Secure PIN library

Diana Mirila

Resum– La seguretat de les nostres dades ha de ser una prioritat. Aquesta prioritat no només ha de ser dels desenvolupadors, sinó que també ha de ser nostre. Tant els desenvolupadors d'aplicacions han de ser capaços de protegir les nostres dades, com els usuaris també hem de ser capaços, amb les eines donades pels desenvolupadors, de protegir o mitigar alguns aspectes delicats del nostre ús de la tecnologia. Per aquesta raó s'ha decidit dur a terme un projecte en el qual es pogués introduir una contrasenya o pin, el qual es pot reutilitzar i implementar en diverses aplicacions i que permeten als usuaris introduir la seva contrasenya sense perill de què aquesta pugui ser observada per d'altres.
Paraules clau– Seguretat; llibreria; aplicacions mòbils; pin; Android.

Abstract– The security of our data must be a priority. This priority should not only belong to the developers, but it should also be ours. Both application developers must be able to protect our data, but users must also be able, with the tools given by developers, to protect or mitigate some sensitive aspects of our use of technology. For this reason, we has decided to carry out a project in which a password or pin could be entered, which can be reused and implemented in various applications and which allow users to enter their password without the risk of it being observed by others.

Keywords– Security; library; movile applications; pin; Android.

1 Introducció

AQUEST treball de final de grau ve proposat per l'empresa Build38, la qual ens proposa desenvolupar una llibreria que permeti la introducció d'un PIN de seguretat en diverses aplicacions mòbils. Tanmateix, l'empresa també suggereix fer un estudi de vulnerabilitats per tal de definir quines mesures s'han de tenir en compte a l'hora de pensar en la seguretat de les aplicacions mòbils.

Build38 és una empresa que utilitza una tecnologia basada en la intel·ligència artificial que va desenvolupar una de les mesures de protecció d'aplicacions més efectives del mercat. Aquesta mesura té les aplicacions de protecció contra codi maliciós (malware), hackers i cibercriminals.

Segons la mateixa empresa, ells proveeixen una solució programari dissenyada que permet el creixement econòmic, una cosa molt necessària avui dia. Build38 està especialitzat en seguretat mòbil per a un entorn global.[1]

Els beneficis que l'empresa comenta de l'ús de la llibreria creada per ells són, entre d'altres, els següents:

- Una integració fàcil i ràpida: La integració permet a les aplicacions aplicar un mode de protecció propi usant el SDK de l'empresa.
- Sempre al dia: Un cop la solució està integrada, les aplicacions rebran constantment les darreres actualitzacions de seguretat i també estan constantment monitoritzades.

Avui dia, les aplicacions mòbils són la base de l'experiència digital que no només són utilitzades per a l'oci sinó també per al negoci. Així que cal comentar que a mesura que el nombre d'aplicacions creix implica noves amenaces en la seguretat que apareixen.

La seguretat a les aplicacions es veu actualment com un requisit d'aquestes; no obstant això, moltes vegades no som conscients de la importància de mantenir unes mesures de seguretat a les aplicacions. Segons l'article de Build38, una sola bretxa de seguretat en una aplicació pot costar al negoci de mitjana uns 4,35 milions de dòlars.[2]

Així mateix, també hem de comentar que els ciberatacs han augmentat un 600%, fet que comporta que hi ha noves maneres d'explotar una possible vulnerabilitat que tingui l'aplicació que s'hagi desenvolupat.

Segons un estudi realitzat per Osterman, més del 85% de les aplicacions comercials tenen vulnerabilitats crítiques que poden ser explotades de manera maliciosa.

Per acabar amb les estadístiques, un estudi publicat per Trustwave ens mostra que un 99,7% de les aplicacions web tenen almenys un buit a la seguretat.

Tenint en compte que un mòbil fa servir de mitjana 9 aplicacions al dia i 30 aplicacions al mes, això ens indica la

• E-mail de contacte: 1496855@uab.cat
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Jordi Ventayol Marimon
• Curs 2022/23

criticitat de l'assumpte.

