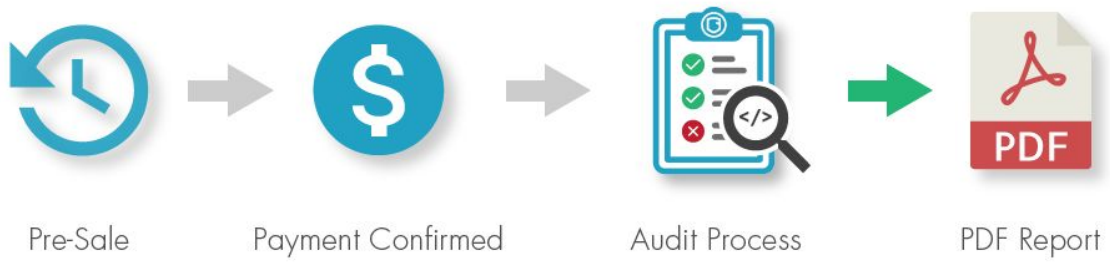
Auditorias De Smart Contracts

Conteúdo

Auditorias de smart contracts. O Processo de Auditoria de A-Z

Nas linhas abaixo, você aprenderá qual é o processo de auditoria completo da Coinfabrik, desde a pré-venda até o envio da auditoria.



A) Pré-venda

Somos contatados pelos clientes solicitando uma auditoria. Pedimos que enviem o código e a documentação para avaliar sua extensão, complexidade, viabilidade e propósito. Uma vez analisado por nossos especialistas, nós faremos um orçamento. Em seguida, enviamos por email o valor e a duração da auditoria. Se o cliente concordar com os termos, enviamos uma proposta formal com todos os detalhes da auditoria, preço, prazos e instruções de pagamento.

B) Pagamento Confirmado

Assim que recebermos o pagamento inicial começamos a trabalhar na auditoria. Iniciamos revisando a documentação fornecida e revisando todo o código fonte enviado. Se tivermos alguma dúvida, entramos em contato com o cliente via chat ou e-mail. Alguns clientes requerem contato direto, utilizemos o telegram onde enviamos as dúvidas e os erros encontrados antes do envio do documento de auditoria, para agilizar o processo e fazer com que corrijam os erros antes da entrega do relatório.

C) Começamos a auditar os contratos inteligentes.

Verificamos o código do smart contracts em busca de problemas técnicos de codificação, e, paralelamente, certificamo-nos de que a lógica do smart contracts faz sentido com a documentação entregue pelo cliente e que é muito importante. É por isso que pedimos ao cliente que nos envie a documentação mais detalhada possível, explicando especialmente o propósito do código e sua

motivação. Uma documentação clara nos ajuda em todo o processo de auditoria de segurança e reduz atrasos inesperados. Depois de entender o código e sua finalidade, iniciamos a análise do código. Fazemos o documento pdf descrito abaixo e o enviamos ao cliente para revisão e correção. Geralmente, antes de enviarmos o relatório completo da auditoria ao cliente, enviamos por chat ou e-mail os principais problemas dos smart contracts para agilizar o processo e fazer com que corrijam os erros antes de terminarmos o relatório.

Verificação de código: o projeto ou o propósito por trás dos contratos não é o foco de análise, a menos que vejamos algum problema nesse assunto. Nesse caso, precisaríamos fazer uma consultoria antes da auditoria, pois o caminho escolhido é obviamente um problema futuro para a segurança dos smart contracts. Porém, rodamos um compilador para verificar os avisos do código. E temos nossos próprios scripts que usamos para representar o código e analisar com facilidade. No entanto, esse processo nem sempre fornece informações úteis.

Duração da auditoria: tudo depende do código que você deseja auditar: seu tamanho, complexidade, protocolo, propósito e documentação fornecida. Para se ter uma ideia, se o código é uma simples liquidação coletiva, bem documentada e previamente agendada, demora cerca de 1 a 2 semanas se tudo correr bem e não estivermos trabalhando ao mesmo tempo em outras auditorias. Devido à importância do processo e dos padrões de qualidade que seguimos nas auditorias, apenas algumas pessoas de nossa equipe auditarão os contratos. Caso esta equipe esteja em outro projeto, é necessário aguardar até que estejam disponíveis. Entre em contato conosco e solicite os requisitos de cronograma e envie o link do código para verificar se podemos fazê-lo no prazo desejado.

