PHÂN TÍCH TĨNH THEO NGỮ CẢNH: PHÁT HIỆN VÀ GIẢM CẢNH BÁO SAI TRONG MÃ NGUỒN DỄ BỊ TẤN CÔNG

Nguyễn Hoàng Kim Ngân - 240202010

Tóm tắt

- Lớp: CS2205.NOV2024
- Link Github của nhóm: https://github.com/kimngannguyenhoang1306/CS2205.NOV2024
- Link YouTube video: https://youtu.be/jnNq7FAIprA
- Ảnh + Họ và Tên của các thành viên

  Nguyễn Hoàng Kim Ngân - 240202010

Giới thiệu

Mã nguồn dễ bị tấn công chứa các đoạn mã có thể dẫn đến các lỗ hổng bảo mật như tràn bộ đệm, lỗi con trỏ, và các lỗ hổng quản lý bộ nhớ khác.

Những lỗi này thường xuất hiện do thiếu kiểm tra đầu vào, sử dụng không đúng cách các hàm hệ thống, hoặc quản lý tài nguyên không an toàn.

Mục tiêu

- Phát hiện mã nguồn dễ bị tấn công
- Giảm cảnh báo sai
- Tăng độ chính xác

Nội dung và Phương pháp

Phân tích ngữ cảnh mã nguồn:

- Xây dựng đồ thị luồng điều khiển (CFG).
- Phân tích đồ thị gọi hàm (Call Graph Analysis).
- Xác định cách đoạn mã nguy hiểm được gọi trong các điều kiện khác nhau.

Nội dung và Phương pháp

Phân tích phụ thuộc dữ liệu:

- Theo dõi dữ liệu từ nguồn không tin cậy (Taint Analysis).
- Xác định tham chiếu bộ nhớ (Alias Analysis).
- Mô phỏng thực thi mã giả định (Symbolic Execution).

Nội dung và Phương pháp

Giảm cảnh báo sai:

- Đối sánh mẫu để phát hiện lỗ hổng.
- Giải ràng buộc để kiểm tra điều kiện lỗi.
- Lọc nâng cao để loại bỏ cảnh báo không phù hợp.

Nội dung và Phương pháp

Tích hợp và tối ưu hóa:

- Phân tích mô-đun, xử lý song song, tối ưu hóa lưu trữ.

Nội dung và Phương pháp

Kiểm tra và đánh giá:

- Xây dựng bộ dữ liệu kiểm thử từ CVE.
- Đo lường tỷ lệ cảnh báo sai, độ chính xác và hiệu suất.

Kết quả dự kiến

- Khung phân tích:
  - Giảm >50% tỷ lệ cảnh báo sai so với các công cụ hiện tại.
  - Tăng >30% tốc độ phân tích.
- Sản phẩm nghiên cứu:
  - Công cụ phân tích tĩnh.
  - Bộ dữ liệu kiểm thử.
  - Tài liệu hướng dẫn và báo cáo nghiên cứu.

Tài liệu tham khảo

1. A. Johnson, "Reducing False Positives in Static Analysis with Context-Sensitive Techniques," IEEE Transactions on Software Engineering, 2021.
2. M. S. e. al., "Data Dependency and Control Flow in Vulnerability Detection," ACM SIGSOFT Software Engineering Notes, 2020.
3. K. Lee, "Symbolic Execution for Security Analysis," Proceedings of the IEEE, 2019.