
This is the **published version** of the bachelor thesis:

Olivella Bruguera, Pau. *Disseny, implementació i anàlisi de la capacitat tècnica d'un assistent d'IA local per a auditories de seguretat ofensiva..* Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334432>

under the terms of the  license.

Disseny, implementació i anàlisi de la capacitat tècnica d'un assistent d'IA local per a auditories de seguretat ofensiva.

Pau Olivella Bruguera

29 de juny de 2026

Resum– Aquest projecte presenta el disseny i la implementació de Nika, un assistent d'intel·ligència artificial local per a l'automatització d'auditories de seguretat ofensiva. El sistema analitza informació d'escaneig de xarxa per identificar serveis vulnerables i proporciona els codis d'explotació que permeten aprofitar aquestes vulnerabilitats. Mitjançant l'ús del model Llama 3.1 i una arquitectura de recuperació augmentada per generació (RAG), s'aplica un enfocament híbrid que combina la cerca vectorial amb filtres literals per garantir la màxima precisió en el reconeixement de versions. A més, s'integra una funció de cerca web en temps real per assegurar una millor detecció d'amenaques, garantint la privadesa de les dades en operar íntegrament en local. La validació en entorns controlats confirma la capacitat de l'eina per reduir significativament la càrrega de feina de l'auditor sense comprometre la confidencialitat de les dades.

Paraules clau– Automatització, Ciberseguretat ofensiva, Intel·ligència artificial local, LLM, Privadesa, RAG, Test de penetració, Anàlisi de vulnerabilitats

Abstract– This project presents the design and implementation of Nika, a local AI assistant for automating offensive security audits. The system analyzes network scan data to identify vulnerable services and provides the exploit code necessary to leverage these vulnerabilities. By utilizing Llama 3.1 model and a retrieval augmented generation (RAG) architecture, it employs a hybrid approach that combines vector search with literal filtering to ensure maximum precision in version recognition. Additionally, a real-time web search function is integrated to enhance threat detection, while ensuring data privacy by operating entirely locally. Validation in controlled environments confirms the tool's ability to significantly reduce the auditor's workload without compromising data confidentiality.

Keywords– Automation, Local Artificial Intelligence, LLM, Offensive Cybersecurity, Pentesting, Privacy, RAG, Vulnerability Analysis

1 CONTEXT

L'EVOLUCIÓ dels sistemes d'informació actuals es caracteritza per un avenç tecnològic constant i exponencial. No obstant això, aquesta digitalització accelerada comporta un increment paral·lel de la superfície d'exposició davant possibles ciberamenaces. Actualment, tant les petites com les grans organitzacions s'enfronten a un escenari de risc creixent, on la ciberdelinqüència s'ha professionalitzat fins esdevenir un dels principals reptes de seguretat global. A Espanya, durant el 2025, sense comptar atacs a administracions públiques, es van registrar un total de 122.223 incidents, això representa un augment del 26% respecte al 2024 [1].

En els darrers anys, la intel·ligència artificial (IA) ha emergit com una tecnologia disruptiva que ha transformat completament des del desenvolupament de programari i l'automatització de processos industrials fins a tasques quotidianes a nivell d'usuari. Aquesta evolució s'ha traslladat també al camp de la ciberseguretat, on la IA ha passat d'executar tasques més bàsiques, a esdevenir una eina omnipresent, cada cop més precisa, millorant significativament l'eficiència operativa. [2].

Dins de la ciberseguretat, i més concretament en l'àrea del *Pentesting* o les auditories de seguretat ofensiva, la IA està transformant la manera de treballar. Inicialment, s'emprava principalment en tasques automatitzades de baixa complexitat com podrien ser agilitzar la recerca d'informació pública i l'anàlisi ràpida de codi. Avui dia, però, la millora tècnica dels models ha estat dràstica, passant de

- E-mail de contacte: Pau.OlivellaB@autonoma.cat
- Menció: Tecnologies de la Informació
- Treball tutoritzat per: Enric Alibech Romero (DEIC)
- Curs 2025/26

ser assistents bàsics i una capacitat tècnica molt limitada a eines molt més complexes. Com es pot observar a la Figura 1, *benchmarks* de referència com *MMLU-Pro* documenten aquest salt qualitatiu, on s'ha passat de precisions del 43% en generacions anteriors a superar el 89% en les versions més recents.

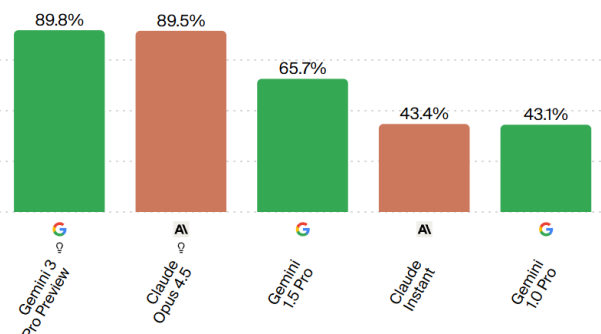


Fig. 1: Comparativa de rendiment i precisió en el Benchmark MMLU-Pro entre models més antics (2023) i més recents (2026) de Gemini i Claude Opus. (Font: adaptat de [3]).

Aquesta progressió permet ara generar scripts més complexos, analitzar ràpidament grans volums de dades i cercar informació crítica amb una fiabilitat superior, automatitzant processos que tradicionalment consumien gran part dels recursos temporals de l'auditor.

Aquesta evolució no només optimitza els recursos temporals de l'auditor en automatitzar processos mecànics, sinó que també aporta un component de creativitat al procés. La IA és capaç de suggerir vectors d'atac innovadors que podrien passar desapercebuts per a un humà per exemple per falta d'experiència o fatiga durant auditories extenses. Aquesta versatilitat es manifesta també en la creació d'entorns de prova personalitzats, on la IA facilita la generació de configuracions i simulacions adaptades específicament a la infraestructura i particularitats de cada objectiu [4].

Davant d'aquest escenari de ràpida evolució, aquest Treball de Final de Grau té l'objectiu d'explorar i analitzar la capacitat d'assistència tècnica actual de la IA local en auditories de seguretat ofensiva. Per fer-ho, es proposa el disseny i la implementació d'un assistent intel·ligent basat en dos pilars tecnològics clau:

- Implementació d'un LLM (*Large Language Model*) local: Per garantir la màxima privadesa i l'integritat de les dades tractades, evitant la filtració d'informació sensible cap a models comercials al núvol.
- Arquitectura RAG (*Retrieval-Augmented Generation*): Integració d'informació tècnica específica del sector per millorar la qualitat, precisió i context de les respostes generades per l'assistent.

2 MOTIVACIÓ DEL TREBALL

La motivació principal d'aquest projecte neix de la necessitat d'optimitzar els processos d'auditoria mitjançant la

delegació de tasques mecàniques a sistemes intel·ligents, permetent que l'auditor focalitzi els seus recursos en l'anàlisi i el criteri humà. En aquest sentit, la implementació d'una IA no s'ha d'entendre com un substitut del professional del *pentesting*, sinó com una evolució metodològica que permet treballar amb una precisió i celeritat superiors [5].

Així mateix, aquest treball respon a la preocupació creixent per la privadesa de les dades en l'ús de la IA generativa. La voluntat d'analitzar si un entorn local pot oferir un suport tècnic comparable a les solucions al núvol, sense comprometre la confidencialitat de la informació sensible de l'auditoria, constitueix un dels pilars centrals d'aquest projecte.

3 OBJECTIUS

L'objectiu principal d'aquest projecte és l'anàlisi i estudi de la capacitat d'optimització dels processos més mecànics de les auditories de seguretat ofensiva i la capacitat tècnica actual d'assistència de la IA en entorns locals. Per això es realitzarà el disseny, desenvolupament i desplegament d'un sistema d'assistència tècnica per tasques de ciberseguretat ofensiva, optimitzant fases com les de reconeixement i anàlisi de vulnerabilitats mitjançant IA generativa.

Per aquest projecte s'han definit els següents objectius:

3.1 Objectius generals (OG)

- OG1: Seleccionar, implementar i configurar un model de llenguatge extens (LLM) en entorn local, garantint la privadesa de les dades tractades durant les auditories.
- OG2: Explorar i prototipar la integració d'un agent de cerca web (OSINT) en temps real que permeti al model contrastar la seva base de coneixement amb informació externa actualitzada com poden ser noves amenaces, exploits, CVE, etc.
- OG3: Desenvolupar una arquitectura RAG (*retrieval-augmented generation*) mitjançant una base de dades vectorial per gestionar documentació tècnica.
- OG4: Validar l'eficàcia de l'assistent en un entorn de laboratori controlat, avaluant la precisió de les respostes generades en la identificació de vulnerabilitats i CVEs.
- OG5: Proposar millores i recomanacions basades en els resultats obtinguts per enfortir el futur desenvolupament d'eines d'IA aplicades a la seguretat ofensiva.

3.2 Objectius d'aprenentatge (OA)

- OA1: Comprendre en profunditat els principis fonamentals dels LLMs, les arquitectures RAG i el seu paper emergent en la ciberseguretat.
- OA2: Adquirir experiència pràctica en la configuració i optimització de models de llenguatge locals segons les limitacions de maquinari disponible.