D) Relatório PDF entregue inclui:

• Um resumo detalhado do que é o processo de auditoria, com o link para o smart contracts a ser auditado, sua função, a que o contrato se propõe e em que protocolo está escrito. Também explica quais são as análises realizadas.

• As descobertas: Descrevemos quais são os problemas encontrados no contrato e qualificamos os problemas pelo nível de gravidade (problemas de gravidade alta, média ou baixa).

• Aprimoramentos: com base nas melhores práticas de código, propomos alguns aprimoramentos para melhorar a semântica do código.

• Conclusão: é uma breve descrição dos itens mais importantes que observamos em todo o documento.

Este pdf será enviado ao cliente, podendo ser corrigido seguindo as nossas recomendações. Se descobrirmos que os erros foram corrigidos com êxito, marcamos esses problemas como corrigidos.

Após a auditoria ser concluída e corrigida pelo cliente: com a permissão do cliente, publicaremos a auditoria em nosso blog na categoria “Security”. O cliente pode fazer um link de volta em seu site e também pode usar nosso logotipo para notificar que seu smart contracts foi auditado pela equipe Coinfabrik.

Por que as Auditorias de smart contracts são tão importantes?

A segurança de smart contracts é uma questão a ser observada com seriedade. Falhas de segurança, erros e ineficiência são muito caros quando você implanta um smart contracts no Blockchain. As empresas estão especialmente preocupadas com seu código de smart contracts porque, uma vez executado, não há como voltar atrás (eles são irreversíveis) e valores podem ficar “presos” no blockchain e inacessíveis perpetuamente.

Assim, para garantir que seu código seja escrito corretamente, as empresas geralmente contratam auditores externos conhecidos (como a Coinfabrik) pois sabem que um problema em seu código pode ter um custo muito expressivo se ignorarem a auditoria. Smart contracts são usados para mover, armazenar ou distribuir fundos, e por isso erros no código e design do smart contracts devem ser corrigidos. Além disso, desde o surgimento dos ICOs nos últimos anos e com seu boom em 2017 e 2018, as auditorias de segurança de smart contracts se tornaram um dos serviços mais solicitados na indústria de blockchain.

Quais são os tipos de contratos inteligentes?

- Contratos totalmente compatíveis com o padrão da cadeia sem transferência de ether (ou seja, tokens ERC20, tokens ERC721): são os mais fáceis de auditar, pois devem seguir um padrão, que restringe seu design. Estar totalmente conectado e sem transferências de ether reduz a gravidade dos ataques e a área de superfície das explorações.
- Contratos totalmente em cadeia com transferência Ether (ou seja, Crowdsales / ICOs): são um pouco mais difíceis de auditar, pois não seguem um padrão. Em alguns casos, como em Crowdsales, existem várias implementações de contrato aberto que os desenvolvedores de contrato podem seguir, o que torna o processo mais fácil graças à familiaridade com o código. Mesmo estando totalmente em cadeira, esses tipo de smart contracts nao sao os mais difíceis de serem auditados. Veja a seguir o porquê.
- Contratos com interação fora da cadeia: são os mais difíceis de auditar, pois envolvem algum componente crítico fora do blockchain. Podem ser oráculos, canais ou qualquer outra coisa que constitua uma parte essencial no fluxo da referida interação do contrato. Auditar totalmente esses contratos pode ser difícil ou até impossível, pois pode haver algum componente fora da cadeia inacessível que precisa ser confiável de antemão. Mesmo assim, podemos auditar esses contratos presumindo que essas peças sejam confiáveis, desde que levados em consideração.

O que acontece quando smart contracts não são devidamente seguros?

Problemas comuns de design:

- Superengenharia / Alta Complexidade: Este é um problema comum no desenvolvimento de softwares modernos em geral, especialmente em projetos desenvolvidos por vários programadores. Esse tipo de problema geralmente está relacionado a desenvolvimentos sem smart contracts e muitas pessoas permitem isso como uma tradeoff para uma implantação mais rápida. Porém, no desenvolvimento de smart contracts isso não é negociável, a alta complexidade aumenta significativamente a área de superfície de explorações, reduz enormemente a capacidade de auditoria e aumenta os recursos usados pelo ambiente que geralmente são muito limitados e caros (ou seja, gás no caso de Ethereum).

Quais são os riscos envolvidos nos contratos inteligentes de vendas coletivas?

