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

Boza Zapater, Jose Angel. *Millora d'escenaris Docker en una distribució Linux virtualitzada per a pràctiques de xarxes*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334419>

under the terms of the  license.

Millora d'escenaris Docker en una distribució Linux virtualitzada per a pràctiques de xarxes

Jose Angel Boza Zapater

Resum— En aquest treball ens encarreguem de partir d'un laboratori de pràctiques que s'ha utilitzat en diverses assignatures del grau d'Enginyeria Informàtica i Enginyeria de Dades per a l'ensenyament del funcionament de les xarxes. A partir d'aquesta base, hi introduïm millores amb l'objectiu de fer-lo molt més complet i d'ampliar els conceptes fonamentals relacionats amb les xarxes i la seva interconnexió. En concret, s'hi incorporen tecnologies i conceptes com NAT, DHCP, DMZ, tallafocs, IPv6, etc. Aquests conceptes permeten enriquir el laboratori i apropar-lo a entorns més propers als que es troben en infraestructures reals. A més, el projecte també inclou un component de conscienciació sobre seguretat. En aquest sentit, es mostra com aquestes tecnologies poden ser explotades per denegar serveis o per capturar informació sensible, i com la manca de mecanismes de seguretat adequats pot posar en risc la confidencialitat, la identitat, la integritat de la informació i la seguretat de la infraestructura digital d'una organització.

Paraules clau— Xarxes, laboratori, NAT, DHCP, DMZ, DNS, VPN, Proxy, tallafocs, IPv6, infraestructura, seguretat, confidencialitat, integritat, identitat, denegació, servei, captura, infraestructura, enverinament, inundació, suplantació.

Abstract— In this paper, we start from a lab exercise that has been used in various courses of the Computer Engineering degree and Data Engineering Degree to teach how networks operate. Based on this foundation, we introduce improvements aimed at making it much more comprehensive and expanding the fundamental concepts related to networks and their interconnection. Specifically, it incorporates technologies and concepts such as NAT, DHCP, DMZ, Firewalls, IPv6, etc. These concepts enrich the lab and bring it closer to environments found in real-world infrastructures. Additionally, the project also includes a security awareness component. In this regard, it shows how these technologies can be exploited to deny services or capture sensitive information, and how the lack of adequate security mechanisms can put at risk the confidentiality, identity, integrity of information, and security of an organization's digital infrastructure.

Index Terms— Networks, lab, NAT, DHCP, DMZ, DNS, VPN, Proxy, firewalls, IPv6, infrastructure, security, confidentiality, integrity, identity, denial, service, capture, infrastructure, poisoning, flooding, spoofing.

1 INTRODUCCIÓ - CONTEXT DEL PROJECTE

El present treball es centra en l'ampliació d'un entorn basat en Docker que simula una infraestructura de xarxa destinada a les pràctiques de laboratori de diverses assignatures. Aquest escenari s'ha utilitzat en els graus d'Enginyeria Informàtica i Enginyeria de Dades per a la formació en xarxes de computadors. Cal precisar, no obstant això, que durant el curs 2025-2026 l'escenari s'ha utilitzat únicament en el grau d'Enginyeria de Dades..

La infraestructura de laboratori presentava certes limitacions pel que fa a l'abast dels continguts de xarxes que s'hi poden implementar, veure Fig. 1. El punt de partida d'aquest projecte es defineix a partir de l'anàlisi de la infraestructura de laboratori existent i dels materials docents [1] proporcionats pel tutor, en Rubén Martínez Vidal. Segons la documentació i els enunciats de pràctiques originals, l'escenari presentava una configuració centrada en aspectes fonamentals de xarxa, però amb un abast limitat que requereix una ampliació. D'una banda, el material previ estableix les bases per a la configuració i l'anàlisi del desplegament de protocols

d'encaminament dinàmic, concretament *RIP* i *OSPF*. D'altra banda, la pràctica original inclou una secció orientada a la seguretat en xarxes, centrada específicament en l'estudi de l'impacte dels atacs de denegació de servei distribuïts (*DDoS*) mitjançant la tècnica d'inundació de segments de sincronització (*SYN flooding*).

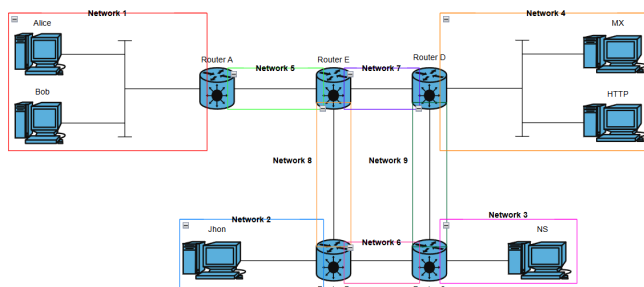


Fig. 1. Diagrama de la infraestructura de xarxa de l'entorn Docker inicial.

2 OBJECTIUS DEL PROJECTE

2.1 Ampliació de tecnologies de xarxa

L'objectiu principal d'aquest treball és l'evolució de l'entorn de simulació basat en *Docker* per a la docència, transformant una infraestructura limitada en un laboratori integral i professional. Es pretén crear un entorn on l'estudiantat pugui aprendre de manera pràctica i visual el funcionament de les xarxes de computadors. Per assolir-ho, la infraestructura s'ha dividit en quatre sub objectius específics, que corresponen a quatre subxarxes diferenciades. Cada xarxa representa un escenari concret i totes elles estan interconnectades simulant l'entorn global d'Internet, veure Fig. 2.

2.1.1 Subobjectiu 1: Xarxa Domèstica (N1)

Aquesta xarxa simula un entorn residencial per demostrar com els proveïdors de serveis d'Internet (ISP) utilitzen mecanismes de configuració i traducció d'adreces per gestionar l'escassetat d'adreces IPv4 públiques i protegir la privacitat. En aquesta xarxa estan implementats els conceptes següents [2, 3]:

- **NAT** (Traducció d'Adreces de Xarxa): Tècnica que permet que múltiples adreces IP privades es comuniquin amb l'exterior sota una única IP pública.
- **DHCP** (Protocol de Configuració Dinàmica d'Amfitrions): Protocol que assigna automàticament adreces IP i altres paràmetres de configuració als dispositius de la xarxa local.

2.1.2 Subobjectiu 2: Xarxa Empresarial (N2)

En aquest escenari es recrea un entorn corporatiu on s'apliquen sistemes de seguretat i optimització per protegir serveis crítics, mantenint alhora l'exposició controlada a Internet de certs actius. També s'hi inclou un servidor de registres (*logs*) per monitorar l'activitat i detectar anomalies. En aquesta xarxa, denominada N2, s'integren quatre tecnologies clau [4][5][7]:

- **Proxy**: Servidor intermediari que gestiona les peticions entre els dispositius interns i Internet, actuant com a pont.
- **DMZ (Zona Desmilitaritzada)**: Subxarxa aïllada que actua com a zona intermèdia entre la xarxa pública (Internet) i la xarxa interna privada.
- **Tallafocs (Firewall)**: Sistema de seguretat que monitora i filtra el tràfic de xarxa segons regles predefinides per controlar els accessos.
- **IPv6**: Implementació de la versió més recent del protocol d'Internet per abordar l'esgotament de les adreces IPv4.