- OA3: Desenvolupar habilitats en programació per la creació d'agents intel·ligents.
- OA4: Adquirir experiència en la transformació de documentació tècnica no estructurada en espais vectorials.
- OA5: Analitzar i assegurar els principis de privadesa i gestió de les dades en el desplegament de models de llenguatge, avaluant críticament els riscos d'exfiltració d'informació sensible en autories de seguretat en comparar implementacions locals i les solucions basades en el núvol.

4 METODOLOGIA I PLANIFICACIÓ

Pel desenvolupament d'aquest projecte i assoliment dels objectius plantejats, s'ha escollit una metodologia de desenvolupament seqüencial o en cascada (*Waterfall*), dividint el projecte en fases clarament diferenciades on cada una depèn de la validació de l'anterior. Aquest enfocament permet assegurar una bona planificació de les tasques a realitzar i la robustesa del nucli de l'assistent abans d'integrar funcionalitats més complexes.

El desenvolupament de la part tècnica del projecte s'estructurarà en les següents línies d'actuació:

- **Fase 1: Implementació del nucli LLM i RAG inicial:** La fase inicial del projecte que consta de la investigació i desplegament d'un model de llenguatge extens en local (com Llama3.1 o Mistral) i la seva integració amb la primera base de dades vectorial alimentada amb el *dataset* de *ExploitDB*.
- **Fase 2: Desenvolupament de l'agent de processament:** Seguidament, la creació d'un script d'orquestració que actuï com a pont entre l'usuari i la IA. Aquest agent tindrà la capacitat d'interpretar informació tècnica semi-estructurada com els outputs de la terminal de *Kali Linux* resultants d'un escaneig d'*Nmap* i decidir de manera autònoma quina informació ha de recuperar de la base de dades per identificar vulnerabilitats i *CVEs*.
- **Fase 3: Millora i Expansió:** En aquesta etapa, segons els resultats de la Fase 2, s'acabaran de millorar els resultats obtinguts, es plantejarà la necessitat d'ampliar les fonts de coneixement de l'agent i la possibilitat de l'addició d'un agent de cerca web per contrastar la base interna amb dades en temps real plantejat en l'OG2.
- **Fase 4: Validació i anàlisi:** Finalment, una fase final de proves intensives en entorns controlats per avaluar la precisió de l'assistent, realitzar millores iteratives i extreure les conclusions definitives del projecte.

Fase	Inici	Final	Durada
Fase 1	23/02	22/03	28 dies
Fase 2	23/03	26/04	35 dies
Fase 3	27/04	10/05	14 dies
Fase 4	11/05	31/05	21 dies

TAULA 1: PLANIFICACIÓ DEL PROJECTE

5 ESTAT DE L'ART

Aquest apartat estableix el marc conceptual i tecnològic necessari per comprendre la implementació de l'assistent intel·ligent del projecte. S'analitzen des dels fonaments de la ciberseguretat ofensiva fins als sistemes d'execució en entorns privats que permeten l'execució de models de llenguatge garantint la privadesa de les dades.

5.1 Fonaments de la ciberseguretat ofensiva

L'avaluació de la seguretat en els sistemes d'informació es fonamenta en la identificació proactiva de debilitats en un sistema *Software* o *Hardware* abans que puguin ser explotades per agents malintencionats [6].

5.1.1 Definició d'auditoria de seguretat i *pentesting*.

L'auditoria de seguretat ofensiva, coneguda tècnicament com a *pentesting* o proves de penetració, consisteix en l'execució d'atacs simulats i autoritzats contra una infraestructura amb l'objectiu d'avaluar-ne la resistència real. En aquest procés, l'auditor adopta el rol d'un adversari real i utilitza les mateixes tècniques i procediments que un ciberdelinqüent per detectar debilitats o vulnerabilitats en els sistemes. Aquesta pràctica és fonamental per a la identificació proactiva de fallades abans que puguin ser explotades malintencionadament, proporcionant la informació necessària perquè siguin corregides i es pugui, d'aquesta manera, mitigar el risc d'un atac real.

5.1.2 Fases del *pentesting*

Per estructurar l'auditoria ofensiva d'una manera rigorosa i repetible, generalment s'empra l'estàndard PTES (*penetration testing execution standard*) [10], el qual divideix el procés en set fases clarament definides:

1. **Interaccions prèvies al compromís (Pre-engagement interactions):** Es defineixen l'abast, els límits i els objectius de l'auditoria per dur a terme amb èxit la prova de penetració. Per exemple, s'acorda si es permeten atacs de denegació de servei (DoS) o si certes infraestructures crítiques queden fora de les proves.
2. **Recollida d'informació (Intelligence gathering):** Es realitza una cerca activa i passiva sobre l'objectiu per identificar la seva superfície d'exposició. Un exemple clau és l'ús d'*Nmap* per descobrir ports oberts i versions del programari en un servidor.
3. **Modelatge d'amenaçes (Threat modeling):** S'analitzen els actius de l'organització i els possibles vectors d'atac des de la perspectiva d'un adversari. Per exemple, es valora si la base de dades és l'objectiu més vulnerable o si ho és el servidor web.
4. **Anàlisi de vulnerabilitats (Vulnerability analysis):** Es busquen fallades de seguretat en els serveis detectats prèviament. Un exemple seria contrastar una versió específica d'un servidor amb la base de dades de *CVEs* per trobar vulnerabilitats conegudes.

5. **Explotació (Exploitation):** Es valida l'existència de la vulnerabilitat mitjançant l'execució d'un *exploit*. Per exemple, l'ús d'un *script* d'ExploitDB per obtenir un accés no autoritzat a una terminal remota.
6. **Post-explotació (Post-exploitation):** Es valora el valor dels sistemes compromesos i la capacitat de mantenir el control. Un exemple és l'intent d'escalada de privilegis des d'un usuari estàndard cap a un rol d'administrador (root).
7. **Informes (Reporting):** Es documenta tot el procés, incloent-hi les tècniques utilitzades i les recomanacions de mitigació. Un informe típic inclouria el detall tècnic de com s'ha compromès el sistema i el nivell de risc associat.

L'assistent desenvolupat en aquest treball optimitza la transició entre fases, facilitant la correlació entre les dades brutes de la fase de Recollida d'informació de l'escaneig de ports i serveis, i la identificació d'amenaques específiques a la fase d'Anàlisi de vulnerabilitats, proporcionant finalment el codi de suport necessari per a la fase d'Explotació.

5.2 Models de llenguatge extensos en entorns locals

5.2.1 Models de llenguatge extensos (LLM)

Els LLM o Models de Llenguatge Extensos, són sistemes d'intel·ligència artificial basats en xarxes neuronals profundes que han estat entrenats amb volums massius de dades per processar i generar text [11]. La seva arquitectura es fonamenta en el Transformer, un model que utilitza mecanismes per comprendre les relacions i el context dins d'una seqüència de paraules, permetent analitzar la informació de manera paral·lela i eficient [12].

A nivell tècnic, el rendiment i la capacitat d'un LLM estan definits pels seus paràmetres, que són les variables internes que el model ajusta durant l'entrenament per aprendre patrons complexos en les dades [11]. Aquests models es caracteritzen per la seva habilitat per extreure informació rellevant de fonts no estructurades, resumir dades tècniques i predir la següent seqüència lògica en un flux de treball. Aquesta capacitat de processament permet que els LLM puguin ser adaptats a entorns especialitzats, com en el cas d'aquest projecte, la ciberseguretat, mitjançant l'ús de prompts específics, facilitant la transició de dades brutes a formats estructurats i interpretables [11], [13].

Els models de llenguatge extensos representen el nucli tecnològic de l'assistent del projecte. No obstant això, la seva implementació en l'àmbit de la ciberseguretat ofensiva requereix un canvi de paradigma des de les solucions basades en el núvol cap a l'execució local per motius de seguretat i privacitat.

5.2.2 Privadesa vs. Núvol (Cloud)

L'ús de models comercials basats en el núvol com GPT o Claude Opus implica l'enviament de dades mitjançant APIs cap a servidors de tercers gestionats pel proveïdor. En una auditoria de seguretat, la informació tractada, que inclou

adreces IP, estructura de la xarxa, codi font vulnerable, entre d'altres, és extremadament sensible. L'exfiltració d'aquestes dades cap a entorns externs trenca de manera directa la cadena de custòdia de la informació, fet que suposa un greu risc de confidencialitat, atès que les plataformes comercials sovint emmagatzemen les entrades dels usuaris per reentrenar els seus models, la informació podria quedar incorporada en el sistema, obrint la porta a que actors maliciosos puguin reconstruir o inferir detalls crítics de la infraestructura mitjançant consultes especialitzades [14]. D'altra banda, el desplegament local garanteix un control total sobre les dades, assegurant que cap informació tècnica surti de l'entorn controlat de l'auditor [15]. Com veiem en la Figura 2, hi ha una tendència d'interès en augment cap als models LLM locals.

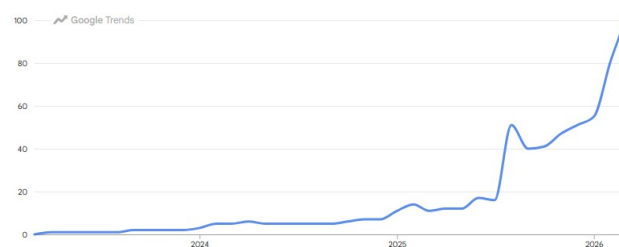


Fig. 2: Evolució de l'interès de cerca per al terme "Local LLM" entre 2023 i 2026. (Font: Extret de Google Trends [16]).

