Разработка инструмента для исследования множества контрактов сети Ethereum

Алексей Прошутинский
группа 444
руководитель: ст.преп. А. Р. Ханов
рецензент: к.т.н А. А.Воробьева

24 мая 2019 г.

СПбГУ

ВВЕДЕНИЕ

- **Предмет исследования:** рабочая сеть Ethereum и код контрактов
- **Проблема:**
  - База данных занимает десятки гигабайт
  - Новые контракты появляются с каждым блоком
  - Не у всех контрактов есть исходный код
  - Поиск похожих контрактов
- **Решение:**
  - База данных с индексом по контрактам

АКТУАЛЬНОСТЬ

- Большое количество контрактов
- Анализаторы работают очень медленно
- Нет актуальной открытой статистики по уязвимостям

Известные уязвимости

- Arithmetic - The DAO, TokenOverflow
- Access control - Parity Wallet
- Reentrancy - The DAO

REENTRANCY

```
1  contract Victim {
2      mapping(address => uint) userbalances;
3
4      function withdraw() {
5          if (userbalances[msg.sender] > 0) {
6              if (msg.sender.call.value(userbalances[msg.sender])()) {
7                  userbalances[msg.sender] = 0;
8              }
9          }
10     }
11
12     function() {
13         userbalances[msg.sender] += msg.value;
14     }
15 }
```

REENTRANCY

```
1  contract Attacker {
2      Victim v;
3
4      function Attacker(address dest) {
5          v = Victim(dest);
6      }
7
8      function attack() {
9          v.call.value(msg.value)();
10         v.withdraw();
11     }
12
13     function() {
14         if (msg.gas > 100000) {
15             v.withdraw();
16         }
17     }
18 }
```

ПОСТАНОВКА ЗАДАЧИ

Цель: Поиск уязвимого множества контрактов сети Ethereum

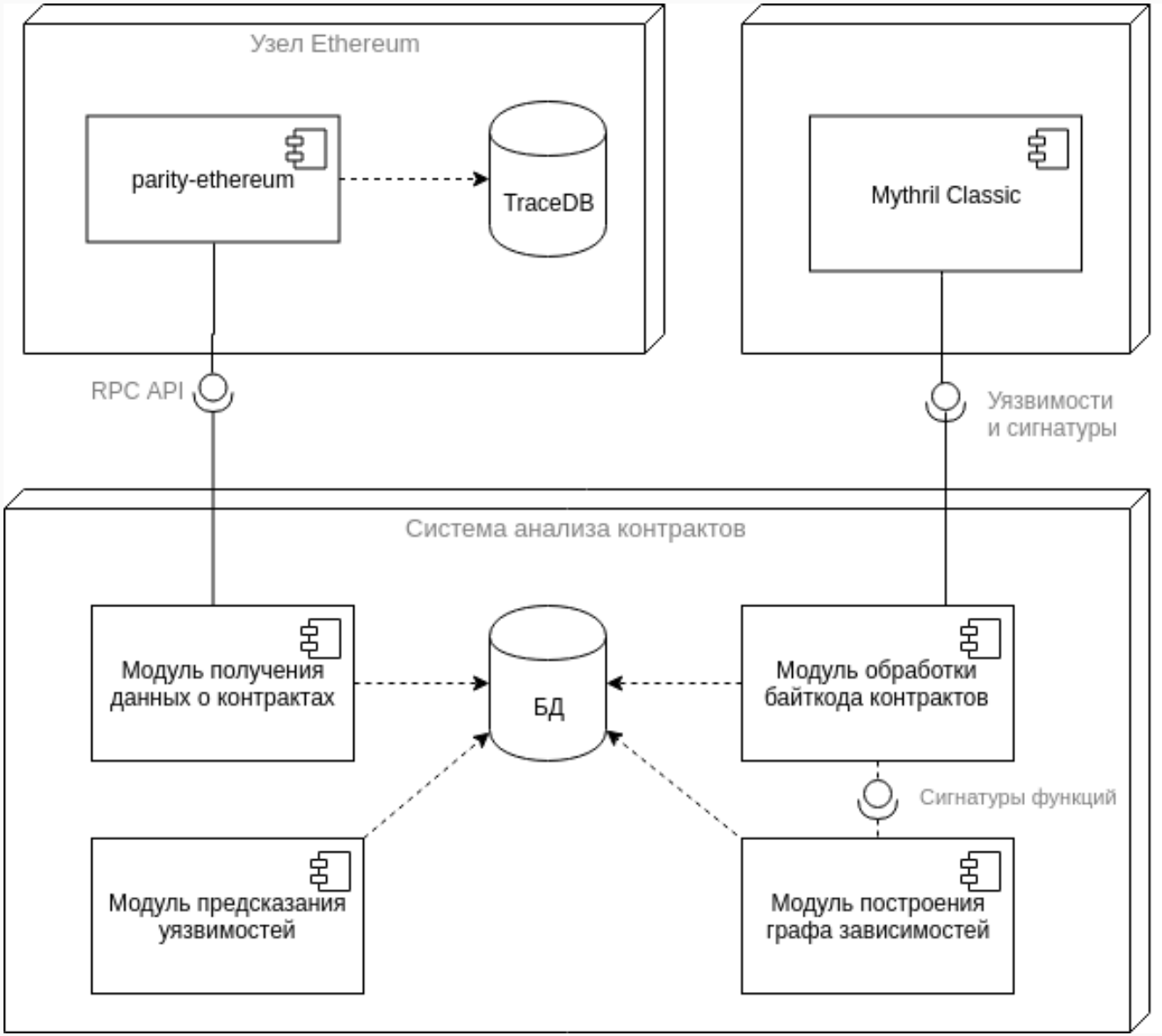
Задачи:

- Разработка модуля для получения информации о контрактах в реальном времени
- Выделение уникальных контрактов
- Получение статистики по контрактам в реальном времени
- Поиск похожих контрактов
- Предсказание уязвимостей для новых контрактов

АНАЛОГИ

- etherscan.io
- Ethereum BigQuery
- blockchair.com

РЕШЕНИЕ


Узел Ethereum
parity-ethereum
TraceDB
Mythril Classic
RPC API
Уязвимости
и сигнатуры
Система анализа контрактов
Модуль получения
данных о контрактах
БД
Модуль обработки
байткода контрактов
Сигнатуры функций
Модуль предсказания
уязвимостей
Модуль построения
графа зависимостей

РЕШЕНИЕ

- Нахождение в блоках транзакций создания контрактов
- Проверка уникальности контракта по хеш-сумме от байткода
- Дизассемблирование и определение сигнатур функций
- Построение графа зависимостей между контрактами
- Вывод контрактов, связанных с известными уязвимыми
- Предсказание уязвимостей на основе обученной модели

БАЗА ДАННЫХ

- Адрес контракта
- Исходный код
- Блок создания
- Сигнатуры функций
- Статистика по похожим контрактам
- Битовый вектор уязвимостей

ПРЕДСКАЗАНИЕ УЯЗВИМОСТЕЙ

- Модель – персептрон
- Исходные данные – кортежи сигнатур функций 93 тысяч контрактов
- Контрольные данные – кортежи уязвимостей 80 тысяч контрактов, полученные с помощью Mythril

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Уязвимость	AUC	ACC
SWC-101	0.81	0.79
SWC-104	0.91	0.97
SWC-105	0.91	0.99
SWC-106	0.94	0.99
SWC-107	0.91	0.87
SWC-110	0.86	0.85
SWC-111	0.89	0.99
SWC-112	0.95	0.99
SWC-113	0.90	0.98
SWC-116	0.89	0.99
SWC-120	0.94	0.99

$$ACC = \frac{TP+TN}{P+N}$$
$$AUC = \int_{x=0}^{1} TPR(FPR^{-1}(x))dx$$
$$TPR = \frac{TP}{P}$$
$$FPR = \frac{TN}{N}$$

РЕЗУЛЬТАТЫ

- Реализован модуль получения информации о контрактах в реальном времени
- Из 13 млн контрактов выделено 200 тыс уникальных
- Выделены сигнатуры функций и собрана статистика по ним
- Построен граф зависимостей
- Обучена модель предсказания уязвимостей
- Доклад на СПИСОК 2019
- github.com/justprosh/ethereum_contract_analyzer