

CONESCAPANHONDURAS2025paper58.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

Document Details

Submission ID

trn:oid:::14348:477772553

Submission Date

Jul 31, 2025, 11:27 PM CST

Download Date

Aug 12, 2025, 2:38 PM CST

File Name

CONESCAPANHONDURAS2025paper58.pdf

File Size

646.3 KB

6 Pages




4,570 Words

28,125 Characters

12% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 11%  Internet sources
- 7%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags




0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

11%  Internet sources
7%  Publications
0%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	conf.miigaik.ru	2%
2	Internet	arxiv.org	1%
3	Internet	www.diva-portal.org	1%
4	Internet	www.hindawi.com	<1%
5	Internet	openaccess.uoc.edu	<1%
6	Internet	www.sju.edu.in	<1%
7	Internet	internationalpubls.com	<1%
8	Internet	joiv.org	<1%
9	Internet	hal.archives-ouvertes.fr	<1%
10	Publication	Joshep Marua. "Corporate Social Responsibility and SDGs: A Bibliometric Analysis ...	<1%
11	Internet	ar5iv.org	<1%

12	Internet	tickelia.com	<1%
13	Internet	assets-eu.researchsquare.com	<1%
14	Internet	ieeexplore.ieee.org	<1%
15	Internet	roderic.uv.es	<1%
16	Internet	www.coursehero.com	<1%
17	Internet	www.mdpi.com	<1%
18	Internet	www.springerprofessional.de	<1%
19	Publication	Ahmed El-Kosairy, Nashwa AbdelBaki. "Next-Gen Cloud Security: IRDS4C's Decepti...	<1%
20	Internet	archive.org	<1%
21	Internet	comunicaciones.poligran.edu.co	<1%
22	Internet	slam.ece.utexas.edu	<1%
23	Internet	www.researchgate.net	<1%
24	Internet	www.techrxiv.org	<1%
25	Internet	docs.google.com	<1%

26	Internet	www.jove.com	<1%
27	Publication	Daniel Gibert, Nikolaos Totosis, Constantinos Patsakis, Quan Le, Giulio Zizzo. "Ass...	<1%
28	Internet	brasil.indymedia.org	<1%
29	Internet	iris.unitn.it	<1%
30	Internet	journals.riverpublishers.com	<1%
31	Internet	oa.upm.es	<1%
32	Internet	rinacional.tecnm.mx	<1%
33	Publication	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in He...	<1%

SPADE++: Adaptive Multimodal Cyber Deception Strategies Using Generative AI

1

line 1: 1st Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 2nd Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

line 1: 3rd Given Name Surname
line 2: dept. name of organization
(of Affiliation)
line 3: name of organization
(of Affiliation)
line 4: City, Country
line 5: email address or ORCID

Abstract—The use of generative artificial intelligence has transformed cyber defense strategies. PADE++ is an adaptative and multimodal framework that automates the creation of realistic and context-aware cyber deception tactics. This approach enhances threat detection, misleads attackers, and reinforces security systems in real time.

Index Terms—Cyber deception, AI, Prompt Engineering, Malware, Cybersecurity.

I. INTRODUCCIÓN

28

La creciente sofisticación y diversidad de los ataques informáticos exige enfoques de defensa más dinámicos y adaptativos. El uso de ataques dirigidos y técnicas avanzadas de la evasión como el malware polimórfico ha superado la capacidad de los mecanismos tradicionales de seguridad, los cuales suelen ser estáticos y reactivos [1], [2]. En este contexto, el ciberengaño emerge como una herramienta complementaria complementaria de la defensa activa, diseñada especialmente para confundir, retrasar y obtener inteligencia sobre los atacantes [3].

16

La automatización y la adaptabilidad se han convertido en elementos clave en la evolución de la defensa cibernética. La incorporación de inteligencia artificial generativa (GenAI) ha transformado procesos tradicionalmente manuales, como la creación de “señuelos” o artefactos de engaño, en operaciones automáticas y altamente personalizadas. No obstante, la mayoría de los enfoques actuales siguen centrados en recursos estáticos, como honeypot o honeytokens, con capacidades limitadas de adaptación frente a amenazas dinámicas [4], [3].

5

El marco SPADE (Structured Prompting for Adaptive Deception Engineering), una solución que combina ingeniería estructura de prompts y modelos de lenguaje de gran escala (LLMs) para generar tácticas de ciberengaño adaptativas, y contextualizadas. Este enfoque ha demostrado resultados prometedores en términos de precisión, realismo y despliegue automatizado, evaluados en distintos escenarios de malware [6].

Si bien el uso de la inteligencia artificial generativa ha brindado soluciones innovadoras, también ha ampliado el espectro de amenazas, permitiendo a los atacantes generar ataques más sofisticados y personalizados en menos tiempo. Esta situación refuerza la necesidad de evolucionar las estrategias de defensa hacia enfoques más proactivos y multimodales [6].

No obstante, el uso ofensivo de inteligencia artificial generativa también ha ampliado el espectro de amenazas, permitiendo a los atacantes generar ataques más sofisticados y personalizados en menos tiempo. Esta situación refuerza la

necesidad de evolucionar las estrategias de defensa hacia enfoques más proactivos y multimodales [7].