Això és degut a les recents optimitzacions que han reduït dràsticament els requisits del maquinari, fent molt més viable l'execució d'aquests models en equips amb recursos computacionals molt més limitats que els requerits anteriorment.

5.2.3 Optimitzacions per a maquinari limitat

L'execució de LLMs en maquinari de consum, com la targeta de vídeo *NVIDIA RTX 3060Ti* utilitzada en aquest projecte, requereix tècniques d'optimització avançades per superar les limitacions de memòria (VRAM). Tot i que el model podria carregar-se en la memòria RAM del sistema, on avui dia l'estàndard ja sol estar entre 16 GB i 32 GB [17], l'execució seria extremadament lenta a causa del coll d'ampolla del bus de dades entre la CPU i la GPU.

Per obtenir una velocitat de resposta funcional, és necessari que el model es carregui íntegrament en la memòria VRAM aprofitant així la gran capacitat de processament paral·lel de la gràfica [18]. D'altra banda, les targetes gràfiques solen tenir una capacitat de memòria inferior, en el cas d'aquest projecte, disposem de 8 GB de VRAM en comparació amb els 32 GB de RAM. Aquesta diferència ens obliga a aplicar diverses estratègies d'optimització per ajustar la càrrega del model als límits físics del maquinari. Tot i l'existència de diversos processos i tecnologies d'optimització que permeten l'execució local, còmoda i eficient de models LLM, aquest apartat es focalitzarà principalment en el mètode més determinant per a la viabilitat tècnica en maquinari de consum: la quantització, així com en l'entorn necessari per a la seva implementació.

La quantització d'acord amb [19], consisteix a reduir la precisió numèrica dels paràmetres del model, passant de formats d'alta precisió com FP32 (coma flotant de 32 bits)

a formats de menor resolució com INT8 (enters de 8 bits) o fins i tot INT4 (enters de 4 bits).

L'avantatge principal per aquest projecte és l'eficiència en l'ús de la memòria de vídeo (VRAM).

- **Reducció d'espai:** Un model de 7.000 milions de paràmetres en FP32 ocuparia uns 28 GB de VRAM. En quantitzar-lo a 4 bits, el seu pes es redueix a uns 4-5 GB amb una pèrdua de qualitat/precisió acceptable d'un 5-15% [20], permetent que s'executi en una targeta de vídeo com la utilitzada en aquest projecte.
- **Velocitat d'inferència:** Les operacions amb valors enters (INT) són molt més ràpids que amb coma flotant (Float Point), reduint el temps de resposta de l'assistent. En l'exemple anterior d'optimització, passant de FP32 a INT4 obtindríem una millora d'inferència aproximada d'entre 3 i 4 vegades més ràpid [20].
- **Consum energètic:** En requerir menys esforç computacional, el consum elèctric i la calor generada pel maquinari es veuen reduïts.

Per complementar aquest procés, l'arquitectura d'aquest projecte es recolza en el motor d'inferència Ollama i el format de fitxer GGUF. Aquesta combinació no només facilita una integració molt directa de l'LLM dins del codi de l'assistent, sinó que aporta una gran estabilitat operativa davant les limitacions físiques del maquinari de consum. El seu avantatge tècnic principal és la capacitat de realitzar una descàrrega híbrida de memòria, si el context de l'escaneig arriba a superar la capacitat de la VRAM de la targeta gràfica, el motor delega de forma automàtica el fragment excendent a la memòria RAM i a la CPU. Tot i que aquesta transferència redueix notablement la velocitat d'inferència, s'evita una excepció crítica per falta de memòria (*Out of Memory*) que aturaria l'execució de l'aplicació.

6 IMPLEMENTACIÓ I ARQUITECTURA

En aquest apartat es detalla el disseny de l'assistent *Nika*. Una eina orquestrada en Python que integra models de llenguatge extensos (LLM) locals, recuperació augmentada per generació (RAG) i capacitats OSINT per a l'automatització intel·ligent de l'anàlisi de vulnerabilitats.

6.1 Arquitectura general del sistema

L'arquitectura de l'assistent ha sigut creada com un sistema d'orquestració modular on cada component s'encarrega d'una tasca específica en la gestió del flux de dades. El sistema no actua com una única consulta estàndard, sinó que els diferents mòduls iteren repetidament fins obtenir resultats per tots els serveis consultats, fins arribar al menú final on se li exposen tots les evidències recopilades a l'auditor amb la possibilitat de proporcionar els *exploits* per cada una de les potencials vulnerabilitats detectades.

Aquesta estructura es divideix en quatre fases principals que defineixen el cicle d'anàlisi de l'script:

1. **Fase d'interfície i processament:** Encarregada de la recepció de dades de l'usuari i la seva estructuració intel·ligent.

2. **Fase de recuperació híbrida i OSINT:** Cerca simultània a la base de dades local d'ExploitDB i enriquiment dels resultats mitjançant cerques web en temps real.
3. **Fase de filtratge i rànkig final:** Ús del LLM per validar la rellevància de cada vulnerabilitat i assignar nivells de confiança.
4. **Fase del menú d'interacció:** Interfície final de selecció i recuperació d'scripts d'exploitació per a l'auditor.

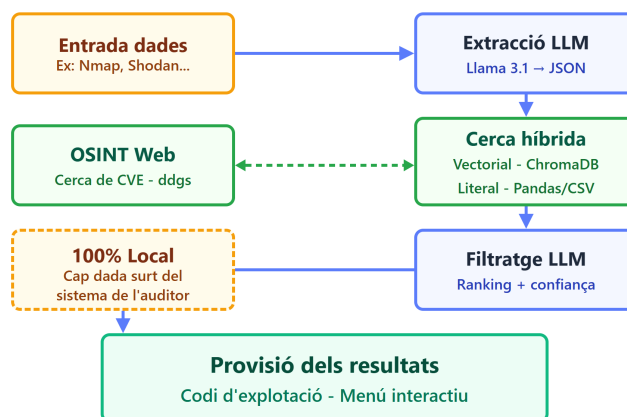


Fig. 3: Representació esquemàtica d'execució de l'assistent Nika. (Font: Elaboració pròpia).

Aquesta organització modular permet separar la lògica de cerca de la lògica de decisió, garantint que el sistema treballi sempre amb un context filtrat i d'alta rellevància, facilitant el manteniment i l'escalabilitat del projecte.

6.2 Selecció del model de llenguatge

S'ha seleccionat el model Llama 3.1 [21] en la versió de 8.000 milions de paràmetres com a nucli central de l'assistent. Aquesta elecció es basa en l'equilibri òptim que ofereix entre la capacitat de raonament, finestra de context i rendiment en entorns locals després de la quantització del model. Gràcies a aquesta tècnica de quantització, permet reduir els requeriments de la memòria VRAM, sense sacrificar dràsticament la precisió necessària per a les tasques d'anàlisi i filtratge necessàries per aquest projecte, podent així executar-lo en maquinari de consum no especialitzat mantenint la privadesa de les dades tractades.

6.3 Processament d'entrada i estructuració

Un dels reptes principals en l'automatització de les fases del *pentesting* que cobreix aquest assistent, és la variabilitat dels outputs que generen eines com Nmap, que sovint contenen informació redundant, metadades irrellevants o formats no estructurats. Per no dependre d'una estructuració determinada a l'hora d'enviar les dades on es troben els serveis i versions a analitzar, el sistema implementa una fase de processament intel·ligent utilitzant les capacitats del model de llenguatge prèviament definit. D'aquesta manera, s'aconsegueix que l'assistent pugui rebre tant el resultat d'un escaneig d'Nmap com un de Shodan o, fins i tot, l'entrada manual de les dades.

En aquesta etapa, el text brut de dades s'envia al model Llama 3.1 mitjançant un *system prompt* altament específic que

l'instrueix per actuar com un extractor de dades determinista. El model identifica els camps crítics per l'anàlisi i els encapsula en un objecte estructurat en format JSON.

Aquesta estructuració és fonamental per a l'eficàcia del sistema, ja que permet:

- **Neteja de dades:** S'eliminen capçaleres i dades redundants que no aporten valor real a la cerca de vulnerabilitats.
- **Precisió en la cerca:** Les consultes posteriors a ExploitDB i les cerques web es realitzen utilitzant exactament els termes de servei i versió detectats, minimitzant el soroll en la fase de recuperació.
- **Simplicitat en la lògica del codi:** El format JSON facilita la gestió d'errors, la traçabilitat de la informació i l'escalabilitat del sistema davant noves funcionalitats.

6.4 Sistema de recuperació híbrida i OSINT

Aquesta fase constitueix el nucli de coneixement de l'assistent, on es combina la informació emmagatzemada localment amb capacitats de cerca externa en temps real. Per garantir que l'auditor rebi els resultats més precisos i actualitzats, el sistema utilitza un enfocament de recuperació en tres nivells que s'executa de manera iterativa per a cada servei identificat.

