Exploring AI’s Role in Smart Contract Security

Speakers: Alice , Daky

ONESAVIE

Who am I

- Product Development Lead @OneSavie Lab
- Member @DeFiHackLabs
- Focus on AI & Web3 Development

https://x.com/ga013077

https://medium.com/@ga013077

ONESAVIE

Who am I

- Security Research Engineer @OneSavie Lab
- OP and White Hat @DeFiHackLabs
- Audit Team member @TaiChi Audit Group
- Focus on Web3 Security Research

@AliceHsu_kou

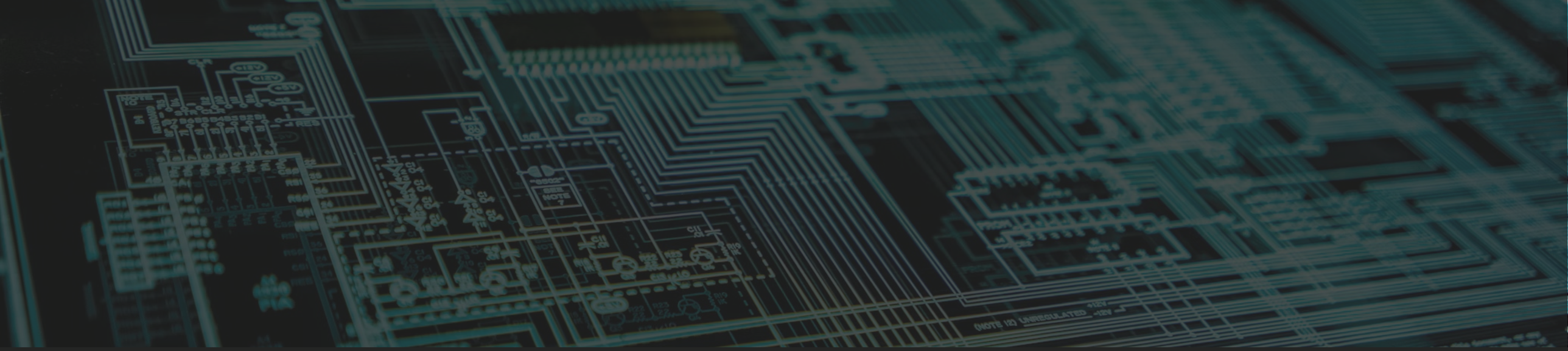
@Alice_Hsu

DEFIHACKLABS
Let's make web3 more secure!

Agenda

- Background
- About Bastet
  - Introduction
  - Methodology
- Conclusion

ONESAVIE

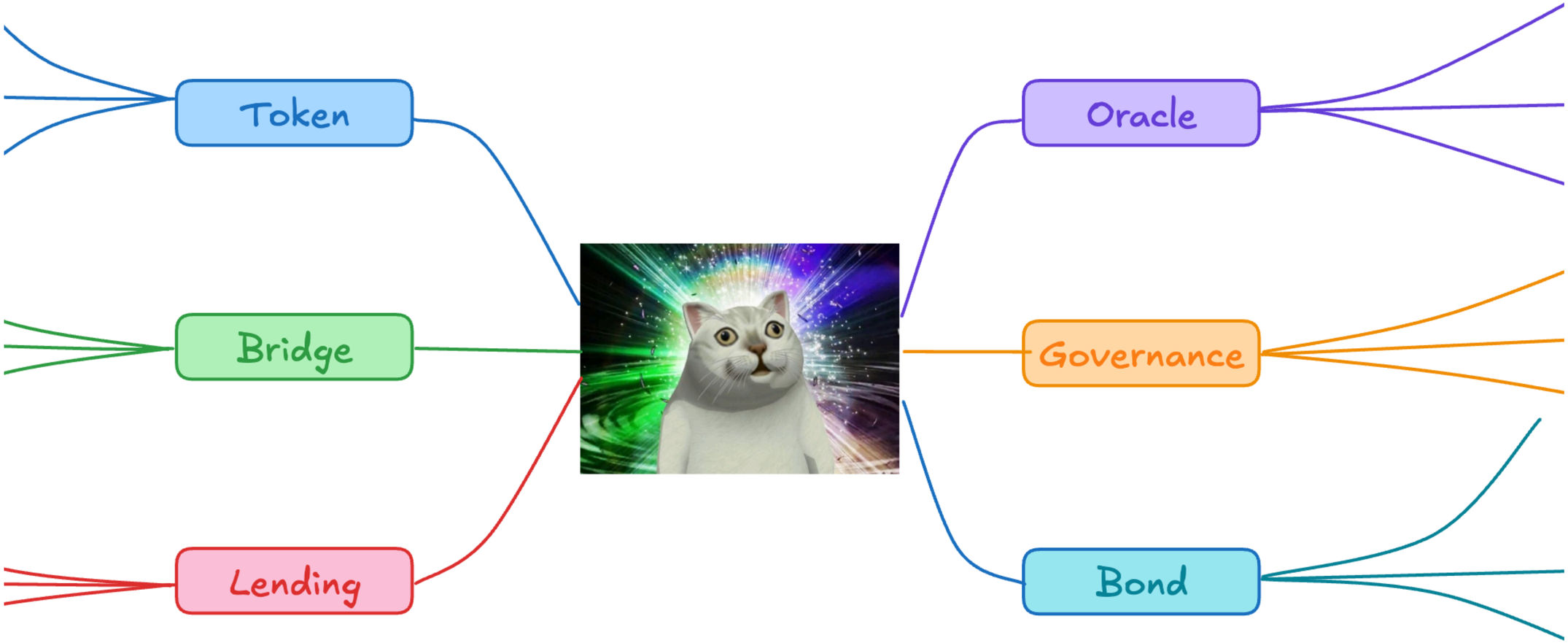


Background

Motivation: Driving Industry Security

- DeFi applications are becoming more diverse, with increasing security awareness and service demand. Together, we aim to promote industry development through the power of the community.
- Expecting to optimize the process from test to audit.