La seguretat, o millor dit la manca d'ella, es pot mostrar usant el codi CVE, i aquestes són les sigles de Çommon Vulnerabilities and Exposures", és un llistat de vulnerabilitats de seguretat conegudes. En aquest llistat, les entrades tenen un codi únic i es detalla una descripció de la vulnerabilitat, que versions de programari afecten, en cas que n'hi hagi, possibles solucions, etc.[3]

2 MOTIVACIÓ I OBJECTIUS

La motivació inicial d'haver escollit aquest treball de fi de grau és deguda al fet que en haver cursat la menció de Tecnologies de la Informació, la seguretat a les xarxes i a les aplicacions és una cosa que sempre he considerat important i interessant, i per això m'agradaria aprofundir en aquest tema. Considero que amb aquest treball de finalització de grau es podrà, no només aprofundir en els coneixements vistos durant les assignatures de la carrera, sinó també desenvolupar característiques per a la llibreria que es considerin necessàries per millorar la seguretat.

Pel que fa als objectius, hem fet una divisió entre l'objectiu principal, el qual és desenvolupar una llibreria Android que aporti seguretat als desenvolupadors d'aplicacions i usuaris quan hagin d'introduir un PIN o dades sensibles. I els objectius secundaris, essent aquests els següents:

- Analitzar les vulnerabilitats a què s'enfronta una aplicació Android.
- Analitzar les característiques de seguretat necessàries per a la protecció de l'entrada de dades.
- Analitzar quines de les mesures de seguretat aporta la llibreria desenvolupada per l'empresa Build38 i quines haurien de ser implementades per la llibreria nova a desenvolupar.
- Desenvolupar la característica per a la introducció del pin que incorpori la protecció de la llibreria de l'empresa Build38 així com la desenvolupada en aquest projecte.
- Aprofundir en els conceptes vists durant els diversos cursos.
- Implementar les mesures de seguretat que no aporti la nostra llibreria T.A.K.

3 METODOLOGIA, PLANIFICACIÓ I PROCÉS DE DESENVOLUPAMENT

La metodologia que s'ha plantejat seguir és scrum, aquesta és una metodologia incremental que divideix els requisits i les tasques de manera similar a Kanban. S'itera sobre blocs de temps curts i fixos, entre dues i quatre setmanes, per aconseguir un resultat complet a cada iteració. El fet que la divisió dels blocs sigui entre dues i quatre setmanes fa que s'ajusti a la divisió dels lliuraments parcials d'aquest projecte.

Les etapes que es fan servir en aquesta metodologia són: planificació de la iteració (planning sprint), execució (sprint), reunió diària (daily meeting) i demostració de resultats (sprint review). Cada iteració per aquestes etapes s'anomena també sprint. L'sprint de planificació correspon al primer lliurament parcial, on es planteja el projecte, la resta dels lliuraments corresponen a la resta dels lliuraments parcials, la reunió diària es canviaria per una reunió setmanal i finalment l'esprint review seria la presentació del treball fet a la tutora als lliuraments parcials i la presentació final en acabar aquest projecte.[**?**]

Tanmateix, a causa de les dificultats trobades en la realització del projecte, aquesta metodologia no s'ha pogut seguir segons com s'havia plantejat.

Les etapes que es van plantejar seguir en aquesta metodologia són, tal com s'ha esmentat anteriorment, les següents:

1. Planificació de la iteració (planning sprint)
2. Execució (sprint)
3. Reunió diària (daily meeting)
4. Demostració de resultats (spring review)

Cada iteració per aquestes etapes s'anomena també sprint. L'sprint de planificació correspon al primer lliurament parcial, on es planteja el projecte, la resta dels lliuraments corresponen a la resta dels lliuraments parcials, la reunió diària es canviaria per una reunió setmanal i finalment l'esprint review seria la presentació del treball fet a la tutora als lliuraments parcials i la presentació final en acabar aquest projecte. Tal com podem observar en la planificació inicial, representada en la figura [Apendice 5] 5.

En la figura 5 les activitats numerades són les següents:

1. Estudi de l'empresa Build38 i de les alternatives que tenen al mercat.
2. Anàlisi de vulnerabilitats d'una aplicació Android.
3. Anàlisi de mesures de seguretat que aporta la llibreria T.A.K de Build38.
4. Definir funcionalitats a desenvolupar.
5. Desenvolupar la introducció del pin de seguretat.
6. Implementar mesures de seguretat que aporti la llibreria T.A.K
7. Implementar mesures de seguretat que no aporti la llibreria T.A.K
8. Aprofundiment en els coneixements sobre la seguretat de les aplicacions.
9. Implementar nous requisits que calguin.
10. Testeig de la llibreria.
11. Testeig i proves amb usuaris reals.
12. Implementació de la versió final.
13. Redactar documentació

Tal com s'ha esmentat anteriorment, no s'ha pogut seguir la planificació inicial a causa de problemes en la realització del projecte. I, per tant, la realització del projecte s'ha realitzat seguint una planificació com la de la figura[Apendice 5] 6.

