



**SOLID**Proof  
*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC**

MADE IN GERMANY

# Audit

## Security Assessment

18. августа 2021 г.

**For**



Отказ от ответственности	3
Описание	5
Участие в проекте	5
Логотип	5
Ссылка на контракт	5
Методология	7
Использованный код из других фреймворков / смарт-контрактов (прямой импорт)	8
Протестированные файлы контрактов	9
Исходные строки	10
Уровень риска	10
Возможности	11
Объем работ	13
График наследования	13
Проверить претензии	14
CallGraph	19
Источники в объеме	20
Критические проблемы	21 год
Высокие проблемы	21 год
Средние проблемы	21 год
Низкие проблемы	21 год
Информационные вопросы	21 год
Комментарии аудита	22
SWC атаки	23

Отчеты SolidProof.io не являются и не должны рассматриваться как «одобрение» или «неодобрение» какого-либо конкретного проекта или команды. Эти отчеты не являются и не должны рассматриваться как указание на экономику или ценность любого «продукта» или «актива», созданного какой-либо командой. SolidProof.io не покрывает тестирование или аудит интеграции с внешним контрактом или услугами (такими как Uniswap, PancakeSwap и т. Д. ...)

**SolidProof.io Audits не дает никаких гарантий относительно абсолютного отсутствия ошибок в анализируемой технологии, а также не дает никаких указаний на владельцев технологий. SolidProof Audits ни в коем случае не следует использовать для принятия решений об инвестициях или участии в каком-либо конкретном проекте. Эти отчеты никоим образом не содержат рекомендаций по инвестициям и не должны использоваться в качестве рекомендаций по инвестициям.**

Отчеты SolidProof.io представляют собой обширный процесс аудита, призванный помочь нашим клиентам повысить качество своего кода при одновременном снижении высокого уровня риска, связанного с криптографическими токенами и технологией блокчейн. Технология блокчейн и криптографические активы представляют собой постоянный высокий уровень риска. Позиция SolidProof заключается в том, что каждая компания и каждое отдельное лицо несут ответственность за свою должную осмотрительность и постоянную безопасность. SolidProof никоим образом не претендует на какие-либо гарантии безопасности или функциональности технологии, которую мы согласны анализировать.

Версия	Дата	Описание
1.0	18. августа 2021 г.	<ul style="list-style-type: none"><li>• Верстка проекта</li><li>• Автоматическое / ручное тестирование безопасности</li><li>• Резюме</li></ul>

## **Сеть**

Умная сеть Binance (BEP20)

## **Веб-сайт**

<https://herobattle.app/>

## **Телеграмма**

<https://t.me/HeroBattleBSC> <https://t.me/HeroBattleBSCNews>

## **Твиттер**

<https://twitter.com/HeroBattleBSC>

## **Facebook**

<https://www.facebook.com/HeroBattle-468380853567164/>

## **Instagram**

[https://www.instagram.com/herobattle\\_bsc/](https://www.instagram.com/herobattle_bsc/)

## **YouTube**

[https://www.youtube.com/channel/UCnR7\\_CAGwsvc3Ewx1LvnbEA](https://www.youtube.com/channel/UCnR7_CAGwsvc3Ewx1LvnbEA)

## **Reddit**

<https://www.reddit.com/user/HeroBattle>

## **Середина**

<https://medium.com/@herobattle>

## Описание

HEROBATTLE станет самым привлекательным игровым проектом на BSC с профессиональными разработчиками игр в сочетании с командой разработчиков игровой платформы на блокчейне. Игроки испытают на себе ролевую модель «ИГРАЙ, чтобы заработать». Разница между HEROBATTLE и традиционной игрой заключается в экономической схеме блокчейна, используемой для вознаграждения наших игроков за их вклад.

## Участие в проекте

15 августа 2021 г. **Боевая команда героев** привлекли Solidproof.io для аудита созданных ими смарт-контрактов. Задание носило технический характер и было сосредоточено на выявлении недостатков безопасности при разработке и выполнении контрактов. **Боевая команда героев** предоставили Solidproof.io доступ к своему репозиторию кода и техническому документу.

