BÁO CÁO TIẾN ĐỘ ĐỒ ÁN MÔN HỌC

**Môn học: Lập trình an toàn và khai thác lỗ hổng phần mềm**

**Tên chủ đề: …………………………………………………………………………**

*Mã nhóm: …………. Mã đề tài: ……….*

**Lớp**: ……….

1. **THÔNG TIN THÀNH VIÊN NHÓM:**

*(Sinh viên liệt kê tất cả các thành viên trong nhóm)*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Vương Đinh Thanh Ngân | 20521649 | [20521649@gm.uit.edu.vn](mailto:20521649@gm.uit.edu.vn) |
| 2 | Lê Minh Nhã | 20521690 | [20521690@gm.uit.edu.vn](mailto:1552yyyy@gm.uit.edu.vn) |
| 3 | Vũ Hoàng Thạch Thiết | 20521957 | [20521957@gm.uit.edu.vn](mailto:20521957@gm.uit.edu.vn) |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-0)**
   1. **Chủ đề nghiên cứu trong lĩnh vực An toàn phần mềm:**

🗹 Phát hiện lỗ hổng bảo mật phần mềm

🗹 Khai thác lỗ hổng bảo mật phần mềm

🞎 Sửa lỗi bảo mật phần mềm tự động

🞎 Lập trình an toàn

🞎 Khác: ……………….

* 1. **Tên bài báo tham khảo chính:**

|  |
| --- |
| Wang, Y., Zhang, C., Xiang, X., Zhao, Z., Li, W., Gong, X., ... & Zou, W. (2018, October). Revery: From proof-of-concept to exploitable. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1914-1927) |

* 1. **Dịch tên Tiếng Việt cho bài báo:**

|  |
| --- |
| Revery: Từ kiểm tra tính khả thi đến có thể khai thác được |

* 1. **Tóm tắt nội dung chính:**

|  |
| --- |
| <mô tả nội dung tóm tắt của bài báo/chủ đề trong vòng 350 từ>  Đề xuất một giải pháp Revery để tìm kiếm các trạng thái có thể khai thác trong các crashing paths, và gây ra việc khai thác tấn công control-flow (điều khiển luồng) dựa trên lỗ hổng heap |

* 1. **Các kỹ thuật chính được mô tả sử dụng trong bài báo:**

|  |
| --- |
| <mô tả các kỹ thuật chính được dùng trong nghiên cứu của bài báo>(liệt kê tóm tắt vai trò của các kỹ thuật đó – kèm hình ảnh minh họa về hệ thống/kỹ thuật/phương pháp)  Bài báo sử dụng 3 kỹ thuật cho 3 giai đoạn chính trong phương pháp đề xuất, bao gồm:   * **Kỹ thuật 01: Layout-contributor digraph**   Mô tả vai trò, nhiệm vụ: Mô tả đặc điểm các lỗ hổng của bố cục bộ nhớ và các đóng góp của chúng, cho phép nhiều phân tích đến khai thác   * **Kỹ thuật 02: Layout-oriented fuzzing**   Mô tả: Tìm kiếm các trạng thái có thể khai thác đường dẫn phân kỳ(diverging paths) , mà không thực thi biểu tượng (symbolic)   * **Kỹ thuật 03: Control-flow stitching**   Mô tả: Kết hợp crashing paths và diverging paths với nhau, tổng hợp các đầu vào EXP có thể kích hoạt cả lỗ hổng và trạng thái có thể khai thác  Hình ảnh minh họa: |

* 1. **Môi trường thực nghiệm của bài báo:**

|  |
| --- |
| <mô tả môi trường thực nghiệm của nhóm tác giả, bao gồm cấu hình máy tính (phần cứng, phần mềm), các chương trình hỗ trợ để hiện thực phương pháp, ngôn ngữ lập trình được sử dụng, các chương trình phần mềm dùng để kiểm tra khả năng/tính năng của phương pháp>   * Cấu hình máy tính: Ubuntu 17.04 (trên server với 115G RAM và Intel Xeon (R) CPU E5-2620 @2.40GHz\*24) * Các công cụ hỗ trợ sẵn có: công cụ phân tích nhị phân angr và Fuzzer AFL phổ biến   + Bao gồm: 1334 dòng code cho phân tích lỗ hổng, 190 dòng code cho khai thác diverging paths với fuzzing, 1249 dòng code cho stitch paths và dẫn đến kết quả khai thác   * Ngôn ngữ lập trình để hiện thực phương pháp: **C** * Đối tượng nghiên cứu (chương trình phần mềm dùng để kiểm tra tính khả thi của phương pháp/tập dữ liệu – nếu có): tác giả dùng tập dữ liệu từ 15 cuộc thi CTF bao gồm 19 chương trình phần mềm chứa lỗ hổng bảo mật      * Tiêu chí đánh giá tính hiệu quả của phương pháp: <vd: số lượng lỗ hổng được phát hiện, thời gian chạy, coverage….>   + Các kết quả thực thi được so sánh với các công cụ như AFL, symbolic execution |

