[

Forest_banner2

1920×845 143 KB

](https://europe1.discourse-cdn.com/business20/uploads/aave/original/2X/1/15fa4ae34b21cc1ae5f79b8fa80a2aad48f04d8d.jpeg)

TL;DR

Start the community discussion around the concept of an

Aave Forest

: a monitoring and prevention framework to monitor and protect Aave against potential exploits.

Context

Public transparent blockchains and especially DeFi protocols, live in a highly adversarial environment, with malicious parties trying to exploit any type of vulnerability to get financial profit.

Additionally, the entry barrier for anybody to perform a good part of the exploits is really low: capital is usually not required in advance given the presence of flash loans, so it boils down to having the expertise to identify the exploit and how to execute it, including pre and post obfuscation.

Since its first version, Aave has been continuously improving all the security procedures, starting with the protocol design and implementation, and going through multiple other mechanisms like extensive testing (in isolation and simulations), independent security reviews, formal verification techniques, or bug bounty programs.

But in a system like Aave, involving external components as tokens or the networks themselves where the protocol lives, it is technically impossible to have 100% security assurance; it is a process of continuous improvement.

Lately, new types of technologies (or their application on DeFi systems) are starting to show results in the fields of live monitoring, so we think it is time for Aave to start moving and integrating them, coordinating them with a framework/architecture we call Aave Forest

.

State-of-art in DeFi monitoring and exploit prevention

Real-time exploits monitoring and prevention is a well-known field in software security. The idea is relatively simple: it is usually possible to infer patterns from exploits and the pre-exploit behavior of the attacker by collecting, analyzing, and processing historic security incidents and their metadata. Afterward, the product of that analysis can be used to detect and/or prevent future incidents.

The amount of data required for these techniques to work (frequently based on techniques like machine learning) is considerable, so even months ago, all platforms trying to create this type of monitoring system for DeFi were not really mature, showing, for example, quite an important rate of false positives, or simply not having enough data to be able to detect meaningful attack patterns.

This has changed lately, with multiple teams showing evident results on the detection of real attacks on DeFi, that if properly integrated into the attacked systems, would have protected them sometimes totally, sometimes partially from loss of funds.

From BGD, we have been following the field and in contact with the teams behind these initiatives. And lately, their results are becoming impressive.

The following is an extract of some of those platforms and their capabilities/results:

Hypernative

https://www.hypernative.io/

- IEarn exploit
- https://twitter.com/HypernativeLabs/status/1646408384814432257
- https://twitter.com/HypernativeLabs/status/1646408384814432257
- Euler exploit

- https://twitter.com/HypernativeLabs/status/1635250158727491584
- https://twitter.com/HypernativeLabs/status/1635250158727491584
- Allbridge exploit
- https://twitter.com/HypernativeLabs/status/1642527379795988481
- https://twitter.com/HypernativeLabs/status/1642527379795988481
- Others
- https://twitter.com/HypernativeLabs/status/1608376668493873153
- https://twitter.com/HypernativeLabs/status/1616026574851297282
- https://twitter.com/HypernativeLabs/status/1615309912711520258
- https://twitter.com/HypernativeLabs/status/1608376668493873153
- https://twitter.com/HypernativeLabs/status/1616026574851297282
- https://twitter.com/HypernativeLabs/status/1615309912711520258

Hexagate

https://www.hexagate.com/

- iearn exploit
- https://twitter.com/hexagate_/status/1646396462559887360
- https://twitter.com/hexagate_/status/1646396462559887360
- Euler exploit
- https://twitter.com/hexagate_/status/1635204228523368452
- https://twitter.com/hexagate_/status/1635204228523368452

Forta

https://forta.org/

- Sushi exploit
- https://twitter.com/FortaNetwork/status/1645522583637426209
- https://twitter.com/FortaNetwork/status/1645522583637426209
- Euler exploit
- https://twitter.com/FortaNetwork/status/1639290581531889665
- https://twitter.com/FortaNetwork/status/1639290581531889665

Spotter (Pessimistic)

https://spotter.pessimistic.io/

- Paraspace exploit
- https://twitter.com/sadspotter/status/1636623959310364673
- https://twitter.com/sadspotter/status/1636623959310364673

Taking these results into account, from a high-level perspective, it is obvious that Aave needs to start exploring and integrating these tools if they give additional security assurance

.

Aave Forest: a wrapper of technologies

Different from other projects from BGD, Aave Forest is more of a framework than an implementation and is powered by different existing native features of Aave, combined with external platforms.

A recap of the different components and actors on this proposed framework is the following:

- Owls.

External platforms to Aave, with DeFi monitoring technology able to detect exploit patterns potentially before they get executed. They are a non-invasive party, as a trustable entity able to notify the Aave community representatives (more later) of any potential incident.

- Aave v3 roles.

The core mechanism of Aave v3, allows granting granular permissions over all the protocol's different risk/security levers, like changing listing new assets, changing risk parameters, freezing an asset, or even pausing a pool.