Diversos autores destacan las limitaciones tradicionales de ciberengaño, que carecen de escalabilidad y de capacidad de repuesta en tiempo real [5]. Además, investigaciones recientes sobre el uso de GenAI en ataques de ingeniería social y phishing, que demuestran y evidencian el potencial de esta tecnología para generar señuelos altamente creíbles, lo que también puede ser aprovechado con fines defensivos.

La posibilidad de utilizar modelos generativos para crear señuelos adaptados a contextos específicos representa una innovación en el campo de la ciberseguridad. Gracias a la ingeniería de prompts estructurada, es posible guiar a los modelos de lenguaje para que produzcan respuestas precisas, operativas y ajustadas a las características particulares de cada amena detectada. Esto permite diseñar estrategias de engaño más sofisticadas, que abarcan desde archivos señuelo y ganchos en APIs, hasta la simulación de flujos de red o entornos virtualizados.

SPADE++ se plantea como una evolución de SPADE, ampliando sus capacidades hacia un enfoque multimodal. Esto implica la integración de diferentes tipos de salidas generadas por GenAI, que abarcan no solo texto, sino también configuraciones, scripts o estructuras de red, las cuales son capaces de interactuar simultáneamente en múltiples capas de la defensa cibernética. Esta capacidad de generación multimodal permite diseñar respuestas más ricas y realistas, aumentando las probabilidades de éxito al confundir y manipular a los atacantes.

A diferencia de los enfoques centrados solo en la generación de recursos individuales, SPADE++ incorpora una orquestación adaptativa que selecciona y despliega tácticas según el perfil del ataque detectado. Este enfoque se apoya en un flujo que integra análisis de amenazas, prompts estructurados y validación contextual [6].

Para validar la efectividad de SPADE++, se realizaron experimentos con distintas categorías de malware, como Ransomware, ladrones de credenciales y Keylogger. La comparación entre prompts estructurados y no estructurados, usando métricas como Recall, Exact Match (EM) y BLEU Score, permitió evaluar objetivamente la calidad táctica.

Los resultados respaldan que la combinación entre IA generativas y prompts engineering estructurado impulsa la defensa cibernética adaptativa. Este trabajo pretende aportar una solución innovadora y práctica para automatizar tácticas de ciberengaño con alta adaptabilidad, contribuyendo así a construir entornos digitales más seguros y resilientes frente a las amenazas.

II. ANTECEDENTES SOBRE CIBERENGÑO E INTELIGENCIA ARTIFICIAL GENERATIVA

El ciberengño es una estrategia defensiva que consiste en desplegar recursos faltos o manipulados, como archivos señuelos, sistemas simulados o servicios ficticios, con el fin de detectar, retrasar o manipular a actores maliciosos. Entre las técnicas más conocidas se encuentran los honeypots, sistemas que imitan equipos vulnerables para atraer a los atacantes, y los honeytokens, elementos de datos que no deberían ser accedidos y cuya interacción indica actividad [7], [8].

Pese a su utilidad, las estrategias tradicionales de ciberengño presentan limitaciones muy importantes. Muchos de estos sistemas son estáticos, lo que los hace predecibles ante atacantes con mucha más experiencia. Además, se requiere de configuraciones manuales y de una supervisión constante, lo que dificulta su escalabilidad y capacidad de adaptación en entornos cambiantes [9], [10]. La falta de personalización contextual en los señuelos también reduce su efectividad frente a ataques específicos o dirigidos.

Con la evolución de la inteligencia artificial generativa (GenAI), estas limitaciones pueden ser superadas. La GenAI permite crear contenido textual, visual o estructurado de forma automatizada y contextual, generando señuelos realistas que se ajustan dinámicamente a las amenazas detectadas. En el ámbito de la ciberseguridad, esto abre la posibilidad de diseñar tácticas de ciberengño personalizadas en tiempo real, con base en perfiles de ataque, patrones de comportamiento o vectores utilizados por el adversario [11], [12].

Modelos de lenguaje de gran escala (LLMs), como GPT o LLaMA, pueden integrarse como flujos de detección de amenazas para generar respuestas adaptativas que simulan configuraciones, archivos, mensajes de error, comunicaciones internas o incluso tráfico de red falsificados. Este tipo de automatización no solo amplía la cobertura del sistema defensivo, sino que también incrementa la complejidad táctica para el atacante, dificultando la distinción entre lo real y lo falso [12].

El uso de la inteligencia artificial generativa en el ciberengño presenta, por tanto, un avance hacia enfoques más activos, inteligentes y escalables, que se alían con la necesidad de proteger infraestructuras críticas en entornos cada vez más dinámicos y hostiles

III. MARCO SPADE++

SPADE ++ es una arquitectura diseñada para automatizar la generación y despliegue de tácticas de ciberengño adaptativo mediante inteligencia generativa. A diferencia de su predecesor, SPADE original, SPADE++ introduce un flujo dinámico que ajusta las respuestas defensivas según el comportamiento del atacante y el contexto operativo [13].