ONESAVIE



ONESAVIE

322 findings found

Search for keywords (title, content)

rounding error

Report Tag

ALL

Source

ALL

Impact

HIGH, MEDIUM

748 findings found

Search for keywords (title, conten

slippage

Report Tag

ALL

Source

ALL

Impact

HIGH, MEDIUM

982 findings found

Search for keywords (title, content)

first deposit

Report Tag

ALL

Source

ALL

Impact

HIGH, MEDIUM

ONESAVIE

[M-01] Missing slippage protection in `AerodromeDexter.sol`

`swapExactTokensForTokens()`

Submitted by alix40, *also found by* 0xKann, 0xlucky, 0xpetern, 0xvd, ABAIKUNANBAEV, Abhan, air_0x, zanderbyte, bharg4v, DharkArtz, dic0de, Hajime, hals, inh3l, John_Femi, kodyvim, lightoasis, Matin, mgf15, MSaptarshi, Mushow, newspacexyz, NexusAudits, PolarizedLight, Rhaydden, Shinobi, smbv-1923, Sparrow, tpiliposian, Tumelo_Crypto, udo, vesko210, waydou, *and* y51r

Issue H-1: Precision differences when calculating userCollateralRatioMantissa causes major issues for some token pairs

Source: #122

Found by

0x52, Bauer, GimelSec, TrungOre, __141345__, ast3ros, bin2chen, ctf_sec, gogo, joestakey, peanuts, usmannk

Issue H-1: Lack of slippage protection leads to loss of protocol funds

Source: #66

Found by

0x6a70, 0x73696d616f, 0xjoi, 0xnbvc, 4b, 4gontuk, Abhan1041, MohammedRizwan, Pheonix, StraawHaat, alphacipher, boringslav, ivanonchain, sakshamguruji, vinica_boy

[H-17] First depositor bug on unmodified Compound fork

Submitted by thekmj, *also found by* nirlin, Ruhum, Qeew, 33audits *and* fs0c.

ONESAVIE

[H-08] Vault.sol is not EIP-4626 compliant

Submitted by 0x52, also found by Bahurum, Jeiwan, Lambda, PwnPatrol, and thebensams

Issue M-3: Contracts of the codebase will not strictly compliant with the ERC-1504.

Source: #155

The protocol has acknowledged this issue.

Found by

0xnbvc, Atharv, AuditorPraise, HackTrace, Kirkeelee, dany.armstrong90, isagixyz, pkqs90

Issue M-6: `previewRedeem` and `redeem` functions deviate from the ERC4626 specification

Source: #577

Found by

0x73696d616f, 0xSurena, BPZ, BTK, Flora, Kalyan-Singh, Nadin, bin2chen, enfrasico, shaka, talfao, xiaoming90

Issue M-5: The `SuperPool` vault is not strictly ERC4626 compliant as it should be

Source: #110

Found by

000000, 0xAadi, 4gontuk, A2-security, Atharv, EgisSecurity, Flare, Kalogerone, Nihavent, Obsidian, Ryonen, S3v3ru5, dany.armstrong90, h2134, hash, iamandreiski, pseudoArtist, xtesias

[M-01] PrincipalToken is not ERC-5095 compliant

Submitted by jnforja, also found by sl1, dimulski, wangxx2026 (1, 2), JohnSmith, 0xLogos, 14si2o_Flint, erosjohn, Aymen0909, Limbooo, Giorgio (1, 2), smaul, ZanyBonzy, Oxhacksmithh, Brenzee, btk, 0xDemon, lsaudit (1, 2), mrudenko, memforvik, Franklin, Shubham, and nmirchev8

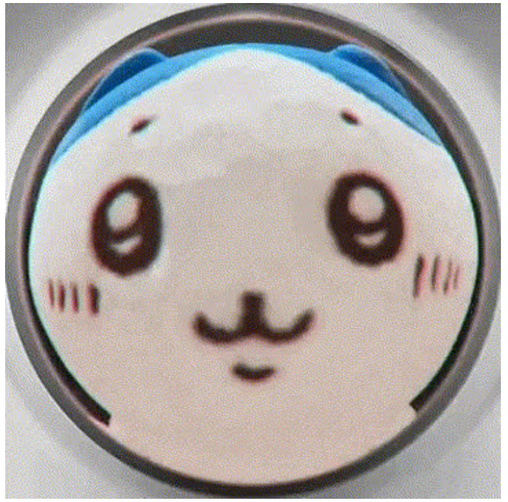
Issue M-7: `ERC721Bridgable` and `ERC1155Bridgable` are not EIP-2981 compliant, and fail to correctly collect or attribute royalties to artists

Source: #214

The protocol has acknowledged this issue.

Found by

Ruhum, h2134, kuprum, novaman33, rndquu, zzykxx



ONESAVIE

21 Mar 4:00 AM - 26 Mar 4:00 AM GMT+8

THORWallet Mitigation Review

An all-in-one DeFi solution that seamlessly combines cross-chain trading,
powerful multichain multisignature solution—all in a single, intuitive platfo

EVM Solidity

Ends in 4 days

20 Mar 4:00 AM - 10 Apr 4:00 AM GMT+8

Starknet Perpetual

Shaping the future with scale and integrity.

