FortaKnight PRD Document

Team Lead: Nicholas Brown

Members: John Lin, Andy Wu, Khalid Mihlar, Alejandro Rojas Rodriguez

Company: Forta Foundation

Introduction:

Web 3.0 and decentralized finance (DeFi) implement cutting-edge blockchain technologies to provide users with financial instruments that are not reliant on centralized financial institutions through the use of smart contracts on a blockchain network. This technology has the potential to revolutionize the financial experience of the user but also presents new, unprecedented challenges. The pseudonymity that blockchain technologies offer is a double-edged sword as it allows hackers to steal large sums of money with minimal traceability. According to a report by Elliptic, DeFi users lost an estimated $10.5 billion to theft in 2021. These hacks are performed primarily by deploying an attacker smart contract that exhibits adversarial behavior on a blockchain network. Our project aims to reinforce the security that the Forta network provides by developing a bot that is capable of detecting attacker smart contracts through both static and dynamic analysis methods. The four attack stages consist of funding, preparation, exploitation, and money laundering. Our bot will seek to detect attacks before the exploitation stage because funds are virtually impossible to recover after they have been stolen due to the nature of the blockchain.

On the bright side, Forta has created a system that allows people to create bots to monitor blockchain activity for malicious activity. Unlike attacks on the Web 2 space, the entire transaction history of every attack is publically available. This means we can

leverage data about previous attacks to develop detection strategies for future attacks. The public nature of the blockchain also means that Forta bots can monitor all activity on the supported blockchain network for attacks. This means that any user can use a Forta bot to monitor any contract on the blockchain. This is particularly useful for people using smart contracts in production because they can use Forta bots to monitor activity related to their smart contracts to detect malicious activity as soon as possible and mitigate the damage before it is too late.

Our Goal:

We are researching pre-existing malware detection strategies used in other areas of computing and are planning to apply these insights to develop a detection strategy for Web 3.0. We will look for high-quality heuristics that can detect malicious activity. As described previously, it is important to have a bot that can detect attackers as quickly as possible because there is little value in being able to detect attacks once attackers have already stolen the assets. It is also important to avoid having a high false positive rate since most transactions on the blockchain are not malicious. A high false positive rate means that the bot will yield a large number of false alerts, causing users to have less trust in our alerts and making real, critical alerts more difficult to spot. Our goal is to strike a balance between speed and accuracy to ensure that our bot is as useful as possible.

We also aim to leverage machine learning techniques to detect suspicious activity. Since we have access to an in-depth dataset of past attacker smart contracts, we will be able to use this data to train a machine-learning model that can detect malicious activity. We will study this data to find useful features that can indicate a

malicious smart contract and use these features in a model. If this approach is more effective than using heuristics in terms of speed and accuracy, then we will use this instead of the heuristics. We can potentially use a combination of both approaches to get better accuracy for our alerts.

Implementation:

We will deploy our bot on the Forta Network which provides a simple and convenient way for users to subscribe to and receive alerts. This will allow users to easily examine activity on the blockchain and keep record of our bot through alert logs. The Forta Network allows us to focus on our detection strategy instead of having to worry about how we will examine the blockchain activity and communicate with our users. We also will be actively updating our bot with new changes, allowing the bot to become more effective as we develop it. We will be using ML models to help increase efficiency and decrease false positive rates.

Assumptions:

