Administració de la seguretat

PID_00275592

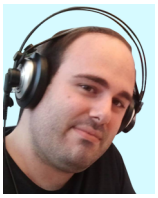
José Manuel Castillo Pedrosa
Javier Panadero Martínez

Temps mínim de dedicació recomanat: 8 hores

Universitat Oberta de Catalunya

José Manuel Castillo Pedrosa

Enginyer en Informàtica i llicenciat en Economia per la Universitat Autònoma de Barcelona (UAB). Des de l'any 2005, és professor col·laborador a la Universitat Oberta de Catalunya (UOC) de l'assignatura Administració de xarxes i sistemes operatius i de treballs finals de grau, participant en tribunals d'avaluació i coautor de materials d'aprenentatge. Més de vint anys d'experiència professional com a responsable de projectes TIC, especialitzat en sistemes Linux i UNIX, clústers d'alta disponibilitat, emmagatzemament, virtualització, sistemes gestors de bases de dades, servidors d'aplicacions, seguretat i *backup*.



Javier Panadero Martínez

Enginyer informàtic i doctor en Computació d'Altes Prestacions per la Universitat Autònoma de Barcelona (UAB). Des de 2019, és professor dels Estudis d'Informàtica, Multimèdia i Telecomunicació de la Universitat Oberta de Catalunya (UOC). Director del màster universitari en Enginyeria Computacional i Matemàtica. Ha elaborat diversos materials sobre administració de sistemes i programació. Els seus interessos de recerca inclouen la computació paral·lela i distribuïda, l'optimització i simulació de sistemes complexos i els algorismes intel·ligents.

Primera edició: setembre 2020

Av. Tibidabo, 39-43, 08035 Barcelona
Autoria: José Manuel Castillo Pedrosa, Javier Panadero Martínez
Producció: FUOC

Índex

Introducció

El concepte de seguretat informàtica és difús i pràcticament inabastable, per la qual cosa serà preferible centrar la nostra atenció en el que podríem anomenar fiabilitat, entesa com la garantia de la qualitat de servei d'un sistema en l'àmbit de les tecnologies de la informació i la comunicació (TIC).

En aquest mòdul veurem quan es pot comprometre aquesta fiabilitat, i també les eines que un administrador té a la seva disposició per tal d'implementar les mesures de defensa amb l'objectiu de prevenir, detectar i respondre adequadament quan es produeixi un incident relacionat amb la seguretat.

Estudiarem la naturalesa dels elements que constitueixen les amenaces més importants: els errors del programari, el codi maliciós, els atacs de denegació de servei, els atacs d'intermediari i l'enginyeria social.

Veurem que un dels principis fonamentals en els quals s'ha de sustentar el disseny de la seguretat és que un sistema TIC, considerat com a conjunt, és tan fiable com el seu component més dèbil. Per tant, s'ha de tenir una visió integral que prengui en consideració la seguretat física del servidor, dels sistemes operatius, de les aplicacions, de les dades en qualsevol dels seus estats (en transmissió, en ús quan s'estan processant o en repòs quan estan emmagatzemades), de la xarxa i dels recursos desplegats al núvol.

És primordial no descuidar el factor humà: ajudar a revisar els processos, participar en el pla de formació i conscienciar els responsables de l'organització de la importància de la seguretat, per tal d'obtenir el seu compromís i recolzament en l'aplicació de totes les mesures necessàries, són tasques que ha d'assumir com a prioritàries l'administrador de sistemes.

Objectius

En els materials didàctics associats a aquest mòdul, l'estudiant trobarà les eines i els continguts necessaris per assolir els objectius següents:

1. Conèixer les qüestions bàsiques que comporta l'administració de la seguretat d'un sistema TIC.

2. Saber quines són les responsabilitats que té un administrador envers els sistemes i les dades, tant les que li són pròpies per la seva funció com les derivades del marc legal.

3. Conèixer què cal fer, una vegada s'ha produït un incident de seguretat, per determinar què ha succeït, establir qui ha estat el presumpte responsable i recuperar els sistemes per restablir els serveis.

1. Seguretat de les TIC

S'entén que quelcom és segur quan està lliure i exempt de risc. Específicament en l'àmbit de les tecnologies de la informació i la comunicació (TIC), resulta complicat formular una definició de seguretat formal i, a més, inviable, a causa dels continus canvis que s'experimenten en aquest camp.

Aquí establirem el concepte de **seguretat** com el conjunt de metodologies, documentació, programari i maquinari que determinen que l'accés als recursos d'un sistema TIC sigui realitzat exclusivament pels agents autoritzats.

En general, no hi ha un sistema perfectament segur a la pràctica. Una frase il·lustre que recull molt bé aquesta idea la va encunyar l'expert en seguretat Eugene H. Spafford l'any 1989:

> «L'únic sistema veritablement segur és aquell que està desconnectat de l'alimentació elèctrica, dins d'un bloc de formigó, i tancat en un búnquer amb vigilants armats, i amb tot i això tinc els meus dubtes».
>
> Spafford (1989).

Aquesta impossibilitat és a causa que els sistemes TIC solen ser altament complexos i, per tant, amb una alta probabilitat que puguin contenir fallades que representin vulnerabilitats.

Un exemple paradigmàtic d'això el representa el programari, que s'ha caracteritzat històricament per tenir errors de disseny i implementació que deriven, com a conseqüència, en l'obtenció de resultats incorrectes i no desitjats en general, i, específicament, en la possibilitat de permetre que els atacants puguin obtenir privilegis sobre els recursos sense autorització.

El factor humà, que juga un paper primordial en els sistemes, també representa un dels seus elements més fràgils quan es considera la seguretat. L'enginyeria social, que és la manipulació psicològica per part dels agents maliciosos sobre les persones per tal que aquestes facin determinades accions, s'ha demostrat que és especialment efectiva per eludir, fins i tot, les mesures de protecció tecnològiques més avançades.

Per acabar-ho de complicar, hi ha diversos i forts incentius per a determinats actors per subvertir la seguretat dels sistemes, com poden ser: el guany econòmic, l'activisme, l'espionatge, la recerca de reconeixement i prestigi en un col·lectiu, el merament recreatiu i la satisfacció de la pròpia curiositat, entre d'altres.

Atès que és del tot impossible garantir la seguretat o inviolabilitat absoluta d'un sistema, en lloc de l'inabastable concepte de seguretat, serà preferible parlar de **fiabilitat**, entesa com la probabilitat que un sistema es comporti tal com s'espera. Considerant això, ens haurem de preguntar: quin és el nivell apropiat de seguretat desitjable i factible a la pràctica per obtenir la màxima fiabilitat dels sistemes? Quina és la inversió requerida per protegir els nostres recursos de la manera més eficient?

S'ha d'acceptar que sempre hi haurà vulnerabilitats que per ser corregides caldria una inversió inviable considerant els costos. Com que els recursos de les organitzacions són limitats, s'ha de passar de considerar la gestió de la seguretat com una qüestió de blanc o negre a una d'escala de grisos, i entrar de ple en la disciplina de la gestió de riscos. En aquest context, entendrem com a **risc** la quantificació o mesura de la magnitud dels danys ocasionats en cas de produir-se un fet determinat. Hi ha moltes maneres d'obtenir aquesta quantificació, però una de senzilla pot ser la següent:

Risc associat a un esdeveniment =
Probabilitat que passi × Pèrdua ocasionada

Hem de considerar, doncs, el risc que representa la inoperativitat, un mal funcionament o pèrdua d'un recurs, i invertir en la seva protecció de manera proporcionada.

La política de seguretat d'una organització determinarà documentalment quins recursos es pretendrà protegir, s'identificaran i definiran clarament les responsabilitats, i s'assignaran aquestes als membres de l'organització que s'estableixi. La comunicació, la implementació i l'execució de la política de seguretat s'hauran d'avaluar i revisar de manera contínua per tal que pugui ser eficaç.

1.1. Principis de seguretat

La seguretat es basa en tres principis generals fonamentals, anomenats en el seu conjunt com la **tríada CIA**.[1] Representen la base en què s'ha de sustentar tota consideració que es faci en relació amb la protecció dels recursos:

[(1)]Sigles en anglès de *confidentiality, integrity, availability*.

- **Confidencialitat**. Només es pot accedir als recursos de manera autoritzada.

- **Integritat**. Només es poden modificar els recursos de manera autoritzada.
- **Disponibilitat**. Els recursos han de romandre accessibles i només per als elements autoritzats.

Un altre concepte important és el de **principi del mínim privilegi**:[2] un agent (un usuari, un procés o un programa, per exemple) només ha de tenir accés als recursos que siguin estrictament necessaris perquè pugui realitzar la seva funció.

(2)En anglès, *principle of least privilege*.

Per exemple, no hauríem de mantenir oberta una sessió com a usuari privilegiat al nostre ordinador per treballar, si no hem de fer cap tasca administrativa que requereixi drets d'administració, ja que així un error humà o del programari poden tenir conseqüències molt més destructives.

1.2. Atacs

Un recurs d'un sistema és un element que té per nosaltres un valor i que, per tant, podem considerar protegir-lo d'alguna determinada manera per intentar salvaguardar-lo. El recurs pot ser **físic** (com per exemple: servidors, cabines de discs, equipaments de xarxa) o **lògic** (entre d'altres: dades, aplicacions, sistemes operatius, capacitat de còmput, espai d'emmagatzematge, amplada de banda).

La debilitat, error o fallada d'un sistema susceptible de ser atacat amb èxit rep el nom de **vulnerabilitat**. N'hi ha de diversos tipus, entre les quals destacarem les següents: físiques, naturals, humanes, de programari, de maquinari i de comunicació.

Definirem **amenaça** com el perill que una vulnerabilitat pugui ser atacada amb èxit (en aquest circumstància, direm que l'amenaça ha estat explotada).

S'anomena **atac** a l'acte deliberat que té com a objectiu transgredir la seguretat d'un sistema. D'aquells atacs que intenten modificar els recursos o afectar el seu funcionament d'alguna manera, en direm actius. Els atacs que només intenten accedir als recursos però sense fer-hi cap modificació ni afectar-los en el seu funcionament, es diuen passius. A banda d'aquesta primera classificació, podem fer la següent agrupació dels tipus d'atacs existents:

1) Interrupció: és un atac actiu contra la disponibilitat en què un agent no autoritzat fa que un recurs es perdi, no pugui utilitzar-se o no s'hi pugui accedir.

Per exemple: destruir físicament un servidor, esborrar un programa o realitzar un atac de denegació de servei.

2) Intercepció: és un atac passiu contra la confidencialitat en què un agent no autoritzat accedeix a un recurs.

Per exemple: copiar un fitxer o espiar una comunicació.

3) Modificació: és un atac passiu contra la integritat en què un agent no autoritzat accedeix a un recurs i, a més, el modifica.

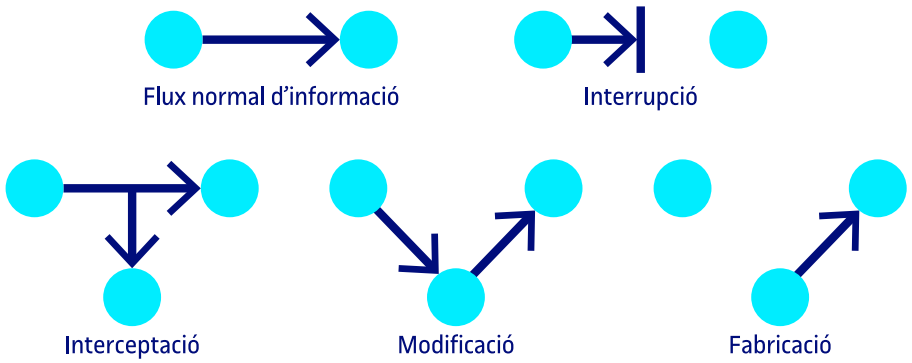
Per exemple: canviar el valor d'un camp d'una taula d'una base de dades.

4) Fabricació: és un atac actiu contra la integritat en què un agent no autoritzat crea o inserta objectes falsificats en un sistema.

Per exemple: afegir un usuari o crear nous registres en una taula d'una base de dades.

A continuació, la figura 1 mostra un exemple visual de cadascun dels atacs presentats.

Figura 1. Representació dels diferents tipus d'atacs que pot patir la comunicació entre l'emissor i el receptor



1.3. Atacants

Un **atacant** és l'agent que realitza un atac. La majoria de vegades, els atacs que pot patir un sistema es produeixen per mans de persones que, amb diversos objectius, intenten accedir o destruir informació confidencial, o directament aconseguir el control del sistema atacat. Conèixer els objectius dels atacants i les seves motivacions resulta, doncs, essencial per detectar les seves accions i prevenir els seus efectes.

Es poden identificar diversos tipus d'atacants:

1) Membres en actiu de la mateixa organització. Tot i que per defecte el personal intern gaudeix de confiança, cal tenir en compte que alguns atacs es poden produir des de dins mateix de l'organització. Sovint molts incidents no són intencionats (tot i que, quan ho són, són molt devastadors), sinó que poden ser accidents provocats pel desconeixement del personal o causats per un error (per exemple, l'esborrament involuntari de dades).

2) Antics membres de l'organització. Una part molt important dels atacs als sistemes són els realitzats per antics treballadors que, abans de deixar l'organització, instal·len tota mena de programari destructiu com, per exemple, virus o bombes lògiques que s'activen en la seva absència. La presència d'aquest tipus de programari no sempre és fàcil de detectar, però almenys sí

que es poden evitar els atacs que l'antic treballador pugui dur a terme des de fora amb el nom d'usuari i la contrasenya de què disposava quan encara era un membre actiu de l'organització. Per tant, com a norma general, cal donar de baixa tots els comptes dels extreballadors com més ràpid millor.

3) Intrusos.[3] Aquestes persones duen a terme normalment atacs passius orientats a obtenir informació confidencial (per exemple, aconseguir l'enunciat d'un examen) o simplement es posen a prova amb la finalitat d'intentar obtenir el control del sistema atacat. A més, si l'atacant és prou hàbil, fins i tot podria esborrar les empremtes de les seves accions en els fitxers que les enregistren (anomenats genèricament **fitxers *log***). Com que aquest tipus d'accions de vegades no produeixen cap rastre, no són fàcilment detectables. Els intrusos solen aprofitar les vulnerabilitats dels sistemes operatius i les aplicacions per aconseguir el control de tot el sistema. Per dur a terme aquestes activitats malicioses, n'hi ha prou amb executar diversos tipus de programari que automatitzen els atacs fins i tot sense que l'intrús necessiti disposar de gaires coneixements tècnics. A més de les eines que hem esmentat, els intrusos disposen d'altres tècniques més senzilles (almenys des del punt de vista tècnic), però igualment efectives.

[3] Són els popularment però incorrectament anomenats *hackers*.

Per exemple, pot resultar molt productiu fer una senzilla recerca de contrasenyes escrites en papers entre la brossa continguda en una paperera, o d'una manera més enginyosa, l'intrús podria suplantar la identitat d'una altra persona per esbrinar la seva contrasenya. Així mateix, un intrús que volgués obtenir una contrasenya en un sistema determinat, podria trucar per telèfon a l'administrador, fer-se passar per una altra persona i demanar la contrasenya amb l'excusa que l'ha oblidat o perdut. En un excés de bona fe, l'administrador podria canviar la contrasenya i lliurar la nova a l'intrús en la mateixa comunicació telefònica.

Les variants d'aquest tipus d'atacs són múltiples i s'inclouen dins el que es denomina *enginyeria social*, és a dir, i com s'ha comentat, la manipulació de les persones per tal que facin determinades accions que en realitat no volen fer, de què parlarem més endavant.

1.4. Contramesures

Una **contramesura** és un mètode per defensar un sistema de possibles atacs. La seguretat considerada en el seu conjunt depèn del disseny i de la implementació de moltes i diverses contramesures, que han de ser complementàries entre si, però també redundants. Un dels principis de disseny de la seguretat és el de la conveniència de disposar les contramesures en una estructura per capes, per tal que, quan un atacant pugui superar una capa de protecció, s'hagi d'enfrontar amb la següent, després amb la que segueix, etc.

Segons la seva naturalesa, establirem els tipus de contramesures següents:

1) Prevenció: són aquelles que protegeixen un sistema durant el seu funcionament.

Per exemple: comunicació i formació, seguretat física, actualització de programari i maquinari, *hardening* de sistemes operatius i aplicacions, auditories, i tallafocs, entre d'altres.

2) Detecció: són les que s'utilitzen per descobrir atacs que s'estan produint contra un sistema.

Per exemple: gestió i anàlisi de *logs*, sistemes de detecció i prevenció d'intrusos, i esquers, per destacar-ne algunes.

3) Recuperació i resposta: aquestes són les que permeten restablir el funcionament d'un sistema un cop produït un incident amb repercussions en l'àmbit de la seguretat.

Per exemple: *backups*, restabliment del servei, plans de recuperació en cas de desastres, investigació d'incidents, execució de plans de resposta preestablerts, notificació a les autoritats en cas d'intrusió o fuita d'informació, entre d'altres.

A les pàgines següents parlarem abastament i amb detall de cada una de les contramesures més importants que es poden implementar en cada àmbit.

2. Control d'accés

Un dels problemes bàsics en el camp de la seguretat és arribar a establir un control efectiu perquè els accessos als recursos sigui realitzat exclusivament per part dels agents autoritzats.

El **control d'accés** és determinar en un sistema que una persona o una part d'un programari té permisos per fer una acció determinada o per accedir a una informació específica.

A les organitzacions hi ha una mena de conflicte d'interessos entre els usuaris d'un sistema, que volen accedir a les aplicacions i a les dades amb les mínimes traves possible per tal de ser el màxim de productius i eficients, i els administradors, que tenen com a missió protegir i salvaguardar els recursos. Entre facilitar un accés totalment lliure i restringir-lo de forma absoluta, hi ha d'haver un punt intermedi en el qual operi el sistema de control d'accés, considerant el principi del mínim privilegi anteriorment esmentat i, alhora, conciliant el rendiment i la seguretat: cada usuari ha de tenir cedits exclusivament els privilegis necessaris per dur a terme la seva feina, ni més ni menys.

A continuació, als subapartats següents descriurem les diferents parts que habitualment constitueixen un sistema de control d'accés: identificació, autenticació, autorització i auditoria.

2.1. Identificació

La primera fase del control d'accés és la identificació: l'agent, que vol fer ús d'un recurs, ha de facilitar la seva identitat i el sistema recollir la manifestació de qui diu ser. Un cop s'ha obtingut aquesta declaració, es procedeix a decidir si se li dona credibilitat o no, mitjançant l'etapa següent: l'autenticació.

2.2. Autenticació

L'**autenticació** és el procés de verificació de la identitat d'un agent: és a dir, assegurar-se que algú és efectivament qui declara ser.

L'estratègia més habitual per dur a terme aquesta tasca és demanar-li a aquest agent quelcom que sabem que només aquell algú ens pot proveir i que, per tant, pot constituir una prova de la seva identitat: alguna dada que només ell conegui (per exemple, una contrasenya), algun objecte que només ell posseeixi (per exemple, una clau que obre un pany) o algun tret físic característic

distintiu que es pugui reconèixer (per exemple, les empremtes digitals). Així doncs, els mecanismes d'autenticació es podran classificar en tres grups, segons el mètode utilitzat per comprovar la identitat (de què en direm *factor d'autenticació*):

1) Factor de coneixement (quelcom que l'usuari sap)

Implica una dada, anomenada *secret*, que només coneix l'usuari (i l'agent que autentica). El principal mecanisme dins d'aquests tipus d'autenticació el representen els sistemes basats en **contrasenyes**. És un dels mètodes més habituals i fàcils d'implementar, però també és un dels més vulnerables, ja que encara que la contrasenya hauria de ser personal i intransferible, és molt fàcil que acabi en poder de persones no autoritzades. D'altra banda, encara que les contrasenyes s'emmagatzemin xifrades, és possible eludir l'autenticació amb diverses tècniques, com es veurà més endavant. També la seva efectivitat depèn del fet que els usuaris escullin contrasenyes fortes: que siguin difícils d'esbrinar i que puguin resistir atacs de diccionari i de força bruta. S'anomena *fortalesa* d'una contrasenya a la mesura de la dificultat que troba un atacant per obtenir-la sense tenir-hi accés previ, i s'expressa habitualment en el nombre d'intents necessaris per aconseguir-la. Com més llarga, complexa i aleatòria sigui una contrasenya, major serà la seva fortalesa.

Tot i que l'assignació de les contrasenyes es basa en el sentit comú, no és sobrer recordar els aspectes següents per fer-les més efectives:

a) Utilitzar contrasenyes fortes:

- Evitar utilitzar paraules que es puguin trobar en un diccionari.
- Evitar utilitzar dades que poden ser conegudes per altres persones (per exemple: nom i cognom de l'usuari, números de documents d'identitat, número de telèfon o dates significatives).
- Fer servir contrasenyes llargues.
- Utilitzar majúscules, minúscules, números i caràcters especials.
- No utilitzar seqüències de teclat del tipus «qwerty».

b) Utilitzar una cadena alfanumèrica que, tenint totes les característiques de les contrasenyes fortes, sigui fàcil de recordar (fent servir mnemotècnics, per exemple), o utilitzar un programa gestor de contrasenyes.

c) Canviar la contrasenya periòdicament.

d) Assegurar-se de canviar les contrasenyes per defecte.

e) No repetir la mateixa contrasenya en comptes diferents.

f) Evitar la reutilització de contrasenyes antigues.

g) Evitar introduir la contrasenya allà on puguem ser observats i no comunicar-la a ningú.

h) No llençar documents amb contrasenyes a la paperera.

2) Factor de propietat (quelcom que l'usuari té)

Es basa en un objecte físic que només pot posseir l'usuari. En aquest grup tindríem els documents d'identitat, les claus que obren un pany, les targetes de crèdit, el terminal del telèfon mòbil, les targetes intel·ligents (*smartcards*), els *tokens* de seguretat, etc.

3) Factor d'existència o inherència (quelcom que l'usuari és)

Comporta un tret físic consubstancial de l'usuari, com els que utilitzen els sistemes biomètrics. Aquests es basen en les característiques físiques de l'usuari que s'ha d'autenticar o en patrons particulars que puguin ser reconeguts (com pot ser, per exemple, la signatura). Entre les diferents característiques que es poden utilitzar per reconèixer un usuari mitjançant les mesures biomètriques, destacarem les següents: les empremtes dactilars, les faccions facials, la morfologia de la retina i l'iris dels ulls, la veu, la geometria de la mà o el traçat dels vasos sanguinis, entre d'altres.

Els desavantatges que presenten aquests sistemes són els següents: solen ser comparativament més cars, poden generar lectures falses (un fals positiu seria acceptar una lectura quan hauria de ser rebutjada i un fals negatiu seria l'inrevés), els usuaris poden mostrar-se reticents al fet que el sistema gestioni les seves dades biomètriques i les lectures poden variar en funció de les condicions ambientals en què es produeixin o de l'evolució de l'estat físic de l'usuari (per exemple, l'edat i el pes).

Encara que els desavantatges esmentats són els més habituals, altres factors d'autenticació menys comuns, però també viables i emprats, especialment quan s'utilitzen conjuntament amb altres factors, són:

- **Factor d'ubicació** (a on està l'usuari): implica que l'usuari està físicament en una localització específica en un moment determinat. Un exemple d'aquest factor el tenim quan un servei web o una companyia financera registra que la ubicació geogràfica de l'usuari en el moment d'accedir o de fer alguna operació no és l'habitual i demana més informació per confirmar la identitat.

- **Factor de comportament** (quelcom que l'usuari fa): consisteix a identificar un comportament característic, com ara els patrons d'escriptura cal·ligràfica o d'utilització d'un teclat d'ordinador.

De tots aquests factors d'autenticació de què hem parlat, se'n pot fer servir només un o, si s'estima que això pugui ser insuficient, utilitzar-ne dos o més a la vegada. Parlaríem, en aquest últim cas, d'autenticació de múltiples factors.

2.3. Autorització

Quan l'agent està autenticat és quan se li podrà atorgar efectivament els privilegis necessaris perquè pugui accedir als recursos que es determinin. En aquesta fase, anomenada *autorització*, s'articula a quina funcionalitat i a quines dades té accés l'usuari, en funció de la seva identitat verificada.

2.4. Auditoria

Aquesta fase, l'auditoria, és transversal a tot el procés i consisteix a mantenir en un registre el seguiment de l'activitat dels agents mentre interactuen amb el sistema, per tal de comprovar que se n'està fent un ús lícit, analitzar possibles incidents i disposar d'evidències en cas de produir-se una violació de la seguretat.

2.5. Models

Hi ha diferents models de sistemes de control d'accés quan parlem de la seva implementació pràctica, d'entre els quals destacarem els següents:

- **Control d'accés discrecional (*discretionary access control*, DAC)**: es restringeix l'accés als objectes (fitxers, directoris, dispositius) en funció de la identitat dels agents (usuaris, programes) i dels grups a què pertanyen. Habitualment els objectes tenen un posseïdor i és aquest qui gestiona directament els privilegis d'accés. Molts sistemes de fitxers de Linux implementen aquest model.

- **Control d'accés obligatori (*mandatory access control*, MAC)**: amb aquest model, el sistema assigna nivells de seguretat a tots els agents i a tots els objectes. Així, un usuari podrà accedir a un objecte determinat només en el cas que els nivells de seguretat de l'un i l'altre ho permetin.

- **Control d'accés basat en rol (*role-based access control*, RBAC)**: es decideix donar accés a un objecte en funció del rol de l'agent.

  Per exemple: els usuaris que pertanyen al grup de gestió acadèmica d'una universitat seran els únics que podran accedir a les dades de matrícula, expedients i beques, i els de gestió econòmica a les de comandes i facturació.

- **Llistes de control d'accés (*access control lists*, ACL)**: són llistes que recullen els usuaris i grups d'un sistema conjuntament amb els privilegis atorgats a cada un.

Per exemple: un directori anomenat «Nòmines» podria tenir una ACL com aquesta: «(Joana: lectura, escriptura; Maria: lectura)», que habilitaria l'accés als usuaris «Joana» i «Maria» a aquest directori amb els privilegis especificats.

3. Amenaces i atacs

Les amenaces sobre la seguretat provenen de **múltiples fonts de vulnerabilitats** que possibiliten diversos atacs. Les més habituals: l'execució de codi arbitrari aprofitant els errors del programari, la utilització de diferents tipus de codi maliciós, els atacs de denegació de servei, els atacs d'intermediari, els atacs de canal lateral i l'ús de l'enginyeria social, entre d'altres.

En aquest apartat parlarem de les vulnerabilitats, les amenaces i les contramesures que es poden adoptar per defensar-se dels atacs.

3.1. Errors i vulnerabilitats del programari

El desenvolupament de programari és una disciplina altament complexa i, a causa d'això i de la manca de comprovacions suficients, és habitual que els programes continguin errors de diversos tipus (anomenats habitualment *bugs*) que fan que, quan s'executen, presentin un comportament diferent a aquell pel qual estaven dissenyats.