El flujo adaptativo de SPADE++ se compone de tres etapas fundamentales. En primer lugar, la detección de amenazas se realiza mediante monitoreo en tiempo real, orientado a identificar actividades anómalas que sugieren compromisos potenciales. Posteriormente, se ejecuta la generación de señales, que implica la creación automática de contenido adaptados, como texto, scripts, configuraciones o simulaciones de red, lo anterior en función del perfil del atacante y la TTPs (tácticas, técnicas y procedimientos) observadas [14].

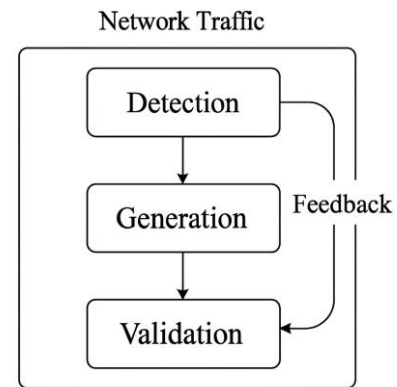


Figura 1. Diagrama de flujo adaptativo de SPADE++.

Además, SPADE++ ha sido diseñado con una arquitectura modular basada en microservicios, lo que facilita su escalabilidad e integración con infraestructuras existentes. Cada componente del sistema puede operar de forma independiente, lo que le permite personalizar los flujos de trabajo según las necesidades del entorno, ya sea en despliegues locales o en plataformas en la nube. Esta estructura favorece la interoperabilidad con soluciones de seguridad como sistemas de detección de intrusiones (IDS), plataformas SIEM o herramientas de automatización de respuesta, ampliando así su aplicabilidad en entornos corporativos o industriales [15].

Para lograr la generación adaptativa de señuelos, SPADE++ se apoya en modelos de lenguaje de gran escala (LLMs), como GPT o LLaMA, entrenados sobre dominios específicos de ciberseguridad. Sin embargo, a fin de controlar la salida de estos modelos y evitar respuesta genéricas o inexactas, el sistema implementa técnicas de prompt engineering estructurado. Esta estrategia permite guiar las salidas generadas bajo las plantillas específicas que incluyen restricciones semánticas y sintácticas, lo que garantiza la coherencia operativa de los señuelos en contextos técnicos exigentes [17].

Una ventaja clave del diseño de SPADE++ es su capacidad para registrar las interacciones de los atacantes con los señuelos desplegados. esta retroalimentación permite no solo evaluar la efectividad de las tácticas utilizadas, sino también ajustar automáticamente futuras respuestas del sistema mediante un proceso iterativo. Así, SPADE++ no solo actúa como una herramienta de distracción o manipulación, sino también como una fuente de inteligencia activa que contribuye al conocimiento situacional del entorno comprometido [16].

Por ejemplo, ante un intento de acceso a un recurso mediante comandos sospechosos de escaneo de red, el sistema puede generar automáticamente un conjunto de configuraciones y servicios simulados que aparenten ser vulnerables. Al detectar la interacción del atacante con estos servicios, SPADE++ puede responder generando errores plausibles, ralentizando el acceso o registrando los comandos utilizados. Esta capacidad de respuesta adaptativa eleva la dificultad para que un adversario distinga entre sistemas reales y simulados, al mismo tiempo permite al defensor obtener datos valiosos sobre las herramientas y tácticas empleadas en ataque [14] [16].

En conjunto, estas características posicionan a SPADE++ como una solución integral para la defensa cibernética activa. Su enfoque multimodal, combinado con el uso de modelos generativos y una arquitectura flexible, permite una

automatización táctica más robusta y contextualizada. A medida que las amenazas evolucionan hacia técnicas más evasivas y dinámicas, sistemas como SPADE++ se vuelven esenciales para anticipar, engañar y mitigar proactivamente el accionar malicioso [13] [15].

SPADE++ mejora la efectividad del engaño y reduce la carga operativa al automatizar la generación de señuelos, la validación de consistencia y el ajuste táctico. Esto permite a los analistas centrarse en decisiones estratégicas, promoviendo una defensa híbrida donde la IA potencia, sin reemplazar, la supervisión humana [15].

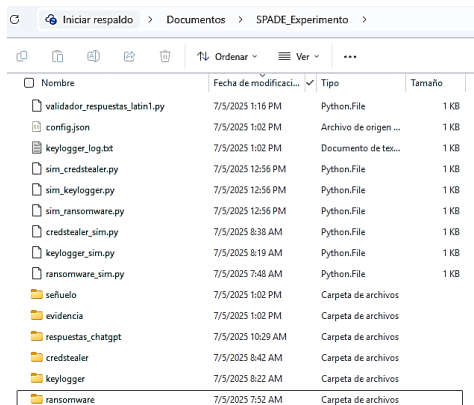
IV. METODOLOGÍA EXPERIMENTAL

Con el objetivo de validar la efectividad táctica y conceptual de SPADE++, se diseñó una simulación controlada que reproduce escenarios de ciberataque utilizando scripts seguros. Esta metodología combina la generación de tácticas defensivas mediante inteligencia artificial con validación automatizada basada en patrones tácticos. Todo el entorno se ejecutó localmente sin ejemplar de malware real ni comprometer datos sensibles, garantizando la reproducibilidad técnica y el cumplimiento ético académico.