### Логотип



### Ссылка на контракт

[#code](https://bscscan.com/address/0xD5c213E95249E3D570E2aCA31a72fB4474D4043b)

# Уязвимость и уровень риска

Риск представляет собой вероятность того, что определенный источник-угроза воспользуется уязвимостью, и влияние этого события на организацию или систему. Уровень риска рассчитывается на основе CVSS версии 3.0.

Уровень	Ценить	Уязвимость	Риск (необходимое действие)
Критический	9 - 10	Уязвимость, которая может нарушить функционирование контракта в ряде сценариев или создает риск того, что контракт может быть разорван.	Немедленное действие снизить уровень риска.
Высокий	7 - 8,9	Уязвимость, которая влияет на желаемый результат при использовании договор или предоставляет возможность использовать контракт непреднамеренным образом.	Реализация корректирующие действия как как можно скорее.
Середина	4 - 6,9	Уязвимость, которая может повлиять на желаемый результат выполнение контракт по конкретному сценарию.	Реализация корректирующие действия в определенный период.
Низкий	2-3,9	Уязвимость, которая не имеет значительное влияние на возможные сценарии для использование контракт и наверное субъективно.	Реализация определенные корректирующие действия или принятие риск.
Информационная	0 - 1,9	Уязвимость, которая имеет информационный символ, но не влияет на код.	Наблюдение, что не определяет уровень риска

# Стратегия и применяемые методы аудита

На протяжении всего процесса проверки было уделено внимание оценке репозитория на предмет проблем, связанных с безопасностью, качества кода и соответствия спецификациям и лучшим практикам. Для этого мы проводим построчную проверку нашей командой опытных пентестеров и разработчиков смарт-контрактов, документируя все обнаруженные проблемы.

## Методология

Процесс аудита состоит из рутинной серии шагов:

1. Проверка кода, которая включает в себя следующее:
  - я) Ознакомьтесь со спецификациями, источниками и инструкциями, предоставленными SolidProof, чтобы убедиться, что мы понимаем размер, объем и функциональность смарт-контракта.
  - II) Ручная проверка кода, которая представляет собой процесс чтения исходного кода построчно в попытке определить потенциальные уязвимости.
  - iii) Сравнение со спецификацией, то есть процесс проверки того, выполняет ли код то, что описывают спецификации, источники и инструкции, предоставленные SolidProof.
2. Тестирование и автоматический анализ, который включает в себя следующее:
  - я) Анализ покрытия тестами, который представляет собой процесс определения того, действительно ли тестовые примеры покрывают код и сколько кода выполняется, когда мы запускаем эти тестовые примеры.
  - II) Символьное выполнение, которое анализирует программу, чтобы определить, какие входные данные вызывают выполнение каждой части программы.
3. Обзор передового опыта, который представляет собой обзор смарт-контрактов с целью повышения эффективности, действенности, уточнения, ремонтпригодности, безопасности и контроля на основе установленных отраслевых и академических практик, рекомендаций и исследований.
4. Конкретные, детализированные и действенные рекомендации, которые помогут вам предпринять шаги для защиты ваших смарт-контрактов.

## Использованный код из других фреймворков / смарт-контрактов (прямой импорт)

Импортированные пакеты:

- OpenZeppelin
  - Адрес
  - Собственный
  - SafeMath
  - SafeMathUint
  - SafeMathInt
  - IterableMapping
  - ERC20
  - IERC20
- Uniswap
  - UniswapV2Factory
  - UniswapV2Pair
  - UniswapV2Router01
  - UniswapV2Router02



## Протестированные файлы контрактов

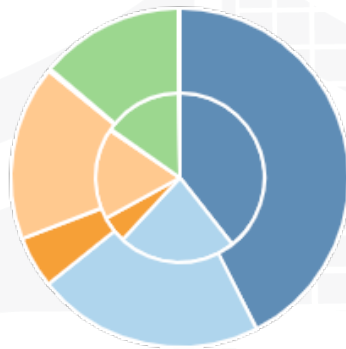
Этот аудит охватил следующие файлы, перечисленные ниже, с хешем SHA-1.