Un ***exploit*** és una peça de codi o un conjunt de dades que, aprofitant un *bug* específic d'un maquinari o un programari determinats, provoca un comportament inesperat o no previst prèviament i que pot ser aprofitat per part d'un atacant, com ara: executar codi arbitrari en un ordinador (*arbitrary code execution*), aconseguir més privilegis d'usuari dels que pertocarien lícitament i així fer accions no autoritzades (escalada de privilegis o *privilege escalation*), i realitzar un atac de denegació de servei, entre d'altres.

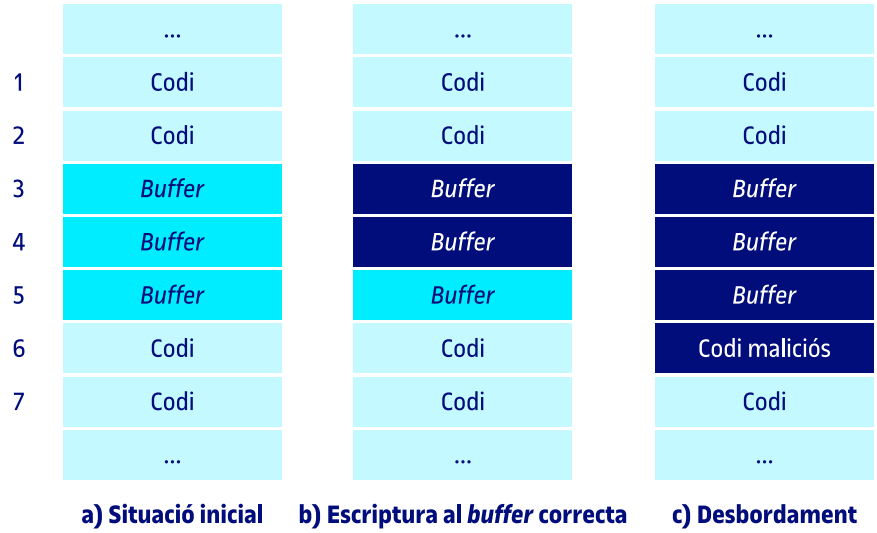
Un ***exploit* remot** funciona contra un sistema vulnerable per mitjà de la xarxa, sense que l'atacant tingui accés local, i un ***exploit* local** requereix accés al sistema vulnerable per ser efectiu.

Alguns dels problemes més habituals que ens podem trobar amb el programari, que tenen afectació en la seguretat, són els que s'assenyalen a continuació:

1) Desbordament del *buffer* (*buffer overflow*): és un tipus de vulnerabilitat molt comuna en programari desenvolupat en els llenguatges C i C++. Es produeix quan un programa reserva un espai de memòria determinat per guardar-hi dades (anomenat *buffer*), però quan hi escriu, rabassa els límits d'aquest espai i sobreescriu parts contigües de la memòria. Si el fragment sobreescrit de manera anòmala es correspon amb una part del codi executable del programa, aquest es pot modificar per incloure-hi codi maliciós.

A la figura 2 es mostra, a mode d'exemple, el contingut de la memòria d'un programa determinat en temps d'execució. Cada rectangle es correspon amb una posició de memòria d'un byte. A les posicions 3, 4 i 5 hi ha reservat un espai de memòria per a un *buffer* de 3 bytes (color blau intens). A la resta de la memòria hi ha codi executable (color blau clar). Una escriptura correcta al *buffer* seria aquella en la qual s'escriuen 3 bytes o menys. Si s'escriuen més de 3 bytes, es produirà un desbordament del *buffer*. Un atacant pot sobreescriure el *buffer* amb un quart byte que contingui codi maliciós (tal com es mostra a la part c de la figura).

Figura 2. Esquema del funcionament del desbordament d'un *buffer*



2) Desbordament de sencer (*integer overflow/underflow*): succeeix quan el resultat d'una operació aritmètica en un programa no es pot representar, per ser massa alt o massa baix, amb el nombre de dígits disponibles pel programa. Aquests resultats incorrectes no estan prevists per la lògica del codi i poden comportar diversos mals funcionaments susceptibles de ser explotats de diverses maneres, com per exemple, per dur a terme atacs de denegació de servei.

3) Situació de competició (*race condition*): són els problemes derivats de quan el programa no duu a terme les tasques que ha de fer en l'ordre que el programador l'havia dissenyat.

4) Fuites (*leakage*) de recursos: aquesta situació anòmala es produeix quan un programa va reservant recursos de la màquina (per exemple, memòria, ports de xarxa o fitxers oberts) però no els allibera.

Com es pot afrontar el problema dels *bugs* del programari? Quines mesures es poden adoptar, sabent que existeixen i previsiblement continuaran existint, per tractar aquests errors i disminuir al màxim possible la seva afectació (i, més específicament, en tot allò que pugui afectar la seguretat)?

L'estratègia més immediata és intentar localitzar els *bugs* de manera sistemàtica i reescriure els programes per solucionar-los, fent auditories de codi amb eines automatitzades. Els programadors poden revisar manualment el programari a la recerca d'errors, però utilitzant eines específiques es pot automatitzar el procés i, en alguns casos, fer-ho de manera molt més eficient. Quan l'anàlisi del programari es duu a terme sense executar-lo, només tractant el codi font, parlarem d'**anàlisi estàtica**. I quan s'examina el codi en temps d'execució, ens hi referirem com a **anàlisi dinàmica**.

Un tipus d'anàlisi que val la pena destacar aquí, per la seva importància, és l'anomenada **enginyeria inversa**, que és el procés de descobrir els principis de funcionament d'un sistema per mitjà de la seva desconstrucció. En seguretat, concretament, consisteix a produir codi llegible a partir d'un fitxer binari per tal de, per exemple, trobar vulnerabilitats i *exploits* o estudiar el comportament del codi maliciós.

Una altra estratègia per afrontar la presència de *bugs* és reescriure el programari en llenguatges de programació segurs quant al tipus (llenguatges *type-safe*) com poden ser Java, Go, Rust i altres, que eviten en temps de compilació els anomenats *errors de tipus*.

Errors de tipus

Aquests errors es produeixen quan, per exemple, s'intenta assignar un valor numèric a una variable de tipus cadena.

3.2. Codi maliciós

El **codi maliciós** (*malware* en anglès) és un programari dissenyat específicament amb l'objectiu de dur a terme algun tipus d'acció nociva en un sistema.

El mètode que el codi maliciós utilitza per propagar-se (o, dit d'una altra manera, per infectar altres programes o sistemes) s'anomena *vector*.

Les accions nocives que duu a terme el codi maliciós (a banda de la pròpia replicació) s'anomenen *payload* i poden ser diverses, com ara, entre d'altres:

- Robar, impedir-ne l'accés o esborrar dades.
- Fer un ús no autoritzat dels recursos computacionals de què disposa l'usuari, com la capacitat de còmput i l'amplada de banda (per exemple, minar criptomonedes des de l'ordinador de la víctima sense el seu coneixement).
- Espiar l'activitat de l'usuari.

El codi maliciós es pot classificar en els tipus següents:

1) Virus: té la capacitat d'inserir una còpia de si mateix en un altre codi executable amb la finalitat de propagar-se. Quan aquest altre programa infectat, anomenat *hoste*, s'executa, el codi del virus també ho fa. El nom prové del seu anàleg en la naturalesa: en biologia, un virus és un organisme que infecta diverses formes de vida i que només es pot desenvolupar i reproduir dins d'una cèl·lula infectada. Hi ha molts tipus de virus: de sector d'inici, de macro, multiplataforma, multiprocés, de compressió (com a forma d'ocultació), interpretats, sobreescrits (destrueixen el fitxer) o afegits (el conserven), i polimòrfics, entre d'altres.

2) Cucs: són capaços de replicar-se propagant-se per mitjà de la xarxa a altres ordinadors, normalment aprofitant-se de vulnerabilitats remotes específiques que detecten i que poden explotar per autoexecutar-se. A diferència dels virus, no deixen una còpia de si mateixos en el codi d'altres fitxers executables.

3) Troians: es presenten davant l'usuari com un programa benigne per tal que aquest l'executi, realitzant finalment accions diferents a les que se suposa que havia de fer (potencialment nocives). El nom prové de la història clàssica del cavall de Troia, que explica com els grecs van aconseguir conquerir la ciutat de Troia, que estaven assetjant, mitjançant un engany. Aquest va consistir a deixar un cavall de fusta a les portes de la ciutat i fer veure que desistien del setge i que marxaven amb els seus vaixells. Llavors els habitants van apoderar-se del cavall com a trofeu de guerra i el van introduir a la ciutat. Dintre del cavall hi havia soldats grecs que van aprofitar la nit per sortir, obrir les portes de la ciutat i deixar que entrés la resta de l'exèrcit, que la va prendre.

4) Bombes lògiques: es mantenen en estat inert fins que es produeix una certa condició que les activen (com ara una data o una combinació de tecles).

5) *Rootkits*: són un conjunt d'eines que serveixen per evitar el fet que un sistema que hagi estat compromès sigui detectat.

6) *Backdoors*: són uns mecanismes ocults, instal·lats per un atacant, que permeten accedir a un sistema evitant els mecanismes normals d'autenticació.

7) *Ransomware*: impedeix l'accés a les dades d'un sistema (normalment xifrant-les), i requereix el pagament d'un rescat perquè es pugui retirar el bloqueig. L'afectació d'aquests atacs ha augmentat a causa del desenvolupament i la popularització de les criptomonedes, ja que les seves transaccions són extremadament difícils de rastrejar.

8) *Adware*: mostra publicitat diversa sense el consentiment de l'usuari.

9) ***Spyware***: recopila informació de l'usuari sense el seu coneixement, intentant mantenir-se ocult (prenent el control de la càmera, enregistrant l'activitat del teclat i la pantalla, interceptant comunicacions, robant credencials o enregistrant l'activitat de navegació per pàgines web).

10) ***Cryptojacking***: utilitza, sense el coneixement de l'usuari, els recursos d'una màquina per minar criptomonedes.

3.3. Denegació de servei

S'anomenen **atacs de denegació de servei** (*denial of service*, DoS) a l'acció iniciada per un atacant amb què es pretén inutilitzar funcionalment el maquinari o el programari, de manera que els seus recursos no siguin accessibles (en local o en remot).

Una manera típica d'aconseguir efectuar aquest atac consisteix a sobrecarregar el sistema objectiu amb peticions espúries de tal manera que les peticions lícites que arribin no puguin ser ateses.

Quan l'atac es produeix per la xarxa, una manera de fer-lo més efectiu és utilitzar moltes fonts per originar el trànsit que provoqui la sobrecàrrega, en comptes de només una. Primer perquè, òbviament, a més cabdal, més possibilitat hi haurà de poder saturar els recursos; i segon, perquè serà més difícil per part de la víctima defensar-se intentant bloquejar els atacs, ja que aquests provenen de diverses localitzacions. Aquests tipus d'atacs s'anomenen **atacs de denegació de servei distribuïts** (DDoS). De manera habitual, s'utilitzen ordinadors compromesos,[4] que estan controlats per l'atacant normalment sense el coneixement de l'usuari i que formen part d'una xarxa anomenada *botnet*. L'atacant pot enviar comandes que seran executades per tots els sistemes de la *botnet*, com per exemple, enviar contínuament peticions a un servidor concret amb la finalitat de saturar-lo. Darrerament hi ha un vector que s'està fent servir cada vegada més per dur a terme atacs de DDoS, com són els dispositius que formen part de l'anomenada *internet de les coses* (domòtica, alarmes, monitors personals de salut, per posar alguns exemples), ja que solen tenir deficiències importants de seguretat perquè no reben actualitzacions (i, per tant, són relativament fàcils de prendre'n el control) i estan sempre connectats.

[4] Anomenats *zombies*.

Un exemple d'atac DoS és el d'**inundació SYN** (SYN *flooding*). Consisteix en l'enviament, per part dels sistemes atacants, d'un gran nombre de sol·licituds d'establiment de connexions TCP per segon. El sistema atacat respon correctament, però no obté resposta, acumula peticions obertes fins a esgotar els seus recursos i col·lapsar-se, deixant llavors d'atendre les sol·licituds legítimes de connexió.

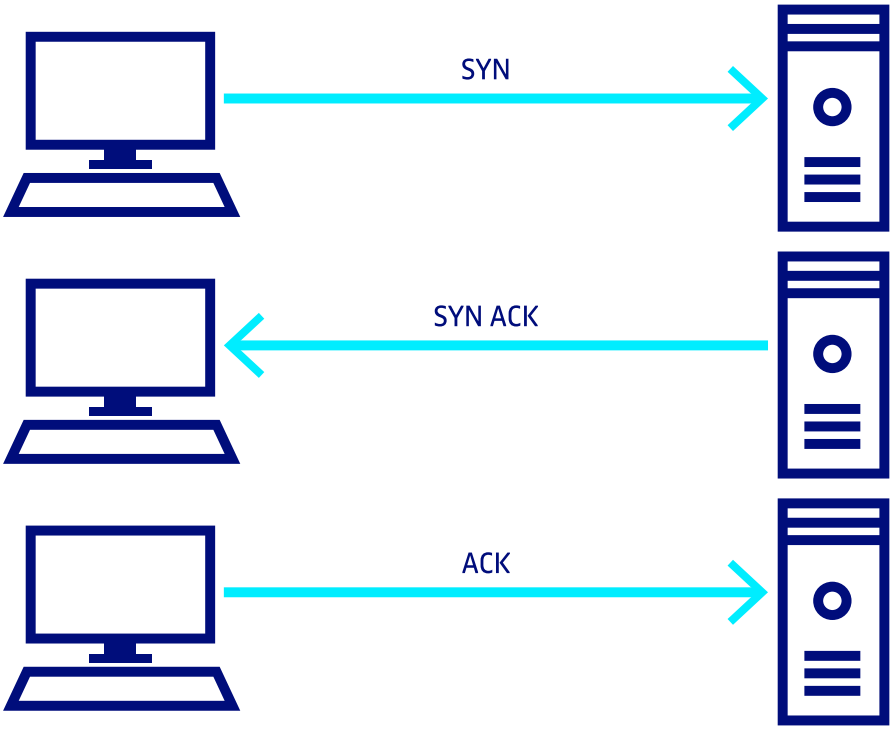
La figura 3 mostra el comportament lícit del protocol TCP per a l'establiment d'una connexió nova:

1) El client envia al servidor una sol·licitud de nova connexió enviant un missatge (els missatges TCP s'anomenen *segments*) amb el *flag* de sincronització (SYN) activat.

2) El servidor respon al client amb un altre segment amb els *flags* de sincronització (SYN) i de justificant de recepció o *acknowledgement* (ACK) activats.

3) El client respon enviant al servidor un segment amb el *flag* ACK activat. Queda establerta en aquest moment la connexió TCP, llesta per a l'intercanvi de dades.

Figura 3. Protocol d'establiment de sessió TCP en tres passos (*handshake*)



El servidor manté en cua d'espera tots els paquets amb SYN que va rebent, fins que són cancel·lats per l'enviament del corresponent ACK per part del client (o expira un temporitzador que regula el temps d'espera). L'atac per inundació SYN es produeix quan els paquets enviats per l'emissor contenen adreces IP errònies i, en conseqüència, el servidor mai no podrà rebre el paquet ACK que alliberaria la cua de recepció. Així, quan aquesta s'omple, les noves sol·licituds legítimes de connexió no es podran servir.

Per intentar defensar-se dels diferents tipus d'atacs DoS i DDoS, l'estratègia consisteix a **monitoritzar la xarxa** per establir una línia de base, que marcarà les condicions del trànsit que es consideraran normals. A partir d'aquesta anàlisi contínua del trànsit, es detectaran les anomalies mitjançant la identificació de diferents patrons, amb l'objectiu d'identificar el trànsit que és generat

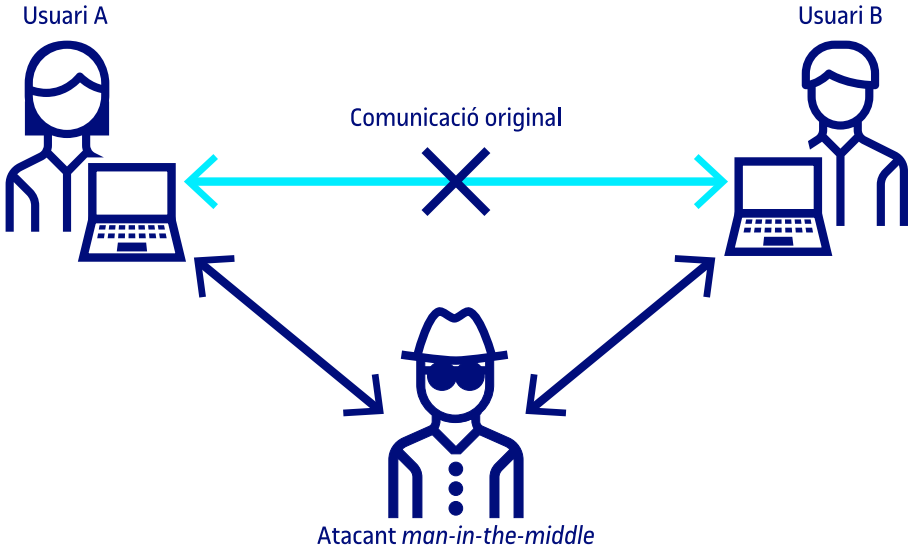
pels humans i el que pugui provenir de bots de sistemes compromesos. Quan s'ha realitzat la detecció de l'atac, es procedeix a adoptar una contramesura, que pot ser el filtratge dels paquets maliciosos o la limitació progressiva de la resposta a un client en funció de les peticions que realitza (*rate limiting*).

3.4. Atac d'intermediari

Un **atac d'intermediari** (*man-in-the-middle attack*) consisteix a interposar-se en una comunicació entre un emissor i un receptor sense que aquests se n'adonin.

La figura 4 descriu un exemple d'aquest tipus d'atac. Secretament, l'atacant intercepta el missatge provinent de l'emissor, l'observa, opcionalment el modifica, i l'envia al receptor. L'emissor i el receptor creuen en tot moment que es comuniquen directament.

Figura 4. Esquema d'atac d'intermediari (*man-in-the-middle*)



Un tipus més específic d'atac d'intermediari és l'anomenat ***atac de replay*** (*replay attack*), en què un missatge vàlid entre l'emissor i el receptor és interceptat per l'atacant i aquest el retransmet en algun altre moment.

Un exemple senzill d'aquesta situació seria quan es produeix la captura del senyal que emet un comandament a distància que obre una porta quan l'està utilitzant un usuari de manera lícita. Llavors l'atacant podrà reemetre el senyal capturat per obrir la porta quan li convingui.

3.5. Atac de canal lateral

Els **atacs de canal lateral** (*side-channel attack*) intenten aprofitar les vulnerabilitats d'un sistema analitzant mesures de paràmetres físics referents a la seva implementació a la pràctica.

Per exemple, obtindrem informació potencialment útil d'un sistema per atacar-lo, aconseguint mètriques mentre realitza una determinada operació: el corrent elèctric consumit, el temps de còmput invertit, les emissions electromagnètiques que es produeixen, l'escalfor produïda, els sons emesos o la llum provinent dels indicadors d'activitat (com, per exemple, els que tenen els ordinadors per a l'accés al disc). Aquest tipus d'atacs són especialment efectius per utilitzar-se contra determinades implementacions pràctiques d'un criptosistema que pot ser segur en la teoria.

Un exemple d'atac lateral seria enregistrar i analitzar els sons que produeixen les tecles quan un usuari les pressiona en un teclat, per tal d'inferir quina tecla correspon a cada so i, així, esbrinar una contrasenya o un pin que es pugui estar introduint.

3.6. Enginyeria social

Ja s'ha introduït breument el concepte d'enginyeria social i s'ha parlat de manera introductòria de la seva importància creixent quan es considera la seguretat dels sistemes.

Recordem que una possible definició d'**enginyeria social** seria la següent: el conjunt de tècniques per tal d'influir en una persona amb l'objectiu que faci una determinada acció.

El principi en què se sustenta l'enginyeria social és el d'intentar aprofitar-se dels biaixos cognitius (prejudicis, distorsions de la percepció, interpretacions incorrectes) que condicionen la pressa de decisions en els éssers humans. Usualment els objectius concrets més habituals de l'atacant que fa servir aquestes tècniques són, entre d'altres: obtenir informació diversa, fer que l'usuari víctima executi algun tipus de codi maliciós i accedir físicament a una localització restringida.

De què serveix mantenir els dispositius i els mecanismes digitals de protecció més avançats en l'àmbit tecnològic d'una organització, si un atacant aconsegueix accedir físicament als servidors després de passar pel control de seguretat vestit d'empleat de manteniment amb una targeta d'identitat falsa, o si obté les credencials d'una aplicació corporativa d'un usuari important trucant-lo per telèfon fent-se passar per un administrador?

A continuació, s'introdueixen breument alguns tipus específics d'atacs d'enginyeria social:

1) *Phishing*: és un dels atacs més coneguts. Consisteix a intentar obtenir informació sensible de manera il·lícita o provocar una determinada acció de la víctima per mitjà d'una comunicació electrònica que simula provenir d'una part que mereix confiança (el correu electrònic i la missatgeria instantània són els mitjans més habituals). Quan l'atac es dissenya per dirigir-lo específicament a una persona concreta, i no a qualsevol víctima genèrica, es diu atac de llança (*spear attack*). Quan l'atacant empra algun mitjà en el qual intervé la veu (per exemple, el telèfon), aquest tipus d'atac s'anomena *vishing*.

2) Suplantació de la identitat: consisteix a fer-se passar per algú altre per tal de poder dur a teme alguna acció no autoritzada.

3) *Dumpster diving/Trashing*: és l'activitat que consisteix a buscar a les escombraries informació impresa, o guardada en qualsevol altre mitjà, que no hagi estat degudament eliminada (credencials, correus electrònics, llistes d'empleats, números de telèfon, registres de trucades, fotos, clients, proveïdors, secrets empresarials, informació dels sistemes, etc.).

4) *Baiting*: és la versió d'un troià digital al món real. Consisteix a deixar a l'abast de la persona a qui es dirigeix l'atac un dispositiu d'emmagatzematge USB o un DVD que contingui codi maliciós, intentant treure profit de la curiositat que se li pugui despertar a la víctima.

5) *Doxing*:[5] és la pràctica de cercar informació privada i potencialment sensible d'una persona o una organització per fer-la pública o utilitzar-la per a altres fins maliciosos. Els recursos utilitzats en aquest tipus d'atacs són les xarxes socials, els cercadors, les bases de dades públiques i altres mètodes d'enginyeria social.

(5)De *dox*, abreviatura de la paraula *documents*.

Quines mesures es poden prendre per intentar defensar-se d'aquests tipus d'atacs d'enginyeria social? A continuació, en presentarem algunes:

- Prioritzar la formació contínua de tots els membres de l'organització en les pràctiques bàsiques de seguretat, establir protocols per a la correcta gestió de la informació i conscienciar la direcció de l'organització. Aquesta, sense cap mena de dubte, és la més important.

- Mantenir la seguretat física operativa i incloure-la en les auditories periòdiques.

- Evitar que es pugui recuperar il·lícitament la informació que no estigui xifrada dels mitjans que han de ser retirats del funcionament actiu o que puguin ser extraviats (dispositius mòbils, còpies de seguretat externalitzades o documents en paper). Per exemple, quan es tracta de mitjans digitals com ara cintes o discs que s'han de llençar, ens hem d'assegurar que

les dades que hi hagi contingudes siguin convenientment eliminades, utilitzant diferents tècniques, essent la més habitual la seva reescriptura perquè siguin impossibles (o el més difícil possible) de recuperar per part de persones no autoritzades. D'aquest mètode d'esborrament es diu *wiping* o *shredding*, utilitzant els termes en anglès. En funció de la seguretat que es vulgui tenir del procés d'eliminació, es poden sobreescriure diferents patrons, diverses i repetides vegades (com més vegades, més difícil seria una hipotètica recuperació il·lícita). Un altre mètode és el *degaussing*, que consisteix a utilitzar un aparell per eliminar el camp magnètic de la cinta o el disc. I també es pot emprar el procediment més contundent, que seria directament la destrucció física del mitjà.

- Evitar posar noms als servidors que facilitin informació a un possible atacant sobre el mateix sistema, el programari que té instal·lat, la funció que realitza o el servei que dona.

  Per exemple, si algú descobreix, mitjançant consultes DNS, que hi ha dues màquines anomenades «sap1» i «sap2» en una organització, podrà pensar que, amb molta probabilitat, aquestes tinguin instal·lat algun programari empresarial de la companyia SAP i que, per tant, contindran informació important i la seva funció pot ser vital per al desenvolupament de l'activitat (en canvi, si les màquines s'anomenen «prc-b01» i «prc-b02» o «hansel» i «gretel», per exemple, no es dona cap pista que pugui facilitar un atac).

4. Seguretat física

En aquest apartat veurem algunes mesures de protecció física que es poden fer servir per evitar els accessos no autoritzats als sistemes. Una organització pot invertir molts diners en programari que detecti i eviti els accessos lògics il·lícits als seus sistemes, però tota aquesta inversió servirà de poc si els recursos físics estan a l'abast de possibles atacants. A més a més, en els últims anys, la qüestió de la seguretat física ha guanyat en complexitat amb la proliferació dels dispositius mòbils (telèfons, tauletes, dispositius d'emmagatzematge i altres).

Una persona no autoritzada que aconsegueixi entrar al centre de processament de dades o CPD, que és un espai físic on hi ha ubicats els sistemes, podrà manipular-los directament i causar enormes danys. D'aquest tipus d'atacs en direm *atacs d'accés directe*. Si en un servidor els sistemes de fitxers no estan xifrats, pot iniciar el sistema des d'un mitjà extern (com un USB *bootable* amb una distribució de Linux) i fer els canvis que vulgui sense tenir les credencials d'administrador. Si els sistemes de fitxers estan xifrats, també pot instal·lar components físics per observar i enregistrar il·lícitament l'activitat, com per exemple amb un *keylogger* USB, col·locar altres sistemes d'espionatge (micròfons o càmeres ocultes), o manipular els elements de seguretat física (alarmes, videovigilància i sensors). I és clar, en qualsevol situació, les pèrdues també es poden estendre a les que pugui causar pel robatori i destrucció de l'equipament, o l'eliminació de dades.

S'ha de considerar, doncs, que la seguretat física és la base fonamental sobre la qual descansa tot el disseny de seguretat lògic. Algunes de les **mesures de prevenció** en aquest àmbit són les següents:

- Mantenir tot l'equipament en una zona d'**accés físic restringit**, amb controls d'accessos i monitorització. L'accés físic als sistemes ha de ser controlat mitjançant un o més sistemes d'autenticació (panys, *smartcards*, sistemes biomètrics, codis d'accés, entre d'altres). També és important assegurar-se que els accessos físics al CPD mitjançant el sostre, els terres tècnics, els conductes de l'aire, o trencant parets o finestres de vidre, no poden ser utilitzats. També és convenient instal·lar un sistema de videovigilància, sensors de presència i alarmes d'intrusió.