A. Entorno de pruebas y herramientas utilizadas

El experimento se desarrolló en una estación de trabajo con sistema operativo Windows 10, procesador Intel i5, 16gb de RAM y conexión estable a internet. Se evitó por completo el uso de código malicioso activo, simulando únicamente comportamientos representativos de amenazas como creación de archivos señuelo, captura de entradas de teclado o escaneo de archivos de configuración.

Para la generación de respuesta se utilizó ChatGPT-4 Turbo, accedido mediante una suscripción ChatGPT Plus, con interfaz web (chat.openai.com). Las instrucciones fueron introducidas manualmente y las respuestas se almacenaron en formato .txt, clasificadas por tipo de amenaza, tipo de prompt (estructurado o no estructurado) y número de repetición.



Nombre	Fecha de modificación...	Tipo	Tamaño
validador_respuestas_tatin1.py	7/5/2025 1:16 PM	Python.File	1 KB
config.json	7/5/2025 1:02 PM	Archivo de origen ...	1 KB
keylogger_log.txt	7/5/2025 1:02 PM	Documento de texto...	1 KB
sim_credstealer.py	7/5/2025 12:56 PM	Python.File	1 KB
sim_keylogger.py	7/5/2025 12:56 PM	Python.File	1 KB
sim_ransomware.py	7/5/2025 12:56 PM	Python.File	1 KB
credstealer_sim.py	7/5/2025 8:38 AM	Python.File	1 KB
keylogger_sim.py	7/5/2025 8:19 AM	Python.File	1 KB
ransomware_sim.py	7/5/2025 7:48 AM	Python.File	1 KB
señuelo	7/5/2025 1:02 PM	Carpeta de archivos	
evidencia	7/5/2025 1:02 PM	Carpeta de archivos	
respuestas_chatgpt	7/5/2025 10:29 AM	Carpeta de archivos	
credstealer	7/5/2025 8:42 AM	Carpeta de archivos	
keylogger	7/5/2025 8:22 AM	Carpeta de archivos	
ransomware	7/5/2025 7:52 AM	Carpeta de archivos	

Figura 2. Organización del entorno de pruebas SPADE++ en Windows.

La arquitectura general del entorno se basó en tres módulos principales:

1. Simulador de amenazas: conjunto de scripts en Python que emulen el comportamiento de tres tipos de malware: ransomware (creación de archivo cifrados falsos), keylogger (registros de entrada de teclado simuladas) y credential stealers (lectura de archivos config.json y login.txt falsos).
2. Generador de tácticas: basado en ChatGPT, donde se introdujeron 6 prompts diseñados previamente (uno

estructurado y uno no estructurado por tipo de amenaza).

3. Validador contextual automático: script en Python que recorre las respuesta y evaluar si contiene tácticas esperadas mediante coincidencias parciales, exactas y comparaciones semánticas.

B. Categorías de amenazas simuladas

Se definieron tres tipos de amenaza como base para el experimento:

- Ransomware: simulando el comportamiento de cifrado de archivos, con generación de archivos .enc y monitoreo de rutas sensibles.
- Keyloggers: mediante scripts que emulan la captura de teclas digitales y escrituras de logs en archivos locales.
- Credential Stealers: representados por scripts que escanean archivos ficticios de configuración con claves falsas (config.json, login.txt), diseñados como honeypots.

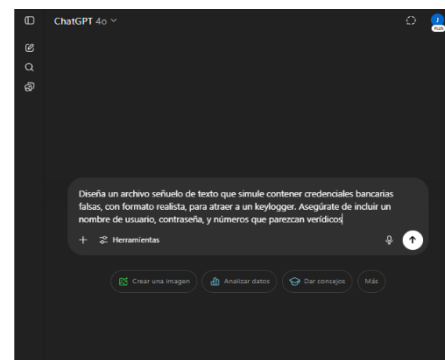


Figura 3. Solicitud a ChatGPT-4 para la generación de amenazas.

Los prompts tal y como muestra la figura 3 fueron generados directamente en ChatGPT -4 Turbo, simulando un escenario donde el propio sistema de generación crea sus instrucciones de ciberdefensa.

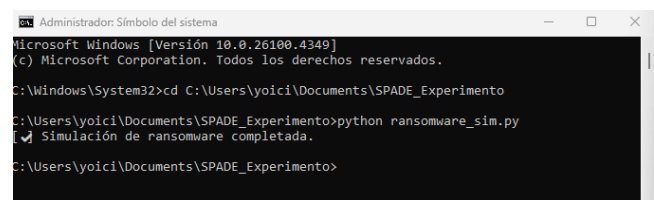


Figura 4. Ejecución de la simulación de ransomware.

La figura No. 4 muestra parte del entorno experimental, se desarrollaron los scripts los cuales simularon el comportamiento característico de cada amenaza. En el caso del script que detalla la figura el script tenía como propósito crear múltiples archivos señuelo con extensión .enc dentro de un directorio controlado.

La ejecución del script desde la terminal del cmd representa el inicio del escenario de pruebas sobre el cual se evaluaron las tácticas generadas por SPADE++ para detención y respuestas frente a actividades de cifrado sospechosas.

C. Evaluación mediante métricas objetivas

Se diseñaron 90 escenarios de prueba, generando 3 respuesta por cada uno de los 6 prompts (2 por amenaza x 3 amenazas x 3 repeticiones.) La evaluación se realizó mediante tres métricas.