6.4.1 Sistema de recuperació híbrida: cerca vectorial i literal

La implementació inicial del sistema de recuperació es basa en el paradigma RAG, utilitzant la base de dades vectorial ChromaDB i el model d'embeddings nomic-embed-text [22] per identificar candidats per proximitat contextual en un espai multidimensional. No obstant això, durant el desenvolupament es van detectar imprecisions en el raonament de versions numèriques (p. ex. la confusió entre les versions 1.2.4 i 2.4.4). Per solucionar-ho, s'ha integrat una capa de cerca literal i estructural mitjançant la llibreria Pandas sobre un fitxer CSV amb l'estructura completa de la base de dades. Aquesta combinació híbrida realitza un filtratge estricte per paraules clau que prioritza la correspondència exacta del programari analitzat, garantint una identificació de vulnerabilitats molt més robusta.

6.4.2 Enriquiment OSINT

Aquest mòdul actua com un mecanisme de redundància mitjançant la llibreria ddgs [23] una solució d'extracció de dades que s'executa localment sense dependre d'infraestructures centrals. Aquesta arquitectura elimina costos operatius, dificulta el bloqueig del servei i garanteix l'anonimat de les consultes en no generar registres en servidors intermediaris. L'agent realitza cerques web per extreure identificadors CVE rellevants esmentats en publicacions de seguretat recents per al servei analitzat. Un cop obtinguts, el sistema fa una nova cerca a la base de dades local utilitzant aquests identificadors com a clau única. Aquest procés permet rescatar vulnerabilitats que, per inconsistències en la seva nomenclatura o descripció original, podrien no haver estat detectades pels mètodes de cerca previs.

6.5 Filtratge intel·ligent, rànquing i interacció

L'última etapa del cicle d'anàlisi és la que transforma el conjunt de dades recuperades en els resultats mostrats finalment a l'auditor de seguretat.

6.5.1 Filtratge i rànquing final amb LLM

Un cop consolidada la llista dels millors candidats, el sistema delega la responsabilitat de la validació final al model de llenguatge extens. En aquesta etapa, el model no actua com un cercador, sinó com un expert en seguretat que avalua la coherència tècnica de cada exploit respecte a l'objectiu real. L'assistent realitza una verificació lògica de versions per determinar l'explotabilitat del programari detectat. Aquest anàlisi permet, per exemple, discernir si un servei amb la versió 2.3.4 es veu afectat per un exploit definit per a versions < 2.4 , una distinció que els mètodes basats exclusivament en similitud de vectors no poden garantir. El LLM avalua cada candidat mitjançant una assignació de confiança (alt, mitjà o baix) fonamentada en la versió, la plataforma i el tipus de vulnerabilitat. Finalment, el sistema genera un rànquing refinat on s'eliminen els falsos positius i es destaquen aquells vectors d'atac amb una viabilitat superior.

6.5.2 Mode interactiu i recuperació d'scripts

La darrera fase del projecte es centra en la presentació de resultats i la facilitat d'accés a la informació per a l'auditor. L'assistent exposa un menú interactiu a la terminal on es resumeixen les vulnerabilitats trobades organitzades per port i servei. Aquesta interfície permet a l'usuari realitzar les següents accions:

- **Selecció i provisió d'exploits:** Permet a l'auditor seleccionar una vulnerabilitat específica a partir dels resultats obtinguts per tal que l'assistent li proporcioni el codi de l'*exploit* corresponent.
- **Resum de vulnerabilitats:** L'assistent genera una vista simplificada i categoritzada de totes les vulnerabilitats detectades en els diferents ports i serveis. Aquesta funció permet a l'usuari tenir una visió global, ràpida i estructurada de l'estat de seguretat de l'objectiu sense haver de revisar tota la informació detallada.
- **Reimpressió detallada de l'anàlisi:** Per a una revisió profunda, el sistema permet tornar a visualitzar la totalitat dels resultats amb tot el detall tècnic recuperat assegurant que l'auditor tingui accés a tot el context necessari en qualsevol moment.

7 DINÀMICA D'EXECUCIÓ DEL SISTEMA

En aquest apartat s'il·lustra el flux d'ús de l'assistent Nika. Aquest recorregut visual permet comprendre com interactuen les diferents capes de l'arquitectura des de la perspectiva de l'auditor, des del primer contacte amb la interfície fins a l'obtenció del recurs d'explotació final.

7.1 Inicialització del sistema i entrada de dades d'auditoria

El procés s'inicia amb l'execució de l'eina. Un cop completada la càrrega dels mòduls d'intel·ligència i les bases de dades, es presenta la pantalla de benvinguda. Aquesta interfície inclou un logotip en format *ASCII Art* i una guia ràpida amb els paràmetres necessaris per començar l'anàlisi de serveis, tal com es mostra a la figura 4.



Automated Vulnerability Identification & Exploit Retrieval System

```
[i] INPUT INSTRUCTIONS:
Please paste the target scan data like:
- Nmap outputs (e.g., '21/tcp open ftp vsftpd 2.3.4')
- Shodan data
- Manual input (e.g., 'Apache 2.4.41...')

[!] Ensure the SOFTWARE NAME and VERSION are visible for better results.

[>] Paste your data and press ENTER Three times to start the analysis:

Starting Nmap 7.94 ( https://nmap.org ) at 2026-04-17 11:23 CET
Nmap scan report for target-host (192.168.56.101)
Host is up (0.00045s latency).
Not shown: 997 closed tcp ports

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.4
80/tcp    open  http     Apache httpd 2.4.17 ((Ubuntu))
8080/tcp  open  http     HttpFileServer httpd 2.3

Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 11.23 seconds
```

Fig. 4: Interfície de línia de comandes de Nika: Menú principal i recepció de dades d'escaneig. (Font: Elaboració pròpia).

El sistema està dissenyat per ser independent de la font d'entrada, l'auditor pot introduir dades en brut manualment o resultats provinents d'eines d'escaneig consolidades com Nmap, Masscan o Shodan, iniciant immediatament el processament intel·ligent de la informació.

7.2 Cicle d'anàlisi automatitzat i recuperació de vulnerabilitats

Un cop s'inicia el procés, l'assistent treballa de manera seqüencial per a cada servei detectat. Durant aquesta fase, s'informa en temps real sobre la cerca d'identificadors CVE mitjançant el mòdul OSINT i es presenten els resultats de la recuperació híbrida.

```
[21] vsftpd 2.3.4

[OSINT] Searching CVEs for vsftpd 2.3.4...
[OSINT] Potential relevant CVEs found: ['cve-2011-2523']

[1] Confidence: HIGH | Type: RCE | ID: 49757
Title : vsftpd 2.3.4 - Backdoor Command Execution
Script : exploits/unix/remote/49757.py
[2] Confidence: HIGH | Type: RCE | ID: 17491
Title : vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
Script : exploits/unix/remote/17491.rb
[3] Confidence: MEDIUM | Type: DoS | ID: 16270
Title : vsftpd 2.3.2 - Denial of Service
Script : exploits/linux/dos/16270.c
```

Fig. 5: Flux de treball actiu de l'assistent durant la identificació de vulnerabilitats. (Font: Elaboració pròpia).

En finalitzar l'anàlisi de tots els serveis, l'eina genera un resum consolidat de les vulnerabilitats potencials. Aquesta vista, il·lustrada a la Figura 6, proporciona la informació crítica de cada element trobat perquè l'auditor pugui avaluar

amb criteri la rellevància i l'impacte de cada vector d'atac identificat.

```
=====
SUMMARY - Type the exploit ID you want to view
=====

Port 21 for vsftpd:
ID 49757 [HIGH] RCE vsftpd 2.3.4 - Backdoor Command Execution
ID 17491 [HIGH] RCE vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)
ID 16270 [MEDIUM] DoS vsftpd 2.3.2 - Denial of Service

Port 80 for Apache:
ID 46676 [HIGH] LPE Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrot
ID 39037 [MEDIUM] DoS Apache 2.4.17 - Denial of Service

Port 8080 for HttpFileServer:
ID 49125 [HIGH] RCE Rejetto HttpFileServer 2.3.x - Remote Command Executio
ID 34668 [HIGH] RCE Rejetto HTTP File Server (HFS) 2.3.x - Remote Command
ID 39161 [HIGH] RCE Rejetto HTTP File Server (HFS) 2.3.x - Remote Command
ID 38859 [MEDIUM] Arbitrary File Upload Rejetto HTTP File Server (HFS) 2.2/2.3 - Arbi
ID 34852 [MEDIUM] RCE Rejetto HTTP File Server (HFS) 2.3a/2.3b/2.3c - Remote
ID 49584 [MEDIUM] RCE HFS (HTTP File Server) 2.3.x - Remote Command Executio
ID 52102 [LOW] RCE Rejetto HTTP File Server 2.3m - Remote Code Execution
=====
```

Fig. 6: Output estructurat de l'assistent per les potencials vulnerabilitats trobades. (Font: Elaboració pròpia).