4 CONCEPTES BÀSICS DE SEGURETAT

Abans de començar amb els objectius del projecte, es ve convenient fer una breu introducció a conceptes clau per tal de poder donar una base a l'anàlisi de vulnerabilitats del següent punt:

En ciberseguretat es parla d'amenaces, vulnerabilitats i risc, però per a moltes persones aquests conceptes no queden clars i solen confondre'ls; per això s'ha considerat fer una diferència clara d'aquests tres conceptes:

Vulnerabilitat: És una debilitat pròpia d'un sistema que permet ser atacat i rebre un dany. Les vulnerabilitats es produeixen de forma habitual per una baixa protecció contra-atacs externs, manca d'actualitzacions, fallades de programació i altres causes similars. A les vulnerabilitats també se les coneix com a forats de seguretat i tenen l'avantatge que poden ser solucionades un cop siguin descobertes. Una vulnerabilitat posa en risc les dades i els sistemes d'una empresa comprometent-ne la integritat, la privadesa i la disponibilitat.[4]

Per altra banda, una amenaça és la possibilitat que un sistema vulnerable sigui atacat i pateixi danys. Les amenaces d'un sistema informàtic provenen principalment d'atacs externs (malware, denegació de servei o injeccions SQL, entre d'altres), de no complir les polítiques de seguretat (connectar dispositius no autoritzats a la xarxa o utilitzar contrasenyes febles) i esdeveniments inesperats (com incendis o robatoris físics, per exemple). A continuació es mostra alguna de les amenaces més comunes:[5]

- Codi maliciós: Aquests atacs codi maliciós ataquen dispositius i servidors per tal de robar informació delicada, com dades bancàries o credencials d'accés. Els atacs ransomware són una de les amenaces més grans avui en dia per als sistemes informàtics de les empreses.
- Robatori d'identitat. Una altra amenaça que posa en risc els sistemes d'una organització és el phishing o el robatori d'identitat. L'amenaça consisteix a enganyar l'usuari perquè faciliti de manera involuntària les credencials d'accés a un tercer que les farà servir de forma fraudulenta.
- Amenaces Persistents Avançades. Són atacs coordinats que es dirigeixen a una empresa per robar les dades. Són una de les amenaces més difícils de detectar, ja que fan servir tècniques d'enginyeria social.
- Denegació de servei: Els atacs DDoS són una amenaça que plana sobre servidors que pretenen ser col·lapsats enviant-los una enorme quantitat de peticions (fent que no puguin atendre-les, i fins i tot que acaben caient).
- Negligència. Els usuaris solen ser l'amenaça més gran per a un sistema informàtic. Els errors humans i el no incompliment de les polítiques i les normes de seguretat de l'empresa posen en perill els sistemes i les dades de l'empresa.

Finalment, els riscos són la possibilitat que un sistema pateixi un incident de seguretat i que una amenaça es materialitzi causant una sèrie de danys. Per mesurar el risc d'un sistema informàtic cal assumir que hi ha una vulnerabilitat davant d'una amenaça. El risc és, doncs, la probabilitat que l'amenaça es materialitzi aprofitant una vulnerabilitat existent.[6]

5 ANÀLISI DE VULNERABILITATS

Al món de la informàtica, una vulnerabilitat és una debilitat existent en un sistema que pot ser utilitzada per una part malintencionada per comprometre la seva seguretat. Aquestes poden ser de tipus maquinari, programari, procedimentals o humanes i poden ser explotades o fetes servir per intrusos o atacants. Algunes de les vulnerabilitats més comunes en els sistemes informàtics són les següents:

