TecnoCampus

Centre universitari adscrit a la

upf. Universitat Pompeu Fabra *Barcelona*

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

Desenvolupament de laboratoris de ciberseguretat

Memòria final

Bruno Lucena Siles
Tutor: Dr. Pere Tuset Peiró

Maig 2025

Agraïments

Vull agrair sobretot al meu tutor, Pere Tuset-Peiró, per fer possible la realització d'aquest projecte. La seva experiència i coneixement han siguin fonamentals pel desenvolupament d'aquest treball.

Resum

Aquest Treball de Final de Grau té com a objectiu el disseny i desenvolupament d'un conjunt de laboratoris pràctics de ciberseguretat, orientats a l'assignatura optativa d'Administració i Seguretat de Sistemes del grau en Enginyeria Informàtica. El projecte neix de la necessitat d'oferir recursos educatius que permetin als estudiants experimentar en un entorn controlat, integrant coneixements teòrics amb la pràctica directa.

Els laboratoris dissenyats aborden àrees fonamentals com la criptografia clàssica, la criptografia moderna, l'exfiltració de dades i l'ús d'eines OSINT com Shodan i Censys. Cada laboratori inclou activitats tècniques específiques que fomenten la comprensió de conceptes com el xifrat simètric i assimètric, la integritat de la informació, la detecció de vulnerabilitats i les tècniques d'atac i defensa en xarxes.

Per al seu desenvolupament s'ha aplicat una metodologia iterativa-incremental amb un enfocament àgil, permetent adaptar el projecte a les necessitats emergents. Els laboratoris s'han implementat amb llenguatge Python, utilitzant llibreries específiques segons l'objectiu de cada activitat.

Els resultats obtinguts han mostrat un bon nivell d'acceptació i utilitat didàctica, reforçant la rellevància del projecte com a eina formativa per a futurs professionals en ciberseguretat.

Resumen

Este Trabajo de Fin de Grado tiene como objetivo el diseño y desarrollo de un conjunto de laboratorios prácticos de ciberseguridad, orientados a la asignatura optativa de Administración y Seguridad de Sistemas del grado en Ingeniería Informática. El proyecto nace de la necesidad de ofrecer recursos educativos que permitan a los estudiantes experimentar en un entorno controlado, integrando conocimientos teóricos con la práctica directa.

Los laboratorios diseñados abordan áreas fundamentales como la criptografía clásica, la criptografía moderna, la exfiltración de datos y el uso de herramientas OSINT como Shodan y Censys. Cada laboratorio incluye actividades técnicas específicas que fomentan la comprensión de conceptos como el cifrado simétrico y asimétrico, la integridad de la información, la detección de vulnerabilidades y las técnicas de ataque y defensa en redes.

Para su desarrollo se ha aplicado una metodología iterativa-incremental con un enfoque ágil, permitiendo adaptar el proyecto a necesidades emergentes. Los laboratorios se han implementado con el lenguaje Python, utilizando librerías específicas según el objetivo de cada actividad.

Los resultados obtenidos han mostrado un buen nivel de aceptación y utilidad didáctica, reforzando la relevancia del proyecto como herramienta formativa para futuros profesionales en ciberseguridad.

Summary

This Final Degree Project aims to design and develop a set of practical cybersecurity laboratories, focused on the optional subject ”System Administration and Security”within the Computer Engineering degree. The project arises from the need to offer educational resources that allow students to experiment in a controlled environment, integrating theoretical knowledge with hands-on practice.

The developed labs cover key areas such as classical cryptography, modern cryptography, data exfiltration, and the use of OSINT tools like Shodan and Censys. Each lab includes specific technical activities to promote understanding of concepts such as symmetric and asymmetric encryption, data integrity, vulnerability detection, and network attack and defense techniques.

An iterative-incremental methodology with an agile approach has been applied for development, allowing the project to adapt to emerging needs. The labs have been implemented in Python, using specific libraries according to the objective of each activity.

The results obtained show a good level of acceptance and didactic usefulness, reinforcing the value of the project as a training tool for future cybersecurity professionals.

Índex

Índex de figures

Glossari

Atacs de força bruta

Tècnica de criptoanàlisi que consisteix a provar sistemàticament totes les combinacions possibles d'una clau o contrasenya fins a trobar la correcta. Aquest tipus d'atac es basa en la força computacional i no en debilitats del sistema, fent-lo eficaç però molt costós en termes de temps i recursos.

CVE

Sigles de Common Vulnerabilities and Exposures, un sistema públic d'identificació estandarditzada de vulnerabilitats en programari i maquinari. Cada CVE inclou un codi únic i una descripció, i permet als professionals de la ciberseguretat referenciar i gestionar vulnerabilitats de manera consistent.

Denial of Service (Denegació de servei)

Tipus d'atac que busca fer inoperatiu un sistema, servei o xarxa sobrecarregant-lo amb peticions o trànsit maliciós. L'objectiu és impedir que els usuaris legítims hi puguin accedir o utilitzar-lo correctament, causant una interrupció temporal o permanent del servei..

Distributed Denial of Service (Denegació de servei distribuït)

Tipus d'atac de denegació de servei en què s'utilitza una xarxa distribuïda de dispositius compromesos (coneguts com a botnet) per enviar grans volums de trànsit de manera simultània contra un objectiu. Això fa més difícil detectar i mitigar l'atac, i n'augmenta la potència i l'efectivitat..

DNS

Sigles de Domain Name System, sistema que tradueix noms de domini llegibles per humans (com `example.com`) a adreces IP que poden ser interpretades pels dispositius de xarxa. Actua com una "agenda" d'Internet per facilitar la navegació..

Enllaços d'hipertext

Els enllaços d'hipertext són elements d'un document digital que permeten navegar entre diferents pàgines o seccions mitjançant un clic, connectant informació de manera no lineal.

HTTP

Sigles d'Hypertext Transfer Protocol, protocol utilitzat per transferir dades a través de la web. Permet la comunicació entre clients (com navegadors) i servidors per carregar pàgines web i altres recursos mitjançant peticions i respostes..

HTTPS

Escala de valoració utilitzada habitualment en enquestes per mesurar el grau d'acord o desacord amb una afirmació. Es basa en una sèrie de punts ordenats (generalment entre 5 i 7) que permeten quantificar opinions, actituds o percepcions..

ICMP

Sigles d'Internet Control Message Protocol, protocol utilitzat principalment per enviar missatges de control, notificació d'errors i comprovació de l'estat de la connexió entre dispositius en una xarxa. Un ús comú és la comanda `ping`, que serveix per verificar la connectivitat amb un altre dispositiu.

INCIBE (Instituto Nacional de Ciberseguridad)

Sigles de l'Institut Nacional de Ciberseguretat d'Espanya, organisme públic encarregat de promoure la ciberseguretat en l'àmbit ciutadà, empresarial i institucional. Ofereix serveis d'assistència, formació i conscienciació, i desenvolupa accions per protegir la societat davant riscos i amenaces digitals..

Likert

Escala de valoració utilitzada habitualment en enquestes per mesurar el grau d'acord o desacord amb una afirmació. Es basa en una sèrie de punts ordenats (generalment entre 5 i 7) que permeten quantificar opinions, actituds o percepcions..

Màquina virtual

Entorn de computació simulat per programari que imita el funcionament d'un ordinador físic, permetent executar sistemes operatius i aplicacions de manera aïllada dins d'un altre sistema..

Ofuscades

Significa que una informació ha estat alterada o encriptada de manera que sigui difícil d'entendre per humans o sistemes sense la clau adequada, utilitzant tècniques per ocultar el seu contingut original.

Pings

Els pings formen part del protocol ICMP els quals són missatges de sol·licitud i resposta enviats per comprovar la connectivitat entre dos dispositius en una xarxa, mesurant el temps de resposta i detectant possibles problemes de comunicació.

Protocols de xarxa

És un conjunt de regles i normes que defineixen com es comuniquen els dispositius dins d'una xarxa. Aquests protocols estableixen les normes per a l'enviament, la recepció i la interpretació de dades entre ordinadors, servidors, routers i altres dispositius connectats.

Taula ASCII

La taula ASCII (American Standard Code for Information Interchange) és una taula que assigna valors numèrics a cada caràcter de text. Aquest sistema de codificació permet que els ordinadors interpretin i representin caràcters com lletres, números, signes de puntuació i símbols.

Threads

En informàtica, un fil d'execució (thread en anglès) és la unitat més petita de processament que pot ser programada pels sistemes operatius, i que permet a un procés executar diferents tasques al mateix temps..

VPN

Xarxa privada virtual que crea un canal segur de comunicació sobre una xarxa pública com Internet. Mitjançant tècniques de xifrat, una VPN permet protegir les dades transmeses, ocultar la ubicació de l'usuari i accedir a recursos de xarxes internes de manera remota..

1 Objecte del projecte

L'objectiu del present projecte és el disseny i desenvolupament d'un conjunt de laboratoris de ciberseguretat orientats a l'assignatura optativa Administració i Seguretat de Sistemes, de 4t curs del Grau en Enginyeria Informàtica en Sistemes d'Informació (GEISI). Aquestes activitats estan pensades per a ser implementades durant el tercer trimestre del curs acadèmic 2024-25, amb la finalitat que els estudiants adquireixin coneixements fonamentals en àrees clau de la protecció de sistemes informàtics, comunicacions i dades.

Els laboratoris es duran a terme en un entorn controlat que permetrà als estudiants experimentar amb tècniques d'atac i defensa, millorant la comprensió dels riscos associats a la seguretat digital i fomentant l'adopció de bones pràctiques.

L'objectiu final és proporcionar recursos d'aprenentatge innovadors que facilitin la preparació dels estudiants per afrontar reptes reals en la protecció d'infraestructures digitals. Així mateix, el projecte busca fomentar el desenvolupament de professionals amb capacitats pràctiques i conceptuals en ciberseguretat, un àmbit de creixent rellevància en l'era digital.

1.1 Motivació

El present Treball Final de Grau neix de la necessitat creixent d'entorns controlats per a la formació pràctica en ciberseguretat. En un món cada vegada més digitalitzat, on les amenaces informàtiques evolucionen constantment, resulta imprescindible comptar amb professionals ben preparats que puguin fer front als reptes actuals i futurs en matèria de seguretat digital. La motivació principal d'aquest projecte és desenvolupar un conjunt de laboratoris de ciberseguretat que permeti simular escenaris reals d'atacs informàtics en un entorn segur i controlat. Aquest tipus d'infraestructura és fonamental per diverses raons:

- Permet als estudiants aprendre mitjançant la pràctica directa, consolidant els coneixements teòrics adquirits durant el grau.
- Facilita l'experimentació amb diferents vectors d'atac i tècniques de defensa sense riscos legals ni afectacions a sistemes reals.
- Contribueix a la recerca en ciberseguretat, possibilitant l'anàlisi de vulnerabilitats i el desenvolupament de contramesures.

El projecte també neix de la detecció d'una mancança en els recursos educatius actuals del centre en matèria de ciberseguretat. L'ensenyament teòric de la ciberseguretat no sempre va acompanyat de la corresponent pràctica per limitacions, tant de temps com de recursos. Amb aquest TFG, es pretén contribuir a omplir aquest buit i oferir un conjunt d'eines i laboratoris pràctics valuosos per enfortir tant a la docència com la recerca.

Resumint, aquest treball busca apropar la pràctica real de la ciberseguretat als estudiants, creant una unió entre la teoria acadèmica i l'aplicació professional. D'aquesta manera, permetrà formar especialistes en aquest àmbit millor preparats per afrontar els reptes de seguretat digital que enfronten les organitzacions actualment i en el futur.

2 Estat de l'art: context, antecedents i necessitats d'informació

A continuació, s'analitzen en profunditat els antecedents del present projecte, així com els conceptes bàsics necessaris per entendre el procediment que es pretén seguir i poder iniciar el desenvolupament detallat del projecte.

2.1 Context

Actualment, la ciberseguretat és una prioritat per a organitzacions de tots els sectors. Amb l'augment constant d'atacs i bretxes de seguretat, és essencial comptar amb eines i professionals capaços de protegir els sistemes informàtics i les dades que gestionen. La necessitat d'aquestes mesures es fa evident si es compara l'evolució de la connectivitat global: mentre que l'any 2005 hi havia aproximadament mil milions de persones connectades a Internet, el 2023 aquesta xifra s'ha multiplicat per cinc. Aquest creixement ha anat acompanyat d'un increment proporcional en el nombre de ciberatacs, que ja formen part del dia a dia [1].

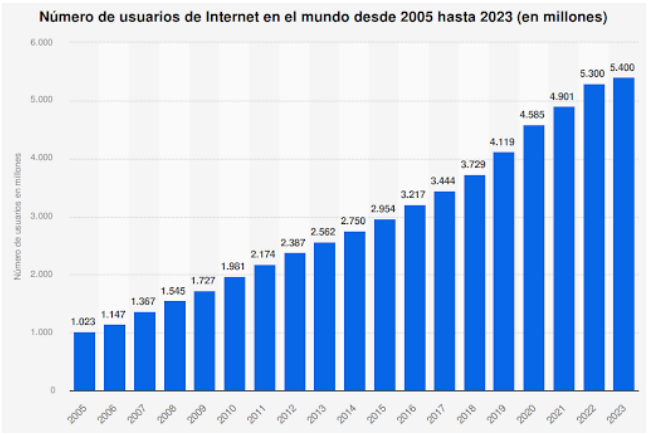

Figura 2.1: Registre dels usuaris connectats al món 2005-2023.

Font: https://es.statista.com/estadisticas/541434/numero-mundial-de-usuarios-de-internet/

No només els dispositius connectats a la xarxa són vulnerables; fins i tot elements físics com les memòries USB (Universal Serial Bus) poden esdevenir vectors d'atac. Un exemple significatiu es va produir l'any 2008, quan el Departament de Defensa dels Estats Units va patir un ciberatac després que un soldat, destinat a l'Orient Mitjà, connectés una memòria USB recollida en un aparcament a un ordinador militar vinculat a la xarxa central [2].

El dispositiu havia estat col·locat intencionadament per un agent estranger i contenia un malware conegut com Agent.BTZ [3].

Aquest programari maliciós va permetre l'accés no autoritzat a les xarxes militars, facilitant l'exfiltració de dades i obrint una porta d'entrada remota per als atacants. El malware es va propagar de manera silenciosa, afectant diversos sistemes connectats.

Davant d'aquest tipus d'escenaris, es considera fonamental establir polítiques estrictes de control de dispositius externs, formar els usuaris en bones pràctiques de seguretat i implementar plans de contingència per prevenir i mitigar possibles ciberatacs.

Tot i saber l'anterior xifra, també s'ha d'entendre que existeix una gran diversitat d'aquest tipus d'atacs on cal saber les seves diferències per tal de no confondre'ls [4]:

- **Atacs de xarxa**: Els atacs a la xarxa són intents malintencionats per entrar sense autorització a la xarxa d'una organització. Hi ha dues categories principals: passius i actius. Els atacs passius consisteixen a interceptar i monitoritzar informació confidencial sense alterar les dades, mentre que els atacs actius consisteixen a accedir a les dades i modificar-les, podent causar danys. Als atacs a la xarxa, l'objectiu és trencar el perímetre de la xarxa corporativa i obtenir accés als sistemes interns. Alguns exemples són [5]:
  - *Man-in-the-Middle (MitM)*: Atac on s'intercepta informació la comunicació entre dues parts per tal d'espiar o modificar dades entre altres accions.
  - *Denial of Service (DoS) i Distributed Denial of Service (DDoS)*: Atac on es vol saturar un sistema o una xarxa per deixar-la inoperativa.
- **Atacs a aplicacions i serveis web**: Un atac web es produeix quan un actor maliciós aprofita les vulnerabilitats d'un lloc web per obtenir accés no autoritzat, obtenir informació confidencial, introduir contingut maliciós o alterar el contingut del lloc web. A l'estar constantment en línia, també les fa constantment vulnerables als atacs. A més, com que sempre han d'estar accessibles públicament, no poden protegir-se mitjançant tallafocs convencionals. Per aquest motiu és imprescindible que les aplicacions i serveis web implementin mesures de seguretat eficaces per defensar-se contra aquestes amenaces. Alguns exemples són:
  - *SQL Injection*: Inserció de codi SQL maliciós en aplicacions vulnerables per accedir a informació de bases de dades.
  - *Cross-Site Scripting (XSS)*: Inserció de scripts o fragment de codi maliciós en llocs web per afectar a altres usuaris.

- **Atac de malware**: Programari maliciós que s'instal·la en un dispositiu sense el coneixement de l'usuari. Pot incrustar-se en el codi, propagant-se a altres programes útils o transmetent-se a través d'Internet per comprometre la seguretat del sistema. Alguns exemples són:
  - *Virus*: Programa que infecten altres fitxers per propagar-se i causar danys.
  - *Trojan*: Programes que semblen ser legítims i inofensius però, que amaguen funcionalitats malicioses.
- **Enginyeria social**: Tipus d'atac que explota la interacció humana i les emocions per manipular l'objectiu. L'atacant enganya la víctima per obtenir informació sensible o comprometre la seguretat d'un entorn virtual. Normalment, aquest tipus d'atac segueix diverses fases [6]:
  1. **Recopilació d'informació**: L'atacant investiga l'objectiu per identificar vulnerabilitats o detalls útils.
  2. **Construcció de confiança**: Mitjançant accions o comunicacions, l'atacant es guanya la confiança de la víctima.
  3. **Manipulació final**: S'utilitza la confiança obtinguda per convèncer la víctima de compartir informació confidencial o trencar les polítiques de seguretat.