StarkNet Cairo

Ends in 19 days

18 Mar 4:00 AM - 25 Mar 4:00 AM GMT+8

Nudge.xyz

Reallocating assets, liquidity, and activity onchain

EVM Solidity

Ends in 3 days

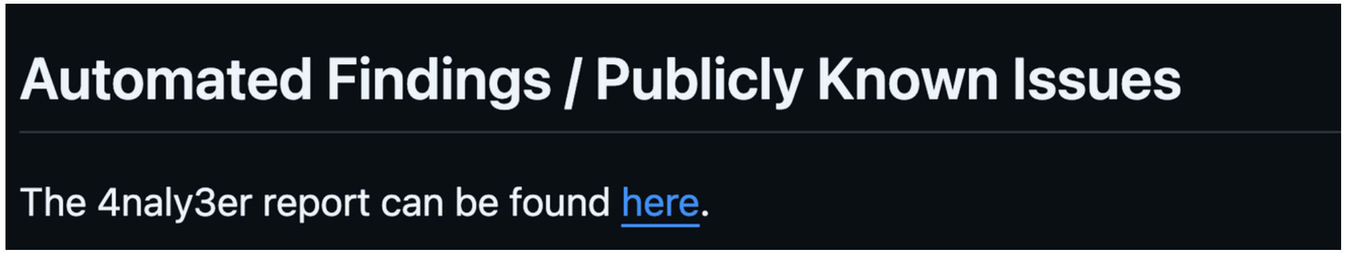
ONESAVIE

Test -> Audit

ONESAVIE

Common Smart Contract Analysis Tool

- Mostly static analysis tools
- Rules are maintained using **AST** or **Regex**



ONESAVIE

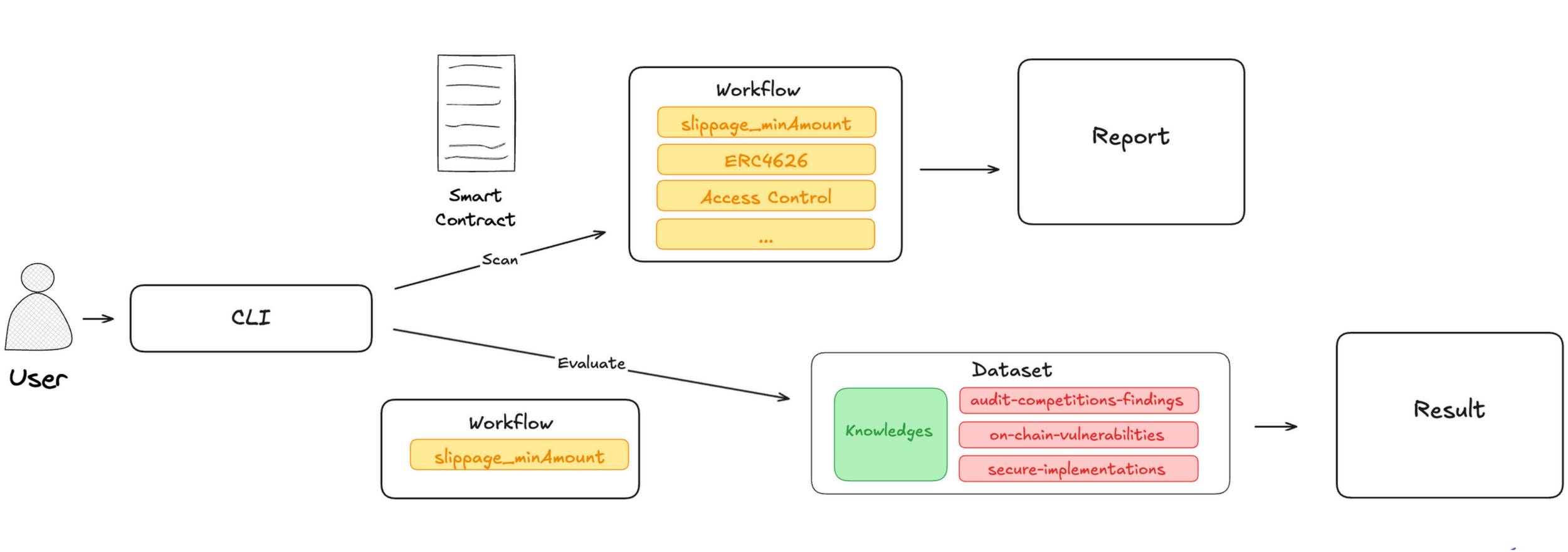
About Bastet

Introduction

- This open source project Bastet focus on two parts:
  - Vulnerability dataset + AI Audit benchmark tool
  - AI Vulnerability identification process

ONESAVIE

User Scenario



ONESAVIE

How to Select Targets ?

- The vulnerability selection focuses on **common issues** that are **hard to maintain** with static analyzers.
- Designing processes to enhance AI's accuracy in detecting vulnerabilities

ONESAVIE

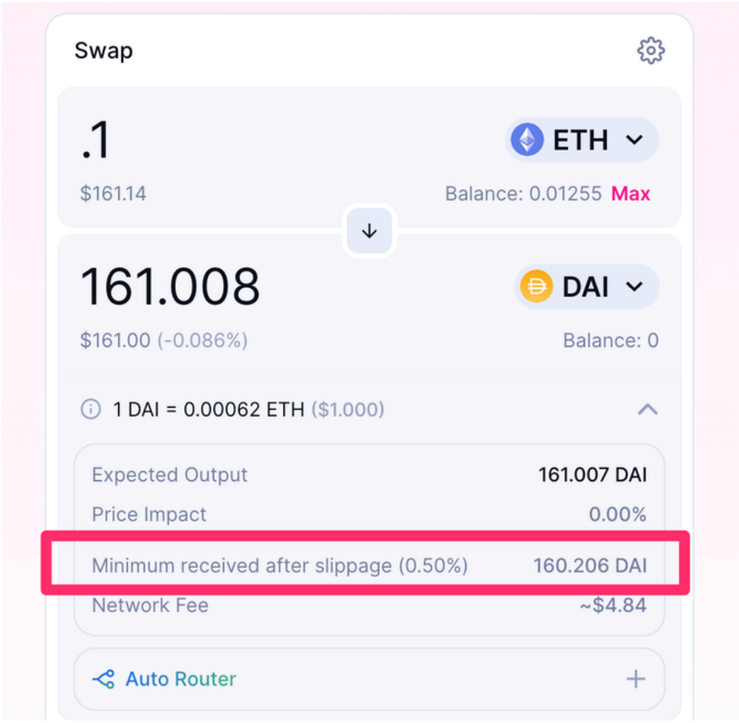
Methodology

- What we need to prepare
  - Knowledges
  - Prompt Engineering Technique
  - Validation

ONESAVIE

Knowledges - Slippage

- Slippage refers to the difference between the **expected price** of a transaction and the **actual price** at which the transaction is executed.