7.3 Interfície d'interacció final i recuperació d'scripts d'exploitació

Finalment, el sistema ofereix un menú interactiu que permet a l'auditor passar de l'anàlisi a la possible fase d'exploitació. L'usuari pot seleccionar qualsevol vulnerabilitat d'interès i sol·licitar el codi font de l'*exploit* corresponent introduint el seu identificador (ID) a la consola.

```
Commands: <ID> view exploit | 'r' summary | 'R' detailed original | 'q' quit
> 49125

Exploit ID : 49125
Title : Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
Type : RCE | Confidence: HIGH
Path : exploits/windows/webapps/49125.py

START OF SCRIPT

# Exploit Title: Rejetto HttpFileServer 2.3.x - Remote Command Execution (3)
# Google Dork: intext:"httpfileserver 2.3"
# Date: 28-11-2020
# Remote: Yes
# Exploit Author: Åscar Andreu
# Vendor Homepage: http://rejetto.com/
# Software Link: http://sourceforge.net/projects/hfs/
# Version: 2.3.x
# Tested on: Windows Server 2008 , Windows 8, Windows 7
# CVE : CVE-2014-6287

#!/usr/bin/python3

# Usage : python3 Exploit.py <RHOST> <Target RPORT> <Command>
# Example: python3 HttpFileServer_2.3.x_rce.py 10.10.10.8 80 "c:\windows\SysNati
ive\WindowsPowershell\1.0\powershell.exe IEX (New-Object Net.WebClient).Downlo
adString('http://10.10.14.4/shells/mini-reverse.ps1')"

import urllib3
import sys
import urllib.parse

try:
    http = urllib3.PoolManager()
    url = f'http://{sys.argv[1]}:{sys.argv[2]}?search=%00{{.+.exec|{urllib.
parse.quote(sys.argv[3])}}}'
    print(url)
    response = http.request('GET', url)

except Exception as ex:
    print("Usage: python3 HttpFileServer_2.3.x_rce.py RHOST RPORT command")
    print(ex)

END OF SCRIPT
```

Fig. 7: Visualització del codi font de l'*exploit* sol·licitat mitjançant el menú interactiu. (Font: Elaboració pròpia).

Com s'aprecia a la Figura 7, el sistema recupera el recurs directament del repositori local i el mostra per pantalla. A més, el menú inclou funcionalitats per optimitzar l'experiència d'usuari, com l'opció de generar resums ràpids de totes les possibles vulnerabilitats trobades, tornar a llistar els resultats detallats o finalitzar l'execució de l'eina.

8 PROVES DE VALIDACIÓ

En aquest apartat es presenten les proves de validació del projecte, l'objectiu és avaluar la capacitat tècnica de l'assis-

tent Nika en la identificació de vulnerabilitats i la provisió de recursos d'exploració.

8.1 Configuració del laboratori

El disseny de l'escenari de laboratori s'ha estructurat mitjançant la virtualització per garantir un entorn controlat, segur i fàcilment replicable per a l'execució de les diverses proves de concepte. La infraestructura principal es basa en una màquina virtual gestionada amb el programari *VMware*, la qual allotja una distribució d'*Ubuntu Server* que actua com a màquina objectiu per al desplegament de la infraestructura vulnerable. Aquesta configuració permet simular la superfície d'exposició d'un servidor real en un entorn aïllat, assegurant que les activitats d'auditoria no interfereixin amb altres sistemes o xarxes externes.

Per a la realització de les tasques d'auditoria ofensiva, s'utilitza una estació de treball independent equipada amb la distribució *Kali Linux*, la qual actua com l'origen dels atacs simulats i les consultes de reconeixement. El flux de treball s'inicia realitzant un escaneig de ports i serveis des d'aquesta màquina externa a la màquina objectiu per obtenir les dades brutes que serviran d'entrada a l'assistent.

8.2 Infraestructura de vulnerabilitats

Per la creació dels diferents escenaris del laboratori, s'ha integrat l'eina *Vulnlab*, un projecte de codi obert dissenyat per al desplegament ràpid d'entorns vulnerables. El nucli d'aquesta eina es basa en la tecnologia de contenidors *Docker*, la qual permet instanciar serveis i aplicacions amb configuracions de seguretat febles o versions de programari amb vulnerabilitats conegudes. Aquesta modularitat és un factor determinant per a la validació del projecte, ja que permet activar i desactivar contenidors específics de manera ràpida i simple, segons la necessitat de cada prova, garantint també que cada escenari d'auditoria sigui independent i lliure de residus de proves anteriors.

8.3 Escenaris de prova

L'avaluació de l'assistent s'ha estructurat en diferents situacions de prova dissenyades per comprovar la capacitat i robustesa de l'arquitectura davant d'entorns reals. El primer grup de proves es focalitza en verificar la versatilitat de l'eina davant de tecnologies que no s'havien utilitzat durant la fase de desenvolupament. Mitjançant la combinació de serveis i aplicacions utilitzats habitualment, com ara instàncies de *MySQL*, *Tomcat* o servidors *Apache HTTP*, s'avalua si l'orquestrador és capaç de mantenir la mateixa taxa d'encert i consistència sense patir una degradació en el rendiment.

Un altre escenari crític tracta el comportament de la lògica de decisió i del rànquing intel·ligent quan les dades recollides per l'escaneig de xarxa presenten ambigüitats o informació incompleta, com ara la total absència d'un número de versió en un servei obert com *Redis* o *Oracle WebLogic Server*. Aquestes proves són essencials per mesurar l'eficàcia de la redundància implementada a través de la cerca web en temps real, determinant si l'assistent és capaç d'analitzar correctament les consultes externes

per inferir amenaces potencials i, simultàniament, si el model de llenguatge *Llama 3.1* reacciona de manera segura rebaixant el nivell de confiança (*LOW*) a causa de la falta de certesa de la versió.

Finalment, es dissenyen escenaris on s'activen de manera simultània nombrosos ports i serveis, incloent estructures complexes com *Nginx*, *PostgreSQL* i *Apache ActiveMQ*. Aquest enfocament permet sotmetre l'*script* d'orquestració a una prova amb múltiples dades, avaluant si l'assistent és capaç de processar correctament una jerarquia extensa de ports sense creuar informació contextual entre ells ni patir saturacions en la finestra de context del model.

8.4 Conclusions de les proves de validació

Com es pot observar a la taula 2 i en més detall en l'Apèndix B, l'anàlisi de les dades obtingudes al laboratori confirma la solidesa i la viabilitat tècnica de l'assistent Nika.

Criteri d'avaluació	Resultats
Casos d'estudi reals avaluats	18
Taxa de detecció	83,33%
Eficiència del filtratge de soroll	~88,43%
Temps mitjà d'inferència	34,1 s

TAULA 2: RESUM DELS RESULTATS DE VALIDACIÓ.

Si s'exclouen els entorns de control no vulnerables i les proves dissenyades intencionadament sense versió (implementades per avaluar la maduresa del sistema davant d'entorns atípics), el sistema assoleix una taxa de detecció del 83,33%. Aquest valor es calcula a partir del percentatge de casos on l'assistent és capaç de llistar correctament la vulnerabilitat objectiu de l'entorn o, en el seu defecte, una equivalència de la mateixa versió plenament funcional. Respecte al 16,67% d'errors restants, el motiu ha estat l'absència física dels *scripts* dins d'*ExploitDB*, fet que demostra que, tot i treballar amb un dels millors repositori de referència, l'èxit depèn directament d'aquesta font.

D'altra banda, la taxa de reducció de falsos positius del ~88,43% valida l'èxit de l'enfocament de cerca híbrida junt amb el filtratge de candidats. Aquest percentatge representa la diferència entre el volum total dels 40 candidats inicials capturats semànticament per la base vectorial, la cerca literal o el mòdul *OSINT*, i la llista final que l'assistent acaba mostrant a l'auditor (4,625 vulnerabilitats de mitjana per servei).

Pel que fa al rendiment, el temps mitjà d'inferència de 34,1 segons per servei demostra ser altament competitiu per a auditories reals. Aquesta mitjana engloba tant anàlisis de casos més individuals i directes, com ara el cas de la validació de *Redis* executada en només 30,22s, com entorns multi-port més complexos com a exemple, l'escenari combinat de quatre serveis simultanis (*PostgreSQL*, *nginx*, *Jetty* i *ActiveMQ*) que va requerir un temps total de 2 minuts i 16,48 segons. Com a conclusió, els resultats demostren que l'arquitectura compleix amb els objectius establerts, consolidant Nika com un sistema robust, ràpid i privat capaç d'optimitzar el temps de decisió de l'auditor.

9 LÍNIES DE TREBALL FUTUR I ESCALABILITAT EN ENTORNS CORPORATIUS

L'arquitectura modular de Nika posseeix un gran potencial d'expansió en múltiples direccions. Tot i la immensitat de vies de desenvolupament aplicables, en aquest apartat es presenta una selecció delimitada de les línies de futur i vies d'escalabilitat que aportarien un major impacte al projecte.

9.1 Línies de treball futur

Resiliència en la recuperació i diversificació de fonts:

La dependència exclusiva d'*ExploitDB* es podria mitigar integrant repositoris addicionals. En cas de no localitzar un codi d'exploitació actiu, el sistema podria proporcionar informació addicional com enllaços a *wroteups* de referència, guies tècniques i altres fonts de valor per agilitzar l'anàlisi de l'auditor independentment de la disponibilitat de l'script a la base de dades local.

Expansió a altres fases i automatització: Una de les línies de futur més rellevants és l'expansió de l'assistent cap a d'altres fases operatives de l'auditoria amb l'objectiu d'agilitzar el flux de treball global. Per exemple, en l'etapa de documentació, i aprofitant l'estructura interna en format JSON, els rànquings de confiança i els tipus de vulnerabilitats, el sistema es podria programar per exportar automàticament esborradors d'informes tècnics estandarditzats. Això il·lustra com l'arquitectura de l'eina es pot aplicar per reduir dràsticament la càrrega administrativa i en els blocs més mecànics del *pentesting*.

