Smart Contract Audit Report: IMBTC

This report summarizes the findings of a comprehensive audit conducted on the IMBTC smart contract us

1. Unchecked Input in `addMinter` and `removeMinter` Functions
- **Severity**: High
- **Description**: The `addMinter` and `removeMinter` functions in the `MinterRole` contract do not perfor
- **Impact**: This vulnerability allows an attacker to:
  - **DoS attack**: Submit a large number of fake addresses to the `addMinter` function, increasing gas
  - **Minter manipulation**: Remove all existing minters through `removeMinter`, potentially disrupting th
- **Mitigation**: Implement input validation and sanitization checks for the `address` parameter in both fu

2. Unchecked Pausing in the `Pausable` Contract
- **Severity**: Medium
- **Description**: While the `pause` function requires the sender to be the owner, it does not check the se
- **Impact**: An attacker could use a transaction with a high gas limit and low gas price to perform a DoS
- **Mitigation**: Add a check for the sender's balance before allowing the owner to pause or unpause the

3. Unchecked Transferability Toggling in the `SwitchTransferable` Contract
- **Severity**: High
- **Description**: The `enableTransfer` and `disableTransfer` functions do not check the input `address` b
- **Impact**: This vulnerability allows an attacker to:
  - **Manipulate transferability**: Submit fake addresses to `enableTransfer`, unintentionally changing th
  - **Disable all transfers**: Use `disableTransfer` to disable all token transfers for every address, preven
- **Mitigation**: Implement input validation and sanitization checks for the `address` parameter in both fu

4. Potential Revenue Address Manipulation
- **Severity**: Medium
- **Description**: The `setRevenueAddress` function allows the contract owner to change the revenue ad
- **Impact**: If the owner's account is compromised, an attacker could change the revenue address to the
- **Mitigation**:
  - Implement a multisig wallet or a DAO for the contract owner to mitigate single-point failure.
  - Consider adding a mechanism to prevent sudden changes to the revenue address, requiring a delay

5. Potential Minting to Invalid Addresses
- **Severity**: Medium
- **Description**: The `mint` function does not check if the recipient is a valid address.
- **Impact**: If the recipient is a zero address or a contract without the necessary interface, minted tokens
- **Mitigation**: Add a check to ensure that the recipient address is valid before minting tokens.

6. ERC1820 Registry Vulnerability
- **Severity**: Medium
- **Description**: The contract relies on the ERC1820 registry for interface implementations.
- **Impact**: If the ERC1820 registry is compromised, it could lead to loss of funds or unexpected contrac
- **Mitigation**: While difficult to directly mitigate, it is crucial to stay informed of any potential vulnerabiliti

Recommendations

- **Address the identified vulnerabilities**: Implement the suggested mitigations for the vulnerabilities outl
- **Conduct regular security audits**: Engage a reputable security auditing firm to perform regular audits t
- **Enhance contract owner security**: Employ a multisig wallet or a DAO for the contract owner to mitiga

- **Implement best practices**: Ensure adherence to established Solidity coding best practices to prevent

This report highlights potential vulnerabilities in the IMBTC smart contract. The identified risks should be a