ONESAVIE

Knowledges - Improper slippage settings

- If there is no slippage check, MEV can monitor transactions and perform a **sandwich attack** before and after the transaction, profiting from the price fluctuations.

ONESAVIE

How to perform detection using static analyzers?

1. Use **findAll** to search for specific functions like **addLiquidity**.
2. Find the location of parameters like **minAmountOut** in the function.
3. Write corresponding conditional checks for each type of improper slippage setting scenario.

ONESAVIE

What are the challenges of doing this?

- There are variations in the function names for swaps and slippage parameters across different protocols.

- The same issues cannot be detected in different programming languages.

ONESAVIE

What method are we using?

- Example, human brain learns to recognize cats.
- Knowledge Base:
  - A cat's appearance typically features **soft fur** and **a sleek, agile body**.
  - They have **four legs**, with **sharp claws** that are perfect for climbing and hunting.
  - Cats' **ears are upright**, and their **noses are small and sensitive**.
  - A cat's **tail is long and flexible**, often used to maintain balance.

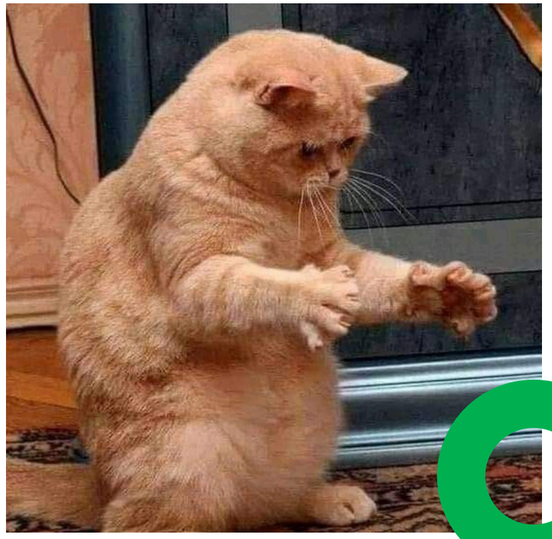
ONESAVIE

AI Might Think ...