Para explicar melhor quais são os riscos de construir um ICO/IEO com problemas de segurança e quais são suas consequências, sugerimos a leitura sobre a história e consequências do DAO ou DAO História e consequências. Foi utilizado um bug no código do smart contract para pegar (de forma indevida) mais éters do caller que poderia lidar com o roubo de 3,6 milhões de ETH em poucas horas.

Dois problemas críticos foram usados com sucesso pelo invasor:

- Não foram consideradas chamadas recursivas no contrato.
- O smart contract enviou os Ethers primeiro e depois atualiza o saldo.

Portanto, o código de solidez deve ser escrito correta e cuidadosamente para evitar que algo semelhante aconteça com seus tokens.

E por isso é tão importante realizar auditorias de segurança externas para verificar o código e tentar encontrar qualquer vulnerabilidade antes de executar o código, quando muitas vezes o dano é irreversível. Sugerimos que as auditorias de segurança não sejam consideradas um custo, mas um investimento. A possibilidade de perdas de fundos é ilimitada caso haja um ataque no código.

É necessário agendamento para auditar seus contratos pela Coinfabrik?

Certamente! Temos uma quantidade limitada de auditores em nossa equipe que pode estar trabalhando em outras auditorias, por isso é importante reservar sua vaga para que o processo seja feito no prazo necessário. Antes de perguntar sobre os preços, informe-nos quando você precisar que a auditoria seja concluída e quando gostaria de começar a fazê-la.

Vulnerabilidades de smart contracts e ataques sofridos

- Reordering attack.
- Replay attack.
- Stack overflow e underflow
- Short address attack

Como economizar gás em contratos inteligentes?

Economizar gas é necessário para construir um smart contract eficiente. É um dos principais problemas que os desenvolvedores enfrentam porque nem todos sabem fazer isso da maneira correta. A primeira coisa a entender é como otimizar o código para economia de gas e quais instruções são as que mais consomem gas e como evitar ou minimizar seu uso.

O gas gasto em cada instrução pode ser referenciado nos 2 seguintes documentos do Google:

Custo do gas 1

Gas Custos 2

Abaixo estão nossas recomendações para seus contratos para economizar *gas*:

- Não use seu contrato como um armazenamento de dados.
- Considere que criar contrato é uma das instruções mais caras. Não crie contratos vazios.
- O armazenamento de novos dados gasta 20.000 gas e 5.000 gas para modificar os dados. Evite declarações redundantes de declarações caras, como SSTORE.

Por exemplo:

A.

```
uint256 public sum;
// ....

for (uint256 i = 0; i < 50; ++i) {
 // instructions
 ++sum;
}
```

Você está fazendo 50 iterações na variável sum. Esta execução está consumindo 20.000 * 50 de gas, tornando esse código ineficiente.

B.

```
for (uint256 i = 0; i < 50; ++i) {
 // instructions
}
sum += 50;
```

Considerando que, neste exemplo, você está executando o comando de armazenamento apenas uma vez no final da interação. O código faz o mesmo, mas consome 50 vezes menos gas.

- Chamada de contratos externos para consumo de 1.400 gas: não separe vários contratos se não for necessário. Você pode resolver isso criando várias heranças.
- Use o algoritmo hash integrado mais barato, amenos que seja necessário. Consumo de gas: *ripemd160> sha256> keccak256*. Use o que consome menos.
- Certifique-se de transferir para um endereço conhecido (call, send, transfer). Se você enviar para um endereço nunca visto, poderá ter que pagar mais 25.000 de gas.
- Tente combinar os loops quando puder.

Ex:
a)

```
function looping ( uint x ){
   uint a = 0;
   uint b = 0;
   for ( uint i = 0 ; i < x ; i++)
       a += i;
   for ( uint j = 0 ; j < x ; j++)
       b -= j; }
```

b)
Se você combinar ambas as instruções, você pode economizar gas.

```
function looping ( uint x ){
   uint a = 0;
   uint b = 0;
   for ( uint i = 0 ; i < x ; i++) {
      a += i;
      b -= i; }
// ...
}
```

Use variáveis locais para economizar gas quando você tiver declarações de iteração:

```
uint count=0;
function a1 ( uint x ){
    for ( uint i = 0 ; i < x ; i++)
        count += i; }
```

- No exemplo acima, a variável global “count” faz n iterações e consomem n vezes o gas usando uma variável local.

```
uint count = 0;
function a1 ( uint x ){
    uint local = 0;
    for ( uint i = 0 ; i < x ; i++)
        local += i;
    count += local;
}
```

• Nunca realize um call em uma biblioteca que faz algo simples quando pode ser feito facilmente dentro do contrato. Você evitaria acionar a biblioteca e executá-la, o que gasta muito gas desnecessariamente.