- Buffer overflow: error de programari que es produeix quan un programa no controla adequadament la quantitat de dades que es copien sobre una àrea de memòria reservada a aquest efecte (buffer). Si aquesta quantitat és superior a la capacitat preassignada, els bytes sobrants s'emmagatzemen en zones de memòria adjacents, i en sobreescriuen el contingut original, que probablement pertanyien a dades o codi emmagatzemats en memòria. Això constitueix una fallada de programació.
- Condició de carrera: Es dona quan un o més threads executen seccions que un altre thread podria fer servir alhora, i en aquestes modifica variables que podrien afectar l'execució de la resta de threads. Es poden produir quan diversos threads no accedeixen en exclusió mútua a un recurs compartit.
- Error de format en cadenes: error que passa quan la cadena de format printf(%d, %s) fet ús a la printf()funció s'usa de manera incorrecta. Això és perquè l'ordinador reconeix el valor d'entrada com a caràcter de format en lloc d'un caràcter. Aquesta vulnerabilitat és molt comuna en el llenguatge de programació C/C++.
- Cross Site Scripting: tipus d'injecció, en què s'afegeixen scripts maliciosos en llocs web que, altrament, serien benignes i fiables. Els atacs XSS ocorren quan un atacant utilitza una aplicació web per enviar codi maliciós, generalment en forma de seqüència d'ordres del costat del navegador, a un usuari final diferent. Les falles que permeten que aquests atacs tinguin èxit estan força esteses i ocorren a qualsevol lloc on una aplicació web faci servir l'entrada d'un usuari dins de la sortida que genera sense validar-la o codificar-la.
- Injecció de SQL: La injecció de SQL generalment passa quan se us demana a un usuari que introduïu una dada, com el vostre nom d'usuari/ID d'usuari, l'usuari li dona una instrucció SQL que, sense saber-ho, s'executarà a la base de dades.

Tot i que cal comentar que aquestes vulnerabilitats no només apliquen a les aplicacions web, sinó que són vulnerabilitats del sistema informàtic.

Tal com es comenta quan es defineix el concepte de vulnerabilitat, aquestes també poden ser humanes, amb aquest tipus de vulnerabilitat a allò que ens referim és a una vulnerabilitat on el problema està en l'ús que li dona l'usuari i com això pot afectar a la seguretat de la plicació.

En aquesta categoria podria entrar qualsevol vulnerabilitat que depengui de l'usuari com ara: Ús de contrasenya o pin poc segur, quan l'usuari utilitza una contrasenya o pin poc segur, aquest pot ser fàcilment endevinat en cas que sigui una contrasenya o pin amb un significat per a l'usuari o bé si la contrasenya o pin és fàcilment fisurable. Ús de la contrasenya o del pin en un lloc públic i visible a ulls d'algun possible atacant. Aquesta vulnerabilitat fa que si una persona aliena observi la posició de les tecles i aconseguirho amb la contrasenya o amb el pin.

Aquestes vulnerabilitats, però, afecten més a les aplicacions i/o pàgines web. Les vulnerabilitats descrites a continuació són més rellevants en les aplicacions mòbils.

Cal tenir en compte que les aplicacions mòbils, com també les webs, poden sofrir de vulnerabilitats en el codi. Entre les vulnerabilitats que poden ser presents en les aplicacions mòbils tenim les següents:

- Intents de còpia del codi font: Donat que les aplicacions mòbils solen emmagatzemar informació dels usuaris, aquesta informació ha d'estar ofuscada per tal que no sigui de fàcil accés i, per tant, segura.[7]

- Les crides que fa l'aplicació al servidor i viceversa, també poden ser un aspecte sensible de les aplicacions mòbils. Per aquesta raó, es treballa la protecció de les trucades de Webservice per evitar perills. Aquest és un focus perillós en l'àmbit de les aplicacions mòbils.[7]

- Falta de SSL. Igual que passa amb les pàgines web amb el certificat SSL, si una aplicació no en disposa, no és segura. Per això és important que quan una empresa desenvolupa una aplicació mòbil, disposi del certificat SSL. [7] SSL és l'acrònim de Secure Sockets Layer (capa de sockets segurs), la tecnologia estàndard per mantenir segura una connexió a Internet, així com per protegir qualsevol informació confidencial que s'envia entre dos sistemes i impedir que es llegeixin i modifiquin qualsevol dada que transfereixi, inclosa informació que pogués considerar-se personal. [8]

- Encriptació dèbil és quan la comunicació a través de la xarxa és sense cap mena de xifratge o un de molt feble que no garanteixen la privadesa i seguretat de les converses. Es tracta d'un aspecte molt important a tenir en compte durant el desenvolupament de l'aplicació mòbil, ja que si l'algoritme d'encriptació no és prou fort serà un blanc fàcil per als hackers.[7]

- Codi obert: Quan desenvolupem una aplicació mòbil amb codi obert, ens exposem a riscos que aquest codi sigui utilitzat per explotar el codi i l'aplicació.