  Alguns exemples són:
  - *Phishing*: Engany mitjançant correus electrònics o llocs web falsos per robar credencials o dades.
  - *Spear Phising*: Atacs de phishing dirigits específicament a una persona o una organització.
- **Atacs a dispositius i IoT**: Ciberatac que aprofita les vulnerabilitats de seguretat dels dispositius IoT per obtenir accés no autoritzat a les dades confidencials dels usuaris. Els atacants solen instal·lar malware en aquests dispositius, manipular la seva funcionalitat o explotar els seus punts dèbils per accedir a informació important d'una organització o empresa. Alguns exemples són:
  - *Botnets*: Xarxes de dispositius infectats utilitzats per llançar atacs coordinats.
  - *Side-Channel Attacks*: Explotació de fuites d'informació en dispositius físics.

- **Atacs criptogràfics**: Són intents maliciosos de comprometre la seguretat dels sistemes criptogràfics, amb l'objectiu d'explotar vulnerabilitats i obtenir accés no autoritzat a informació confidencial. Aquests atacs suposen una amenaça important per la confidencialitat, integritat i disponibilitat de les dades xifrades [7]. Alguns exemples són:
  - *Criptoanàlisi*: Intent de trencar xifrats per accedir a informació protegida.
  - *Brute force*: Intent sistemàtic de provar totes les combinacions possibles de contrasenyes.
  - *Replay Attacks*: Reutilització de dades interceptades en comunicacions per accedir a sistemes.

Així doncs, davant la creixent amenaça, és essencial que els professionals i futurs experts en tecnologia disposin d'eines per entendre i afrontar els riscos digitals. Tot i això, moltes formacions teòriques no ofereixen espais on aplicar aquests coneixements de manera pràctica. Aquest projecte neix amb la voluntat d'omplir aquest buit, proporcionant un entorn d'aprenentatge actiu on els participants puguin experimentar amb situacions simulades de ciberseguretat, comprendre els atacs més comuns i aprendre a defensar-se'n amb plans de contingència.

2.2 Antecedents

En l'actualitat, existeixen una gran quantitat d'institucions que ofereixen cursos per tal de formar tant a treballadors com a la població en general en pràctiques per minimitzar els possibles riscos a les infraestructures o a la mateixa informació. Els exemples més propers en el marc nacional són cursos que ofereixen tant entitats públiques com privades, com els programes de Formació Professional, els Graus Universitaris, entre d'altres.

D'altra banda, també coexisteixen altres espais com les plataformes virtuals, que proporcionen informació i casos pràctics per preparar als usuaris que consumeixen els seus continguts.

En aquest context, destaquen casos com *Hack The Box* i *TryHackMe* a escala global, així com iniciatives nacionals com les de *l'INCIBE (Instituto Nacional de Ciberseguridad)*. INCIBE és un organisme públic d'Espanya encarregat de contribuir amb un conjunt d'accions que protegeixin tant ciutadans com empreses i organismes públics. Una de les accions que ofereix és la formació i la disponibilitat de recursos tant per a professionals de les tecnologies de la informació com per al públic general [8] [9].

La importància d'aquest tipus de formació es fa evident davant l'augment d'atacs cibernètics que poden comprometre la seguretat global. Un cas amb bastant impacte va ser el de Stuxnet, un malware descobert el 2010 a la planta nuclear de Natanz, a l'Iran. Aquest cuc informàtic, detectat inicialment per inspectors de l'Agència Internacional d'Energia Atòmica, no buscava extreure informació, sinó sabotejar les centrifugadores del reactor nuclear. Fins i tot, els tècnics de la central que reemplaçaven les màquines es van trobar desconcertats en el moment del descobriment [10].

La causa d'aquest sabotatge, el cuc informàtic anomenat Stuxnet, no va ser descoberta fins cinc mesos després, quan es va repetir el mateix problema. Durant la investigació sobre com aquest malware podria haver arribat al sistema, es va divulgar que probablement va penetrar a través d'una memòria USB infectada, inserida en un ordinador connectat a la xarxa. Stuxnet es va propagar per la xarxa fins a trobar i comprometre el programari que controlava les centrifugadores. Aquest malware alterava el funcionament de les màquines, provocant que unes 1.000 acabessin desintegrant-se [11].

Per evitar deteccions, Stuxnet va simular dades falses als sistemes de registre de les centrifugadores, enganyant els tècnics sobre l'estat de les màquines. L'atac, presumptament orquestrat pels governs dels Estats Units i Israel, tenia com a objectiu frenar el programa nuclear iranià, generant tensions i posant de manifest la vulnerabilitat de les infraestructures crítiques.

Aquest cas destaca la necessitat creixent de professionals especialitzats en ciberseguretat. La sofisticació de Stuxnet subratlla la importància d'iniciatives formatives pràctiques i avançades que permetin anticipar i combatre aquest tipus d'amenaces en el futur.

Tenint en compte aquests casos i la documentació disponible en diverses plataformes i laboratoris, es va decidir desenvolupar un conjunt de laboratoris per establir una base sòlida i facilitar el disseny de futurs laboratoris enfocats a la ciberseguretat.

2.3 Estat de l'art

En aquest apartat es presenta la investigació realitzada per identificar les eines i tecnologies actuals en l'àmbit de la ciberseguretat. Aquestes eines poden oferir pràctiques i coneixements valuosos tant per a estudiants com per a qualsevol persona interessada en aquesta disciplina.

Es detallen les funcionalitats i utilitats principals de cada eina, així com el seu enfocament específic en el desenvolupament de laboratoris pràctics, orientats a la formació i la resolució de problemàtiques reals relacionades amb la ciberseguretat.

2.3.1 Investigació

En primer lloc, abans de determinar les eines o tecnologies a utilitzar en els laboratoris, s'ha realitzat una investigació entre totes les possibilitats que existien a Internet que fossin viables i a l'abast del meu coneixement entre tota la documentació que hi ha penjada a Internet. Sobre les eines que es van investigar, les que oferien més possibilitats de poder ser viables i que a la vegada, fossin prou interessats per gent que vol iniciar-se en el món de la ciberseguretat van ser:

- **Criptografia**: Eina que ha sigut utilitzada des del segle V a.C pels romans. Aquesta tecnologia és encarregada d'estudiar i idear mètodes per enviar informació de tal manera que només el destinatari o el receptor de la informació pugui ser l'únic que pugui veure aquesta. De manera més tècnica, el principal objectiu és proporcionar a dos elements que poden ser persones o màquines, puguin establir una comunicació de manera segura per un canal de comunicació que possiblement no és fiable, per tal que una entitat no pugui accedir a la informació que es vol transmetre entre les dues parts [12].
- **Phishing**: És un atac on un atacant fa ús de tècniques d'enginyeria social cap a un usuari simulant que és una entitat legítima com pot ser un banc, una xarxa social, etc. amb l'objectiu de robar-li informació privada i així aprofitar-se d'ell econòmicament.

- **Exfiltració de dades**: Consisteix en l'acte maliciós de transferir dades des d'un dispositiu com un ordinador o un mòbil a una ubicació externa sense autorització.

  Aquest acte pot ser realitzat per actors interns d'una organització, com empleats descontents o bastant negligents, o per atacants que han aconseguit infiltrar-se en la xarxa. Les vies per transferir-les poden variar des de simples tàctiques com l'ús d'unitats USB fins a tècniques sofisticades que implica l'ús de software maliciós i atacs cibernètics molt avançats.

Posteriorment, després d'investigar i veure tota la documentació penjada, es va concloure en realitzar els següents conjunts de laboratoris:

- **Criptografia clàssica**
- **Criptografia moderna**
- **Exfiltració de dades**
- **OSINT i vulnerabilitats CVE**

Aquesta sèrie de laboratoris proporciona a l'estudiant o individu que els realitzi una base sòlida i un conjunt ampli de coneixements en l'àmbit de la ciberseguretat. A més, fomenta la seva curiositat i motivació per continuar aprofundint en la matèria, explorant i aprenent de manera autònoma a través de recursos addicionals disponibles a la xarxa.

Criptografia clàssica

L'objectiu principal de la criptografia és enviar dades que siguin "secretes" i que cap entitat que no sigui l'emissor o el receptor puguin veure-les. Per fer-ho possible, cal encriptar aquestes dades, és a dir, transformar-les en un format que no sigui llegible de manera evident ni per una persona ni per una màquina. A més, per dur-lo a terme, cal decidir una manera, una tècnica o una "clau" que només han de saber tant l'emissor com el receptor.

Els motius principals per seleccionar el laboratori de criptografia clàssica i criptoanàlisi són els següents:

- **Fonaments clau en la ciberseguretat**. La criptografia és l'element imprescindible en qualsevol transmissió i emmagatzemament de dades.
- **Diversitat de possibles creacions**. Aquesta elecció et dona un abast bastant ampli de possibilitats per experimentar amb altres tecnologies com poden ser protocols de comunicació, mètodes d'intercanvi de claus, tipus de xifrats, etc.
- **Aplicacions pràctiques**: Ajuda els estudiants a entendre com es protegeixen dades sensibles en sistemes moderns com HTTPS o VPNs.

Tenint en compte aquests avantatges, és fonamental explicar les bases de la criptografia per entendre com funciona actualment i així que un futur s'entenguin millor tècniques o algoritmes avançats. Per aquest motiu, en aquest laboratori s'estudien algunes de les primeres tècniques criptogràfiques desenvolupades per la humanitat per ocultar informació i evitar que tercers no autoritzats la poguessin llegir.

A més d'aquestes tècniques, també s'introdueix la criptoanàlisi. El criptoanàlisi consisteix en l'estudi dels sistemes criptogràfics amb l'objectiu de trobar vulnerabilitats que permetin trencar la seva seguretat sense conèixer la seva clau secreta. A partir dels algoritmes que es treballin, s'intenta realitzar la criptoanàlisi d'aquests per tal d'intentar trencar-los.

Criptografia moderna

A l'anterior laboratori s'han explicat les bases de la criptografia i com funciona la criptoanàlisi. En aquest s'explica com ha evolucionat la criptografia i quines són algunes de les tècniques que s'utilitzen avui dia o que es feien servir fa un temps, des de l'existència dels primers ordinadors. Amb l'inici de l'existència d'aquests, també va sorgir l'aparició de dues categories de criptografia: la criptografia simètrica i l'asimètrica.

La criptografia simètrica utilitza una única clau per xifrar i desxifrar la informació. Això significa que tant l'emissor com el receptor han de compartir aquesta clau de manera segura abans de començar a comunicar-se. Aquesta tècnica és molt ràpida i eficient, la qual cosa la fa ideal per xifrar grans volums de dades, com fitxers o transmissions en temps real. No obstant això, la seva principal limitació és la necessitat de protegir la clau compartida, ja que si un tercer l'intercepta, pot desxifrar tota la informació.

És cert que la criptografia simètrica ja s'utilitzava abans que es definís formalment aquest concepte. Tanmateix, els sistemes emprats en aquell moment eren molt més senzills i, en comparació amb els estàndards actuals, presentaven vulnerabilitats significatives. La majoria d'aquells algoritmes serien fàcilment trencats pels ordinadors actuals, sigui mitjançant atacs per força bruta o tècniques de criptoanàlisi més avançades.

D'altra banda, la criptografia asimètrica utilitza un parell de claus: una clau pública i una clau privada. La clau pública es fa servir per xifrar el missatge, però només la clau privada corresponent pot desxifrar-lo. Això elimina la necessitat de compartir claus secretes, ja que només la clau pública és accessible a tothom. Tot i ser més segura per a intercanvis de claus i autenticació, la criptografia asimètrica és més lenta i menys eficient per xifrar grans volums de dades.

Finalment, un altre concepte utilitzat tant en la criptografia simètrica com en l'asimètrica és la funció de hash. Una funció de hash és un algorisme matemàtic que transforma un conjunt de dades en una representació alfanumèrica de longitud fixa. Aquesta representació, anomenada resum o digest, és pràcticament única per a cada entrada i varia de manera dràstica davant qualsevol modificació, per petita que sigui. Aquesta propietat permet detectar canvis no autoritzats i verificar la integritat de la informació.

Els principals motius per escollir aquest laboratori són els següents:

- **Distinció entre criptografia simètrica i asimètrica**: Es vol aprofundir en la diferència entre aquests dos tipus de criptografia i comprendre els seus avantatges i limitacions.

- **Limitacions de la criptografia simètrica**: Es destaca el problema de compartir la clau de manera segura, que és una debilitat important en aquest sistema.
- **Importància de la criptografia asimètrica**: Es mostra com aquest model permet una comunicació més segura mitjançant l'ús de claus públiques i privades, eliminant la necessitat de compartir una clau secreta prèviament.
- **Introducció al concepte de funció de hash**: Es considera rellevant introduir les funcions de hash per mostrar com es pot garantir la integritat de la informació. Aquestes funcions permeten detectar modificacions no autoritzades en les dades i són essencials en processos com les signatures digitals o l'emmagatzematge segur de contrasenyes.

Exfiltració de dades

Tot i que sembla que la paraula exfiltració sigui molt complexa d'entendre, si es recorre a un diccionari per entendre el seu significat no ho és tant. Bàsicament, vol dir quan ocorre la còpia o transferència no autoritzada de dades d'un servidor o un ordinador d'un individu cap al d'un altre extern. L'objectiu principal de l'acció és robar informació confidencial que poden provocar problemes greus a l'individu o a l'organització de la qual s'extreu [13]. Els motius principals per haver escollit el laboratori d'exfiltració de dades són els següents:

- **Comprensió de riscos crítics**: L'exfiltració de dades és un dels objectius més comuns dels atacants i un problema greu per a les organitzacions.
- **Simulació d'atacs reals**: Els estudiants aprenen a entendre com es poden extreure dades sensibles d'un sistema, cosa que ajuda a identificar punts febles en les defenses.
- **Conscienciació sobre la protecció de dades**: Aquesta pràctica fomenta la comprensió de la importància de polítiques de seguretat com el xifrat, la segmentació de xarxes i el monitoratge.

Les tècniques emprades en aquest laboratori inclouen mètodes com els atacs de phishing, l'enginyeria social o la generació de programari maliciós amb l'objectiu d'interrompre el funcionament habitual d'un dispositiu o d'obtenir informació confidencial. Aquestes tècniques constitueixen vies per obtenir dades d'un sistema, d'una entitat o d'un dispositiu, i poden introduir-se a través de diferents vectors, com ara dispositius USB, descàrregues de programari amb codi maliciós o l'explotació de vulnerabilitats en serveis accessibles des d'Internet.

Tot i això, un cop s'ha obtingut la informació, encara cal exfiltrar-la, és a dir, enviar-la cap a una ubicació sota control de l'atacant, com pot ser un servidor extern. Per l'enviament es vol enfocar amb l'ús d'alguns dels principals protocols de xarxa que existeixen en la comunicació entre xarxes: ICMP, DNS i HTTP. La majoria dels dispositius connectats a una xarxa fan ús d'aquests. L'elecció d'aquests protocols es deu al fet que són àmpliament utilitzats en la comunicació entre dispositius i, sovint, no són bloquejats pels sistemes de seguretat convencionals. Per tant, els atacants poden aprofitar-los per exfiltrar informació sense aixecar sospites [14] [15].

OSINT

L'OSINT que són les sigles en anglès per Open Source Intelligence (Inteligència de Fonts Obertes) i es refereix al conjunt de tècniques i eines que s'utilitzen per recopilar informació pública, analitzar dades i relacionar-les per convertir-les en coneixement útil. Els motius principals per escollir el laboratori de OSINT són els següents:

- **Importància en la reconeixença**: L'ús d'eines com Shodan o Censys ensenya als estudiants a recollir informació pública sobre sistemes i xarxes.
- **Exploració de vulnerabilitats**: L'ús de bases de dades com CVE (Common Vulnerabilities and Exposures) connecta la teoria amb vulnerabilitats reals, ajudant a entendre com identificar i mitigar riscos.
- **Ús de les APIs per a l'automatització de consultes**: Tant Shodan com Censys ofereixen una interfície de programació d'aplicacions (API) que permet automatitzar consultes, extreure dades i analitzar-les de manera estructurada.

A l'hora de fer aquest laboratori, es vol centrar en la utilització de dues eines de recopilació de dades en relació amb l'IoT, l'Internet de les Coses. Tant Shodan com Censys són dos motors de cerca que permeten descobrir dispositius connectats a Internet i proporcionar informació detallada sobre ells [16] [17].

El seu ús és legal, sempre que s'utilitzi de manera ètica com per la investigació de seguretat, la supervisió de la infraestructura en línia i la protecció contra amenaces cibernètiques. No obstant això, la seva capacitat per buscar dispositius i sistemes connectats a Internet planteja preocupacions sobre la privacitat i la seguretat.

A més, incloent-hi també CVE, es podrà llistar totes les vulnerabilitats conegudes associades als dispositius o serveis identificats.