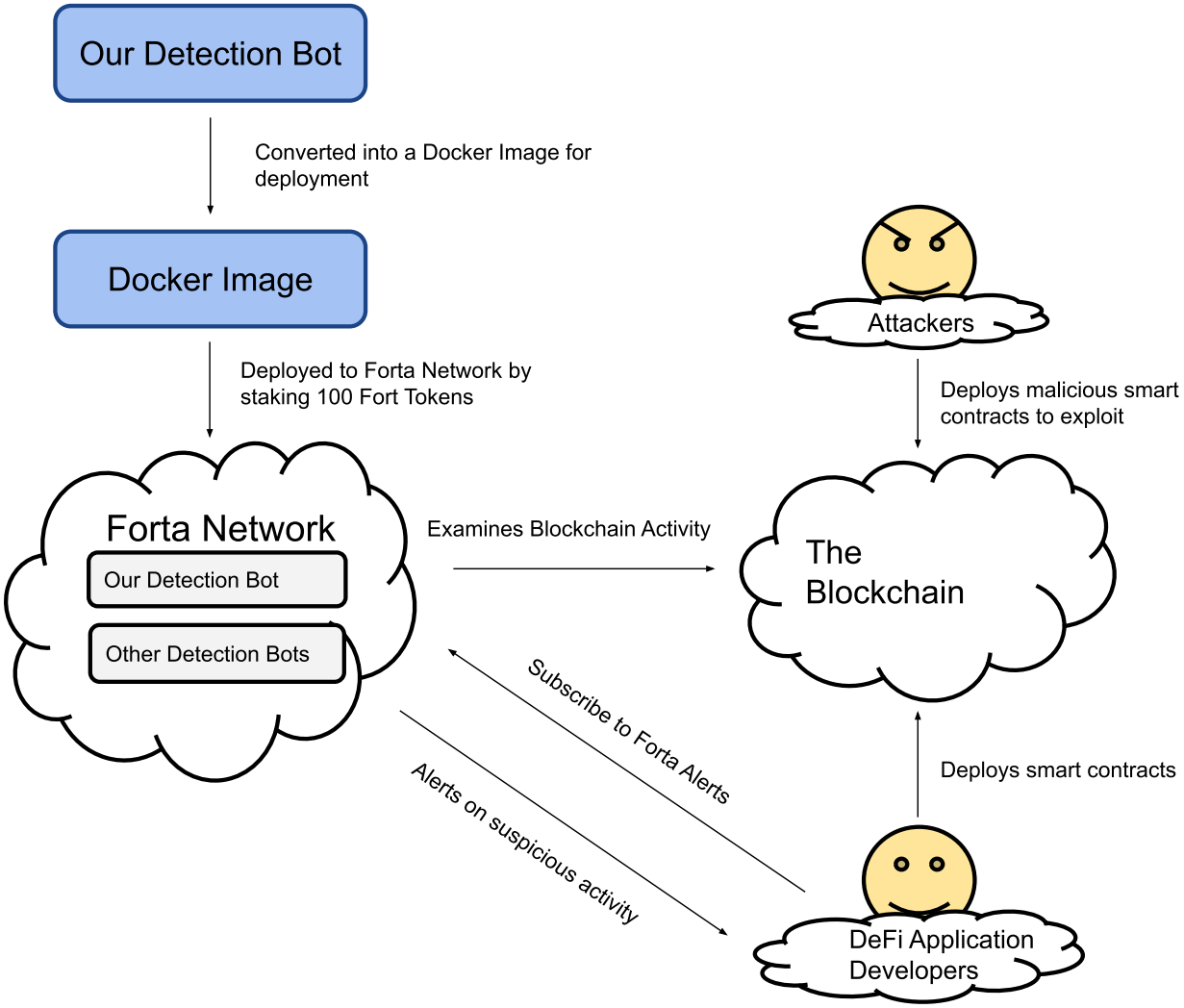
Since this is an emerging field, there is little documentation about prior techniques to create attacker contract detection bots, which makes our task more difficult. This means that we will have to identify patterns that attacker bots have in common ourselves, or by utilizing machine learning techniques to detect attacker smart contracts.

Our use of machine learning hinges on the assumption that patterns in previous malicious attacks will be useful in predicting future attacks. The Web 3 space is constantly changing, so it is likely that future attacks will look completely different from any past attack. It is also possible that attacks in the past are so different from each

other that no useful patterns will emerge from our analysis. We are confident that machine learning will be a useful technique for detecting attacks, and if attacks change drastically over time it will be relatively simple to retrain the model on newer attacks to improve its accuracy. As the Web 3 space becomes more stable in the future, we expect concrete attack patterns to emerge which will allow us to more confidently detect certain classes of attacks.