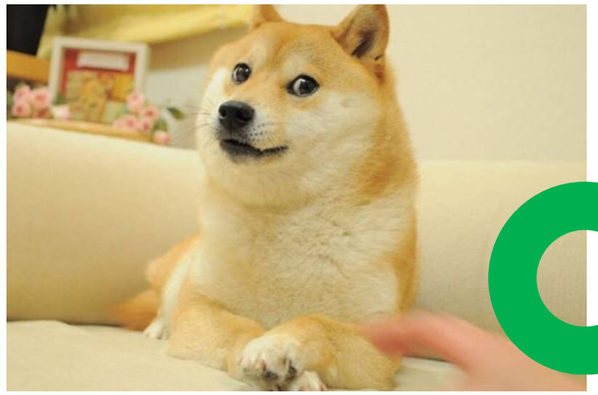
1

2

3



4



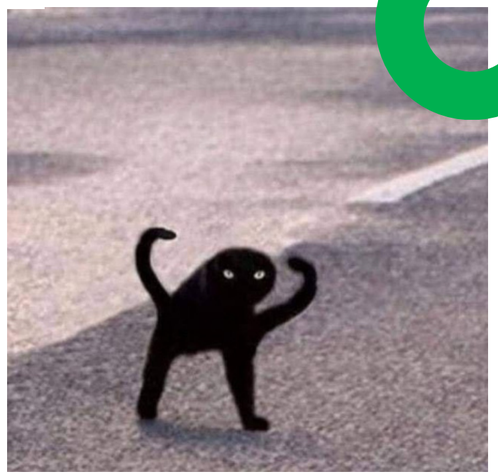
5

6

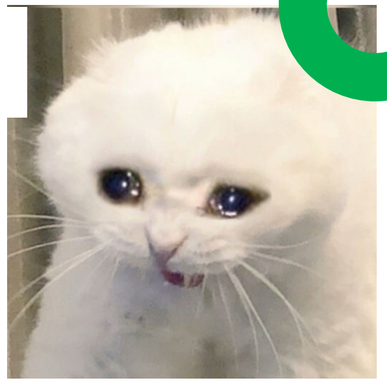
ONESAVIE

The Answer

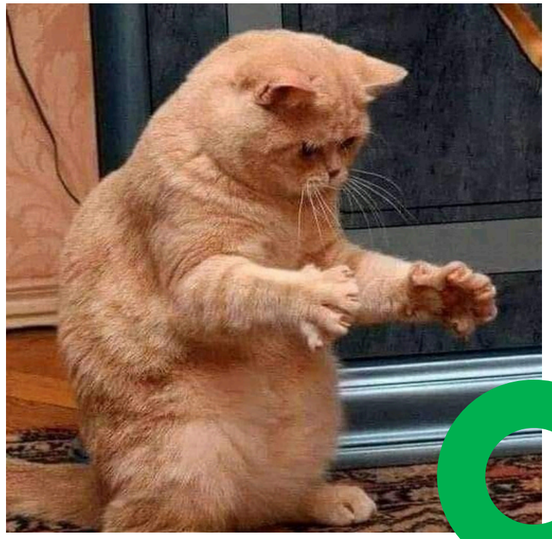
1



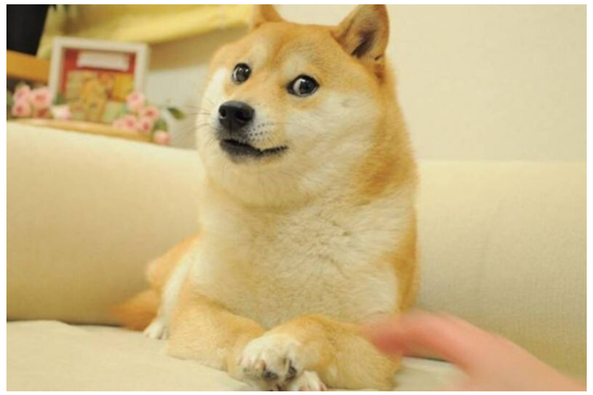
2



3



4



5

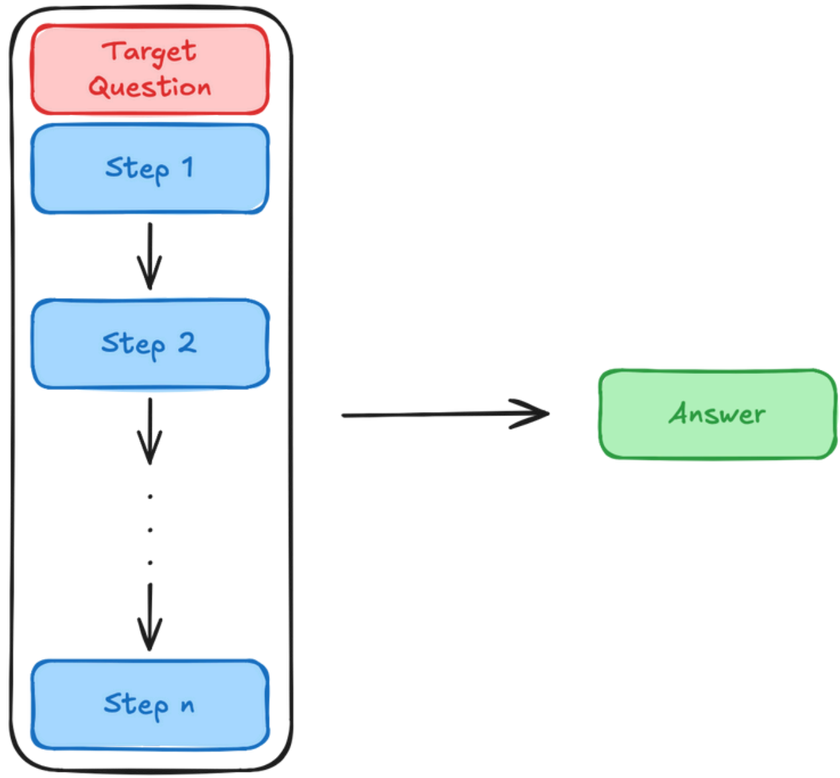
6

ONESAVIE

Prompt Engineering Technique – CoT

- By using the CoT method, the reasoning ability of large language models can be improved.

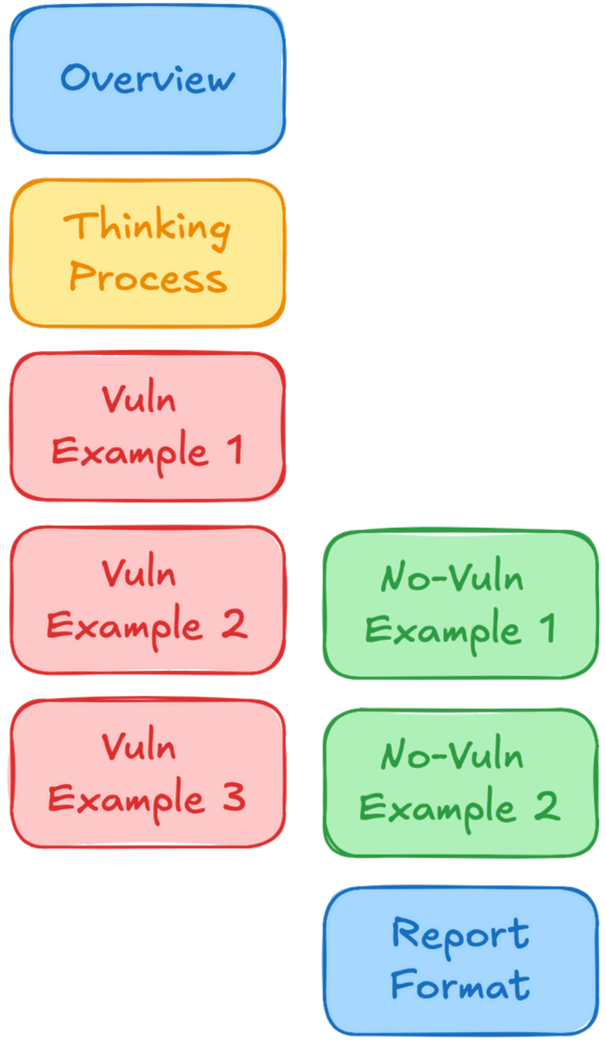
- By breaking down the reasoning process, we help the model think step-by-step, with each step closely tied to the root causes of vulnerabilities.



ONESAVIE

Prompt Engineering Technique – CoT

- The LLM analyzes each function according to the specified steps.
- The LLM details the thought process in the examples.
- Generate results based on the report format.



ONESAVIE

Bastet's Output

- Summary
- Severity
- Vulnerability Details
  - Description
  - Code Snippet
- Recommendation

Bastet Security Report

Missing Slippage Protection for Token Swap

Severity: High

Vulnerability Details

- **File Name:** `IbbtcVaultZap.sol`
- **Function Name:** `deposit`
- **Description:**
  The function uses `add_liquidity` from the `CURVE_IBBTC_DEPOSIT_ZAP` contract without specifying a minimum amount of LP tokens to receive.
  This exposes the user to potential slippage, which could result in receiving fewer tokens than expected due to price fluctuations.

Code Snippet

```
uint256 vaultDepositAmount = ICurveZap(CURVE_IBBTC_DEPOSIT_ZAP).add_liquidity(
    CURVE_IBBTC_METAPOOL,
    depositAmounts,
    0,
    address(this)
);
```

Recommendation

Introduce a parameter for the minimum amount of LP tokens to receive (e.g., _minAmountOut) and use this parameter in the add_liquidity function to ensure slippage protection.