- Mantenir un **inventari** de tots els elements de l'equipament i fer revisions periòdiques.

- Protegir el **cablejat** de la xarxa i fer-ne inspeccions periòdiques.

- Triar una **topologia** de xarxa adequada a les nostres necessitats de seguretat.

- Garantir la **seguretat** del maquinari de la xarxa (encaminadors, concentradors, tallafocs, connectors, etc.).

- Utilitzar **contrasenyes** en els BIOS i els protectors de pantalla.

Hi ha **amenaces** no causades per l'acció humana, que també s'han de tenir en compte a l'hora de dissenyar la protecció dels elements físics de la nostra infraestructura, com ara:

- **Incendis**: s'han d'instal·lar sistemes de detecció de fum i d'extinció d'incendis.

- **Condicions ambientals**: mantenir la temperatura i la humitat adequades és imprescindible perquè l'equipament del CPD es mantingui operatiu durant el període de vida útil previst. Convé instal·lar sensors ambientals i assegurar-se que són funcionals, que generen les alarmes de manera oportuna i que aquestes són ateses pel personal responsable seguint els procediments establerts.

- **Pols**: l'acumulació de pols en els ventiladors dels equips provoquen un augment de la temperatura d'operació que pot tenir com a conseqüència mals funcionaments i avaries. Els components òptics dels elements de la xarxa són especialment sensibles a la pols, podent-se generar errors de comunicació.

- **Problemes amb el corrent elèctric**: instal·lar un sistema d'alimentació ininterrompuda, filtres per evitar pics de tensió i mecanismes de protecció contra els efectes de les tempestes elèctriques.

- **Inundacions**: instal·lar sensors que detectin l'acumulació d'aigua a les instal·lacions.

- **Huracans i terratrèmols**: els armaris (*racks*) on hi ha instal·lats els servidors, equipaments de xarxa i altres sistemes, han d'estar degudament instal·lats i fixats al terra, observant totes les mesures de seguretat.

És important disposar d'un pla per restaurar l'activitat dels sistemes quan una d'aquestes amenaces es fa efectiva, de tal manera que el temps de recuperació i les pèrdues produïdes estiguin dins d'uns barems acceptats establerts prèviament. Un exemple de les mesures que es poden adoptar en aquest sentit és mantenir els sistemes redundats amb les dades replicades en una altra ubicació física o al núvol.

5. Seguretat del servidor

El següent nivell, un cop revisada la seguretat física, és el que concerneix al servidor. En aquest apartat comentarem les qüestions relatives a la seguretat del servidor, des del moment en què es dissenya el seu desplegament, i inicialment s'instal·len i configuren el sistema operatiu i les aplicacions, fins quan es passa a producció i s'han d'anar executant el conjunt de tasques administratives rutinàries necessàries per mantenir el correcte funcionament dels serveis.

Parlarem del *hardening* del sistema operatiu, de les tasques de l'administrador en relació amb la seguretat, dels sistemes de fitxers, de les contrasenyes i de les intrusions.

5.1. *Hardening*

Hardening (traduït com a enduriment) és el procés de reduir en un sistema la seva exposició a possibles atacs (anomenada *superfície d'atac*) amb l'objectiu de fer-lo més segur. Com menys elements d'interacció permeti un sistema (com menys exposat estigui), menys possibilitats hi haurà que es pugui utilitzar algun error per violar la seva seguretat.

En el cas dels sistemes operatius, el *hardening* consisteix, més específicament, a dur a terme una sèrie de mesures en el moment de la instal·lació, entre les quals podem destacar:

1) Crear diferents particions, volums i sistemes de fitxers per separar les dades i així limitar l'afectació en cas de produir-se algun accident o error.

2) Xifrar els sistemes de fitxers que calgui per tal de garantir la confidencialitat.

3) Desinstal·lar o inhabilitar el programari i el maquinari que sabem que no s'utilitzarà.

4) Canviar el nom dels usuaris administradors, si el sistema ho permet.

5) Restringir les vies amb les quals els usuaris administradors poden accedir al sistema (per exemple, impossibilitant que puguin establir una sessió directament des de la xarxa).

6) Establir una política de fortalesa i expiració de contrasenyes, i canviar les contrasenyes per defecte.

7) Eliminar els comptes d'usuari que no s'hauran de fer servir.

8) Desactivar les tasques programades i els serveis innecessaris.

9) Configurar el programari instal·lat de la manera més restrictiva possible i habilitar l'encriptació a tots els serveis de la xarxa que ho permetin.

10) Comprovar els ports de xarxa oberts a l'escolta i restringir les interfícies associades.

11) Instal·lar, configurar i habilitar un tallafoc i un sistema de detecció i prevenció d'intrusions de *host*.

Vegeu també

Dels tallafocs i els sistemes de detecció i prevenció d'intrusions de *host*, en parlem a l'apartat «Seguretat de la xarxa».

12) Establir quotes d'ús per als recursos per evitar atacs de denegació de servei.

13) Instal·lar i configurar el sistema de còpies de seguretat i verificar el seu funcionament.

14) Documentar la instal·lació i salvaguardar els fitxers de configuració.

5.2. Tasques de l'administrador

Un cop el servidor està funcionant, l'administrador és responsable de dur a terme una sèrie de tasques per mantenir la seguretat. Consisteixen en un exercici continu de pràctiques periòdiques i sistemàtiques:

1) Mantenir un o diversos entorns aïllats i dedicats de desenvolupant o de test en els quals provar els canvis que s'hagin de realitzar posteriorment en producció.

2) Instal·lar actualitzacions i *patches* del sistema operatiu i les aplicacions. Convé estar al corrent de les vulnerabilitats que es van publicant per prendre ràpidament mesures adequades.

3) Mantenir el programari antivirus i de detecció i prevenció d'intrusos contínuament actualitzats.

4) Monitoritzar el rendiment dels sistemes per detectar possibles patrons sospitosos (com ara un ús intensiu i anòmal de CPU o memòria, trànsit de xarxa inusual, o sessions iniciades a hores poc habituals).

5) Revisar i registrar les configuracions i documentar els canvis que es vagin produint.

6) Fer auditories locals i tests d'intrusió utilitzant escàners de vulnerabilitats.

7) Comprovar la seguretat dels sistemes de fitxers.

Per exemple, localitzant executables amb el *setuid/setgid* activat, fitxers i directoris amb noms estranys, sense restricció d'escriptura o amb propietaris sospitosos.

8) Comprovar que no hi hagi ports escoltant a la xarxa que no estiguin documentats.

9) Comprovar que no hi hagi usuaris amb privilegis d'administració no documentats.

10) Eliminar immediatament els comptes d'usuari del personal que hagi causat baixa a l'organització (guardant les dades el temps establert).

11) Implementar una bona política de contrasenyes (fent que els usuaris escullin contrasenyes fortes i forçant la seva expiració periòdica).

12) Fer un seguiment de l'activitat del sistema analitzant els registres i els informes d'auditoria. S'anomena *logging* el procediment mitjançant el qual s'enregistren (per exemple, en un fitxer o en una base de dades) les activitats que tenen lloc en un sistema o en una aplicació. La importància dels *logs* és evident, ja que ens permetrà esbrinar què ha passat en un sistema i, si cal, prendre les mesures que s'estimin adients en cas d'un incident. És molt important plantejar quines aplicacions han d'enregistrar els *logs*, quan i amb quin nivell de detall ho han de fer, i també quan s'han d'eliminar o migrar a un altre mitjà d'emmagatzematge.

13) Establir contramesures quan es detectin intents d'accés no autoritzats.

Per exemple, configurant polítiques que bloquegin temporalment els comptes d'usuari després de diversos intents d'obrir la sessió amb credencials incorrectes.

14) Auditar el programari instal·lat per assegurar-se que està pertinentment llicenciat.

15) Fer còpies de seguretat i verificar periòdicament que es poden recuperar de manera efectiva.

16) Participar en l'elaboració, revisió i execució, en cas de necessitat, d'un pla de recuperació davant d'un desastre (*disaster recovery*). Aquest consisteix en una sèrie de procediments i eines per tal de recuperar la continuïtat dels serveis TIC en cas de produir-se una disrupció greu a causa de la destrucció de la infraestructura per un desastre (terratrèmol, inundació o incendi, per exemple).

5.3. El sistema de fitxers

En els sistemes de fitxers tradicionals de Linux, cada fitxer pertany a un usuari i a un grup determinats, anomenats *propietaris*. En el moment de creació d'un fitxer, a aquest se li assignaran dos atributs: l'usuari propietari (l'usuari que ha creat el fitxer) i el grup propietari (el grup principal al qual pertany l'usuari que ha creat el fitxer).

Els permisos que es poden assignar sobre un fitxer permeten **tres nivells d'accessos**: lectura, escriptura i execució.

1) Lectura: en cas dels fitxers, permet llegir el seu contingut. Si es tracta d'un directori, permet veure els noms dels fitxers i directoris que hi ha dintre. Es representa amb la lletra «r» (*read*) i el valor numèric 4.

2) Escriptura: en cas d'un fitxer, permet escriure, modificar i esborrar el contingut. Si és un directori, permet crear, esborrar i canviar el nom dels fitxers i directoris que contingui. Es representa amb la lletra «w» (*write*) i el valor numèric 2.

3) Execució: en un fitxer, permet executar-lo. En un directori, permet entrar dintre i veure les metadades (amb les comandes «cd» i «ls», per exemple). Es representa amb la lletra «x» (*execute*) i el valor numèric 1.

Els permisos expressats amb lletres serien la concatenació de les lletres representatives de cada nivell d'accés (posant un «-» si el permís no està assignat), i expressats en valor numèric, la suma dels valors numèrics de cada nivell d'accés. Els valors numèrics es representen en sistema octal: això vol dir que les xifres van del 0 al 7, no del 0 al 9 com el sistema decimal. A la taula 1 podem veure alguns exemples.

Taula 1. Exemples de permisos en un sistema de fitxers Linux

Lletres	Numèric	Permisos atorgats
r--	4	Lectura
-w-	2	Escriptura
--x	1	Execució
rw-	4 + 2 = 6	Lectura i escriptura
r-x	4 + 1 = 5	Lectura i execució
rwx	4 + 2 + 1 = 7	Lectura, escriptura i execució

Aquests permisos s'atorguen per part del propietari (o l'administrador) als usuaris del sistema, agrupats en tres grans classes:

1) Usuari propietari: els permisos fan referència al propietari del fitxer.

2) Grup: els permisos fan referència al grup propietari del fitxer.
3) Públic: els permisos fan referència a tota la resta d'usuaris del sistema.

Per tant, cada fitxer o directori tindrà cinc valors assignats:

a) un usuari propietari,
b) un grup propietari,
c) permisos assignats a l'usuari propietari (usuari),
d) permisos assignats al grup propietari (grup), i
e) permisos assignats a la resta d'usuaris (públic).

Finalment, parlarem breument dels permisos *setuid* i *setgid*. Quan s'assignen a un fitxer permetran executar-lo amb els privilegis del seu usuari o grup propietaris, respectivament. És un mecanisme que permet fer una tasca amb privilegis més elevats. Per exemple, si un fitxer executable que fa una funció determinada té com a propietari l'usuari *root* (l'administrador) i el permís *setuid* assignat, qualsevol usuari no privilegiat que l'executi ho farà amb permisos de *root*.

5.4. Contrasenyes en sistemes Linux

En els sistemes Linux tots els usuaris tenen una entrada a cada un dels fitxers «/etc/passwd» i «/etc/shadow». Per al bon funcionament del sistema, «/etc/passwd» ha de tenir permisos de lectura per a tots els usuaris i «/etc/shadow» ha de tenir permisos de lectura per a l'usuari i el grup d'administració exclusivament, ja que és en aquest últim on es guarden les contrasenyes xifrades dels usuaris.

Les entrades del fitxer «/etc/passwd» tenen el format que es pot veure a continuació (el símbol «:» actua d'element separador entre els diferents camps):

```
nom_usuari:x:UID:GID:informació:directori_de_treball:shell
```

Els camps contenen la informació següent:

1) Nom d'usuari (*login name*): és una cadena que identifica l'usuari, que li ha de permetre iniciar la sessió.

2) Contrasenya xifrada: antigament, en aquest camp es guardava la contrasenya xifrada, tot i que actualment, el més habitual és que contingui el valor «x», que vol dir que la contrasenya xifrada s'ubica al fitxer «/etc/shadow».

3) UID (*user* ID): identificador numèric de l'usuari. El valor 0 sempre correspon a l'usuari administrador del sistema (anomenat *root*).

4) GID (*group* ID): identificador numèric del grup principal (la informació dels grups es guarda al fitxer «/etc/group»).

5) Informació (aquest camp també s'anomena històricament GECOS): conté una cadena amb un comentari amb informació de l'usuari (nom, ubicació, telèfon, etc.).

6) Directori de treball: és la ruta del directori principal de l'usuari (anomenat en anglès *home directory*), on es guardaran les seves dades.

7) *Shell*: és la ruta de l'intèrpret de comandes per defecte.

Com s'ha dit, cada línia del fitxer «/etc/shadow» es correspon amb un usuari i conté informació sobre la contrasenya xifrada de cada un i diversos camps amb informació relativa a l'expiració i bloqueig del compte:

```
nom_usuari:informació_contrasenya_xifrada:exp1:exp2:exp3:exp4:exp5:exp6
```

1) Nom d'usuari: és una cadena que identifica l'usuari.

2) Informació de la contrasenya xifrada: aquí es guarda un identificador de l'algorisme de xifrat que s'utilitza per xifrar la contrasenya, un valor anomenat *salt* (que explicarem de seguida) i la mateixa contrasenya xifrada. S'utilitza el format següent (el caràcter $ actua d'element separador):

```
$id_algorisme_xifrat$salt$contrasenya
```

3) Camps d'expiració i bloqueig del compte: com per exemple, valors numèrics expressats en dies que indiquen quan va ser l'última vegada que es va canviar la contrasenya i quan expirarà.

5.4.1. Atacs a contrasenyes

La finalitat d'aquest tipus d'atac consisteix a esbrinar, desxifrar, esborrar, modificar o inserir contrasenyes en el fitxer que les emmagatzema.

Malgrat l'existència de molts mecanismes d'autenticació, el cert és que avui en dia la **via d'entrada més comuna** per accedir a un sistema és l'ús del nom d'usuari acompanyat de la corresponent contrasenya. En conseqüència, la política de gestió i manteniment de contrasenyes és vital per garantir la seguretat.

Quan un usuari entra al sistema, la contrasenya del fitxer «/etc/shadow» no es desxifra (ja que l'algorisme de xifrat és unidireccional), sinó que es xifra la contrasenya introduïda per l'usuari fent servir el mateix algorisme de xifrat simètric i es compara amb la contrasenya xifrada del fitxer «/etc/shadow». En cas que coincideixin, l'usuari estarà autoritzat a entrar.

Del procés d'obtenir una contrasenya, que ha estat xifrada i emmagatzemada o tramesa per la xarxa, se'n diu ***cracking* de contrasenyes**. Atès el caràcter unidireccional de l'algorisme de xifrat, la manera més evident de trencar les contrasenyes del fitxer «/etc/shadow» serà mitjançant un atac de força bruta. Consisteix a anar provant reiteradament totes les combinacions possibles per formar una contrasenya, explorant tot l'arbre de possibilitats, amb l'esperança de trobar la correcta. Pot funcionar amb contrasenyes de poca longitud, però a mesura que s'incrementa la fortalesa de les contrasenyes, resulta més difícil obtenir l'èxit amb l'atac, ja que es multipliquen exponencialment el nombre de possibilitats.

Una vegada es disposa del fitxer amb les contrasenyes xifrades, es podran mirar d'esbrinar xifrant totes les paraules contingudes en un diccionari (s'anomenen d'aquesta manera els fitxers que contenen molts mots d'un idioma determinat o d'un tema concret, com ara esports o música) i, comparant el resultat amb les contrasenyes xifrades del fitxer «/etc/shadow». Si alguna de les contrasenyes xifrades coincideix amb el resultat de xifrar un mot del diccionari, haurem obtingut una clau d'accés al sistema d'una manera no autoritzada. Aquest tipus d'atac s'anomena ***de diccionari***.

En realitat, el procés de xifrar tots els mots d'un diccionari és més complex del que s'ha explicat, ja que no hi ha un únic xifrat per a cada mot. A l'hora de xifrar un mot (és a dir, el moment en què es va crear o es va canviar la contrasenya), cal tenir en compte un valor numèric aleatori (anomenat *salt* en anglès) que proporcionen diverses codificacions diferents per a cada mot.

Així doncs, cada mot del diccionari haurà de ser codificat amb els possibles valors de *salt* per assegurar que no ens deixem cap possibilitat per explorar. Val a dir, però, que la presència dels bits de *salt* no dificulta (computacionalment no representa un cost insalvable) el trencament de les contrasenyes, però permet que dos usuaris que tinguin la mateixa contrasenya apareguin xifrats d'una manera diferent en el fitxer «/etc/shadow».

Una manera d'agilitar l'atac de diccionari és utilitzant un **diccionari precomputat**. Consisteix a tenir xifrades totes les entrades d'un diccionari, obtenint dues columnes en una taula: la possible contrasenya en clar (sense xifrar) i la mateixa possible contrasenya xifrada. Per crackejar una contrasenya només s'haurà de fer una cerca a la segona columna, amb un cost computacional molt

baix en comparació d'un atac bàsic per força bruta. Aquestes taules necessiten un gran espai de disc, encara que s'han desenvolupat estratègies més eficients com les *rainbow tables*.

Un altra vegada, l'**ús de contrasenyes fortes** dificulta en gran manera els atacs basats en l'ús de diccionaris. En aquest sentit, l'administrador disposa de diverses eines que permeten l'anomenada *comprovació proactiva de contrasenyes*, la qual permetrà rebutjar aquelles que, segons una sèrie de criteris establerts, siguin considerades febles. Així doncs, en cas que un usuari esculli una contrasenya que no satisfà aquests criteris, es veurà obligat a triar-ne una altra.

A més, l'administrador també hauria d'executar amb una certa periodicitat (i amb l'autorització per fer-ho), eines com per exemple John the Ripper per fer atacs de diccionari sobre el mateix fitxer de contrasenyes i poder comprovar d'aquesta manera la robustesa. Utilitats com aquesta automatitzen el procediment d'atac basat en diccionaris i, fins i tot, permeten dur a terme atacs de força bruta, efectius quan les contrasenyes tenen un nombre de caràcters molt reduït.

5.5. Intrusions

Malgrat totes les mesures de seguretat que es pugin implementar, de vegades l'atacant té èxit i aconsegueix prendre el control del servidor. D'aquest fet en direm **intrusió**.

Les **fases** en què sol constar una intrusió en un sistema per part d'un atacant són les següents:

1) Etapa prèvia: recollida d'informació, per tots els mitjans disponibles.

2) Atac inicial.

3) Accés complet al sistema.

4) Instal·lació de mecanismes per obtenir informació i facilitar futurs accessos de l'atacant sense ser detectat.

5) Eliminació d'empremtes.

En cas de detectar-se una intrusió en un dels nostres sistemes, convé seguir un **protocol d'actuació** semblant al que aquí es proposa, amb l'objectiu de minimitzar l'aturada del servei (i la conseqüent pèrdua econòmica) i preservar les evidències que permetin estudiar la naturalesa de l'atac. Consta de les accions següents:

1) Desconnexió de l'equip atacat de la xarxa: amb això intentem evitar que l'intrús causi més danys i que pugui eliminar (si encara no ho ha fet) les empremtes de les seves accions.

2) Fer una **còpia** completa a baix nivell dels dispositius d'emmagatzematge.

3) Restaurar completament el sistema:

a) Reinstal·lar el sistema operatiu i les aplicacions.

b) Aplicar les actualitzacions (*patches*) per solucionar la vulnerabilitat de què s'ha servit l'atacant per introduir-se al sistema.

c) Restaurar les dades des de la còpia de seguretat.

4) Recopilar i analitzar totes les dades possibles sobre l'atac: *logs*, programari instal·lat per l'atacant, porta d'entrada que ha fet servir, etc.

5) Notificar l'atac als usuaris amb la finalitat que canviïn les contrasenyes dels comptes com més aviat millor.

6) Si es detecta que la màquina ha estat utilitzada com a trampolí per atacar altres sistemes, cal avisar els seus responsables. També cal notificar l'atac a l'equip directiu de l'organització del sistema atacat i, en cas que es consideri necessari, denunciar els fets a la policia (tots els cossos policials de l'Estat disposen d'unitats especialitzades en aquest tipus de delicte) i notificar-ho al Computer Emergency Response Team (CERT).

6. Seguretat de les dades

Per evitar els atacs contra la confidencialitat i les tècniques d'espionatge, es poden fer servir diversos **mètodes criptogràfics**. A continuació, definirem els criptosistemes de clau privada i clau pública, les funcions resum i la signatura digital. A més, estudiarem les implicacions que poden tenir aquests elements en la seguretat global dels sistemes.

Una **xifra o criptosistema** és un mètode secret d'escriptura, mitjançant el qual un text en clar es transforma en un text xifrat o criptograma. El procés de transformar un text en clar en text xifrat s'anomena *xifratge*, i el procés invers, és a dir, la transformació del text xifrat en text en clar, s'anomena *desxifratge*. Tant el xifratge com el desxifratge són controlats per una o més claus criptogràfiques.

S'anomena *criptografia* a la ciència i estudi de l'escriptura secreta. Juntament amb la criptoanàlisi (tècnica que té com a objectiu esbrinar la clau d'un criptograma a partir del text en clar i del text xifrat), formen el que es coneix amb el nom de *criptologia*.

Per protegir la confidencialitat de les dades (emmagatzemades o circulant per la xarxa) es poden fer servir criptosistemes de clau privada (simètrics) o de clau pública (asimètrics).

6.1. Criptosistemes de clau privada

Els criptosistemes de clau privada són aquells en els quals l'emissor i el receptor comparteixen **una única clau**. És a dir, el receptor podrà desxifrar el missatge rebut si i només si coneix la clau amb la qual l'emissor ha xifrat el missatge.

L'algorisme més representatiu dels criptosistemes de clau privada és el *Data Encryption Standard* (DES), de l'any 1977. Aquest algorisme xifra la informació en blocs de 64 bits de llargada fent servir claus de 56 bits. Actualment està en desús, ja que no és segur. En lloc del DES s'utilitza una variant anomenada Triple DES o altres algorismes com, per exemple, IDEA, CAST, Blowfish, etc. No obstant això, l'actual estàndard (des de l'any 2002), adoptat com a tal pel Govern dels Estats Units d'Amèrica, és l'anomenat *Advanced Encryption Standard* (AES), representat per l'algorisme Rijndael. Les especificacions de l'AES (que no coincideixen exactament amb el seu representant, l'algorisme Rijndael) determinen una mida de bloc fix de 128 bits i mides de clau de 128, 192 o 256 bits.

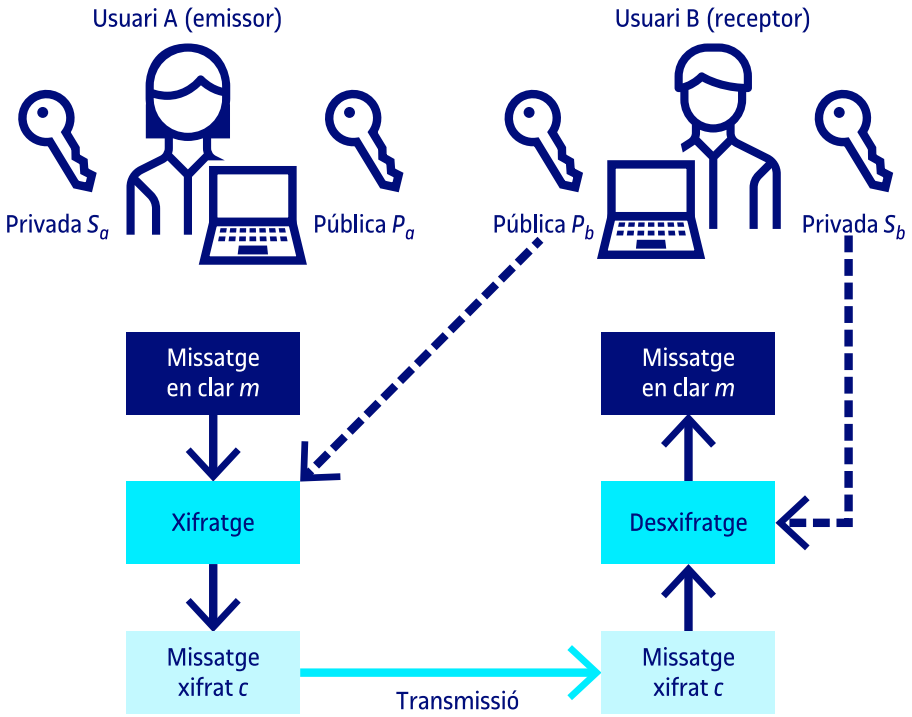
6.2. Criptosistemes de clau pública

La criptografia de clau pública va ser introduïda per Diffie i Hellman, l'any 1976.

Els criptosistemes de clau pública són un tipus de criptosistemes on cada usuari u té associada **una parella de claus <P_u, S_u>**. La clau pública, P_u, és accessible per a tots els usuaris (per exemple, pot aparèixer en un directori públic o es pot intercanviar directament entre els intervinents en la comunicació), mentre que la clau privada, S_u, solament és coneguda per l'usuari u.

La figura 5 mostra un exemple de funcionament d'aquest sistema criptogràfic. Quan un usuari A vol enviar un missatge a un usuari B, xifra el missatge fent servir la clau pública de B (recordem que aquesta clau és accessible per a tothom). Quan el receptor rebi el missatge, únicament el podrà desxifrar utilitzant la seva clau privada (la qual està exclusivament en poder seu).

Figura 5. Comunicació en un criptosistema de clau pública



El criptosistema de clau pública més conegut és l'anomenat RSA, creat per Rivest, Shamir i Adleman el 1978, però n'hi ha d'altres com, per exemple, el DSA (*Digital Signature Algorithm*). Aquests tipus de criptosistemes es basen en funcions matemàtiques unidireccionals (no utilitzen substitucions o transposicions) i són lents si es comparen amb els de clau privada, motiu pel qual se solen fer servir per intercanviar claus simètriques en els protocols de comunicació, però no per xifrar informació.

