

Phòng thí nghiệm An toàn thông tin



Meta-Path Based Attentional Graph Learning Model for Vulnerability Detection



G-07

Phòng thí nghiệm An toàn thông tin (InSecLab)
Trường ĐH Công nghệ Thông Tin, ĐHQG Tp. HCM



Nội dung báo cáo



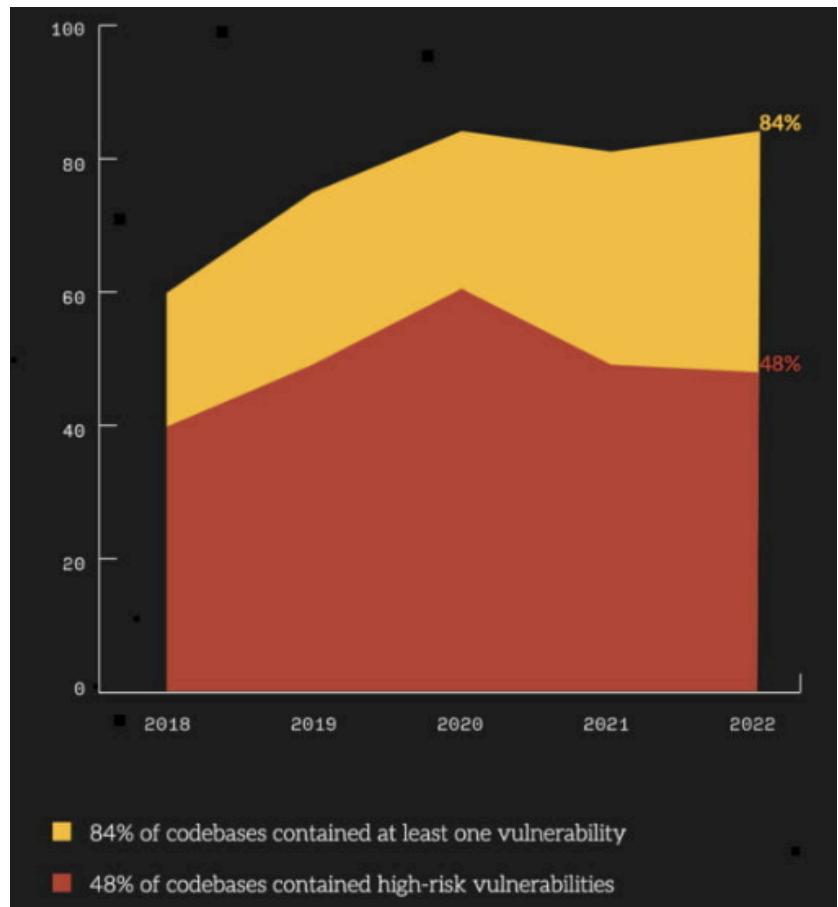
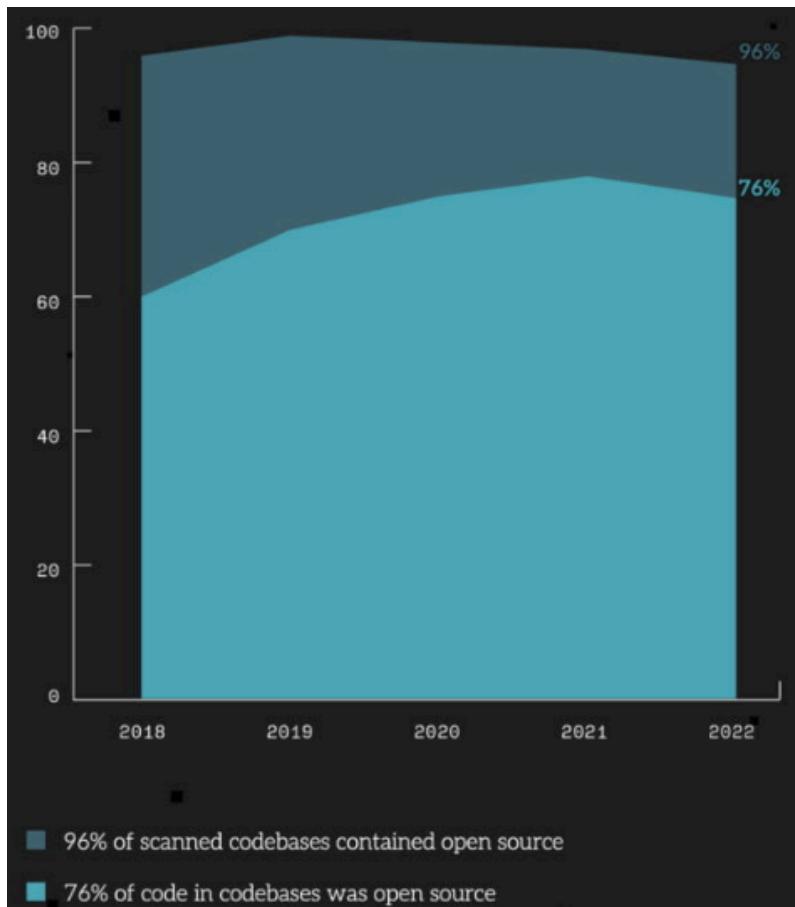
- **Phần I: Ngữ cảnh**
- Phần II: Giải pháp
- Phần III: Hướng phát triển
- Phần IV: Demo



Ngữ cảnh



“Phân tích rủi ro và an ninh nguồn mở” (OSSRA) năm 2023 của Synopsys, 96% cơ sở mã mà Synopsys đánh giá vào năm 2022 chứa mã nguồn mở. Điều đáng chú ý, có đến 84% cơ sở mã được quét chứa ít nhất một lỗ hổng mã nguồn mở đã biết và gần một nửa trong số đó (48%) chứa các lỗ hổng rủi ro cao (High-risk vulnerabilities).