* 1. **Kết quả thực nghiệm của bài báo:**

|  |
| --- |
| <mô tả ngắn gọn kết quả thực nghiệm của bài báo, tự nhận xét về khả năng, ưu và nhược điểm của phương pháp được đề cập trong bài báo>  - Kết quả: Họ đã thành công trong việc tìm kiếm các trạng thái có thể khai thác trong diverging paths thay vì crashing paths (họ đã kết hợp cả diverging paths và crashing paths lại với nhau), với giải pháp “layout-oriented fuzzing” và “control-flow stitching” -> thành công trong các khai thác làm việc cho một số lỗ hổng nhất định (ở đây là lỗ hổng trên các chương trình CTF)  - Ưu điểm:  + Thời gian chạy nhanh hơn  + Khai thác dựa trên Heap  + Có thể tìm kiếm các states không chỉ trong crashing paths mà còn ở diverging paths và tạo việc khai thác control-flow khi có thể  - Nhược điểm:  + Chỉ hỗ chợ cho ‘Object Level Corruption Dectection”  + Dựa vào Angr để thực hiện “symbolic execution”, mà nó được viết lại các chức năng thư viện trong python và nối các chức năng ban đầu do đó nó không hỗ trợ hầu hết các chương trình thực tế. Nên họ chỉ đánh giá Revery trên các chương trình CTF |

* 1. **Công việc/tính năng/kỹ thuật mà nhóm thực hiện lập trình và triển khai cho demo:**

|  |
| --- |
| <liệt kê kế hoạch của nhóm, các công việc mà nhóm dự định thực hiện cho đề tài dựa trên phân tích phương pháp/hệ thống được sử dụng trong bài báo đã tham khảo>  - Tìm hiểu về công cụ AFL, và công cụ phân tích nhị phân QEMU  - Dùng các công cụ trên để khai thác các thử thách CTF  <liệt kê các công việc đã thực hiện+ kèm theo kết quả của công việc này>  - Đã cài đặt được tool AFL trên Kali Linux  -> tìm hiểu cách hoạt động của AFL  <các công việc cần thực hiện tiếp theo>  - Dùng công cụ AFL khai thác một số thử thách CTF |

* 1. **Các khó khăn, thách thức hiện tại khi thực hiện:**

|  |
| --- |
| <mô tả khó khăn của nhóm (nếu có), ….>  - Bài báo không có public framework, code lại thì quá cực vì kiến thức hạn hẹp, nên nhóm chúng em tìm hiểu một phần có liên quan đến việc phát triển khung làm việc của đề tài trong bài báo |

1. **TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH SO VỚI KẾ HOẠCH THỰC HIỆN:**

|  |
| --- |
| <VD:>  ~40-50% |

1. **NHẬT KÝ PHÂN CÔNG NHIỆM VỤ:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Phân công nhiệm vụ** |
| 1 | Đọc hiểu và phân tích bài báo  Viết báo cáo | Vương Đinh Thanh Ngân, Lê Minh Nhã, Vũ Hoàng Thạch Thiết |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

BÁO CÁO CHI TIẾT

**<phần này chỉ dùng cho báo cáo đề tài môn học ở giai đoạn cuối kì>**

**Phần bên dưới của báo cáo này là tài liệu báo cáo tổng kết - chi tiết của nhóm thực hiện cho đề tài này.**

Mô tả các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, số liệu thống kê trong bảng biểu, có giải thích)

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

*Ví dụ: [*NT101.K11.ANTT*]-Exe01\_Group03.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung tương ứng theo mô tả [↑](#footnote-ref-0)