

Registration Smart Contract

```
Parameter Registration.registerEntity(address,Registration.EntityType)._entityAddress (Registration.sol#31) is not in mixedCase
Parameter Registration.registerEntity(address,Registration.EntityType)._entityType (Registration.sol#31) is not in mixedCase
Parameter Registration.removeEntity(address)._entityAddress (Registration.sol#41) is not in mixedCase
Parameter Registration.getEntity(address)._entityAddress (Registration.sol#47) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Registration.sol analyzed (1 contracts with 81 detectors), 4 result(s) found
```

Child Labour Data Management Smart Contract

```
ChildLabourDataManagement.verifyViolationReport(uint256,bool) (ChildLabourDataManagement.sol#305-312) uses timestamp for comparisons
  Dangerous comparisons:
    - require(bool,string)(! violationReports[_violationReportId].isInspected,This report has already been inspected) (ChildLabourDataManagement.sol#307)
ChildLabourDataManagement.getChildData(uint256,uint256) (ChildLabourDataManagement.sol#332-343) uses timestamp for comparisons
  Dangerous comparisons:
    - require(bool,string)(report.isVerified,Cannot retrieve child data because the violation report Id has not been verified) (ChildLabourDataManagement.sol#339)
ChildLabourDataManagement.completeRemediation(uint256,uint256) (ChildLabourDataManagement.sol#389-399) uses timestamp for comparisons
  Dangerous comparisons:
    - require(bool,string)(block.timestamp >= remediationdata.remediationCompletion,The time window for remediation is still open) (ChildLabourDataManagement.sol#393)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp
ChildLabourDataManagement.sol analyzed (4 contracts with 81 detectors), 3 result(s) found
```

Verifier ZKP Smart Contract

```
Pairing.addition(Pairing.G1Point,Pairing.G1Point) (Verifier.sol#40-53) uses assembly
  - INLINE ASM (Verifier.sol#47-51)
Pairing.scalar_mul(Pairing.G1Point,uint256) (Verifier.sol#58-70) uses assembly
  - INLINE ASM (Verifier.sol#64-68)
Pairing.pairing(Pairing.G1Point[],Pairing.G2Point[]) (Verifier.sol#75-98) uses assembly
  - INLINE ASM (Verifier.sol#91-95)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
Verifier.sol analyzed (2 contracts with 81 detectors), 3 result(s) found
```