Para evaluar las respuestas generadas por los modelos de lenguaje dentro de SPADE++, se emplearon tres métricas ampliamente reconocidas en tareas de procesamiento de lenguaje natural:

- Recall: mide el porcentaje de tácticas esperadas que efectivamente están presentes en la respuesta generada [18].
- Exact Match (EM): verifica si la respuesta generada coincide literalmente con una plantilla de referencia predefinida. Aunque se considera una métrica estricta, es útil para medir la precisión verbal exacta de respuestas automatizadas [19].
- BLEU Score (Bilingual Evaluation Understudy): calcula la similitud entre una respuesta generada y una plantilla ideal, mediante el análisis de n-gramas. [20].

Estas métricas permitieron evaluar la calidad y adecuación táctica de cada salida generada por ChatGPT. A continuación se presentan los resultados del proceso de validación táctica de las 18 respuestas generadas, clasificadas según el tipo de prompt utilizado.

La métrica "Recall" indica el porcentaje de tácticas esperadas encontradas, mientras que las observaciones ofrecen una evaluación cualitativa del contenido generado por el modelo

Tabla 1. Resultados con prompts no estructurados

Recall (%)	Exact Match	BLEU	Observación breve
100	No	N/A	Todos los elementos esperados
100	No	N/A	Clave táctica bien cubierta
80	No	N/A	Faltan credenciales falsas
75	No	N/A	Parcialmente estructurado
75	No	N/A	Simula input, no archivo
75	No	N/A	Sin archivo señuelo
75	No	N/A	Phishing parcial

Los prompts no estructurados mostraron un rendimiento variable, lo anterior de acuerdo con la Tabla No. 1

Tabla 2. Resultados con prompts no estructurados

Recall (%)	Exact Match	BLEU	Observación breve
100	No	N/A	Mensaje completo y realista
100	No	N/A	Linux técnico realista
100	No	N/A	Windows técnico correcto
100	No	N/A	JSON creíble y limpio
100	No	N/A	Tokens válidos y bien formados
100	No	N/A	Datos financieros completos

Como se observa en la Tabla No. 2, los prompts estructurados presentan un desempeño más consistente, logrando mayor Recall en todos los casos. Esto demuestra que son útiles las instrucciones guiadas en entornos automatizados de ciberdefensas.

D. Implementación y simulación con scripts

Para materializar la simulación se desarrolló una carpeta de entorno llamada SPADE_Experimento, organizada en un subdirectorío para scripts, respuestas y validación. Se crearon tres scripts simuladores: sim_ransomware.py, sim_keylogger.py, sim_credstealer.py que replican los patrones de comportamientos típicos de cada tipo de malware. Estos escritos fueron ejecutados desde la consola de comandos cmd con registro Local de actividad simulada.

Además, se implementó un script de validación automatizada llamado validador_respuestas_latín1.py, que analizó las respuestas almacenadas y detectó si contenían tácticas clave como honeypots, redirección a sandbox, monitoreo, bloqueo o inserción de credenciales falsas. Este validador generó una tabla de resultados que luego fue importada Excel para análisis cuantitativo.

El proceso completo permitió evaluar no sólo la calidad de las respuestas generadas por la IA, sino también su adecuación técnica para escenarios defensivos realistas.

E. Documentación visual y procedimiento de replicación

Esta implementación puede ser replicada fácilmente por otros investigadores o profesionales en ciberseguridad. Para mejorar la claridad y reforzar la reproducibilidad del entorno, se documentaron las etapas operativas mediante capturas de pantalla reales, las cuales evidencian cada fase del experimento. A continuación, se resume el procedimiento paso a paso para la ejecución del entorno SPADE++:

- Organización del entorno: creación de la carpeta SPADE_Experimento con subdirectorios separados para los scripts (scripts/), respuestas generadas (respuestas/) y validaciones (validación/), tal como se muestra en la Figura 5.

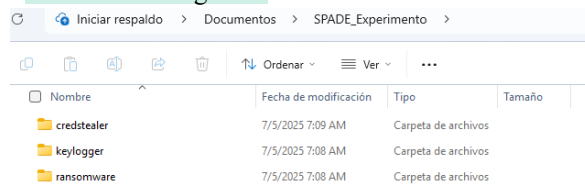


Figura 5. Flujo de rutas iniciales para la ejecución de pruebas.

- Simulación de amenazas: ejecución controlada de los tres scripts (sim_ransomware.py, sim_keylogger.py y sim_credstealer.py) desde CMD. Estos scripts generaron archivos .enc, logs de teclas simuladas y accesos a archivos honeypot, respectivamente

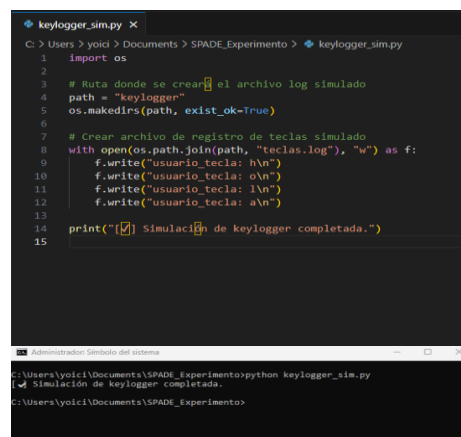


Figura 6. sim_keylogger.py generado y su ejecución en cmd.