1. **MAGNET** ra đời nhằm khắc phục những hạn chế của các phương pháp phát hiện lỗi trước đó:

- Các phương pháp cũ thường **không xem xét** đồng thời các loại nút và cạnh khác nhau, tức là **các mối quan hệ không đồng nhất**, mà chúng rất hữu ích cho việc nắm bắt các mẫu mã chứa lỗi.
- GNN bị **hạn chế trong việc xử lý các mối quan hệ giữa các nút ở xa**, vì GNN chủ yếu sử dụng thông tin lân cận để truyền thông điệp. Do số lượng nút lớn và độ sâu của các đồ thị dựa trên AST, các phương pháp hiện tại vẫn **đối mặt với thách thức trong việc học các phụ thuộc tầm xa**

-> **MAGNET** được phát triển để **giải quyết hạn chế** này bằng cách sử dụng đồ thị meta-path và mạng nơ ron đồ thị chú ý phân cấp dựa trên meta-path để xử lí các vấn đề này.



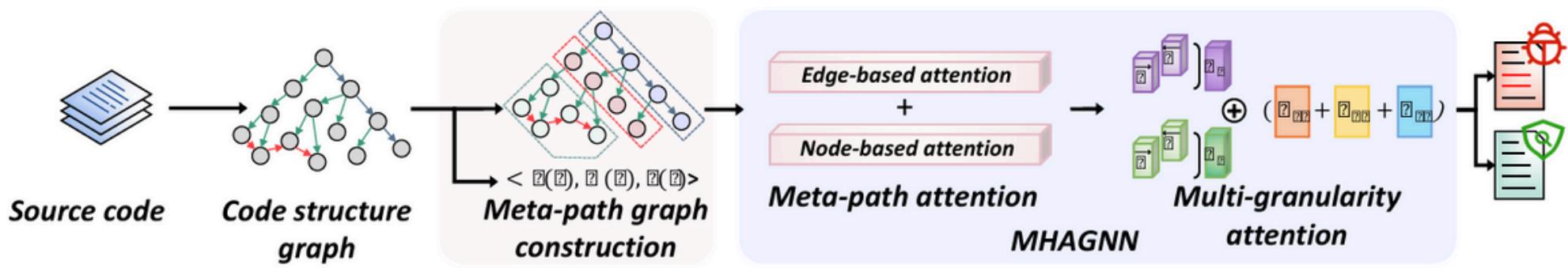
Nội dung báo cáo



- Phần I: Ngữ cảnh
- **Phần II: Giải pháp**
- Phần III: Hướng phát triển
- Phần IV: Demo



Giải pháp



Hình 2: Mô hình tổng quan của Magnet



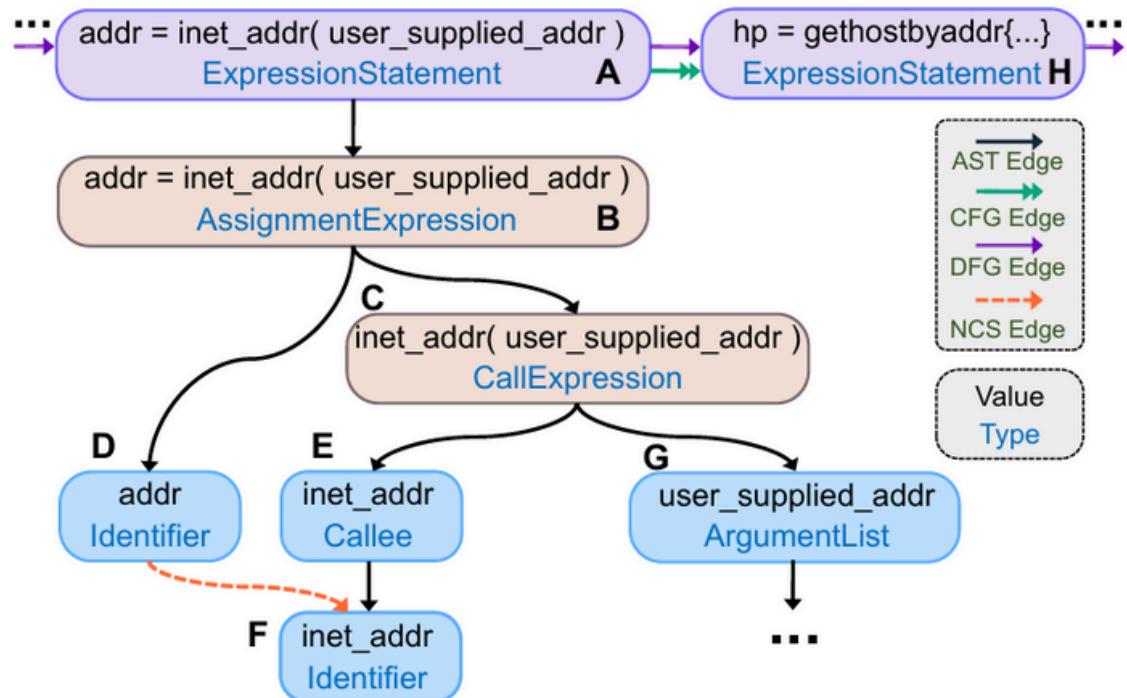
Giải pháp - Meta-path



- **AST:** Biểu diễn cấu trúc cú pháp của mã nguồn.
- **CFG:** Biểu diễn luồng điều khiển.
- **DFG:** Biểu diễn luồng dữ liệu.
- **NCS:** Chuỗi mã tự nhiên.

```
void host_lookup(char *user_supplied_addr)
{
    struct hostent *hp;
    in_addr_t *addr;
    char hostname[64];
    in_addr_t inet_addr(const char *cp);
    validate_addr_form(user_supplied_addr);
    addr = inet_addr(user_supplied_addr);
    hp = gethostbyaddr( addr, sizeof(struct in_addr), AF_INET);
    strcpy(hostname, hp->h_name);
}
```