*Файл с другим хешем был изменен намеренно или иным образом после проверки безопасности. Другой хеш может (но не обязательно) указывать на изменившееся состояние или потенциальную уязвимость, которые выходят за рамки этого обзора.*

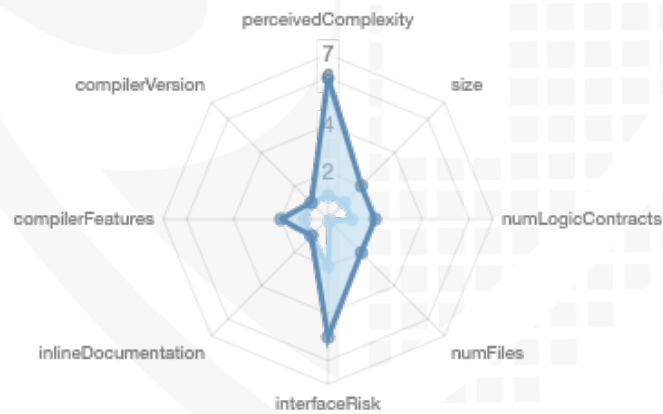
File Name	SHA-1 Hash
contracts/IERC20Metadata.sol	c98275f7bd2c6cb0eaeef0dab83be92848ee3c09b
contracts/DividendPayingToken.sol	82c11e2c0a4fc6ef445be6c6e83264f80b41088b
contracts/IUniswapV2Pair.sol	a345c94b2ca004ec5d68ea2fa95dc90d7c0b7e7a
contracts/SafeMathUint.sol	a4719406da9f9075421f53e2b82a916ff9ecaad8
contracts/Context.sol	d5f8c0eeac877f69ba9ada0f9925abecab9acdae
contracts/IUniswapV2Factory.sol	6dfd768eef85f94ca3bf7669a26ff22a8be8ea2e
contracts/DividendPayingTokenInterface.sol	4cd1b102451c554b62939ff3c289d5b7076262ae
contracts/SafeMath.sol	47a55484f7d77a847c0d2638da16315e6404c85c
contracts/IUniswapV2Router.sol	2c8abb8b8215d88897e9011ec4b03e5f4243c71f
contracts/SafeMathInt.sol	a0dc6f88e3a6ae8d51b551ac37b8cd93a53f2545
contracts/Ownable.sol	36cff8d460e596f55fcd07c93f6540e77f94a49f
contracts/IterableMapping.sol	1887d8153ee4f844e42b669ad50069f9310a5fcc
contracts/hero_battle.sol	960e4a20b049cddb3e0d9fa1f288a341876fe11b
contracts/ERC20.sol	35a78d307b1936c2ddc0cb609ffa1e8d15ae5ccf
contracts/DividendTracker.sol	c7bff1efbd3ccbb0fc4f59f5e2f186b89e0bbbf8
contracts/DividendPayingTokenOptionalInterface.sol	5feafe9dfe56ab29f8cca48beac4f1d75b15565e
contracts/IERC20.sol	1bdf256058bece58d0b44372a3e77d6d5df48cc3

# Метрики

## Исходные строки



## Уровень риска



## Возможности

### Компоненты

Контракты	Библиотеки	Интерфейсы	Абстрактный
5	4	8	1

### Открытые функции

В этом разделе перечислены функции, которые явно объявлены общедоступными или оплачиваемыми.

Обратите внимание, что методы получения для общедоступных переменных состояния не включены.