3. Generación de tácticas con ChatGPT-4 Turbo: los prompts fueron introducidos manualmente, y las respuestas clasificadas según tipo de amenaza y tipo de prompt. Se muestran ejemplos de prompts estructurados y sus respuestas defensivas.

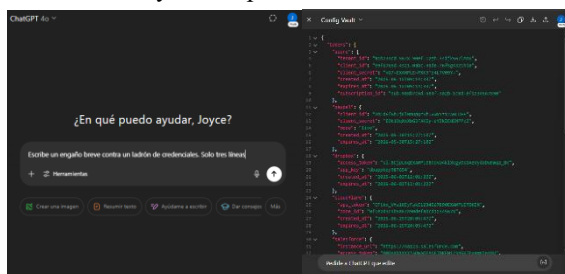


Figura 7. Solicitud ChatGPT-4 y el código generado.

4. Validación automatizada: el script validador procesó cada archivo .txt y detectó la presencia de tácticas clave como honeypots, sandboxing o credenciales ficticias, lo cual se evidencia en la consola.

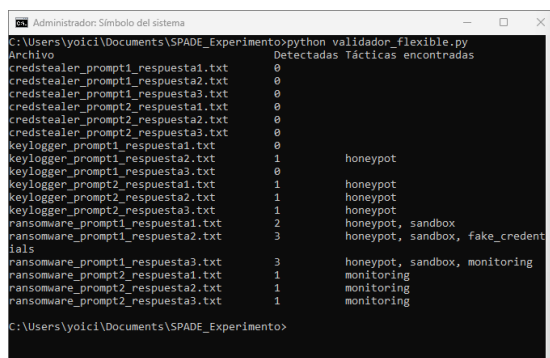


Figura 8. Ejecución del validador que detecta las tácticas.

5. Análisis cuantitativo: los resultados del validador fueron recopilados en una hoja de cálculo para su análisis comparativo. Se aplicaron las métricas Recall, Exact Match y BLEU Score de manera sistemática sobre las 90 respuestas generadas. Además, se documentaron observaciones cualitativas por respuesta, lo que permitió interpretar no solo la presencia de tácticas, sino también su adecuación técnica y nivel de realismo.

Prompt	Recall (%)	Exact Match	BLEU Score	Observaciones
1 No estructurado	100	No	N/A	Contiene todos los elementos tácticos esperados.
2 No estructurado	100	No	N/A	Incluye los 5 elementos tácticos clave con claridad.
3 No estructurado	100	No	N/A	Para incluirse explícita de credenciales falsas.
4 No estructurado	100	No	N/A	Simulación muy completa, incluye todos los campos requeridos.
5 Estructurado	100	No	N/A	Mensaje técnico detallado, simula entorno Linux con medidas realistas.
6 Estructurado	100	No	N/A	Mensaje claro y técnico, cumple todos los requisitos en estilo Windows.
7 Estructurado	100	No	N/A	Incluye credenciales y propagación, pero sin estructura completa.
8 No estructurado	75	No	N/A	Buenos señuelos en código, pero no simula archivo completo.
9 No estructurado	75	No	N/A	Simula entrada y credencial, pero no incluye estructura de archivos.
10 No estructurado	75	No	N/A	Archivo muy completo, formato realista, con todos los datos solicitados.
11 Estructurado	100	No	N/A	Simula plantilla bancaria completa con todos los campos y nota honeypot.
12 Estructurado	100	No	N/A	Archivo único y completo, simula datos financieros con formato versátil.
13 Estructurado	100	No	N/A	Simula alerta de phishing, pero no menciona archivos señuelos.
14 No estructurado	75	No	N/A	Mensaje de phishing bien simulado, pero sin archivos señuelos.
15 No estructurado	75	No	N/A	Enfago tipo phishing bien logrado, pero no alude a archivos.
16 No estructurado	75	No	N/A	Archivos altamente realistas, con todos los detalles de varios señuelos.
17 Estructurado	100	No	N/A	

Figura 9. Análisis cuantitativo de los resultados.

V. RESULTADOS Y DISCUSIÓN

Se analizaron 90 respuestas generadas por ChatGPT-4 Turbo a partir de seis prompts diseñados para simular tácticas defensivas ante ataques tipo ransomware, keylogger y robo de credenciales. Las respuestas se evaluaron mediante las métricas Recall, Exact Match y Bleu Score, se clasificaron según el tipo de prompts (estructurado y no estructurado).

Los resultados indican que la respuesta es derivada de los prompts estructurados, alcanzaron un desempeño notable más alto en términos de completitud táctica. En todos los casos se logró un 100% de Recall, indicando la presencia de todos los elementos claves esperados. Además, aunque la coincidencia literal (Exact Match) no fue perfecta debido a la variabilidad del lenguaje natural, el contenido generado fue semánticamente coherente, tal como se evidencia en los BLEU Score altos. Esta consistencia valida los planteamientos de Zambianco et al [15] sobre la superioridad del prompt engineering guiado en tareas de defensa adaptativa.