2.1.3 Subobjectiu 3: Xarxa de Serveis (N3)

Aquesta xarxa, anomenada N3, simula la infraestructura que proveeix serveis essencials per a la navegació, la seguretat i l'anonimat a la xarxa, basant-se en dos pilars [2][3][7]:

- **DNS (Sistema de noms de domini)**: Sistema encarregat de traduir noms de domini (com *www.google.com*) en adreces IP numèriques.
- **VPN (Xarxa Privada Virtual)**: Tecnologia que crea un túnel xifrat entre el dispositiu i un servidor remot per garantir la confidencialitat del tràfic.

2.2 Subobjectiu 4: Ampliació d'atacs i vulnerabilitats (N4)

En aquest apartat s'introdueix la figura d'un atacant extern situat en una xarxa específica (N4). Des d'aquest terminal, s'executen diverses amenaces dirigides a les tecnologies descrites anteriorment per visualitzar-ne les vulnerabilitats i l'impacte (pèrdues econòmiques, interrupció del servei o robatori d'informació). Els atacs simulats són [8][9][10][11]:

- **Enverinament de Taules d'Enrutament (*Routing Tables Poisoning*)**: Manipulació de les rutes dels encaminadors amb informació falsa per desviar el tràfic.
- **Suplantació d'IP (*IP Spoofing*)**: Falsificació de l'adreça d'origen d'un paquet per enganyar el destinatari sobre la seva procedència.
- **Home al mig i enverinament ARP (*MitM + ARP Spoofing*)**: Intercepció de comunicacions mitjançant missatges ARP falsos que vinculen la MAC de l'atacant amb la IP del encaminador.
- **Atac de denegació de servei distribuït (*DDoS*)**: Saturació d'un servei mitjançant un volum massiu de tràfic. Es diferenciaren dues variants:
 - **SYN Flood**: Explota el procés de connexió TCP (*Three-Way Handshake*).
 - **ICMP Flood**: Col·lapsa la víctima mitjançant l'enviament massiu de paquets de control (*ping*).
- **Enverinament de memòria cache DNS (*DNS Cache Poisoning*)**: Introducció d'entrades falses en un servidor DNS per redirigir usuaris cap a llocs maliciosos.
- **Injecció de tràfic via Proxy**: Manipulació del contingut que circula a través d'un servidor intermediari compromès.

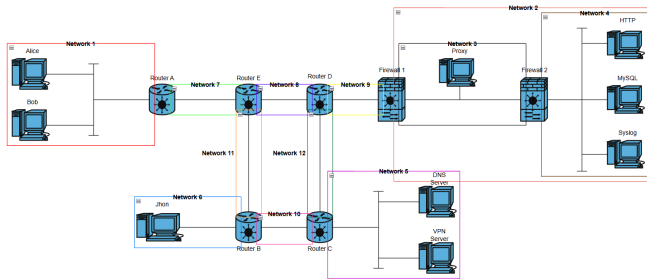


Fig. 2. Diagrama de la infraestructura de xarxa de l'entorn Docker amb les millores implementades.

2.3 Prioritats de desenvolupament

Tot i que les xarxes es poden implementar de manera independent, la xarxa *N4* depèn de l'existència prèvia dels serveis que pretén atacar. Per establir una planificació adient que asseguri el màxim de funcionalitats operatives, s'ha definit un ordre de prioritats basat en la facilitat d'aplicació i el valor docent aportat:

- Xarxa Domèstica (*N1*)
- Xarxa Empresarial (*N2*)
- Xarxa de Serveis (*N3*)
- Atacs i vulnerabilitat (*N4*)

2.4 Criteris d'assoliment

Es considera que un objectiu ha estat assolit quan es compleixen els següents punts de verificació:

2.4.1 Implementació de serveis i infraestructura

- **Desplegament funcional:** El servei ha d'estar configurat i actiu dins del seu contenidor *Docker*.
- **Verificació operativa:** Demostració que el servei respon correctament (ex: recepció d'*IP* via *DHCP* o filtratge del *Proxy*).
- **Interconnexió:** Assegurar la traçabilitat del tràfic entre subxarxes a través dels encaminadors configurats.

2.4.2 Implementació d'atacs i seguretat (*N4*)

Atès que *Docker* comparteix el nucli de la màquina amfitriona, l'assoliment es basa en:

- **Màxima cobertura viable:** Implementació dels atacs que la tecnologia de contenidors permeti sense comprometre l'estabilitat del sistema amfitrió.
- **Documentació de limitacions:** Si un atac no és viable per restriccions de *Docker*, l'objectiu es dona per assolit si s'identifica el bloqueig tècnic i es descriu un possible entorn alternatiu (com *KVM* o *VirtualBox*) on sí que ho seria.
- **Evidència de vulnerabilitat:** Presentació de captures de tràfic o registres (*logs*) que acreditin l'èxit de l'atac.

3 ESTAT DE L'ART

Existeix un ventall molt ampli de plataformes sobre les quals s'hauria pogut desenvolupar un laboratori de xarxes virtualitzat. Abans d'adoptar *Docker* com a eina central, es van valorar i descartar diverses alternatives consolidades en l'àmbit acadèmic i professional. A continuació se'n presenten les principals i es justifica l'elecció final.

3.1 Virtualització tradicional amb hipervisors

Solucions de tipus hipervisor de Tipus 2, com *Oracle VirtualBox* o *VMware Workstation*, haurien permès un aïllament complet de sistemes operatius mitjançant màquines virtuals (*VM*). Aquest aïllament és superior al dels contenidors i facilita determinats atacs de baix nivell. Tanmateix, es van descartar pel seu alt consum de recursos (*CPU*, *memòria RAM* i *emmagatzematge*), que dificulta desplegar una topologia de múltiples nodes en un ordinador estàndard.

3.2 Simuladors de xarxes específics

Plataformes com *GNS3* i *Cisco Packet Tracer* són estàndards de la indústria per simular topologies d'encaminament complexes interconnectant imatges de sistemes operatius de xarxa reals (com *Cisco IOS*). Tot i ser eines molt potents, es van desestimar perquè requereixen una corba d'aprenentatge elevada per a la configuració inicial, exigeixen la gestió de llicències de programari privatiu per a determinats nodes i afegeixen una rigidesa que dificulta la programació lliure d'scripts d'atac a baix nivell.