Общественные	К оплате
143	5

Внешний	Внутренний	Частный	Чистый	Вид
92	119	7	25	60

### Переменные состояния

Общий	Общественные
36	21 год

### Возможности

Твердость Версии наблюдаемый	Эксперимент аль Features	Жесткая банка Получать Фонды	Использует сборка	Имеет Разрушаемый Контракты
^ 0.6.2		да	**** (0 asm блоки)	

Переводы ETH	Низкий- Уровень Звонки	Делегат Вызов	Использует Хеш Функция s	ECRecov э	Новый/ Создавать/ Create2

да					да → NewCo ntract: Дивиден dTracke р
----	--	--	--	--	---



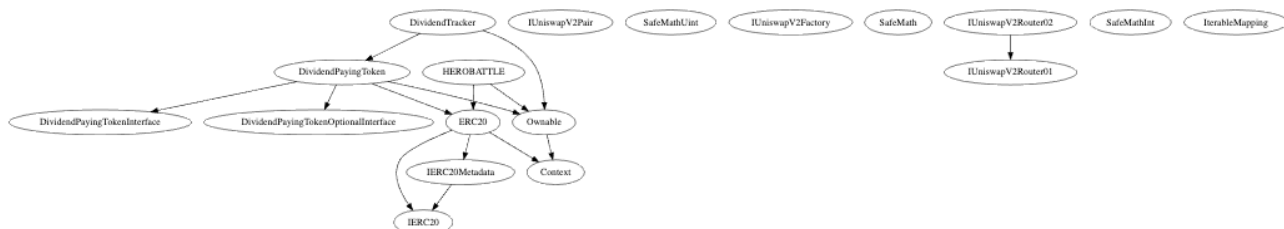
## Объем работ

Команда вышеупомянутого токена предоставила нам файлы, которые необходимо протестировать (Github, Bscscan, Etherscan, файлы и т. Д.). Объем аудита - это основной контракт (обычно то же имя, что и команда с добавлением .sol).

Мы проверим следующие утверждения:

1. Правильная реализация стандартного токена Deployer
2. не может создавать новые токены Deployer не может
3. сжечь или заблокировать средства пользователя
4. Deployer не может приостановить контракт Общая
5. проверка (Smart Contract Security)

## График наследования



## Проверить претензии

## Правильная реализация стандарта Token

Tested ✓	Verified ✓
	верифицированный

Функция	Описание	Существуют	протестированные	Проверено
TotalSupply	предоставляет информацию об общем поставка токенов	✓	✓	✓
Баланс	обеспечивает баланс счета счет владельца	✓	✓	✓
Передача	выполняет переводы указанного количества токенов на указанное адрес	✓	✓	✓
Трансфер от	выполняет переводы указанного количество токенов из указанного адрес ✓	✓		✓
Утвердить	разрешить спонсору забрать набор количество токенов из указанного учетная запись ✓	✓		✓
Разрешение	возвращает заданное количество токенов из спонсор собственнику ✓	✓	✓	✓

## Необязательные реализации

Функция	Описание	Существуют	протестированные	Проверено
отказаться от владения	Владелец отказывается от права собственности на больше доверия	✓	✓	✗

## Deployer не может чеканить новые токены

Имя	Существовать	Проверено	Проверено	Файл
Deployer не может МЯТА	✓	✓	✓	Главный
Комментарий	Линия: -			

Максимальное / общее предложение:

```

constructor() public ERC20("HEROBATTLE TOKEN", "HRB") {

    dividendTracker = new DividendTracker();
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x10ED43C718714eb63d5aA57B78B54704E256024E);
    // Create a uniswap pair for this new token
    address _uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory())
        .createPair(address(this), _uniswapV2Router.WETH());

    uniswapV2Router = _uniswapV2Router;
    uniswapV2Pair = _uniswapV2Pair;

    _setAutomatedMarketMakerPair(_uniswapV2Pair, true);

    // exclude from receiving dividends
    dividendTracker.excludeFromDividends(address(dividendTracker));
    dividendTracker.excludeFromDividends(address(this));
    dividendTracker.excludeFromDividends(owner());
    dividendTracker.excludeFromDividends(deadWallet);
    dividendTracker.excludeFromDividends(address(_uniswapV2Router));

    // exclude from paying fees or having max transaction amount
    excludeFromFees(owner(), true);
    excludeFromFees(_marketingWalletAddress, true);
    excludeFromFees(address(this), true);

    /*
        _mint is an internal function in ERC20.sol that is only called here,
        and CANNOT be called ever again
    */
    _mint(owner(), 100000000 * (10**18));
}