3 Objectius i organització

L'objectiu principal d'aquest projecte és el desenvolupament i la implementació d'un conjunt de laboratoris de ciberseguretat que permetin als estudiants aprendre i practicar tècniques de seguretat mitjançant simulacions realistes en un entorn controlat. A continuació es presenten els objectius específics i el públic potencial del projecte.

3.1 Objectius principals

- **Desenvolupar un conjunt de laboratoris de ciberseguretat** que abordin diferents àrees de seguretat de xarxes i sistemes, com ara criptografia, exfiltració de dades, etc.
- **Proporcionar una metodologia d'ensenyament progressiva i interactiva** per a l'aprenentatge pràctic dels conceptes de ciberseguretat mitjançant l'execució de simulacions i pràctiques realistes.
- **Elaborar materials de suport** que incloguin guies pas a pas, instruccions i explicacions per a facilitar la comprensió de cada laboratori per part dels estudiants.
- **Promoure un enfocament pràctic de l'aprenentatge** centrat en el desenvolupament d'habilitats tècniques en l'àmbit de la ciberseguretat a través d'un model de l·learning by doing".
- **Avaluar l'impacte educatiu dels laboratoris** creats mitjançant el feedback dels estudiants i usuaris i la seva aplicació en l'ensenyament i formació en ciberseguretat.

3.2 Públic potencial

- **Estudiants de l'assignatura de Seguretat i Administració de Sistemes** que volen millorar les seves habilitats en la protecció de sistemes i xarxes a través de pràctiques aplicades.
- **Professionals i professors** que necessitin materials i recursos per a la formació de futurs experts en ciberseguretat.
- **Entitats educatives i centres de formació** que vulguin integrar laboratoris de ciberseguretat en els seus programes d'estudi o formació especialitzada.

3.3 Organització i metodologia de treball

Per desenvolupar aquest TFG, s'ha estudiat quines metodologies estan més alineades al propòsit del TFG. Per aquesta raó, s'ha decidit utilitzar una metodologia iterativa-incremental amb un enfocament àgil per assegurar que el projecte es pugui adaptar a canvis i necessitats emergents. El procés es divideix en etapes clares, però amb la flexibilitat d'ajustar el nombre i la complexitat dels laboratoris en funció del progrés i dels resultats obtinguts.

1. **Fase de planificació**

   En aquesta fase es realitza una recerca inicial per tal de fer una selecció dels laboratoris que cobreixin habilitats pràctiques rellevants per l'assignatura d'Administració i Seguretat de Sistemes. Una vegada seleccionats, es decideix quins són els més adients d'acord amb l'ús de tècniques actuals de ciberseguretat i la relació amb casos d'ús reals.

   (a) Definir els temes dels laboratoris inicials

   (b) Crear un pla inicial amb el nombre estimat de laboratoris

2. **Fase de Recerca i Anàlisi**

   Aquesta fase consisteix en la recerca de tecnologies i metodologies adequades per al desenvolupament dels laboratoris. Es consulta fonts acadèmiques i documentació oficial d'eines de seguretat per garantir la rellevància.

   (a) Recerca d'informació d'eines i tècniques per cada laboratori

   (b) Analitzar diferents enfocaments per dissenyar els laboratoris

3. **Fase de Disseny i Desenvolupament de Laboratoris**

   Per desenvolupar els diferents laboratoris, s'utilitza eines com GitHub per al control de versions com per mirar documentació i exemples; StackOverflow per consultar solucions a problemes tècnics que surtin durant el desenvolupament d'aquestes, etc.

   (a) Dissenyar i implementar els laboratoris

   (b) Cada laboratori es desenvolupa de manera independent

   (c) Revisions setmanals per ajustar i ampliar el seu contingut

4. **Fase de Validació i Avaluació**

   Aquesta fase té com a objectiu validar la rellevància i adequació dels laboratoris mitjançant proves amb estudiants de l'assignatura d'Administració i Seguretat de Sistemes. Durant aquesta fase, es fa un anàlisi sobre si els estudiants entenen els conceptes fonamentals de cada laboratori i al mateix temps, es recull informació per a futures millores.

   (a) Provar els laboratoris amb els estudiants de l'assignatura de Seguretat i Administració de Sistemes

   (b) Recollir el feedback dels estudiants

   (c) Realitzar informe d'avaluació dels resultats i possibles millores.

5. **Fase de Documentació i Lliurament**

   (a) Documentar el procés de desenvolupament de cada laboratori

   (b) Afegir el feedback dels laboratoris fets per l'estudiant.

4 Desenvolupament

En aquest apartat, s'explica el desenvolupament i les possibles solucions del diferents laboratoris que prèviament s'han introduït. En els documents dels laboratoris inclosos als annexos no s'hi ha incorporat el nom de l'estudiant autor del treball. Únicament hi consten els noms del tutor, Pere Tuset-Peiró, i del professor responsable de l'assignatura, Michael Pilgermann. Aquesta decisió s'ha pres perquè, de manera paral·lela a la realització del Treball de Final de Grau, també es cursa l'assignatura en què s'implementen aquests laboratoris. Per coherència acadèmica, s'ha considerat més apropiat no incloure el nom de l'estudiant en el material docent.

4.1 Laboratori 1: Criptografia clàssica i criptoanàlisi

4.1.1 Resum

Aquest primer laboratori proporciona una introducció a la criptografia, demostrant com les tècniques clàssiques, originalment desenvolupades a l'antiga Grècia, es poden utilitzar per aconseguir la confidencialitat en transmetre informació per canals insegurs. Es començara explicant i posant en pràctica diversos mètodes criptogràfics clàssics: els xifrats Cèsar, Vigenère i Rail Fence, tots basats en senzills mecanismes de substitució i transposició.

Tot i això, també es mostra com aquestes tècniques clàssiques són actualment vulnerables a diferents tipus d'atacs a causa de l'augment de la potència de càlcul, cosa que redueix significativament el temps necessari per recuperar el missatge original sense la clau de xifrat. Per mostrar-ho, s'explora dues tècniques habituals de criptoanàlisi: la força bruta i l'anàlisi de freqüències.

Per posar aquests coneixements a prova, es realitzen un conjunt d'activitats amb Python on s'usen els diferents algoritmes clàssics descrits. A més, per demostrar les seves vulnerabilitats, s'ha d'implementar les tècniques de criptoanàlisi per tal de trencar-los.

Al realitzar i completar aquestes activitats, s'hauria de comprendre perquè les tècniques de criptografia clàssica no són adequades per a les aplicacions modernes. A més, és important tenir en compte que la criptografia clàssica no proporciona cap protecció quant a la integritat dels missatges; una entitat maliciosa podria modificar un missatge xifrat sense detectar-lo ni l'emissor ni el receptor. Aquesta limitació s'estudia més a fons al laboratori de criptografia

moderna, on es presenten mètodes criptogràfics contemporanis que ofereixen garanties de seguretat més amplies.

En respecte amb els diferents tipus de hackers segons les característiques del laboratori, tenint en compte els principis que es treballen, aquests es vinculen amb els aspectes defensius i la protecció de sistemes, habitualment associats amb el rol del barret blau (Blue Hat) [18].

4.1.2 Explicació teòrica

Per dur a terme aquest laboratori, primer es va analitzar quins algoritmes de criptografia clàssica eren més adequats. D'entre les diferents opcions disponibles, s'han seleccionat els xifrats César, Vigènere i Rail Fence. Un cop seleccionats, aquests algoritmes s'expliquen en la part teòrica del laboratori amb l'objectiu que els estudiants entenguin el seu funcionament i la seva aplicació pràctica.

La principal característica dels tres algoritmes és la seva simplicitat i fàcil implementació. Tots tres algoritmes són de fàcil comprensió i poden ser implementats amb pocs recursos. Permeten introduir els conceptes de substitució i transposició d'una manera clara i intuïtiva. Tot i que tots fan ús d'una clau per xifrar i desxifrar, el mecanisme intern de cada un és diferent:

- **El xifrat Cèsar** utilitza un xifrat per desplaçament on cada lletra es substitueix per una altra segons un desplaçament fix en l'alfabet, però mantenint l'ordre del text original [19].

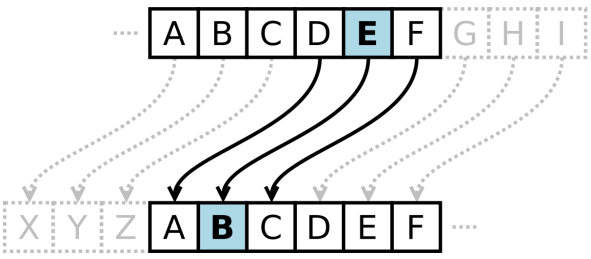

Figura 4.1: Captura del xifrat César

Font: https://en.wikipedia.org/wiki/Caesar_cipher

- **El xifrat Vigènere** fa ús d'un xifrat per substitució poli alfabètica, és a dir, utilitza més d'un alfabet. Cada lletra del missatge es xifra amb una lletra diferent de l'alfabet, determinada per una paraula clau que es repeteix al llarg del text.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figura 4.2: Quadrat del xifrat Vigenère

Font: https://www.researchgate.net/figure/The-Vigenere-table-10-11_fig1_341734578

Per exemplificar el funcionament del xifrat Vigènere, es pren la paraula **"SEGURETAT"** i la clau **"CLAU"**. La clau es repeteix tantes vegades com calgui per cobrir la longitud de la paraula.

- Paraula a encriptar: S E G U R E T A T
- Clau d'encriptació: C L A U C L A U C

A continuació, s'aplica una substitució on per cada lletra de paraula a encriptar es busca la seva lletra en l'alfabet per columnes. La intersecció de les dues lletres a partir dels alfabets dona la lletra encriptada. Llavors, queda així l'encriptació:

- ’S’ (fila) + ’C’ (columna) → **U**
- ’E’ (fila) + ’L’ (columna) → **P**
- ’G’ (fila) + ’A’ (columna) → **G**
- ’U’ (fila) + ’U’ (columna) → **O**
- ’R’ (fila) + ’C’ (columna) → **T**
- ’E’ (fila) + ’L’ (columna) → **P**
- ’T’ (fila) + ’A’ (columan) → **T**
- ’A’ (fila) + ’U’ (columna) → **U**
- ’T’ (fila) + ’C’ (columna) → **V**

El resultat final del xifrat és: **”UPGOTPTUV”**.

- **El xifrat Rail Fence** utilitza un xifrat per transposició on les lletres del text es reordenen seguint un patró en forma de zig-zag determinat per la clau, sense modificar-ne el valor dins de l’alfabet [20].

  Per exemple, si el missatge fos **”HELLO WORLD”** amb 3 raïls, així quedaria el missatge encriptat:

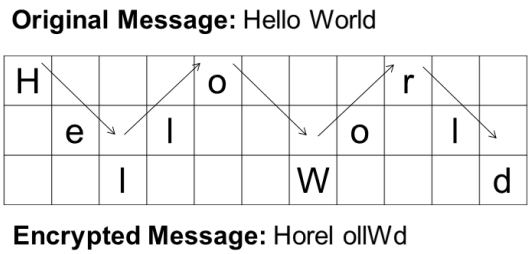

Figura 4.3: Encriptació de HELLO WORLD utilitzant el xifrat Rail Fence

Font: https://www.researchgate.net/figure/Encrypting-using-Rail-Fence-Cipher5_fig5_333480277

La introducció d’aquest conjunt d’algoritmes permet presentar nocions bàsiques de la criptografia com l’ús de claus secretes (com el nombre de desplaçaments) i el concepte de confidencialitat. Aquests mètodes faciliten la comprensió dels principis fonamentals del procés de xifrat i desxifrat, i creen una base sòlida per a l’estudi de sistemes més complexos.

A més, aquests algoritmes són útils per exemplificar diferents tècniques d'atac que es poden aplicar als sistemes de xifrat clàssic:

- Atacs de força bruta consisteixen a provar totes les claus possibles fins a trobar la correcta, explotant tot el conjunt de claus, ja que sol ser reduït en xifrats com el Cèsar.
- Tècniques de criptoanàlisi, com l'anàlisi de freqüència, on utilitza les lletres, els grups de lletres o les paraules més comunes per deduir el missatge original sense conèixer la clau [21].

Un cop establerts els algoritmes a fer servir, es defineixen els diferents apartats per garantir que els estudiants adquireixin un coneixement sòlid sobre els sistemes de criptografia clàssica i la criptoanàlisi. Així, el laboratori es divideix en les següents activitats que s'han de realitzar en Python:

1. Desenvolupament del algoritme Cèsar
2. Desenvolupament del algoritme Vigènere
3. Desenvolupament del algoritme Rail Fence
4. Implementació de l'atac de força bruta contra el xifrat Cèsar
5. Implementació de l'atac d'anàlisis de freqüència contra el xifrat Cèsar
6. Exploració d'un altre mètode de criptoanàlisi avançat per trencar el xifrat Cèsar més efectiu

4.1.3 Desenvolupament del algoritme Cèsar

Aquesta primera activitat requereix implementar dos programes independents que apliquin el xifrat Cèsar: un per realitzar l'encriptació i un altre per efectuar el desencriptat. L'objectiu és que l'estudiant entengui el funcionament bàsic d'aquest algorisme de substitució i pugui visualitzar el procés mitjançant la seva aplicació sobre un text curt.

Els programes han de rebre tres paràmetres per terminal: el text que es vol xifrar o desxifrar, la clau numèrica utilitzada per al desplaçament i el nom de l'arxiu de sortida. A més, es limita el xifrat als caràcters de l'alfabet llatí per tal de repetir el comportament del xifrat original. Els caràcters que no pertanyin a aquest conjunt no es modifiquen.

Tant per al xifrat com per al desxifrat, es fa ús d'un "array"amb les lletres de l'alfabet llatí. En el procés d'encriptació, cada lletra del missatge se substitueix per una altra ubicada un nombre determinat de posicions més endavant dins de l'array, segons el valor de la clau.

```python
def encrypt_caesar(message, moves):
    encrypt_message = ""
    for character in message.upper():
        if character in alphabet_mayus:
            # Find the position of the character in the alphabet
            position = (alphabet_mayus.index(character) + moves) % len(alphabet_mayus)
            encrypt_message += alphabet_mayus[position]

        elif character in alphabet_minus:
            position = (alphabet_minus.index(character) + moves) % len(alphabet_minus)
            encrypt_message += alphabet_minus[position]

        else:
            encrypt_message += character  # Maintain the character if it is not in the alphabet
    return encrypt_message
```

Figura 4.4: Captura de la funció d'encriptació de text amb l'algoritme César

En el cas de desencriptació, el desplaçament es fa en sentit contrari per recuperar el missatge original. Aquest plantejament permet entendre com opera el desplaçament mono alfabètic i visualitzar els efectes del xifrat sobre el text.

```
def decrypt_caesar(encrypted_message, moves):
    decrypt_message = ""
    for character in encrypted_message:
        if character in alphabet_mayus:
            # Calculate the new position by subtracting the number of moves
            position = (alphabet_mayus.index(character) - moves) % len(alphabet_mayus)
            decrypt_message += alphabet_mayus[position]
        elif character in alphabet_minus:
            position = (alphabet_minus.index(character) - moves) % len(alphabet_minus)
            decrypt_message += alphabet_minus[position]
        else:
            decrypt_message += character  # Maintain the character if it is not in the alphabet
    return decrypt_message
```

Figura 4.5: Captura de la funció de desencriptació de text amb l'algoritme César

És important tenir en compte que, en desencriptar un text, cal utilitzar la tècnica i clau correctes ja que, en cas contrari, el text resultant no es desxifra correctament.

4.1.4 Desenvolupament de l'algoritme Vigènere

Aquesta segona activitat no difereix gaire de l'anterior. L'única modificació que s'introdueix és el tipus de xifrat emprat, que en aquest cas és el xifrat Vigènere. Es continuen utilitzant els tres paràmetres d'entrada, però canvia el tipus d'alfabet, ja que ara es fa servir taula ASCII.

Com s'ha indicat anteriorment, la clau d'aquest xifrat, a diferència del xifrat Cèsar i del Rail Fence, sol ser una paraula o una cadena alfanumèrica. Per realitzar el xifrat, una estratègia habitual consisteix a repetir la clau tantes vegades com calgui fins que la seva longitud sigui igual o superior a la del text. Aquesta repetició permet aplicar una substitució específica per a cada caràcter. Tot seguit, cal retallar la clau perquè tingui exactament la mateixa longitud que el text.

```
def vigenere_encrypt(text, key):
    cipherText = ""

    while len(key) < len(text):
        key+=key

    key = key[:len(text)]
```

Figura 4.6: Captura 1 de la funció d'encriptació de text amb l'algoritme Vigènere

Per a cada caràcter del text, es calcula el desplaçament a aplicar basant-se en el valor ASCII del caràcter corresponent de la clau. Se'n resta 32 per restringir l'operació als caràcters imprimibles. A continuació, s'aplica el càlcul del caràcter xifrat mitjançant els passos següents:

1. Es calcula el desplaçament sobre l'espai de 95 caràcters disponibles.

2. Es fa una suma per assegurar que el valor resultant estigui dins del rang.

3. Es converteix de nou a caràcter, ja que estava en format ASCII.

```
for i, char in enumerate(text):
    shift = ord(key[i]) - 32
    encrypted_char = chr(((ord(char) - 32 + shift) % 95) + 32)
    cipherText += encrypted_char
return cipherText
```

Figura 4.7: Captura 2 de la funció d'encriptació de text amb l'algoritme Vigènere

Finalment, el caràcter s'inclou en el text final.

