Implementació d'Autenticació de doble factor (2FA) mitjançant OTPs

Xavier Molina Garcia

Resum– En aquest projecte s'ha implantat un sistema d'autenticació de doble factor en connexions ssh. En concret s'ha fet ús dels codis d'un sol ús, més coneguts com a OTP (One Time Password). Aquests codis es generen de forma aleatòria a través d'un software codi obert anomenat LinOTP, aquest software permet adaptar múltiples mòduls en funció de les necessitats de cada cas, concretament s'ha utilitzat el mòdul PAM (Pluggable Authentication Modules). Es completa amb un escenari possiblement real, on es comunica un grup d'alumnes universitaris amb el laboratori des d'on realitzen les pràctiques, de manera remota des de fora del campus. La utilitat del desenvolupament recau en la possibilitat d'aplicar-ho en un sistema empresarial real. On d'una forma relativament senzilla es pot afegir una barrera extra de seguretat per accedir als sistemes corporatius.

Paraules clau– Autenticació, doble factor, seguretat, codis d'un sol ús, One Time Password, LinOTP, Pluggable Authentication Modules, xifratge, criptografia.

Abstract– A two-factor authentication system has been implemented in this project on ssh connections. In particular, single-use codes have been used, better known as OTP (One Time Password). These codes are generated randomly through open-source software called LinOTP, which allows multiple modules to be adapted depending on the needs of each case, specifically the PAM module (Pluggable Authentication Modules) has been used. It is completed with a possibly real scenario, where one group of university students is communicated with the lab from where they perform the practices, remotely from outside the campus. The usefulness of development lies in the possibility of implementing it in a real business system. Where in a relatively simple way an extra security barrier can be added to access corporate systems.

Keywords– Authentication, double factor, security, One-time password, LinOTP, Pluggable Authentication Modules, Encryption, cryptography.

1 Introducció - Context del treball

ACTUALMENT en el món tot està informatitzat, des de la pàgina web de qualsevol comerç passant per les xarxes socials o els sistemes bancaris, etc. I una de les premisses més importants és la seguretat de les dades que emmagatzemen aquests sistemes, no volem que ningú hi accedeixi sense permís.

Pensem per un moment quina és l'única barrera que existeix per accedir al nostre compte d'Instagram, una contrasenya, més llarga o més curta però no deixa de ser una combinació alfa numèrica que hem escollit nosaltres. I aquesta combinació de números i lletres, segurament per comoditat, estigui repetida en el teu correu electrònic o en el portal de Facebook.

Amb aquesta premissa al davant, deduïm que les contrasenyes no són suficientment segures per si soles. Tots volem que la nostra informació estigui emmagatzemada de la forma més segura possible. I amb aquest estudi s'investiga sobre la possibilitat d'afegir una mesura de seguretat extra, que no permeti accedir a usuaris no legítims encara que hagin estat capaços d'aconseguir les credencials d'accés a un sistema.

En primer lloc, que és autenticarse? En sistemes informàtics, és el proces per el qual un individu certifica

• E-mail de contacte: xavier.molinag@e-campus.uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Angel Elbaz Sanz (DEIC)
• Curs 2020/2021

que és qui diu ser. La proposta està basada en sistemes d'autenticació de doble factor. Els sistemes de doble factor tenen dues barreres de protecció per poder verificar l'identitat de l'usuari i poden fer servir multituds de tècniques d'autenticació diferents, com contrasenyes alfa numèriques, sistemes biomètrics, credencials d'un sol ús, etc. Nosaltres concretament utilitzarem One Time Password, que consisteixen en un segon codi que arriba a l'usuari a través d'un correu electrònic, un SMS o una aplicació que ens permet validar d'una forma més estricta la identitat de l'individu. Prèviament, es farà un repàs pels sistemes d'autenticació clàssics d'un factor amb les seves variants, igualment pels sistemes de doble factor i quines combinacions són les més utilitzades actualment.

En l'àmbit pràctic s'implementa aquest procés de millora de la seguretat amb un escenari possiblement real, on es comunica un grup d'alumnes universitaris amb el laboratori des d'on realitzen les pràctiques, de manera remota des de fora del campus. Aquests alumnes es connecten a través del protocol SSH i a través del software LinOTP afegim el segon pas de l'autenticació.

2 Estat de l'art

La ciberseguretat és un àmbit de la tecnologia que ha cobrat una importància clau en els últims anys. Tant és així que en el 2018 aquest sector de mercat va moure més de 5.000 milions de dòlars. Com a tal és un sector que engloba moltes tecnologies diferents.

Centrant-nos específicament en els sistemes d'autenticació basats en les contrasenyes que tothom coneixem. Segons SplashData, que realitza estudis sobre les contrasenyes més utilitzades, al 2019 la contrasenya més freqüent va ser "123456", seguit de "123456789" i "qwerty".

Estudis com aquests demostren any rere any que els sistemes d'un factor com els de contrasenya alfa-numèrica no són prou segurs actualment. A més, molts cops, un mateix individu té la mateixa contrasenya per diferents plataformes. Fent que quan una contrasenya es veu compromesa els altres portals també ho estiguin.

3 Objectius

La realització d'aquest projecte vol aconseguir diversos objectius. Cadascun d'ells té una prioritat i criticitat diferent, per tant si no s'arribessin a complir tots, tindríem uns punts clau i mínims pel bon desenvolupament del projecte.

- Analitzar la viabilitat dels mètodes d'autenticació de doble factor (concretament OTP's).
- Generar la documentació necessària per tenir un coneixement teòric profund sobre els sistemes d'autenticació.
  - Autenticació d'un factor.
  - Autenticació de doble factor.
- Implantació d'un prototip d'autenticació de doble factor mitjançant OTP's i provar-ho amb màquines virtuals.
  - En el cas que la implantació es compliqui i no s'arribin als terminis establerts en la planificació del projecte s'especificarà el model teòric realitzat i una demo esquemàtica per mostrar el flux de comportament previst pel programa.

A més de l'elaboració del projecte també es busca assolir uns objectius personals i tècnics. En l'àmbit personal observem els següents:

- Aprendre a gestionar un projecte des de zero, tant la planificació inicial, l'execució del mateix i la documentació que es generi.
- Conèixer millor una part de la seguretat informàtica, àmbit el qual m'hi voldria dedicar.
- Posar a prova la meva capacitat de portar a terme un projecte així sense altres intervencions (a part de la del tutor).

Per últim, els objectius en l'apartat tècnic serien els següents:

- Millorar els meus coneixements sobre estàndards de seguretat.
- Conèixer en profunditat els sistemes d'autenticació.

