras, com restrições apropriadas.

8. Funções `recoverBEP20FromContract` e `recoverBNBfromContract`

- **Descrição**: Permitem ao dono recuperar tokens BEP-20 ou BNB presos no contrato.
- **Análise**:
  - **Modificador**: `onlyOwner`.
  - Os fundos recuperados são enviados para o `Marketingwallet`.
  - **Alerta**: Essa funcionalidade dá ao dono controle total sobre quaisquer ativos presos no contrato, o que pode ser explorado para retirada de fundos de forma não transparente.
  - **Segurança**: Implementação segura, mas o uso deve ser monitorado.

9. Função `_transfer` (Sobrescrita)

- **Descrição**: Gerencia transferências de tokens, aplicando taxas dinâmicas para compras e vendas.
- **Análise**:
  - Verifica se o trading está ativado, exceto para endereços isentos.
  - Calcula taxas com base em `buytaxes` (compra) ou `sellTaxes` (venda).
  - **Alerta**: A lógica de taxas é aplicada apenas se o remetente ou destinatário não estiverem na lista `exemptFee`. Como várias carteiras estão isentas, elas podem transferir grandes quantidades sem pagar taxas.
  - **Segurança**: A função é bem implementada, mas a isenção de taxas para múltiplas carteiras é um risco.

10. Função `Liquify`

- **Descrição**: Converte tokens em ETH e adiciona liquidez ao par PancakeSwap.
- **Análise**:

- **Modificador**: `lockTheSwap` (evita reentrância).
- Divide os tokens entre liquidez, marketing, buyback e dev, enviando ETH para as respectivas carteiras.
- **Segurança**: A função é protegida contra reentrância e bem estruturada.

11. Funções `swapTokensForETH` e `addLiquidity`

- **Descrição**: Funções auxiliares para trocar tokens por ETH e adicionar liquidez.
- **Análise**:
  - Interagem com o roteador PancakeSwap de forma padrão.
  - **Segurança**: Funções seguras, sem vulnerabilidades aparentes.

Análise de Carteiras Isentas de Taxas

O contrato define várias carteiras como isentas de taxas no construtor e permite que o dono adicione mais via `excludeFromFee`. Os endereços isentos são: - Deployer (`msg.sender`) - Endereço do contrato (`address(this)`) - `Marketingwallet` (`0x2517f3BAa7357f4dF1A81a86ae21aDdBAB6b7973`) - `Buyback` (`0x8646CacbD1D6624Ab85B16e8424AC2e007762dfD`) - `deadWallet` (`0x000000000000000000000000000000000000dEaD`) - `DevWallet` (`0x54D5daF7ae5E6Af1C39306781E269Aa21e0`) - Endereços fixos adicionais: `0x87D02f93BC0A97c4586ED0df351F7ea1F18a4335`, `0xc9Ef8217cE5ba596aFEe16e4f7130b1c4Cd66A87`, `0x2FEcF6FE979463EAA45619B6eE9C1c460a53e4E9`

Risco: A isenção de taxas para múltiplas carteiras, especialmente endereços fixos não identificados, pode permitir manipulação de mercado ou dumping de tokens. Além disso, o endereço `liquifyAdmin` (`0x18d018e8E3265E471eaf7601E4484Ef89341D894`) tem privilégios para forçar liquidez, criando um ponto de centralização.

Verificação de Códigos Ocultos

- **Análise**: O contrato não apresenta códigos ocultos ou funcionalidades não documentadas. Todas as funções são transparentes, e o código segue o padrão BEP-20 com extensões para taxas e liquidez.
- **Conclusão**: Nenhum código oculto encontrado.

Análise dos Holders (Top 1,000)

Os principais detentores do token, conforme fornecidos, são: 1. `0x000...dEaD` (25,44%) - Endereço de queima, esperado. 2. `0xa31767BF...0B2Eb5A55` (25,26%) - Endereço não identificado. 3. `0x18d018e8...89341D894` (24,09%) - Coincide com o `liquifyAdmin`, indicando que este endereço detém uma quantidade significativa de tokens. 4. PancakeSwap V2 (5,11%) - Par de liquidez, esperado. 5. Outros endereços com participações menores.

Observação: O `liquifyAdmin` ser um dos maiores detentores é um ponto de preocupação, pois combina privilégios administrativos com uma grande quantidade de tokens, aumentando o risco de centralização.

Monitoramento de Transações

O contrato está implantado no endereço `0x66b6b2ed21a0bfb3f84b120401074abfc4f0c08d`. Analisando as transações via BscScan: - As transações são consistentes com um token BEP-20, incluindo transferências, adição de liquidez e queima de tokens. - **Alerta**: Algumas transações mostram grandes transferências de carteiras isentas (como o `liquifyAdmin`), o que confirma o risco de manipulação devido à isenção de taxas. - Não há atividades suspeitas que indiquem exploits ou comportamento malicioso direto no contrato.

Conclusão e Nota de Segurança

- **Pontos Positivos**:
  - O contrato segue boas práticas para tokens BEP-20, com proteção contra reentrância (`lockTheSwap`).
  - Limites razoáveis para taxas (máximo 30%).
  - Nenhum código oculto encontrado.
- **Pontos de Preocupação**:
  - Múltiplas carteiras isentas de taxas, incluindo endereços fixos não identificados.
  - O `liquifyAdmin` tem privilégios significativos e detém uma grande quantidade de tokens, criando centralização.
  - Funções como `recoverBEP20FromContract` e `recoverBNBfromContract` dão ao dono controle total sobre ativos presos no contrato.
- **Nota de Segurança**: 85/100
  A pontuação reflete a boa implementação técnica, mas é reduzida devido à centralização causada pelas carteiras isentas e pelo `liquifyAdmin`.

Nota Final: Esta auditoria avalia apenas a segurança do código e não valida a credibilidade da equipe por trás do projeto.