3.3 Entorns de simulació lleugers

Mininet és molt utilitzat en l'àmbit acadèmic per a la investigació de xarxes definides per programari (*SDN*). Es va descartar perquè, a diferència de *Docker*, no aïlla els sistemes de fitxers dels nodes de manera nativa ni ofereix la flexibilitat necessària per desplegar fàcilment serveis independents —com un servidor *Nginx* o un proxy complet— dins de cada element de la xarxa d'una manera tan modular.

3.4 Justificació de l'elecció: *Docker*

Amb l'objectiu de maximitzar l'accessibilitat i simplificar l'experiència tant del professorat com de l'alumnat, s'ha escollit *Docker* com a eina central. Els contenidors comparteixen el nucli de la màquina amfitriona, fet que permet desplegar una topologia complexa de múltiples xarxes i nodes sense un consum excessiu de recursos. A més, l'estudiant pot replicar exactament el mateix escenari en qualsevol sistema operatiu amb un simple fitxer de configuració, i la inicialització de la infraestructura és gairebé instantània, facilitant el cicle d'assaig i error propi de l'aprenentatge de xarxes.

Cal advertir, també, dues limitacions inherents als contenidors: en compartir el nucli de l'amfitrió, alguns atacs a nivell de capa de xarxa o manipulacions profundes

del stack TCP/IP poden veure's limitats; i l'aïllament de seguretat no és tan rígid com el d'un hipervisor.

3 METODOLOGIA DE DESENVOLUPAMENT

En aquest projecte s'ha aplicat una metodologia iterativa i incremental basada en el marc de treball *Scrum*. Aquest enfocament permet organitzar les funcionalitats del laboratori de manera modular i segura, garantint l'existència d'una versió de producció permanentment funcional i testejada mentre que es depuraven noves característiques en la versió de desenvolupament.

3.1 Justificació de la metodologia

S'ha triat *Scrum* com a marc àgil enfront de models tradicionals com el *Waterfall* (en cascada) per la seva capacitat d'adaptació. En un entorn de xarxes virtualitzades amb *Docker*, poden sorgir dependències o conflictes tècnics imprevistos; *Scrum* permet gestionar aquestes incerteses mitjançant cicles curts i revisions constants.

3.2 Flux de treball

Per gestionar aquest flux de treball, s'ha utilitzat un sistema de control de versions (*Git*) amb una estructura de branques diferenciada. El repositori compta amb una branca principal (*main*) que representa en tot moment l'estat productiu i verificat del laboratori. Per a cada xarxa (*N1*, *N2*, *N3* i *N4*) s'ha creat una branca específica on s'ha dut a terme el desenvolupament i la depuració corresponents. Un cop les funcionalitats d'una branca eren completament operatives i havien superat les verificacions de connectivitat i seguretat, els canvis es fusionaven manualment a *main* mitjançant un merge directe. El desenvolupament l'hem estructurat en *Sprints* (iteracions), cadascun centrat en el desplegament d'una xarxa específica (*N1*, *N2*, *N3* o *N4*). El cicle de treball dins de cada *Sprint* ha seguit les fases següents:

1. **Especificació i disseny:** Redacció dels objectius de la xarxa, descripció de components i llista de requisits tècnics.
2. **Implementació:** Configuració progressiva en *Docker*, incorporant correccions del tutor i mantenint un registre de canvis.
3. **Verificació i Validació:** Tests de connectivitat i seguretat; un cop validats, els canvis es fusionen amb la branca de producció.
4. **Documentació de fase:** Generació d'un fitxer detallat amb referències bibliogràfiques, registre de problemàtiques/solucions i documentació tècnica per a l'alumne.

Paral·lelament al desenvolupament de cada *Sprint*, s'ha dissenyat una capa de *Quality Assurance* (QA) automatitzada composta per dos scripts principals. El

script *test-lab.sh* agrupa les verificacions d'infraestructura per categories (estat de contenidors, *DHCP*, enrutament, *NAT*, serveis, *IPv6*, *VPN*, etc.) i emet un resultat binari *PASS/FAIL* per a cada cas. El script *test-attacks.sh* cobreix els sis vectors d'atac implementats, estructurant cada verificació en tres fases seqüencials: inici de l'atac, comprovació de l'efecte observable i aturada amb retorn a l'estat legítim. A més de les verificacions binàries, s'emeten mètriques quantitatives al fitxer *metrics.csv*. Ambdós scripts s'executen al final de cada *Sprint* com a criteri de validació previ a la fusió de la branca a *main*.

4. PLANIFICACIÓ TEMPORAL

Taula 1: Cronograma del Projecte: Sprints i Fites

Sprint	Setmanes	Tasca Principal	Descripció i Fites
S1	1 – 4	Fonaments i Disseny	Recollida d'informació, anàlisi de la infraestructura inicial i redacció de l'informe inicial.
S2	4 – 5	Implementació N1	Configuració de la xarxa domèstica.
S3	5 – 8	Implementació N2	Configuració de la xarxa empresarial i redacció de l'informe de progrés (I).
S4	8 – 9	Implementació N3	Configuració de la xarxa de serveis.
S5	9 – 13	Implementació N4	Execució d'escenaris d'atac i seguretat.
Final	13 – 14	Consolidació	Unificació de la documentació i preparació del lliurament final.

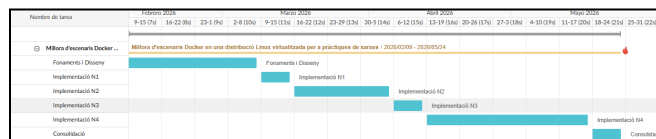


Fig. 3. Diagrama de Gantt amb la temporització i planificació de les fases del projecte.

5 RESULTATS

S'han assolit satisfactòriament totes les funcionalitats i fites definides en la planificació original dins dels terminis establerts. La naturalesa modular de l'entorn ha permès una integració àgil, consolidant un desplegament d'infraestructura estructurat al voltant dels eixos tecnològics que es detallen a continuació.

5.1 Xarxa Domèstica (N1)

S'ha configurat el *NAT* amb l'eina *iptables* en el node *Router A*, afegint les regles de post-processament necessàries per traduir les adreces de la xarxa interna cap a l'exterior. Per al *DHCP* s'ha desplegat el servei *isc-dhcp-server*, definint el rang d'adreçament dinàmic, la màscara de xarxa, la porta d'enllaç i el temps de concessió de l'adreça per als clients.

Durant el desplegament es va detectar un conflicte en l'assignació d'*IP* als nodes *Alice* i *Bob*: *Docker* assignava adreces estàtiques definides al fitxer *docker-compose.yml*, impedit que el servei *DHCP* gestionés el rang establert. La solució va consistir a eliminar la configuració d'*IP* fixa en la definició dels serveis, permetent que els contenidors fessin la petició directament al servidor *DHCP*.

Per tant, s'ha assolit el Subobjectiu 1 (Xarxa Domèstica), ja que s'ha desplegat una xarxa local amb assignació dinàmica d'adreces IP mitjançant DHCP i un encaminador amb traducció d'adreces NAT que proporciona connectivitat amb l'exterior. Això permet que múltiples dispositius de la xarxa interna comparteixin una única IP pública, simulant fidelment el comportament d'un entorn residencial gestionat per un ISP real.