4 Metodologia

Per la metodologia de desenvolupament utilitzarem un tipus waterfall, ja que considerem que és un projecte molt acotat i és necessari realitzar un estudi teòric previ abans de començar a preparar la implantació del sistema. Es definiran uns temps i uns punts de control per saber si el projecte es va desenvolupant en els terminis correctes.

El treball implicarà una metodologia deductiva on es farà èmfasi en el coneixement teòric i la bona documentació prèvia, un procés que es realitzarà durant el període de creació de l'informe de progrés 1. Després centrar-nos en el nostre experiment particular, que es desenvoluparà durant el període del informe de progrés 2. Consistirà en la implantació del sistema de doble factor esmentat anteriorment. Per últim es recollirà tota la documentació generada en el informe final i es realitzarà l'exposició davant del tribunal. El model per argumentar si els objectius del treball es compleixen serà de caràcter quantitatiu, valorant si s'estan assolint o no, en base al bon funcionament o no de la implantació del programari. També es realitzaran les tasques de manera síncrona i amb un ordre predefinit, com es detalla a la planificació del projecte, ja que estem treballant en un model waterfall. A més, el diagrama de Gantt generat permetrà veure de manera gràfica si s'estan assolint els terminis establerts al inici del projecte.

5 ESTUDI DELS SISTEMES D'AUTENTICACIÓ

L'autenticació és el procés pel qual es verifica que algú és qui diu ser. Aquest procés consisteix en que el provador (persona que s'està identificant) coneix o té alguna cosa que el verificador (sistema que verifica la identitat del provador) ha de validar.

En referència específica a la seguretat en xarxes de dades, l'autenticació és un dels tres passos fonamentals a seguir. Parlem d'autenticació, autorització i auditoria. En seguretat informàtica correspon a l'acrònim AAA, que reuneix una sèrie de protocols de control d'accés.

Existeixen múltiples sistemes d'autenticació[1], els més comuns en els sistemes informàtics són els basats en alguna cosa coneguda, com seria una contrasenya alfanumèrica. També hi ha els basats en alguna cosa posseïda com una targeta acreditativa (DNI, SmartCard, etc) i per últim el basats en una característica física com l'empremta digital, el reconeixement fàcil, el reconeixement de veu, etc [2].

5.1 Autenticació d'un factor

Són aquells sistemes d'autenticació que només requereixen un sol pas per verificar la identitat de l'usuari, sigui del tipus que sigui. Són els sistemes més simples que existeixen i també els més econòmics, això fa que la majoria de sistemes que necessiten implementar un sistema d'autenticació optin per aquest tipus.

També és important destacar que les autenticacions d'un sol factor normalment es produeixen en forma de codi numèric o contrasenya alfanumèrica. Ho podem observar en qualsevol portal web o d'altres aplicatius. Per altra banda també existeixen casos (sobretot en autenticació en telefonia) d'autenticacions d'un sol pas biomètriques. Són molt pràctiques per l'usuari, però requereixen un sistema alternatiu d'accés ràpid, ja que si l'element biomètric es troba modificat no hagem de fer un procés de restauració de credencials. En aquest àmbit s'acaba recorrin al codi numèric comentat anteriorment. Un exemple molt comú és l'autenticació per empremta dactilar que molts dispositius mòbils implementen actualment.

5.2 Problemes d'autenticació d'un sol factor

Aquests sistemes tenen el problema de ser els més fàcils de saltar, ja que per les seves característiques només tenen una barrera de protecció. Si un atacant aconsegueix les credencials hi podrà accedir. Habitualment estan basats en sistemes d'usuari i contrasenya, si aquestes contrasenyes no són prou segures el sistema es veurà compromès.

S'ha de tenir en compte que com al cas anterior, si l'autenticació és biomètrica serà molt difícil que un atacant pugui autenticar-se, però al tenir un sistema d'autenticació alternatiu en forma de codi estaríem en la mateixa problemàtica que presenten les autenticacions de contrasenyes.

5.3 Autenticació de dos factors

Són aquells sistemes que requereixen dos passos per verificar la identitat de l'usuari. Es combinen amb sistemes d'autenticació diferents, un exemple seria una contrasenya i a més la verificació biomètrica d'empremta dactilar.

L'autenticació de dos factors agrega una barrera més de seguretat i complica el fet que un atacant aconsegueixi accedir al sistema. I perquè el sistema d'autenticació sigui segur, ha d'estar compost per la combinació d'alguna cosa que coneixes més alguna cosa que tens. Això fa que encara que allò que coneixes es vegi compromès, sense allò que tens no es pugui accedir al sistema[3][4].

Però no s'ha de confondre amb un sistema d'autenticació de dos passos, que difereix en que el segon pas, no és alguna cosa que tens sinó una segona contrasenya amb una validesa temporal, i pot arribar a l'usuari a través d'un correu electrònic, un SMS o una altra aplicació[5].

5.4 Problemes d'autenticació de dos factors

En el cas d'estar utilitzant autenticacions de doble factor amb quelcom que tens i no estiguis en possessió d'aquesta cosa en el moment de la teva autenticació, no hi podràs accedir encara que siguis l'usuari legítim.

Un exemple clar és quan estem utilitzant un token usb o una smartcard. Si perdem aquest article no podrem verificar la nostra identitat. En el cas d'estar utilitzant un sistema biomètric, si aquest element biomètric pateix un canvi com un tall o una ferida no serà reconegut com a vàlid.

Per últim en el cas de les verificacions per dos passos a través de OTP és necessari assegurar-se que les comunicacions són segures, ja que el codi temporal s'ha de subministrar a l'usuari a través d'un canal de comunicació segur. I així evitar atacs de "Man in the Middle"que podrien realitzar esnifades de la xarxa.

6 ESTUDI D'APLICACIONS OTP EXISTENTS

6.1 Algoritmes

Les aplicacions OTP requereixen diferents algoritmes per dur a terme la seva funció. Es necessita poder xifrar la informació que enviarem per la xarxa, per això utilitzarem l'algoritme SHA, que converteix la contrasenya que l'usuari ha d'introduir en una cadena Hash. A l'utilitzar el protocol SSH s'utilitza algoritme AES, que també xifra informació, per encriptar la informació que viatja en les comunicacions entre servidor i client SSH. Per últim l'algoritme MD5, que servirà per comprovar que no s'han efectuat canvis en el missatge mentre viatjava per la xarxa.