6.2.1. Signatura digital

Un avantatge molt important d'aquest tipus de criptosistema és que permet la incorporació de la signatura digital. Cada usuari podrà signar digitalment el seu missatge amb la seva clau privada i aquesta signatura podrà ser verificada més tard de manera que l'usuari que l'ha originat no pugui negar que s'ha produït (propietat de no-repudi).

Per poder explicar el mecanisme de la signatura digital, caldrà definir prèviament el concepte de **funció *hash***.

Una **funció *hash* (o funció resum)** és una funció matemàtica que fa correspondre una representació de mida fixa, anomenada *valor resum*, a un missatge *m* de mida variable.

Per exemple, el que es pot veure a continuació és el resultat d'aplicar una funció resum a un fitxer anomenat «MD5.txt:

```
MD5.txt 89736DF30DC47A7D5AC22662DC3B5E9C
```

Les funcions resum han de ser unidireccionals. Una funció resum o funció *hash* (*H*) és unidireccional si per a qualsevol missatge *m'* del recorregut de *H*, és «difícil» (des del punt de vista computacional) trobar *m* de tal manera que $m' = H(m)$.

Els algorismes MD5 i SHA són els que es fan servir més per implementar les funcions resum. A més de l'algorisme SHA-0 (el precursor) i SHA-1, hi ha diverses variants (SHA-224, SHA-256, SHA-384 i SHA-512), totes elles identificades com a SHA-2. A diferència dels seus predecessors, SHA-3 no ha estat dissenyat per l'NSA (National Security Agency, l'agència d'intel·ligència dels EUA) i utilitza la mateixa longitud de *hash* que SHA-2.

A continuació, descriurem el funcionament del protocol de la signatura digital amb funcions resum. Suposem que l'usuari A vol signar el missatge *m* i enviar-lo a l'usuari B.

1) L'usuari A calcula el resum de *m*.

2) A continuació, l'usuari A signa el resum amb la seva clau privada i obté *s*. L'usuari B rebrà el missatge *m* i el resum signat *s*. Si l'usuari B volgués verificar l'origen del missatge rebut, caldria fer les accions següents.

3) L'usuari B calcula el resum de *m*.

4) A continuació, l'usuari B desxifra el resum signat, *s*, fent servir la clau pública de l'usuari A. Si aquest valor coincideix amb el calculat en el pas 3, aleshores *s* és una signatura digital vàlida per al valor resum de *m*.

6.2.2. Certificat digital

A l'hora d'utilitzar la clau pública d'un usuari, com podem saber que és vàlida? Hem de disposar d'algun mecanisme per assegurar-nos que la clau pública pertany realment a l'usuari en concret i que, a més, està vigent (perquè les claus criptogràfiques han de tenir una vida útil limitada). Per resoldre aquest problema es requereix la participació d'una tercera part (anomenada *autoritat de certificació* o CA, *Certificate Authority*) que confirmi l'autenticitat de la clau pública d'un usuari amb l'expedició d'un certificat digital.

El **certificat digital** és un document (un conjunt de dades en un fitxer) que serveix com a prova de propietat d'una clau pública per a tots els participants de les comunicacions que confien en la CA que l'ha emès.

Un certificat no és una prova d'identitat ni serveix per xifrar missatges utilitzant-lo directament (sinó que, més exactament, s'ha de fer servir la clau pública que conté).

Al certificat s'inclou la informació següent: la identitat del propietari, la pròpia clau, el període de vigència de la clau i la firma digital de la CA. Com que els participants en les comunicacions confien en aquesta autoritat, si la firma del certificat és vàlida, es considera segur utilitzar la clau per comunicar-se amb el seu propietari.

La CA té la responsabilitat de verificar que la informació que conté un certificat és correcta abans d'emetre'l per facilitar-lo al seu propietari. De la mateixa manera, també li correspon la missió de revocar el certificat en cas que deixi de ser vàlid (per exemple, en cas que la seguretat del parell de claus del propietari hagi estat compromesa).

6.3. Esteganografia

S'anomena **esteganografia** al conjunt de tècniques que permeten amagar informació, de manera que només l'emissor i el receptor siguin coneixedors de la seva existència.

A diferència de la criptografia, l'esteganografia amaga dades entre altres dades, però no les modifica de manera que siguin illegibles.

A tall d'exemple, mitjançant l'ús de tècniques esteganogràfiques, un fitxer d'una imatge digitalitzada podria ocultar dins seu un fitxer de text amb informació secreta. Des del

punt de vista de l'usuari que examina la imatge, no es podria apreciar cap diferència entre la imatge original i la imatge que oculta les dades confidencials; els dos fitxers tindrien la mateixa mida i tot.

En general, qualsevol fitxer, tant si és una imatge com un document o, fins i tot, un fitxer de so, és susceptible d'amagar algun tipus d'informació. Encara que les diferències entre el fitxer original i el fitxer esteganografiat siguin pràcticament inapreciables, òbviament hi són. Una de les tècniques que es pot fer servir per ocultar informació en un fitxer consisteix a alterar els bits menys significatius del fitxer original, de manera que en aquestes alteracions s'emmagatzemi precisament la informació que es vol ocultar. La mida del fitxer esteganografiat serà exactament la mateixa que la del fitxer original, però el contingut serà lleugerament diferent i els canvis difícilment detectables.

Si es localitza un fitxer xifrat es pot pensar que s'hi amaga alguna cosa confidencial (encara que desxifrar-lo sigui molt complex o gairebé impossible), però en el cas de l'esteganografia, l'anàlisi superficial de les dades ni tan sols pot arribar a crear sospites que hi hagi informació rellevant. Una tècnica senzilla que es pot fer servir per localitzar fitxers que continguin informació oculta consisteix en la comparació dels valors resum dels fitxers sospitosos amb els valors resum dels fitxers originals.

7. Seguretat de la xarxa

Com és conegut, una xarxa és un conglomerat de molts elements heterogenis. Per tant, no podem confiar la seguretat d'un sistema tan complex a l'acumulació de mesures de control en el punt més evident: el servidor. Així doncs, pel que fa a la seguretat de la xarxa (entesa de la manera més genèrica possible), s'hauria de tenir en compte els punts següents:

1) Criptografia: l'ús de la criptografia per mantenir la confidencialitat de les dades que es transmeten per la xarxa, des del seu origen al seu destí.

2) Seguretat de les **topologies** i els tipus de xarxa.

3) Seguretat del **maquinari** de la xarxa: cal destacar que el tallafoc i l'encaminador poden esdevenir els punts més crítics d'una xarxa des del punt de vista de possibles atacs externs.

4) Sistema de **control d'accés** a LAN basat en l'autenticació: mitjançant aquest sistema, els dispositius (en lloc dels usuaris) que volen connectar-se al medi comú s'hauran d'autenticar (basant-se en l'adreça MAC del dispositiu). Aquest mètode requereix tres components:

- **Client**: és el dispositiu (per exemple, un portàtil) que desitja connectar-se a la LAN. Consisteix en un programari instal·lat o integrat en el dispositiu que es vol autenticar.

- **Autenticador**: és l'element que controla l'accés físic al medi, basant-se en l'estat d'autenticació del client. L'estat inicial dels ports de l'autenticador és «no controlat». Si el procés d'autenticació finalitza positivament, aleshores el port canvia el seu estat a «controlat» i el dispositiu és autoritzat a accedir al medi.

- **Servidor d'autenticació**: és el dispositiu de «confiança» que s'encarregarà d'efectuar la validació de la identitat del client. Notificarà el resultat a l'autenticador.

Feta aquesta introducció, a partir d'aquí tractarem els mecanismes, els procediments i els dispositius per gestionar la seguretat de la xarxa més utilitzats: tallafoc, *proxies* i NAT, IDS/IPS, esquers, VPN, detectors, monitors, escàners i tests d'intrusió.

7.1. Tallafocs

En el món de la construcció, un tallafoc és una paret gruixuda que divideix un edifici o el separa dels que té a la vora i que serveix per evitar la propagació del foc. També se'n diu així de la franja de terreny que es manté sense vegetació als boscs i a les zones de cultiu per obstaculitzar l'avenç dels incendis.

Un **tallafoc**, com a sistema de seguretat TIC, fa una funció similar: protegeix xarxes i ordinadors interposant barreres, controlant el trànsit que entra i surt segons unes regles definides (en funció de l'origen, el destí, el tipus o el contingut, entre d'altres).

Segons on estiguin desplegats, n'hi ha de dos tipus:

1) Tallafocs de xarxa: controla el trànsit entre dues o més xarxes.
2) Tallafocs de *host*: controla el trànsit d'un servidor.

Els tallafocs poden treballar a diversos nivells:

1) Filtratge de paquets sense gestió de l'estat (*stateless firewall*): actuen a les capes baixes de la pila TCP/IP. Les regles de decisió (per exemple, rebutjar un paquet o deixar-lo passar) fan servir la informació que contenen les capçaleres IP de cada paquet considerat individualment.

2) Filtratge de paquets amb gestió de l'estat (*stateful firewall*): també treballen a les capes baixes de la pila TCP/IP, tot i que fa un pas més enllà, mantenint un registre del flux de les connexions i no solament dels paquets individuals.

3) Filtratge a nivell d'aplicació: treballen a la capa d'aplicació de la pila TCP/IP.

Cal considerar, a l'hora d'instal·lar un tallafoc, els aspectes següents:

- No s'han d'emprar en lloc d'altres eines, sinó conjuntament amb aquestes. Hem de tenir en compte que el tallafoc serà el punt que rebrà gran part dels atacs sobre els nostres sistemes.

- Centralitza una bona part de les mesures de seguretat de la xarxa en un únic sistema (no cal que sigui un únic dispositiu) i, si es veu compromès, la xarxa quedarà exposada als atacs dels intrusos.

- Pot proporcionar una falsa sensació de seguretat als administradors. No per instal·lar un tallafoc podem assumir que la xarxa és segura i prescindir de vigilar la seguretat dels equips interns de la xarxa.

En general, les decisions bàsiques en el disseny d'un tallafoc són:

- La configuració i el nivell de seguretat potencial del tallafoc estarà relacionada amb l'ús del dispositiu. Així, la política serà diferent si connecta dues subxarxes diferents, que si ha de filtrar els paquets de l'organització amb l'exterior.

- S'ha de definir i implementar, per mitjà de la política de seguretat, el nivell de monitorització i de control desitjat a l'organització. S'ha d'indicar, bàsicament, què s'ha de permetre i què s'ha de denegar. Hi ha dues possibilitats:
  - **Política restrictiva**: es denega tot allò que explícitament no es permet.
  - **Política permissiva**: es permet tot, excepte el que s'ha negat explícitament.

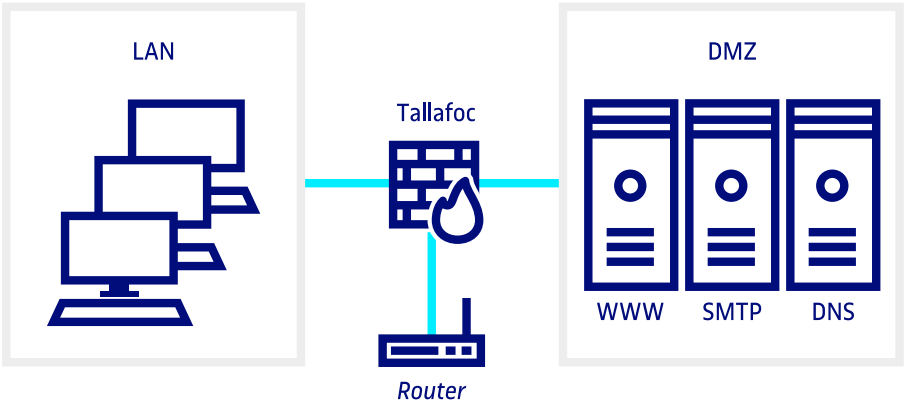
Hi ha diverses arquitectures de tallafocs, però aquí ens centrarem en les DMZ.

Una **DMZ** (de l'expressió en anglès *demilitarized zone* o zona desmilitaritzada) és una subxarxa aïllada interposada entre la subxarxa interna i la xarxa pública.

A la DMZ es connecten els servidors que allotgen els serveis que han de ser accessibles des de l'exterior i que, per tant, són els més susceptibles de rebre atacs i resultar compromesos (com ara, el correu o el *front-end* web). Així, les màquines connectades a la subxarxa privada interna (LAN) no tenen exposició directa a la xarxa pública (internet), ni tampoc als servidors de la DMZ, fet que incrementa la seva protecció. Amb aquest sistema de segmentació es determinen, doncs, tres zones: la xarxa pública (insegura), la subxarxa privada interna (segura) i la DMZ (de seguretat intermèdia).

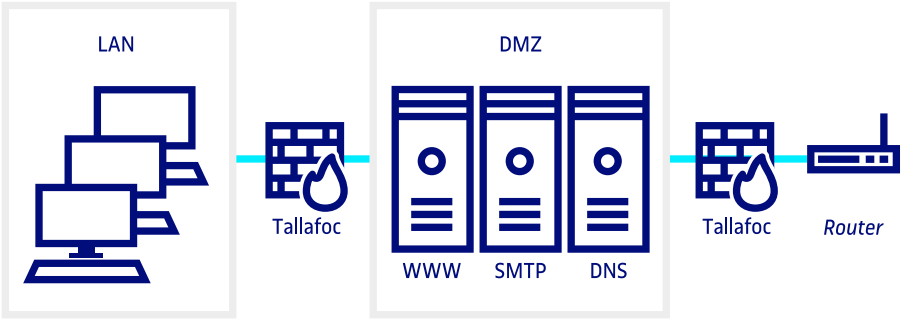
Hi ha diverses implementacions de l'arquitectura DMZ, però en podem destacar dues: amb un i dos tallafocs. En el primer cas, es tindria un tallafoc amb tres interfícies de xarxa, com es pot observar a la figura 6.

Figura 6. DMZ amb un sol tallafoc



La figura 7 il·lustra la implementació amb dos tallafocs (un d'exterior, o perimetral, i un altre d'interior, o de *back-end*), la qual es considera més segura. Té una disponibilitat més alta (en el disseny anterior, el tallafoc resultava un punt únic de fallada) i, a més, un atacant hauria de comprometre la seguretat de dos dispositius en comptes d'un per accedir a la subxarxa privada interna.

Figura 7. DMZ amb dos tallafocs



7.2. *Proxies*

Un *proxy* és un servei que actua com a **intermediari** entre un client i un servidor en una xarxa.

Els més habituals són els *proxies* web, encara que també s'utilitzen de molts altres protocols (SMTP, DNS, SOCKS). El client no envia la petició directament al servidor, sinó que ho fa al *proxy*, que pren la decisió de redirigir-la en nom del client cap al servidor o rebutjar-la. Quan la resposta del servidor arriba al *proxy*, de la mateixa manera, la redirigeix al client. L'objectiu dels *proxies* és augmentar la privacitat, evitar mecanismes de censura, controlar els accessos als recursos de xarxa, filtrar continguts, fer la distribució de la càrrega i millorar el rendiment (amb els *proxies* que disposen de *cache*).

7.3. NAT

La traducció d'adreces de xarxa (*network address translation*, NAT) és una funcionalitat dels encaminadors que permet **canviar l'espai d'adreces IP** dels paquets que hi circulen modificant les seves capçaleres. Gràcies a aquest mètode, tots els equips d'una xarxa poden tenir adreces privades i utilitzar una passarel·la amb funcionalitat de NAT i una IP pública per tenir sortida a internet. Aquesta tècnica incrementa la seguretat dels sistemes, ja que aquests no seran accessibles directament des de l'exterior i, a més, tot el trànsit provinent de la xarxa circularà per internet amb la IP de la passarel·la. Precisament per això, aquesta tècnica és molt útil, ja que no ha d'utilitzar adreces IPv4 públiques per a tots els equips, que, recordem, són molt escasses.

7.4. Sistemes de detecció i prevenció d'intrusos (IDS i IPS)

Un sistema de detecció d'intrusos (*intrusion detection system*, IDS) és un dispositiu que **monitoritza** l'activitat d'un sistema i identifica esdeveniments que puguin ser de caràcter maliciós, generant un registre i possiblement activant una alarma.

1) Segons el **sistema d'implementació**, podem establir tres grups:

a) Basats en xarxa: monitoritzen una xarxa i solen ser elements passius.

b) Basats en *host*: monitoritzen un *host* (o un conjunt d'aquests) i permeten un control més detallat, registrant els processos i usuaris implicats en les activitats capturades per l'IDS. Consumeixen recursos del *host* i incrementen el flux d'informació per mitjà de la xarxa.

c) Basats en aplicacions: monitoritzen els *logs* d'una aplicació específica per detectar activitats sospitoses. Consumeixen molts recursos del *host*.

2) Segons el **mecanisme d'anàlisi** que empren, podem distingir entre dos tipus:

a) Basats en patrons: de forma similar als paquets de programari d'antivirus, aquests tipus d'IDS monitoritzen la xarxa a la recerca de patrons (signatures) que permetin identificar un atac ja conegut. Aquests tipus d'IDS requereixen que les bases de dades de signatures d'atac estiguin constantment actualitzades.

b) Basats en anomalies (tècniques heurístiques): en aquest cas, l'IDS cercarà comportaments considerats sospitosos, com ara: intent d'identificació remota de sistemes operatius i aplicacions per part d'un atacant (*fingerprinting*), escaneigs de ports, localització de vulnerabilitats, ús d'*exploits*, atacs de denegació de servei, paquets mal formats, etc.

7.4.1. Snort

Snort és un programari de codi obert per a la detecció i prevenció d'intrusos en una xarxa, analitzant el trànsit en temps real. Utilitza un llenguatge basat en regles per poder definir alertes quan es detecta una activitat a la xarxa que no es correspon amb la que es considera normal. Pot funcionar en diferents modes: només detector, registre de paquets (guardant l'activitat de la xarxa en un fitxer o base de dades), IDS (analitzant el trànsit i aplicant les regles per generar alertes informatives) i IPS (el trànsit que es considera maliciós després de l'anàlisi es descarta).

7.5. Esquers i xarxes d'esquers (*honeypots* i *honeynets*)

Un esquer (*honeypot* en anglès) és un sistema que s'ofereix de forma deliberada a l'accés públic amb la finalitat d'estudiar les pautes dels possibles atacants que pugui tenir.

Per tant, aquests tipus de sistemes no podran contenir cap informació important i necessitaran eines passives d'auditoria que puguin permetre conèixer, amb posterioritat a l'atac, què és el que ha passat. Freqüentment, aquests tipus de sistemes també contenen directoris o noms de fitxers amb identificacions llamineres que despertin la curiositat dels atacants. A més de la seva finalitat d'anàlisi, també poden utilitzar-se per distreure l'atenció dels possibles atacants del veritable sistema, el qual hauria d'estar aïllat i protegit. Generalment, els esquers no estan completament protegits i les aplicacions i els dispositius es configuren amb les opcions per defecte, les quals solen presentar múltiples forats de seguretat.

La generalització del concepte d'esquer a una xarxa s'anomena *honeynet*. En aquest cas, els atacants, a més de servidors no completament protegits, també poden trobar altres dispositius a la xarxa, com ara encaminadors o tallafocs.

7.6. Xarxes privades virtuals

Una **xarxa privada virtual** (*virtual private network*, VPN) és un programari que estén una xarxa privada per mitjà d'una xarxa pública (com per exemple, internet), permetent als usuaris en remot transmetre i rebre informació com si estiguessin connectats directament a la xarxa privada. Per garantir la confidencialitat de les dades és habitual que les VPN xifrin el trànsit.

D'aquesta manera els treballadors d'una organització poden accedir de manera segura a les aplicacions corporatives connectant-se mitjançant la VPN des de qualsevol lloc del món. A més, el seu ús és útil per evitar els mecanismes de censura a internet que determinats governs imposen als seus ciutadans. També serveixen per evitar les restriccions geogràfiques que activen alguns serveis d'internet (com ara d'*streaming* o diaris digitals).

Una VPN es crea establint una connexió virtual punt a punt utilitzant un protocol de *tunnelling* sobre les xarxes existents. La tècnica de *tunnelling* consisteix a encapsular un protocol de xarxa sobre un altre.

La característica que converteix la connexió «pública» en «privada» és el que s'anomena un *túnel*, terme referit al fet que únicament ambdós extrems són capaços de veure el que es transmet per aquest, convenientment encriptat i

protegit de la resta d'internet. La **tecnologia de túnel** xifra i encapsula els protocols de xarxa que s'utilitzen en els extrems sobre el protocol IP. D'aquesta forma podem operar com si es tractés d'un enllaç dedicat convencional, de forma transparent a l'usuari.

El protocol més estès per a la creació d'una VPN és ***Internet Protocol Security*** (IPSec). Consisteix en un conjunt d'estàndards que comproven, autentiquen i encripten les dades en els paquets IP, i les protegeixen en les transmissions. En definitiva, IPSec aporta al trànsit IP la propietat de confidencialitat mitjançant l'encriptació, d'integritat mitjançant el rebuig del trànsit modificat il·lícitament, i també d'autenticació i prevenció contra els atacs de reproducció. IPSec utilitza certificats (signats digitalment per una entitat emissora de certificats) per comprovar la identitat d'un usuari, equip o servei, i enllacen de forma segura una clau pública amb l'entitat que disposa de la clau privada corresponent.

El protocol té dues formes operacionals:

1) Mode transport: emprat per protegir connexions individuals d'usuaris remots. Les comunicacions s'encripten entre un ordinador remot (el client VPN) i el servidor de VPN. Aquesta configuració pot ser d'interès, per exemple, quan l'organització disposa de dades confidencials que haurien de romandre ocultes per a molts usuaris. D'aquesta manera, se separen les dades confidencials gràcies al servidor VPN, de forma que només hi puguin accedir els usuaris autoritzats.

2) Mode túnel: les comunicacions s'encripten entre dos dispositius de tipus encaminador (o un encaminador i el servidor de VPN), amb el qual es protegeixen totes les comunicacions de tots els ordinadors situats darrere de cada encaminador.

A més de les VPN basades en la xarxa pública, també cal esmentar les VPN de confiança, en les quals l'extensió es porta a terme sobre una xarxa privada de confiança, i per tant, permet estalviar d'encriptar el flux d'informació que circula per mitjà del túnel. Els protocols emprats en aquests tipus de xarxes són diversos i poden ser: *Asynchronous Transfer Mode* (ATM), *Multi-Protocol Label Switching* (MPLS) i *Layer 2 Forwarding* (L2F).

VPN

Altres solucions esteses de VPN, a banda d'IPSec, són OpenVPN i WireGuard.

7.7. Detectors

S'anomena ***detectors*** al programari que permet la captura i l'enregistrament de les dades que circulen per una xarxa. El seu funcionament es basa en l'activació del mode promiscu de les interfícies de xarxa.

Amb l'activació d'aquest mode, una estació de treball podrà monitoritzar, a més dels paquets de dades que s'hi adrecen d'una manera específica, el trànsit sencer de la xarxa. Això inclou, per exemple, la captura d'informació que no circuli xifrada, com ara noms d'usuari, contrasenyes, correus electrònics o qualsevol altre document confidencial.

L'activitat dels detectors és difícil de descobrir perquè no deixen gaires empremtes. No podem tenir constància de la informació que pot haver estat interceptada per l'acció dels detectors, però es poden fer servir mesures de protecció contra la fuita d'informació, com per exemple segmentar el trànsit de la xarxa i, sobretot, la criptografia.

Finalment, notem que els detectors tenen molts avantatges per a l'administrador del sistema, no solament per monitoritzar, per exemple, el flux d'informació que circula per la xarxa, sinó per ajudar a investigar i diagnosticar l'origen d'algun problema de funcionament que es pugui presentar, inclosos els relacionats amb la seguretat.

Figura 8. El programari detector Tcpdump

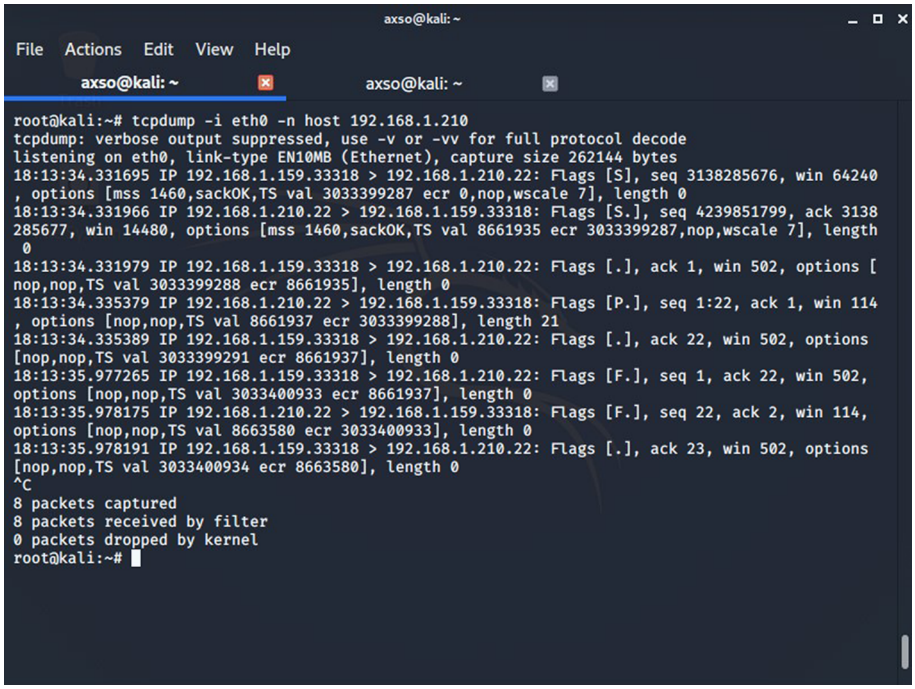
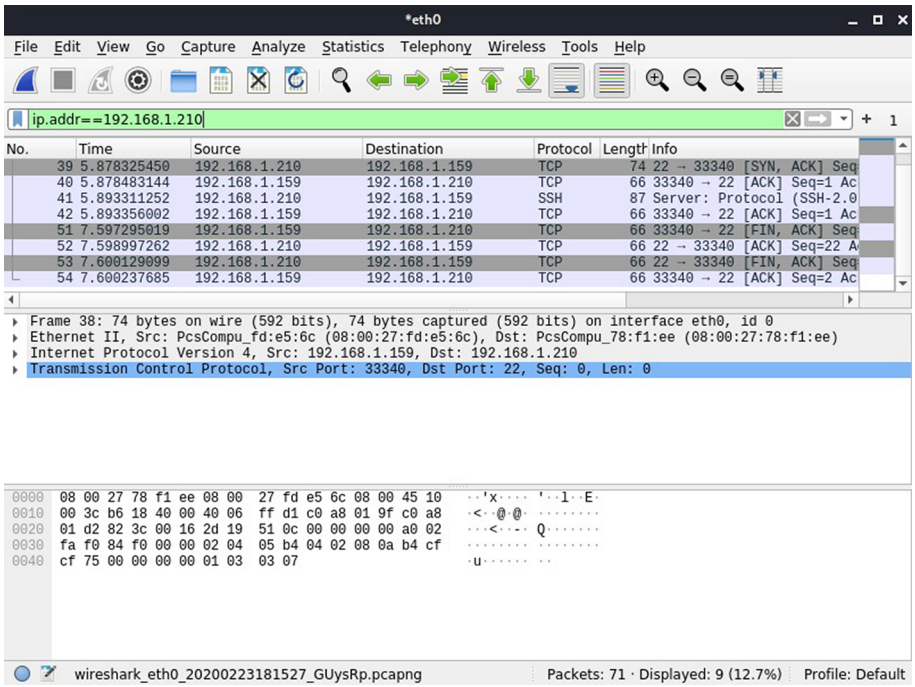


Figura 9. El programari detector Wireshark



Els detectors més habitualment utilitzats són Tcpdump (figura 8) i Wireshark (figura 9), dels quals s'acompanyen captures de pantalla en ple funcionament.