```

```

function _mint(address account, uint256 amount) internal virtual {
    require(account != address(0), "ERC20: mint to the zero address");

    _beforeTokenTransfer(address(0), account, amount);

    _totalSupply = _totalSupply.add(amount);
    _balances[account] = _balances[account].add(amount);
    emit Transfer(address(0), account, amount);
}

```

### Deployer не может сжечь или заблокировать средства пользователя

Имя	Существовать	Проверено	Проверено
Deployer не может замок	✓	✓	✓
Deployer не может гореть	✓	✓	✓

1. approve	→
2. blacklistAddress	→
3. claim	→
4. decreaseAllowance	→
5. excludeFromDividends	→
6. excludeFromFees	→
7. excludeMultipleAccountsFromFees	→
8. increaseAllowance	→
9. processDividendTracker	→
10. renounceOwnership	→
11. setAutomatedMarketMakerPair	→
12. setEnableFee	→
13. setFee	→
14. setMarketingWallet	→
15. setSwapTokensAtAmount	→
16. setTokenAddress	→
17. setTokenForMarketing	→
18. transfer	→
19. transferFrom	→
20. transferOwnership	→
21. updateClaimWait	→
22. updateDividendTracker	→
23. updateGasForProcessing	→
24. updateUniswapV2Router	→





## Развертыватель не может приостановить контракт





Имя	Существовать	Проверено	Проверено
Deployer не может Пауза	✓	✓	✓

1. approve	→
2. blacklistAddress	→
3. claim	→
4. decreaseAllowance	→
5. excludeFromDividends	→
6. excludeFromFees	→
7. excludeMultipleAccountsFromFees	→
8. increaseAllowance	→
9. processDividendTracker	→
10. renounceOwnership	→
11. setAutomatedMarketMakerPair	→
12. setEnableFee	→
13. setFee	→
14. setMarketingWallet	→
15. setSwapTokensAtAmount	→
16. setTokenAddress	→
17. setTokenForMarketing	→
18. transfer	→
19. transferFrom	→
20. transferOwnership	→
21. updateClaimWait	→
22. updateDividendTracker	→
23. updateGasForProcessing	→
24. updateUniswapV2Router	→