Hình 3. Source Code



Hình 4. Code Structure Graph

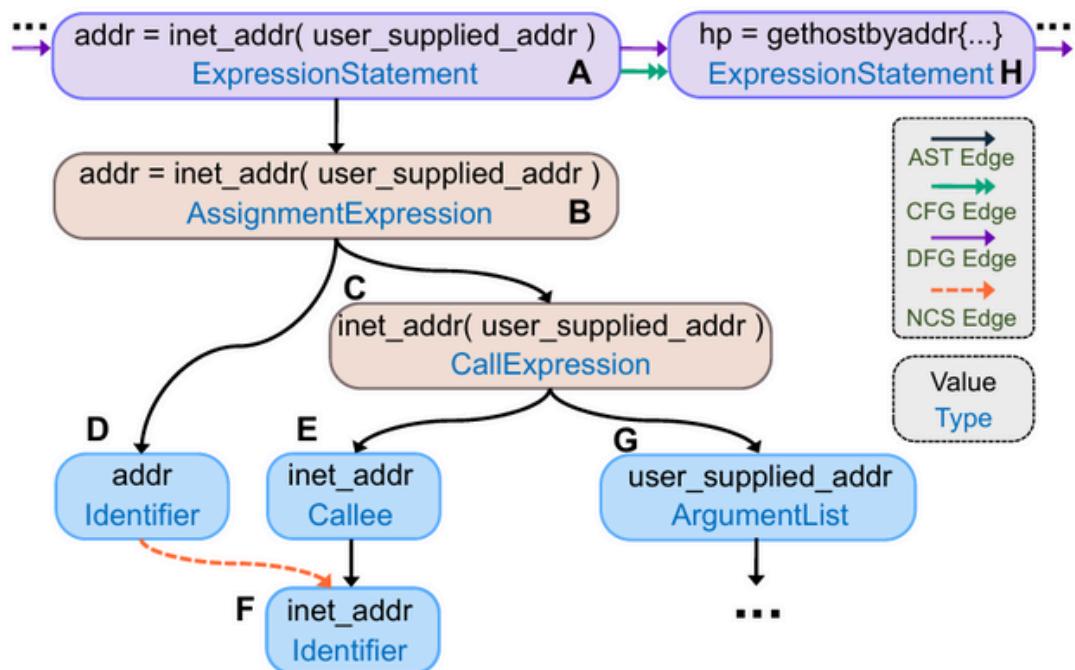


Giải pháp - Meta-path



Trong ngữ cảnh của MAGNET, mỗi meta-path biểu thị cho một mối quan hệ phức tạp trong đồ thị cấu trúc mã.

- MAGNET nhóm các loại nút trong đồ thị cấu trúc mã thành **3 mức độ chi tiết (granularity)**:
- Statement:** Biểu diễn cả câu lệnh.
- Expression:** Bao gồm hai hoặc nhiều toán tử/toán hạng.
- Symbol:** Các nút còn lại được phân loại là nút "Symbol" cho đơn giản.



Giải pháp - Meta-path

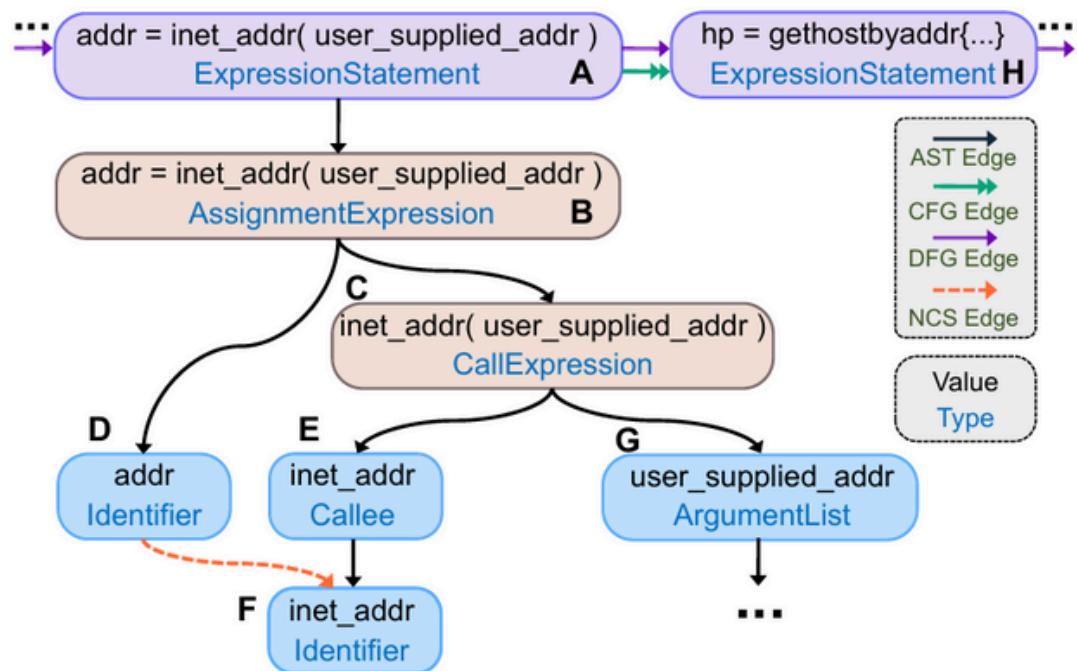


Trong ngữ cảnh của MAGNET, mỗi meta-path biểu thị cho một mối quan hệ phức tạp trong đồ thị cấu trúc mã.