6.1.1 SHA

SHA de les sigles en angles Secure Hash Algorithm, és una família de funcions hash publicades per el NIST (Nacional Institute of Standards and Technology). La primera versió anomenada SHA-0 va ser creada al 1993 i al llarg dels anys

s'han anat publicant noves versions.

SHA-0 va ser publicat al 1993 i tenia una longitud de 160 bits però al 1998 es va trobar una vulnerabilitat que afectava l'algoritme i es va substituir per SHA-1.

SHA-1 va ser publicat al 1995 també amb una longitud de 160 bits, però segons l'única publicació oficial al respecte els canvis realitzats a l'algoritme, es redueixen les probabilitats de col·lisió. Des de 2005 no es considera segur arran d'una vulnerabilitat de MD5.

SHA-2 va ser publicat al 2001 i inclou canvis significatius respecte a la versió anterior, consta d'una família de sis funcions (SHA-224, SHA-256, SHA-384, SHA-512, SHA512/224, SHA-512/256) que varien entre paraules de 32 i 64 bits i bucles que s'executen entre 64 i 80 rondes. Al 2011 es va produir un ataca que va ser capaç de trencar la resistència de preimatge per SHA-512 (entre 57 i 80 rondes) i SHA-256 (entre 52 i 64 rondes), encara que actualment no s'han trobat col·lisions de sortida.

SHA-3 va ser publicat al 2015 i és l'última publicació d'aquesta família d'algoritmes. A diferència dels anteriors algoritmes, SHA-3 no és una evolució de la versió anterior sinó que esta construït de manera diferent. Utilitza una estructura d'esponja i és un subconjunt de la família criptogràfica Keccak, dissenyada per Guido Bartoni, Joan Daemen, Michaël Peeters i Gilles Van Assche. Els seus creadors suggereixen utilitzar la funció més ràpida Kangaroo Twelve, amb paràmetres més ajustats i nou mode d'arbre hash sense despeses generals addicionals per missatges de petit format.[6][7][8]

6.1.2 AES

De les sigles en angles Advanced Encryption Standard, és un esquema de xifratge per blocs creat per el NIST i actualment és un estàndard des del 2002. Al contrari que el seu predecessor DES, AES és ràpid tant en execució software com hardware, ràpid d'implementar i exigeix pocs recursos de còmput, fent que sigui escalable. AES realitza operacions en una matriu 4x4 amb blocs de 128 bits, seguint els següents passos:

1. Expansió de la clau (Esquema de Rijndael)
2. Afegir clau de rondes
3. Rondes
   (a) Substitució no lineal de cada byte en relació a la taula de cerques.
   (b) Transposició de files de manera cíclica.
   (c) Barreja de columnes, combinant els quatre bytes de cada columna de forma no lineal.
   (d) Combinar cada byte amb la clau de rondes.
4. Per últim un cop s'ha extret la taula final, s'executa una ultima ronda d'aquesta taula amb els passos anteriors (a, b, d).

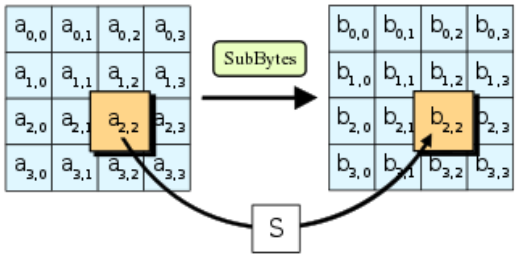

Fig. 1: Substitució no lineal

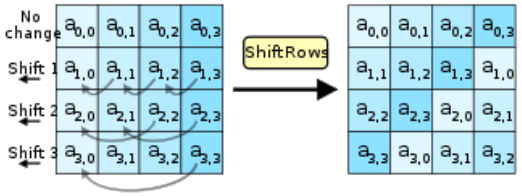

Fig. 2: Transposició cíclica

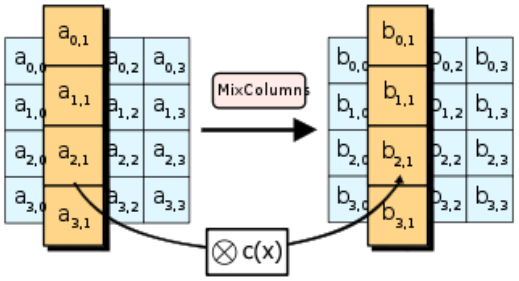

Fig. 3: Barreja de columnes

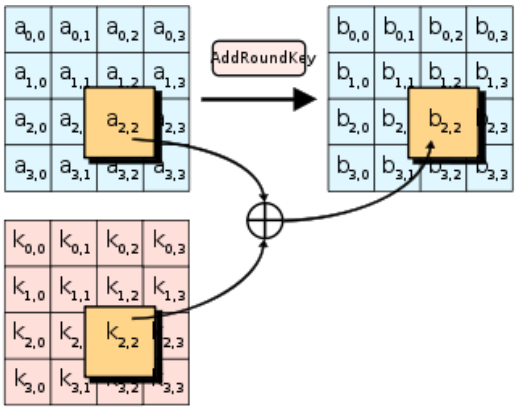

Fig. 4: Combinació bytes i claus

Encara que existeix una preocupació per la integritat de AES en el que respecta a la seva ruptura per força bruta i la seva integritat matemàtica, actualment no s'ha aconseguit trencar la seguretat del mateix. Un teòric atac denominat XSL, mostra una potencial debilitat de AES però la comunitat matemàtica ha trobat problemes en la integritat d'aquesta demostració teòrica. [9]

6.1.3 MD5

De les sigles en angles Message Digest Algorithm 5, és un algoritme de reducció criptogràfica de 128 bits. Un dels seus usos més comuns és la comprovació de que un arxiu no hagi estat modificat.

El sistema UNIX utilitza MD5 per calcular el hash de les claus dels usuaris. Al disc s'emmagatzema el resultat del MD5 de la clau que introdueix l'usuari en donar-se d'alta. A l'hora de realitzar l'autenticació es compara el hash MD5 de la clau introduïda amb el hash emmagatzemat en el sistema. El funcionament de MD5 es basa en els següents 5 passos:

1. Addicció de bits, el missatge es estes fins arribar a 448 mòdul 512.

2. S'ajusta per tenir un missatge exacte de 64 bits.

3. Inicialització del buffer de MD, un buffer de quatre paraules (A, B, C, D) s'utilitza per calcular el resum del missatge i s'inicialitzen amb uns valors exactes en forma hexadecimal.

```
palabra A: 01 23 45 67
palabra B: 89 ab cd ef
palabra C: fe dc ba 98
palabra D: 76 54 32 10
```

Fig. 5: Valors exadecimals exactes

4. Processat del missatge en blocs de 16 paraules

$$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$$

Fig. 6: Processat dels codis de 16 paraules

5. Sortida del resum del missatge produïda per A, B, C i D. Des del bytede menor pes de A fins el byte de major pes de D.