5.2 Xarxa Empresarial (N2)

La xarxa inicial (192.168.4.0/24) s'ha dividit en tres subxarxes amb màscara /26: una zona externa (192.168.4.0/26) connectada al Router D i al Firewall perimetral; una DMZ (192.168.4.64/26) on s'ubica el servidor proxy; i una xarxa interna (192.168.4.128/26) on resideixen els serveis crítics HTTP, MySQL i Syslog.

S'ha implementat una arquitectura de defensa en profunditat amb doble tallafocs. El Firewall Extern actua com a perímetre exterior, filtrant tot el trànsit entrant amb política DROP per defecte i permetent només connexions HTTP/HTTPS cap al proxy mitjançant regles DNAT (ports 80 i 443 cap a 192.168.4.66). El Firewall Intern protegeix la xarxa interna amb regles granulars que només permeten `proxy→http:80`, `proxy→mysql:3306` i nodes `DMZ→syslog:514 UDP`.

Els serveis desplegats inclouen un proxy invers (nginx) a la DMZ, un servidor web (apache2) a la xarxa interna, una base de dades (mariadb) amb accés restringit només des del proxy i un servidor centralitzat de logs (syslog-ng) en UDP 514. La xarxa interna suporta IPv4 i IPv6 (fd00:4::/64), amb adreces com fd00:4::130 (http), fd00:4::131 (mysql) i fd00:4::132 (syslog). S'han afegit rutes estàtiques en Router D cap a les subxarxes DMZ i interna via Firewall Extern, de manera que el protocol RIP no necessita anunciar aquestes zones protegides.

És en aquesta xarxa on s'han trobat més incidències, derivades de casuístiques de Docker i de la inicialització dinàmica de taules d'enrutament:

- **Processos zombie en FRR:** els daemons *zebra* i *ripd* generaven processos zombie que bloquejaven la convergència RIP. S'ha solucionat integrant *tini* als *Dockerfiles* dels routers, actuant com a init mínim que adopta processos orfes i neteja zombies automàticament.
- **Assignació aleatòria d'interfícies:** Docker no garanteix l'ordre *eth0/eth1/eth2*, fet que provocava que *scripts* amb interfícies fixades fallessin. S'ha implementat detecció dinàmica d'interfícies mitjançant *awk*.
- **Bloqueig del CDN d'Alpine per l'ISP:** durant determinats esdeveniments l'ISP bloquejava dominis de Cloudflare —incloent *dl-cdn.alpinelinux.org*— degut

als bloquejos que genera la Lliga de futbol quan hi ha partit, impeding la descàrrega d'imatges i provocant bucles de reinicis. S'ha migrat la instal·lació de paquets als respectius *Dockerfiles*, convertint les imatges en autosuficients.

Per tant, s'ha assolit el Subobjectiu 2 (Xarxa Empresarial), ja que s'ha desplegat una arquitectura de defensa en profunditat amb doble tallafocs, una DMZ amb servidor proxy invers i serveis crítics a la xarxa interna (HTTP, MySQL, Syslog) amb suport *dual-stack IPv4/IPv6*. Aquesta configuració reflecteix l'estructura perimetral d'una xarxa corporativa real, on l'exposició dels actius interns a Internet es controla i segmenta mitjançant polítiques de filtratge granulars.

5.3 Xarxa de Serveis (N3)

S'ha desplegat un servei DNS amb *dnsmasq*, que combina les funcions de *forwarder* i servidor de cache, permetent la resolució de noms tant per a dominis locals del laboratori com per a consultes externes. Per a la VPN s'ha implementat *WireGuard*, que destaca per la seva eficiència en el processament de paquets al nucli de Linux i la simplicitat en la gestió de claus criptogràfiques, ideal per simular accessos remots segurs.

S'han resolt dues incidències rellevants. La primera, la sobreescritura del fitxer */etc/resolv.conf* pel daemon de Docker, que feia fallar silenciosament el client *udhcpd*; la solució va consistir a forçar l'escriptura dels paràmetres DNS mitjançant una redirecció directa un cop finalitzat *udhcpd*. La segona, l'error *REFUSED* en consultes *AAAA* que bloquejava *getaddrinfo()* (i, per tant, eines com *wget*): es va resoldre configurant *dnsmasq* com a autoritatiu per al domini local, de manera que respon amb *NXDOMAIN* per als registres *IPv6* no definits i permet que el client continuï amb la resolució *IPv4*.

Per tant, s'ha assolit el Subobjectiu 3 (Xarxa de Serveis), ja que s'ha desplegat un servidor DNS funcional amb capacitats de reenviament i memòria cau per a dominis locals i externs, i s'ha establert un túnel VPN xifrat punt a punt amb *WireGuard*. Ambdós serveis constitueixen la infraestructura fonamental de privacitat i resolució de noms sobre la qual recolzen la navegació segura i l'accés remot en entorns reals.

5.4 Atacs i vulnerabilitats (N4)

Abans d'implementar els atacs, es van adaptar diversos aspectes de la infraestructura: es va afegir una segona interfície de xarxa al contenidor de John dins la xarxa N1 (necessària per als atacs que requereixen presència a la mateixa xarxa que les víctimes), es va instal·lar la biblioteca *Scapy* via *pip3* (per evitar conflictes de dependències amb *apk* d'Alpine) i es va dotar tots els *scripts* d'un esquelet comú de daemonització operable des de l'amfitrió a través del script de control *netctrl.sh*.

- **Enverinament de taules d'enrutament (RIP poisoning):** el node *John* injecta anuncis *RIPv2* falsificats cap a l'adreça multicast *224.0.0.9*, anunciant rutes cap a subxarxes crítiques amb mètrica 1 i establint-se com a next hop. En aturar l'atac, envia els mateixos anuncis amb mètrica 16 (distància infinita) per forçar la reconvergència cap a les rutes legítimes en menys de 30 segons.
- **Suplantació d'IP (IP spoofing):** *John* genera paquets *ICMP* i segments *TCP SYN* amb adreça *IP* d'origen falsificada cap al *DNS* i el *Firewall Exterior*, demostrant que l'atribució d'identitat basada només en la *IP* d'origen és fàcilment manipulable i evidenciant la necessitat de solucions criptogràfiques (xifratge, signatures i certificats *PKI*).
- **MitM + ARP spoofing:** després d'un escaneig *ARP* de la xarxa *N1*, *John* enverina les memòries *cache ARP* d'*Alice*, *Bob* i *Router A*, mantenint actiu l'*ip_forwarding* per actuar com a node intermediari de manera transparent. En finalitzar, restaura les taules *ARP* amb paquets legítims.
- **DDoS multivectorial (SYN/ICMP/UDP flood):** tres fils en paral·lel generen inundació cap a actius crítics. Es va constatar el col·lapse temporal del *proxy*, amb un pic superior a 180 connexions simultànies en estat *SYN_RECV*, confirmant la saturació del *backlog* del sistema operatiu.
- **Enverinament de memòria cache DNS (DNS cache poisoning):** recreació de la vulnerabilitat de *Kaminsky* amb *dnsmasq* configurat amb port de sortida fix (5353) i *upstream* inexistent. *John* inunda amb 65.536 paquets *UDP* falsificats (un per cada valor de *TxID*) emprant raw sockets, aconseguint injectar una resposta fraudulenta amb *TTL* de 30 segons.
- **Injecció de tràfic via proxy (HTTP response hijacking):** simulant que *John* ha compromès el *proxy*, es reescriu *nginx.conf* substituint *proxy_pass* per un return 200 amb *HTML* controlat per l'atacant i s'aplica una recàrrega en calent. En finalitzar, l'script restaura el fitxer original des del *backup nginx.conf.bak*.