- Mỗi meta-path là một bộ ba $(\tau(s), \Psi(e), \tau(t))$, trong đó:

- $\tau(s)$: Loại nút của nút nguồn (source node) s.
- $\Psi(e)$: Loại cạnh của cạnh e.
- $\tau(t)$: Loại nút của nút đích (target node) t.

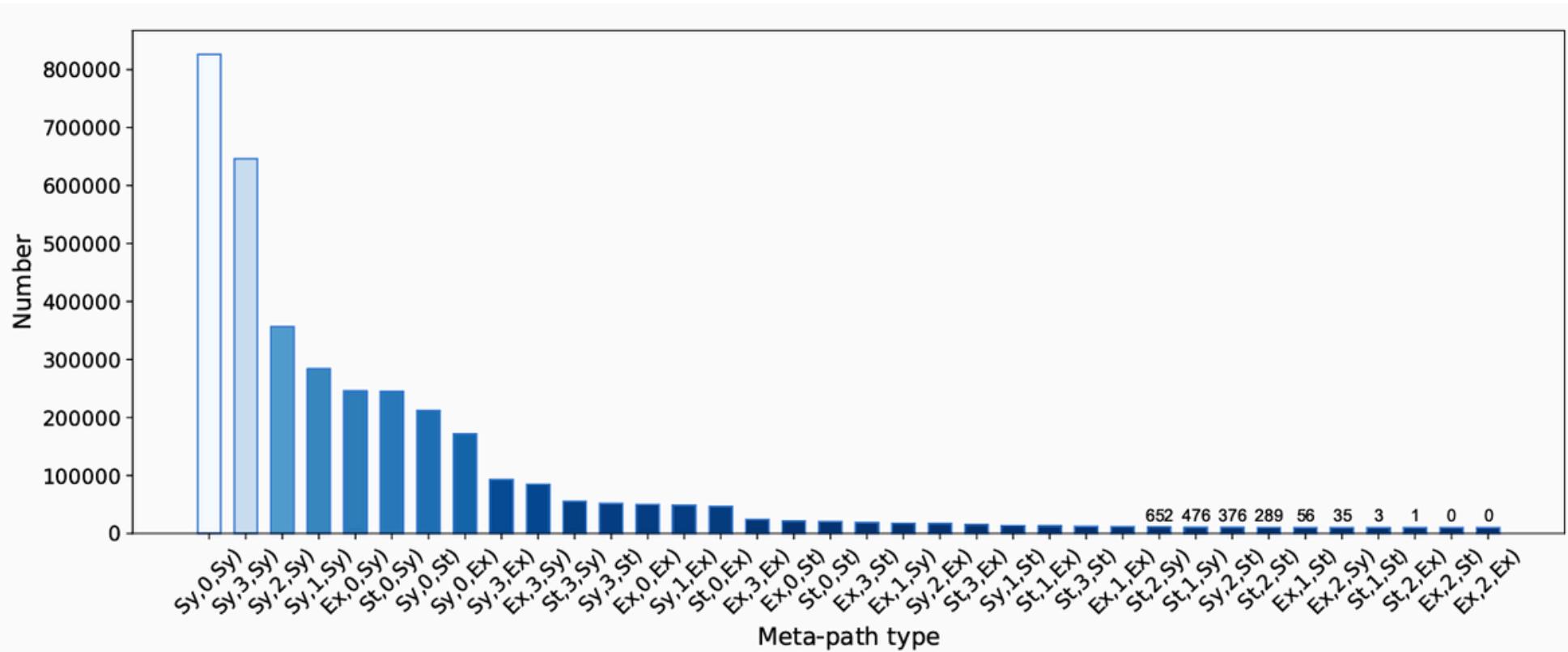
EX: Meta-path (Statement, AST, Expression) biểu thị mối quan hệ giữa một nút thuộc loại Statement và một nút thuộc loại Expression thông qua một cạnh AST.



Giải pháp - Meta-path



- Để tạo điều kiện thuận lợi cho việc học biểu diễn của các mối quan hệ phức tạp trong đồ thị, MAGNET đã **lọc bỏ các loại meta-path hiếm gặp** và chỉ **32 loại meta-path phổ biến nhất** được giữ lại để xây dựng đồ thị meta-path -> Việc này giúp **giảm độ phức tạp** của mô hình và **tăng hiệu quả** học biểu diễn đồ thị, từ đó **cải thiện hiệu suất phát hiện lối**.