6.2 Característiques

Per desenvolupar un software dedicat a l'autenticació en dos passos, segons el que hem vist prèviament, és important treballar amb un llenguatge de programació que sigui eficient i que permeti el desenvolupament orientat a sistemes. Per això moltes aplicacions enfocades a l'autenticació d'usuaris a l'hora d'accedir al sistema operatiu estan desenvolupades en llenguatge C. Per altra banda també és possible trobar-se amb la casuística de voler realitzar una doble autenticació a nivell d'un portal web, en aquest àmbit trobem programaris desenvolupats en llenguatge Python. Al ser un llenguatge interpretat i no compilat permet aplicar-se d'una forma més transversal en processos d'autenticació que no disposen d'una doble autenticació.

També s'ha de tenir en compte la seguretat de les comunicacions entre el proveïdor de la segona contrasenya i l'usuari que rep el codi. Per evitar atacs de "man in the middle" que facin que encara que tinguem una autenticació de dos passos, el possible atacant pugui robar les dues contrasenyes. Per això utilitzem algoritmes com el SHA que permeten xifrar un missatge i d'altres com el MD5 que ens permeten verificar que aquest missatge no ha estat modificat durant el transport cap a l'usuari final.

Uns termes importants quan parlem de OTP és conèixer el significat de les sigles HOTP i TOTP:

- HOTP: La OTP basada en esdeveniments, és a dir, basada en HMAC. És l'algoritme original que s'utilitza en autenticacions d'aquest tipus, i que depèn de dos tipus d'informacions. La clau secreta la qual només la coneix el token i el servidor, i un comptador que s'emmagatzema tant al token com al servidor. Aquest comptador s'incrementa al utilitzar-se el OTP o al generar un nou codi.

- TOTP: La OTP basada en temps, és a dir, esta basada en una HOTP, però el seu comptador utilitza una conta enrere abans d'incrementar-se.

Des d'una perspectiva de seguretat el sistema TOTP és clarament superior i per això la majoria d'aplicacions actuals utilitzen aquest sistema. Una problemàtica que podria sorgir però, és que la connexió entre usuari i servidor no fos prou bona, fent que la finestra de temps no fos suficient perquè l'usuari pogués autenticar-se. Augmentant una mica la finestra quan es donin aquest tipus de casos podria ser una solució rapida i senzilla. S'ha de vigilar no augmentar massa la finestra de temps, ja que en el cas que un atacant estigues intentat robar el codi, podria tenir temps per fer el robatori de claus i identificar-se il·legítimament.

6.3 Elecció de programari

Després d'analitzar una sèrie de programaris open source per autenticar-se en el sistema operatiu Linux mitjançant OTP's. Hem escollit l'eina LinOTP per els següents motius:

LinOTP és una solució a nivell d'empresa per una autenticació de caràcter fort, desenvolupada i mantinguda per KeyIdentity GmbH, capaç d'escalar des d'instal·lacions petites i individuals fins escenaris de mitjanes companyies proveïdors de sistemes Cloud. Això és possible a través de la modularitat de LinOTP. Al voltant d'un servidor amb una sèrie d'interfícies definides fent que el procés d'integració de LinOTP en el teu sistema sigui més senzill. Quan les necessitats del client canvien i necessiten ser escalades, LinOTP és capaç de créixer per adaptar-se a les noves necessitats de l'entorn. LinOTP no només és capaç de créixer mitjançant la seva modularitat, sinó que a més es un valor de futur. Permet fer modificacions en l'emmagatzematge d'usuari sense fer canvis en el nucli del sistema. LinOTP està desenvolupat principalment amb Python però proporciona una API de tipus REST.[10]

- És open source encara que té versions de pagament amb funcionalitats esteses.

- Ens proporciona una interfície gràfica.

- És un programari lleuger que no requereix massa recursos de còmput.

- Proporciona flexibilitat a l'hora de crear els tokens per l'autenticació, podent variar entre diferents algoritmes i mètodes de xifratge.

Els processos d'instal·lació i configuració es realitza a través del terminal. A més inclou altres funcionalitats com són:

- Integració total amb SSH amb PAM
- Integració amb altres bases de dades de tercers.

6.3.1 Flux de dades

El procés ha seguir un cop s'hagi implementat l'aplicació serà el següent. Inicialment l'usuari haurà d'introduir les seves credencials d'inici de sessió mitjançant el terminal. Seguidament si aquestes són valides es passaria a generar el codi OTP a través de l'aplicació LinOTP, de no ser correctes l'usuari haurà de tornar a introduir les credencials d'inici de sessió. Un cop generat el codi OTP s'haurà d'introduir en el procés d'autenticació, de ser correcte l'usuari ja estaria autenticat i el procés d'inici de sessió hauria finalitzat. De no ser correcte el codi escrit per l'usuari l'hauria de tornar a escriure. Si no aconsegueix realitzar correctament l'autenticació i el temps de validesa del codi OTP expira, se'n generarà un de nou perquè l'usuari el pugui escriure en el procés d'autenticació.

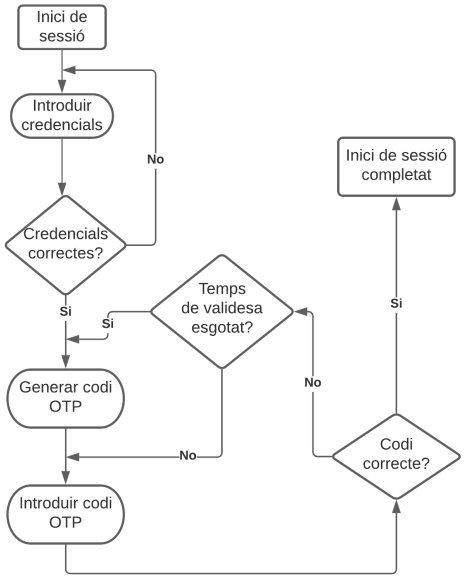

Fig. 7: Diagrama de flux

7 CAS PRÀCTIC

Per donar una visió més clara sobre aplicacions reals d'aquest desenvolupament, establirem un possible entorn real on veure reflectida tota la part pràctica realitzada.