En el desenvolupament d'aquests vectors es van identificar limitacions de programari i protocol que es van resoldre: el mòdul *scapy.contrib.rip* no estava integrat a la distribució de *Scapy*, fet que es va resoldre codificant les capçaleres *RIPv2* de manera nativa amb *struct.pack*; *FRR* rebutjava la injecció de la ruta per defecte (0.0.0.0/0) i les actualitzacions en mode *unicast*, motiu pel qual es va reorientar l'atac cap a rutes específiques i es va forçar l'enviament multicast amb *sendp()*; i en el *MitM* es va afegir un filtre programàtic per restringir els objectius al rang *DHCP* (.10 a .99), descartant adreces internes del bridge de *Docker*.

Per tant, s'ha assolit el Subobjectiu 4 (Atacs i vulnerabilitats), ja que s'han implementat i verificat sis vectors d'atac sobre la infraestructura desplegada: enverinament de taules d'enrutament, suplantació d'IP, *MitM* amb *ARP spoofing*, *DDoS multivectorial*, enverinament de memòria *cache DNS* i segrest de respostes *HTTP* via *proxy*. Cada vector ha estat validat amb evidències quantitatives o captures de tràfic anotades, i les limitacions inherents a *Docker* han estat documentades i justificades amb alternatives viables.

5.5 Tests d'assoliment i QA

Per verificar que les tecnologies implementades funcionen i que els vectors d'atac produeixen els efectes documentats, s'ha dissenyat una capa de *Quality Assurance* en tres nivells: una bateria automatitzada de tests d'infraestructura, una bateria de tests d'atacs amb emissió de mètriques quantitatives i un conjunt de captures de tràfic anotades com a evidència visual.

El script *test-lab.sh* executa 65 verificacions automatitzades distribuïdes en tretze categories (Taula 2).

Taula 2: Resum de verificacions de QA per categoria

Categoria	Verificacions	Tipus comprovació
Estat dels contenidors	17	Docker
DHCP	2	isc-dhcp-server (router_a)
Enrutament dinàmic RIP	5	FRR ripd
Enrutament dinàmic OSPF	4	FRR ospfd
DNS	7	dnsmasq
NAT	2	iptables MASQUERADE
Flux HTTP complet	3	nginx + apache2
Seguretat perimetral	4	firewall.extern / firewall.intern
Serveis interns	7	apache2, nginx, mariadb, syslog-ng
MySQL	3	mariadb
Syslog	2	syslog-ng UDP/514
IPv6	6	pila dual-stack fd00::4::/64
VPN WireGuard	14	WireGuard wg0

El script *test-attacks.sh* cobreix els sis vectors d'atac implementats, estructurant cada verificació en tres fases seqüencials: inici de l'atac, comprovació de l'efecte observable i aturada amb retorn a l'estat legítim. La creació d'ambdós scripts forma part integral del cicle de cada *Sprint*: s'executen com a criteri de validació previ a la fusió de cada branca a main, garantint que cap canvi trenca la infraestructura existent.

A més de les verificacions binàries *PASS/FAIL*, el script *test-attacks.sh* emet mètriques quantitatives al fitxer *metrics.csv* (Taula 3). Cada mètrica temporal s'ha obtingut a partir de n=5 execucions independents, calculant la mitjana aritmètica com a valor representatiu i la *Desviació Absoluta Mediana (MAD)* com a indicador de dispersió. S'ha triat el *MAD* en lloc de la desviació estàndard perquè, amb mostres petites, un únic valor atípic —provocat per *jitter* del planificador de *Docker* o pics puntuals de càrrega— pot distorsionar significativament la desviació estàndard.

Els llindars de la Taula 3 s'han definit seguint dos criteris diferenciats. Per als vectors basats en protocols estàndard, els llindars deriven de les especificacions tècniques: els valors de *rip_poison_time* (≤40s) i

rip_post_withdrawal_verification_latency (≤ 30 s) es fonamenten en el temporitzador d'actualització de *RIPv2* definit a la *RFC 2453* (cicle de 30 segons, amb marge addicional per a la propagació entre nodes); el *kaminsky_poison_time* (≤ 30 s) es defineix com el temps màxim raonable per transmetre els 65.536 paquets *UDP* que cobreixen l'espai complet de *TxID* de *DNS* (2¹⁶). Per als vectors de capa d'enllaç i de saturació, els llindars s'han establert empíricament a partir de les observacions durant el desenvolupament.

Una mètrica requereix una nota semàntica explícita. En els 0.17s de *arp_manual_flush_time* mesuren el temps necessari per esborrar manualment l'entrada corrupta de la taula *ARP* i forçar que el dispositiu torni a demanar la *MAC* correcta. Ara bé, aquesta intervenció manual no és necessària en un escenari real: si ningú actua, l'entrada falsa roman activa durant aproximadament 60 segons fins que el sistema operatiu la descarta de manera automàtica. Això significa que l'atacant pot desconnectar-se i l'enverinament continua funcionant durant un minut, cosa que constitueix una vulnerabilitat rellevant que s'ha documentat explícitament a la Taula 3 com a *arp_natural_persistence*.

Table 3: Mètriques d'avaluació dels atacs

Escenari	Mètrica	Valor ($\pm$ MAD)	Unitat	Llindar	Resultat
rip	poison time	2,46 ($\pm 8,0$ ms)	s	≤ 40	PASS
arp	poison time	7,08 ($\pm 14,0$ ms)	s	≤ 10	PASS
	manual flush time	0,17 ($\pm 1,0$ ms)	s	≤ 10	PASS
	natural persistence (no intervention)	~ 60	s		INFO
flood	time to SYN_RECV saturation	8,48 ($\pm 9,0$ ms)	s	≤ 10	INFO
	peak SYN_RECV connections	366,00 ($\pm 6,0$)	connections		INFO
	recovery time	7,20 ($\pm 2,0$ ms)	s	≤ 10	PASS
dns	time to cache poisoning	2,47 ($\pm 6,0$ ms)	s	≤ 30	PASS
	packets sent (TxID flood)	65.536,00 ($\pm 0,0$)	packets	65.536	INFO

Aquesta capa de *QA* ha permès detectar errors de configuració i validar objectivament que els criteris d'assoliment definits a la secció 2.4 s'han satisfet. Els components que no es poden avaluar de manera quantitativa s'han documentat mitjançant captures de tràfic anotades que demostren gràficament el contingut, l'enviament i la recepció dels paquets.