System Architecture Overview

High-Level Diagram



User Interactions and Design:

This detection bot will be deployed on the Forta Network, so Forta provides the necessary tools for user interaction by default. Forta has a streamlined bot subscription system that allows users to easily access the bots and view findings about their smart contracts. Therefore, a minimal implementation of user interactions and design will be needed for this project. Users can navigate through the Forta website and find the bot search feature. From the search bar, they can either search by bot name or developer ID in order to find and subscribe to our detection bot.

However, Forta requires a few user features that we must implement. The bot will have fairly detailed documentation that describes which blockchain network the bot runs on, what alerts it provides, and how to properly install/develop the bot. The documentation should cater to both beginner blockchain users (with high-level bot descriptions and instructions) as well as experienced users (with developmental details). We will do this by having an overview that gives a high-level description of how the bot works that will be easy to understand for any user. We will also provide specific details about our implementation to cater to experienced users so that they can understand how our bot works. This will help users gain more trust in our bot. Additionally, the bot will record alerts that will appear under our Forta bot page, providing information such as transaction address, attack severity, and output alert. The data will be compiled into graphs such as the number of daily alerts and alert severity. Lastly, the source code will be provided as a link on the Forta bot page.

Requirements:

1. As a user, I can subscribe to get an alert before an unauthorized smart contract exploits my contract protocols so that I can allow my system to deploy necessary resources to stop the exploitation before any money is lost.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/1

   - Scenario 1: The bot detects an unauthorized smart contract before the exploitation phase and alerts the user system to deploy resources to stop exploitation by the malicious smart contract.
   - Scenario 2: The bot detects an unauthorized smart contract, but fails to alert the user system before the exploitation phase. The bot will deploy a warning message stating that an attack has, or may be currently occurring.

2. As a user, I can access the bot detection logs so that I can check on all attacks that have occurred on my system and check bot activity if the system doesn't correctly detect an attack.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/2

   - Scenario 1: The user requests an activity log, and the bot relays all detected attacks in chronological order.
   - Scenario 2: The user requests an activity log, and the bot relays all timely detected attacks, delayed detections, and money lost (if any)

3. As a user, I can search for a specific attack that has occurred on my system by clicking on the navbar so that I can instantly look up the attack instead of looking through the whole log.

GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/3

4. As a user, I can filter detection bot logs by day, week, month, or year so that I can limit the amount of detection bot logs I see because I am interested in detection bot logs that happen on a specific date range.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/4

5. As a user, I can download the bot detection logs so that I can have a copy of the bot detection logs.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/5

6. As a user, I will be able to filter the types of alerts I subscribe to by severity (high, medium, low) so that I can focus on alerts of a specific severity.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/6

7. As a user, I can click on the “Source Code’ button so that I can see the bot’s source code and verify that the code is not malicious.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/7

8. As a user, I can click on the GitHub Repository link to learn more information about the bot and how it was developed

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/8

9. As a user, I can look at the bot information so that I can know the bot’s ID, the networks the bot has scanned, and when the bot was created or updated.

   GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/9

10. As a user, I can verify that the alert system I’ve subscribed to is working correctly to ensure that I don’t miss a potential alert.

    GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/10

11. As a user, I will be able to easily filter through the logs so I can access the information that I requested

    GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/11
12. As a user, I can access other bot data to see how it compares to the bot that applies to my account

    GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/12
13. As a user, I can review the bot's alerting history so that I am aware of any potential changes in behavior/performance through changes in the bot's alerting patterns.

    GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/13
14. As a user, I can see the developer's wallet address so that I can see how credible they are as a developer because sometimes wallet addresses are linked to crypto/NFT scams, such as "rug pull" scams.

    GitHub Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/14
15. As a developer, I have MATIC in my polygon wallet so that I can pay the transaction fees needed to upload the bot.

    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/15
16. As a developer, I have 100 Fort tokens in my wallet so that I can stake the bot which is necessary for it to run.

    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/16
17. As a developer, I have uploaded a docker image of a working bot to the Forta image repository so that it is visible on the Forta website.