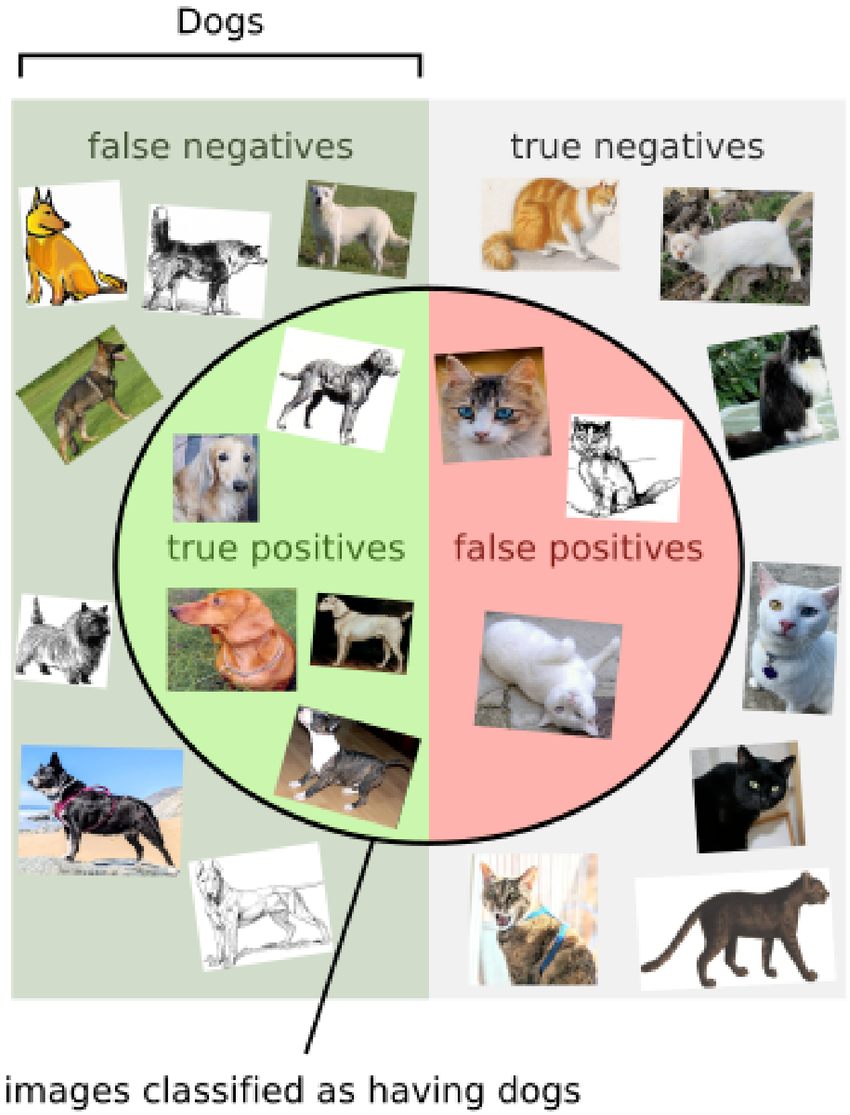
ONESAVIE

Test Results - example

- Accuracy
  - How to calculate?

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}}$$

$$= \frac{TP + TN}{TP + TN + FP + FN}$$



ONESAVIE

Test Results

- The evaluation result of minAmountOut workflow is
  - True Positive: 12
  - True Negative: 27
  - False Positive: 2
  - False Negative: 5

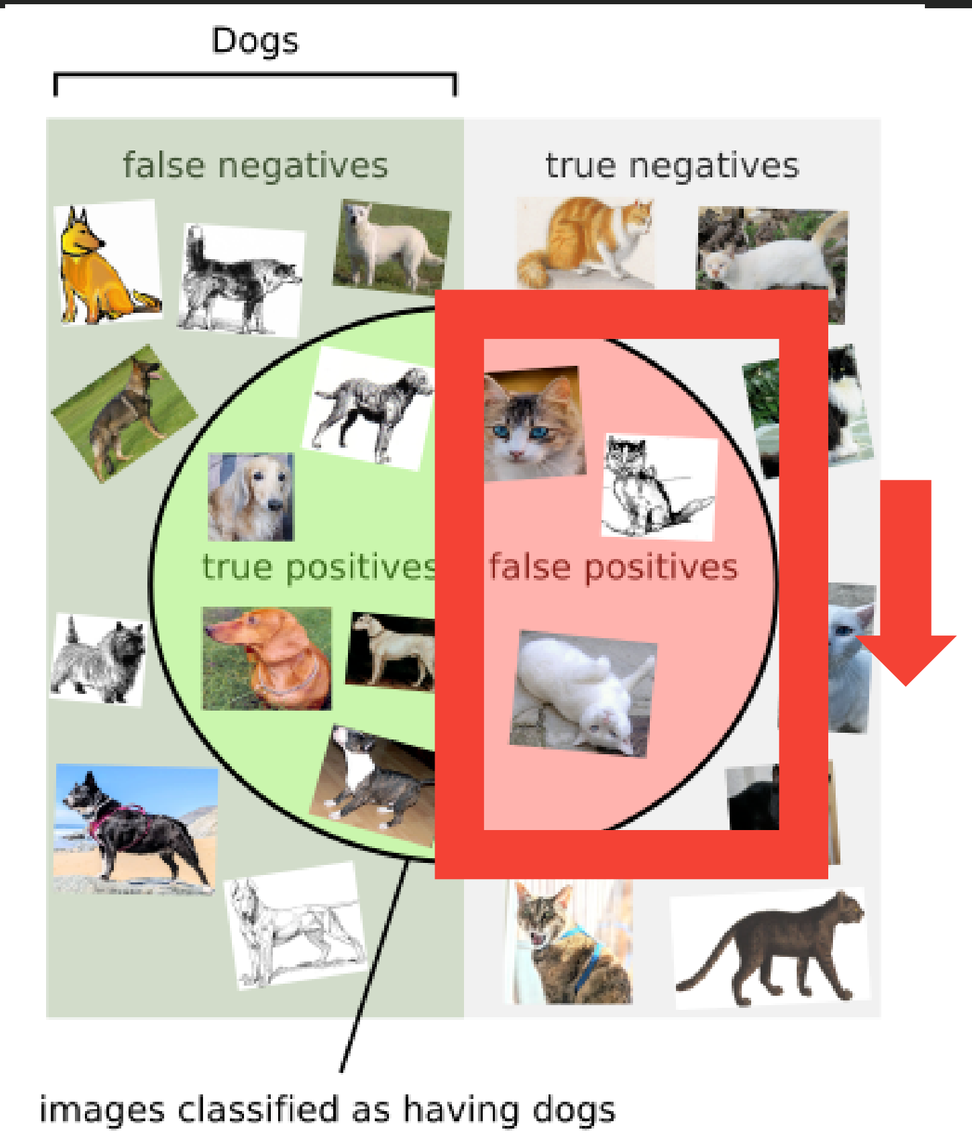
Metric	Value
True Positive	12
True Negative	27
False Positive	2
False Negative	5