• Use bytes1 fixos a bytes32 se puder limitar o número de bytes. Nunca use bytes [] a menos que seja estritamente necessário. Bytes [] ocupa 31 bytes por elemento toda vez que você o aciona.

• Ao usar operadores &&, e quando se realiza esta operação a && b, você deve considerar duas coisas. 1) O custo de cada declaração: Se a for menos caro, você economiza gás fazendo um short-circuiting quando a for falso eb, que é mais caro de operar.

2) Certifique-se de que a tem mais probabilidade de retornar falso do que b, porque se a for falso, b não é considerado e você tem mais chances de economizar gás.

• Quando você usa OR || operador: a || b, considere duas questões.

1. O custo de cada afirmação: Se a for menos caro, você economiza gas fazendo um short-circuiting quando a for verdadeiro e b não é considerado porque a operação completa é verdadeira. Assim você evita executar b, que é mais caro do que a.
2. Certifique-se de que a tem mais probabilidade de retornar verdadeiro do que b, porque se a for verdadeiro, b não é considerado e você tem mais chances de economizar gas. Mais informações sobre como economizar gas podem ser encontradas em:

- [Solidity] Smart Contract Gas Saving
- How to write an optimized (gas-cost) smart contract?
- Under-Optimized Smart Contracts Devour Your Money

É caro auditar contratos inteligentes?

A auditoria de contratos inteligentes é uma das tarefas mais críticas em projetos de blockchain porque a segurança deve ser a prioridade para evitar problemas importantes após a execução do código. Não fazer isso pode fazer com que milhares de dólares não vão a lugar nenhum e se percam, ou ainda, na melhor das hipóteses, os contratos desperdicem mais gas do que o necessário devido à ineficiência.

Lembre-se de que, ao gerenciar contratos inteligentes, geralmente está envolvido muito dinheiro das pessoas que confiaram em seu projeto. É importante visualizar o problema em que sua empresa estaria envolvida se depois de fazer uma crown sale o código fosse hackeado e todo o dinheiro fosse perdido. Portanto, é por isso que todas as trocas online importantes exigem que você execute uma auditoria

externa em seus tokens e contratos inteligentes, a fim de proteger seus investidores e suas empresas que listam seus tokens.

Com essas questões em mente, a empresa de auditoria é contatada para encontrar possíveis problemas de segurança, geralmente em um tempo muito curto (especialmente em crowd sales, onde o tempo é importante). É uma grande responsabilidade encontrar bugs, possíveis falhas e problemas de ineficiência nos contratos antes de serem executados. É um serviço aprofundado que precisa ser feito corretamente pois valores estão envolvidos nas transações, e se um erro não for encontrado e o contrato for executado, não há como voltar atrás (são irreversíveis) e os fundos poderiam ser perdidos para sempre, como afirmado acima.

Além disso, tendo em conta o quão sensível é, as auditorias devem ser realizadas por especialistas na área de programação cujos perfis estão à altura do trabalho exaustivo e meticuloso e devem ter experiência tanto em programação de smart contracts como em segurança.

Nossa equipe possui um departamento dedicado que trabalha exclusivamente com auditorias de segurança. Somos um dos líderes neste serviço em todo o mundo, tendo feito mais de 50 auditorias nesse segmento.

Quanto custa uma auditoria de smart contracts?

Para saber o custo da auditoria de seus contratos inteligentes, é necessário nos enviar o código (de preferência enviar o código do github) e o cronograma desejável de conclusão. Responderemos com os preços e o tempo que levaremos para finalizar o processo. Leve em consideração que um serviço como a auditoria não é um serviço de preço fixo pois cada código é diferente e sempre precisamos verificá-lo além do tipo de código a auditar antes de citá-lo.

Quem são os auditores de smart contracts na Coinfabrik?

Atualmente, temos 4 auditores de smart contracts em nossa equipe, sendo 3 deles 100% dedicados a esta tarefa. Ao menos 2 integrantes da equipe desenvolvem contratos inteligentes em outros projetos e auditam os contratos em seu tempo livre, ajudando os outros 3 auditores a encontrar bugs / problemas no código.