Ens situem en l'àmbit de la Universitat, en una assignatura de tercer curs, on a causa de les restriccions per la pandèmia no es poden fer les classes pràctiques de laboratori. Ens trobem en la situació que es necessita un programari específic que només trobem a les màquines del laboratori, i voldríem poder connectar-nos de manera segura des de fora de la universitat.

Per realitzar la comunicació els alumnes hauran de connectar-se a través de SSH des de les seves màquines a les màquines del laboratori.

Per realitzar les proves d'autenticació del nostre cas real a traves de ssh utilitzarem linotp amb una base de dades Mysql, un servidor apache 2 i el mòdul d'autenticació PAM. [Fig.8]

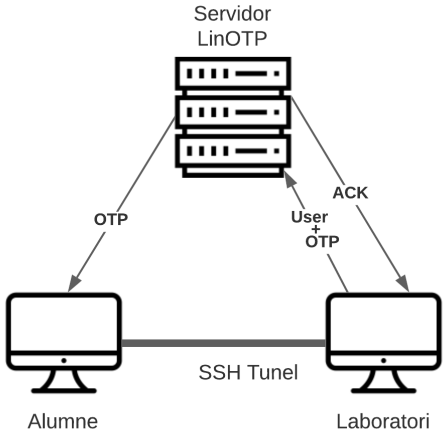

Fig. 8: Esquema de funcionament

7.1 Creació de l'entorn de proves

7.1.1 Màquines virtuals

Per el desenvolupament pràctic del projecte s'utilitzaran entorns virtuals creats amb l'eina VirtualBox.[11][12]

VirtualBox es un potent producte de virtualització x86 i AMD64/intel64 per us empresarial i particular. Es un software de codi obert sota els termes de la GNU General Public License (GPL) versió 2. Actualment aquest programa s'executa en hosts de tipus Windows, Linux, Mac i Solaris. Amés de ser compatible amb una gran de quantitat de sistemes operatius convidats, com Windows Server, OpenSolaris, etc.

VirtualBox es continua desenvolupant activament amb actualitzacions freqüents i te una llista de característiques cada cop major. Oracle garanteix la qualitat professional però la comunitat pot aportar-hi millores.

En el nostre cas s'utilitzarà una distribució Ubuntu de Linux per fer les nostres proves. La obtenció d'aquesta distribució de Linux s'ha fet a través de el distribuïdor oficial Ubuntu.[13]

7.1.2 Instal·lació de LinOTP

- Des del Terminal de Linux realitzem els següents passos: Apèndix A.1
- Instal·lació de SSH Server en l'amfitrió: Apèndix A.2
- Instal·lació de SSH Client en el client: Apèndix A.3

- Configuració dels adaptadors de xarxa de VirtualBox: Apèndix A.4

7.1.3 Comprovació de les funcionalitats

Primerament establim una connexió ssh estàndard sense cap tipus d'afegit. Únicament se'ns demana usuari i contrasenya per poder accedir a la maquina host.[Fig.9]

Al connectar-nos a la màquina de destí només es demana la contrasenya d'accés. Per fer unes comprovacions accedim al directori Documentos i creem un directori anomenat "prova", un cop fet això tanquem la connexió i comprovem que al nostre directori Documentos no hi existeix.

Fet això comprovem que prèviament no hi existia el directori "prova"a la carpeta Documentos, i a través de la connexió ssh des de l'altre màquina l'hem creat.[Fig.10]

Seguidament afegirem un token a la nostra base de dades que contindrà la informació de la nostra màquina, aquest token es pot generar amb el mòdul PAM de forma automàtica, registrant la informació de l'usuari i aplicant una configuració de OTP predefinida, amb un codi de tipus TOTP i un xifratge SHA256.

A continuació només hem de carregar o escanejar el codi QR a l'aplicació LinOTP, de forma automàtica quedarà creat el token. Fet això ja tindríem establert una configuració de 2FA al nostre usuari de ssh, però cal un mètode de comunicació entre l'usuari i el servidor. Existeixen diferents formes com hem vist anteriorment (Email, SMS, etc) en el nostre cas per facilitar la vida a un possible usuari hem fet servir una aplicació de generació de OTP com es Google Authenticator. Aquesta eina també pot escanejar el codi que hem generat prèviament i des de l'aplicació vincular la generació de codis amb LinOTP.

D'aquesta forma podrem visualitzar els codis OTP inclús sense tenir connexió a internet en el nostre telèfon, ja que la clau de generació de codis entre LinOTP i Google Authenticator és la mateixa, i per tant en el moment de l'autenticació el codi es validarà en el servidor per comprovar que és correcte.[Fig.13]

Actualment ens demana el password d'usuari i el Verification code, que és el codi que tindrem generat en el nostre telèfon mòbil i que haurem d'introduir. Si tot és correcte la connexió queda establerta. De la mateixa manera que hem fet anteriorment, fem una prova creant un nou directori "prova2"al directori Escritorio, i tanquem la connexió.[Fig.11][Fig.12]

Per últim comprovem a la màquina de destí que efectivament tenim creat el nou directori "prova2"a l'escriptori.

8 Conclusions

Després de l'estudi teòric i del desenvolupament pràctic del projecte, podem dir que s'han aconseguit els objectius generals. En l'àmbit personal he pogut aprendre a autogestionar el temps i abordar els canvis que s'han produït. Això m'ha permès adquirir nous hàbits de treball i de pensament.

En quant al tema principal del projecte, que consistia en la implantació d'un sistema de doble autenticació mitjançant OTPs. Hem pogut realitzar una investigació teòrica sobretot el que comporta un sistema d'aquest tipus. S'han pogut estudiar els tipus d'autenticacions que existeixen en l'actualitat, els avantatges i inconvenients de cadascun d'ells i en concret els de doble factor. També hem pogut estudiar els algoritmes i protocols que implanten aquest tipus de sistemes, on són necessaris processos d'encriptació d'informació, processos de transmissió d'informació de forma segura, entre d'altres. Amb tot això hem pogut estudiar diversos sistemes de tipus OTP, per poder escollir-ne un a implantar en el nostre projecte.

En la part pràctica del desenvolupament, OTPClient va ser el programari escollit inicialment, ja que en primer lloc es un programa totalment de codi obert i implementa totes les funcionalitats que poden tenir les autenticacions de tipus OTP. Base de dades encriptada en local, creació de codis tipus TOTP i HOTP, creació de tokens de forma manual o pre-configurada mitjançant codis QR, etc. Però conforme avançava el desenvolupament pràctic OTPClient no oferia totes les possibilitats que es necessitaven per dur a terme un simulació en un entorn real. Per això es va decidir canviar a LinOTP, un programa molt més complet i complex que permetia un ventall més gran de possibilitats. Per el nostre cas vam fer us de la instal·lació base de LinOTP i dels mòduls de PAM per realitzar l'autenticació per SSH. Amés també vam utilitzar el mòdul de enviament de correus per simular aquest entorn real de la universitat.