- Hooking: És el conjunt de tècniques emprades per alterar el comportament d'un programari mitjançant la intercepció de les trucades a funcions, missatges o esdeveniments entre components. És a dir, que mitjançant l'ús d'eines com Frida, Intropsy o Xposed Framework es pot modificar el comportament del sistema sense necessitat de modificar l'APK.[9]

- Rooting: Els atacants poden 'rootejar' un dispositiu per eludir la zona de proves de l'aplicació Android. Això pot permetre l'accés a les dades emmagatzemades al dispositiu que, de manera normal, tindrien accés restringit. De manera similar, el programari maliciós pot aprofitar les debilitats conegudes d'Android per obtenir permisos elevats en un dispositiu mentre s'executa. Hi ha dos tipus de rooting: [10]

  - Soft root: Es basa en una vulnerabilitat d'escalada de privilegis al nucli de Linux o en una aplicació que s'executa com a root. Quan l'eina que realitza el rooting ha aconseguit permisos de root, té accés il·limitat al sistema de fitxers. Això generalment ho fan les eines One Click, que s'instal·len al dispositiu i activen la vulnerabilitat en iniciar-se.[10]
  - Hard root: Es basa en la capacitat d'actualitzar el microprogramari del dispositiu. Això també permet l'accés complet al sistema de fitxers. Una arrel dura requereix un dispositiu que tingui un carregador d'arrencada que es pugui desbloquejar o una vulnerabilitat al carregador d'arrencada.[10]

Pot ser l'exemple més clar de vulnerabilitat que va afectar a gran escala, no només a les aplicacions mòbils sinó a totes les aplicacions que feien servir la llibreria, és la vulnerabilitat de log4j.

Log4Shell, la qual està recollida en les següents entrades del CVE: CVE-2021-44228, CVE-2021-45046 i CVE-2021-45105, és una vulnerabilitat d'execució remota de codi (RCE) que permet als agents maliciosos executar codi Java arbitrari, prenent el control d'un servidor de destinació.[11]

Aquesta vulnerabilitat es va comunicar per primera vegada a Apache el 24 de novembre de 2021.

Log4Shell és una vulnerabilitat per injecció de Java Naming and Directory Interface™ (JNDI), que permet l'execució remota de codi (RCE). En incloure dades que no són de confiança al missatge registrat d'una versió afectada d'Apache Log4j, un atacant pot establir una connexió a un servidor maliciós mitjançant una cerca de JNDI. El resultat: accés complet al sistema des de qualsevol lloc del món.[11]

Atès que la cerca de JNDI admet diferents tipus de directoris, com el servei de noms de domini (DNS), el protocol lleuger d'accés a directoris (LDAP), el qual proporciona informació valuosa com els dispositius de xarxa de l'organització, la invocació del mètode remot (RMI) i el protocol inter-ORB (IIOP), Log4Shell pot portar a altres amenaces com per exemple, randsomware, denegació de servei, entre d'altres.[11]

Algunes de les vulnerabilitats descrites en aquest apartat tenen solució gràcies a la llibreria T.A.K, tal com descriurem en el següent apartat.

6 MESURES DE SEGURETAT DE LA LLIBRERÍA T.A.K

La llibreria desenvolupada per l'empresa Build38 ofereix una solució de seguretat que s'executa en dispositius Android i iOS, implementada a una biblioteca fàcil d'integrar.

La llibreria T.A.K ofereix als usuaris finals una protecció dels vostres comptes i dades personals, en cas que el dispositiu hagi estat robat o el codi maliciós s'hagi obert camí al

dispositiu. Dit d'una altra manera, una solució de seguretat que respecta la seva privadesa.