7.8. Monitorització de la xarxa

La **monitorització** consisteix a comprovar l'estat d'una xarxa de manera contínua per detectar problemes de funcionament (baix rendiment, fallades d'elements o altres situacions considerades anòmales respecte d'un comportament habitual). Quan es detecta una incidència, es genera una alarma que permetrà alertar l'equip d'administradors perquè puguin estudiar el problema i treballar en una solució.

Funciona enviant periòdicament, des d'una sonda, peticions de servei als elements monitoritzats i analitzant la resposta obtinguda: en funció de si aquesta té uns paràmetres dintre o fora d'uns límits establerts, es considera que l'element està operatiu (funciona correctament), inoperatiu (no funciona correctament) o en estat degradat (presenta una funcionalitat limitada).

La petició de servei enviada des de la sonda a un element a monitoritzar pot variar el seu nivell de detall i la regularitat amb què es fa, segons el grau de coneixement que es requereixi assolir del seu estat. Per exemple, per monitoritzar l'estat d'un servidor web, es podrien enviar diferents peticions de servei, per exemple cada cinc minuts:

- Obrir i tancar una connexió TCP al port 443 (HTTPS), si només es vol veure si el servidor web està escoltant i responent.

- Enviar una petició HTTPS i comprovar la resposta obtinguda, si es vol analitzar que el recurs que s'ha demanat i s'ha servit és l'esperat.

Es poden recollir diferents mètriques que siguin útils a l'hora de detectar anomalies i assegurar la qualitat del servei, com ara el temps de resposta, la disponibilitat i l'*uptime*.

Monitorització

Algunes solucions de programari habitualment emprades per a la monitorització són: Nagios, Zabbix, Ganglia, Cacti, Netdisco, Netdata, entre d'altres.

7.9. Escàners de xarxa i de vulnerabilitats

Els **escàners** són eines de seguretat que serveixen per detectar les vulnerabilitats d'un sistema. En general, es poden dividir en dues categories: els escàners de sistema i els escàners de xarxa.

Els **escàners de sistema** s'utilitzen per detectar les vulnerabilitats del servidor localment: problemes de configuració, permisos erronis, contrasenyes febles, etc. Els **escàners de xarxa** analitzen els serveis i ports disponibles de sistemes remots a la recerca de debilitats conegudes que puguin ser aprofitades per atacants (en certa manera, doncs, automatitzen les tasques que duria a terme un intrús remot).

Un port de xarxa indica un punt pel qual entra o surt la informació d'un sistema. Els protocols d'internet utilitzen emissor i receptor, ports d'emissió i recepció comuns en ambdós extrems de la comunicació.

L'anomenat *escaneig de ports* consisteix a descobrir quins ports estan oberts en una màquina remota. Els ports oberts constitueixen una informació molt interessant per als possibles intrusos, ja que les vulnerabilitats dels serveis de xarxa que estan en funcionament poden permetre, si són atacades amb èxit, l'accés no autoritzat al sistema. L'assignació dels ports corresponents als serveis més habituals no és arbitrària, sinó que es determina per part de la Internet Assigned Numbers Authority (IANA).

Exemples d'assignació de ports a serveis d'internet:

- Port TCP 25: SMTP
- Port TCP/UDP 53: DNS
- Port TCP 80: HTTP
- Port TCP 143: IMAP
- Port TCP 443: HTTPS

Els ports entre el 0 i el 1023 són els anomenats *ports de sistema* o *ports reconeguts* (*well-known ports*) i són els utilitzats per processos de sistema que ofereixen serveis per mitjà de la xarxa.

Els ports situats en el rang del 1024 fins al 49151 s'anomenen *ports registrats*. Són assignats per la IANA, si una entitat ho sol·licita per fer l'associació a un servei específic. Al contrari dels ports reconeguts, no necessiten privilegis d'administrador per obrir-los.

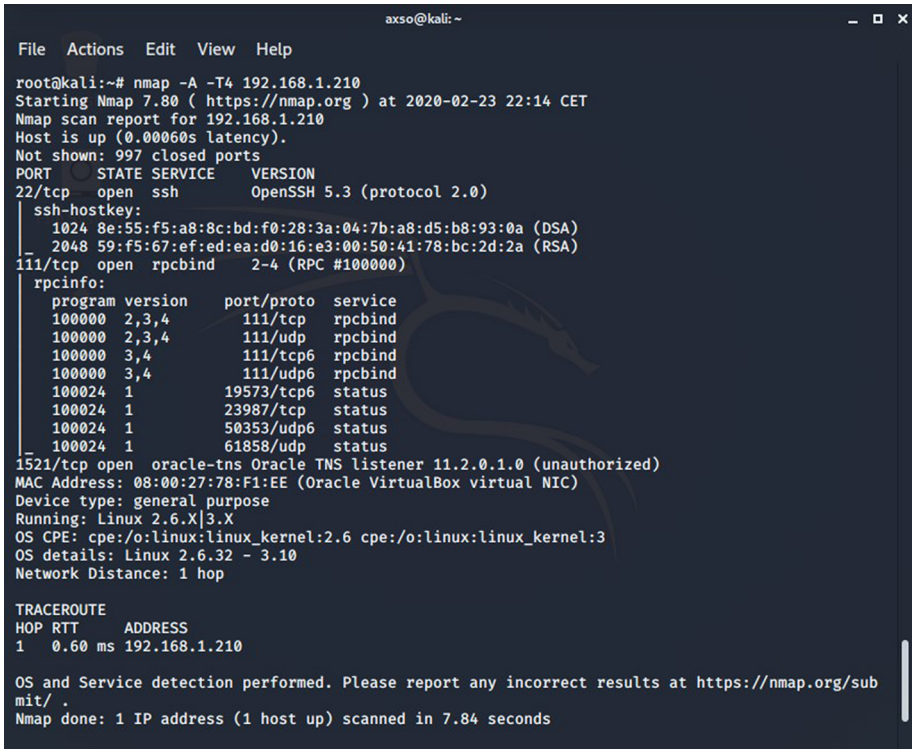
Els ports del 49152 al 65535 són els anomenats *ports dinàmics o privats*, no es poden registrar per mitjà de la IANA i s'utilitzen per a serveis privats o de caràcter provisional.

Tot i que els escàners són eines de molta utilitat per als administradors de sistemes, val a dir que els atacants també en poden fer un ús maliciós. Els escàners permeten l'automatització de centenars de proves per localitzar les vulnerabilitats d'un sistema.

7.9.1. Nmap

Nmap és l'escàner de xarxa més utilitzat. Permet descobrir sistemes connectats a una xarxa i saber quins ports tenen oberts, identificar els sistemes operatius que corren i els serveis disponibles, i, per mitjà d'un motor d'*scripting* (anomenat NSE), realitzar certes funcionalitats de detecció i explotació de vulnerabilitats. La figura 10 mostra un exemple de sortida amb informació que mostra l'escàner.

Figura 10. Escaneig de ports amb Nmap

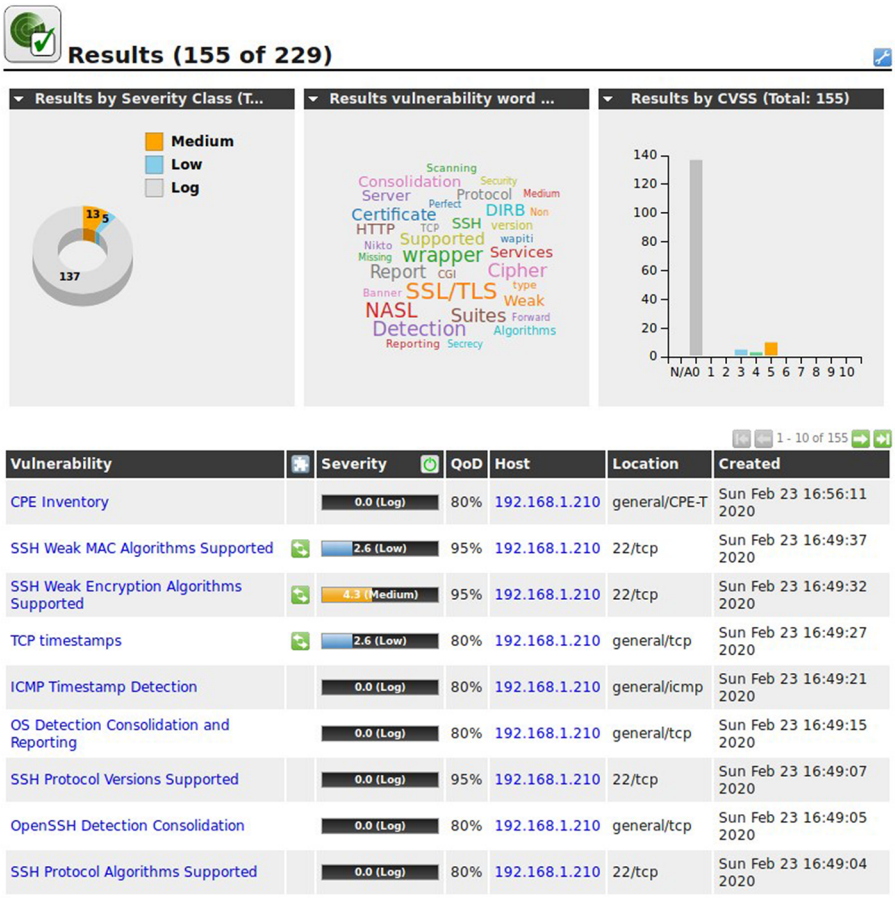


7.9.2. OpenVAS

OpenVAS (*open vulnerability assessment system*) és un escàner de vulnerabilitats format per un conjunt d'eines de programari lliure per analitzar els sistemes amb l'objectiu de localitzar i gestionar les possibles vulnerabilitats que hi puguin haver presents. Es va iniciar com un *fork* (un projecte nou que neix a partir d'un altre) de Nessus.

Té funcionalitats molt útils com el fet de permetre ajustar la velocitat dels escaneigs i la inclusió d'un llenguatge de programació propi per automatitzar tasques. La figura 11 mostra un informe d'escaneig d'un sistema fet amb OpenVas. Com es pot apreciar, té una interfície molt amigable, cosa que fa que sigui molt fàcil interpretar la informació que proporciona.

Figura 11. Informe d'escaneig d'un sistema fet amb OpenVAS



7.10. Tests d'intrusió (*pentesting*)

Un **test d'intrusió** (*pentest*, en anglès) és un conjunt de procediments amb els quals es pretén buscar vulnerabilitats en un sistema utilitzant tècniques d'atac, amb l'autorització expressa del seu responsable, per tal d'avaluar i documentar el seu nivell de seguretat.

L'objectiu és fer una anàlisi completa de riscos, identificant les fortaleses i les debilitats, involucrant tots els components del sistema (físics, lògics, processos i persones).

A la part final del test s'elabora un seguit de documentació que es lliura al responsable dels sistemes, en què es recullen els problemes que s'hagin localitzat, una valoració del seu impacte i suggeriments sobre les contramesures a adoptar.

En funció dels nivells de coneixement i d'accés previs en relació amb els sistemes objecte del test d'intrusió que tingui el personal tècnic que s'encarregarà de fer-lo, es pot establir la divisió següent:

- **Tests d'intrusió de capsa negra (*black box*)**: nivells de coneixement i d'accés previs mínims (els equiparables als que pugui tenir una persona aliena a l'organització a la qual pertanyen els sistemes).

- **Tests d'intrusió de capsa gris (*grey box*)**: nivells de coneixement i d'accés previs intermedis (els equiparables als d'un usuari pertanyent a l'organització).

- **Tests de capsa blanca (*white box*)**: nivell de coneixement i d'accés previs avançats.

Hi ha diversos marcs de treball i metodologies estàndard per fer els tests d'intrusió, dels quals indiquem els més importants: *Open Source Security Testing Methodology Manual* (OSSTMM), *Penetration Testing Execution Standard* (PTES), *NIST Special Publication 800-115*, *Information System Security Assessment Framework* (ISSAF) i *OWASP Testing Guide*, entre d'altres.

7.10.1. Fases dels tests d'intrusió

Un test d'intrusió consisteix habitualment en un conjunt de procediments comuns i quasi estandarditzats, que es poden agrupar en les fases consecutives següents:

1) Fase preliminar: en aquesta fase s'estableixen, treballant conjuntament amb els responsables dels sistemes objecte del test: l'abast de la intervenció, els acords contractuals sobre els serveis a prestar i sobre la confidencialitat, i les condicions del test (per exemple, el test podrà comportar una pèrdua de servei o causar algun tipus d'afectació?). En aquesta etapa es recullen i es validen legalment els permisos i autoritzacions necessaris per fer el test.

2) Fase de reconeixement: l'objectiu d'aquesta fase és doble:

a) Obtenir informació que pugui ser rellevant per trobar vulnerabilitats, de dues maneres:

- Reconeixement passiu o extern: sense interactuar directament amb el sistema auditat, fent-ho com si es tractés d'algú totalment aliè a l'organització. Es poden utilitzar en aquesta fase atacs d'enginyeria social o diverses fonts externes, com ara cercadors, xarxes socials, bases de dades (per exemple, Google, Shodan o Builtwith). Les dades recollides que utilitzen fonts públiques s'anomenen OSINT (*open-source intelligence*).

- Reconeixement actiu o intern: interactuant directament amb el sistema auditat.

b) Elaborar un model de les possibles amenaces, identificant els actius més valuosos i més vulnerables, i establir quines amenaces els poden afectar.

3) Fase d'anàlisi de vulnerabilitats: s'utilitzen escàners de vulnerabilitats i altres eines per conèixer amb el màxim detall possible els sistemes: xarxes, elements connectats, sistemes operatius, ports oberts, serveis disponibles, etc.

4) Fase d'explotació de les vulnerabilitats: es realitzen atacs contra les vulnerabilitats localitzades per veure si es poden explotar de manera efectiva. Amb l'eina Metasploit es pot automatitzar l'explotació de vulnerabilitats conegudes. Quan s'aconsegueix el control d'un sistema per mitjà d'un atac exitós:

a) S'estableixen els mecanismes per mantenir aquest control en el futur, intentant fer-ho de manera indetectable (per exemple, instal·lant *rootkits*).

b) S'eliminen les evidències de la intrusió, com ara: restes derivades de l'atac, *logs* d'accessos, etc.

c) Es recull el màxim d'informació possible continguda al sistema i es documenta.

d) S'accedeix des del sistema de què s'ha pres control a d'altres que puguin estar a la mateixa xarxa (evitant així, per exemple, els tallafocs que possiblement estiguin desplegats), per repetir el mateix procés reiteradament. D'aquest tècnica se'n diu *pivoting*.

5) Redacció de l'informe i presentació dels resultats: es documenta en un informe escrit el procés realitzat, aportant tota la informació recollida, i es presenta als responsables del sistema incloent un pla d'acció per solucionar els problemes detectats.

8. Seguretat del núvol

En aquest apartat tractarem de manera introductòria les particularitats sobre la seguretat que ha de tenir en compte l'administrador de sistemes TIC quan es treballa amb el sistema de virtualització de contenidors, quan a l'organització se segueix la metodologia DevOps per desenvolupar i desplegar programari, i quan s'utilitzen els recursos d'un núvol públic.

8.1. DevOps

DevOps és una metodologia en la qual els equips de treball combinen les funcions específiques de desenvolupament de programari (Dev) i les d'operacions TIC (Ops) amb l'objectiu de fer més curt el cicle complet de planificació, creació, comprovació i desplegament de sistemes d'informació (habitualment programari).

Per fer-la efectiva se segueix el conjunt de pràctiques d'enginyeria del programari anomenat **integració i desplegament continus** o CI/CD (*continuous integration / continuous deployment*). Consisteix a fer que els cicles de producció del programari siguin més curts i freqüents (fins i tot, diverses vegades al dia), assegurant que es puguin alliberar versions en qualsevol moment perquè siguin desplegades en producció de forma automàtica. L'objectiu és introduir canvis en el servei ofert amb més agilitat, velocitat i freqüència (augmentant, així, la satisfacció dels usuaris), però també reduir costos, temps i risc (en teoria, es redueix risc aplicant en el programari molts canvis petits, en comptes de pocs canvis grans, com es fa en el paradigma tradicional).

Encara que no és exclusiu d'aquesta manera de treballar, al mateix temps s'ha estès molt la utilització de diversos sistemes de virtualització, implementats tant en la infraestructura física *on-premises*, com al núvol. D'aquests, els que representen més avantatges aquí per la seva funcionalitat, facilitat d'ús, escalabilitat i seguretat, són els contenidors.

Els **contenidors** són sistemes de virtualització a nivell de sistema operatiu, és a dir, és el nucli que disposa de la funcionalitat de poder mantenir entorns d'execució aïllats un de l'altre i de la instància global. A causa d'això consumeixen comparativament menys recursos que, per exemple, les màquines virtuals.

Docker és una de les tecnologies de contenidors més populars, tot i ser relativament recent. És un sistema per empaquetar i distribuir programari en què cada contenidor té incorporats tots els requeriments: dependències, biblioteques, fitxers de configuració, etc.

Kubernetes és un paquet de programari que permet l'automatització del desplegament, l'escalat i la gestió de contenidors[6] en clústers. Segurament és el d'ús més estès, encara que n'hi ha altres, com ara Docker Swarm. S'encarrega de desplegar en diversos servidors treballadors els contenidors que contenen les aplicacions que s'executaran, proveint escalabilitat i alta disponibilitat de manera dinàmica.

(6)En anglès, *container orchestrator*.

Algunes de les **mesures de seguretat** que es poden aplicar a nivell dels diversos components del gestor de contenidors quan es treballa amb la metodologia CI/CD són les següents:

1) Contenidors:

a) És important mantenir les dades fora del contenidor (els contenidors són immutables i es creen i destrueixen contínuament).

b) Mantenir els components distribuïts en contenidors separats, en la mesura que es pugui, reduint així la superfície d'atac de cada un.

c) Per fer més difícil que els atacants puguin escapar de l'entorn d'execució d'un contenidor utilitzant algun *bug* del gestor de contenidors o del nucli del servidor hoste, és convenient que les aplicacions que corren dintre del contenidor ho facin amb un usuari no privilegiat.

d) Convé que el sistema de fitxers arrel del contenidor sigui de només lectura (així un atacant no podrà reescriure un fitxer binari amb codi maliciós).

2) Cadena de construcció (*build pipeline*):

a) Un cop s'ha acabat la part estrictament de desenvolupament del programari en un cicle, abans de compilar, crear els paquets i la imatge, és convenient fer una anàlisi estàtica automatitzada del codi.

b) Si s'utilitza un nombre limitat d'imatges base per als contenidors i que, a més, siguin el més senzilles possible, eliminant tot allò que no sigui estrictament necessari, s'aconsegueix reduir la superfície d'atac.

c) Convé disposar d'un procediment repetible i que pugui ser automatitzat per recrear les imatges des de zero.

d) Abans de donar-les per bones per a la seva publicació, és important escanejar les imatges per trobar-hi vulnerabilitats conegudes (per exemple, paquets de programari no actualitzats), localitzar l'existència de secrets (contrasenyes en clar o claus criptogràfiques), evitar fuites d'informació sensible (utilitzant la cerca per paraules clau o expressions regulars) i comprovar si hi ha elements incorrectes en els fitxers de configuració del programari inclòs. Si les anàlisis troben incidències, s'ha de descartar la imatge i evitar que continuï el procés de la cadena i que arribi al registre i a producció. Per dur a terme aquestes funcions es pot utilitzar programari com ara Anchore Engine, CoreOS Clair i OpenSCAP.

e) Per obtenir una millor traçabilitat, la persona o equip que s'ha encarregat de la producció de la imatge, la firmen criptogràficament.

f) Si la imatge ha passat per tots els controls, es publica al registre d'imatges i es pot desplegar en producció.

3) Xarxa:

a) Com en un sistema tradicional, convé implementar tallafocs i un sistema de monitorització que permeti detectar i respondre a amenaces de seguretat. També enregistrar i analitzar els intents d'accessos indeguts.

b) Per obtenir un nivell de control més fi, resulta una bona pràctica gestionar polítiques de seguretat de xarxa a nivell de contenidor, per restringir l'accés i determinar quin contenidor es pot comunicar amb quin, utilitzant, per exemple, eines com Calico.

4) Servidor hoste i gestor de contenidors:

a) Fer un *hardening* adequat del sistema operatiu on hi ha allotjats els contenidors, especialment si algun sistema de fitxers és accessible des dels contenidors. També cal considerar si és possible utilitzar una arquitectura *serverless*.

b) Fer el més segur possible el gestor de contenidors, restringint els accessos a l'API.

c) Protegir elements clau, com ara la base de dades central de Kubernetes, anomenada etcd, on es guarda tota la informació del clúster. Convé implementar un pla de còpies de seguretat adequat.

d) Utilitzar RBAC (*role based access control*) per assignar rols a cada usuari i al compte de servei.

e) Convé prestar especial atenció a protegir el registre d'imatges, un dels objectius més perseguits pels atacants, ja que tenir-ne el control permetrà manipular les imatges.

f) Utilitzar eines com Seccomp o Falco per poder restringir l'accés a determinades crides del sistema del servidor hoste fetes des de cada contenidor.

5) En temps d'execució (*runtime*):

a) Assegurar-se que només entren a producció els contenidors que estan degudament firmats criptogràficament.

b) Escanejar sistemàticament les imatges en temps d'execució per identificar canvis que s'hagin produït respecte a la imatge original, per determinar si poden haver estat conseqüència d'un atac. També escanejar per a la cerca de vulnerabilitats que s'hagin conegut posteriorment al desplegament de les imatges. Generar alertes per poder retirar de producció els contenidors que es considerin insegurs, enviant un informe als desenvolupadors perquè es pugui analitzar el problema.

c) Utilitzar mecanismes que incrementin el grau d'aïllament que ja incorpora el mateix gestor de contenidors, com per exemple, utilitzant Kata Containers o gVisor.

8.2. Núvol públic

A banda de les qüestions habituals relacionades amb la seguretat en el paradigma tradicional de computació, de les quals la gran majoria també estaran presents quan tinguem allotjats els nostres recursos en un núvol públic, s'hauran de tenir en compte alguns aspectes que en són particulars:

1) Necessàriament haurem de dipositar la nostra confiança en un proveïdor de serveis al núvol i aquesta necessitat de compartició de responsabilitats amb una altra part aliena cada vegada s'anirà incrementant perquè és habitual anar passant al núvol més parts de la infraestructura i més crítiques. Això representa un risc que s'ha de tenir en compte i avaluar.

2) Aquest model de responsabilitat compartida es concreta en el fet que el proveïdor és l'encarregat de la seguretat de la infraestructura física i del sistema de virtualització (hipervisor, gestor de contenidors, orquestradors), i el client de la seguretat dels recursos virtuals que contracta i de les aplicacions que desplega.

3) S'ha de tenir present el perill que pot comportar el fet de quedar atrapat en la relació que es té amb un proveïdor, en el sentit de no poder traspassar fàcilment els serveis que hi tenim contractats a algun altre lloc (de les situacions en què hi ha dificultats d'aquest tipus, en direm *vendor lock-in*).

4) S'ha de controlar que la facturació que fa el proveïdor pels serveis contractats sigui la correcta i no es produeixi cap frau.

5) Hi ha el risc que el personal contractat pel proveïdor pugui utilitzar els seus privilegis d'administració de manera abusiva amb finalitats malicioses i això ens pugui afectar.

6) Abans de contractar els serveis d'un proveïdor, hauríem de fer un estudi del seu historial en què es faci referència a la qualitat del servei ofert. Els serveis contractats hauran de recollir uns acords de nivell de servei (*service-level agreement*, SLA) que s'ajustin a les necessitats de la nostra organització: la qualitat del servei requerida ha d'estar perfectament establerta.

7) En un entorn virtual es desconeix amb qui es comparteixen els recursos. Els proveïdors no haurien d'accedir al contingut de les màquines virtuals dels seus clients perquè estan obligats a preservar la seva confidencialitat, però monitoritzen l'ús de recursos per detectar qualsevol comportament anòmal (com podria ser un atac). També els clients dels serveis de computació al núvol haurien de monitoritzar tots els paràmetres que considerin oportuns, incloent-hi els relacionats amb la qualitat del servei.

8) Els *bugs* que pugui tenir el sistema de virtualització utilitzat pel proveïdor poden comprometre l'aïllament lògic que hi ha d'haver entre les màquines virtuals desplegades a un mateix servidor físic.

9) S'ha d'utilitzar l'encriptació sempre que es pugui i ha de quedar establert de qui és la custòdia de les claus: el client o el proveïdor.

10) Els recursos virtuals (com ara els discos virtuals) que es deixen d'utilitzar, especialment en el cas que no disposin de xifrat, s'han d'esborrar de manera segura per tal que no es pugui accedir al seu contingut per part d'algun altre client del proveïdor quan es torni a assignar.

11) Hem de tenir present quina és la legislació del país on resideixen les dades, i segons aquesta, tenir clar a qui pertanyen legalment les dades. Convé considerar quin paper pot adoptar el seu govern i altres institucions en cas d'incidents.

12) Tenim algun requeriment legal (o d'un altre tipus) que ens obligui a restringir la localització geogràfica de les dades quan estan allotjades al núvol?

13) En cas de desastre, quin pla de recuperació té establert el proveïdor?

14) Cal establir un pla d'actuació en cas que el proveïdor cessi la seva activitat.

9. Seguretat de la web

La tecnologia en què es basa la web ha anat evolucionant de manera contínua des del seu naixement. Al principi, el flux de comunicació era unidireccional, del servidor al client, i el contingut era estàtic. Avui dia, els llocs webs són aplicacions interactives i complexes en què el flux d'informació és bidireccional. Utilitzem la web per realitzar moltes tasques de les quals hem arribat a dependre en el nostre dia a dia, sovint tractant amb dades de caràcter sensible (per exemple, fer transaccions bancàries, fer compres, accedir a entreteniment o accedir a cursos i estudis).