- Rangers.

Smart contracts or entities trusted by the community who have Aave v3 roles in executing certain protective actions over the protocol. For example, a Ranger could have held a role for being able to freeze an Aave reserve if an Owl would detected an upcoming exploit.

These 3 components connect to each other in a pretty simple way:

- The Aave community gives v3 roles to Rangers to allow them to, for example, freeze or pause a pool.
- Owls monitor Aave and everything around that could indicate an upcoming exploit.
- Rangers receive "alerts" from Owls whenever there is any symptom of danger and execute whatever transactions they are programmed with to protect Aave.

But Aave puts important trust in the Owls and Rangers, no?

Systems like Aave and its permissions' granularity are incredibly powerful in terms of trust management and decentralization for the following reasons:

- Trust can be limited and isolated.

An entity can be able to, let's say, freeze a subset of listed assets, but technically incapable at the same time to affect anyhow else the system.

- Actions can be purely protective in nature.

For example, freezing disables new deposits and borrowings but keeps active repayments, withdrawals, and liquidations, so it is quite non-damaging for users. Even more, whenever an asset gets frozen, it can be unfrozen relatively fast (with some simple changes in the Aave v3 permissions).

- The Aave decentralized governance is a super-admin of any type of sub-permissions.

Any entity on which trust is deposited can get that trust removed at any time the Aave governance system decides to.

With these trust-minimization levers in place, the focus turns into a different perspective: can we maximize the protocol's security by using the smart contracts mechanisms available?

In this case, the answer is quite clear: if there is any non-0 probability of preventing potential exploits on the protocol, it is certainly worth giving certain trust to expert entities (Owls/Rangers) to help with it

.

What would a monitoring + protection flow look like?

Let's assume an attack with the following characteristics:

- Targeting Aave v3 Ethereum.
- Based on some 0-day on the logic, allowing for artificial inflation of the value of a position's collateral, and borrowing of all available liquidity, but only possible in "chunks" of ~$1m each transaction.
- Requiring only a big flash loan from Aave v2.

A potential flow without any protection could be the following:

1. The attacker deploys a smart contract that will be used to perform the attack.
2. Immediately after (or even on the same transaction), he triggers the exploit by calling a function on the contract deployed on 1).
3. Repeat 2) until there is no available liquidity on Aave.

With protection, and even if it would depend quite a lot on the tooling and specifics of the attack, the flow could look:

1. The attacker deploys a smart contract that will be used to perform the attack.
2. An Aave Owl detects from the mempool of Ethereum the contract prepared to start the execution of an attack.
3. An Aave Ranger holding freezing permissions calls freezeReserve()

on all assets of the pool.

1. As borrowing is disabled, the attack will revert.

But will Owls and Rangers be fast enough?

With the techniques of attackers getting more sophisticated every day, this is not easy to answer, but as commented before, analysis of past events shows the following:

- Attacks tend to be less sophisticated than they could be.
- It is quite easy to leave "traces" before an attack happens.
- Monitoring technology is faster than exploit execution.

So again, even if not possible at the moment to have 100% protection given the open nature of the blockchain infrastructure, results show that at least partial protection is possible, which is still a net positive outcome.

So, who will be the Aave Owls and Rangers?

From BGD we have been following the ecosystem of monitoring and protection during the last months, and new participants continuously appear.

If the community believes this system is worth pursuing, we will evaluate more in deep different solutions in the market, obviously prioritizing speed of integration with those that showed historic accuracy.

It is still an open topic (but possible) if Aave Owls should be Rangers at the same time. But we think that as an initial step, it could be the most optimal outcome if a single platform could provide monitoring and execution, with maximum reliability.

Even if that is not the case, even the sole integration of official Owls into Aave DAO procedures could give important benefits and assurances.

Next steps

This is an early project/framework, but quite aligned in connecting Aave with other initiatives like theRisk Stewards. So, first of all, we would like to get feedback from the community regarding this direction.

If there is a positive reaction, we propose the following next steps:

- We will start doing deeper research on all potential candidates for Owls/Rangers

. Given that we are continuously looking into ways of improving the security of Aave, we have already done some initial research about different entities, like the ones presented before in this post.

But we encourage any alternatives/candidates to both comment on this post (preferably without entering into contests with others) and contact us on Twitter or hi@bgdlabs.com, to check out the platforms.

- Define grants model for external entities

. The Aave DAO already runs a successful grants program which usually is the initial step for new contributors. As Owls/Rangers will most probably be a similar case, propose a standardized grants system for their service, being through the Aave Grants DAO or apart.

- Once we define additional technical specifications, Snapshot votes to approve the initial sets of Owls and Rangers, together with the initial Aave v3 roles to give to Rangers if applicable

. BGD will propose this initial set, taking into account both the underlying infrastructure and how well it fits into the protocol’s requirements and limitations.

- On-chain proposal for granting roles to Rangers

.