5.6 Discussió de resultats

S'ha aconseguit desenvolupar un entorn de laboratori complet, complex i funcional que permet materialitzar de forma pràctica grans conceptes fonamentals de l'enginyeria de xarxes. L'arquitectura dissenyada il·lustra empíricament dinàmiques essencials com el descobriment i reconeixement de nodes en la capa d'enllaç, els fluxos de transport de dades interns, la interconnexió de topologies mitjançant passarel·les i l'encaminament dinàmic inter-nodes basat en taules d'encaminament.

El projecte posa de manifest la complexitat oculta d'arquitectures que utilitzem de manera quotidiana, com la resolució jeràrquica de noms de domini (*DNS*) o la implementació de seguretat i privacitat mitjançant túnels xifrats (*VPN*). Des d'una perspectiva de disseny, la infraestructura emula fidelment l'organització d'una xarxa domèstica —mitjançant la mitigació de l'escassetat

d'adreces *IPv4* públiques a través de *NAT*— i l'estructura perimetral d'una xarxa empresarial dissenyada per protegir els actius interns de l'exposició directa a Internet. Finalment, es consolida com un banc de proves ideal per a l'anàlisi de vectors d'atac clàssics que, malgrat la seva antiguitat, continuen representant un repte de seguretat crític si no es configuren les mitigacions adequades.

Els continguts pràctics i teòrics integrats convergeixen de manera directa amb el teixit docent d'assignatures troncales de la menció de *Tecnologies de la Informació (TI)*, com *Xarxes*, *Tecnologies de l'Arquitectura d'Internet (TAI)*, *Infraestructures de Tecnologies de la Informació (ITX)* i *Garantia de la Informació i Seguretat (GIS)*. També s'hi incorporen competències de *Gestió d'Administració de Xarxes (GAX)* que, tot i pertànyer a una altra menció, comparteixen la mateixa família de coneixement.

Cal destacar que, si bé l'exposició pràctica directa a vectors d'atac reals és sovint limitada durant el grau, el bagatge acadèmic adquirit ha proporcionat les competències clau en resolució de problemes i recerca bibliogràfica, indispensables per avaluar de forma crítica quins components tecnològics presenten el major valor didàctic. Tot i néixer a l'Escola d'Enginyeria, l'abast del laboratori no es limita a l'àmbit intern: s'ha estructurat com una plataforma oberta, ideal per a qualsevol professional o entusiasta que vulgui experimentar, analitzar protocols o adaptar i desplegar nous vectors d'atac i tecnologies emergents sobre el seu propi *stack* de coneixement. Esdevé, en essència, un entorn de simulació viu i escalable.

6 CONCLUSIONS

Les fites i els objectius establerts a l'inici del projecte s'han assolit completament dins dels terminis estimats. L'èxit en l'execució, malgrat la considerable complexitat d'un laboratori on convergeixen múltiples conceptes de xarxa i sistemes, és fruit d'una planificació rigorosa i d'una organització estructural orientada a minimitzar els riscos. Durant la fase d'integració final es van requerir lleugeres reestructuracions de la topologia no previstes inicialment, però fàcilment adaptables gràcies a l'estructura inicial sòlida del projecte.

En termes d'aportació concreta, aquest treball lliura una eina de laboratori funcional i autocontinguda per a la docència d'enginyeria de xarxes, estructurada en quatre escenaris interconnectats que permeten:

1. Analitzar i configurar tecnologies de xarxa domèstica, experimentant de primera mà el funcionament de *NAT* i *DHCP* en un entorn que simula la gestió d'adreces *IPv4* pròpia d'un *ISP* real.

2. Desplegar i comprendre arquitectures de seguretat empresarial, posant en pràctica la segmentació en DMZ, la defensa en profunditat amb doble tallafocs, el *proxy* invers, el servidor centralitzat de *logs* i la coexistència IPv4/IPv6.
3. Explorar serveis fonamentals d'Internet, configurant un servidor DNS amb reenviament i cache i establint túnels xifrats punt a punt amb *WireGuard*.
4. Estudiar i executar vectors d'atac clàssics de manera controlada —RIP poisoning, IP spoofing, ARP spoofing, DDoS multivectorial, DNS cache poisoning i segrest de respostes HTTP—.

Cadascun d'aquests escenaris cobreix els continguts d'assignatures troncal de la menció de *Tecnologies de la Informació* —Xarxes, TAI, ITX, GIS i GAX— amb tecnologies de codi obert plenament vigents en l'àmbit professional, garantint una transferència directa al mercat laboral.

Atès el caràcter modular i escalable de l'entorn, s'identifiquen les següents línies de treball futur, ordenades de major a menor prioritat segons la viabilitat tècnica dins de *Docker* i el valor docent aportat:

- **IDS/IPS:** integració d'un sistema de detecció i prevenció d'intrusions amb *Snort* o *Suricata*, tancant el cicle ofensiva-defensa amb els atacs de N4 com a casos de prova naturals [13][14].
- **PKI i TLS:** desplegament d'una infraestructura de clau pública amb una autoritat de certificació interna i HTTPS/TLS als serveis existents, completant la narrativa defensiva enfront de la suplantació d'identitat [15][16].
- **Monitorització:** incorporació de *Nagios*, *Zabbix* o *Prometheus+Grafana* per visualitzar en temps real l'impacte dels atacs de denegació de servei, complementant el servidor de *logs syslog-ng* ja desplegat [17].
- **Atacs IPv6:** ampliació dels escenaris de N4 amb vectors específics d'IPv6 com *NDP spoofing* o atacs *SLAAC*, aprofitant la infraestructura *dual-stack* ja operativa [18][19][20].