Pel que fa a la desencriptació, el procés és molt similar al del xifrat Cèsar. L'única diferència significativa és que, en lloc de sumar el desplaçament, es resta per recuperar el caràcter original.

```
def vigenere_decrypt(text, key):
    plainText = ""

    while len(key) < len(text):
        key += key

    key = key[:len(text)]

    for i, char in enumerate(text):
        shift = ord(key[i]) - 32
        decrypted_char = chr(((ord(char) - 32 - shift) % 95) + 32)
        plainText += decrypted_char

    return plainText
```

Figura 4.8: Captura de la funció de desencriptació de text amb l'algoritme Vigènere

4.1.5 Desenvolupament del algoritme Rail Fence

Aquesta tercera activitat té un guió bastant similar a les dues darreres. L'únic que canvia és el tipus de xifrat que s'utilitza, en aquest cas el xifrat Rail Fence. Es manté igual que les anteriors els tres paràmetres d'entrada i, però canvia igual que l'altre el tipus d'alfabet que s'usa, la taula ASCII.

L'algoritme d'encriptació, en comparació amb el de desencriptació, és més simple. Per aquest, s'ha de crear tantes arrays buides depenent del número introduït per la terminal. A continuació, s'ha de passar per tots els caràcters i anar-los introduint d'un en un en les diferents "arrays". Quan s'arribi fins al número màxim introduït, llavors s'anirà omplint les arrays en direcció contrària fins a arribar a 0, on es tornarà a repetir el mateix patró fins a passar per tots els caràcters. Una vegada acabat, s'uniran tots els arrays per crear un string amb tots ells.

Pel que fa a la funció desencriptació, primer es crea una matriu buida amb tantes files com raïls indicats i tantes columnes com caràcters té el missatge xifrat. A continuació, es recorre aquesta matriu simulant el recorregut en zig-zag que hauria seguit el text original durant el xifrat, marcant amb un símbol les posicions on s'ubicaran els caràcters.

```
def decrypt_rail_fence(encrypted_message, rails):
    if rails < 2:
        return encrypted_message  # No decryption needed for 1 rail
    # Determine the zigzag pattern
    railArray = []
    for i in range(rails):
        railArray.append([''] * len(encrypted_message))

    row = 0
    step = 1

    # Mark positions where characters will be placed
    for col in range(len(encrypted_message)):
        railArray[row][col] = '*'
        if row == 0:
            step = 1
        elif row == rails - 1:
            step = -1
        row += step
```

Figura 4.9: Captura 1 de la funció de desencriptació de text amb l'algoritme Rail Fence

Un cop establert el patró, s'omplen les posicions marcades amb els caràcters del missatge xifrat, de manera que es reprodueix la disposició original. Finalment, es torna a recórrer la matriu en zig-zag per extreure els caràcters en l'ordre correcte i reconstruir el text original desxifrat.

```
# Fill the marked positions with the actual characters
index = 0
for r in range(rails):
    for c in range(len(encrypted_message)):
        if railArray[r][c] == '*' and index < len(encrypted_message):
            railArray[r][c] = encrypted_message[index]
            index += 1

# Read the characters in zigzag order
row = 0
step = 1
decrypt_message = []
for col in range(len(encrypted_message)):
    decrypt_message.append(railArray[row][col])
    if row == 0:
        step = 1
    elif row == rails - 1:
        step = -1
    row += step

return ''.join(decrypt_message)
```

Figura 4.10: Captura 2 de la funció de desencriptació de text amb l'algoritme Rail Fence

4.1.6 Implementació de l'atac de força bruta contra el xifrat Cèsar

En el quart exercici d'aquest laboratori es demana a l'estudiant que desenvolupi el codi necessari per dur a terme un atac de força bruta amb el xifrat Cèsar amb l'objectiu de desxifrar un text emmagatzemat en un fitxer situat a la carpeta del projecte.

Per afegir un nivell de dificultat addicional, se li requereix fer ús de la llibreria NLTK de Python *(Natural Language Toolkit)*, un conjunt d'eines i programes de codi obert que permeten dur a terme múltiples tasques relacionades amb el processament del llenguatge natural [22].

Amb aquesta llibreria, l'estudiant ha de descarregar un diccionari de paraules en anglès ja que el text encriptat està en aquest idioma. Un cop completada la descàrrega, ha de dur a terme l'atac de força bruta, que consisteix a provar totes les possibles combinacions per trobar la clau correcta. Per cada iteració, ha d'analitzar les paraules obtingudes i comparar-les amb les paraules del diccionari descarregat prèviament.

Per entendre el que s'ha de fer de manera visual, s'adjunta aquest diagrama:

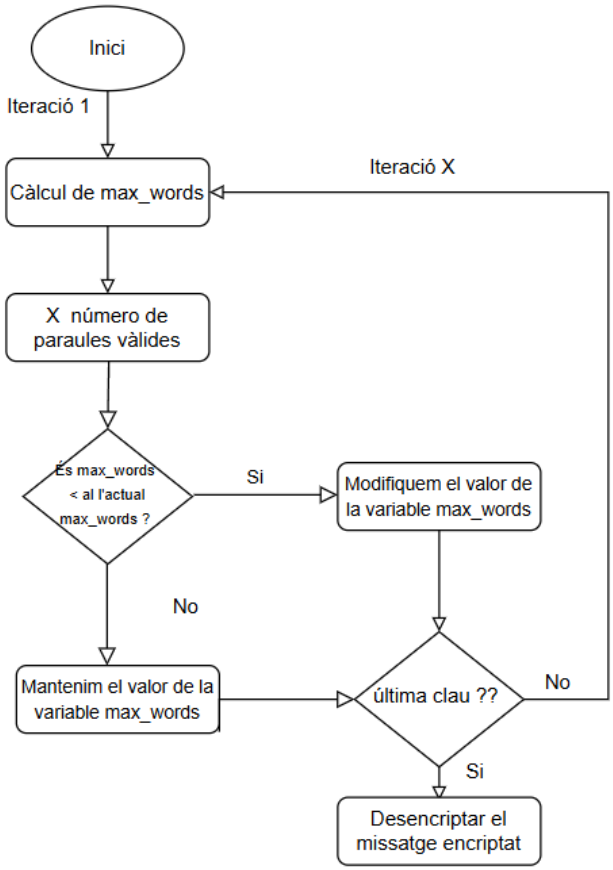

Figura 4.11: Diagrama de l'atac de força bruta de l'activitat 4

Després de provar totes les possibilitats, el programa ha de mostrar quina iteració presenta el major nombre de coincidències amb el diccionari, i finalment, exhibir el text desxifrat.

És important destacar que, en el cas del xifrat Cèsar, el nombre de claus possibles és limitat, ja que es basa en l'alfabet llatí, que només conté 26 lletres. Per tant, aquest serà el nombre màxim d'iteracions.

```python
# Variables to store the best decryption
best_movement = None
max_valid_words = 0

# Bruteforce attack to decrypt without knowing the key
print("\nTesting all the possible keys to do a brute-force attack:")
for move in range(1, len(alphabet_mayus)):
    test_message = decrypt_message(encrypted_message, move)
    valid_words = count_words_valid(test_message)

    print(f"Movement {move}: {test_message} (Number of word valids: {valid_words})")

    if valid_words > max_valid_words:
        max_valid_words = valid_words
        best_movement = move
```

Figura 4.12: Captura de codi funció en relació amb l'atac de força bruta

Tot i que aquest exercici no presenta una corba d'aprenentatge elevada, té com a objectiu demostrar un aspecte clau. En els atacs de força bruta, on s'utilitzen eines que cobreixen moltes més possibilitats que en exercicis anteriors, es fa indispensable disposar d'un mecanisme de verificació. Aquest mecanisme ha de permetre determinar, en cada iteració, quina de les opcions s'apropa més a la solució.

4.1.7 Implementació de l'atac d'anàlisis de freqüència contra el xifrat Cèsar

El quart apartat del laboratori consisteix a desenvolupar un algoritme de criptoanàlisi. La criptoanàlisi és la disciplina que s'encarrega d'estudiar i analitzar sistemes criptogràfics amb l'objectiu de trencar-los o trobar vulnerabilitats sense conèixer prèviament la clau.

L'exercici anterior, basat en l'atac de força bruta, era un exemple de criptoanàlisi, tot i existeix un debat sobre si es pot considerar una tècnica de criptoanàlisi real. És cert que és una forma pràctica d'intentar desxifrar missatges, però el problema és que no intenta entendre ni explotar febleses del sistema. A més, tampoc requereix un coneixement matemàtic ni tècniques d'anàlisi, aspectes que sí que compleixen les veritables tècniques de criptoanàlisi.

Aquest exercici se centra en l'anàlisi de freqüència d'un text xifrat amb el mètode Cèsar. L'anàlisi de freqüència es basa en l'estudi de la freqüència d'aparició de les lletres o combinacions en un text xifrat. Generalment, funciona bastant bé contra xifrats senzills com el Cèsar.

L'activitat consisteix a comparar dos textos, en aquest cas dos llibres: un encriptat *(Finis Mundi)* i un de referència sense xifrar *(El Quixot)*. A partir d'aquesta comparació, es pretén desxifrar el text encriptat. La selecció d'aquests dos llibres es deu al fet que són textos extensos i reflecteixen característiques generals de la llengua castellana.

Per iniciar l'activitat, primer, l'estudiant ha de crear una funció que calculi el percentatge d'aparició de cada lletra en ambdós textos i mostrar els resultats mitjançant un gràfic de barres.

```python
def calculate_frequencies(text):
    text = text.lower()

    # Remove all characters that are not letters
    letters_only = ""
    for char in text:
        if char in alphabet_minus or char in alphabet_mayus:
            letters_only += char

    total_letters = len(letters_only)
    count = Counter(letters_only)
    frequencies = {}

    # Calculate the frequency of each letter in the text
    for letter in string.ascii_lowercase:
        if total_letters > 0:
            frequencies[letter] = (count[letter] / total_letters) * 100
        else:
            frequencies[letter] = 0
    return frequencies
```

Figura 4.13: Captura de codi de la funció de càlcul de freqüències

A continuació, es comparen i s'ordenen les lletres en funció de la seva freqüència en cada text. Un cop ordenades, es realitza la substitució de manera sistemàtica: la lletra més freqüent del text xifrat es reemplaça per la més freqüent del text de referència, i així successivament fins a cobrir totes les lletres.

En executar el programa, es comprova que aquest tipus de criptoanàlisi no és el més precís. Tot i que algunes lletres el text es descodifiquen correctament i permeten llegir certes paraules, d'altres no ho fan. El motiu és que, encara que ambdós textos compleixen característiques generals del castellà, l'ús de determinades lletres pot variar entre ells.

4.1.8 Exploració d'un altre mètode de criptoanàlisi avançat per trencar el xifrat Cèsar més efectiu

Aquesta darrera activitat és una prolongació de l'anterior. Pel fet que no es pot acabar de desxifrar completament el text encriptat, es proposa aplicar tècniques complementàries. Una primera aproximació que els estudiants poden utilitzar consisteix a calcular la mitjana dels desplaçaments observats de cada lletra en comparar les freqüències de les lletres en els dos llistats.

Primer, cal comptar el total de cops que cada lletra apareix en cadascun dels textos i emmagatzemar aquest comptatge en un array per lletra.

```python
# Function that counts the times each letter appears in a text
def analyze_counts(text):
    # Remove all characters that are not letters
    letters_only = ""
    for char in text:
        if char in alphabet_minus:
            letters_only += char

    return Counter(letters_only)
```

Figura 4.14: Captura de codi de la funció de càlcul d'aparició de lletres

I a continuació, passar els dos arrays amb el comptador d'aparició de les lletres.

```python
# Function to calculate the average offset between the most common words
def calculate_average_offset(book_counts, cypher_counts):
    offsets = []

    common_cypher_words  = cypher_counts.most_common()
    common_book_words  = book_counts.most_common()

    # Compare the frequencies of the most common words
    for i in range(min(len(common_cypher_words), len(common_book_words))):
        cypher_word = common_cypher_words[i][0]
        book_word = common_book_words[i][0]

        if cypher_word in alphabet_mayus and book_word in alphabet_mayus:
            alphabet = alphabet_mayus
        else:
            alphabet = alphabet_minus

        cypher_index = alphabet.index(cypher_word)
        book_index = alphabet.index(book_word)

        print(f"Offset {cypher_word} -> {book_word}: {cypher_index, book_index} ,{(cypher_index - book_index) % len(alphabet_mayus)}")
        offset = (cypher_index - book_index) % len(alphabet_mayus)
        offsets.append(offset)

    # Calculate the average offset
    if offsets:
        return round(sum(offsets) / len(offsets))
    return 0
```

Figura 4.15: Captura de codi de la funció de càlcul de la mitjana de desplaçament

Tot i semblar una bona opció, continua sent imprecisa, ja que no té en compte patrons lingüístics més complexos.

Com a alternativa, és proposa analitzar exclusivament les paraules d'una sola lletra, ja que són molt freqüents en el castellà i poden proporcionar un desplaçament mitjà més fiable.

```
# Function to extract the one letter words from a text
def one_letter_words_func(text):
    words = text.lower().split()
    one_letter_words = []
    for word in words:
        if len(word) == 1 and word in string.ascii_lowercase:
            one_letter_words.append(word)

    count = Counter(one_letter_words)
    return count
```

Figura 4.16: Captura de codi de la funció per extreure les paraules d'una sola lletra

L'ús d'aquestes diferents tècniques no té com a únic objectiu desxifrar el text, sinó demostrar que l'eficàcia de la criptoanàlisi depèn tant de l'algoritme utilitzat com de la quantitat de dades disponibles.

4.2 Laboratori 2: Laboratori de criptografia moderna

4.2.1 Resum

Aquest laboratori explora les tècniques modernes de criptografia, les quals es basen en els conceptes de la criptografia simètrica clàssica, però presenten una major robustesa davant dels atacs de força bruta i criptoanàlisi. D'entre els diferents algorismes existents, com DES (*Data Encryption Standard*) o 3DES (*Triple Data Encryption Standard*), el laboratori se centra en l'ús de l'AES (Advanced Encryption Standard), considerat l'estàndard actual en xifrat simètric. L'algorisme AES proporciona una base sòlida per garantir la confidencialitat tant en la transmissió com en l'emmagatzematge de dades.

A més, s'introdueix el concepte de funció hash, una eina essencial per verificar la integritat dels fitxers o missatges. Aquestes funcions, com MD5 (*Message-Digest Algorithm 5*) o SHA-256 (*Secure Hash Algorithm 256-bit*), generen una sortida de mida fixa a partir d'una entrada determinada. Qualsevol modificació, per petita que sigui, altera completament el valor del hash. Aquesta propietat permet detectar canvis no autoritzats en la informació.

També s'aprofundeix en la criptografia asimètrica, que es diferencia de la simètrica per l'ús d'un parell de claus: una pública i una privada. Les dades xifrades amb una clau només es poden desxifrar amb l'altra clau del parell. Aquesta característica permet establir canals de comunicació segurs sense necessitat de compartir claus prèviament. A més, fa possible la implementació de firmes digitals, que permeten verificar l'autenticitat i la integritat d'un missatge. En aquest laboratori s'utilitza l'algorisme RSA (Rivest-Shamir-Adleman) com a representant d'aquest tipus de criptografia.

Per posar en pràctica aquests coneixements, es desenvolupen diferents activitats amb Python. S'hi implementen funcionalitats de xifrat i desxifrat utilitzant AES i RSA, així com la generació i comparació de valors hash. Aquestes pràctiques permeten comprovar l'efectivitat dels mètodes criptogràfics moderns i entendre millor les seves aplicacions. Igual que en el laboratori de criptografia clàssica, aquest també es fonamenta en els principis de seguretat de la informació. Els continguts es vinculen amb el rol del barret blau (Blue Hat), ja que s'orienten a la protecció i defensa dels sistemes mitjançant mecanismes criptogràfics avançats [18].

4.2.2 Explicació teòrica

Per realitzar aquest laboratori, es va fer una anàlisi inicial per abordar els diferents aspectes de la criptografia moderna. Aquest tipus de criptografia millora molts aspectes en què la criptografia clàssica tenia mancances i debilitats, com ara l'autenticació i la integritat de les dades.

Entre els aspectes més importants, es treballen els següents perquè els estudiants adquireixin uns coneixements sòlids sobre els elements essencials:

- **Diferència entre criptografia simètrica i asimètrica**

  La criptografia simètrica moderna és l'evolució de la criptografia clàssica, amb característiques que la fan més resistent als intents de desxifrat. Les claus són més llargues i segures gràcies a millores en els algoritmes. Un dels algoritmes més utilitzats actualment és l'AES (Advanced Encryption Standard), que permet crear claus de diferents mides segons el nivell de seguretat i el consum de recursos desitjat. A més, aquest serà fet servir en aquest laboratori.