Por que a Coinfabrik é uma boa opção para auditar contratos inteligentes?

Somos uma empresa de segurança web conhecida mundialmente e trabalhamos no setor de segurança há mais de 20 anos. Iniciamos com a empresa Nektra.com e então com a Coinfabrik. Temos 6 anos de experiência trabalhando em segurança de blockchain e auditamos mais de 50 contratos inteligentes nesse tempo, os quais todos seguem executando sem problemas desde que foram verificados por nossa equipe.

Consideramos a segurança algo sério, e nosso logotipo é bem conhecido no mercado de blockchain, e ser auditado por nós traria ao seu smart contract confiança e prestígio para seus investidores/clientes. Auditamos ICOs importantes como Status, mona.co, Patientory, que juntos arrecadaram centenas de milhões e tiveram sucesso em captar fundos em suas vendas coletivas ais quais foram auditadas por nós.

Além disso, fomos listados em:

Top security audit companies list

Top 7 Blockchain Development Companies in 2020

GoodFirms Unveiled the Leaders in Development, Designing and Marketing for Q1 2019

15 Latin American Bitcoin & Blockchain Startups You Need to Know

7 companias en Argentina para desarollar tecnologia blockchain

Top 10 Crypto Web Development Company

An Overview Of Latin America's Blockchain Adoption

Buenos Aires: la capital bitcoin de Latinoamerica

Crypto expert talks Libra misunderstandings and Nahmii potential

Meet the Bitcoin Foundation's Newest Core Security Auditor, Sergio Demian Lerner

Los argentinos que ya estan en la blockchain

Decoder: Is "Dogethereum" is another joke? And what does it mean for Dogecoin?

Especialistas divergem sobre as aplicações das criptografias quânticas

Nektra SpyStudio goes freeware

HEX Cryptocurrency Lauch Completes Successful First Month, with well Over a Billion Dollars in Bitcoin Claimed & 50k Ethereum Transformed

List of Blockchain Consulting Companies

Etherparty Launches World's First Consumer-Ready Smart Contract Application, 'Rocket'

Nós também colaboramos com Solidstamp, enviando todas as nossas auditorias públicas de smart contracts as quais estão listadas no blog.

Últimas Auditorias da Coinfabrik por ano

2020 Smart Contract Audits

YFFII Smart Contract Audit

Wise Smart Contract Audit

YELD Security Incident Report

Icherry Smart Contract Audit

Katana Security Audit

Securitize Smart Contract Audits

Feyorra Smart Contract Audits

TOSC Smart Contract Audits

HEX Financial Audit

2019 Smart Contract Audits

Polymath Core Audit

HEX Smart Contract Audit

Money on Chain Security Audit IV

Bitpara Smart Contract Audit

Money on Chain Security Audit III

Timvi Smart Contract Audit

Money on Chain Gas Cost

Aeternity Grand Final Report

Nahmii Security Audit

ArcadierX Security Audit

EasyPool Smart Contract Security Audit v2

2018 Smart Contract Audits

Inlock Smart Contract Audit Report

Bricks4us Token Audit

Zeex Token Auditing Report

Stasis Token Smart Contract Audit

Pixie Token Sale Audit

Casper CST Token

Dreamteam Token

CryptoCup

EtherParty

Inbest

Cryptosolartech

Dreamteam Token Sale

Ripio Credit Network (RCN) v2

Send (SDT)

Rightmesh

Beluga Pay (BBI)

Ripio Credit Network (RCN) | BasicCosigner

2017 Smart Contract Audits

Theta Token Sale

DM Token

CDX

Worldbit

Ripio Credit Network v1

Realisto

Flixxo

Patientory (PTOY)

Conclusão

O processo de auditoria de smart contracts foi planejado para ser o mais direto possível, pois sabemos que na maioria das vezes as auditorias são urgentes devido às vendas coletivas da ICO que têm seus prazos fixos a serem respeitados.

Sendo assim, para agilizar o processo nos certificamos de que o cliente entenda toda a documentação e código necessários, o custo da auditoria, como se dá o processo e seu ETA (prazo de entrega). Nossos auditores serão notificados com os detalhes e cronogramas e iniciarão a auditoria após confirmação. Pode-se abrir um grupo de telegram para discutir o código ou documentação caso nossa equipe tenha alguma dúvida.