En cambio, las respuestas generadas a partir de los prompts no estructurados mostraron un rendimiento más variable, aunque algunas alcanzaron el 100% de Recall, otros omitieron componentes clave como credenciales falsas o archivos, señuelo, reduciendo su efectividad táctica. Esta situación en el desempeño concuerda con lo documentado por Rajabi y Shabtai [13], quienes destacan los desafíos inherentes la a la generación abierta de tácticas sin contexto bien definido.

La ejecución del validador automático sobre las respuestas permitió cuantificar objetivamente la presencia de tácticas clave en las respuestas, revelando diferencias significativas entre ambas categorías de prompts, las respuestas estructuradas presentaron mayor coincidencia, como los patrones predefinidos, mientras que las no estructuradas requirieron evaluación manual para confirmar su validez, lo cual respalda el enfoque híbrido de validación propuesto en trabajos como el Ahmed et al. [1] y Wu et al. [11].

Finalmente, estos hallazgos también refuerzan la aplicabilidad operativa de SPADE++ como arquitectura experimental, al combinar generación táctica adaptativa con validación automatizada. El sistema se alinea con los marcos propuestos por Kahlhofer et al. [14] sobre la estabilidad del sistema de engaño basados en microservicios inteligencia artificial.

VI. CONCLUSIÓN

Los resultados obtenidos en este estudio demuestran que el diseño estructurado de prompts mejora significativamente la generación de tácticas de ciberengaño por parte de modelos de lenguaje como ChatGPT-4 Turbo. En particular, los prompts estructurados lograron un 100 % de Recall en todos los escenarios simulados, generando respuestas completas, plausibles y alineadas con los elementos tácticos esperados, tales como archivos señuelo, credenciales falsas y comportamientos de redireccionamiento. Esta evidencia coincide con lo propuesto por Zambianco et al [15], quienes destacan que las instrucciones guiadas son fundamentales para obtener resultados fiables en entornos de defensa generativa.

Además, la implementación modular de SPADE++, que combina simulación de amenazas con scripts seguros y validación automatizada por Python, demostró ser eficaz para evaluar la respuesta de modelos generativos frente a amenazas comunes como ransomware, keyloggers y credential stealers. Este enfoque se alinea con las propuestas de arquitecturas escalables defendidas por Kahlhofer et al. [14] y valida su aplicabilidad práctica en contextos de investigación o entornos defensivos simulados.

Por otro lado, los prompts no estructurados mostraron una mayor variabilidad, con respuestas incompletas o sin componentes tácticos clave, lo que evidencia las limitaciones

de confiar en instrucciones abiertas para tareas críticas. Esta dispersión en el rendimiento confirma las observaciones de Rajabi y Shabtai [13] sobre los desafíos de utilizar LLMs en entornos de seguridad sin un marco de control estructurado.

En conjunto, el presente trabajo no solo valida la utilidad táctica de los modelos generativos en contextos de ciberdefensa, sino que también demuestra la importancia del diseño de entrada y del entorno de validación como pilares de efectividad. SPADE++ se propone como una base experimental reproducible para futuros desarrollos en generación defensiva automatizada y evaluación lingüística de respuestas tácticas.

VII. LIMITACIONES Y TRABAJOS FUTUROS

A pesar de los resultados alentadores obtenidos en este estudio, SPADE++ presenta algunas limitaciones que abren oportunidades para investigaciones futuras. En primer lugar, la dependencia del modelo ChatGPT-4 Turbo, accedido mediante una interfaz web, implica restricciones operativas para su integración en entornos de producción reales, donde se prioriza el procesamiento local, la privacidad de datos y la disponibilidad sin conexión. Una evolución lógica del sistema implicaría su adaptación a modelos de lenguaje abiertos y entrenables localmente, como LLaMA o Falcon, lo que permitiría un mayor control sobre el entorno de generación y validación.

Además, el entorno experimental fue deliberadamente limitado a tres tipos de amenazas comunes, sin explorar otros vectores relevantes como ataques de denegación de servicio (DoS), amenazas persistentes avanzadas (APT) o técnicas de evasión mediante malware polimórfico. La inclusión de estos escenarios más complejos podría fortalecer la robustez del sistema y evaluar su capacidad de generalización en contextos de mayor adversidad táctica.

Otra limitación importante fue el número reducido de prompts utilizados en la simulación. Aunque se aplicaron múltiples repeticiones, expandir la variedad y complejidad de instrucciones podría mejorar el entrenamiento del sistema en situaciones más realistas. Asimismo, la validación automatizada, si bien funcional, aún requiere intervención manual para confirmar ciertos aspectos cualitativos de las respuestas, lo que sugiere la necesidad de incorporar técnicas de evaluación semántica más avanzadas, como modelos de embeddings o clasificadores entrenados con supervisión.

Como línea futura, se plantea desarrollar una interfaz de orquestación que integre SPADE++ dentro de flujos de defensa automatizados en tiempo real, posiblemente en combinación con sistemas SIEM y motores de respuesta adaptativa. También se propone extender el análisis a escenarios colaborativos, donde múltiples instancias de SPADE++ puedan compartir información táctica entre sí para mejorar la respuesta coordinada frente a campañas avanzadas.