  Tot i aquestes millores, la criptografia simètrica encara presenta limitacions en un dels tres principis bàsics de la seguretat de la informació: la integritat. No garanteix que un conjunt de dades no hagi estat modificat, ni permet comprovar la identitat de l'emissor, ja que no incorpora mecanismes d'autenticació.

  La criptografia asimètrica soluciona aquestes mancances introduint el concepte de clau pública. A diferència de la simètrica, on la mateixa clau serveix per xifrar i desxifrar, en l'asimètrica es divideixen les responsabilitats: la clau pública, que es pot compartir lliurement, s'utilitza per xifrar, mentre que la clau privada, que cal mantenir en secret, serveix per desxifrar.

  Amb aquest model, ja no és necessari un canal segur per compartir claus secretes, i només cal un parell de claus per usuari, fet que simplifica la gestió. A més, l'ús de signatures digitals amb la clau privada permet verificar la identitat del remitent.

  En aquest laboratori, per a qualsevol pràctica de criptografia asimètrica s'ha utilitzat l'algoritme RSA (*Rivest-Shamir-Adleman*). RSA és un dels més coneguts i utilitzats, ja que ofereix una combinació molt útil de xifrat i signatures digitals. A més, és relativament senzill d'implementar amb les biblioteques de Python, cosa que el fa realment pràctic per a entorns educatius i laboratoris.

- **Hashes i autenticació de missatges**

  Els hashes són cadenes alfanumèriques de longitud fixa que representen un conjunt de dades. Es generen mitjançant una funció hash, que és un algoritme matemàtic. Una característica essencial dels hashes és que, per a un conjunt concret de dades, sempre es generarà el mateix hash, i fins i tot un petit canvi en les dades donarà lloc a un hash completament diferent. Això fa que els hashes siguin molt útils per garantir la integritat de les dades, ja que permeten detectar qualsevol modificació [23].

  Tanmateix, els hashes no resolen el problema de l'autenticació: és a dir, no permeten verificar que les dades provenen d'una entitat en què confiem i que posseeix una clau secreta. Per això, l'autenticació de missatges és essencial, perquè ens permet assegurar-nos que les dades no només són íntegres, sinó també fiables pel que fa a la seva procedència.

  En referència els hashes, el hash a utilitzar serà el MD5 per calcular i verificar la integritat dels missatges. Tot i que MD5 ja no es considera segur per a aplicacions criptogràfiques crítiques, encara és bastant usats en entorns no tan crítics com la verificació de la integritat de fitxers. A més, la seva implementació és senzilla i ràpida, la qual cosa el fa molt adequat per un laboratori didàctic.

4.2.3 Activitat 1: Encriptació amb AES

L'objectiu d'aquesta activitat és desenvolupar un programa en Python que utilitzi l'algoritme AES per encriptar i desencriptar arxius. L'activitat pretén que l'estudiant entengui el funcionament bàsic de l'AES, així com les seves principals característiques.

El primer pas de l'activitat és comprovar si ja existeixen claus generades prèviament i guardades en un fitxer local. En cas contrari, el programa ha de generar aquestes claus i desar-les per poder reutilitzar-les més endavant en la fase de desencriptació. Aquest pas és important, ja que si es generés una clau nova cada vegada, no seria possible desencriptar correctament el contingut encriptat amb una clau anterior.

```
if os.path.exists("key_iv.txt"):
    with open("key_iv.txt", "rb") as file:
        key, iv = file.read(32), file.read(16)
else:
    key = os.urandom(32)
    iv = os.urandom(16)

    with open("key_iv.txt", "wb") as file:
        file.write(key + iv)
```

Figura 4.17: Captura del codi que llegeix o crea les claus de l'algoritme AES

Un cop disponibles les claus, ja que AES requereix una clau i un vector d'inicialització (IV) per realitzar el xifrat i desxifrat, es llegeix el contingut de l'arxiu i preparar-lo per al procés d'encriptació. Com que AES és un algoritme de bloc, és necessari aplicar un sistema de padding (emplenat) per assegurar que el contingut tingui una longitud múltiple de la mida del bloc. Amb això preparat, es pot encriptar el missatge i guardar-lo en un fitxer local.

```python
def encrypt_aes(text, key, iv):
    # Pad the text to be a multiple of the block size
    padder = padding.PKCS7(128).padder()  # 128 bits = 16 bytes
    padded_text = padder.update(text) + padder.finalize()

    # Create a cipher object using the key and IV
    cipherEncryptor = Cipher(algorithms.AES(key), modes.CBC(iv))
    encryptor = cipherEncryptor.encryptor()

    # Encrypt the padded text
    ct = encryptor.update(padded_text) + encryptor.finalize()
    print("Ciphertext:", ct)
    return ct
```

Figura 4.18: Captura del codi que encripta dades amb l'algoritme AES

La segona part de l'activitat consisteix a desencriptar el contingut. Per fer-ho, primer es recuperen les claus prèviament guardades. Si no es fes així i es generessin claus noves, la desencriptació fallaria, ja que les claus serien diferents. Un cop recuperades les claus, es llegeix el contingut encriptat i es procedeix al desxifrat. Finalment, es retira el padding afegit anteriorment per obtenir el text original, igual al que es tenia abans de l'encriptació.

```python
def decrypt_aes(text, key, iv):

    # Create a cipher object using the key and IV
    cipherDecryptor = Cipher(algorithms.AES(key), modes.CBC(iv))
    decryptor = cipherDecryptor.decryptor()

    # Decrypt the text
    decrypted_text  = decryptor.update(text) + decryptor.finalize()

    # Unpad the decrypted text
    unpadder = padding.PKCS7(128).unpadder()  # 128 bits = 16 bytes
    unpadded_text = unpadder.update(decrypted_text) + unpadder.finalize()

    decrypted_message = unpadded_text.decode('utf-8')

    print("Decrypted message:", decrypted_message)
    return decrypted_message
```

Figura 4.19: Captura del codi que desencripta dades amb l'algoritme AES

4.2.4 Activitat 2: Signatures amb MD5

Aquesta activitat consisteix a desenvolupar un programa en Python que utilitzi la funció de hash MD5 per calcular el hash d'un arxiu i verificar si els hashes de diferents arxius coincideixen. L'objectiu és que l'estudiant entengui com es generen els hashes i la seva importància per garantir la integritat dels arxius, és a dir, assegurar-se que no han estat modificats.

La primera funcionalitat desenvolupada és el càlcul del hash d'un arxiu. Aquesta operació és força senzilla, ja que la llibreria de Python *hashlib* proporciona funcions que permeten generar el hash de manera directa i eficient. Un cop calculat, el programa imprimeix per la terminal tant el valor en format hexadecimal com en format binari. Cal destacar que el format original del hash és una cadena de bytes (format binari), que no és directament llegible per a les persones. Per aquest motiu, sovint es converteix a format hexadecimal, que és molt més comprensible i pràctic per visualitzar o comparar.

La segona funcionalitat permet comparar els hashes de dos arxius. Per fer-ho, només cal passar al programa els dos fitxers que es volen comparar. El programa calcula els hashes de cadascun i després els compara. Si els hashes coincideixen, això indica que els arxius són idèntics; en canvi, si són diferents, significa que almenys un dels fitxers ha estat modificat.

```python
# Function to verify if two messages have the same MD5 hash
def verifyHashMessage(message1, message2):
    # Create a new md5 hash object from the first message
    message1 = hashlib.md5(message1.encode())
    print("Hash in hexadecimal format:")
    print(message1.hexdigest())

    # Create a new md5 hash object from the second message
    message2 = hashlib.md5(message2.encode())
    print("Hash in hexadecimal format:")
    print(message2.hexdigest())

    # Compare the two hashes
    if(message1.digest() == message2.digest()):
        print("The hash is the same. There was no change in the message.")
    else:
        print("The hash is different. The message was changed.")
```

Figura 4.20: Captura del codi verificar si el contingut de dos arxius és idèntic

4.2.5 Activitat 3: Encriptació amb RSA

Aquesta activitat té com a objectiu desenvolupar un programa en Python que faci ús de l'algoritme RSA per encriptar i desencriptar arxius. L'activitat pretén que l'estudiant entengui el funcionament bàsic de la criptografia asimètrica i la diferència amb la criptografia simètrica, especialment pel que fa a la gestió de claus i a les seves aplicacions pràctiques.

El primer pas del programa és la generació o lectura dels parells de claus RSA (clau pública i clau privada) per a dos participants: A i B. Aquestes claus es generen utilitzant la llibreria *cryptography*, especificant una mida de clau de 4096 bits, que determina tant el nivell de seguretat com l'eficiència del procés d'encriptació i desencriptació [24].

```python
def generate_keys():
    private_key = rsa.generate_private_key(
        public_exponent=65537,
        key_size=4096
    )
    public_key = private_key.public_key()
    return private_key, public_key
```

Figura 4.21: Captura del codi per generar les claus amb RSA

Si les claus ja s'han creat prèviament, es llegeixen d'un arxiu mitjançant un procés de deserialització, que converteix les claus des del seu format binari a un format útil per al programa. En cas contrari, s'han de generar de nou i després serialitzar-les, és a dir, convertir-les a format binari per poder-les guardar correctament en un fitxer.

Per realitzar l'encriptació, el programa divideix el missatge original en fragments petits, també anomenats chuncks, de 214 bytes, ja que l'algoritme RSA només pot gestionar una quantitat limitada de dades per operació (la mida depèn de la clau i del padding utilitzat). A continuació, cada fragment es xifra amb la clau pública del destinatari (B), fent servir l'esquema un esquema de padding OAEP (*Optimal Asymmetric Encryption Padding*) amb la funció de hash SHA-256 per assegurar la confidencialitat i resistència contra atacs. Els fragments xifrats s'emmagatzemen en una llista de blocs encriptats per ser posteriorment desats en un fitxer.

Un cop finalitzada l'encriptació, es pot efectuar l'operació inversa: la desencriptació. Tot i que el programa es desenvolupa dins el mateix script, en aquest punt se simula que el participant B rep el missatge. Per això, primer es llegeixen les claus des de l'arxiu corresponent. A continuació, B utilitza la seva clau privada per desencriptar cada bloc i reconstruir el missatge original. Finalment, el programa imprimeix el missatge desencriptat per comprovar que coincideix amb l'original i també el desa en un fitxer.

```python
def decrypt_message(encrypted_blocks, private_key_b):
    decrypted_data = b""
    for encrypted_part in encrypted_blocks:
        decrypted_part = private_key_b.decrypt(
            encrypted_part,
            padding.OAEP(
                mgf=padding.MGF1(algorithm=hashes.SHA256()),
                algorithm=hashes.SHA256(),
                label=None
            )
        )
        decrypted_data += decrypted_part

    print("Decrypted message received")
    return decrypted_data
```

Figura 4.22: Captura del codi per desencriptar amb RSA

4.2.6 Activitat 4: Signatures amb RSA

Per finalitzar amb les activitats de criptografia moderna, la darrera activitat amplia la funcionalitat de l'activitat anterior incorporant el concepte de signatura digital. L'objectiu és demostrar com, mitjançant l'ús de claus asimètriques, es pot garantir no només la confidencialitat, sinó també l'autenticitat i integritat del missatge. En aquest cas, es continua fent servir RSA tant per xifrar com per signar digitalment el contingut.

El primer pas del programa és la generació o lectura dels parells de claus RSA per als dos participants (A i B), tal com es va fer a l'activitat anterior. Un cop generades o desades, aquestes claus es fan servir per xifrar i signar el missatge.

Per tal de garantir la integritat i autenticitat, el participant A genera una signatura digital del missatge. Aquesta signatura es crea a partir del contingut del missatge utilitzant un algorisme de hash segur (SHA-256) combinat amb el padding PSS (Probabilistic Signature Scheme), que proporciona seguretat addicional mitjançant l'ús d'un salt aleatori. A partir del hash resultant, aquest és xifrat amb la clau privada d'A, i el resultat crea la signatura digital. A continuació, aquesta signatura es concatena amb el missatge original mitjançant un separador per tal de facilitar la separació en desencriptar. Tot seguit, es prepara el conjunt per ser xifrat.

```python
def generate_signature(private_key,message):

    signature = private_key.sign(
        message,
        padding.PSS(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            salt_length=padding.PSS.MAX_LENGTH
        ),
        hashes.SHA256()
    )
    return signature
```

Figura 4.23: Captura del codi per generar les signatures amb RSA

Igual que en l'activitat 3, el contingut (missatge + signatura) es divideix en chunks i es xifra amb la clau pública de B utilitzant OAEP i SHA-256. Els blocs encriptats s'emmagatzemen en un fitxer local.

Aleshores, canviant de rol, el participant B rep el missatge. Aquest es fa servir la seva clau privada per desencriptar cada bloc i reconstruir les dades originals. Tot seguit, cal separar el missatge de la signatura i es verifica que la signatura és vàlida. Per fer-ho, B calcula el hash del missatge i el compara amb el hash desencriptat utilitzant la clau pública d'A. Si coincideixen, això vol dir que el missatge no ha estat modificat i que prové realment de l'emissor esperat (A).

```python
def verify_signature(decrypted_data, client_public_key):
    message, signature = decrypted_data.split(b"||SIGNATURE||")

    try:
        client_public_key.verify(
            signature,
            message,
            padding.PSS(
                mgf=padding.MGF1(algorithm=hashes.SHA256()),
                salt_length=padding.PSS.MAX_LENGTH
            ),
            hashes.SHA256()
        )
        print("The firm is valid")
        return message.decode()
    except Exception as e:
        print("The firm is not valid:", e)
        return None
```

Figura 4.24: Captura del codi per verificar les signatures amb RSA

Aquest procés és essencial en entorns on és necessari assegurar-se que la informació no ha estat alterada durant la transmissió i que l'origen és legítim. Permet garantir tant la integritat com l'autenticitat, afegint un nivell de seguretat fonamental en comunicacions modernes.

4.3 Laboratori 3: Exfiltració de dades

4.3.1 Resum

Al llarg del laboratori s'analitzen diverses tècniques i protocols de xarxa habitualment utilitzats per dur a terme exfiltració de dades. D'una banda, s'estudien dos enfocaments concrets: el registre de pulsacions de teclat (keylogging) i la transferència encoberta de fitxers. D'altra banda, s'explora com es poden aprofitar protocols habituals de xarxa com ICMP, HTTP i DNS per exfiltrar dades. Aquests protocols, àmpliament utilitzats en entorns de producció, sovint tenen permís per travessar tallafocs i altres mecanismes de protecció. Això els converteix en canals potencials per a atacs encoberts.

El laboratori també incorpora l'estudi de mesures defensives per prevenir aquests atacs, com la formació en seguretat, el monitoratge de trànsit o el desplegament de tallafocs de nova generació.

Per posar en pràctica aquests coneixements, l'objectiu del laboratori es entendre tots aquests coneixements a partir de la proposició de diferents activitats en Python. Aquestes inclouen la creació d'una arquitectura client-servidor que simula l'exfiltració de dades mitjançant ICMP i HTTP, així com la codificació i descodificació de missatges i fitxers. A més, també inclou una activitat teòrica centrada en l'anàlisi de l'exfiltració mitjançant el protocol DNS, amb l'objectiu de descriure com es poden incrustar dades dins de les consultes o respostes DNS. Finalment, el laboratori conclou amb un conjunt de qüestions orientades a la reflexió i a la detecció de possibles mesures de defensa.

Aquest laboratori presenta característiques pròpies del barret vermell (Red Hat), ja que simula accions ofensives que podrien ser realitzades per un atacant. No obstant això, també incorpora la perspectiva del barret blau (Blue Hat), atès que permet analitzar les vulnerabilitats exposades i plantejar estratègies de detecció i mitigació. Aquesta doble mirada permet adquirir una visió més àmplia i crítica sobre els riscos associats a la seguretat dels sistemes [18].

4.3.2 Teoria del laboratori

En relació amb el laboratori d'exfiltració de dades, es va introduir el tema explicant en què consisteix i la seva importància en diversos àmbits, tant personals com laborals.

A continuació, es van descriure les tècniques més comunes per exfiltrar dades, com els atacs de phishing i el malware. L'atac de phishing consisteix que l'atacant es fa passar per un individu de confiança per enganyar la víctima i aconseguir que reveli informació confidencial, com noms d'usuari, contrasenyes o dades bancàries, per accedir a serveis amb elles [25]. D'altra banda, un malware és un programari dissenyat per interrompre el funcionament normal d'un dispositiu i dur a terme accions perjudicials per a l'usuari. En el cas de l'exfiltració de dades, aquest tipus de programari es pot utilitzar per robar informació.

A més d'explicar les tècniques d'atac, també es van presentar diverses mesures de prevenció per evitar aquestes amenaces. Algunes d'aquestes mesures inclouen la formació en matèria de seguretat als usuaris, perquè puguin identificar intents d'exfiltració de dades per part d'atacants i, l'ús de firewalls de nova generació, que proporcionen una seguretat avançada per analitzar i bloquejar connexions no autoritzades [26].

Un cop explicats els tipus d'atacs i les mesures de prevenció, es van introduir els protocols de xarxa principals que s'utilitzaren en aquest laboratori i els atacs relacionats amb ells.