- Accuracy: (12+27)/46 = 84.7%

ONESAVIE

Test Results

- High Accuracy is good
- Less False Positive (FP) is good
- Save Developer and Researcher's Time



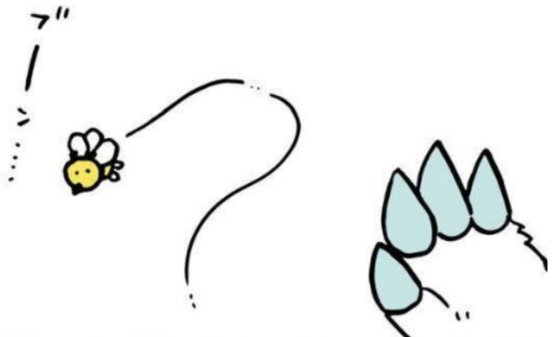
Can Bastet One Day Replace Human Researchers?

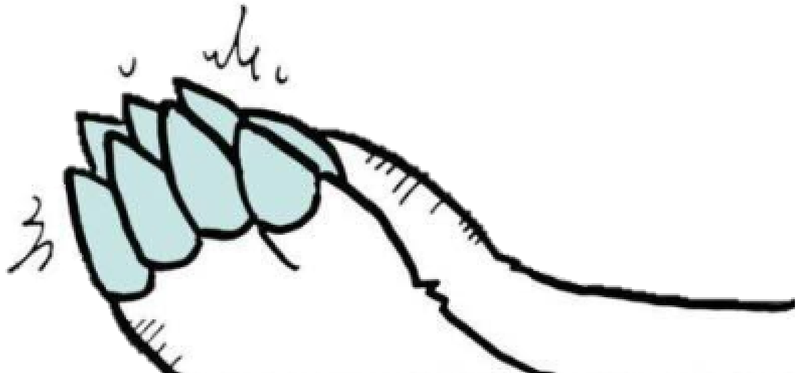
- The goal of Bastet is to assist researchers to focus on higher level issues such as systemic risks, logic errors, integration issues, and more.
- As a tool, it will undoubtedly become stronger over time.

ONESAVIE

What Does Bastet Need To Become Stronger?

- Data
- Knowledge Base
- Methodology



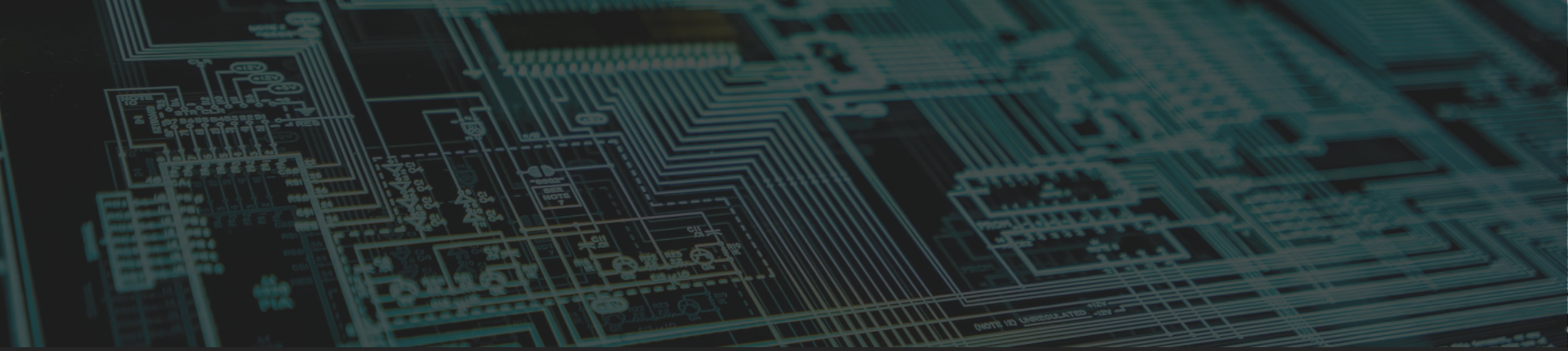


ONESAVIE

What if the vulnerabilities that are not in the knowledge base?

- Bastet is highly skilled at finding details and implementation errors in code.
- We are also working closely with the community.
- Bastet has already identified **valid** findings in every audit, including **high** and **medium** severity issues.

