PHƯƠNG PHÁP TẠO BIẾN THỂ MÃ ĐỘC WINDOWS DỰA TRÊN HỌC TĂNG CƯỜNG CÓ KIỂM CHỨNG CHỨC NĂNG BẰNG PHÂN TÍCH ĐỘNG

Lê Trọng Nhân - 240101060

Tóm tắt

- Lớp: CS2205.CH190
- Link Github của nhóm: *https://github.com/letrongnhanlp/CS2205.CH190*
- Link YouTube video: *https://youtu.be/jyXXSeXHu6Q*
- Họ và Tên: Lê Trọng Nhân

Giới thiệu

- Gần đây, các mô hình học máy/học sâu đã được ứng dụng hiệu quả trong phát hiện lỗ hổng và mã độc. Tuy nhiên, chúng dễ bị đánh lừa bởi các biến đổi tinh vi, đặc biệt là những biến đổi do chính học máy tạo ra. Một thách thức lớn là các phương pháp hiện tại thường làm mất chức năng gốc của mã độc sau khi biến đổi.
- Trước thực tế này, nghiên cứu đặt ra câu hỏi: Liệu mô hình tạo biến thể mã độc Windows kết hợp cơ chế kiểm tra tính toàn vẹn chức năng bằng phân tích động có thể tạo ra các biến thể đa dạng, vừa qua mặt được hệ thống phát hiện lại vừa đảm bảo duy trì chức năng ban đầu hay không?

Mục tiêu

- Nghiên cứu và ứng dụng học tăng cường (RL) vào bài toán tạo biến thể mã độc Windows, tập trung vào việc bảo toàn chức năng gốc sau biến đổi.
- Xây dựng hệ thống tạo biến thể tự động tích hợp cơ chế kiểm chứng chức năng dựa trên phân tích động (API call analysis).
- Đánh giá hiệu quả hệ thống thông qua khả năng qua mặt mô hình phát hiện mã độc và tỷ lệ bảo toàn chức năng so với các phương pháp khác.

Nội dung và Phương pháp

Nội dung nghiên cứu cơ sở lý thuyết:

- Tìm hiểu định dạng PE file, đặc điểm mã độc Windows.
- Phân tích các kỹ thuật biến đổi mã độc (code obfuscation, FGSM, RL).
- Nghiên cứu học tăng cường và ứng dụng trong tạo biến thể bảo toàn chức năng mã độc Window.

Phương pháp:

- Đọc các công trình về phân tích và phát hiện mã độc Windows ở cả hướng động và tĩnh, nắm được các phương pháp và các thông tin thường được sử dụng để phát hiện mã độc.
- Đọc các công trình về áp dụng học tăng cường, đặc biệt là trong việc phát hiện mã độc (Windows hoặc các loại khác)

Nội dung và Phương pháp

Nội dung xây dựng hệ thống tạo biến thể dựa trên học tăng cường có kiểm chứng chức năng

- Thiết kế không gian hành động (action space) phù hợp.
- Phát triển mô-đun kiểm chứng chức năng bằng phân tích API call (so sánh vector API qua cosine similarity hoặc các mô-đun kiểm chứng).

Phương pháp:

- Sử dụng môi trường RL với hàm reward dựa trên:
  - Khả năng qua mặt trình phát hiện (MalConv).
  - Kiểm tra chức năng (so sánh API call gốc và biến thể).
- Triển khai các phép biến đổi mã độc (dùng thư viện LIEF, PEfile) và tích hợp sandbox (Cuckoo) để kiểm tra chức năng của mã độc.

Nội dung và Phương pháp

Nội dung thực nghiệm và đánh giá kết quả

- Đánh giá mô-đun kiểm tra chức năng.
- So sánh với phương pháp tạo biến thể khác

Phương pháp:

- Thu thập dữ liệu đầu vào cho toàn bộ mô hình là mẫu chương trình thực thi malware (dưới dạng file PE).
- Đánh giá khả năng kiểm chứng chức năng của mô-đun đã phát triển
- Thực nghiệm so sánh hiệu quả của mô hình RL và một số phương pháp tạo biến thể mã độc khác như việc chọn ngẫu nhiên action.

Kết quả dự kiến

- Độ chính xác của mô-đun kiểm chứng chức năng đạt kết quả cao (lớn hơn 90%)
- Hoàn thành huấn luyện agent RL có khả năng biến đổi mã độc vượt mặt tốt hơn hoặc bằng các phương pháp hiện tại nhưng khả năng tạo bảo toàn chức năng ban đầu tốt hơn.

Tài liệu tham khảo

[1] Edward Raff, Jon Barker, Jared Sylvester, Robert Brandon, Bryan Catanzaro, Charles Nicholas: Malware Detection by Eating a Whole EXE. arXiv:1710.09435

[2] J. Yuste, E. G. Pardo and J. Tapiador: "Optimization of code caves in malware binaries to evade machine learning detectors," Computers & Security, vol. 116, no. Elsevier, p. 102643, 2022.

[3] K. Lucas, M. Sharif, L. Bauer, M. K. Reiter and S. Shintre: "Malware makeover: Breaking ml-based static analysis by modifying executable bytes," Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, pp. 744-758, 2021.

[4] Hyrum S. Anderson, Anant Kharkar, Bobby Filar, David Evans, and Phil Roth: Learning to evade static pe machine learning malware models via rl. ArXiv, 2018.

[5] Z. Fang, J. Wang, B. Li, S. Wu, Y. Zhou, and H. Huang: Evading anti-malware engines with deep reinforcement learning. IEEE Access, 7:48867–48879, 2019.

[6] Labaca-Castro, Raphael and Franz, Sebastian and Rodosek, Gabi Dreo:AIMED-RL: Exploring Adversarial Malware Examples with Reinforcement Learning. Joint European Conference on Machine Learning and Knowledge Discovery in Databases

[7] Hyrum S. Anderson, Phil Roth. EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models. arXiv:1804.04637v2