Protocol ICMP i ICMP Tunneling

Primer, s'explica el protocol ICMP, que serveix per determinar si un dispositiu és accessible i mesurar el temps de resposta entre nodes de la xarxa, entre altres funcions. Després d'una breu explicació, s'introdueix la tècnica d'exfiltració de dades que utilitza aquest protocol anomenada ICMP Tunneling.

Aquest mètode consisteix a transferir dades de manera encoberta des d'un dispositiu compromès cap al servidor de l'atacant. És especialment efectiu perquè molts firewalls i sistemes de detecció d'intrusions permeten el tràfic ICMP per la seva utilitat en la funcionalitat de la xarxa. Com que ICMP es fa servir per enviar pings que no suposen cap mal als dispositius, dins d'aquests paquets es poden incrustar les dades a exfiltrar [27].

Figura 4.25: Captura del ICMP Tunneling

Font: https://www.vectra.ai/detections/icmp-tunnel

Protocol DNS i DNS Tunneling

En segon lloc, s'explica el protocol DNS, que permet als usuaris navegar per Internet utilitzant noms de domini en comptes d'adreces IP numèriques. Posteriorment, s'introdueix la tècnica d'exfiltració de dades relacionada amb el protocol DNS anomenada DNS Tunneling.

Aquest mètode és més complex, ja que no es tracta simplement de transferir dades d'un punt a un altre, sinó que requereix un procés previ. Primer, l'atacant ha de registrar un domini i configurar un servidor de noms que apunti cap al seu propi servidor. Després, la víctima és infectada amb un malware, que fa una petició DNS inicial per obtenir l'adreça IP del servidor de l'atacant [28].

Un cop establerta la connexió, el malware comença a exfiltrar dades fent-se passar per consultes a subdominis del domini principal, incrustant les dades en el text del subdomini.

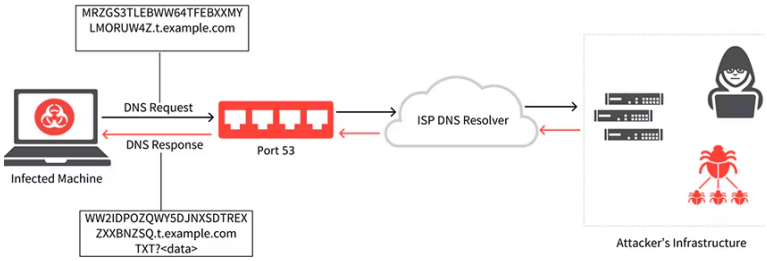

Figura 4.26: Captura del DNS Tunneling

Font: https://www.paloaltonetworks.lat/cyberpedia/what-is-dns-tunneling

Protocol HTTP i HTTP Tunneling

Finalment, es va explicar el protocol HTTP, que és la base de la navegació web i s'utilitza principalment per carregar pàgines webs mitjançant enllaços d'hipertext. Llavors, la tècnica d'exfiltració de dades que fa servir aquest protocol s'anomena HTTP Tunneling.

Aquest mètode es basa en un malware actiu en el dispositiu de la víctima, que utilitza peticions HTTP per incrustar les dades a exfiltrar. Aquestes peticions són enviades a un servidor controlat per l'atacant, on es desencripten i s'extreuen les dades [29] [30].

Aquest mètode és efectiu perquè tant els firewalls com els sistemes de detecció d'intrusions permeten el trànsit HTTP per la seva importància en la comunicació web. A més, si s'utilitza HTTPS, que incorpora xifrat TLS, es dificulta la detecció d'anomalies.

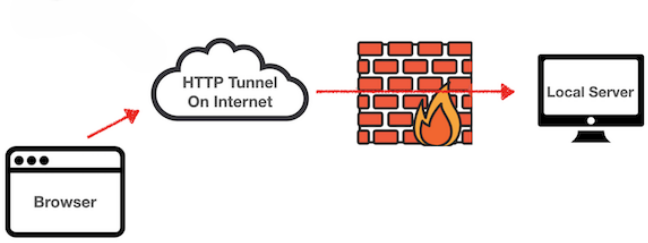

Figura 4.27: Captura del HTTP Tunneling

Font: https://medium.com/@embbnux/building-a-http-tunnel-with-websocket-and-node-js-98068b0225d3

Tipus de malware utilitzats

Després d'explicar els protocols i tècniques d'exfiltració previs, es van introduir dos tipus de malware que es poden utilitzar per robar dades:

- **Keylogger**: Un malware que registra totes les pulsacions de tecles d'un dispositiu de manera silenciosa. Al ser recopilades, aquestes dades són enviades a l'atacant mitjançant algun dels protocols explicats. També s'explica com implementar aquest malware a la pràctica [31].
- **Exfiltració de fitxers**: Procés mitjançant el qual es roba informació d'un sistema de manera no autoritzada. Per dur-lo a terme, un atacant segueix tres passos:
  1. Accedir al fitxer mitjançant accés remot, malware o explotació de vulnerabilitats.
  2. Preparar l'extracció, xifrant-lo, dividint-lo o amagant-lo per evitar ser detectat.
  3. Enviar les dades a través de diversos canals de comunicació sense ser detectat.

Objectiu de la pràctica i guia d'implementació

L'objectiu principal de la pràctica és exfiltrar dades en un entorn controlat i segur, sense risc per als estudiants. Les tècniques explicades es simplificaran perquè siguin més fàcils d'entendre i implementar.

Per tal de realitzar la pràctica correctament, els estudiants hauran de crear una comunicació client-servidor i utilitzar dues tècniques d'exfiltració amb dos dels protocols explicats:

- Realitzar un keylogger fent ús del protocol ICMP.
- Realitzar una exfiltració de fitxers fent ús del protocol HTTP.

A més, es demana l'ús de diferents llibreries per implementar cada protocol correctament. L'explicació es dividirà en dues parts:

- **Configuració de la banda del client**: Inclourà el format del paquet de cada protocol i com incrustar-hi les dades.
- **Configuració de la banda del servidor**: Especificarà com rebre i processar les dades exfiltrades.

A més, per cada banda, també s'anomenaran algunes de les funcions que han d'utilitzar per realitzar la implementació.

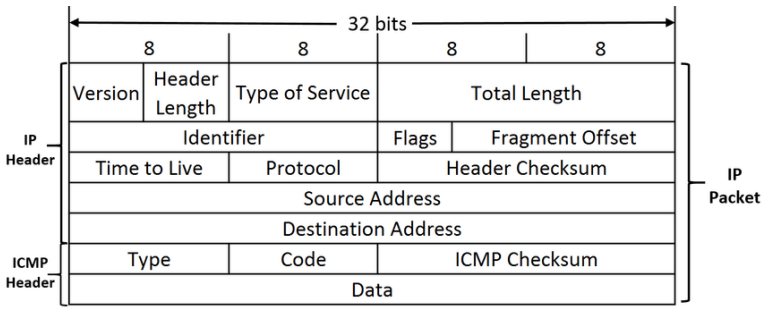

Figura 4.28: Format de la capçalera del protocol DNS

Font: https://www.researchgate.net/figure/CMP-packet-structure_fig5_316727741

També es detallen característiques addicionals per als dos tipus de malware. Per exemple, en el cas del keylogger, caldrà utilitzar els threads mitjançant una llibreria per aturar la lectura de les tecles i l'enviament de dades. En el cas de l'exfiltració de fitxers, es definirà el directori on s'han de trobar els fitxers i com accedir-hi. Finalment, les dades exfiltrades hauran d'estar xifrades i ofuscades per evitar que siguin llegibles. Un cop arribin al servidor, s'hauran de desencriptar i desar en fitxers.

Aquest laboratori no es limita a la implementació tècnica i la documentació del desenvolupament. A més de les pràctiques relacionades amb els protocols ICMP i HTTP, s'inclou una activitat d'anàlisi vinculada al protocol DNS. En aquesta activitat, s'ha de descriure com el plantejament una tècnica d'exfiltració de dades mitjançant DNS, detallant el funcionament del mètode, la manera d'incrustar la informació en les consultes o respostes DNS i els possibles reptes tècnics.

Finalment, l'activitat incorpora un conjunt de preguntes reflexives orientades a consolidar el coneixement adquirit. Algunes d'aquestes qüestions se centren en les mesures de protecció que poden implementar-se per prevenir exfiltracions de dades en entorns reals.

4.4 Laboratori 4: Laboratori d'OSINT

4.4.1 Resum

Aquest laboratori introdueix l'OSINT (Open Source Intelligence), un conjunt de tècniques orientades a la recopilació i anàlisi d'informació pública procedent de fonts obertes. Aquest tipus de reconeixement permet detectar serveis exposats, identificar vulnerabilitats conegudes i comprendre la superfície d'atac d'una infraestructura sense necessitat d'explotar cap sistema.

L'objectiu principal del laboratori és familiaritzar-se amb eines d'OSINT com Shodan i Censys, i entendre com es poden utilitzar per identificar dispositius i serveis connectats a Internet, així com vulnerabilitats conegudes associades. A més, també es busca desenvolupar la capacitat d'interpretar resultats i representar-los de manera visual mitjançant gràfics i mapes.

Per assolir aquests objectius, es proposen quatre activitats principals. En primer lloc, s'utilitza la interfície web de Shodan o Censys per realitzar cerques bàsiques i entendre el funcionament dels filtres, les consultes i les metadades. En segon lloc, es redacta un informe comparatiu entre ambdues plataformes, identificant les diferències en la manera com obtenen, processen i presenten la informació. La tercera activitat consisteix en desenvolupar un script en Python que consulta l'API d'una de les plataformes i genera un informe gràfic amb estadístiques com la distribució geogràfica, la presència de vulnerabilitats o les organitzacions més exposades. Finalment, s'aplica la funcionalitat facets de l'API de Shodan per generar un resum estadístic eficient dels serveis, sistemes operatius i tipus de dispositius més comuns en una ciutat espanyola.

Aquest laboratori presenta característiques de dos barrets principals de la ciberseguretat. En primer lloc, pel que fa al barret vermell (Red Hat), l'OSINT forma part de la fase de reconeixement que un atacant podria dur a terme per identificar vulnerabilitats en sistemes exposats a Internet. D'altra banda, també es relaciona amb el barret blau (Blue Hat), ja que les eines utilitzades permeten identificar exposicions de serveis propis amb l'objectiu de reforçar-ne la seguretat i prevenir riscos [18].

4.4.2 Teoria del laboratori

Com s'ha explicat prèviament, OSINT correspon a les sigles d'Open Source Intelligence, és a dir, intel·ligència de codi obert. És una eina, o conjunt d'eines, que ha anat guanyant rellevància en l'obtenció d'informació precisa, rellevant i, sobretot, pública. No cal hackejar un sistema ni explotar-ne les vulnerabilitats: tota aquesta informació és accessible a través de pàgines web, xarxes socials, articles, entre d'altres.

En relació amb això, actualment existeixen milions de dispositius connectats a Internet que presenten mesures de seguretat molt pobres, posant en risc tant les persones com les seves dades. Mitjançant OSINT, investigadors i especialistes poden escanejar una gran quantitat de dispositius per identificar-ne les debilitats i investigar quin contingut de la seva infraestructura està públicament accessible.

Per dur a terme aquests escaneigs, algunes de les eines més potents són Shodan i Censys. Ambdues permeten visualitzar quins sistemes estan connectats a Internet, quin programari utilitzen, si està desactualitzat, i detectar configuracions errònies. Per tant, OSINT no implica vulnerar sistemes, sinó observar allò que ja és visible.

Amb aquesta base, el laboratori comença amb una breu introducció al concepte d'OSINT i la seva evolució històrica fins a l'actualitat. Per exemple, el terme OSINT va ser emprat per primera vegada a finals dels anys 1980 per l'exèrcit dels Estats Units [32].

A continuació, s'expliquen les característiques essencials d'OSINT aplicades a la ciberseguretat, destacant-ne tant els avantatges com els inconvenients. Pel que fa als avantatges, OSINT permet accedir a grans volums de dades, sovint en temps real. Tanmateix, el fet que aquestes dades siguin públiques implica que també poden ser explotades per actors amb intencions malicioses [33].

Tot seguit, s'introdueixen les eines utilitzades al laboratori per escanejar dispositius i sistemes: Censys i Shodan. S'explica quan van ser creades i algunes de les seves funcionalitats generals. No obstant això, es procura no aprofundir-hi en excés, ja que una de les activitats consisteix precisament a fer un informe comparatiu entre ambdues eines. També s'hi descriuen els dos principals mètodes d'utilització: a través d'una interfície web per cerques manuals i mitjançant una API per a desenvolupadors.

Ambdós mètodes funcionen de manera similar, com si es tractés d'un cercador com Google. Quan s'introdueix un terme, com ara Windows, el sistema retorna tots els resultats relacionats amb aquest terme. Tot i això, Shodan i Censys operen amb una sintaxi de consultes específica (queries), similar a les consultes que es fan en una base de dades.

Cada sistema, és a dir, Shodan i Censys, disposa de filtres similars, però també presenta diferències, alguns filtres són exclusius d'un sistema i no tenen equivalència directa en l'altre.

Un exemple de consulta en Shodan seria cercar dispositius localitzats a Espanya.

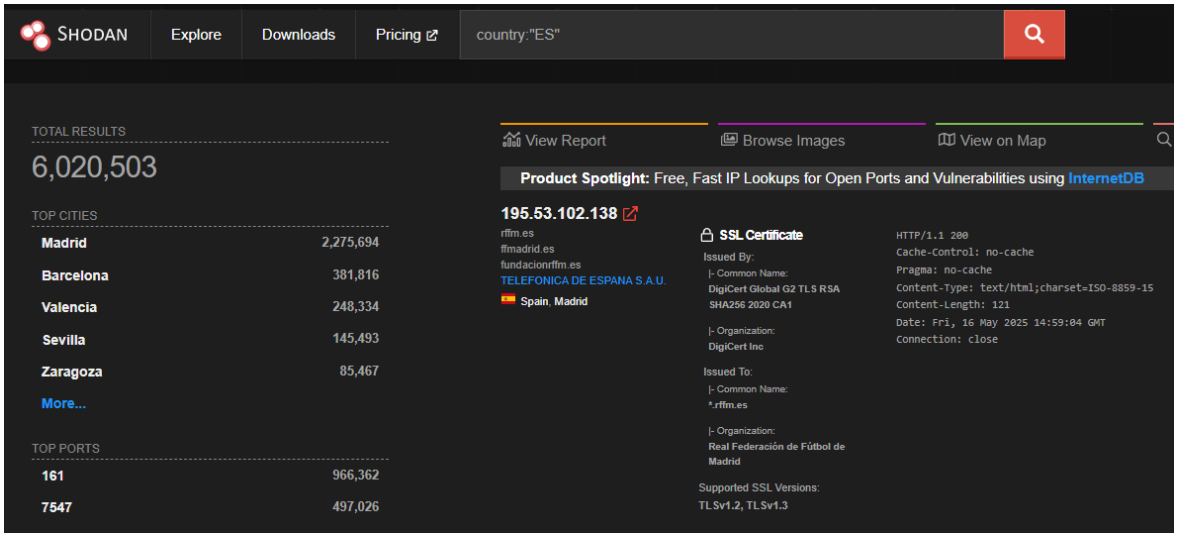

Figura 4.29: Captura de query en Shodan

En el cas de Censys, la consulta seria molt similar, tot i que la sintaxi varia lleugerament.

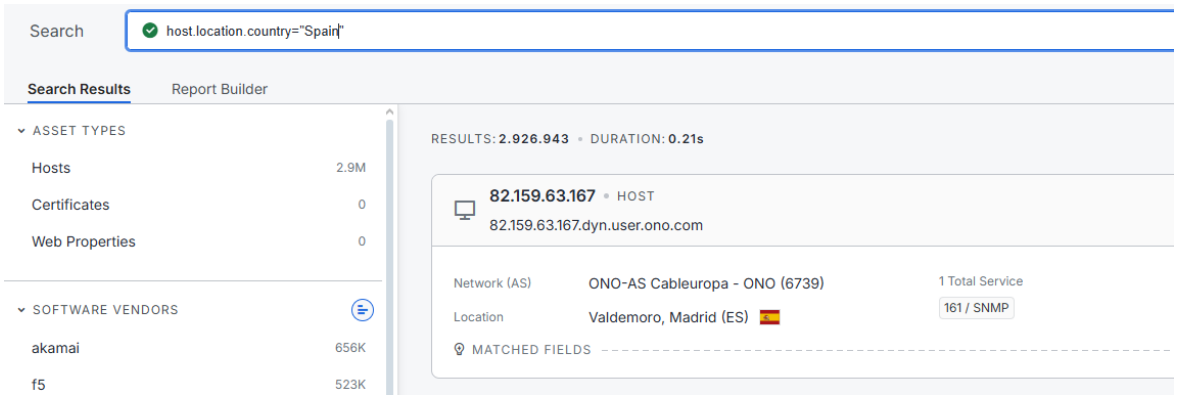

Figura 4.30: Captura de query en Censys

És important remarcar que, tot i que les dues consultes siguin equivalents, els primers resultats que es mostren poden ser diferents, ja que cada sistema utilitza un mecanisme de cerca propi.