En les aplicacions web són comuns els problemes de seguretat relacionats amb el procés d'autenticació, és a dir, l'inici de la sessió d'usuari (contrasenyes insegures o atacs de força bruta, per exemple). També són habituals els que tenen a veure amb el control d'accessos, que són els que es presenten quan la funcionalitat o les dades amb què treballa l'aplicació no es protegeixen de manera correcta, i un atacant pot realitzar accions amb privilegis que no li pertoquen.

En aquest apartat parlarem detalladament d'aquestes vulnerabilitats i de què es pot fer per evitar-les i protegir-se dels atacs que les tenen com a objectiu.

9.1. El protocol HTTP

L'**HTTP** (*hypertext transfer protocol*) és el principal protocol de xarxa que fa servir la web, el qual permet a un client accedir a recursos allotjats en un servidor (per exemple, documents o imatges) per mitjà d'una sèrie d'intercanvis de missatges (anomenats transaccions).

Un missatge HTTP consisteix en una o diverses línies de text que s'intercanvien el client i el servidor, amb el format següent:

1) una o més capçaleres
2) una línia en blanc que fa de separador i, opcionalment,
3) un cos del missatge.

Hi ha dos tipus de missatges HTTP: les peticions i les respostes.

9.1.1. Petició HTTP

Les peticions HTTP són els **missatges que envia el client** cap al servidor web. Un exemple concret d'una petició HTTP seria el que es mostra a continuació:

```
GET /cv/assignatura?asid=290 HTTP/1.1
Host: llocweb.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: text/html,application/xhtml+xml,application/xml,image/webp,*/*
Accept-Language: en-US,en,es-ES,es
Accept-Encoding: gzip, deflate
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Cookie: SessionId=06b4i8e850lf4m09a3a5m6ql9i
```

La primera línia de la petició HTTP conté tres components, que són els següents:

1) Mètode: és el tipus d'acció que s'ha de dur a terme amb un recurs determinat. El mètode més habitual és `GET`, que serveix per recuperar un recurs des d'un servidor web. Els mètodes `POST` i `PUT` envien dades al servidor web per crear o modificar un recurs. Altres mètodes són: `HEAD`, `DELETE` i `OPTIONS`. En l'exemple que estem tractant, el mètode és `GET`.

2) URL del recurs: el localitzador uniforme de recursos (*universal resource locator*, URL) és un identificador únic que especifica on està ubicat el recurs al qual es vol accedir. De manera opcional, es pot concatenar a l'URL el nom d'un o diversos paràmetres amb els seus respectius valors. En l'exemple de petició HTTP d'abans, l'URL és «/cv/assignatura», amb un paràmetre inclòs anomenat «asid» amb el valor «290». El format genèric dels URL és el següent (els components entre [ ] són opcionals):

```
protocol://host[:port]/[ruta/]fitxer[?paràmetre=valor]
```

3) Versió del protocol HTTP: en l'exemple, la versió és 1.1.

Després d'aquesta primera línia de la qual hem parlat, hi ha altres capçaleres (una per línia). Podem destacar les següents:

- `Referer`: indica l'URL des d'on s'ha originat la petició (per exemple, la pàgina que contenia l'enllaç on ha clicat l'usuari per arribar al recurs).
- `User-agent`: dona informació sobre el navegador del client.
- `Host`: indica el nom del servidor que allotja el recurs.
- `Cookie`: especifica una galeta (de les quals parlarem més endavant).

9.1.2. Resposta HTTP

A continuació es mostra un exemple de resposta HTTP (el **missatge que envia el servidor** cap al client després de rebre una petició HTTP).

```
HTTP/1.1 200 OK
```

```
Date: Sun, 02 Feb 2020 12:53:20 GMT
Server: Apache
Set-Cookie: SessionId=06b4i8e850lf4m09a3a5m6ql9i
Accept-Ranges: bytes
Content-Length: 89
Pragma: no-cache
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<html>
<header><title>Assignatura XYZ</title></header>
<body>
Benvinguts a XYZ
</body>
</html>
```

La primera línia conté també tres components, com el cas de la petició:

1) Versió del protocol HTTP: en l'exemple, la versió és 1.1.

2) Codi d'estat: aquest és un valor que es correspon amb el resultat de la petició (per exemple, el «200» de l'exemple vol dir que la petició s'ha pogut servir amb èxit).

3) Cadena complementària que descriu el codi d'estat: normalment el navegador no la processa i simplement la descarta. En l'exemple és `OK`.

Després de la primera, vegem la resta de capçaleres (una per línia), de les quals destaquem algunes d'interès:

- `Server`: és el *banner* del programari utilitzat com a servidor web. En aquest cas, Apache.
- `Set-cookie`: especifica una galeta.
- `Pragma`: és una directriu perquè el navegador no guardi la resposta a la *cache*.
- `Content-length`: és la longitud de la resposta en bytes.

A continuació de les capçaleres, i separat per una línia en blanc, en el cos de la resposta, es retorna al client el document sol·licitat. En l'exemple, veiem que es tracta d'un document HTML.

9.1.3. Galetes (*cookies*)

Les **galetes** (*cookies*) són arxius que contenen informació que generen els llocs web que un usuari visita i que s'emmagatzemen localment a l'ordinador client.

El seu objectiu és guardar informació per millorar l'experiència de navegació i oferir més funcionalitats, com ara:

- **Gestió de les sessions d'usuari**: mitjançant les galetes, els llocs web saben si un usuari manté una sessió oberta i amb quin compte. Les galetes representen el mètode més habitual per gestionar les sessions d'usuari d'un lloc web.

- **Personalització**: mantenen informació sobre l'usuari per tal que el lloc web pugui presentar un contingut dinàmic més rellevant (per exemple, guardant les seves preferències de l'aplicació o portant un carret de compra amb productes).

- ***Tracking***: es fa servir per monitoritzar els hàbits de navegació.

Les galetes són elements importants per a la seguretat perquè si un atacant té accés a les galetes d'un usuari, pot aconseguir accedir amb les seves credencials al lloc web que les ha generat (per exemple, mitjançant atacs de tipus *cross-site scripting* i *cross-site request forgery*, dels quals parlarem més endavant).

9.1.4. HTTPS

HTTPS és una extensió del protocol HTTP que permet xifrar els missatges entre el client i el servidor utilitzant els protocols criptogràfics de *transport layer security* (TLS). Facilita la creació d'un canal segur sobre un d'intrínsecament insegur, com és internet, i evita l'espionatge de les comunicacions i els atacs d'intermediari.

9.2. Model d'objectes del document (DOM)

El **model d'objectes del document** (*document object model*, DOM) és un estàndard per llegir, canviar, afegir i esborrar elements d'un document HTML.

Aquest model, constituït en forma d'arbre d'objectes, és el que fan servir els navegadors per renderitzar un document HTML amb tots els seus elements.

DOM és una representació abstracta d'un document HTML (i també el seu URL i les galetes) a la qual els *scripts* que s'executen al navegador poden accedir per consultar el contingut i també modificar-lo utilitzant una API. DOM també inclou un model d'esdeveniments, de tal manera que es poden gestionar programàticament les accions que duu a terme l'usuari, com ara l'enviament d'informació per mitjà d'un formulari, el moviment del ratolí, etc.

9.3. Política del mateix origen

Imaginem que estem accedint amb el navegador al lloc web del nostre banc per consultar el saldo del compte corrent que hi tenim. Podria ser que un *script* d'una pàgina web elaborada per un atacant que tenim oberta en una altra pestanya, pogués accedir a les dades i a la funcionalitat del lloc web del banc, i fer operacions de manera subreptícia com, per exemple, robar les nostres dades financeres o, fins i tot, fer una transferència bancària?

Per evitar que un *script* d'una pàgina web maliciosa accedeixi a les dades i funcionalitats d'una altra pàgina, els navegadors implementen una política de seguretat anomenada política del mateix origen (*same origin policy*, SOP). Segons aquesta, el navegador web només permet que els scripts de la pàgina X accedeixin a dades de la pàgina Y si ambdues tenen el mateix origen. Tenir el mateix origen vol dir que el protocol, el port i el *host* dels seus respectius URL han de ser els mateixos.

Per exemple, aquests dos URL tenen el mateix origen (el protocol, el *host* i el port són iguals):

`https://llocweb.edu/assig1` i `https://llocweb.edu/assig2`

En canvi, en els exemples següents, no tenen el mateix origen:

`http://llocweb.edu/assig3` i `https://llocweb.edu/assig4` (el protocol no és el mateix: un és HTTP i l'altre HTTPS).

`http://llocweb.edu:8080/assig5` i `http://llocweb.edu/assig6` (el port és diferent).

9.4. Components de les aplicacions web

Les aplicacions en si solen estar formades pels components següents:

1) Autenticació: els llocs web habitualment utilitzen noms d'usuaris i contrasenyes com a mecanisme d'autenticació. Quan es requereix un grau més alt de seguretat, es poden fer servir mecanismes complementaris. Aquí els atacants poden identificar noms d'usuaris, esbrinar contrasenyes, fer atacs de força bruta o utilitzar credencials per defecte, per exemple.

2) Gestió de les sessions d'usuari: el protocol HTTP no gestiona l'estat (és *stateless*), això vol dir que les transaccions entre el client i el servidor són independents entre si. Com associar un conjunt de transaccions amb una sessió especifica d'un usuari concret? Per a cada usuari que accedeix a una aplicació

web, aquesta crea i manté una sessió. Una sessió es representarà al servidor com una estructura de dades que conté en cada moment l'estat de la interacció que està realitzant l'usuari amb l'aplicació web. Per identificar cada sessió, el servidor genera un *token*, que és simplement una cadena de text, i el fa arribar al navegador del client. A partir de llavors, el *token* s'envia automàticament de tornada cap al servidor amb totes les peticions HTTP que faci aquest usuari en aquesta sessió.

Les galetes, com s'ha comentat prèviament, són el mètode més àmpliament utilitzat per transmetre els *tokens* de sessió, encara que hi ha altres mecanismes, com ara els paràmetres a l'URL de cada petició o els camps de formularis amagats (*hidden forms*).

En aquesta part de l'aplicació web, l'atacant intenta comprometre el *token* de sessió. Si acaba tenint-hi accés, pot fer-se passar com a usuari víctima i utilitzar l'aplicació com si s'hagués autenticat ell mateix.

3) Control d'accés: en aquesta part de la lògica de l'aplicació web, es gestiona la manera com els usuaris poden accedir a les funcionalitats.

9.5. Arquitectura de les aplicacions web

Les aplicacions web solen estar desplegades basades en una arquitectura de tres capes:

1) Capa de presentació (*front-end*): un o diversos servidors web s'encarreguen de servir el contingut estàtic directament al client. Dirigeix les peticions de contingut dinàmic cap a la capa de servidors d'aplicacions, i un cop rep la resposta, la serveix al client.

2) Capa de negoci (*middle-tier*): un o diversos servidors d'aplicacions, que contenen la lògica de negoci, generen el contingut dinàmic accedint a les dades del *back-end*. Un cop generat el contingut, s'envia al *front-end*.

3) Capa de dades (*back-end*): un o diversos servidors de bases de dades, on hi ha emmagatzemades les dades amb què treballen les aplicacions.

9.6. L'entrada d'usuari

Com ja s'ha comentat, un dels problemes principals de la seguretat en els sistemes TIC és que els usuaris poden introduir dades de manera arbitrària a les aplicacions, fet que pot resultar en un comportament no previst prèviament pels dissenyadors. Hem de tenir present, doncs, quan es desenvolupa una aplicació web o es fa un estudi sobre la seva seguretat, que tota entrada d'usuari és potencialment perillosa.

Com podem afrontar aquest fet? Com s'ha de gestionar des de l'aplicació l'entrada de dades que fa l'usuari per minimitzar el risc de seguretat que representa? Hi ha diferents aproximacions per fer-ho, entre les quals destaquem les següents:

- **Rebutjar els elements nocius coneguts**: aquesta tècnica consisteix a rebutjar les dades introduïdes per l'usuari en el cas que estiguin recollides en una «llista negra», i si no ho estan, acceptar-les.

- **Acceptar els elements coneguts no nocius**: de manera anàloga, aquí s'accepten les dades introduïdes per l'usuari només en el cas que estiguin recollides en una «llista blanca», i si no ho estan, es rebutgen. És un sistema més segur que l'anterior per ser més restrictiu.

- **Higienització** (*sanitization*): en comptes de rebutjar les dades d'usuari, amb aquesta tècnica es manipula l'entrada d'usuari per modificar o eliminar les parts que es considerin nocives.

9.7. Mecanismes de defensa

Perquè l'administrador pugui articular els mecanismes de defensa adients per a les aplicacions web, s'han de tenir en compte el principis d'acció següents:

1) Assegurar-se que els usuaris tinguin, exclusivament, accés a la funcionalitat a què estan autoritzats (autenticació, gestió de les sessions, control d'accessos).

2) Controlar i gestionar les dades introduïdes per l'usuari per prevenir que puguin causar que l'aplicació tingui un comportament no desitjat.

3) No revelar més informació sobre el nostre sistema de la qual sigui necessària (i que es pugui utilitzar per part d'un atacant). Convé no publicar, per exemple, el diferent programari que s'està utilitzant al servidor, les seves versions, les funcionalitats instal·lades, el llenguatge amb el qual estan desenvolupades les aplicacions web, etc.

Per exemple, es pot configurar el servidor web Apache HTTP Server perquè no mostri la seva versió a la resposta d'una petició HTTP. En comptes de tornar aquesta capçalera mostrada a continuació, que informa de quin programari de servidor web està instal·lat, quina versió, el sistema operatiu, i la versió d'altres components del sistema:

```
Server: Apache/2.4.41 (Ubuntu) OpenSSL/1.1.1c
```

Seria més convenient configurar-lo perquè retorni aquesta altra, amb molta menys informació:

```
Server: Apache
```

També és important que les aplicacions gestionin de manera controlada els errors que es puguin generar, mostrant a l'usuari un missatge degudament format i no els missatges d'error generats pel servidor web o el servidor d'aplicacions (que poden donar molta informació a un atacant).

4) Monitoritzar l'activitat i gestionar les alertes. S'han de comprovar els *logs* de l'aplicació, del servidor d'aplicacions i del servidor web per detectar comportaments anòmals potencialment indicadors d'algun problema, com per exemple:

- Moltes peticions realitzades des d'una adreça o un rang d'adreces IP específics.
- Moltes peticions del mateix tipus realitzades des de diverses adreces IP.
- Peticions que, pel seu contingut, es puguin considerar com a malicioses (cadenes o patrons que identifiquen un atac).

5) Configurar un tallafocs d'aplicació web (*web application firewall*, WAF). Aquest tipus de tallafocs és capaç d'analitzar el trànsit HTTP, detectar el que sigui maliciós i bloquejar-lo abans que arribi al servidor web.

6) Reacció en cas d'atac. Es poden prendre diferents mesures reactives en cas de detectar un atac:

- Bloqueig de les IP de les quals prové l'atac.
- Baixada progressiva de la velocitat de resposta a les peticions HTTP malicioses.
- Finalització immediata de les sessions d'usuari sospitoses.

9.8. Injecció de codi SQL

Les aplicacions web accedeixen a bases de dades per emmagatzemar i recuperar informació que serveix per generar la sortida dinàmica que es presenta a l'usuari.

Un **atac d'injecció SQL** consisteix a incloure codi maliciós a les sentències SQL que utilitza un lloc web utilitzant un mecanisme d'entrada de dades d'usuari (normalment un formulari).

Per exemple, tenim una pàgina web a on es pot buscar en un catàleg els productes pel seu identificador numèric, que l'usuari ha d'introduir en un formulari. El codi que gestiona l'entrada del formulari és el següent:

```
producteIDtxt=getRequestString ("ProducteID")
consultaSQL="SELECT * FROM Productes WHERE ProducteID = " + producteIDtxt
```

Si al formulari s'introdueix un número (per exemple, 8990, com es mostra a continuació), que és el tipus d'entrada que espera l'aplicació, la sentència SQL que es formaria seria la següent:

```
Identificador de producte: 8990
```

```
SELECT * FROM Productes WHERE ProducteID = 8990;
```

No obstant això, si un atacant introdueix, en comptes d'un valor numèric, la cadena següent, la sentència SQL manipulada indegudament seria la que es mostra a continuació. Quan s'executi, s'esborrarà la taula «Productes» de la base de dades:

```
Identificador de producte: 8990; DROP TABLE Productes
SELECT * FROM Productes WHERE ProducteID = 8990; DROP TABLE Productes;
```

9.9. *Cross-site scripting*

Cross-site scripting (XSS) és un atac que consisteix a injectar codi maliciós a una pàgina web lícita (però vulnerable) visitada per un usuari.

Aquest tipus de vulnerabilitat també radica en el fet que l'aplicació web no controli degudament l'entrada de dades per part de l'usuari. A partir d'aquesta entrada, es genera la sortida que s'envia a l'usuari i que es mostrarà mitjançant el seu navegador. Si un atacant construeix una entrada de tal manera que la sortida generada sigui un codi interpretable per un navegador web, aquest codi s'executarà al navegador de l'usuari víctima amb els permisos de la sessió que tingui oberta en aquell moment.

En aquest tipus de vulnerabilitat, l'atacant és capaç, amb el codi maliciós injectat d'aquesta manera, de **capturar el *token*** de sessió de l'usuari víctima. Així, el resultat de l'atac pot ser el robatori de dades privades de l'usuari o la modificació il·lícita del contingut mostrat per l'aplicació web (*defacing*), entre d'altres.

Perquè l'atac tingui èxit, el codi maliciós ha d'estar injectat a la pàgina web lícita i no pot tenir un URL qualsevol, a causa de la restricció imposada als navegadors per la política del mateix origen.

Els tipus d'atacs XSS més importants que podem trobar són els següents:

1) XSS no persistent (altrament dit *reflectit* o *indirecte*): el codi maliciós està contingut a la petició HTTP.

Exemple: tenim un **URL** lícit d'una pàgina web vulnerable, que accepta entrades d'usuari arbitràries per mitjà d'un paràmetre. En el cas d'ús lícit que es mostra a continuació, l'URL té el valor `Seguretat Web` pel paràmetre `busca`:

```
URL lícit:
https://web-vulnerable.com/busca?text=Seguretat+Web

HTML generat quan s'accedeix a l'URL:
... <body> Buscant: Seguretat Web </body> ...
```

Ara bé, si un atacant genera l'URL següent i aconsegueix que l'usuari víctima hi accedeixi amb el seu navegador (per mitjà de l'enginyeria social, per exemple), executarà el codi maliciós contingut entre les etiquetes `<script>` i `</script>` amb els privilegis de la sessió que tingui oberta:

```
URL il·lícit, generat per un atacant:
https://web-vulnerable.com/busca?text=<script>codi_maliciós</script>
```

```
HTML generat quan s'accedeix a l'URL:
...<body><script> /* Codi maliciós */ </script></body>...
```

2) XSS persistent (també anomenat *emmagatzemat* o *directe*): el codi maliciós està contingut a la base de dades de l'aplicació web. Poden ser vulnerables a aquest atac les aplicacions web que emmagatzemen dades que introdueixen els usuaris per tal de ser recuperades en un futur i que, per tant, es guarden en bases de dades (per exemple, els missatges d'un fòrum d'internet o les característiques d'un producte d'una botiga en línia).

Un exemple d'atac d'aquest tipus seria el següent: un atacant deixa un missatge de text en una web que té un fòrum en línia, per mitjà d'un formulari. En el text inclou codi maliciós, entre dues etiquetes `<script>` i `</script>`. En cas que l'aplicació del fòrum sigui vulnerable, tots els usuaris que llegeixin el missatge que ha deixat l'atacant executaran el codi maliciós que conté al seu navegador.

A diferència dels XSS reflectits, no cal convèncer l'usuari perquè accedeixi a un URL determinat, i per això, són vulnerabilitats més perilloses.

9.10. *Cross-site request forgery*

Cross-site request forgery (CSRF) és un atac que consisteix a fer que un usuari accedeixi a un lloc web maliciós controlat per l'atacant i innocu en aparença, que, quan el visita, fa que el navegador de la víctima enviï una petició a una aplicació web vulnerable per dur-hi a terme una acció determinada sense coneixement de l'usuari objecte de l'atac.

Perquè aquest tingui èxit, l'usuari ha de tenir una sessió oberta en l'aplicació web vulnerable.

Per exemple, suposem el cas d'una aplicació web d'una botiga en línia en què l'URL següent, només accessible per als usuaris amb privilegis d'administrador, estableix el preu que té un producte (en el cas que es mostra, accedir a aquest URL amb una sessió d'administrador establiria el preu de 30 a l'ítem número 900:

```
https://web-vulnerable.com/botiga/admin/edita?item=900&preu=30
```

Si mentre un administrador té una sessió oberta a aquesta web, se li fa visitar (mitjançant un atac d'enginyeria social) una altra pàgina web creada per l'atacant (`https://web-atacant.com`), que contingui el codi HTML següent, el preu del producte 900 passarà a ser 0:

```
...
<img src="https://web-vulnerable.com/botiga/admin/edita?item=900&?preu=0">
...
```

Ens podem preguntar: no hauria d'evitar-se aquest atac amb la política del mateix origen? La resposta és que no, perquè des del lloc web atacant només s'envia una petició al lloc web vulnerable, no s'accedeix ni es processa el seu contingut (que és el que impediria la política del mateix origen).

10. Aspectes legals. Marc jurídic penal i extrapenal. El «delicte informàtic»

El «delicte informàtic» no apareix explícitament definit en l'actual codi penal (1995) ni a les reformes posteriors que se n'han fet i, per tant, no es podrà parlar de «delicte informàtic» pròpiament dit, sinó de delictes fets amb el concurs de la informàtica o les noves tecnologies, en els quals l'ordinador s'erigeix com a mitjà d'execució del delicte o com a objectiu d'aquesta activitat (per exemple, una intrusió en un sistema informàtic). L'objectiu d'aquest apartat no és alliçonar els administradors o directors dels departaments de TIC, sinó solament fer-los conèixer les responsabilitats en què poden incórrer a causa del seu treball i, com a fita principal, dotar-los de mecanismes que, en cas d'accions delictives que tenen per objecte els sistemes que administren, dels quals són responsables, els permetin denunciar els delictes de què han estat víctimes i sol·licitar les actuacions legals pertinents.

D'altra banda, tampoc no es pretén fer un recull excessivament generós en llenguatge jurídic, ni aprofundir en possibles sentències relacionades amb els delictes que s'explicaran en aquest mòdul. La legislació actual encara presenta buits pel que fa als mal anomenats «delictes informàtics», de manera que solament s'oferiran directrius bàsiques, més aviat relacionades amb el sentit comú i els articles del codi penal (entre altres normes), que no pas amb la complexa normativa que es va generant entorn d'aquesta nova problemàtica.

El vessant tecnològic o científic dels estudis d'enginyeria sovint deixa de banda el vessant social de l'aplicació dels avenços que es van produint en aquestes disciplines. Consegüentment, l'administrador d'un sistema pot ser molt competent en el treball tècnic, però és possible que tingui molts dubtes a l'hora de tractar problemes com els següents:

- Si el meu cap em demana que li mostri el contingut de la bústia de correu personal d'un treballador, tinc l'obligació de fer-ho?

- S'ha produït un accés no autoritzat al servidor i els intrusos han modificat la pàgina web del departament. Aquest fet és denunciable? A qui ho he de denunciar?

- El servidor emmagatzema dades de caràcter personal. S'han de protegir amb algunes mesures de seguretat determinades?

- És legal la utilització d'escàners (entesos com a eines d'administració de sistemes)?

- Puc penjar a internet una pàgina web amb les fotografies i logotips del meu grup de música preferit?

- Com puc denunciar l'ús de còpies no autoritzades de programari?

- Puc fer servir eines criptogràfiques per protegir la informació?

- Els administradors de sistemes d'hostatge són responsables dels continguts que allotgen les pàgines web dels clients?

En aquest apartat intentarem orientar-vos en relació amb els dubtes que s'han expressat, si bé cal ser conscient que no hi ha una única línia d'actuació i que les particularitats de cada cas fan que calgui ser molt prudent a l'hora d'enfrontar-se amb aquest tipus de problemes. En definitiva, cal tenir molt present que no tot allò que és tècnicament possible és legal, i que el desconeixement de les normes no exonera de responsabilitat (penal o no) el treballador informàtic.

10.1. Marc jurídic penal de les conductes il·lícites vinculades a la informàtica

En aquest subapartat s'estudiaran les sancions previstes pel codi penal (moltes vegades, penes privatives de llibertat). Com es veurà, algunes de les accions plantejades pels dubtes de l'apartat anterior poden originar responsabilitat penal. Altres, però, tindran la consideració d'extrapenals, entenent amb aquest nom, la branca de l'ordenament jurídic que conté sancions menys greus que les previstes pel dret penal (dret administratiu, dret civil, dret laboral, etc.).

10.1.1. Delictes contra la intimitat

L'article 197.1 de l'actual codi penal (a partir d'ara CP) assimila la **intercepció del correu electrònic** amb la violació de la correspondència.

Així doncs, seran constitutives de delicte les conductes següents:

- L'apoderament de papers, cartes, missatges de correu electrònic o qualsevol altre document o efectes personals.

- La intercepció de les telecomunicacions.

- La utilització d'artificis tècnics d'escolta, transmissió, gravació o reproducció de so, o de qualsevol altre senyal de comunicació.

Per ser constitutives de delicte, aquestes activitats s'han de produir sense el consentiment de l'afectat (ni autorització judicial motivada) i amb la intenció de descobrir-ne els secrets o vulnerar-ne la intimitat.

Per tant, obrir la bústia d'un correu electrònic que no sigui el nostre propi i llegir els missatges que s'hi emmagatzemen podria esdevenir una conducta constitutiva de delicte. Cal anar amb molt de compte amb aquest tipus d'accions i, com a norma general, mai no s'ha de llegir cap correu electrònic que no vagi adreçat a nosaltres mateixos.

El rerefons de la intercepció empresarial del correu electrònic gairebé sempre és el mateix: el dret de les organitzacions a controlar els seus mitjans de producció. En aquest sentit, diverses sentències que s'han dictat en els tribunals en relació amb l'ús dels mitjans de l'empresa amb finalitats personals, s'han pronunciat a favor de l'empresa, ja que s'entén que els mitjans pertanyen a l'empresa i que aquest no és un lloc adient per enviar i rebre missatges de caràcter privat (o fer altres activitats personals, com ara l'ús dels jocs que s'inclouen en els sistemes operatius).

Una manera útil per fer saber als usuaris d'una organització quins són els usos correctes dels mitjans de l'empresa, i les seves limitacions, consisteix en l'ús de contractes en els quals s'especifica, per exemple, quines obligacions i responsabilitats té un usuari d'un compte de correu electrònic. D'altra banda, també és important que els sindicats en tinguin coneixement i que, per tant, els treballadors sàpiguen que se'ls pot sotmetre a certes mesures de control, les quals, més que basar-se en l'obertura dels correus electrònics, ho haurien de fer en l'ús de controls menys lesius, com per exemple, l'estudi del nombre de bytes transmesos, entre d'altres.