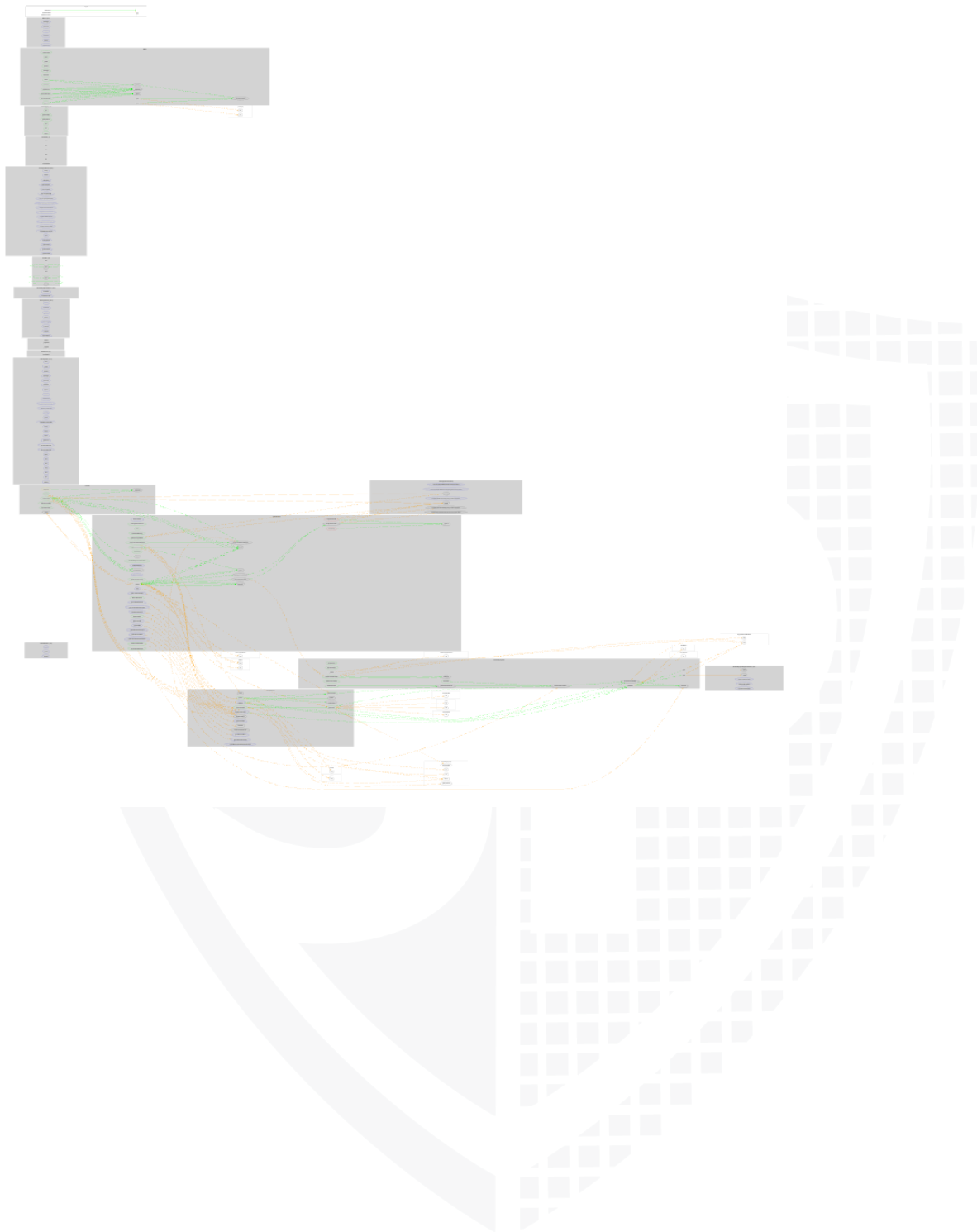
## Общий осмотр (безопасность смарт-контрактов)

Тес 	Вер 

### Легенда

Атрибут	Условное обозначение
Проверено / Проверено	
Частично проверено	
Непроверено / Не проверено	
Недоступен	

# CallGraph



## Источники в объеме

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	contracts/IERC20Metadata.sol	_____	1	27	16	4	15	9	
	contracts/DividendPayingToken.sol	1	_____	177	177	90	51	82	
	contracts/IUniswapV2Pair.sol	_____	1	54	9	5	1	55	_____
	contracts/SafeMathUint.sol	1	_____	15	15	8	5	3	_____
	contracts/Context.sol	1	_____	24	24	10	12	1	_____
	contracts/IUniswapV2Factory.sol	_____	1	19	8	4	1	17	_____
	contracts/DividendPayingTokenInterface.sol	_____	1	36	13	3	16	5	_____
	contracts/SafeMath.sol	1	_____	146	146	39	93	10	
	contracts/IUniswapV2Router.sol	_____	2	142	7	4	2	64	
	contracts/SafeMathInt.sol	1	_____	92	92	33	47	16	_____
	contracts/Ownable.sol	1	_____	57	57	27	21	25	_____
	contracts/IEnumerableMapping.sol	1	_____	63	63	49	2	19	_____
	contracts/hero_battle.sol	1	_____	440	418	289	22	294	
	contracts/ERC20.sol	1	_____	310	294	85	178	82	_____
	contracts/DividendTracker.sol	1	_____	221	203	144	2	95	_____
	contracts/DividendPayingTokenOptionalInterface.sol	_____	1	25	13	3	14	7	_____
	contracts/IERC20.sol	_____	1	81	26	17	57	13	
	<b>Totals</b>	<b>10</b>	<b>8</b>	<b>1929</b>	<b>1581</b>	<b>814</b>	<b>539</b>	<b>797</b>	