Per poder crear un entorn de proves on implementar aquest sistema es va fer ús de l'eina VirtualBox, que ens permetia una virtualització d'un entorn Linux. En concret unes distribucions de tipus Ubuntu de Linux, ja que són de les més utilitzades.

La implementació de LinOTP es va desenvolupar de forma satisfactòria, les proves sobre les seves funcionalitats es van realitzar mitjançant connexions SSH entre dues màquines virtuals.

Aquest projecte pot ser un exemple d'un cas d'us en el que es podria aplicar una barrera més de seguretat en un entorn educatiu general. LinOTP permet d'altres formes d'implementació i d'un ventall de possibilitats molt ampli. Agafant la base de coneixements d'aquest projecte es podria proposar la millora i evolució de les funcionalitats, com podrien ser implementar el mateix sistema per l'autenticació dels alumnes directament al campus virtual.

Agraïments

En primer lloc nomenar al meu tutor del treball de fi de grau Àngel Elbaz, que m'ha guiat durant tot el procés i m'ha donat bones indicacions a l'hora de realitzar el projecte.

A la meva mare Antònia, que va ser de gran ajuda per veure si el projecte s'estava redactant de manera entenedora

per una persona no del sector.

I a la meva parella Vicky per també la col·laboració en què el projecte es realitzés d'una manera entenedora des de un punt de vista d'una persona que coneix l'àmbit de la Universitat i del grau en Enginyeria Informàtica.

REFERÈNCIES

[1] Digicert.com. 2020. Two-Factor Authentication — Digicert.Com. [online] Disponible a: https://www.digicert.com/kb/tfa/two-factor-authentication.htm [Consultat el 26 Setembre 2020].

[2] Evidian.com. 2020. [online] Disponible a: https://www.evidian.com/pdf/wp-strongauth-es.pdf [Consultat el 26 Setembre 2020].

[3] Cybersecurity.ieee.org. 2020. Design Best Practices For An Authentication System – IEEE Cybersecurity. [online] Disponible a: https://cybersecurity.ieee.org/blog/2016/06/02/design-best-practices-for-an-authentication-system/ [Consultat el 26 Setembre 2020].

[4] INCIBE. 2020. Dos Mejor Que Uno: Doble Factor Para Acceder A Servicios Críticos. [online] Disponible a: https://www.incibe.es/protege-tu-empresa/blog/dos-mejor-uno-doble-factor-acceder-servicios-criticos [Consultat el 26 Setembre 2020].

[5] Kaspersky.es. 2020. La Autentificación De Doble Factor En SMS Es Insegura, ¿Qué Alternativas Hay?. [online] Disponible a: https://www.kaspersky.es/blog/2fa-practical-guide/17187/ [Consultat el 26 Setembre 2020].

[6] "Secure Hash Algorithm", Es.wikipedia.org, 2020. [Online]. Disponible a: https://es.wikipedia.org/wiki/SecureHashAlgorithm. [Consultat el 26 Octubre 2020].

[7] "SHA-1", En.wikipedia.org, 2020. [Online]. Disponible a: https://en.wikipedia.org/wiki/SHA-1. [Consultat el 26 Octubre 2020].

[8] "MD5", En.wikipedia.org, 2020. [Online]. Disponible a: https://es.wikipedia.org/wiki/MD5. [Consultat el 26 Octubre 2020].

[9] "AES", En.wikipedia.org, 2020. [Online]. Disponible a: https://es.wikipedia.org/wiki/AdvancedEncryptionStandard [Consultat el 26 Octubre 2020].

[10] L·LinOTP -", Linotp.org, 2020. [Online]. Disponible a: https://www.linotp.org/. [Consultat el 23 Octubre 2020].

[11] "How to install and use the open source OTPClient 2FA tool on Linux", TechRepublic, 2020. [Online]. Disponible a: https://www.techrepublic.com/article/how-to-install-and-use-the-open-source-otpclient-2fa-tool-on-linux/. [Consultat el 23 Octubre 2020].

[12] "Welcome to VirtualBox", VirtualBox, 2020. [Online]. Disponible a: https://www.virtualbox.org/ [Consultat el 19 Novembre 2020].

[13] Übuntu Desktop", En. https://ubuntu.com/download/desktop, 2020. [Online]. Disponible a: https://ubuntu.com/download/desktop. [Consultat el 19 Novembre 2020].

APÈNDIX

A.1 Instl·lació de LinOTP

1. Com que som usuaris amb permisos root, en primera instancia aplicarem un login en consola però no haver d'autenticar-nos a cada pas següent.

   sudo -i

2. Afegirem el repositori per poder descarregar la informació.

   add-apt-repository ppa:linotp/stable

3. Actualitzem el repositori i la resta de paquets per si hi ha alguna modificació.

   apt-get Update

4. Fem la instal·lació de una base de dades per emmagatzemar la informació necessaria. En el nostre cas triem un MyQSL.

   apt-get install mysql-server

5. Fem instal·lació bàsica de LinOTP abans d'afegir els paquets específics per a cada situació.

   apt-get install linotp

6. Un cop realitzats aquets passos podem accedir al nostre cuadre de comandament de LinOTP mitjançant la següent URL:

   http://iplocalhost/manage

7. Per habilitar l'encriptació i autenticació del servidor de LinOTP, es pot utilitzar com un servidor web. Seguidament instal·lem i configurem Apache2 per fer servir LinOTP de una manera encriptada i autenticada.

   apt-get install apache2

   apt-get install libapache2-mod-wsgi

8. Automàticament es crea una configuració bàsica a la ruta

   /usr/share/doc/linotp/examples/apache2-linotp2

   Copiarem aquesta configuració al nostre directori de configuracions

/etc/apache2/available/linotp2

9. Ens desplaçarem al directori esmentat anteriorment

   cd /etc/apache2/available/

10. I activem la configuració

    a2ensite linotp2

11. Un cop realitzada la configuració base de LinOTP afegirem els moduls especifics per el model que volem dur a terme. Haurem d'habilitar el modul PAM Authentication. Descarregarem l'ultima versió disponible del paquet que podem trobar al repositori GitHub de LinOTP