9.2 Escalabilitat en entorns corporatius

Desplegament de models a gran escala: En disposar dels recursos i el pressupost propis d'un entorn corporatiu, es podria prescindir de la quantització agressiva requerida pel maquinari de consum limitat. Això permetria allotjar models de paràmetres massius en servidors de càlcul dedicats, augmentant dràsticament la finestra de context de l'assistent, la seva precisió en l'anàlisi d'arquitectures complexes i eliminant gran part de les al·lucinacions residuals.

Arquitectura centralitzada i multiusuari: El sistema es podria transformar d'un script d'execució individual a una arquitectura distribuïda basada en microserveis dins de la xarxa privada de l'organització. Aquesta transició permetria desplegar una API interna de caràcter privat perquè múltiples auditors poguessin interactuar de manera concurrent amb l'assistent des de qualsevol punt de la infraestructura de l'empresa.

10 CONCLUSIONS

El desenvolupament de l'assistent Nika ha finalitzat de manera molt satisfactòria, confirmant que és totalment viable optimitzar tasques de les auditories de seguretat ofensiva mitjançant intel·ligència artificial en entorns locals. Amb aquest projecte, s'ha aconseguit dissenyar i desplegar un sistema modular capaç de processar outputs d'escaneig de xarxa, identificar serveis vulnerables de

forma eficient i proveir directament els codis d'exploitació corresponents. Tot això s'ha assolit garantint el control absolut i la privadesa de les dades gràcies a l'ús d'un model local adaptat a maquinari de consum, un fet que demostra que no cal dependre de solucions al núvol per obtenir un suport tècnic de qualitat en el camp de la ciberseguretat ofensiva.

D'altra banda, cal destacar la incorporació del mòdul de cerca web en temps real (OSINT). Tot i que en la proposta inicial es va plantejar principalment com un objectiu d'exploració i prototipatge, finalment s'ha pogut integrar com una eina funcional dins del flux de l'assistent. Aquesta característica permet al sistema consultar fonts externes en temps real per enriquir i contrastar la informació de les vulnerabilitats, fet que aporta un gran valor durant la identificació d'amenaçes.

A nivell personal, aquest ha estat el projecte acadèmic més gran que he realitzat fins ara. M'ha exigat d'una bona organització del temps i una gran inversió d'hores de treball. A més, m'ha obligat a investigar a fons un tema el qual no havia treballat mai amb anterioritat, com són els models d'intel·ligència artificial. Tot i la gran quantitat de feina que ha suposat, he gaudit molt realitzant aquest projecte i crec que m'ha aportat un gran valor, tant a nivell personal com per al meu futur professional.

ÚS DE LA IA GENERATIVA

Més enllà del desenvolupament íntegre del projecte, el qual ha estat centrat en aquesta tecnologia, s'ha utilitzat la intel·ligència artificial generativa com a eina de suport en les tasques de cerca d'informació de fonts bibliogràfiques i documentació tècnica, així com en l'assistència en l'escriptura i correcció gramatical d'aquest article. També, s'ha comptat amb el suport d'eines d'assistència a la programació integrades a l'entorn de desenvolupament per a l'autocompletat de codi i l'optimització de funcions, actuant com un suport tècnic per agilitzar la implementació de la lògica del sistema.

AGRAÏMENTS

En primer lloc, vull agrair al meu tutor, Enric Alibech, pels seus consells, propostes i revisions. El seu suport en l'enfocament i estructuració d'aquest treball m'ha ajudat a perfeccionar el contingut i a aconseguir que el resultat final fos el millor possible.

També vull donar les gràcies a la meva família, per haver-me fet costat i haver-me donat suport en tot moment. De manera molt especial, vull donar-li les gràcies a la meva mare, que tot i haver passat per moments complicats, amb la finalització d'aquest projecte puc dir que estic a les portes de considerar-me enginyer gràcies a ella.

Finalment, vull agrair a totes les persones que han dedicat el temps de llegir aquest article. Desenvolupar aquest projecte durant els últims mesos ha estat un camí exigent, ple de reptes i frustracions esporàdiques, però també ha

estat una experiència profundament enriquidora la qual he gaudit molt i m'alegra poder compartir amb vosaltres.

Moltes gràcies.

REFERÈNCIES

- [1] INCIBE, “INCIBE detectó más de 122.000 incidentes de ciberseguridad en 2025,” *INCIBE - Sala de Prensa*, Febrer 2026. [En línia]. Disponible a: <https://www.incibe.es/incibe/sala-de-prensa/incibe-detecto-mas-de-122000-incidentes-de-ciberseguridad-en-2025>
- [2] Darktrace, “The State of AI Cybersecurity 2026: Unveiling insights from over 1,500 security leaders,” *Darktrace Blog*, Febrer 2026. [En línia]. Disponible a: <https://www.darktrace.com/es/blog/the-state-of-ai-cybersecurity-2026>
- [3] Artificial Analysis, “MMLU-Pro Leaderboard and Evaluations,” *artificialanalysis.ai*, 2026. [En línia]. Disponible a: <https://artificialanalysis.ai/evaluations/mmlu-pro>.
- [4] E. Hilario, S. Azam, J. Sundaram, et al., “Generative AI for pentesting: the good, the bad, the ugly,” *Int. J. Inf. Secur.*, vol. 23, pp. 2075–2097, Març 2024. doi: 10.1007/s10207-024-00835-x
- [5] PortSwigger, “The future of pentesting is human x AI – and it’s already in Burp Suite Professional,” *PortSwigger Blog*, Octubre 2025. [En línia]. Disponible a: <https://portswigger.net/blog/the-future-of-pentesting-is-human-x-ai-and-its-already-in-burp-suite-professional>
- [6] INCIBE, “Pentesting”, INCIBE: Aprende Ciberseguridad, Febrer 2026. [En línia]. Disponible a: <https://www.incibe.es/aprendeciberseguridad/pentesting>
- [7] KELA, “Blue Team vs. Red Team in Cybersecurity: Differences Explained,” *KELA Academy*, Juliol 2025. [En línia]. Disponible a: <https://www.kelacyber.com/academy/cti/blue-team-vs-red-team-in-cybersecurity-differences-explained/>
- [8] Intelequia, “Red Team y Blue Team: Funciones y diferencias en ciberseguridad,” *Intelequia Blog*, s.d. [En línia]. Disponible a: <https://intelequia.com/es/blog/post/red-team-y-blue-team-funciones-y-diferencias-en-ciberseguridad>
- [9] INCIBE, “¿Qué es el pentesting? Auditando la seguridad de tus sistemas,” *INCIBE Blog*, Juliol 2019. [En línia]. Disponible a: <https://www.incibe.es/empresas/blog/el-pentesting-auditando-seguridad-tus-sistemas>
- [10] Ciberseguridad Euskadi, “Penetration Testing Execution Standard (PTES),” *Cyberzaintza*, Octubre 2021. [En línia]. Disponible a: <https://www.euskadi.eus/gobierno-vasco/-/ciberpedia/marcos-de-referencia/penetration-testing-execution-standard-ptes/>
- [11] C. Stryker, “¿Qué son los grandes modelos de lenguaje (LLM)?,” *IBM Think*, s.d. [En línia]. Disponible a: <https://www.ibm.com/es-es/think/topics/large-language-models>
- [12] IBM, “¿Qué es un modelo transformador?,” *IBM Think*, Novembre 2025 [En línia]. Disponible a: <https://www.ibm.com/es-es/think/topics/transformer-model>
- [13] SAP, “¿Qué es un modelo de lenguaje extenso?,” *SAP Insights*, Juliol 2024. [En línia]. Disponible a: <https://www.sap.com/spain/resources/what-is-large-language-model>
- [14] Universidad Isabel I, “Riesgos de la IA: filtración de datos sensibles,” *Blog UI1*, Març 2026. [En línia]. Disponible a: <https://www.ui1.es/blog-ui1/riesgos-de-la-ia-filtracion-de-datos-sensibles>
- [15] Aimultiple, “Cloud LLM vs Local LLMs: Examples & Benefits,” *Aimultiple Blog*, Febrer 2026. [En línia]. Disponible a: <https://aimultiple.com/cloud-llm>
- [16] Google Trends, “Explora lo que el mundo está buscando,” *Google Trends*, 2024. [En línia]. Disponible a: <https://trends.google.es/>
- [17] TechPlanet, “¿Cuánta RAM necesitas en 2026? 16, 32 o 64 GB,” *TechPlanet*, Febrer 2026. [En línia]. Disponible a: https://techplanet.es/cuanta-ram-necesitas/#Recomendacion_final
- [18] M. Schwarzmüller, “Why LLMs Need GPUs and VRAM,” *Maximilian Schwarzmüller Blog*, Maig 2025. [En línia]. Disponible a: <https://maximilian-schwarzmuller.com/articles/llms-gpu-cpu-vram-ram/>
- [19] K. Dhanakotti, “Exploring Quantization in Large Language Models (LLMs): Concepts and techniques,” *Medium*, Agost 2024. [En línia]. Disponible a: <https://medium.com/data-science-at-microsoft/exploring-quantization-in-large-language-models-llms-concepts-and-techniques-4e513ebf50ee>
- [20] Local AI Zone, “AI Model Quantization 2025: Master Compression Techniques for Maximum Performance & Efficiency,” *Local AI Zone Guides*, Octubre 2025. [En línia]. Disponible a: <https://local-ai-zone.github.io/guides/what-is-ai-quantization-q4-k-m-q8-gguf-guide-2025.html>
- [21] Ollama, “Llama 3.1,” *Ollama Library*, Novembre 2024. [En línia]. Disponible a: <https://ollama.com/library/llama3.1>
- [22] Ollama, “Nomic Embed Text,” *Ollama Library*, Febrer 2024. [En línia]. Disponible a: <https://ollama.com/library/nomic-embed-text>
- [23] PyPI, “ddgs 0.2.2,” *Python Package Index*, Maig 2023. [En línia]. Disponible a: <https://pypi.org/project/ddgs/>