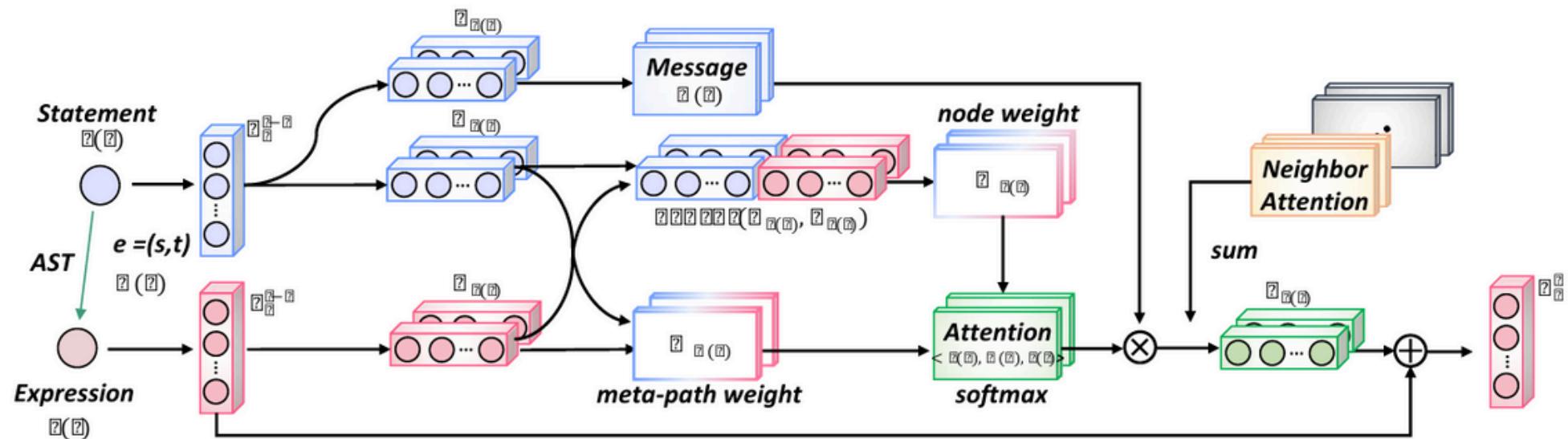


Giải pháp - Mạng nơ ron đồ thị MHAGNN



MHAGNN bao gồm 2 module chính : **Cơ chế chú ý Meta-path** và **Cơ chế chú ý đa mức độ (Multi-granularity attention)**

1. **Cơ chế chú ý meta-path:** Module này tập trung vào việc học biểu diễn của các mối quan hệ phức tạp, được thể hiện bởi các bộ ba ($\tau(s)$, $\psi(e)$, $\tau(t)$) trong meta-path.
 - Chú ý dựa trên nút (Node-based attention)
 - Chú ý dựa trên cạnh (Edge-based attention)
- > Kết hợp điểm chú ý nút và cạnh để tạo điểm chú ý meta-path



Giải pháp - Mạng nơ ron đồ thị MHAGNN



Chú ý dựa trên nút (Node-based attention) : Đánh giá mức độ quan trọng của loại nút đối với biểu diễn cấu trúc đồ thị.

- Đầu tiên, ta tính toán vectơ đại diện cho mỗi quan hệ giữa **nút nguồn s** và **nút đích t**. Vectơ này được tạo ra bằng cách **nối vectơ embedding** của nút s ($K^l(s)$) với vectơ embedding của nút t ($Q^l(t)$).
- Sau đó, ta **nhân vectơ** này với **ma trận trọng số** $W^l \tau(t)$ tương ứng với loại nút của nút đích t. Phép nhân này giúp đánh giá mức độ quan trọng của loại nút t đối với mối quan hệ giữa s và t.
- Cuối cùng, **kết quả được đưa qua hàm sigmoid** để thu được điểm attention nằm trong khoảng từ 0 đến 1.

$$Att_{node}^l = \sigma \left(W_{\tau(t)}^l \cdot \left(K^l(s) || Q^l(t) \right) \right) \quad (1)$$

$$K^l(s) = Linear_{\tau(s)}(h_s^{l-1}) \quad (2)$$

$$Q^l(t) = Linear_{\tau(t)}(h_t^{l-1}) \quad (3)$$





Chú ý dựa trên cạnh (Edge-based attention) : Đánh giá mức độ quan trọng của loại cạnh trong việc kết nối nút nguồn và nút đích.

- Ta cũng tính toán vectơ đại diện cho mỗi quan hệ giữa nút nguồn s và nút đích t, sử dụng vectơ embedding của chúng ($KI(s)$ và $QI(t)$).
- Vectơ này được nhân với ma trận trọng số $W\psi(e)$ tương ứng với loại cạnh e.
- Kết quả được nhân thêm với tham số $\mu(\psi(e))$ đại diện cho mức độ quan trọng của loại cạnh e.
- Cuối cùng, kết quả được chia cho $\sqrt{d/h}$ để chuẩn hóa

$$Att_{edge}^l = \left(K^l(s) W_{\psi(e)} Q^l(t)^T \right) \cdot \frac{\mu(\psi(e))}{\sqrt{\frac{d}{h}}} \quad (4)$$





Chú ý dựa trên Meta-path (Meta-path attention) : Kết hợp node-based attention và edge-based attention để biểu diễn quan hệ không đồng nhất giữa các nút.

- Đầu tiên, ta cộng điểm node-based attention và điểm edge-based attention để thu được điểm attention tổng hợp cho mỗi quan hệ giữa nút s và t.
- Sau đó, ta nối kết quả và áp dụng hàm softmax để chuẩn hóa điểm attention, đảm bảo tổng các điểm attention bằng 1.

$$Att_{(\tau(s), \psi(e), \tau(t))}^l = \text{softmax} \left(\|_1^H \left(Att_{edge}^l + Att_{node}^l \right) \right) \quad (5)$$