    LinOTP-Release-2.12.1-.tar.gz

    Descomprimirem l'arxiu

    Tar –xzvf LinOTP-Release-2.12.1-.tar.gz

    I realitzarem un canvi de directori

    Cd LinOTP-Release-2.12.1/authmodules

    Descomprimirem el arxiu següent

    tar –xzvf LinOTPAuth.tar.gz

    I ens mourem al directori

    Cd LinOTPAuth/libpam-linotp/src/

    A continuació executarem les següents comandes per compilar, configurar i instal·lar el mòdul PAM

    Libtoolize

    Aclocal

    Automake –add-missing

    Autoconf

    ./configure

    Make install

    Actualment tindrem el mòdul PAM instal·lat al directori

    /lib/security/pamlinotp.sp

12. Per acabar de configurar pamlinotp utilitzarem la configuració predefinida i recomanada per els desenvolupadors, la podem trobar en l'arxiu

    /etc/pam.d/auth-common

    Realitzarem una copia i el renombrarem a

    Auth-linotp

A.2 Instal·lació de SSH Server en l'amfitrió

1. Introduirem les següents comandes per instal·lar openssh server a la maquina amfitrió

   Apt-get update

   Apt-get install openssh-server

2. Amb la següent comanda provarem que el sistema de server ssh està correctament configurat.

   Systemctl status ssh

   Haurem de comprovar que l'estat de la casella "Active" es trobi en l'estat "Running".

A.3 Instal·lació de SSH Client en el client

1. Introduirem les següents comandes per instal·lar openssh server a la maquina client

   Apt-get update

   Apt-get install openssh-client

2. Amb la següent comanda provarem que el sistema de server ssh està correctament configurat.

   Systemctl status ssh

   Haurem de comprovar que l'estat de la casella "Active" es trobi en l'estat "Running".

A.4 Configuració dels adaptadors de xarxa de VirtualBox

Per la nostre simulació realitzarem una comunicació entre les dues màquines virtuals que ens servirà per poder establir la comunicació mitjançant el port d'escolta de SSH o el port d'escolta a traves de IP.

Per realitzar la configuració de ports apagarem les màquines virtuals i accedirem al menú de configuració, menú de xarxa, adaptador 1 i opcions avançades. Accedirem al menú de reenviament de ports i afegirem una nova entrada.[Taula.1]

D'aquesta forma podrem accedir mitjançant SSH a la maquina virtual per simular el procés d'autenticació.

En un cas d'accés des de una altre màquina s'hauria d'especificar la ip publica i el port 8080 per accedir mitjançant Internet.

A.5 Fotografies per comprovació de funcionalitats

TAULA 1: CONFIGURACIÓ VBOX

Protocol	IP amfitrió	Port amfitrió	IP Convidat	Port Convidat
TCP	127.0.0.1	2222	10.0.2.15	22

