Proof of concept

Flash Loan attack detection system

Study on the complexity of Flash Loans

Detecting Flash Loans attacks is one of the most complex challenges in blockchain security. Unlike traditional attacks, these exploits are extremely difficult to identify for several fundamental reasons.

Firstly, a Flash Loan takes place in a few microseconds, often occupying less than a second. This ultra-rapidity makes detection almost impossible using conventional methods. The attacker borrows and repays an astronomical amount in the same transaction block, erasing all traces almost instantly.

Secondly, these attacks exploit design flaws in smart contracts, using collateral-free lending mechanisms to perform complex arbitrage. An attacker can potentially manipulate several financial protocols simultaneously, creating interactions so fast and sophisticated that they defy traditional analysis.

Flash Loans represent one of the most sophisticated threats in the cryptocurrency ecosystem today. These attacks exploit the blockchain's unique ability to make instant loans without collateral, enabling complex financial transactions in fractions of a second.

Detection mechanism

I have developed a detection system based on three complementary analyses: analysis of transactional patterns, analysis of cryptographic signatures and fingerprints, and quantitative analysis of financial flows.

Method 1: Analysis of transactional patterns

Observing transaction sequences is our first line of defense. A Flash Loan is generally characterized by a series of rapid, interconnected transactions. We therefore monitor the overall dynamics of the transactions, their interactions and their timing.

An advanced algorithm scans each block, detecting financial movements that depart from the usual patterns. Warning signs include multiple transfers in a matter of microseconds, unusual interactions between contracts, and abnormal transaction volumes.

Method 2: Cryptographic signatures and fingerprints

As each blockchain transaction has a unique signature, I have developed a database of signatures characteristic of Flash Loans and, for the POC, the signature of Euler Finance. By comparing the signatures of suspect transactions with these references, we can quickly identify potential attempts to exploit them.

This method makes it possible to target potential Fast Loan contracts of unknown origin. The patterns of contract calls and the sequences of methods invoked become our main clues.

Method 3: Quantitative analysis of financial flows

In addition to patterns, the amounts transferred must be rigorously examined. A Flash Loan is often characterized by massive, ultra-rapid financial movements. The aim of the algorithm is to be able to calculate this flow in real time. For the moment, it only focuses on large sums, but in the future it will be necessary to look at three fundamental points in this analysis:

- The velocity of transactions
- Deviations from normal financial flows
- The consistency of amounts between different contracts

Technical infrastructure used

To meet the needs of my Flash Loans detection system. I chose to use the **Ethers.js** library, which uses JSON RPC calls to interact with the blockchain in real time and retrieve the necessary data.

To connect to the Ethers RPC, I use **INFURA**, a reputable provider of real-time data, which allows me to obtain the information I need to detect attacks. For the POC, I haven't needed to use any other libraries for the time being because it provides me with all the information I need to analyse the block at our stage of progress.

As a basis for developing my system, I used **Node.js**, which lets me build a virtual server and set up a REST API based on **Javascript** and **TypeScript** to define the types of our variables and functions.

On this server, I've created an API route to provide a programming interface for users and other systems wishing to integrate my detection system.

In addition to this, I've created custom hexadecimal decoders that allow me to transform the raw data from the blockchain into usable information.

Dynamic addition of a severity level filter

Simple detection is not enough. I've added a dynamic scoring system that assigns a level to each indicator. A high score does not automatically mean an attack, but triggers different levels of alert and investigation. If the level is ‘Low’, we have no suspicion of an attack, from ‘Medium’, the suspicion is not serious but deserves to be monitored, but from ‘Hard’ or ‘Critical’ an alert can be triggered because it means there is a potential danger.

What are the solutions for improving the system?

Blockchain security is a constantly evolving field. This system is not a long-term solution, but rather a means of analyzing the situation at the time. It needs to be developed so that it is capable of learning and adapting to future developments. Regular updates are needed to transform this detection tool into an evolving shield.

The main problem with analyzing the complete block is that it can take several seconds to return a response. This is due to the large number of transactions processed in the block. It would therefore be beneficial to use this system on an oracle-type mechanism in order to be able to analyse

transactions in real time and thus have less volume to manage, which would enable instant processing and greater accuracy.

As far as the three different methods are concerned, for the moment this is a first version that needs to be improved.

Signature detection needs to be improved, because for the moment I've created a basic database. Modifying this database would give us a much better chance of detecting Fast Loan type contracts within blocks, because not all of them use these signature systems alone. Signature detection is complex without the support of the professional companies concerned.

For multi-transaction detection, for the moment I've set a detection level for a succession of more than 2 transactions, but it would be necessary to analyse numerous hacks of this type to be able to define a minimum number for the alert threshold.

The analysis of the amount of the transaction can be used as additional support to support the other two methods in order to establish whether or not we need to execute an alert to avoid false alerts.

There are a large number of different types of smart contract, some of which do not implement the ERC20 standards, so it is difficult to analyse these contracts automatically because they do not comply with the standards. It is therefore difficult to identify a Flash Loan without prior verification. An algorithm whose role is to analyse the content of contracts based on a given address would enable us to strengthen our analysis and obtain much greater precision.

For the time being, I've built on a foundation that could be developed in the future and I'd be delighted to be able to contribute to it because I find these subjects very interesting and, above all, useful for the protection of others.