ONESAVIE



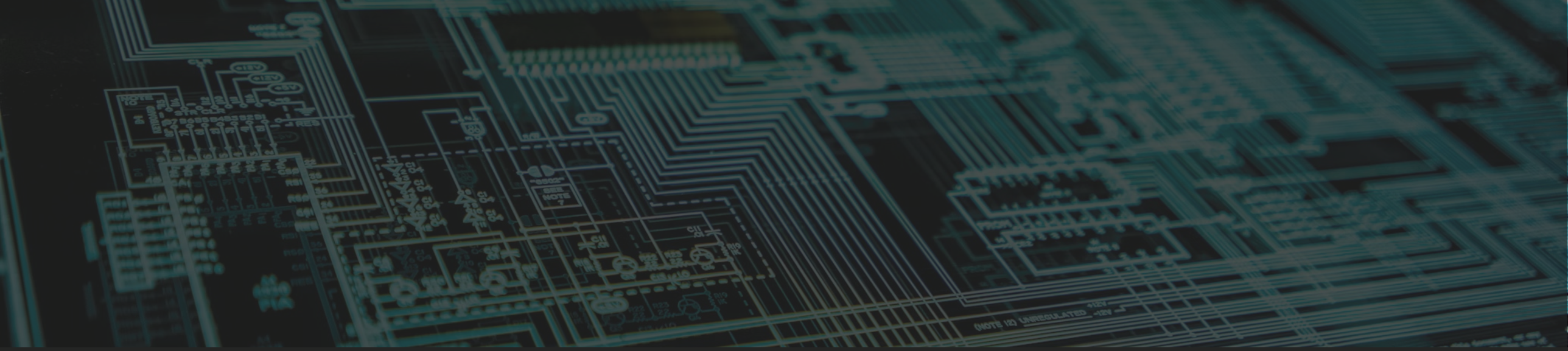
Conclusion

Takeaway

Bastet features:

- Dataset + Evaluation
- AI workflows

ONESAVIE



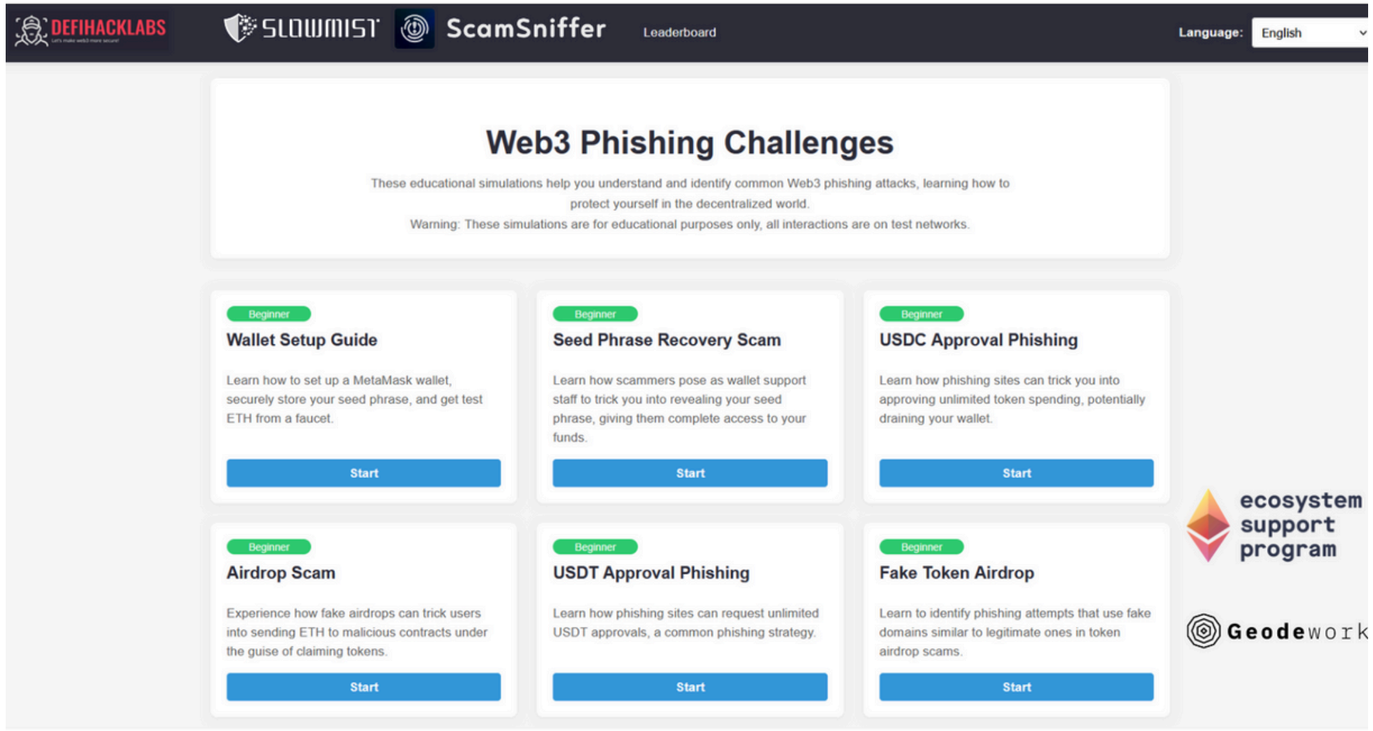
DeFiHackLabs Time

- PhishQuest: Interactive Web3 Anti-Phishing Training Platform
- Founding Contributors: DeFiHackLabs、SlowMist、ScamSniffer
- Supported by: Ethereum Foundation Grant
- Current Status: Prototype
- Features:

  Gamification Design

  30+ Challenges

  Community Collaboration
- Release: Before July



Special Thanks

We are inspired by the spirit of community, open source,

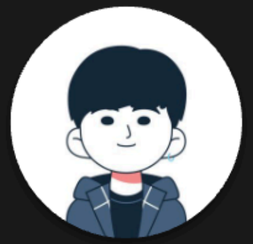
and selflessness, and hope to gather collective intelligence.

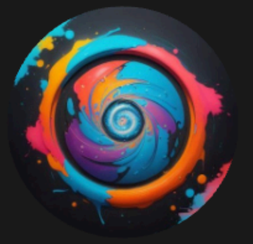
Together Strong!!!

ONESAVIE

DEFIHACKLABS

Let's make web3 more secure!

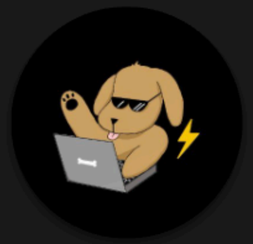



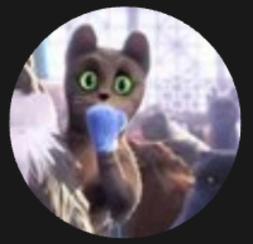










DEFIHACKLABS

Let's make web3 more secure!