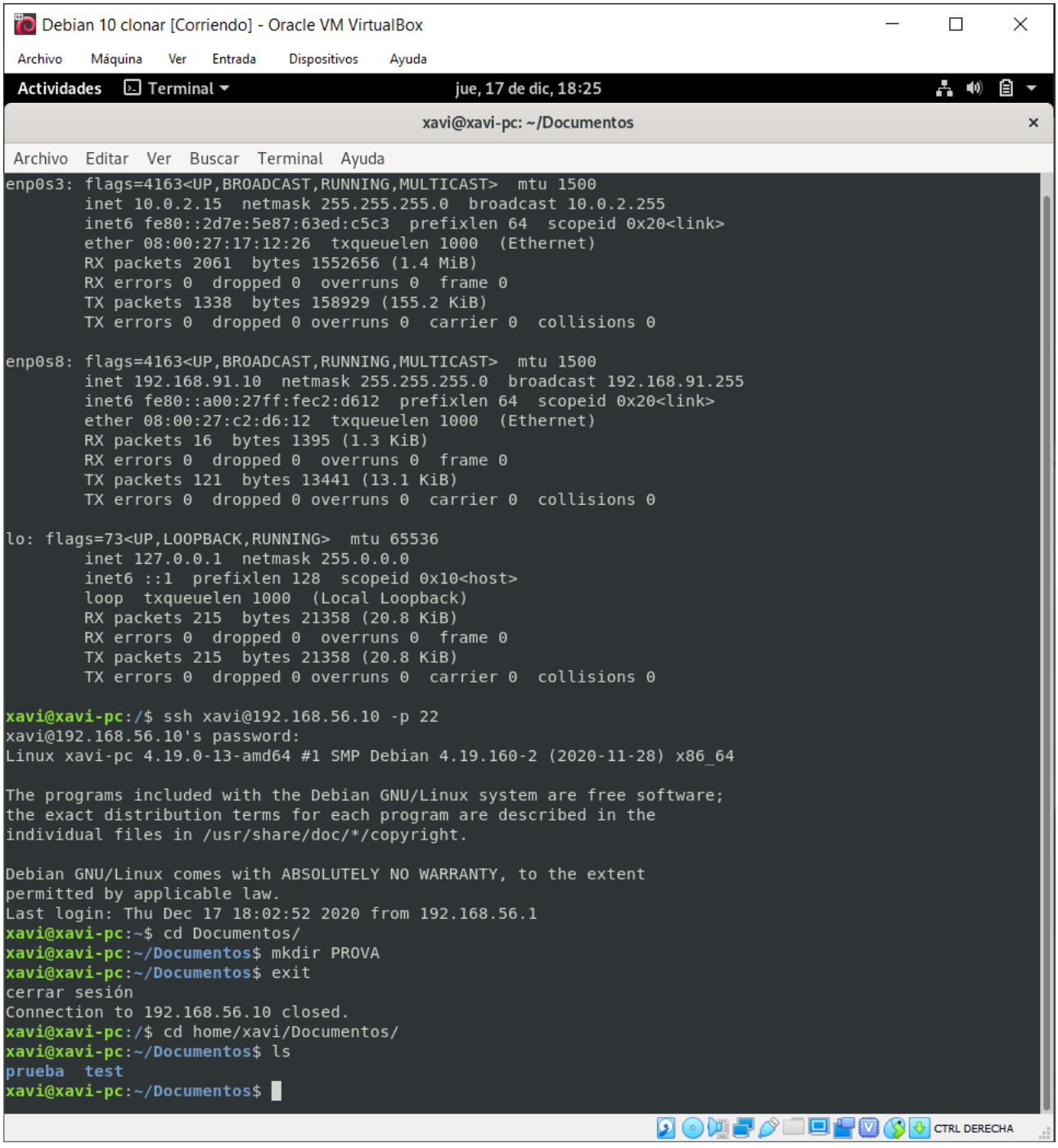

Fig. 9: Autenticació sense OTP

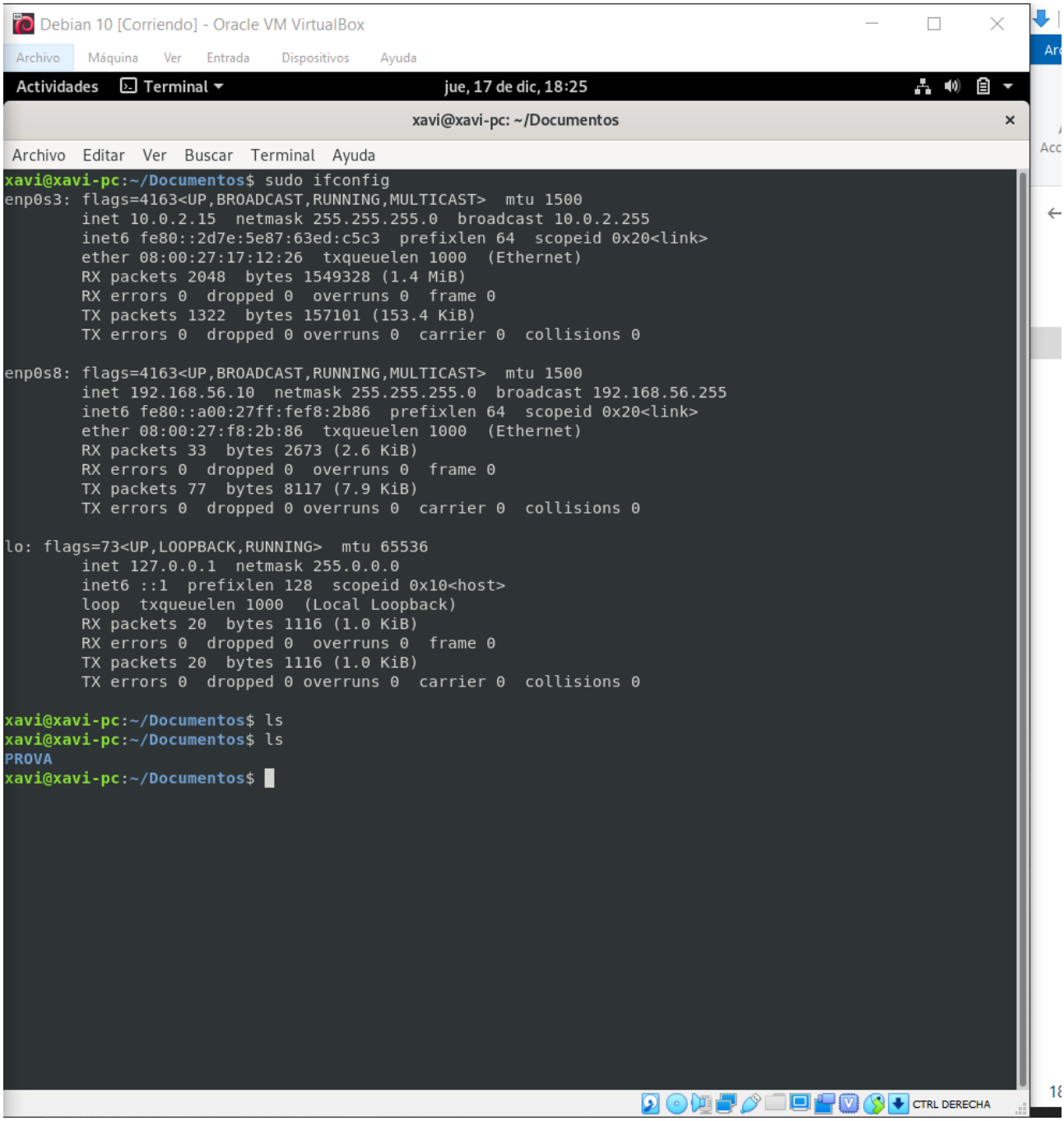

Fig. 10: Comprovació de SSH sense OTP

```
xavi@xavi-pc:~/Documentos$ ssh xavi@192.168.56.10 -p 22
Password:
Verification code:
Linux xavi-pc 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 17 18:23:08 2020 from 192.168.56.1
xavi@xavi-pc:~$ ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  Vídeos
xavi@xavi-pc:~$ cd Es
-bash: cd: Es: No existe el fichero o el directorio
xavi@xavi-pc:~$ cd Escritorio/
xavi@xavi-pc:~/Escritorio$ ls
xavi@xavi-pc:~/Escritorio$ mkdir prova2
xavi@xavi-pc:~/Escritorio$ exit
cerrar sesión
Connection to 192.168.56.10 closed.
```

Fig. 11: Autenticació amb OTP

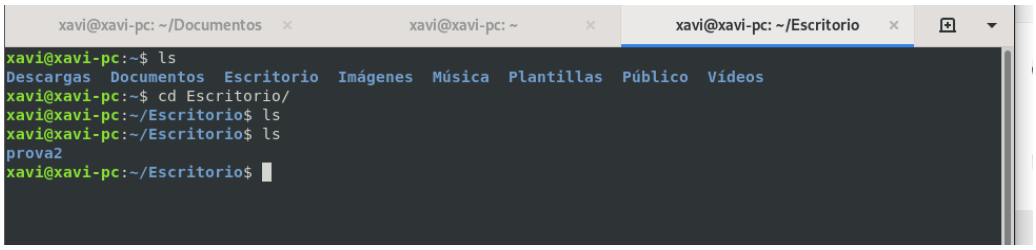

Fig. 12: Comprovació de SSH amb OTP

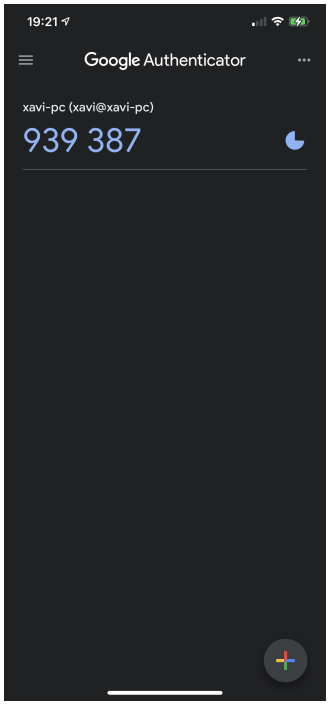

Fig. 13: Generador de OTP