Proof of concept

Système de détection des attaques Flash Loan

Etude sur la complexité des Flash Loans

La détection des attaques de Flash Loans représente l'un des défis les plus complexes de la sécurité blockchain. Contrairement aux attaques traditionnelles, ces exploits sont extrêmement difficiles à identifier pour plusieurs raisons fondamentales.

Premièrement, un Flash Loan se déroule en quelques microsecondes, occupant souvent moins d'une seconde. Cette ultra-rapidité rend la détection presque impossible avec des méthodes conventionnelles. L'attaquant emprunte et rembourse un montant astronomique dans le même bloc de transaction, effaçant presque instantanément toute trace.

Deuxièmement, ces attaques exploitent des failles de conception dans des smart contracts, utilisant des mécanismes de prêt sans garantie pour réaliser des arbitrages complexes. Un attaquant peut potentiellement manipuler plusieurs protocoles financiers simultanément, créant des interactions si rapides et sophistiquées qu'elles défient toute analyse traditionnelle.

Les Flash Loans représentent aujourd'hui l'une des menaces les plus sophistiquées dans l'écosystème des cryptomonnaies. Ces attaques exploitent la capacité unique des blockchain à réaliser des emprunts instantanés sans garantie collatérale, permettant des opérations financières complexes en quelques fractions de seconde.

Mécanisme de détection

J’ai développé un système de détection qui repose sur trois analyses complémentaires : l'analyse des patterns transactionnels, l'analyse des signatures et empreintes cryptographiques, et l'analyse quantitative des flux financiers.

Méthode 1: Analyse des patterns transactionnels

L'observation des séquences de transactions constitue notre première ligne de défense. Un Flash Loan se caractérise généralement par une série d'opérations rapides et interconnectées. Nous surveillons donc la dynamique globale des transactions, leurs interactions, et leurs temporalités.

Un algorithme avancé scrute chaque bloc, détectant les mouvements financiers qui sortent des schémas habituels. Les signaux d'alerte incluent des transferts multiples en quelques microsecondes, des interactions entre contrats inhabituelles, et des volumes de transactions anormaux.

Méthode 2: Signatures et empreintes cryptographiques

Chaque transaction blockchain possédant une signature unique, j’ai développé une base de données de signatures caractéristiques des Flash Loans ainsi que pour le POC, la signature de Euler Finance.

En comparant les signatures des transactions suspectes avec ces références, cela permet de rapidement identifier des tentatives potentielles d'exploitation.

Cette méthode permet de cibler de potentiels contract de Fast Loan dont on ne connaîtrait pas la provenance. Les modèles d'appels de contrats, les séquences de méthodes invoquées deviennent nos indices principaux.

Méthode 3: Analyse quantitative des flux financiers

Au-delà des patterns, il faut examiner rigoureusement les montants transférés. Un Flash Loan se caractérise souvent par des mouvements financiers massifs et ultra-rapides. L'objectif de l'algorithme est de pouvoir calculer ce flux en temps réel. Pour le moment, il se concentre uniquement sur des sommes importantes, mais dans le futur, il faudrait se pencher sur trois points fondamentaux concernant cette analyse:

- La vélocité des transactions
- Les écarts par rapport aux flux financiers normaux
- La cohérence des montants entre différents contrats

Infrastructure technique utilisée

Pour répondre aux besoins concernant mon système de détection de Flash Loans. J'ai choisi d'utiliser la librairie **Ethers.js**, qui utiliser des appels JSON RPC pour les interactions avec la blockchain en temps réel récupérer les données nécessaires.

Pour me connecter au RPC d'Ethers, j'utilise **INFURA**, un fournisseur réputé de données en temps réel, ce qui me permet d'obtenir les informations nécessaires pour détecter les attaques. Pour le POC, je n'ai pour le moment pas eu le besoin de faire appel à d'autres librairies car elle me fournit toutes les informations nécessaires concernant l'analyse du bloc à notre stade d'avancement.

J'ai utilisé comme base pour développer mon système, **Node.js** qui me permet de construire un server virtuel et de monter une REST API sur une base de **Javascript** et de **TypeScript** pour définir les types de nos variables et des fonctions.

Sur ce server, j'ai créé une route API pour fournir une interface de programmation pour les utilisateurs et les autres systèmes qui souhaitent intégrer mon système de détection.

En plus de cela, j'ai crée en support des décodeurs hexadécimaux personnalisés qui me permettent de transformer les données brutes de la blockchain en informations exploitables.

Ajout d'un filtre du niveau de sévérité de manière dynamique

La simple détection ne suffit pas. J'ai ajouté à cela un système de scoring dynamique qui permet de donner un niveau pour chaque indicateur. Un score élevé ne signifie pas automatiquement une attaque, mais déclenche différents niveaux d'alerte et d'investigation, si le niveau est « Low », nous n'avons pas de suspicion d'attaque, à partir de « Medium », la suspicion n'est pas gravissime mais mérite d'être surveillée, mais à partir de « Hard » ou « Critique » une alerte peut-être déclenchée car cela signifie qu'il y'a un potentiel danger.

Quelles seraient le solutions pour améliorer le système

La sécurité blockchain est un domaine en constante mutation. Ce système n'est pas une solution à long terme mais permet d'analyser la situation sur le moment, il doit être développer pour être capable d'apprendre et de s'adapter pour les évolutions futures. Des mises à jour régulières doivent être réalisées afin de pourvoir transformer cet outil de détection en un bouclier évolutif.

La principale problématique concernant l'analyse du bloc complet, est qu'il peut mettre plusieurs secondes à retourner une réponse. Ce phénomène est due au traitement d'une grande quantité de transactions dans le bloc. Il serait donc bénéfique d'utiliser ce système sur un mécanisme type oracle afin de pouvoir analyser les transactions en temps réel et ainsi avoir moins de volume à gérer, ce qui permettrait un traitement instantané et plus de précision.

Concernant les trois différentes méthodes, il s'agit pour le moment d'une première version qu'il faudrait améliorer.

Il faudrait améliorer la détection de signatures car pour le moment j'ai crée une base de donnée basique, modifier cette base nous permettrait d'avoir beaucoup plus de chance de détecter des contracts de type Fast Loan au sein des blocs car tous, n'utilisent pas uniquement ces système de signature. Cette detection de signature est complexe sans l'appui des entreprises professionnelles concernées.

Pour les detections multi transactions, pour le moment j'ai mis un niveau de détection pour une succession supérieur à 2 transactions, mais il faudrait définir en analysant de nombreux hacks de ce type pour pouvoir définir un nombre minimal pour le seuil d'alerte.

L'analyse du montant de la transaction peut-être utilisé en tant que support supplémentaire pour appuyer les deux autres méthodes afin d'établir si nous devons exécuter une alerte ou non pour éviter les fausses alertes.

Il existe un grand nombre de smart contracts de différent types, certains n'implémentent pas les normes ERC20, il est donc difficile d'analyser ces contracts automatiquement car il ne respectent pas les standards. Il est donc difficile d'identifier un Flash Loans sans vérification préalable. Un algorithme ayant pour rôle d'analyser le contenu des contracts en fonction d'une adresse donnée, permettrait de renforcer notre analyse et d'obtenir beaucoup plus de précision sur notre analyse.

Pour le moment je me suis appuyé sur une base qui peut réellement être développée dans le futur et je serai ravi de pouvoir y contribuer car je trouve ces sujets très intéressants et surtout utiles pour la protection d'autrui.