--



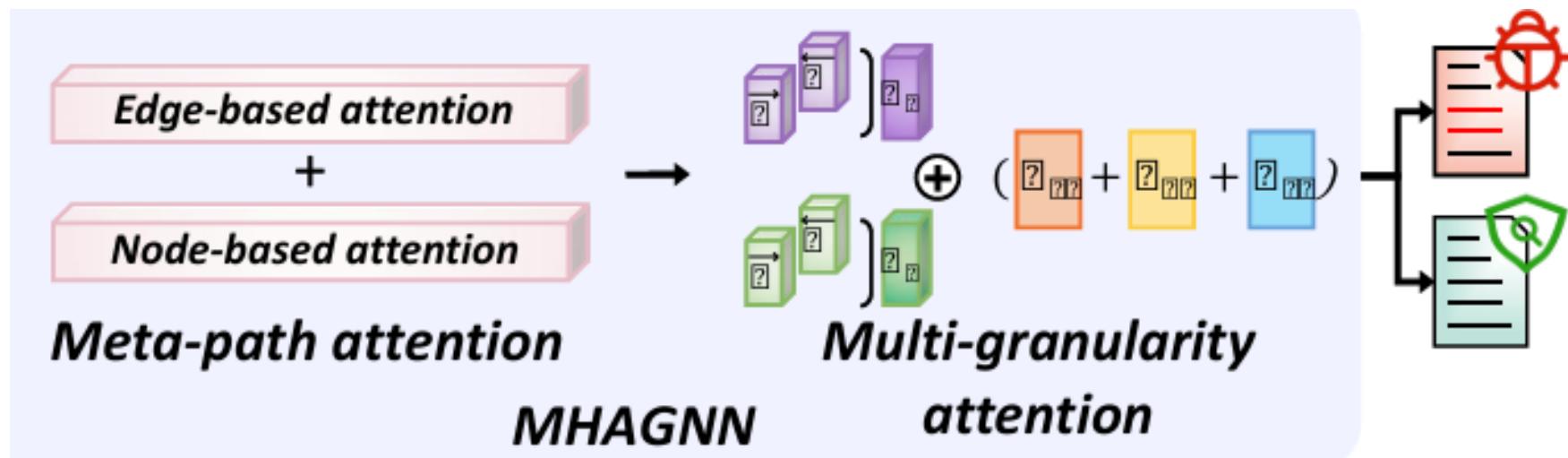
Giải pháp - Mạng nơ ron đồ thị MHAGNN



MHAGNN bao gồm 2 module chính : **Cơ chế chú ý Meta-path** và **Cơ chế chú ý đa mức độ (Multi-granularity attention)**

2. **Cơ chế chú ý đa mức độ (Multi-granularity attention)**: Module này được thiết kế để nắm bắt phụ thuộc tầm xa trong đồ thị meta-path.

- **Áp dụng average-pooling và max-pooling**: Trích xuất thông tin tổng hợp từ các nút ở các mức độ chi tiết (granularity) khác nhau.
 - **Tính điểm chú ý cho mỗi mức độ chi tiết**.
- > **Kết hợp thông tin từ các mức độ chi tiết khác nhau**: Tạo biểu diễn đồ thị cuối cùng, MHAGNN dự đoán xem mã nguồn có chứa lỗi hay không.





Multi-granularity attention: Nắm bắt phụ thuộc phạm vi xa trong đồ thị meta-path bằng cách xem xét mức độ quan trọng của các quan hệ không đồng nhất ở các mức độ chi tiết (granularities) khác nhau.

- Đầu tiên, ta tính toán đặc trưng cho mỗi mức độ chi tiết i (F_i) bằng cách ghép nối (concat) các vectơ embedding của các nút thuộc loại i .
- Sau đó, ta áp dụng lớp average-pooling và max-pooling lên F_i để thu được hai vectơ đặc trưng đại diện cho mức độ chi tiết i .
- Hai vectơ này được nhân với trọng số $\omega_{1,i}$ và $\omega_{2,i}$ tương ứng, sau đó cộng lại với nhau.
- Cuối cùng, kết quả được đưa qua mạng perceptron đa lớp (MLP) và hàm sigmoid để thu được điểm multi-granularity attention cho mức độ chi tiết i

$$M = \sigma (MLP(\omega_{1,i} \cdot AvgPool(F_i) + \omega_{2,i} \cdot MaxPool(F_i)))$$
$$(i = st, ex, sy) \quad (8)$$



Giải pháp



- Nhìn chung, MAGNET đạt kết quả tốt hơn và vượt trội hơn tất cả sáu phương pháp được so sánh trên cả ba bộ dữ liệu về chỉ số F1 và Recall
- Ba phương pháp dựa trên đồ thị (Devign, Reveal và IVDetect) cho kết quả tốt hơn so với ba phương pháp dựa trên token.

Metrics(%) \ Dataset	FFMPeg+Qemu [14]				Reveal [18]				Fan <i>et al.</i> . [37]			
Baseline	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
VulDeePecker	49.61	46.05	32.55	38.14	76.37	21.13	13.10	16.17	81.19	38.44	12.75	19.15
Russell <i>et al.</i>	57.60	54.76	40.72	46.71	68.51	16.21	52.68	24.79	86.85	14.86	26.97	19.17
SySeVR	47.85	46.06	58.81	51.66	74.33	40.07	24.94	30.74	90.10	30.91	14.08	19.34
Devign	56.89	52.50	64.67	57.95	87.49	31.55	36.65	33.91	92.78	30.61	15.96	20.98
Reveal	61.07	55.50	70.70	62.19	81.77	31.55	61.14	41.62	87.14	17.22	34.04	22.87
IVDetect	57.26	52.37	57.55	54.84	-	-	-	-	-	-	-	-
MAGNET	63.28	56.27	80.15	66.12	91.60	42.86	61.68	50.57	91.38	22.71	38.92	28.68

Hình 5: So sánh hiệu suất với các phương pháp cũ



Nội dung báo cáo



- Phần I: Ngữ cảnh
- Phần II: Giải pháp
- **Phần III: Hướng phát triển**
- Phần IV: Demo



Hướng phát triển



- Việc xây dựng đồ thị meta-path và tính toán điểm chú ý meta-path có thể **tốn kém về mặt tính toán**, đặc biệt khi xử lý các dự án phần mềm lớn với đồ thị cấu trúc phức tạp -> Nghiên cứu các kỹ thuật tối ưu hóa để cải thiện hiệu suất của MAGNET, ví dụ như sử dụng các phương pháp lấy mẫu đồ thị (graph sampling) hoặc xấp xỉ điểm chú ý (attention approximation).