Per acabar la part teòrica del laboratori, s'introdueix el concepte de CVE (Common Vulnerabilities and Exposures) o vulnerabilitats i exposicions comunes. Aquestes fan referència a una llista d'amenaces de seguretat públicament conegudes que poden permetre a un atacant accedir als sistemes, a les xarxes o injectar-hi codi maliciós. Les CVEs ja havien estat explicades en un laboratori anterior, però en tornar-les a utilitzar en aquest, cal repassar-ne el concepte. Amb aquesta darrera explicació, es dona pas a la realització de les diferents activitats pràctiques del laboratori.

4.4.3 Activitat 1: Ús de la interfície de cerca web per escanejar

La primera activitat del laboratori es divideix en dues parts diferenciades. En primer lloc, l'objectiu de la part inicial és que els estudiants es familiaritzin amb la interfície de cerca web d'una de les dues eines, Shodan o Censys, per tal d'entendre el seu funcionament i la tipologia de dades que proporciona. Aquesta primera part consisteix a seguir una sèrie de punts concrets que detallen les accions a realitzar dins la plataforma escollida. A partir d'aquestes accions, l'estudiant ha de respondre un conjunt de preguntes per demostrar la comprensió de la informació obtinguda i la manera d'utilitzar l'eina.

Algunes de les preguntes que es plantegen en aquest breu tutorial estan relacionades amb introduir consultes (queries) a la barra de cerca, així com la interacció amb les barres laterals per afegir nous filtres a la consulta inicial. A més, també es demana a l'estudiant que accedeixi al contingut detallat que conté una IP i que observin quins serveis tenen oberts o quines vulnerabilitats presenta.

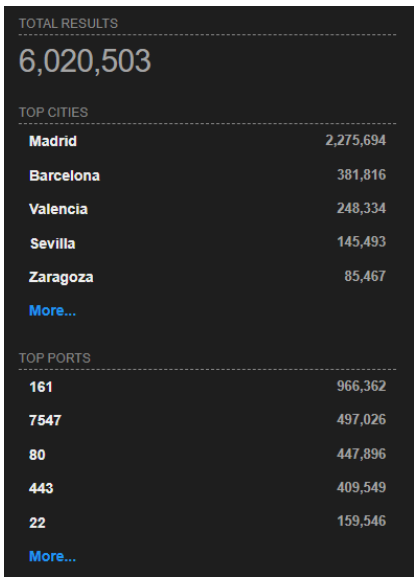

Figura 4.31: Captura de la barra lateral de Shodan

En segon lloc, un cop superada la introducció, es proposa un conjunt de frases relacionades amb dispositius i sistemes connectats a Internet. Aquestes frases han estat formulades amb la intenció que l'estudiant sigui capaç de traduir-les en consultes que puguin ser executades a través de les interfícies web de Shodan o Censys. L'activitat té com a objectiu posar en pràctica la capacitat de transformació de necessitats d'informació en consultes OSINT concretes, utilitzant les opcions de filtratge i els operadors de cerca que ofereixen aquestes plataformes.

Per tal de realitzar les consultes d'aquesta activitat, algunes s'han extret de pàgines web i fòrums especialitzats. D'altres han sorgit a partir de la combinació de consultes senzilles o s'han creat inspirant-se en exemples similars, amb l'objectiu d'elaborar consultes més complexes. Cal destacar que, en alguns casos, Censys requereix una subscripció de tipus Premium, i per tant, els resultats d'algunes consultes no es poden visualitzar sense disposar de la llicència corresponent.

4.4.4 Activitat 2: Informe sobre les diferències entre Shodan i Censys

La segona activitat consisteix, com indica el títol, a realitzar un informe sobre les diferències entre les eines Shodan i Censys. És cert que, en fer una mateixa consulta, com per exemple cercar dispositius amb el sistema operatiu Windows, el primer resultat que ofereix cada eina pot ser diferent. Fins i tot, repetint exactament la mateixa cerca dins una mateixa plataforma, els resultats poden variar lleugerament entre execucions.

Tanmateix, l'aspecte clau d'aquesta activitat no és el contingut concret dels resultats, sinó entendre com funciona el procés de cerca en cada cas. Cadascuna de les dues eines aplica mètodes diferents de l'hora d'obtenir i mostrar la informació, fet que genera resultats i perspectives distintes.

Per aquest motiu, es demana a l'estudiant que analitzi i exposi aquestes diferències mitjançant un text argumentat. Per exemple, una de les principals distincions està en la manera de recollir les dades: mentre Shodan es focalitza en la indexació de dispositius a partir de la detecció de banners i metadades exposades per serveis actius, Censys posa èmfasi en la recopilació estructurada d'informació sobre protocols i serveis, incloent-hi la captura de certificats SSL i la identificació de vulnerabilitats [34].

Per tal de facilitar aquesta tasca comparativa, s'adjunten diversos enllaços i documents que poden ser utilitzats com a font d'informació i suport per a l'elaboració de l'informe.

4.4.5 Activitat 3: Informe de dispositius amb l'API de Shodan i Censys

La tercera activitat del laboratori consisteix a realitzar diferents consultes a l'API de Shodan mitjançant Python, i elaborar un informe a partir de la informació obtinguda. Per dur a terme l'activitat, és necessari disposar d'una clau API (API Key) de la plataforma escollida, la qual permet accedir a les seves funcionalitats programàtiques.

En la petició a l'API es limita el nombre de resultats a una mostra màxima de 1000 dispositius. Aquesta limitació es justifica pel fet que les claus d'accés tenen un nombre restringit de consultes; per tant, obtenir massa dades podria provocar l'esgotament del crèdit disponible i dificultar la feina dels altres companys. Per aquest motiu, es demana als estudiants que emmagatzemin els resultats en un arxiu de tipus JSON, ja que és una manera senzilla, eficient i compatible amb moltes llibreries de Python.

A partir d'aquesta mostra de dades, s'ha de generar un conjunt de gràfics i mapes que representin visualment la informació més rellevant. Per exemple, un dels gràfics proposats és la generació d'un mapa de punts geogràfics que indica la localització dels dispositius detectats. A més de marcar la posició sobre el mapa, s'utilitzen colors diferents per indicar la densitat de dispositius en cada coordenada. Aquesta diferenciació visual facilita la identificació de zones amb alta concentració de dispositius potencialment vulnerables.

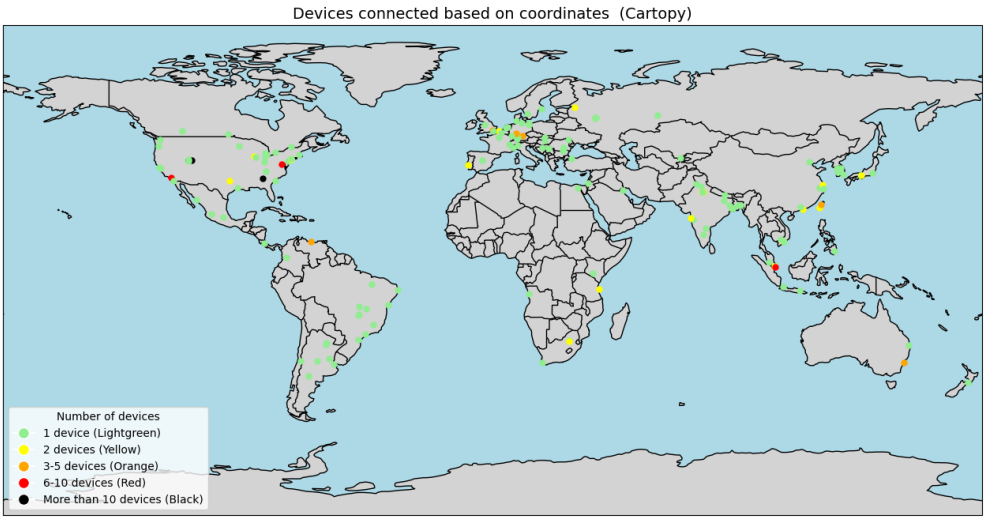

Figura 4.32: Mapa on es mostren tots els dispositius escanejats

Per altra banda, un dels gràfics proposats mostra el top 10 de països amb més IPs vulnerables. Aquest gràfic de barres representa quins dispositius, un cop escanejats, disposen d'alguna vulnerabilitat identificada mitjançant una CVE. Aquest tipus de gràfic permet visualitzar fàcilment en quins països es concentra un major nombre de vulnerabilitats i ajuda a interpretar millor les dades obtingudes.

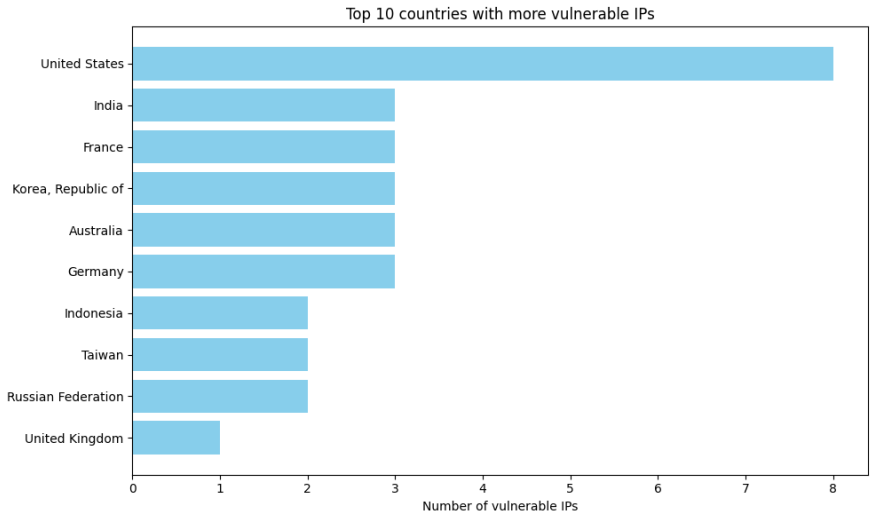

Figura 4.33: Gràfic amb el top 10 de països amb més IPs vulnerables

Per generar aquests gràfics i mapes, l'estudiant haurà de fer ús de diverses llibreries de Python. A cada gràfic s'haurà d'incorporar una breu explicació que en descrigui el contingut i les conclusions que se'n poden extreure.

Finalment, el fet de limitar aquesta activitat a una única petició respon a la idea que l'estudiant ja s'ha familiaritzat amb les cerques manuals a la primera activitat. Ara, el focus es posa en l'explotació del potencial analític i visual de les dades, i no tant en la interacció directa amb la plataforma.

4.4.6 Activitat 4: Facets amb l'API de Shodan

La darrera activitat del laboratori continua treballant amb l'API de Shodan mitjançant Python, però introdueix una funcionalitat addicional molt rellevant per a l'anàlisi de dades: l'eina facets. Aquesta eina permet agrupar i comptar els resultats obtinguts a partir d'atributs específics. En lloc de retornar totes les IPs trobades en una cerca, facets ofereix un resum estadístic que permet analitzar la informació de manera molt més eficient i estructurada.

A partir d'aquesta funcionalitat, es planteja als estudiants el desenvolupament d'un script en Python que permeti a l'usuari introduir el nom d'una ciutat espanyola. A partir d'aquesta entrada, el programa ha d'obtenir i mostrar un conjunt de dades estadístiques associades a dispositius detectats en aquesta ubicació concreta. Concretament, es demana que l'script sigui capaç de mostrar:

- Els 5 ports oberts més comuns.
- Les 10 organitzacions més utilitzades.
- Els 6 sistemes operatius més detectats.
- Els 8 tipus de dispositius més freqüents (com ara càmeres IP, impressores, rúters, etc.).

A més de la generació d'aquestes estadístiques, també es requereix que la informació s'acompanyi de gràfics visuals per tal de facilitar-ne la comprensió a l'usuari. Aquests gràfics han de representar clarament els valors obtinguts, cosa que permet una anàlisi intuïtiva i visual del paisatge digital de la ciutat seleccionada.

Aquesta activitat posa èmfasi en l'ús intel·ligent de dades agregades i en la representació gràfica com a eina de suport per a la presa de decisions i la detecció de patrons de vulnerabilitat en l'àmbit territorial.

5 Anàlisi de resultats

Amb la finalitat d'avaluar l'efectivitat, la claredat i l'impacte dels laboratoris desenvolupats, es va elaborar un formulari estructurat destinat als estudiants de l'assignatura que havien realitzat les activitats. L'objectiu principal és obtenir una valoració crítica i detallada sobre diversos aspectes dels laboratoris, així com detectar àrees de millora i recollir percepcions generals sobre l'experiència formativa. En aquest anàlisi només s'enfoca en el laboratoris de criptografia clàssica i moderna ja que els dos darrers estan presentats i estan en procés de desenvolupament, o encara no han sigut realitzats per l'alumnat.

El qüestionari es divideix en dues parts diferenciades, una per a cada laboratori. Cada secció inclou preguntes tancades amb escales de resposta del tipus Likert de 5 punts, i també espais per comentaris oberts. Finalment, s'inclou una secció general sobre cadascun dels laboratoris. Els valors emprats en les escales eren: (1 = valoració molt baixa / molt insatisfactòria, 5 = valoració molt alta / molt satisfactòria). Per cada pregunta, el text varia segons la pregunta, però en general, l'escala que s'utilitza és la comentada prèviament [35].

Les preguntes i respostes formulades eren les següents:

- Com valoraries, en general, l'experiència de fer el laboratori de criptografia clàssica / moderna? (1 = Molt insatisfet, 2 = Insatisfet, 3 = Neutral, 4 = Satisfet, 5 = Molt satisfet)
- El contingut del laboratori era adequat al teu nivell de coneixements? (1 = Molt inadequat, 2 = inadequat, 3 = Neutral, 4 = adequat, 5 = Molt adequat)
- Has trobat clares les instruccions i explicacions del laboratori de criptografia clàssica / moderna? (1 = Gens clar, 2 = Lleugerament clar, 3 = Moderadament clar, 4 = Majoritàriament clar, 5 = Molt clar)
- Quines dificultats vas tenir durant la realització d'aquest laboratori?
- Creus que aquest laboratori t'ha ajudat a entendre millor els conceptes de la criptografia clàssica / moderna? (1 = Gens útil, 2 = Poc útil, 3 = Moderadament útil, 4 = Útil, 5 = Molt útil)
- Què milloraríeu o afegiríeu al laboratori de criptografia clàssica / moderna per fer-lo més útil o interessant?

A partir d'aquí, son preguntes de la secció general.

- Quin dels dos laboratoris t'ha semblat més interessant? Per què?
- T'ha motivat a voler aprendre més sobre criptografia o ciberseguretat? (1 = Gens, 2 = Lleugerament, 3 = Moderadament, 4 = Bastant, 5 = Molt)
- Recomanaries aquests laboratoris a altres estudiants? (1 = Definitivament no, 2 = Probablement no, 3 = No n'estic segur, 4 = Probablement sí, 5 = Definitivament sí)
- Hi ha algun comentari addicional que voldries fer sobre els laboratoris?

Cal destacar que aquests formularis s'han passat als 19 estudiants que cursen l'assignatura d'Administració i Seguretat de Sistemes, on entre ells, en el moment de fer l'anàlisi, només l'han respost 12 al formulari.

Els resultats obtinguts reflecteixen, en línies generals, una valoració positiva per part dels estudiants. Pel que fa a l'experiència global, tots els participants situen les seves respostes entre els nivells 3 i 5, tant en el laboratori de criptografia clàssica com en el de moderna. Es destaca que el laboratori de criptografia moderna rep una valoració lleugerament superior, especialment s'entenc per la seva percepció de ser més aplicable a entorns tecnològics actuals.

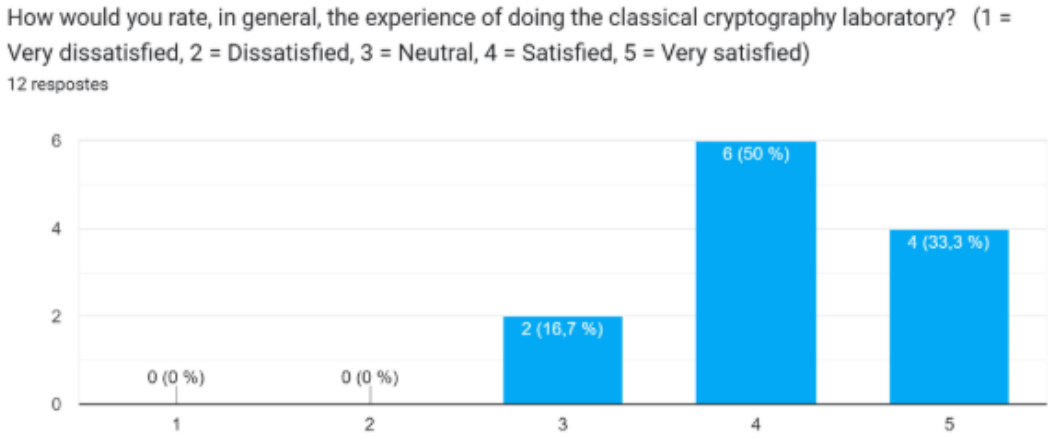

Figura 5.1: Pregunta de com valoraries, en general, l'experiència de fer el laboratori de criptografia clàssica

Respecte a la claredat de les instruccions, les respostes mostren que, en general, són prou entenedores. Tot i això, es detecta algun cas en què les instruccions del laboratori de criptografia clàssica han estat considerades poc clares, i diversos estudiants han assenyalat la manca d'exemples de codi com una dificultat rellevant.