APÈNDIX

A GLOSARI DE TERMES TÈCNICS I SIGLES

Amb l'objectiu de facilitar la lectura i assegurar la claredat del text, aquest apartat presenta una definició breu i senzilla dels conceptes tècnics, les sigles i els acrònims utilitzats al llarg de l'article, servint com a guia de referència per a aquells lectors que no estiguin familiaritzats amb aquesta terminologia.

AI-lucinacions: Fenomen en el qual un model d'intel·ligència artificial genera respostes incorrectes, inventades o sense justificació real, mostrant-les de manera aparentment coherent i amb una falsa aparença de certesa.

API (*application programming interface*): Interfície de programació d'aplicacions, conjunt de regles i protocols que permet a diferents programes comunicar-se i intercanviar dades entre si.

Benchmarks: Proves estandarditzades dissenyades per mesurar, avaluar i comparar de forma objectiva el rendiment i la precisió de sistemes de programari o maquinari.

ChromaDB: Base de dades vectorial de codi obert especialitzada en l'emmagatzematge i recuperació ràpida de vectors d'embeddings per a arquitectures RAG.

CSV (*comma-separated values*): Format de fitxer de text pla senzill utilitzat per emmagatzemar dades estructurades en forma de taula on els camps es separen mitjançant comes o altres delimitadors.

CVE (*common vulnerabilities and exposures*): Identificador únic, universal i públic utilitzat per catalogar i estandarditzar una vulnerabilitat de seguretat registrada internacionalment.

DB (*database*): Base de dades (BD), és a dir, un sistema estructurat utilitzat per organitzar, emmagatzemar, gestionar i recuperar grans volums d'informació de manera eficient.

Docker: Tecnologia de virtualització lleugera basada en contenidors que permet empaquetar, distribuir i executar aplicacions de forma aïllada i portable.

DoS (*denial of service*): Atac de denegació de servei que té com a objectiu saturar els recursos d'un sistema o xarxa per fer que quedi completament inoperable per als usuaris legítims.

Embedding: Representació numèrica en forma de vector multidimensional que captura el significat semàntic, el context i la relació de paraules o textos estructurats.

Exploit: Codi, programari o seqüència de comandes dissenyada específicament per aprofitar una vulnerabilitat tècnica i aconseguir un accés o comportament no autoritzat.

ExploitDB: Repositori públic i arxiu local regulat que consolida codis d'exploitació i exploits associats a vulnerabilitats de seguretat conegudes.

GGUF (*gpt-generated unified format*): Format de fitxer binari dissenyat per a la càrrega ultra-ràpida, optimització i inferència eficient de models LLM en equips locals amb maquinari de consum.

JSON (*javascript object notation*): Format de text lleuger, intercanviable i fàcil de llegir per a humans i màquines, utilitzat

per l'assistent per estructurar dades mitjançant claus i valors.

LLM (*large language model*): Model de llenguatge extens basat en arquitectures de xarxes neuronals profundes, entrenat amb grans volums de text per processar i generar llenguatge natural.

LPE (*local privilege escalation*): Escalada de privilegis local, un tipus de vulnerabilitat o atac on un usuari amb permisos limitats aconsegueix elevar els seus drets fins a nivells superiors.

MMLU-Pro: Prova de referència avançada i complexa dissenyada per avaluar la capacitat operativa, el coneixement multidisciplinari i el raonament dels models d'intel·ligència artificial.

Nmap: Eina de codi obert de referència utilitzada per a l'auditoria i exploració de xarxes, que permet descobrir hosts actius, ports oberts, serveis i versions de programari.

Nomic-embed-text: Model de generació d'embeddings de text d'alt rendiment optimitzat per a tasques de cerca semàntica i recuperació de context en arquitectures RAG.

Ollama: Motor d'inferència lleuger i local que facilita la gestió, configuració i execució de diversos models de llenguatge extensos directament en la màquina de l'usuari.

OSINT (*open source intelligence*): Intel·ligència de fonts obertes, la qual consisteix en un conjunt de metodologies i tècniques que tenen l'objectiu de recopilar i analitzar dades d'accés públic per extreure conclusions útils.

Pandas: Llibreria de programació per a Python especialitzada en la manipulació de dades, utilitzada en aquest projecte per realitzar el filtratge literal i ràpid sobre bases d'informació.

Pentesting (*penetration testing*): Prova de penetració o auditoria de seguretat ofensiva on es simulen atacs reals autoritzats contra una infraestructura per identificar-ne les debilitats.

Prompts: Textos, instruccions o línies de comandament introduïdes de manera específica per interactuar, guiar el context i obtenir una resposta determinada d'un model d'IA.

Python: Llenguatge de programació d'alt nivell, modular i interpretat, amb el qual s'ha programat íntegrament la lògica i l'orquestració de l'assistent Nika.

RAG (*retrieval-augmented generation*): Tècnica que millora la precisió de les respostes d'un model LLM injectant-li text contextual rellevant recuperat d'una base de dades externa.

RAM (*random access memory*): Memòria d'accés aleatori del sistema, utilitzada per emmagatzemar temporalment les dades dels processos operatius i aplicacions actives de l'ordinador.

RCE (*remote code execution*): Execució remota de codi, un tipus d'atac o vulnerabilitat crítica que obre la porta a que un atacant extern executi ordres i comandes arbitràries en un servidor remot.

Root: Nom per defecte del compte d'usuari de màxima jerarquia que posseeix el control absolut i tots els privilegis d'administració en sistemes operatius basats en Unix i Linux.

Script: Fitxer de text que conté un conjunt d'instruccions escrites en codi de programació que s'executen de manera automatitzada i seqüencial per un intèrpret.

Shodan: Motor de cerca especialitzat orientat a localitzar dispositius, routers, servidors i qualsevol mena d'infraestructura de programari exposada públicament a Internet.

Vulnerabilitat: Fallada, error de disseny o defecte de configuració en un actiu de programari o maquinari que compromet la seva seguretat i pot ser explotat per un atacant.

VRAM (video random access memory): Memòria d'accés aleatori de la targeta gràfica, crítica en aquest projecte per carregar els paràmetres del model LLM i fer un processament d'alta velocitat.

Writeups: Documents explicatius de caràcter tècnic redactats per analistes on es detalla el procediment lògic seguit pas a pas per resoldre un repte o explotar una vulnerabilitat de seguretat.

B RESULTATS DE LES PROVES DE VALIDACIÓ

B.1 OpenSSH 10.2p1. CVE Objectiu: Cap

Valoració: Cap (Servei de control).

CVEs rellevants no objectiu: Com que es tracta del servei utilitzat legítimament per a la gestió del propi laboratori, no s'esperava cap troballa crítica. No obstant això, s'envia a l'assistent com a prova de control. L'eina actua correctament: identifica vulnerabilitats potencials històriques, però a causa de la versió actualitzada de l'entorn, els assigna de forma madura una prioritat *MEDIUM / LOW*, demostrant un bon filtratge de soroll.

B.2 Apache 2.4.49. CVE Objectiu: CVE-2021-41773

Valoració: Confiança: *HIGH*. Resposta excel·lent. L'assistent identifica de forma exacta l'*script* crític de *path traversal* i RCE (*Exploit ID 50383*) corresponent al laboratori actiu.

CVEs rellevants no objectiu: **CVE-2021-42013** (Confiança: *MEDIUM*). Vulnerabilitat addicional detectada que, tot i estar catalogada originalment per a la versió 2.4.50, segueix sent vulnerable en la versió del laboratori. És una identificació excel·lent amb una confiança *MEDIUM* preventiva atès que la versió no és totalment exacta.

B.3 Drupal 8.5.0. CVE Objectiu: CVE-2018-7600

Valoració: Confiança: *HIGH*. Resposta excel·lent. L'assistent identifica de forma precisa múltiples lliuraments de l'*script* de *drupalgeddon2* (*Exploit IDs 44482, 44449, 44448*) coincidents amb el laboratori.

CVEs rellevants no objectiu: **CVE-2019-6340** (Confiança: *MEDIUM*). Detecció crítica addicional d'execució remota de comandes (RCE) identificada correctament per proximitat dins de la branca 8.5.x auditada.

B.4 Joomla 3.4.5. CVE Objectiu: CVE-2015-8562

Valoració: Confiança: *HIGH*. Detecció correcta. L'assistent localitza l'*script* d'injecció d'objectes (*Exploit ID 38977*) que replica el vector d'entrada del contenidor actiu.