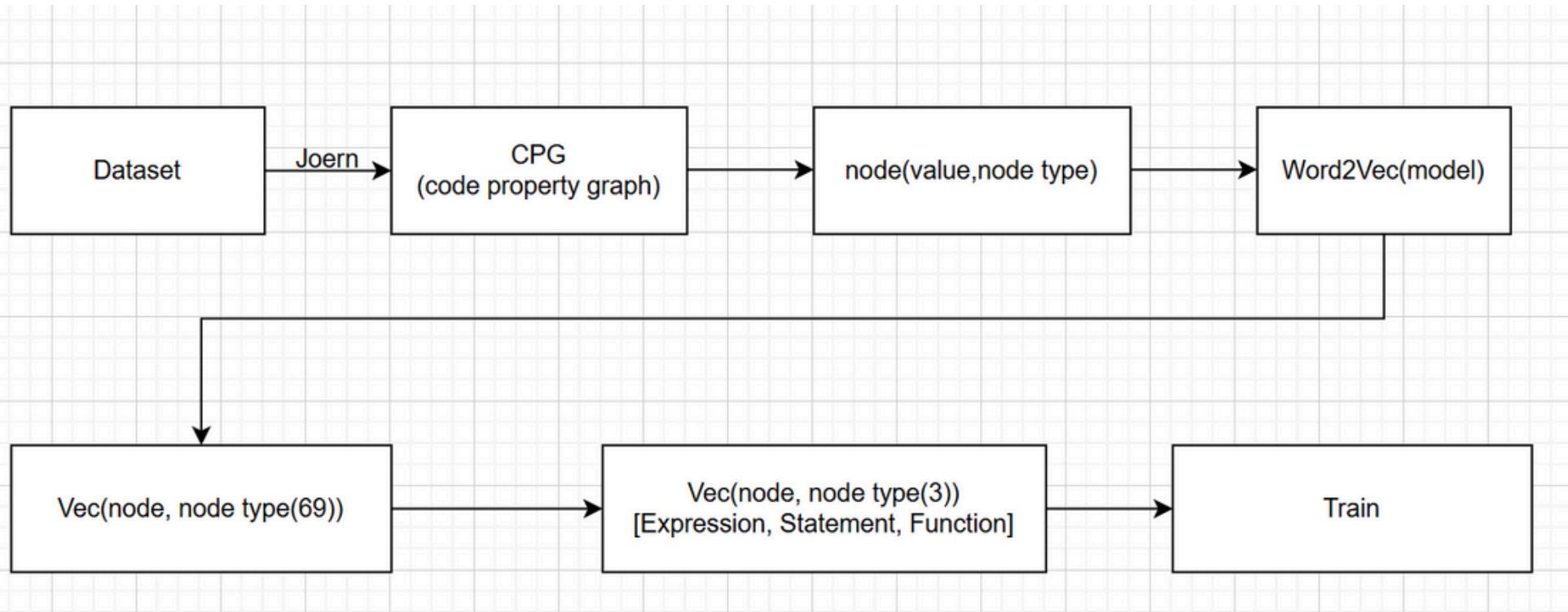
Nội dung báo cáo



- Phần I: Ngữ cảnh
- Phần II: Giải pháp
- Phần III: Hướng phát triển
- **Phần IV: Demo**



Tiền xử lý dữ liệu



Training configs



- batch_size = 512.
- lr = 5e-4, epoch = 10;21, patience = 30.
- opt ='Adam', weight_decay=1.2e-6,lr=5e-4, optim = Adam(model.parameters(), weight_decay=1.3e-6).



Kết quả



	FFMpeg-Quemu		Chrome-Debian	
Test Specimen Type	Vulnerables	Non-Vulnerables	Vulnerables	Non-Vulnerables
Number of Test Specimens	12460	14858	2240	20494

	FFMpeg-Quemu			Chrome-Debian		
Baseline	Accuracy	Precision	F1 score	Accuracy	Precision	F1 score
MAGNET	45.69	45.69	62.72	86.84	33.75	33.21



Kết quả



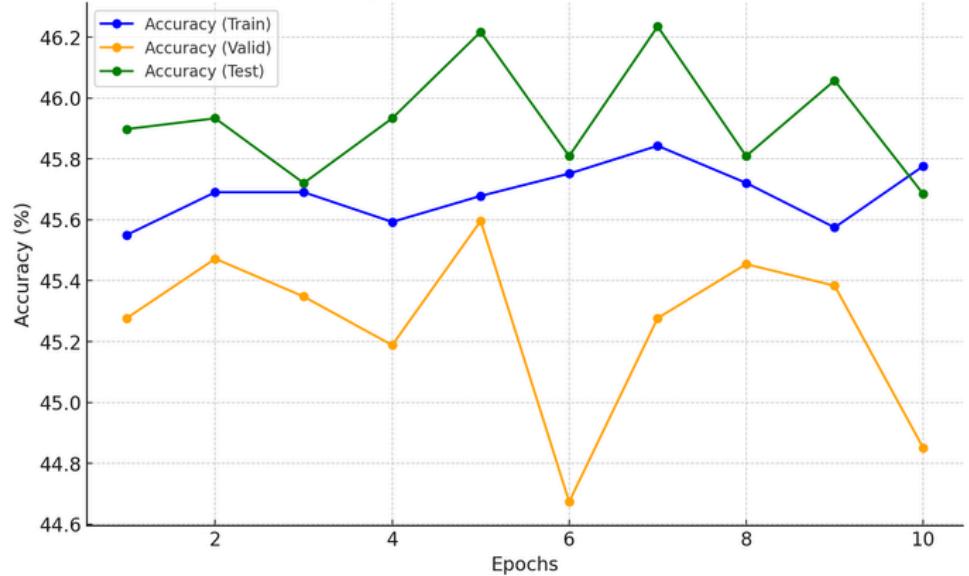
Kết quả thực nghiệm trên tập Chrome-Debian