### Легенда

Атрибут	Описание
Линии	итоговые строки исходного блока
nLines	нормализованные строки исходного блока (например, нормализует функции, охватывающие несколько строк)
nSLOC	нормализованные строки исходного кода (только строки исходного кода; без комментариев, без пустых строк)
Строки комментариев	строки, содержащие одиночные или блочные комментарии
Оценка сложности	настраиваемая оценка сложности, полученная из операторов кода, которые, как известно, вносят сложность кода (ветки, циклы, вызовы, внешние интерфейсы, ...)

## Результаты аудита

# АУДИТ ПРОЙДЕН

### Критические проблемы

- критических проблем не обнаружено -

### Высокие проблемы

- серьезных проблем не обнаружено -

### Средние проблемы

- Средних проблем не обнаружено -

### Низкие проблемы

Проблема	Файл	Тип	Линия	Описание
# 1	hero_battle.sol	Отсутствует нулевой адрес Проверка (проверка на отсутствие нуля)	184, 134, 161, 56	Убедитесь, что адрес не равен нулю.
# 2	hero_battle.sol	Задана плавающая прагма	2	Текущая прагма Solidity директива - "" ^ 0.6.2 "".
# 3	дивиденды_acker.sol	Задана плавающая прагма	2	Текущая прагма Solidity директива - "" ^ 0.6.2 "".
# 4	дивиденды_acker.sol	Отсутствует нулевой адрес Проверка (проверка на отсутствие нуля)	43, 74, 146, 210	Убедитесь, что адрес не равен нулю.

### Информационные вопросы

Проблема	Файл	Тип	Линия	Описание
# 1	hero_battle.sol	Переменные состояния, которые можно объявить постоянными (констебл-состояния)	23	Добавьте атрибуты `constant` к переменные состояния, которые никогда не меняются.

## Комментарии аудита

18. августа 2021 г.:

- Все еще есть владелец (Владелец еще не отказался от права собственности)



## SWC атаки

Я БЫ	Заголовок	Отношения	Положение дел
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">6</a>	Незашифрованный Личные данные В сети	<a href="#">CWE-767: доступ к критической частной переменной через открытый метод</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">5</a>	Код без Эффекты	<a href="#">CWE-1164: несоответствующий код</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">4</a>	Сообщение вызов с участием жестко запрограммированный газ количество	<a href="#">CWE-655: Неправильно Инициализация</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">3</a>	Хеш-коллизии С несколькими Переменная Длина Аргументы	<a href="#">CWE-294: Обход аутентификации путем захвата-воспроизведения</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">2</a>	Непредвиденный Баланс эфира	<a href="#">CWE-667: неправильная блокировка</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">1</a>	Присутствие неиспользованный переменные	<a href="#">CWE-1164: несоответствующий код</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-13</a> <a href="#">0</a>	Справа налево- Переопределить контроль персонаж (U + 202E)	<a href="#">CWE-451: Пользовательский интерфейс (UI) Искажение важной информации</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">9</a>	Типографский Ошибка	<a href="#">CWE-480: Использование неправильного оператора</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">8</a>	DoS с блокировкой Лимит газа	<a href="#">CWE-400: неконтролируемый Потребление ресурсов</a>	прошедший

<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">7</a>	Произвольный прыжок с функцией Тип Переменная	<a href="#">CWE-695: Использование функциональности низкого уровня</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">5</a>	Неверно Наследование порядок	<a href="#">CWE-696: неправильный порядок действий</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">4</a>	Написать в Произвольный Место хранения Место нахождения	<a href="#">CWE-123: условие записи-где- то</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">3</a>	Требование Нарушение	<a href="#">CWE-573: неправильное следование спецификации вызывающим абонентом</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">2</a>	Отсутствие надлежащего Подпись Проверка	<a href="#">CWE-345: Недостаточно Проверка данных Подлинность</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">1</a>	Отсутствует Защита против Подпись Воспроизведение атак	<a href="#">CWE-347: неправильная проверка криптографической подписи</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-12</a> <a href="#">0</a>	Слабые источники случайности fromChain Атрибуты	<a href="#">CWE-330: Использование недостаточно случайных значений</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">9</a>	Затенение Переменные состояния	<a href="#">CWE-710: Несоблюдение стандартов кодирования</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">8</a>	Неверно Конструктор Имя	<a href="#">CWE-665: Неправильно Инициализация</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">7</a>	Подпись Пластичность	<a href="#">CWE-347: неправильная проверка криптографической подписи</a>	прошедший