Cal tenir present, però, que l'ús de *Docker* com a plataforma de virtualització imposa limitacions estructurals que condicionen l'abast del laboratori tant pel que fa a tecnologies de xarxa com a vectors d'atac. En compartir el nucli de la màquina amfitriona i abstraure la capa d'enllaç mitjançant *bridges* virtuals, no és possible implementar tecnologies que requereixen un control real sobre la capa 2, com ara les xarxes de VLAN amb etiquetatge 802.1Q, la configuració de *trunks* entre

commutadors, el protocol *STP* (*Spanning Tree Protocol*) per a la prevenció de bucles, o l'agregació d'enllaços amb *LACP*. Aquestes tecnologies requeririen eines com *GNS3* amb imatges de commutadors reals, o entorns basats en hipervisors amb VMs actuant com a dispositius de xarxa complets. En el mateix sentit, l'aïllament parcial del kernel limita la reproducció d'atacs que necessiten un control total de la pila TCP/IP, com el segrest de rutes *BGP* (*BGP hijacking*), l'explotació de vulnerabilitats de kernel, o determinats vectors d'IPv6 que operen a nivell de trama *Ethernet*. En un entorn de virtualització completa com *KVM* o *VirtualBox*, on cada node disposa d'un sistema operatiu i una pila de xarxa independents, aquests escenaris serien plenament viables. Malgrat aquestes restriccions, *Docker* continua essent l'opció òptima per als objectius docents d'aquest laboratori, ja que el seu baix consum de recursos i la seva portabilitat permeten replicar l'entorn en qualsevol equip estàndard, maximitzant l'accessibilitat per a l'alumnat.

7. IA I LA SEVA FUNCIÓ EN EL DESENVOLUPAMENT

En el context actual, la integració de la Intel·ligència Artificial (IA) en el desenvolupament de projectes tecnològics és una realitat innegable. Tanmateix, per garantir la integritat acadèmica d'aquest treball, s'han definit unes pautes estrictes d'ús. La IA s'utilitza com una eina de suport i optimització de tasques repetitives, i no com un substitut del pensament crític o de l'autoria del projecte. El seu paper es fonamenta en els eixos següents:

7.1 Suport a la redacció i correcció lingüística

Tal com s'especifica en la normativa del treball, s'ha utilitzat la IA per supervisar la qualitat del text. Això inclou la millora de la redacció, l'adequació de l'estil acadèmic, la claredat expositiva i la correcció lingüística de la documentació i dels informes presentats. L'objectiu és garantir una comunicació professional i polida de les idees pròpies.

7.2 Assistència en el desenvolupament tècnic

En l'àmbit de la implementació, la IA actua com un assistent tècnic en diversos marcs d'actuació:

- **Assistència en codi:** Generació d'estructures base, expressions regulars (*regex*) i automatització de fragments de codi repetitius per agilitzar la configuració de l'entorn *Docker*. També a tingut un gran paper en el desenvolupament dels atacs de la xarxa N4, amb molt pocs coneixements de com portar a terme aquesta part de la pràctica a nivell pràctic, ha ajudat a comprendre la base d'aquests problemes i com es el pensament lateral d'aquestes vulnerabilitats i com sapiguer com funcionen aquests processos ajuda a prevenir uns altres.
- **Depuració (Debugging):** Identificació d'errors sintàctics o lògics, anàlisi de les causes d'errors de configuració de xarxa i revisió de bones pràctiques en l'escriptura de fitxers o *scripts*.

- **Disseny de proves:** Suggeriment de casos límit i generació de dades de prova per validar la robustesa de les subxarxes i els protocols implementats. També a sigut de molta ajuda per fer que la visualització dels resultats dels tests a través de la terminal.
- **Recerca i aprenentatge:** Suport en la comprensió de conceptes tècnics complexos extrets de la documentació oficial o de projectes de codi obert, així com l'anàlisi de referències bibliogràfiques per a la seva aplicació pràctica.

7.3 Validació i criteri propi

L'ús de la IA generativa es limita estrictament a una eina d'acceleració. En cap cas se n'accepten els resultats de forma automàtica; tota la informació o el codi generat per la IA és objecte de dubte metòdic, essent revisat, validat i adaptat per l'autor. El criteri de l'estudiant preval sempre per assegurar que el resultat final reflecteix un aprenentatge real i una execució conscient.

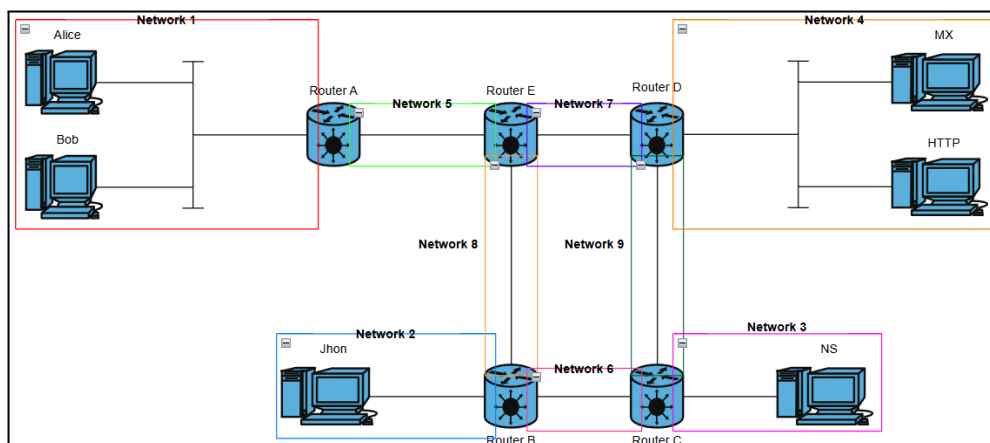
BIBLIOGRAFIA

- [1] Martínez Vidal, R. (2024). *Lab assignment 4: Advanced Networking concepts* [Material de curs]. Computer Networks, Curs 2024-2025.
- [2] Borrego, C., & Robles, S. (s.d.). *Tema 1: Xarxes Privades Virtuals* [Material de curs]. Tecnologies Avançades d'Internet.
- [3] Robles, S. (2022). *4 - Application Protocols* [Material de curs]. Computer Networks.
- [4] Suppi, R. (s.d.). *Tema 3: Seguridad en redes: Proxy, Proxy Server* [Material de curs]. Escola d'Enginyeria, Universitat Autònoma de Barcelona.
- [5] Suppi, R. (s.d.). *Tema 3.3: Seguridad en redes: Firewalls* [Material de curs]. Escola d'Enginyeria, Universitat Autònoma de Barcelona.
- [6] Robles, S., & Borrego, C. (s.d.). *Tema 5: Protocol d'Internet versió 6* [Material de curs]. Tecnologies Avançades d'Internet.
- [7] Suppi, R. (s.d.). *Tema 3.4: Seguridad en redes: VPN, IDS, Seguridad en WIFI* [Material de curs]. Escola d'Enginyeria, Universitat Autònoma de Barcelona.
- [8] Fortinet. (s.d.). *Top 20 most common types of cybersecurity attacks*. <https://www.fortinet.com/resources/cyberglossary/types-of-cyber-attacks>
- [9] Erickson, J. (2008). *Hacking: The art of exploitation* (2a ed.). No Starch Press.
- [10] Mirkovic, J., Dietrich, S., Dittrich, D., & Reiher, P. (2004). *Internet denial of service: Attack and defense mechanisms*. Prentice Hall PTR.
- [11] Stuttard, D., & Pinto, M. (2011). *The web application hacker's handbook* (2a ed.). Wiley.
- [12] Enginyeria del Software. (s.d.). *Tema 1: Principis de l'Enginyeria del Software* [Material de curs].
- [13] Bejtlich, R. (2013). *The practice of network security monitoring: Understanding incident detection and response*. No Starch Press.
- [14] Open Information Security Foundation. (2024). *Suricata documentation (v7)*. <https://docs.suricata.io>
- [15] Ristić, I. (2022). *Bulletproof TLS and PKI* (2a ed.). Feisty Duck.
- [16] Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3* (RFC 8446). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc8446>
- [17] Brazil, B. (2018). *Prometheus: Up & running: Infrastructure and application performance monitoring*. O'Reilly Media.
- [18] Hogg, S., & Vyncke, E. (2009). *IPv6 security*. Cisco Press.
- [19] Narten, T., Nordmark, E., Simpson, W., & Soliman, H. (2007). *Neighbor Discovery for IP version 6 (IPv6)* (RFC 4861). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc4861>
- [20] Chown, T., & Venaas, S. (2011). *Rogue IPv6 Router Advertisement*