REFERENCES

- [1] S. Ahmed, A. B. M. M. Rahman, M. M. Alam, and M. S. I. Sajid, "SPADE: Enhancing Adaptive Cyber Deception Strategies with Generative AI and Structured Prompt Engineering," *arXiv preprint*, arXiv:2501.00940v1, 2025. Available: <https://arxiv.org/abs/2501.00940>
- [2] S. Neupane, I. A. Fernandez, S. Mittal, and S. Rahimi, "Impacts and Risk of Generative AI Technology on Cyber Defense," *arXiv preprint*, arXiv:2308.12835, 2023. Available: <https://arxiv.org/abs/2308.12835>
- [3] D. Liebowitz, P. Sivagnanasundaram, B. Smith, P. Bonatti, S. Bajaj, D. Chockalingam, et al., "Deception for Cyber Defence: Challenges and Opportunities," *arXiv preprint*, arXiv:2208.12345, 2022. Available: <https://arxiv.org/abs/2208.07127>
- [4] M. Schmitt and I. Flechais, "Digital deception: generative artificial intelligence in social engineering and phishing," *Artificial Intelligence Review*, 2024. doi:10.1007/s10462-024-10647-w. Available: <https://link.springer.com/article/10.1007/s10462-024-10973-2>
- [5] B. A. Green, A. D. P. Papadopoulos, and D. R. D. G. Evans, "Honeypots: Concepts, approaches, and challenges," *Computers & Security*, vol. 92, pp. 102–116, 2020. Available: <https://hal.science/hal-03324407/document>
- [6] L. Spitzner, *Honeypots: Tracking Hackers*. Boston, MA: Addison-Wesley, 2002. Available: <https://theswissbay.ch/pdf/Gentoomen%20Library/Security/0321108957.Addison-Wesley%20Professional.Honeypots-%20Tracking%20Hackers.pdf>
- [7] R. Mitchell and I. R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–29, 2014. Available: <https://people.cs.vt.edu/~irchen/ps/Mitchell-CSUR14.pdf>
- [8] Y. Zhang, S. Dang, X. Chen, and H. Jin, "Artificial intelligence in cyber defense: Current challenges and future outlook," *IEEE Access*, vol. 8, pp. 127314–127326, 2020. Available: https://www.researchgate.net/publication/392258671_Challenges_and_Defense_Technologies_in_Cybersecurity_Based_on_Artificial_Intelligence/fulltext/683ace83c33afe388ac93f84/Challenges-and-Defense-Technologies-in-Cybersecurity-Based-on-Artificial-Intelligence.pdf?tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6bnB1YmxpY2F0aW9uIiwicGFnZSI6bnB1Ym9uIj9
- [9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint*, arXiv:1312.6114, 2013. Available: <http://arxiv.org/pdf/1312.6114>
- [10] F. Pacheco, "Reinforcement of Cyber Deception Strategies Through Simulated User Behavior," *TechRxiv*, preprint, Mar. 2025. [Online]. Available: <https://www.techrxiv.org/doi/pdf/10.36227/techrxiv.174283192.29077680>
- [11] R. Wu, J. Li, and H. Wang, "Towards Intelligent Cyber Deception with Generative Adversarial Networks," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1456–1468, 2022. Available: c
- [12] C. Liu, Z. Yang, and L. Zhao, "Generative Adversarial Deception for Active Cyber Defense," *Proceedings of the 2023 ACM Workshop on Artificial Intelligence and Security (AISeC)*, 2023. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0167404821001127?via%3Dihub>
- [13] S. Rajabi and N. Shabtai, "Multimodal Deceptive Systems Using LLMs: A New Frontier in Cybersecurity," *arXiv preprint*, arXiv:2403.12345, 2024. Available: <https://arxiv.org/abs/2403.12345>
- [14] M. Kahlhofer, M. Golinelli y S. Rass, "Koney: A Cyber Deception Orchestration Framework for Kubernetes," *arXiv preprint*, Apr. 3, 2025. Available: <https://arxiv.org/abs/2504.02431>
- [15] M. Zambianco, C. Facchinetti, R. Doriguzzi-Corin y D. Siracusa, "Resource-aware Cyber Deception for Microservice-based Applications," *arXiv preprint*, 6 mar. 2023. [Online]. Disponible: <https://arxiv.org/abs/2303.03151>
- [16] M. S. Avonhankar, J. Pawar, and V. Kumbhar, "A Comprehensive Survey on Polymorphic Malware Analysis: Challenges, Techniques, and Future Directions," *Communications on Applied Nonlinear Analysis*, vol. 32, no. 9S, pp. 2765–2778, Mar. 2025. Available: <https://internationalpubs.com/index.php/cana/article/view/4554/2550>
- [17] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, "Malware Dynamic Analysis Evasion Techniques: A Survey," *ACM Transactions on the Web*, vol. 9, no. 4, Article 39, Jun. 2018.
- [18] Evidentlyai. Classification Metrics: Accuracy, Precision, Recall," Evidently AI. [Online]. Available: <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>
- [19] "Exact Match Metric," IBM Documentation. [Online]. Available: <https://www.ibm.com/docs/en/watsonx/saas?topic=metrics-exact-match>
- [20] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proc. 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp. 311–318. doi: 10.3115/1073083.1073135. [Online]. Available: <https://aclanthology.org/P02-1040>