<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">6</a>	Отметка времени Зависимость	<a href="#">CWE-829: включение</a> <a href="#">Функциональность из недоверенной</a> <a href="#">сферы управления</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">5</a>	Авторизация через tx.origin	<a href="#">CWE-477: Использование</a> <a href="#">устаревшей функции</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">4</a>	Сделка порядок Зависимость	<a href="#">CWE-362: одновременно</a> <a href="#">Выполнение с использованием общего доступа</a> <a href="#">Ресурс с неподходящим</a> <a href="#">Синхронизация ('Race</a> <a href="#">Состояние')</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">3</a>	DoS с ошибкой Вызов	<a href="#">CWE-703: неправильная проверка</a> <a href="#">или обработка исключительных</a> <a href="#">условий</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">2</a>	Делегатозвонить Ненадежный Callee	<a href="#">CWE-829: включение</a> <a href="#">Функциональность из недоверенной</a> <a href="#">сферы управления</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-111</a>	Использование Устарело Твердость Функции	<a href="#">CWE-477: Использование</a> <a href="#">устаревшей функции</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-11</a> <a href="#">0</a>	Заявить о нарушении	<a href="#">CWE-670: Всегда некорректная</a> <a href="#">реализация потока управления</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">9</a>	Неинициализированный Указатель хранилища	<a href="#">CWE-824: Доступ</a> <a href="#">Неинициализированный указатель</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">8</a>	Переменная состояния Дефолт Видимость	<a href="#">CWE-710: Несоблюдение</a> <a href="#">стандартов кодирования</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">7</a>	Реентерабельность	<a href="#">CWE-841: Неправильно</a> <a href="#">Обеспечение соблюдения</a> <a href="#">поведенческой работы</a>	прошедший
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">6</a>	Незащищенный САМОСТРОЕНИЕ Инструкция T	<a href="#">CWE-284: Неправильный контроль</a> <a href="#">доступа</a>	прошедший

<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">5</a>	Незащищенный Эфир Снятие	<a href="#">CWE-284: Неправильный контроль доступа</a>	ПРОШЕДШИЙ
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">4</a>	Непроверенный вызов Возвращаемое значение	<a href="#">CWE-252: непроверенное возвращаемое значение</a>	ПРОШЕДШИЙ
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">3</a>	Плавающий Прагма	<a href="#">CWE-664: Неправильный контроль ресурса через его Продолжительность жизни</a>	НЕТ ПРОШЕДШИЙ
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">2</a>	Устаревший Компилятор Версия	<a href="#">CWE-937: Использование компонентов с известными уязвимостями</a>	ПРОШЕДШИЙ
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">1</a>	Целое число Переполнение и Под течением	<a href="#">CWE-682: неправильный расчет</a>	ПРОШЕДШИЙ
<a href="#">Ю3</a> <a href="#">C-10</a> <a href="#">0</a>	Функция Дефолт Видимость	<a href="#">CWE-710: Несоблюдение стандартов кодирования</a>	ПРОШЕДШИЙ

The logo features the words "SolidProof" in a white, handwritten-style script. The "P" is particularly large and stylized, with a long horizontal stroke that extends to the left. The background is a solid blue color with a faint, large shield emblem. The shield has a grid-like pattern on its right side and a solid blue area on its left side.

SolidProof

**Blockchain Security | Smart Contract Audits | KYC**

A small horizontal bar representing the German flag, with black, red, and gold stripes.

MADE IN GERMANY