10.1.2. Usurpació i cessió de dades reservades de caràcter personal

La resta d'apartats de l'article 197 CP 23 (i els articles 198, 199 i 200 CP) tipifiquen com a conductes delictives l'accés, utilització, modificació, revelació, difusió o cessió de dades reservades de caràcter personal que estiguin emmagatzemades en fitxers, suports informàtics, electrònics o telemàtics, sempre que aquestes conductes siguin fetes per persones no autoritzades (conductes anomenades, genèricament, **abusos informàtics sobre dades personals**).

Explícitament es fa esment de l'agreujant d'aquestes conductes quan les dades objecte del delicte són de caràcter personal que revelin ideologia, religió, creences, salut, origen racial o vida sexual. Altres agreujants que cal tenir en compte es produeixen quan la víctima és un menor d'edat o incapaç, o la persona que comet el delicte és el responsable dels fitxers que hi estan involucrats. Mereix una especial consideració l'article 199.2, en el qual es castiga la conducta del professional que, incomplint l'obligació de reserva, divulga els secrets d'una altra persona.

10.1.3. Delicte de frau informàtic

L'article 248.2 CP castiga la conducta de qui, emprant qualsevol manipulació informàtica, aconsegueixi la **transferència no consentida** de qualsevol actiu patrimonial, amb ànim de lucre i perjudici sobre un tercer. La Llei 15/2003, per la qual es va aprovar la reforma del codi penal de l'any 1995, introdueix el càstig per a les conductes preparatòries per a la comissió de delictes de frau informàtic. Així doncs, també es castiga la fabricació, facilitació o la mera possessió de programari específic destinat a la comissió del delicte de frau informàtic.

10.1.4. Delicte d'ús abusiu d'equipaments

L'article 256 CP castiga l'ús de qualsevol equipament terminal de telecomunicacions sense el consentiment del seu titular, sempre que li ocasioni un perjudici superior a quantre-cents euros. Aquesta quantitat va ser establerta per la Llei 15/2003.

10.1.5. Delicte de danys

Segons l'article 264.2 CP, el delicte de danys consisteix en la **destrucció, alteració, inutilització o qualsevol altra modalitat** que impliqui el dany de dades, programari o documents electrònics emmagatzemats en xarxes, suports o sistemes informàtics.

El delicte de danys és un dels delictes «informàtics» més freqüents i sovint té repercussions econòmiques molt importants a les organitzacions afectades.

Els danys produïts en un sistema informàtic s'han de poder valorar i és essencial adjuntar una valoració d'aquests danys quan es denuncia l'acció delictiva davant d'un cos policial. La valoració dels danys és un procés complex de dur a terme i pot abastar diferents aspectes: cost de restauració d'una pàgina web, pèrdues en concepte de publicitat no emesa (lucre cessant) o per serveis que no s'han pogut prestar, entre d'altres.

10.1.6. Delictes contra la propietat intel·lectual

Segons l'article 270 CP, les conductes relatives als delictes contra la propietat intel·lectual són aquelles en què **es reprodueix, plagia, distribueix o comunica públicament**, tant d'una manera total com parcial, una obra literària, artística o científica sense l'autorització dels titulars dels drets de propietat intel·lectual de l'obra.

Aquestes condicions s'apliquen independentment del suport en què s'hagi enregistrat l'obra, siguin textos, programari, vídeos, sons, gràfics o qualsevol altre fitxer relacionat. És a dir, els delictes relatius a la venda, distribució o fabricació de còpies no autoritzades de programari són delictes contra la propietat intel·lectual.

Vegem alguns exemples de delictes contra la propietat intel·lectual:

- Reproducció íntegra de programari i venda al marge dels drets de llicència.
- Instal·lació de còpies no autoritzades de programari en un ordinador en el moment de la seva compra.
- Publicació del codi font de programari, contingut piratejat (programari, música, llibres, pel·lícules) a internet, al marge dels drets de llicència d'aquestes obres.
- Utilització d'una llicència de programari per només un sol ordinador per donar servei a tota la xarxa.
- Trencament dels mecanismes de protecció que permeten el funcionament correcte del programari (aquestes tècniques reben el nom genèric de *cracking*).

El mateix article 270 CP preveu penes per a qui faci circular o disposi de qualsevol mitjà específicament dissenyat per anul·lar qualsevol dispositiu tècnic de protecció del programari.

Amb la reforma de la Llei 15/2003, els cossos policials poden actuar d'ofici en la persecució d'aquest tipus de delicte. D'altra banda, un particular, atès que normalment no disposa dels drets de propietat intel·lectual, no pot denunciar directament aquests tipus de delicte. No obstant això, és possible fer-ho de manera indirecta per mitjà d'organitzacions com ara la Business Software Alliance (BSA).

Pel que fa a la creació de programari, també hi ha algunes consideracions que cal tenir en compte. Segons el tipus de contracte al qual estigui subjecte el treballador, el programari que desenvolupi per a una organització pertany a l'empresa i, en conseqüència, si el treballador abandona l'organització, no es pot emportar el programari que ha creat en el seu antic lloc de treball. Com en el cas de la utilització del correu electrònic, seria recomanable que el contracte de treball especifiqués aquesta qüestió.

10.1.7. Delicte de revelació de secrets d'empresa

Segons l'article 278.1 CP, fa revelació de secrets d'empresa qui, amb la finalitat de descobrir un secret d'empresa, intercepti qualsevol tipus de telecomunicació o utilitzi artificis tècnics d'escolta, transmissió, gravació o enregistrament del so, imatge o de qualsevol altre senyal de comunicació.

10.1.8. Delicte de defraudació dels interessos econòmics dels prestadors de serveis

La defraudació dels interessos econòmics dels prestadors de serveis és un nou delicte, introduït arran de la reforma 15/2003 del codi penal. L'article 286 CP conté **quatre modalitats de comissió**:

1) Es castiga la facilitació de l'accés «intel·ligible» a serveis de radiodifusió sonora o televisiva, prestats a distància per via electrònica, mitjançant la facilitació, importació, distribució, possessió de programes o equipaments informàtics, destinats a fer possible l'esmentat accés. Aquesta modalitat inclou la instal·lació, manteniment o substitució d'aquests equipaments amb finalitats comercials.

2) Es castiga l'alteració o duplicació del número d'identificació de l'equip de telecomunicacions, amb ànim de lucre.

3) Es castiga la facilitació de l'esmentat accés a una pluralitat de persones per mitjà de qualsevol publicació pública, encara que sigui sense ànim de lucre.

4) Finalment, també es castiga la utilització dels equipaments o programaris que permeten l'accés, i també la utilització d'equipaments alterats, independentment de la quantia de la defraudació.

10.1.9. Altres delictes

A més dels delictes que hem descrit, és evident que molts d'altres també es poden dur a terme amb el concurs de la tecnologia:

- Amenaces i coaccions (per xats o mitjançant el correu electrònic).
- Estafes electròniques.
- Falsedat documental (alteracions i simulacions de documents públics o privats).
- Difusió de pornografia infantil a internet.

En relació amb aquest últim delicte (article 189.1 CP), la Llei 15/2003 ha ampliat notablement el tipus delictiu. Així, la mera possessió (encara que no estigui destinada a la venda) de pornografia infantil ja està castigada (la difusió, la creació i la venda ja ho estaven). A més, s'introdueixen certs agreujants, com per exemple, la utilització de menors de tretze anys, entre d'altres. Així mateix,

també es castiga la producció, venda, distribució i exhibició, de material en què, tot i que no hi apareguin directament menors d'edat, s'hagi modificat la veu o la imatge amb la finalitat que el contingut sigui relatiu a la pornografia infantil.

Si un sistema informàtic és víctima de qualsevol d'aquests delictes, o, per exemple, es descobreix que el sistema és utilitzat com a plataforma de distribució de còpies de programari no autoritzades o de pornografia infantil, s'ha de denunciar immediatament a la comissaria de policia més pròxima, tenint en compte el protocol d'actuació següent:

1) Adjunció dels fitxers *log* (locals) relacionats amb el delicte comès.

2) En cas que s'hagi produït un delicte de danys, cal adjuntar una valoració dels danys ocasionats.

3) Actuar amb rapidesa (els proveïdors d'internet no emmagatzemen indefinidament els fitxers *log* dels seus servidors).

4) En cas que aquesta acció delictiva s'hagi produït per correu electrònic, cal adjuntar les capçaleres completes del correu rebut.

5) Si l'administrador ho considera necessari (per exemple, descobreix pornografia infantil en un servidor de la seva responsabilitat), pot clonar el disc dur del servidor per preservar l'evidència digital i reinstal·lar el sistema per evitar que el delicte es continuï produint.

10.1.10. Ús d'eines de seguretat

Pel que fa a les diverses eines de seguretat disponibles:

- **Escàners de xarxa o de sistema**: tot i la possibilitat de dotar els escàners d'un ús maliciós, els seus beneficis són evidents quant a les tasques que ha de fer l'administrador. La fiabilitat d'un sistema informàtic no es pot basar en la ignorància dels defectes que presenta i, per tant, els escàners esdevenen eines de gran valor en mans dels administradors. Ara bé, des del punt de vista legal, es poden fer servir? No hi ha cap llei en contra de l'ús dels escàners, si bé s'ha generat una interessant polèmica al seu entorn. Algunes opinions consideren que l'ús dels escàners és equivalent a anar a un domicili particular i fer servir la força per obrir la porta. D'altres creuen que pel sol fet de tenir una ubicació a internet, ja es dona el consentiment implícit per «escanejar» la localització.

- **Eines criptogràfiques**: pel que fa a la criptografia, tampoc no hi ha cap llei que en prohibeixi l'ús al nostre país. Segons l'article 52 de la Llei general de telecomunicacions, Espanya disposa d'un règim de llibertat de xifratge per protegir qualsevol dada que circuli per una xarxa. D'altra banda,

aquest mateix article deixa la porta oberta a la definició de mecanismes de control com, per exemple, l'obligació de notificar a l'Estat els algorismes criptogràfics que es facin servir.

10.2. Marc jurídic extrapenal

Hi ha un seguit de lleis que delimiten el marc jurídic que és d'aplicació en l'àmbit de les TIC: la Llei orgànica de protecció de dades personals i garantia dels drets digitals, la Llei de serveis de la societat de la informació i comerç electrònic, la Llei de signatura electrònica, la Llei de conservació de dades relatives a les comunicacions electròniques i a les xarxes públiques de comunicacions, i la Llei de propietat intel·lectual.

10.2.1. Llei orgànica de protecció de dades personals i garantia dels drets digitals

La llei orgànica 3/2018 de protecció de dades personals i garantia dels drets digitals (LOPD-GDD) té per objectiu adaptar el dret intern espanyol al reglament general de protecció de dades (RGPD).

L'RGDP és un reglament europeu (2016/679) sobre la protecció de les persones físiques respecte al tractament i lliure circulació de les dades personals. Reconeix un conjunt de «drets digitals» dels ciutadans i amplia l'abast d'aplicació de la llei de protecció de dades de la Unió Europea a totes les organitzacions (sigui quina sigui la seva procedència) que tinguin dades dels seus residents. Tota entitat (pública o privada) que dugui a terme un tractament de dades personals ha de complir amb l'RGPD. Té com a objectiu unificar la legislació de tots els països membres i garantir la lliure circulació de les dades dins la UE. Va entrar en vigor el 25 de maig de 2016 i en aplicació obligatòria el 25 de maig de 2018.

A l'article 8 de la Carta dels drets fonamentals de la Unió Europea es proclama que tota persona té dret a la **protecció de les dades de caràcter personal** que li concerneixen, i que el tractament d'aquestes dades haurà de ser lleial, per a finalitats concretes i sobre la base del coneixement de la persona afectada. A més a més, tota persona té dret a accedir i rectificar aquestes dades.

La Constitució espanyola també recull aquest dret fonamental a l'article 18: «la llei limitarà l'ús de la informàtica per garantir l'honor i la intimitat personal i familiar dels ciutadans i el ple exercici dels seus drets».

L'Agència Espanyola de Protecció de Dades és l'organisme públic encarregat de vetllar pel compliment de l'RGPD, salvaguardar els drets dels ciutadans, i informar i respondre a consultes.

L'RGDP estableix una sèrie de conceptes:

- **Dades personals**: tota la informació sobre una persona física identificada o identificable.

- **Persona física identificable**: tota persona de la qual es pugui obtenir la seva identitat mitjançant un identificador (un nom, un número d'identificació, dades de localització, un identificador en línia, o un o diversos elements propis de la identitat física, fisiològica, genètica, psíquica, econòmica, cultural o social).

- **Tractament**: qualsevol operació realitzada sobre les dades personals.

- **Fitxer**: conjunt estructurat de dades personals, en suport informàtic o en paper, accessible segons uns criteris determinats.

- **Responsable del tractament**: persona física o jurídica, autoritat pública, servei o un altre organisme que, sol o juntament amb d'altres, determina les finalitats i mitjans del tractament.

- **Encarregat del tractament**: persona física o jurídica, autoritat pública, servei o un altre organisme que tracta les dades personals per compte del responsable del tractament (per exemple: els serveis externalitzats contractats per les empreses per fer feines de comptabilitat, nòmines, gestió de la documentació, etc.).

- **Responsable de protecció de dades**: és la persona encarregada de la gestió, control, supervisió i seguiment del compliment de l'RGPD.

- **Treballador**: és l'usuari de les dades personals a causa de les seves obligacions laborals.

- **Interessat**: els ciutadans afectats pel tractament de les dades.

A diferència de com ho feia l'antiga LOPD, no s'estableixen nivells de seguretat preestablerts, sinó que s'apliquen en funció d'una valoració de risc de l'entitat que duu a terme el tractament de les dades personals, considerant si les dades són sensibles, la seva magnitud, l'àmbit geogràfic, i la freqüència i la durada del tractament.

Les fases del tractament de les dades personals són:

1) Sol·licitud de dades i ingesta als fitxers. S'informa a l'interessat de la naturalesa del tractament.

2) Tractament efectiu de les dades.

3) Supressió o bloqueig de les dades. Quan acaba la finalitat per la qual s'han recollit i tractat les dades, s'ha de procedir a la seva eliminació, o si hi ha alguna obligació legal per mantenir-les, al seu bloqueig.

La LOPD estableix tres tipus d'infraccions (lleus, greus i molt greus), amb sancions que poden arribar fins als vint milions d'euros o al 4 % de la facturació de l'entitat infractora.

El responsable del tractament és el garant de l'acompliment (i de demostrar-ho) dels **principis bàsics de l'RGPD**:

- **Licitud, lleialtat i transparència**: perquè el tractament sigui lícit ha de complir un dels condicionants següents:
  - es disposa del consentiment de l'interessat per al tractament de les seves dades personals per a una finalitat específica,
  - el tractament és necessari per a l'execució d'un contracte,
  - el tractament és necessari per protegir interessos vitals de l'interessat,
  - el tractament és necessari per a l'acompliment d'una obligació legal,
  - el responsable del tractament té un interès legítim prevalent sobre els interessos generals, o
  - el tractament és necessari per al compliment de les funcions de l'Administració pública.

- **Limitació de la finalitat**: el tractament de les dades ha de ser per a una finalitat determinada i explícita.

- **Minimització de les dades**: només s'han d'obtenir i tractar les dades estrictament necessàries per a la finalitat declarada. Les dades sobreres s'han de rebutjar o eliminar immediatament.

- **Exactitud**: les dades s'han de mantenir fidels i actualitzades, havent-se d'eliminar les que siguin obsoletes.

- **Limitació del període de conservació**: les dades s'han de custodiar exclusivament durant el temps requerit per a la finalitat del tractament.

- **Integritat i confidencialitat**: durant el temps que hagi de durar el tractament, s'han de dur a terme totes les mesures tècniques i organitzatives per tal de mantenir la integritat i la confidencialitat.

Tots els ciutadans disposen del dret de ser informats de manera transparent, concisa i clara, i amb caràcter previ, sobre el tractament de les dades que s'haurà de fer. I també, un cop siguin objecte del tractament, de sol·licitar l'accés, la rectificació, la supressió, la portabilitat i la limitació o l'oposició del tractament.

10.2.2. Llei de serveis de la societat de la informació i del comerç electrònic

La Llei 34/2002, d'11 de juliol, de serveis de la societat d'informació i comerç electrònic (LSSICE) representa el desenvolupament al nostre país de la directiva de la Unió Europea sobre el comerç electrònic.

L'**LSSICE** regula els serveis oferts pels operadors de telecomunicacions, els proveïdors d'accés a internet, portals i inclou, entre d'altres, el comerç electrònic.

Alguns trets característiques que la defineixen són els següents:

- Prohibició del correu electrònic no sol·licitat o no consentit (*spam*). L'incompliment d'aquesta prohibició pot comportar sancions de fins a cent- cinquanta mil euros.

- Regulació de qualsevol activitat que generi ingressos o permeti l'obtenció de beneficis econòmics (inclusió de publicitat en una pàgina web, botigues virtuals, patrocinis, etc.).

- Sancions (aplicades per l'Agència de Protecció de Dades) de fins a sis-cents mil euros per a les infraccions considerades molt greus.

- Obligatorietat de denunciar fets il·lícits i suspensió de la transmissió i allotjament de continguts il·lícits (mitjançant sol·licitud).

- Definició de les responsabilitats dels proveïdors d'internet. Per exemple, en el cas d'hostatge i *linking*, els proveïdors no tindran cap responsabilitat sobre la informació emmagatzemada, sempre que no tinguin coneixement que aquesta informació sigui il·lícita, o, si en tenen coneixement, han d'actuar amb la màxima diligència per impossibilitar l'accés o eliminar el contingut il·lícit.

10.2.3. Llei de signatura electrònica

La signatura electrònica és una matèria que queda regulada a l'Estat espanyol mitjançant el Reial decret llei 14/1999, de 17 de setembre, basat en la directiva europea que estableix el marc comunitari per a la signatura electrònica. Aquest decret llei determina l'eficàcia jurídica de la signatura digital a l'Estat espanyol i l'establiment de les condicions dels serveis de certificació.

Hi ha dos tipus diferents de signatura:

1) Signatura electrònica o digital avançada: permet identificar la persona que signa i detectar qualsevol canvi que es pugui produir de forma posterior a la signatura de les dades.

2) Signatura electrònica o digital reconeguda: consisteix en la firma electrònica avançada, basada en un certificat reconegut i generat mitjançant un dispositiu segur de creació de signatura (els prestadors de serveis de certificació). És equiparable a la signatura manuscrita.

10.2.4. Llei de conservació de *logs*

La Llei 25/2007, del 18 d'octubre, de **conservació de dades** relatives a les comunicacions electròniques i a les xarxes públiques de comunicacions, és la que té com a objectiu regular l'obligació dels operadors que presten serveis de comunicacions electròniques disponibles al públic o que exploten xarxes públiques de comunicacions.

Les dades que s'han de conservar per part dels operadors són les necessàries per identificar l'origen, el destí, la data, l'hora, la durada, el tipus, l'equip de comunicació utilitzat pels usuaris i la seva localització.

En caràcter general, els operadors hauran de conservar aquestes dades durant un període de dotze mesos des de la data en la qual s'ha produït la comunicació. Aquest es podrà reduir a sis mesos o ampliar a dos anys com permet la Directiva 2006/24/CE.

10.2.5. Llei de propietat intel·lectual

El dret a la protecció de la propietat intel·lectual és un dret fonamental reconegut a l'article 27 de la Declaració Universal dels Drets Humans.

En l'àmbit de l'Estat, la Llei de propietat intel·lectual és la normativa que protegeix els **interessos morals i materials dels creadors** de produccions científiques, literàries o artístiques de caràcter original, siguin aquestes de naturalesa

tangible (en qualsevol dels mitjans) o intangible. D'aquesta manera, pel sol fet de la seva creació, li correspon a l'autor una sèrie de drets, entre els quals, el de la seva explotació i disposició segons la seva voluntat.

La Llei recull la formulació dels drets morals (que reconeix l'autor com a propietari i a poder-se oposar a qualsevol modificació de la seva obra) i els drets patrimonials (que li permet obtenir una retribució).

11. Informàtica forense

Una vegada descrit el marc jurídic en el qual s'ajusten les conductes il·lícites relacionades amb l'ús de les tecnologies de la informació, s'estudiaran breument les metodologies de treball que es poden emprar, una vegada ha succeït l'incident, amb la finalitat d'esbrinar què ha ocorregut i qui n'ha estat el presumpte autor. Aquestes tècniques es recullen en una disciplina situada a cavall entre el marc jurídic i la tecnologia, anomenada *informàtica forense*. Les empremtes que permeten reconstruir l'execució d'un fet (el qual no ha de ser necessàriament constitutiu de delicte) estan emmagatzemades en suports digitals i s'anomenen genèricament evidències digitals.

L'**evidència digital** presenta, bàsicament, les propietats següents:

- Es pot modificar o eliminar fàcilment.
- És possible obtenir una còpia exacta d'un arxiu sense deixar cap empremta d'aquesta acció.
- L'adquisició de l'evidència pot comportar l'alteració dels suports digitals originals.

L'anàlisi forense va aparèixer a causa de la necessitat d'aportar elements rellevants en els processos judicials en què les noves tecnologies estaven presents, sigui com a objectius finals (per exemple, una intrusió amb danys en un sistema TIC) o com a mitjà (per exemple, l'enviament d'amenaces mitjançant el correu electrònic). La finalitat, en qualsevol cas, consisteix a respondre la clàssica línia argumental policial: què, quan, on, qui, com i per què.

Més precisament, es podria definir l'**anàlisi forense** com el procés d'aplicar el mètode científic als sistemes TIC amb la finalitat d'assegurar, identificar, preservar, analitzar i presentar l'evidència digital, de forma que sigui acceptada en un procés judicial.

Naturalment, la informàtica forense va més enllà dels processos judicials i, moltes vegades, els informes elaborats pels experts analistes no tindran com a objectiu final la seva presentació davant dels tribunals, sinó també l'empresa privada.

11.1. Assegurament de l'escena de l'esdeveniment

Aquesta fase únicament serà preceptiva en el decurs d'una actuació policial. No obstant això, les recomanacions que es donaran poden ser de molta utilitat per a qualsevol pèrit que hagi d'intervenir al lloc dels fets. La finalitat d'aquesta etapa consisteix a assegurar l'escena de l'esdeveniment i restringir-ne l'accés perquè ningú pugui alterar-la. Els referents policials són evidents, encara que seguir les recomanacions que ara es descriuran permetrà preservar, en qualsevol cas, l'evidència, i també facilitar-ne l'anàlisi posterior:

1) Identificar l'escena on s'ha produït el fet a investigar i establir un perímetre de seguretat.

2) Fer una llista amb els sistemes involucrats en el succés.

3) Restringir l'accés de persones i equipaments informàtics a l'interior del perímetre.

4) Fotografiar o enregistrar en vídeo l'escena del succés. També pot ser molt útil representar esquemàticament la topografia de la xarxa d'ordinadors.

5) Mantenir l'estat dels dispositius. Algunes vegades pot ser molt important fotografiar o enregistrar el contingut dels monitors en funcionament, i també la identificació i adquisició de les evidències volàtils, per exemple, l'extracció del contingut de la memòria per saber quins processos estaven en execució en aquell moment.

6) Interrompre les connexions de xarxa.

7) Comprovar i desconnectar les connexions sense fil, ja que podrien permetre connexions remotes als equips objecte d'investigació.

8) Si hi ha impressores en funcionament, permetre que acabin la impressió.

9) Anotar la data i hora del sistema abans d'apagar-lo. Aquestes dades també es poden fotografiar o enregistrar en vídeo.

10) Apagar els dispositius en funcionament, traient el cable d'alimentació o mitjançant el procediment d'apagament normal. L'expert haurà d'avaluar en cada cas quin és el mètode més adequat que ofereix més garanties de preservació de la prova.

11) Etiquetar cables i components. A més, cal tenir present que alguns dispositius requereixen un cablejat molt específic, sense el qual no serà possible analitzar l'aparell al laboratori, ja que no es podrà posar en funcionament.

Algunes vegades, l'assegurament de l'escena es produeix en el decurs d'una entrada i perquisició al lloc dels fets en presència de membres de les Forces i Cossos de Seguretat de l'Estat. En aquest cas, l'entrada comptarà amb la presència del secretari judicial, amb la qual cosa es pot fer constar en acta la data i hora del sistema, entre altres comprovacions de les quals el secretari judicial podria donar fe i, per tant, podria estalviar a l'analista alguns processos de documentació, fotografies o enregistraments de vídeo.

11.2. Identificació de l'evidència digital

S'anomena *identificació de l'evidència digital* el procés d'identificació i localització de les evidències que s'han de recollir per ser analitzades posteriorment.

Aquest procés no és tan trivial com pot semblar a primera vista ja que, tot sovint, l'expert es trobarà amb configuracions de sistemes complexos amb molts dispositius o, simplement, amb usuaris que guarden molts suports susceptibles de ser analitzats (per exemple, un particular addicte a emmagatzemar tot tipus de contingut descarregat d'internet). En conseqüència, l'analista haurà de trobar una solució de compromís entre la qualitat, la validesa de la prova i el temps de què disposa per recollir les evidències.

En primer lloc, l'expert haurà d'identificar el sistema informàtic (un únic PC, una xarxa local, una formació de discs en RAID, etc.) amb la finalitat de saber on s'emmagatzemen les evidències digitals que poden ser d'utilitat per a l'anàlisi. Aquestes poden estar en ordinadors locals, en suports com discs durs externs, en servidors remots, o fins i tot en la memòria RAM dels equipaments en funcionament. Aquest tipus d'evidències, les volàtils (en essència, aquelles que desapareixen en absència d'alimentació elèctrica), són les que haurà d'intentar preservar en primera instància, en els casos en què calgui.

També, en aquest instant, convindrà valorar la possibilitat de fer una «anàlisi en calent» a la recerca d'evidències que d'altra forma es perdrien si s'aturés el sistema. No obstant això, cal tenir present que aquesta mena d'anàlisi pot comportar la pèrdua d'altres evidències, i també la invalidació de la prova en un procediment judicial, ja que l'anàlisi en calent implica la manipulació del dispositiu original i, si no es fa amb les eines forenses adequades, es pot alterar l'evidència.

11.3. Preservació de les evidències digitals

Atesa la facilitat amb què les evidències digitals es poden modificar o eliminar, aquesta fase es converteix en la baula més crítica de tot el procediment. És evident que és del tot impossible obtenir una «instantània» completa de tot un sistema informàtic en un moment concret (la naturalesa intrínseca de les

evidències volàtils així ho determina), encara que sortosament per a l'analista, en la gran majoria de vegades, les proves determinants estan emmagatzemades en el sistema de fitxers, el qual continuarà conservant l'evidència malgrat la manca d'alimentació elèctrica.