    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/17

18. As a developer, I have added stake to my bot on the Forta Website so that it can run.

    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/18

19. As a developer, I have assembled a dataset in a useful format to use for machine learning training.

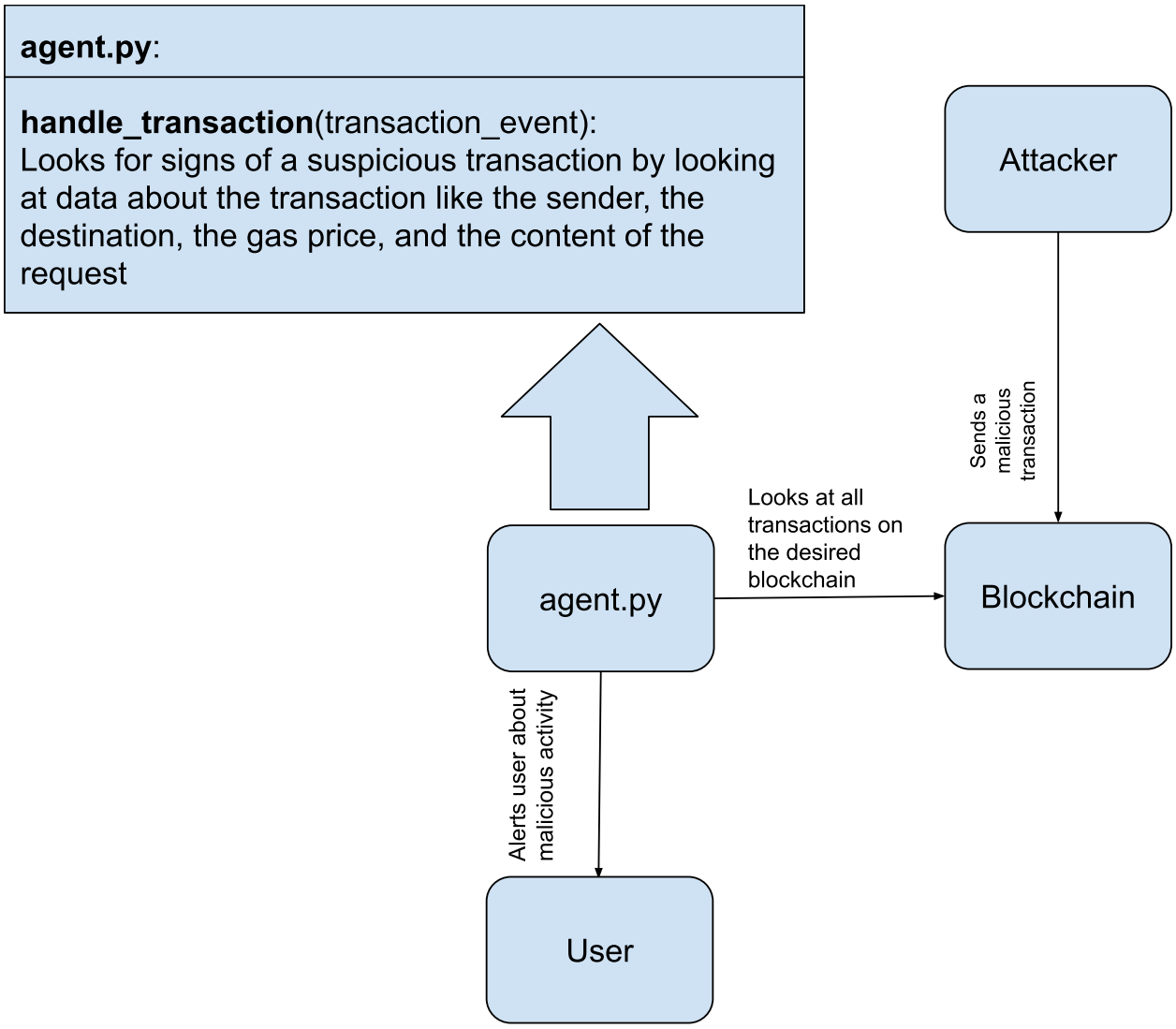
    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/19

20. As a developer, I have trained a machine learning model to predict whether a given bytecode is part of a malicious smart contract