Aquesta observació es repeteix també en les respostes sobre la criptografia moderna, on s'indica que l'absència de guies o d'exemples concrets, ha dificultat la seva implementació. Tots aquests aspectes es tindran en compte de cara a la impartició futura de l'assignatura.

Quant a l'adequació del contingut, les puntuacions oscil·len entre el 3 i el 5, cosa que suggereix que, tot i ser adequats, els laboratoris poden beneficiar-se d'ajustos per adequar-los als estudiants. Entre les propostes, sobretot per el laboratori de criptografia clàssica, existeix un cas on un estudiant suggereix afegir alguna activitat d'atac de força bruta per trencar tant el xifrat Vigènere, com el Rail Fence. Un altre suggeriment és el de combinar la criptografia amb una altra tecnologia. Pel que fa al laboratori de criptografia moderna, només destacar afegir més explicacions i exemples de d'implementacions.

D'altra banda, la gran majoria considera que els dos laboratoris de criptografia els han ajudat a comprendre millor els conceptes treballats.

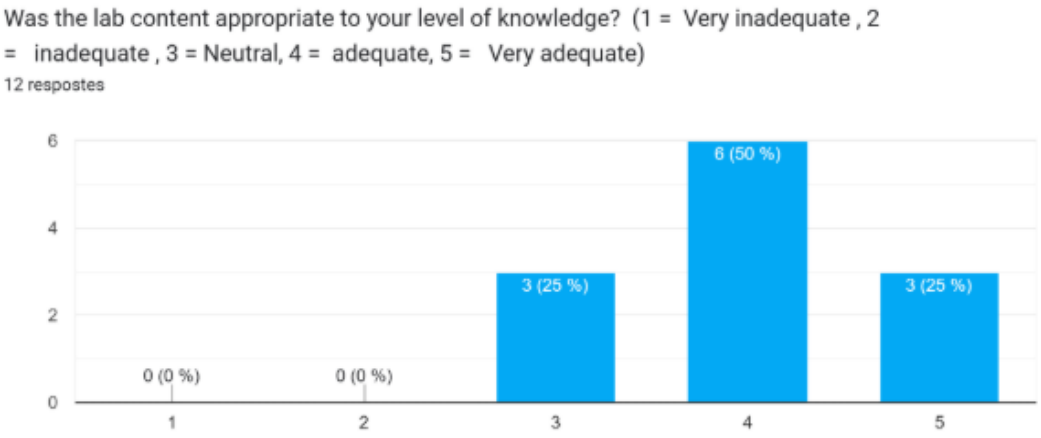

Figura 5.2: Pregunta del laboratori de criptografia moderna de si el contingut del laboratori era adequat al teu nivell de coneixements?

A més, la comparació entre els dos laboratoris revela que la majoria dels participants troba més interessant el laboratori de criptografia moderna, valorant especialment per la relació directa amb casos d'ús reals. Pel que fa a la motivació per continuar aprenent, en general, indiquen que els laboratoris els han motivat moderadament o molt, mentre que existeix una excepció on un estudiant no s'ha sentit motivat.

Finalment, a la pregunta sobre si recomanarien els laboratoris a altres estudiants, quatre dels cinc participants responen afirmativament, mentre que un estudiant expressa dubtes. Aquest resultat reafirma l'interès i la utilitat de les activitats, tot i que apunta a la necessitat de petites millores, explicades prèviament, per millorar la seva experiència global.

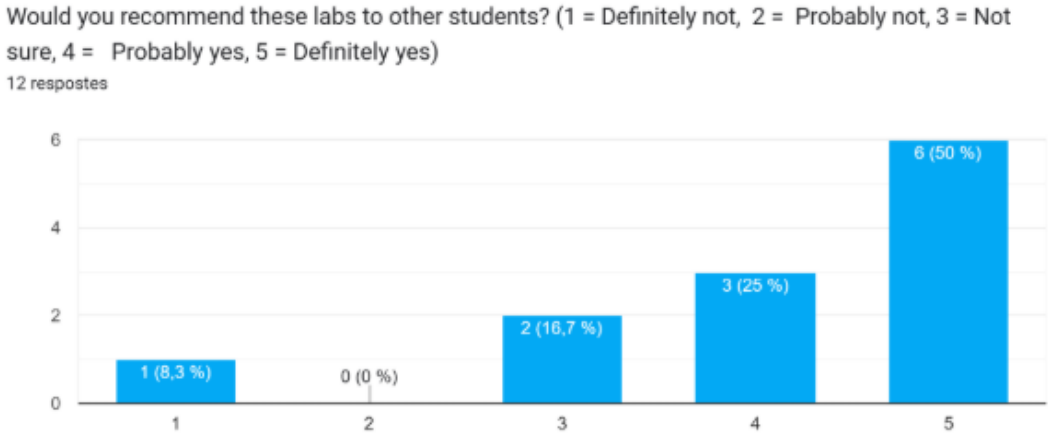

Figura 5.3: Pregunta de si recomanaries aquests laboratoris a altres estudiants?

Pel que fa als comentaris finals, un estudiant destaca que la durada dels laboratoris és adequada i no considera necessari allargar-los més.

En conclusió, els resultats dels formularis mostren que els laboratoris desenvolupats han assolit el seu objectiu principal, que és facilitar la comprensió pràctica dels conceptes de criptografia, alhora que proporcionen una base sòlida per realitzar millores futures i adaptar millor les activitats a les necessitats de l'alumnat.

6 Conclusions i treball futur

6.1 Conclusions

La valoració global del Treball de Final de Grau és satisfactòria. La metodologia iterativa i incremental emprada ha resultat útil, ja que ha permès incorporar millores i actualitzacions progressives als laboratoris amb l'objectiu d'oferir activitats més completes i adaptades a l'aprenentatge pràctic.

Tot i això, la planificació inicial no s'ha pogut complir íntegrament a causa de les dificultats trobades en el desenvolupament de determinats laboratoris previstos en les primeres fases. Aquestes desviacions han obligat a reorganitzar el contingut i prioritzar aquells laboratoris que aportaven més valor pedagògic i tècnic dins del temps disponible.

Els laboratoris desenvolupats proporcionen una base sòlida per continuar aprofundint en àmbits avançats de la ciberseguretat, com ara l'anàlisi de vulnerabilitats, les tècniques de defensa activa o la gestió de la intel·ligència d'amenaces. A més, ofereixen escenaris útils tant per a la pràctica docent com per a l'autoformació d'estudiants i professionals interessats en aquest camp.

La integració dels laboratoris dins l'assignatura d'Administració i Seguretat de Sistemes ha generat una resposta positiva per part dels estudiants participants. Mitjançant els qüestionaris de valoració, s'han identificat els punts forts i les àrees de millora de cada activitat, la qual cosa ha permès aplicar ajustos i enriquir-ne el contingut.

El procés de desenvolupament ha representat una oportunitat d'aprenentatge continu, que ha contribuït a consolidar i ampliar els coneixements adquirits al llarg dels estudis universitaris. La superació dels reptes plantejats, alguns dels quals inicialment inabastables, ha representat una evolució significativa en l'àmbit tècnic i personal.

Finalment, aquest projecte pot servir com a porta d'entrada a noves oportunitats tant professionals com acadèmiques, així com a recurs formatiu per a tercers.

6.2 Treball futur

En aquest apartat, es descriuen possibles ampliacions del treball o activitats que, per limitacions de temps o abast, no s'han inclòs en aquesta versió del Treball de Final de Grau:

- **Laboratori de comunicacions futures**: S'ha desenvolupat una activitat basada en el protocol criptogràfic Diffie-Hellman, però no s'ha incorporat al laboratori de criptografia moderna a causa de la càrrega de treball associada a la resta d'activitats. Aquest protocol permet establir una clau secreta compartida entre dues entitats a través d'un canal públic, sense transmetre la clau directament. La seva implementació implica diversos intercanvis de claus i una combinació matemàtica que dona com a resultat una clau comuna. Encara que una entitat maliciosa intercepte l'intercanvi, no pot derivar la clau compartida sense conèixer les claus privades de les dues parts.
- **Laboratori de Phishing**: Es planteja la creació d'un laboratori enfocat a la simulació d'atacs de phishing utilitzant la plataforma GoPhish. Aquest entorn permet dissenyar i gestionar campanyes de correu electrònic maliciós de manera controlada. L'objectiu principal és mostrar el funcionament d'aquest tipus d'atacs, analitzar els mecanismes d'enginyeria social que fan servir i estudiar com poden ser detectats i mitigats en entorns reals.
- **Laboratori de DDOS**: Es proposa un laboratori orientat a la simulació i anàlisi d'atacs de denegació de servei distribuïda (DDoS). Aquests atacs tenen com a objectiu saturar un servei o una aplicació enviant un volum massiu de tràfic a través d'una xarxa de dispositius compromesos (botnet). L'activitat permetria entendre el funcionament d'aquests atacs, els seus efectes i les estratègies bàsiques de detecció i defensa.
- **Securització de serveis**: Es proposa un laboratori on es proporciona una màquina virtual amb serveis vulnerables que l'estudiant ha d'analitzar i protegir. L'activitat permetria aplicar pràctiques com el reforç de configuracions, l'anàlisi de ports oberts, la gestió de permisos, la limitació de serveis innecessaris i la implementació de controls d'accés per reduir la superfície d'atac.

Bibliografia

[1] S. G. Bordonado. "DETECCIÓN Y ANÁLISIS DE ARTEFACTOS EN LOS PRINCIPALES TIPOS DE CIBERATAQUES." [En línia] [consulta: 2024/12/14]. (2021), Disponible: https://repositorio.uam.es/bitstream/handle/10486/698264/garcia_bordonado_sergio_tfg.pdf?sequence=1&isAllowed=y.

[2] Wikipedia. "2008 malware infection of the United States Department of Defense." [En línia] [consulta: 2025/05/22]. (), Disponible: https://en.wikipedia.org/wiki/2008_malware_infection_of_the_United_States_Department_of_Defense.

[3] Wikipedia. "Agent.BTZ." [En línia] [consulta: 2025/05/22]. (), Disponible: https://en.wikipedia.org/wiki/Agent.BTZ.

[4] S. Marshall. "Top 10 Most Common Types of Network Attacks." [En línia] [consulta: 2024/12/14]. (2024), Disponible: https://www.lepide.com/blog/common-types-of-network-attacks/.

[5] 21y4d. "5 common web attacks: How to exploit and defend against them." [En linia] [consulta: 2024/12/14]. (2024), Disponible: https://www.hackthebox.com/blog/5-common-web-attacks.

[6] Fortinet. "What are Social Engineering Attacks?" [En linia] [consulta: 2024/12/14]. (), Disponible: https://www.fortinet.com/lat/resources/cyberglossary/social-engineering.

[7] S. Burge. "8 Types of Attack in Cryptography." [En linia] [consulta: 2024/12/14]. (2024), Disponible: https://internationalsecurityjournal.com/types-of-attack-in-cryptography.

[8] Hack The Box. "Hack The Box - Cyber Security Training." [En línia] [consulta: 2025/05/22]. (), Disponible: https://www.hackthebox.com.

[9] Try Hack me. "Try Hack me - Learn Cyber Security." [En línia] [consulta: 2025/05/22]. (), Disponible: https://tryhackme.com.

[10] B. N. Mundo, "El virus que tomó control de mil máquinas y les ordenó autodestruirse - BBC News Mundo," *BBC News Mundo*, 2015, [En linia] [consulta: 2024/12/14]. Disponible: https://www.bbc.com/mundo/noticias/2015/10/151007_iwonder_finde_tecnologia_virus_stuxnet.

[11] M. Baezner, "CSS CYBER DEFENSE HOTSPOT ANALYSIS," Center for Security Studies (CSS), ETH Zürich, inf. tèc., 2017, [En linia] [consulta: 2024/12/14]. Disponible: https://doi.org/10.3929/ethz-b-000200661.

[12] G. G. Paredes, "Introducción a la criptografía," *Revista Digital Universitaria*, vol. 7, núm. 7, jul. de 2006, Artículo n.º 55, [En linia] [consulta: 2024/11/14]. Disponible: https://biblat.unam.mx/hevila/Revistadigitaluniversitaria/2006/vol7/no7/5.pdf.

[13] Cloudflare. "¿Qué es la exfiltración de datos?" [En linia] [consulta: 2024/12/19]. (), Disponible: https://www.cloudflare.com/es-es/learning/security/what-is-data-exfiltration/.

[14] Proofpoint. "Fuga de información o exfiltración - Significado y prevención." [En linia] [consulta: 2024/12/19]. (), Disponible: https://www.proofpoint.com/es/threat-reference/data-exfiltration.

[15] Amazon. "¿Qué es ICMP?" [En linia] [consulta: 2024/12/19]. (), Disponible: https://aws.amazon.com/es/what-is/icmp/.

[16] J. Snow. "Shodan y Censys: los peligrosos motores de búsqueda del Internet de las Cosas." [En linia] [consulta: 2024/12/19]. (2016), Disponible: https://www.kaspersky.es/blog/shodan-censys/7827.

[17] V. Arranz. "Shodan: Explorando las Profundidades de Internet." [En línia] [consulta: 2024/12/19]. (), Disponible: https://www.campuscyberseguridad.com/blog/item/176-shodan-explorando-las-profundidades-internet.

[18] SoftwareSecured. "Building a Balanced Security Team: The 7 Hacker Hats Explained." [En línia] [consulta: 2025/05/22]. (), Disponible: https://www.softwaresecured.com/post/the-7-hats-of-hacking.

[19] Wikipedia. "Caesar Cipher." [En línia] [consulta: 2025/03/29]. (), Disponible: https://en.wikipedia.org/wiki/Caesar_cipher.

[20] S. K. '03. "Railfence Cipher." [En línia] [consulta: 2025/03/29]. (2010), Disponible: https://www.cs.trincoll.edu/~crypto/historical/railfence.html.

[21] GeekforGeeks. "Cryptanalysis and Types of Attacks." [En línia] [consulta: 2025/03/21]. (2024), Disponible: https://www.geeksforgeeks.org/cryptanalysis-and-types-of-attacks/.

[22] NLTK. "NLTK :: Natural Language Toolkit." [En línia] [consulta: 2025/05/22]. (), Disponible: https://www.nltk.org.

[23] S. Group. "¿Qué es un hash y cómo funciona?" [En línia] [consulta: 2025/03/28]. (), Disponible: https://www.signaturit.com/es/blog/que-es-un-hash/.

[24] Cryptography. "Welcome to pyca/cryptography." [En línia] [consulta: 2025/05/22]. (), Disponible: https://cryptography.io/en/latest/.

[25] Cloudflare. "¿Qué es la exfiltración de datos?" [En línia] [consulta: 2025/03/1]. (), Disponible: https://www.cloudflare.com/es-es/learning/security/what-is-data-exfiltration/.

[26] SentinelOne. "Data Exfiltration." [En línia] [consulta: 2025/03/1]. (2024), Disponible: https://www.sentinelone.com/cybersecurity-101/cybersecurity/data-exfiltration/.

[27] M. Report. "ICMP Tunneling y Riesgos de Ciberseguridad." [En línia] [consulta: 2025/02/22]. (2024), Disponible: https://mineryreport.com/blog/icmp-tunneling-riesgo-ciberseguridad/.

[28] P. A. Networks. "¿Qué es el DNS Tunneling?" [En línia] [consulta: 2025/02/22]. (), Disponible: https://www.paloaltonetworks.es/cyberpedia/what-is-dns-tunneling.

[29] Wikipedia. "HTTP Tunnel." [En línia] [consulta: 2025/02/22]. (), Disponible: https://en.wikipedia.org/wiki/HTTP_tunnel.

[30] PortSwigger. "Request Tunneling." [En línia] [consulta: 2025/02/22]. (), Disponible: https://portswigger.net/web-security/request-smuggling/advanced/request-tunnelling.

[31] Avast. "¿Qué es un Keylogger?" [En línia] [consulta: 2025/02/25]. (), Disponible: https://www.avast.com/es-es/c-keylogger.

[32] osint.industries. "OSINT Basics: What is OSINT?" [En línia] [consulta: 2025/03/28]. (), Disponible: https://www.osint.industries/post/osint-basics-what-is-osint.

[33] Y.-W. Hwang, I.-Y. Lee, H. Kim, H. Lee i D. Kim, "Current Status and Security Trend of OSINT," *Wiley Online Library*, febr. de 2022, [En línia] [consulta: 2025/03/28]. Disponible: https://onlinelibrary.wiley.com/doi/epdf/10.1155/2022/1290129.

[34] J. Krupp, J. Schrittwieser, T. Schneider i C. Rossow, "Understanding the Ecosystem of Internet Exposure Notifications," a *MADWeb Workshop on Measurements, Attacks, and Defenses for the Web (co-located with NDSS)*, [En línia] [consulta: 2025/03/28], 2021. Disponible: https://www.ndss-symposium.org/wp-content/uploads/madweb2021_23009_paper.pdf.

[35] Wikipedia. “Escala Likert.” [En línia] [consulta: 2025/05/22]. (), Disponible: https://es.wikipedia.org/wiki/Escala_Likert.