A diferència d'altres proves (per exemple, una anàlisi biològica d'ADN), l'evidència digital es pot duplicar o clonar de manera exacta (a nivell de bits), incloent-hi els arxius ocults, eliminats i no sobreescrits, i fins i tot l'anomenat *slack space* (al qual ens referirem posteriorment), possibles particions ocultes, o l'espai no assignat del disc dur. Així, en virtut d'aquesta característica, i també com a garantia de preservació de la prova, l'analista actuant acabarà realitzant un **clon de l'evidència**, sigui a l'escena del succés o a les dependències del laboratori.

A primera vista resulta temptador ajornar la clonació dels suports informàtics al moment en què aquests arribin al laboratori (ja que és on es podrà fer el procés amb tota mena de garanties i sense presses), però això no sempre serà possible. Si, per exemple, les evidències es localitzen al servidor d'una empresa, no és possible precintar l'equipament perquè aleshores l'empresa hauria d'aturar la seva activitat. En aquests casos, és preferible aturar momentàniament l'activitat de l'empresa i obtenir un clon allà mateix, per reprendre tot seguit l'activitat empresarial, o fer una anàlisi en calent, amb els inconvenients que ja s'han explicat.

La còpia o clon s'efectuarà, normalment, sobre dispositius (*pen drives*, discs durs, etc.) aportats per l'analista. L'elecció d'un o altre mitjà dependrà de la quantitat d'informació continguda en els suports originals. Finalment, el programari o maquinari emprat per a l'obtenció del clon calcularà un valor *hash*, que haurà de ser el mateix, tant per al disc dur d'origen com per al de destí, amb la qual cosa es garantirà que el procés de còpia ha funcionat correctament.

A més de l'adquisició de l'evidència, en aquesta etapa també cal documentar qui va preservar l'evidència, on i com es va fer i quan. Tot seguit caldrà empaquetar les evidències, identificant-les de manera unívoca. Aquest procés es duu a terme embalant els paquets amb material protector que pugui protegir les evidències de cops, pluja o qualsevol altre element que pugui malmetre els suports. Aquesta fase posarà fi al transport de les evidències a un lloc segur o a les dependències del laboratori on hagin de ser analitzades. L'embalatge i el transport de les evidències és l'inici de la denominada **cadena de custòdia**, la qual permet garantir la integritat de les proves, des de la seva obtenció fins a la seva disposició a l'autoritat judicial o al laboratori on hagin de ser analitzades. La documentació de la cadena de custòdia permet saber, en qualsevol moment del procés, on han estat emmagatzemades les evidències i qui hi ha tingut accés.

11.4. Anàlisi de les evidències digitals

En aquesta fase, l'expert haurà de respondre les preguntes «policials» introduïdes a l'inici d'aquest apartat. Aquest estudi es fonamentarà, sobretot, en l'anàlisi del contingut dels arxius (dades) i de la informació sobre aquests fitxers (metadades).

Normalment no es fan anàlisis exhaustives dels suports objecte d'interès (seria una tasca inabastable), sinó que els informes pericials es limiten a respondre aquelles qüestions plantejades en els extrems de l'anàlisi.

En general, hi ha quatre categories diferents de dades que són susceptibles de ser analitzades:

1) Dades lògicament accessibles: és a dir, les dades contingudes en arxius directament accessibles. Aquesta anàlisi, no exempta de dificultats, pot no ser gaire senzilla a causa de l'enorme dificultat que hi pot haver a l'hora de discriminar la informació rellevant d'entre molts milers de fitxers, l'existència d'arxius xifrats o la presència de codi maliciós, l'execució del qual podria produir conseqüències inesperades.

2) Dades localitzades en l'anomenat *ambient data*: és a dir, aquelles dades que apareixen en localitzacions no directament visibles i que requereixen l'ús de programaris específics per ser recuperades. Un bon exemple d'aquest tipus de dades és la informació residual que pot estar en parts del disc actualment no assignades a cap arxiu, o aquella informació localitzada a l'*slack space* (espai entre el final lògic d'un fitxer i el final físic d'aquest).

3) Dades que han estat esborrades o eliminades, però que encara no han estat sobreescrites per altres fitxers i que, per tant, són susceptibles de ser recuperades emprant les eines adients a aquesta finalitat.

4) Dades ocultes mitjançant l'esteganografia, les quals són molt més difícils de detectar que els arxius xifrats.

11.5. Presentació i informe

A l'informe elaborat per l'expert es presentaran les **evidències** relacionades amb el cas, la **justificació del procediment** emprat i, el més important, les **conclusions**. Moltes vegades, l'informe serà ratificat en presència del jutge o serà lliurat a empreses i advocats. No obstant això, en cap cas els destinataris de les perícies han de disposar necessàriament de coneixements informàtics per poder comprendre l'informe en profunditat. Per tant, en general mai no s'ha d'emprar un llenguatge excessivament tècnic i, quan calgui fer-ho, caldrà afegir notes aclaridores a peu de pàgina o, fins i tot, redactar glossaris tècnics,

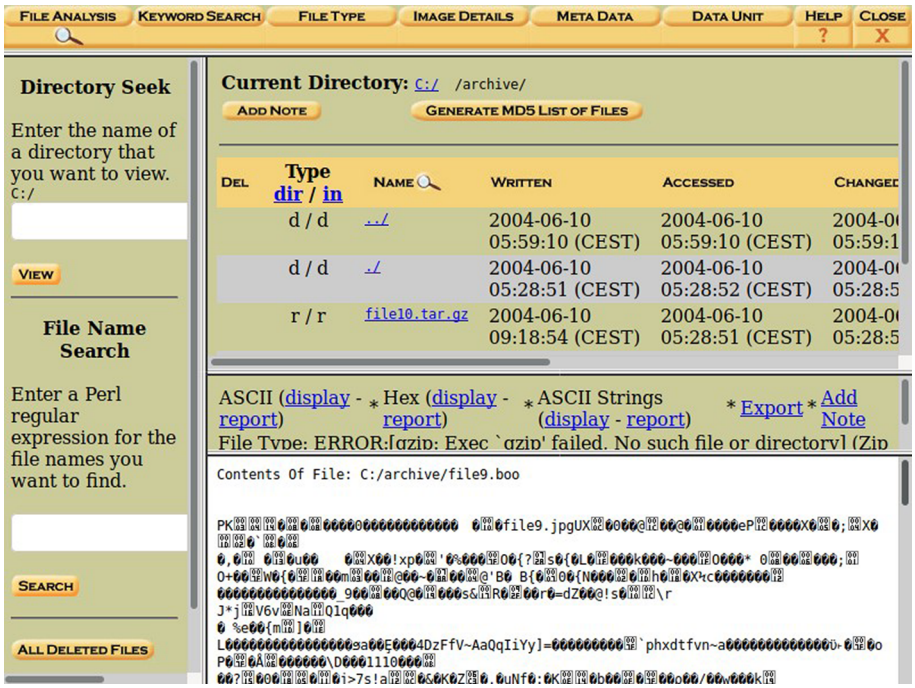
sovint afegits a l'annex de l'informe. En els casos en què els informes hagin de ser defensats davant del jutge, l'analista, a més del rigor tècnic, ha de ser prou hàbil per comunicar el resultat de l'anàlisi de forma concisa i clara.

11.6. Eines d'anàlisi forense

Per fer l'anàlisi de les evidències es poden emprar diverses eines, algunes de les quals ja s'han descrit prèviament. Possiblement, una de les més conegudes és EnCase, de codi propietari, la qual abasta, amb una interfície amigable, totes les fases de l'anàlisi forense, des de l'adquisició dels suports originals i l'anàlisi, fins a la generació automàtica de l'informe final. Parlarem d'aquest programari i d'alguns també importants a continuació.

1) Autopsy i The Sleuth Kit (TSK). **Autopsy** és el *front-end* gràfic de les aplicacions d'anàlisi forense en línia de comandes anomenades The Sleuth Kit (TSK). La figura 12 mostra la seva interfície gràfica, la qual facilita l'experiència d'ús de les eines en línia de comandes TSK. Amb aquest programari es poden realitzar diferents funcions, analitzant les imatges extretes d'ordinadors o telèfons mòbils: anàlisi del temps (establir quin ús ha tingut el dispositiu i quan s'ha produït), filtratge de fitxers utilitzant funcions *hash*, cerques per paraules clau, extracció d'informació dels navegadors web (galetes, marcadors, historial), recuperació de fitxers eliminats, extracció de metadades, localització d'indicadors d'intrusions en un sistema (amb l'eina STIX).

Figura 12. Captura de pantalla d'Autopsy



2) EnCase Forensic és una de les eines d'anàlisi forense més utilitzades arreu del món. Pot extreure dades des de múltiples dispositius pertanyents al mateix cas (ordinadors, telèfons mòbils, tauletes, receptors GPS, serveis al núvol, etc.) i analitzar-les conjuntament per trobar-hi correlacions. Disposa de gestió de

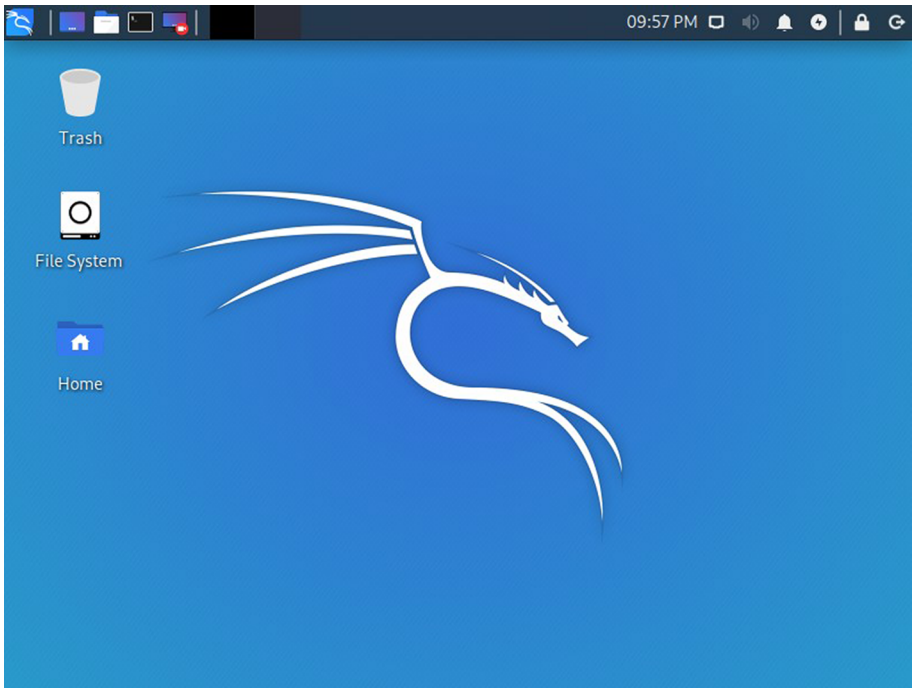
fluxos de treball que ajuda a estandarditzar els processos entre diferents investigadors i casos. Permet indexar contingut en diferents idiomes i cercar per paraules clau amb expressions regulars. Disposa d'un llenguatge de programació anomenat EnScript, que permet ampliar la funcionalitat oferta pel paquet amb desenvolupaments propis de cada usuari. El component anomenat «Evidence Processor» realitza diferents tipus de tractament sobre les imatges de dispositius a processar, com ara l'anàlisi de: entropia (útil, per exemple, per localitzar dades xifrades o codi maliciós, que sol estar ofuscat per evitar la seva detecció), correus electrònics i informació de navegadors web, entre d'altres.

3) X-Ways Forensics és una eina d'anàlisi forense per a entorns Windows. Permet fer imatges amb compressió de dispositius, accedir a diversos tipus de sistemes de fitxers, accedir a la memòria de processos que estan en execució, recuperar informació eliminada, cercar per paraules clau, analitzar màquines en remot, gestionar simultàniament múltiples investigadors i casos, gestionar els *hashs* dels fitxers, identificar fotografies, extreure metadades de diversos tipus, identificar documents protegits amb contrasenya, etc.

12. Kali Linux

Kali Linux és una distribució de Linux basada en Debian especialment creada per utilitzar-se per dur a terme diverses activitats en l'àmbit de la seguretat TIC: proves d'intrusió, auditories de seguretat, informàtica forense i enginyeria inversa, entre d'altres. És habitual fer servir la seva versió Live, que permet iniciar-se des d'un DVD o dispositiu USB sense requerir instal·lació. Com es pot apreciar a la figura 13, té un entorn gràfic d'escriptori molt simple, però conté un gran nombre de programari especialitzat.

Figura 13. Escriptori de Kali Linux



Taula 2. Algunes de les eines més útils incloses a Kali Linux

Eina	Descripció
Nmap	Escàner de xarxa.
Lynis	Eina d'auditoria de seguretat i *hardening*.
Aircrack-ng	Conjunt d'eines per auditar la seguretat de les xarxes WiFi.
THC Hydra	Eina per fer atacs de força bruta contra sistemes d'autenticació en xarxa.
WireShark	Detector de xarxa.
Metasploit Framework	Eina per fer tests de penetració.
Nessus	Escàner de vulnerabilitats.
Burp Suite Scanner	Eina d'anàlisi de seguretat per a aplicacions web.

Eina	Descripció
BeEF	Eina per comprovar la seguretat del navegador web.
SQLMap	Permet automatitzar l'explotació de vulnerabilitats d'injeccions SQL.
John The Ripper	Eina per dur a terme *cracking* de contrasenyes.
Snort	IDS i IPS.
Autopsy i The Sleuth Kit	Eina d'anàlisi forense.
King Phisher	Simulador d'atacs de *phishing*.
Nikto	Escàner de vulnerabilitats de servidors web.
Netcat	Eina per llegir i escriure dades en connexions TCP/IP.
MacChanger	Permet canviar l'adreça MAC dels dispositius de xarxa.
OWASP-ZAP	Eina d'anàlisi de seguretat per a aplicacions web.
HTTrack	Permet clonar una web.
Social Engineering Toolkit (SET)	Eina d'atac d'enginyeria social: *phishing*, *spam*, creació de dispositius USB per a *baiting*, etc.
Bettercap	Duu a terme atacs d'intermediari amb HTTP, HTTPS i altres protocols.

Resum

En aquest mòdul hem estudiat els principis bàsics de l'administració de seguretat d'un sistema informàtic i els hem relacionat amb les possibles responsabilitats que es poden derivar de la vulneració d'aquesta seguretat. Hem dividit el mòdul en els dotze apartats següents:

1) Apartat «Seguretat de les TIC»: dedicat a les definicions bàsiques i els principis fonamentals relatius a la seguretat.

2) Apartat «Control d'accés»: en aquest apartat, hem estudiat els diferents models de control d'accés existents i les fases que els conformen, fent especial èmfasi en l'autenticació i els seus factors.

3) Apartat «Amenaces i atacs»: tracta sobre la naturalesa de les amenaces i els tipus d'atacs existents: els errors del programari (desbordament, situacions de competició, fuites de recursos), el codi maliciós en les seves múltiples variants (virus, cucs, troians, bombes lògiques, *rootkits*, *backdoors*, *ramsonware*, *adware*, *spyware* i *cryptojacking*), els atacs de denegació de servei, els atacs d'intermediari i l'enginyeria social.

4) Apartat «Seguretat física»: es repassen alguns conceptes sobre la seguretat física dels sistemes.

5) Apartat «Seguretat del servidor»: s'analitza la seguretat a nivell de servidor, com ara el *hardening* dels sistemes operatius, les tasques i responsabilitats de l'administrador, el sistema de fitxers, les contrasenyes i els atacs associats, les intrusions i els protocols a seguir quan es produeixen.

6) Apartat «Seguretat de les dades»: es descriuen alguns punts relatius a la seguretat de les dades, introduint conceptes de criptografia de clau pública i privada, la signatura digital, el certificat digital i l'estenografia.

7) Apartat «Seguretat de la xarxa»: es considera, en aquest apartat, la seguretat de les comunicacions en xarxa i els elements dedicats a millorar-la, com els tallafocs, *proxies*, sistemes NAT, sistemes de detecció i prevenció d'intrusos, esquers, xarxes privades virtuals, detectors, sistemes de monitorització, escàners de vulnerabilitats i els tests d'intrusió.

8) Apartat «Seguretat del núvol»: es fa una breu introducció als aspectes a tenir en compte a l'hora de dissenyar els sistemes de seguretat quan s'utilitzen recursos de computació allotjats al núvol.

9) Apartat «Seguretat de la web»: s'analitzen les problemàtiques específiques a la seguretat de les aplicacions web, tot introduint els conceptes més importants de la tecnologia en què es sustenten i tractant les vulnerabilitats i els atacs més habituals.

10) Apartat «Aspectes legals. Marc jurídic penal i extrapenal. El "delicte informàtic"»: descriu les responsabilitats en què pot incórrer l'administrador d'un sistema TIC (tant pel que fa a la que adquireix envers el sistema –maquinari i programari– com per les dades que s'hi emmagatzemen. D'altra banda, també s'enumeren els possibles delictes de què pot ser víctima a l'entorn de treball, i la manera de denunciar-los.

11) Apartat «Informàtica forense»: descriu què es pot fer una vegada ha succeït un problema de seguretat (o, fins i tot, un delicte) per poder esbrinar què ha passat i qui ha estat el presumpte autor. Es defineix el concepte d'informàtica forense, una disciplina situada a cavall entre la informàtica i el dret.

12) Apartat «Kali Linux»: es presenta breument la distribució de Linux Kali, especialitzada en seguretat.

Activitats

1. Dissenyeu un pla de seguretat física per a una organització que vosaltres conegueu (l'organització en què treballeu, l'aula d'informàtica d'una facultat, etc.). Per fer-ho, us podeu orientar en l'esquema següent:

a) Descripció dels recursos físics que es volen protegir.
b) Descripció de l'espai físic on es localitzen els recursos.
c) Descripció del perímetre de seguretat.
d) Enumeració de les amenaces que poden comprometre la seguretat del sistema.
e) Possibles mesures de seguretat contra les amenaces anteriors.
f) Manera d'implementar les mesures anteriors.
g) Càlcul del cost estimat de la implementació de les mesures o millores que cal fer, i també del cost de les dades que cal protegir i la probabilitat que es produeixi un atac.

2. Cerqueu informació sobre les associacions següents, relacionades amb la informàtica forense:

a) International Association of Computer Investigative Specialist (IACIS). Aquesta associació ofereix una certificació internacional (CFEC, Computer Forensic External), adreçada a analistes que no formin part dels cossos policials o judicials.
b) European Network of Forensic Science Institute (ENFSI).
c) International Organisation on Computer Evidence (IOCE).
d) Equipo de Seguridad para la Coordinación de Emergencias en Redes Telemáticas (es-CERT).

3. La funció d'un pèrit judicial consisteix a proporcionar al jutge la informació necessària per ajudar-lo a determinar què ha succeït en el cas que s'investiga. La figura del pèrit judicial s'introdueix a l'article 456 (i següents) de la Llei d'enjudiciament criminal. Cerqueu els articles que defineixen aquesta figura i raoneu les qüestions següents:

a) Penseu que és necessari disposar de la titulació universitària en Informàtica per poder exercir de pèrit?
b) Quins són els drets i deures del pèrit?

Exercicis d'autoavaluació

1. Raoneu breument quina de les tres propietats que ha de satisfer un sistema informàtic «segur» és prioritària en els sistemes següents:
- Una organització de defensa nacional.
- Un sistema de transferència electrònica de diners.
- Un departament d'una universitat.

2. La pàgina web del servidor del departament que administreu ha estat víctima d'un atac i ha estat substituïda per una altra pàgina amb un contingut completament diferent. Quines són les accions que haureu de fer per denunciar el fet davant d'un cos policial?

3. Determineu quins són els enunciats correctes:

a) L'enviament de correu no sol·licitat (*spam*) és una conducta que apareix tipificada en el codi penal.
b) La signatura electrònica avançada té la mateixa consideració que la signatura manuscrita.
c) La intrusió en un sistema informàtic, en si mateixa, no és una conducta tipificada en el codi penal.
d) La figura del responsable del fitxer o tractament és la mateixa que la del tractament del fitxer.

Solucionari

Exercicis d'autoavaluació

1. Confidencialitat, integritat i disponibilitat, respectivament.

2. Accions que cal fer en el cas d'un delicte de danys.

- Desconnexió de la xarxa.
- Fer una còpia de seguretat a baix nivell.
- Compilar tota la informació possible sobre l'atac.
- Restaurar el sistema i aplicar les actualitzacions de programari.
- Fer la notificació a qui es consideri convenient i segons l'atac (al nostre cap, al CERT, a altres administradors d'altres sistemes implicats, als usuaris del nostre sistema, etc.).
- Sol·licitar una valoració dels danys produïts.
- Denunciar el fet a la policia, incloent tota la informació possible sobre l'atac i la valoració dels danys produïts (feta per la mateixa organització o per un pèrit extern).

3. b c

Glossari

atac *m* Acte deliberat que té com a objectiu transgredir la seguretat.

atac de canal lateral *m* Atac que consisteix a aprofitar les vulnerabilitats d'un sistema analitzant paràmetres físics.

atac d'intermediari *m* Atac que consisteix a interposar-se en una comunicació, interceptant els missatges i possiblement modificant-los.

autenticació *f* Verificació de la identitat d'una persona o procés a l'hora d'accedir a un recurs o poder fer una acció determinada.

anàlisi forense *f* Procés d'aplicació del mètode científic als sistemes TIC amb la finalitat d'assegurar, identificar, preservar, analitzar i presentar l'evidència digital de forma que sigui acceptada en un procés judicial.

certificat digital *m* Document electrònic signat per una tercera part o autoritat de certificació que associa una clau pública a una persona.

codi maliciós *m* Programari dissenyat per dur a terme algun tipus d'acció nociva.

confidencialitat *f* Principi de seguretat segons el qual només es pot accedir als recursos de manera autoritzada.

contramesura *f* Mètode de defensa contra atacs.

control d'accés *m* Conjunt de mecanismes per determinar en un sistema si un agent pot accedir a un recurs.

criptosistemes de clau privada *m pl* Criptosistemes en els quals l'emissor i el receptor comparteixen una única clau. És a dir, el receptor podrà desxifrar el missatge rebut únicament si coneix la clau amb la qual ha xifrat el missatge l'emissor.

criptosistemes de clau pública *m pl* Criptosistemes en què cada usuari u té associada una parella de claus $<P_u, S_u>$. La clau pública, P_u, és accessible a tots els usuaris de la xarxa i apareix en un directori públic, mentre que la clau privada, S_u, solament és coneguda per l'usuari u.

dada de caràcter personal *f* Qualsevol informació relativa a les persones. En concret, tota informació numèrica, alfabètica, gràfica, fotogràfica, acústica o de qualsevol altre tipus, susceptible de ser recollida, enregistrada, tractada o transmesa i que concerneix a una persona física identificada o identificable.

denegació de servei *f* Atac que consisteix a fer inaccessibles els recursos d'un sistema.

disponibilitat *f* Principi de seguretat segons el qual els recursos han de romandre accessibles i només per als elements autoritzats.

enginyeria social *f* Conjunt de tècniques per tal d'influir en una persona perquè faci una determinada acció.

esteganografia *f* Conjunt de tècniques que permeten amagar informació.

factor d'autenticació *m* Mètode utilitzat per determinar la identitat d'un agent en un sistema de control d'accés.

fitxer automatitzat *m* Conjunt organitzat de dades que és objecte de tractament automatitzat.

funció *hash* *f* Funció matemàtica que fa correspondre una representació de mida fixa a un missatge m de mida variable.

hardening Procés de reduir la superfície d'atac d'un sistema.

integritat *f* Principi de seguretat segons el qual només es poden modificar els recursos de manera autoritzada.

pla de contingència *m* Protocol d'actuació establert que s'ha d'iniciar quan es produeix una emergència o desastre.

principi del mínim privilegi *m* Principi segons el qual els agents només han de tenir accés als recursos que siguin estrictament necessaris per dur a terme la seva funció.

política de seguretat *f* Conjunt de directrius o estratègies que han de seguir els usuaris en relació amb la seguretat global del sistema TIC.

responsable del fitxer *m i f* Persona física o jurídica, pública o privada, i òrgan administratiu que decideix sobre la finalitat, el contingut i l'ús del tractament.

seguretat informàtica *f* Conjunt constituït per diverses metodologies, documents, programari i maquinari, que determinen que els accessos als recursos d'un sistema TIC siguin duts a terme exclusivament per als elements autoritzats a fer-ho.

spoofing Tècnica d'atac a un sistema en què l'intrús simula una adreça IP d'origen, diferent de l'adreça IP real de l'atacant.

tècnica de l'empremta *f* Activitat que consisteix en la recollida d'informació sobre l'objectiu que es vol atacar utilitzant mètodes indirectes.

test d'intrusió *m* Conjunt de procediments per buscar vulnerabilitats en un sistema utilitzant tècniques d'atac.

vulnerabilitat *f* Debilitat d'un sistema susceptible de ser atacada amb èxit.

Bibliografia

Bosworth, S.; Kabay, M. E.; Whyne, E. (2014). *Computer security handbook*. Nova Jersey: John Wiley & Sons.

Brotherson, L.; Berlin, A. (2017). *Defensive security handbook*. Massachusetts: O'Reilly.

Daimi, K. (2018). *Computer and network security essentials*. Suïssa: Springer.

Easttom, C. (2016). *Computer security fundamentals*. Londres: Pearson Education.

Hadnagy, C. (2018). *Social engineering: the science of human hacking*. Nova Jersey: John Wiley & Sons.

Hodeghatta Rao, U.; Nayak, U. (2014). *The infosec handbook*. Nova York: Apress Open.

Lehtinen, R.; Russell, D.; Gangemi, G. T. (2012). *Computer Security Basics*. Massachusetts: O'Reilly.

Martin, K. M. (2012). *Everyday cryptography*. Oxford: Oxford University Press.

Migga Kizza, J. (2014). *Computer network security and cyber ethics*. Carolina del Nord: McFarland & Company.

Pfleeger, C. P.; Pfleeger, S. L.; Margulies, J. (2015). *Security in computing*. Londres: Pearson Education.

Priyam, P. (2018). *Cloud security Automation*. Birmingham: Packt Publishing.

Stallings, W.; Brown, L. (2012). *Computer security*. Londres: Pearson Education.

Stamp, M. (2011). *Information security*. Nova Jersey: John Wiley & Sons.

Stuttard, D.; Pinto, M. (2011). *The web application hacker's handbook*. Nova Jersey: John Wiley & Sons.

Vacca, J. R. (2017). *Cloud computing security: foundations and challenges*. Florida: CRC Press.

Walker, B. (2019). *Cyber security*. Edició independent.

Zalewski, M. (2012). *The tangled web*. Califòrnia: No Starch Press.