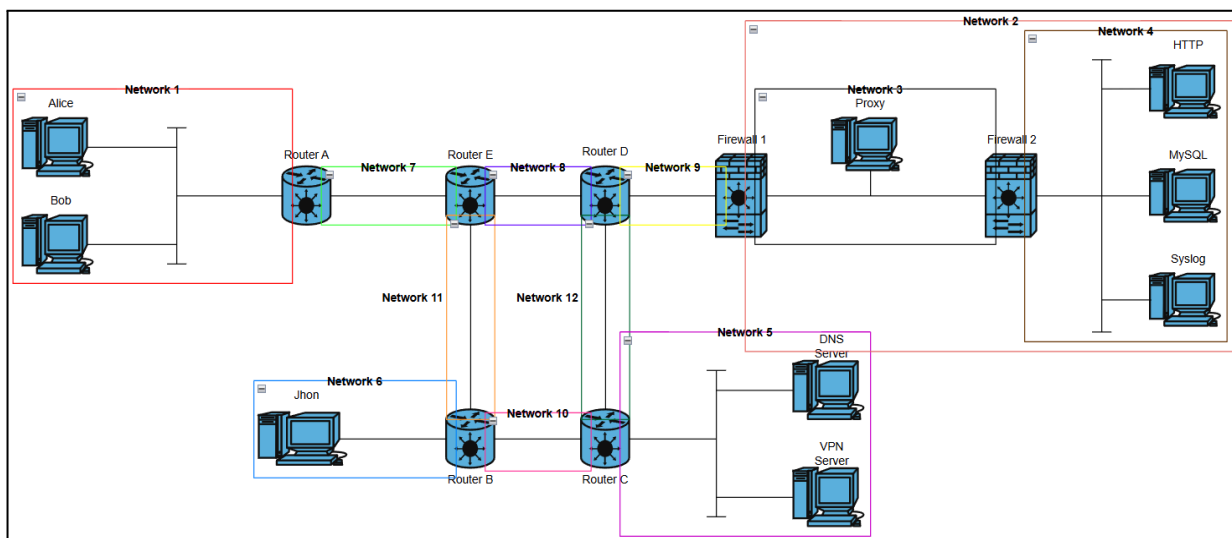
problem statement (RFC 6104). Internet Engineering Task Force. <https://www.rfc-editor.org/rfc/rfc6104>

APÈNDIX

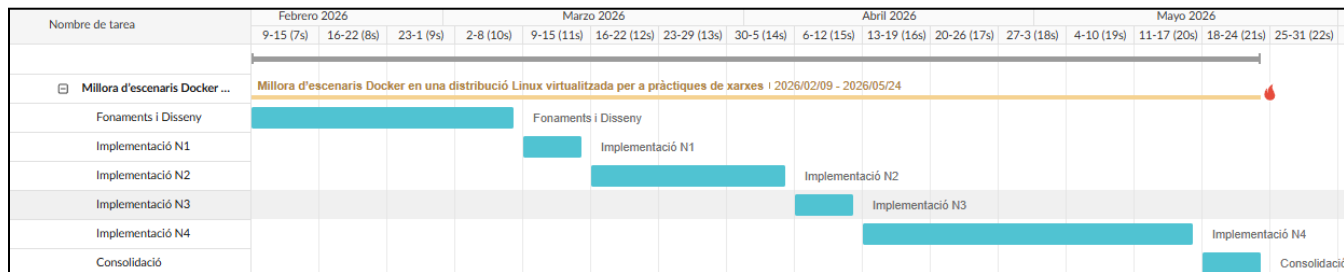
A1. Fig. 1. Diagrama de la infraestructura de xarxa de l'entorn Docker inicial.



A2. Fig. 2. Diagrama de la infraestructura de xarxa de l'entorn Docker amb les millores implementades.



A3. Diagrama de Gantt amb la temporització i planificació de les fases del projecte



A4. Taula 2: Resum de verificacions de QA per categoria

Taula 2: Resum de verificacions de QA per categoria

Categoria	Verificacions	Tipus comprovació
Estat dels contenidors	17	Docker
DHCP	2	isc-dhcp-server (router_a)
Enrutament dinàmic RIP	5	FRR ripd
Enrutament dinàmic OSPF	4	FRR ospfd
DNS	7	dnsmasq
NAT	2	iptables MASQUERADE
Flux HTTP complet	3	nginx + apache2
Seguretat perimetral	4	firewall_extern / firewall_intern
Serveis interns	7	apache2, nginx, mariadb, syslog-ng
MySQL	3	mariadb
Syslog	2	syslog-ng UDP/514
IPv6	6	pila dual-stack fd00:4::/64
VPN WireGuard	14	WireGuard wg0

A5. Taula 3: Mètriques d'avaluació dels atacs

Table 3: Mètriques d'avaluació dels atacs

Escenari	Metrica	Valor ($\pm$ MAD)	Unitat	Llindar	Resultat
rip	poison time	2,46 ($\pm$ 8,0 ms)	s	≤ 40	PASS
arp	poison time	7,08 ($\pm$ 14,0 ms)	s	≤ 10	PASS
	manual flush time	0,17 ($\pm$ 1,0 ms)	s	≤ 10	PASS
	natural persistence (no intervention)	~ 60	s	—	INFO
flood	time to SYN_RECV saturation	8,48 ($\pm$ 9,0 ms)	s	≤ 10	INFO
	peak SYN_RECV connections	366,00 ($\pm$ 6,0)	connections	—	INFO
	recovery time	7,20 ($\pm$ 2,0 ms)	s	≤ 10	PASS
dns	time to cache poisoning	2,47 ($\pm$ 6,0 ms)	s	≤ 30	PASS
	packets sent (TxID flood)	65.536,00 ($\pm$ 0,0)	packets	65.536	INFO