CVEs rellevants no objectiu: **CVE-2015-8566** (Confiança: *HIGH*). Resultat rellevant que comparteix mètode d'exploitació i rang d'afectació sobre capçaleres per a la mateixa branca de programari.

CVE-2014-7228 (Confiança: *MEDIUM*). Identificació vàlida d'un component de tercers (*akeeba kickstart*) vinculat de manera semàntica a la plataforma.

B.5 Apache Solr 8.2.0. CVE Objectiu: CVE-2019-17558

Valoració: Confiança: *HIGH*. Resposta òptima. L'assistent combina amb èxit la informació del mòdul d'OSINT i extreu l'*script* d'execució de comandes remotes per *velocity template* (*Exploit ID 47572*) adaptat a la versió 8.2.0.

CVEs rellevants no objectiu: **CVE-2019-17558** (*Metasploit*) (Confiança: *MEDIUM*). Identificació complementària útil de la mateixa CVE a través de *Metasploit*

B.6 Redis 4.0.14. CVE Objectiu: Exploit ID 47195

Valoració: Confiança: *HIGH*. L'assistent identifica correctament l'*script* d'execució de codi remot (RCE) sense autenticació per a la branca 4.x/5.x de Redis tant en les proves realitzades amb versió explícita com en l'escaneig genèric sense versió indicada.

CVEs rellevants no objectiu: **Exploit ID 48272** (Confiança: *HIGH*). Vulnerabilitat crítica addicional basada en el vector de replicació de Redis que permet l'execució de comandes remotes, essent un resultat altament valuós per a l'auditoria.

B.7 Webmin 1.910. CVE Objectiu: CVE-2019-15107

Valoració: Confiança: *MEDIUM* (Identificació equivalent). Tot i que l'*script* de la vulnerabilitat del *backdoor* de *rpc.cgi* es llista amb confiança *MEDIUM* (*Exploit ID 47330*), el sistema prioritza un *rce* alternatiu completament funcional per a aquesta versió específica (*CVE-2019-12840*) amb confiança *HIGH*.

CVEs rellevants no objectiu: **CVE-2017-15646** (Confiança: *LOW*). Vulnerabilitats secundàries avaluades correctament.

B.8 Apache CouchDB 1.6.0. CVE Objectiu: CVE-2017-12636

Valoració: Confiança: *HIGH*. Resposta correcta. El sistema resol l'ID d'explotació crític d'execució de comandes arbitràries (*Exploit ID 45019*).

CVEs rellevants no objectiu: **Exploit ID 44913** (Confiança: *MEDIUM*). Codi de vulnerabilitat alternatiu vàlid de Python aplicable a versions inferiors a la 2.1.0.

B.9 Zabbix 3.0.3. CVE Objectiu: CVE-2016-10134

Valoració: Confiança: *HIGH* (Identificació equivalent). Èxit de la indexació semàntica. Tot i no llistar el codi textual del CVE objectiu, l'assistent extreu els *scripts* de la injecció *sql* i de l'*rce* de la línia de codi de la branca de treball (*Exploit IDs 40353, 39937*).

CVEs rellevants no objectiu: **Exploit ID 40237** (Confiança: *MEDIUM*). Injecció *sql* complementària i compatible amb la plataforma auditada.

B.10 Sonatype Nexus 3.14.0. CVE Objectiu: CVE-2019-7238

Valoració: Confiança: No detectat. El motor de cerca local no inclou l'ID literal del laboratori a causa d'una manca de coincidència exacta indexada per a aquesta versió.

CVEs rellevants no objectiu: CVE-2020-10199 (Confiança: *HIGH / MEDIUM*). Fals positiu de versió on la descripció del repositori induïx el LLM a llistar *exploits* crítics de Java EL de versions posteriors (v3.21.1).

B.11 Openfire 4.7.4. CVE Objectiu: CVE-2023-32315

Valoració: Confiança: Detecció parcial. Malgrat que el mòdul d'OSINT localitza la vulnerabilitat de *path traversal* externa, la manca de l'*script* físic dins del repositori d'*ExploitDB* produeix que en els resultats finals no trobem la vulnerabilitat objectiu.

CVEs rellevants no objectiu: Exploit ID 40065 (Confiança: *HIGH*). Fals positiu de versió on la fase 3 admet opcions d'una branca antiga (v3.x).

B.12 InfluxDB 1.6.6. CVE Objectiu: CVE-2019-20933

Valoració: Confiança: Parcial. Detectat correctament gràcies al mòdul OSINT però de nou, al no estar catalogat en la base de dades no és capaç de proporcionar més informació en profunditat.

B.13 phpMyAdmin 4.8.1. CVE Objectiu: CVE-2018-12613

Valoració: Confiança: *HIGH*. Resposta excel·lent. L'assistent identifica de forma exacta els *scripts* crítics de *local file inclusion* i *RCE* (*Exploit IDs* 44924, 44928, 50457) corresponents al laboratori actiu.

B.14 Oracle WebLogic Server. CVE Objectiu: CVE-2017-10271

Valoració: Confiança: *LOW*. Èxit del mòdul OSINT i la cerca semàntica. L'assistent és capaç d'intuir i llistar la vulnerabilitat exacta del laboratori (*Exploit ID* 43392) malgrat que Nmap només va reportar metadades de text ("Servlet 2.5; JSP 2.1") i cap versió numèrica neta. L'assignació de la confiança *LOW* és completament prudent i correcta.

CVEs rellevants no objectiu: CVE-2020-14882 (Confiança: *HIGH*). Fals positiu provocat per l'ambigüitat de l'entrada, el model tendeix a assignar nivells de confiança alts a *exploits* molt populars que podrien coincidir, quan es troba amb una manca de dades com en aquest cas.

B.15 Apache Tomcat 8.5.19. CVE Objectiu: CVE-2017-12615

Valoració: Confiança: No detectat. El component de cerca local no aconsegueix prioritzar el CVE objectiu a causa de la saturació d'*scripts* alternatius de Tomcat indexats al repositori local de l'eina.

CVEs rellevants no objectiu: CVE-2019-0232 (Confiança: *HIGH*). Identificació vàlida de *RCE* en el mòdul *CGIServlet* aplicable a la branca 8.5, tot i que limitada en entorns reals

a plataformes Windows.

CVE-2016-5425 / CVE-2016-1240 (Confiança: *HIGH*). Fals positiu de la branca versió del servei.

B.16 MySQL 5.5.23. CVE Objectiu: CVE-2012-2122

Valoració: Confiança: No detectat l'objectiu específic.

CVEs rellevants no objectiu: CVE-2016-6664 / CVE-2016-6663 (Confiança: *HIGH*). Detecció completament vàlida d'escalada de privilegis local (LPE) aplicable de manera estructural a servidors sota entorns MySQL 5.5. **CVE-2016-6662** (Confiança: *HIGH*). Identificació d'un *RCE/LPE* molt proper crític que afecta totes les versions inferiors a la 5.5.52.

B.17 PostgreSQL 9.6.7. CVE Objectiu: CVE-2018-1058

Valoració: Confiança: *HIGH* (Identificació equivalent). Tot i que l'objectiu teòric era el CVE-2018-1058 i com que aquest no es troba dins de la base de dades d'*ExploitDB*, l'assistent assoleix un gran èxit en identificar l'*Exploit ID* 51247 (CVE-2019-9193) (*RCE* per a la branca 9.6). L'eina determina amb total precisió una vulnerabilitat d'execució remota de comandes plenament funcional per a l'entorn actiu.

CVEs rellevants no objectiu: CVE-2019-9193 / Exploit ID 46813 (Confiança: *LOW*). Vulnerabilitat *RCE* complementària del mateix CVE-2019-9193 afectant de nou la versió del laboratori.

B.18 nginx 1.13.2. CVE Objectiu: CVE-2017-7529

Valoració: Confiança: Parcial (Mòdul OSINT). L'assistent demostra un èxit en el mòdul OSINT localitzant el CVE-2017-7529 en la llista de potencials amenaces. Desgraciadament, la correlació no es consolida en el resultat final per limitacions físiques del repositori local, en no disposar de l'*script* a *ExploitDB*.

CVEs rellevants no objectiu: CVE-2013-2028 (Confiança: *HIGH*). Fals positiu de versió provocat per una alta similitud semàntica, tot i que només és aplicable a la v1.3.9-1.4.0.

B.19 Apache ActiveMQ 5.16.3 (Real: 5.11.1). CVE Objectiu: CVE-2015-5254

Valoració: Confiança: *MEDIUM*. Cas altament positiu per a l'eina. Tot i que el servei s'exposava d'una manera confusa que va desencadenar que Nmap el detectés com a "5.16.3 (or earlier)" quan realment era 5.11.1, cosa que fa que l'eina no tingui la versió real a analitzar, l'assistent aporta de manera proactiva *exploits* crítics de la versió real en el laboratori (5.11.1).

CVEs rellevants no objectiu: CVE-2016-3088, CVE-2015-1830 / Exploit ID 40857 (Confiança: *MEDIUM*). Identificació excel·lent d'un vector de *directory traversal* i execució de comandes directament aplicable a la versió real instal·lada 5.11.1.