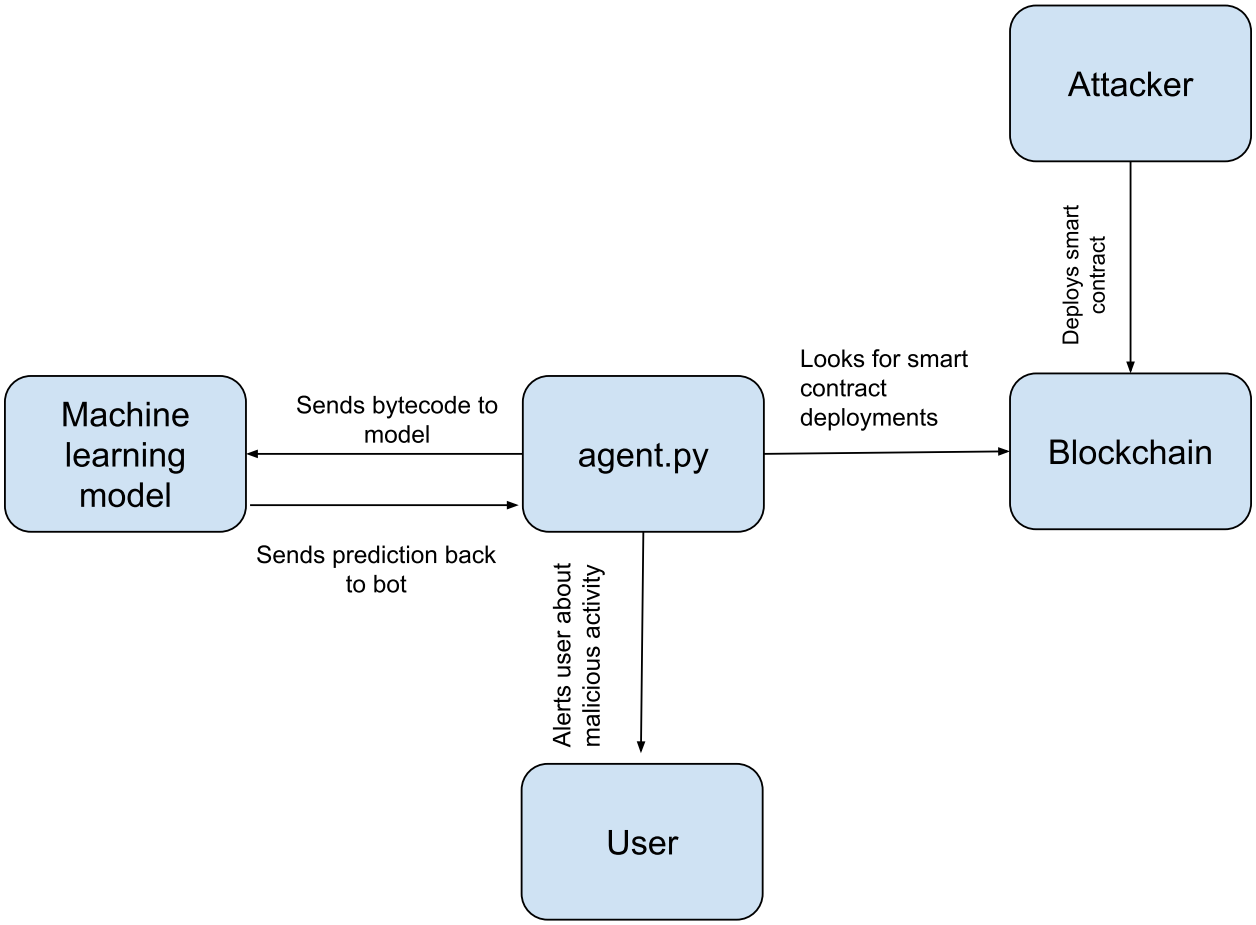
    Github Issue: https://github.com/Sapo-Dorado/FortaKnight/issues/20

System Models

Malicious transaction detection system



Malicious smart contract detection system



Appendices:

GitHub to manage our Code Base, along with their Kanban Board

forta and forta-agent libraries provided by Forta for writing bots

Python 3 to write the bot

Pytorch/numpy libraries for machine learning

Using Solidity to apply the smart contracts for detection