Kết quả

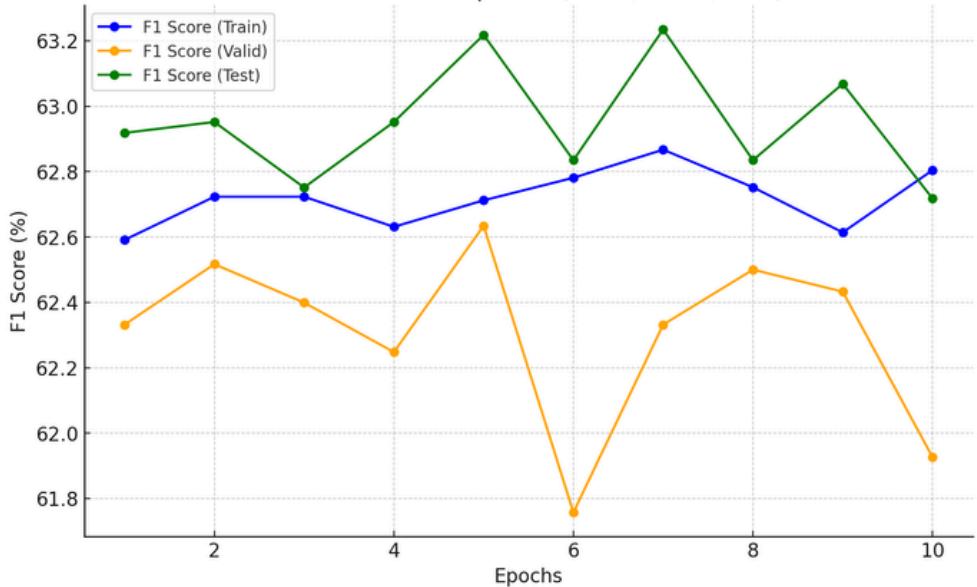


Accuracy Over Epochs (Train, Valid, Test)



Kết quả thực nghiệm trên tập FFMpeg-Quemu

F1 Score Over Epochs (Train, Valid, Test)



Tài liệu tham khảo

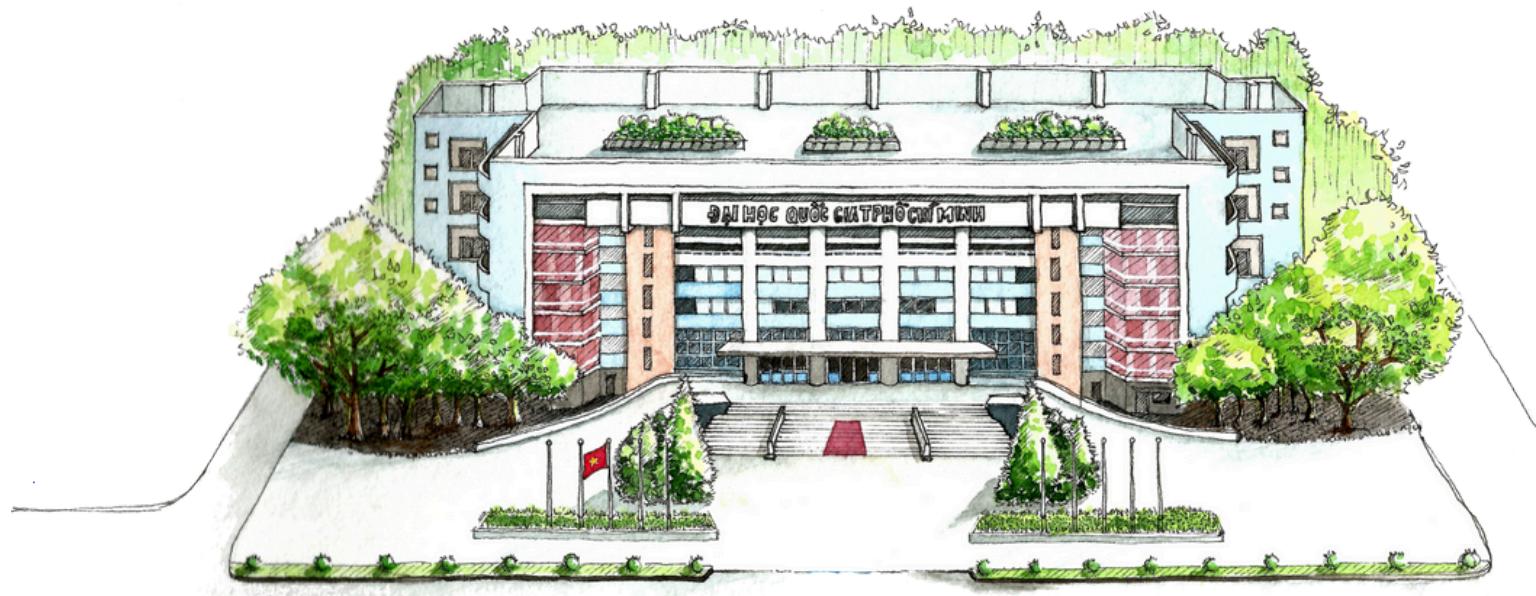


Wen, X. C., Gao, C., Ye, J., Li, Y., Tian, Z., Jia, Y., & Wang, X. (2023). Meta-path based attentional graph learning model for vulnerability detection. *IEEE Transactions on Software Engineering*.





Trường ĐH Công nghệ Thông tin
Đại Học Quốc Gia TP. HCM



Xin cảm ơn.



Nhóm nghiên cứu InSecLab

Phòng Thí nghiệm An toàn thông tin

Email: inseclab@uit.edu.vn

Website: <https://inseclab.uit.edu.vn/>

Fanpage: <https://www.facebook.com/inseclab>