Dins les mesures que aquesta llibreria ens proporciona podem trobar les següents:

- Emmagatzematge segur: aquesta funcionalitat permet emmagatzemar dades d'usuari que es generen després que l'aplicació s'instal·la al dispositiu de l'usuari com ara la baixada de dades d'Internet. Això és degut al fet que la llibreria proporciona una base de dades xifrada vinculada al dispositiu per emmagatzemar parells clau-valor, que es xifra utilitzant claus diferents per a cada usuari final.[12]

- Generació de signatura: T.A.K proporciona la possibilitat de generar signatures sobre la base de dos algorismes de signatura:

    1. ECDSA: Elliptic Curve Digital Signature Algorithm és una variant de l'algorisme DSA (Digital Signature Algorithm) que utilitza operacions sobre punts de corbes el·líptiques. El principal avantatge d'aquest esquema és que requereix números de mida més petits per brindar la mateixa seguretat que DSA o RSA.[12]
    2. ECDSA: Elliptic Curve Digital Signature Algorithm és una variant de l'algorisme DSA (Digital Signature Algorithm) que utilitza operacions sobre punts de corbes el·líptiques. El principal avantatge d'aquest esquema és que requereix números de mida més petits per brindar la mateixa seguretat que DSA o RSA.[12]

- La llibreria T.A.K disponible de dues claus per generar signatures:

    1. La clau privada individual: només disponible després del registre, aquesta clau és única per a cada instància de T.A.K i permet provar que algunes dades es van signar en un dispositiu en concret.[12]
    2. La clau privada per defecte: sempre disponible, compartida entre totes les instàncies amb la mateixa compilació T.A.K. Permet provar que algunes dades van ser signades per una aplicació legítima, és a dir que les dades no han estat manipulades.[12]

- Protector de fitxers: funcionalitat que permet xifrar fitxers per avançat, abans d'afegir-los al projecte. En temps d'execució, els fitxers de recursos xifrats es poden desxifrar mitjançant l'SDK del client T.A.K. Això fa que sigui més difícil per a un atacant extreure o obtenir fitxers confidencials de l'aplicació mòbil, si intenta aplicar enginyeria inversa.[12]

- Canal Segur: funcionalitat que protegeix la comunicació de xarxa entre l'aplicació mòbil i els servidors remots segons l'estàndard TLS. La implementació de T.A.K s'enforteix amb:[12]

    1. Totes les comprovacions de temps d'execució proporcionades per T.A.K
    2. Fer complir l'ús de la fixació de certificats
    3. Habilitar la possibilitat d'habilitar l'autenticació TLS del client, cosa que augmenta la seguretat al costat del backend en garantir que la comunicació provingui de punts finals fiables.

- Hooking: Hooking és un mitjà per executar una funció, sigui abans o després de l'execució de l'aplicació, en lloc del codi existent. Per exemple, es pot escriure una funció per captar el procés d'inici de sessió d'una aplicació mòbil i alterar el flux de treball d'autenticació, sol·licitant als usuaris que tornin a escriure la contrasenya.[12]

  Un altre nom per a aquest tipus d'amenaça és tampering, que pot ser dinàmic o estàtic. Tampering significa fer pegats o canvis en el temps d'execució de l'aplicació per afectar-ne el comportament. Per exemple, és possible que vulgueu desactivar la fixació de SSL o les proteccions binàries que dificulten el procés de prova. La instrumentació de temps d'execució inclou l'addició de ganxos i pegats de temps d'execució per observar el comportament de l'aplicació. Tot i això, en la seguretat de les aplicacions mòbils, el terme es refereix vagament a tota classe de manipulació del temps d'execució, inclosos els mètodes d'anul·lació per canviar el comportament.

  Tal com s'ha comentat anteriorment, el tampering pot ser dinàmic, el qual es fa durant l'execució del codi, o bé estàtic, el qual no es realitza en temps d'execució.

- Root detection or Jailbreak detection: La llibreria reenvia la informació al cloud de T.A.K durant el registre i valida aquestes operacions, posant aquesta informació a disposició del proveïdor de serveis mitjançant la interfície de gestió del frau. Els desenvolupadors d'aplicacions també poden consultar l'estat de root/jailbreak del dispositiu actual mitjançant les trucades d'API proporcionades.[12]

  La detecció de root/jailbreak és un joc de gat i ratolí, on els delinqüents informàtics intenten "amagarl·les seves explotacions de les eines de detecció de root/jailbreak i els defensors intenten trobar noves tècniques per trobar evidències de root/jailbreak.[12]

- La biblioteca estàtica libTAK.a està preparada per a la protecció contra manipulacions (anti-tamper). L'eina UpdateBinary és responsable d'aplicar les adreces de memòria dins de la biblioteca compartida generada "libtak_wrapper.so" mitjançant la informació que es troba dins del fitxer "tak.c.atp" proporcionat. Si falta aquest pas al procés de creació, la protecció antimanipulació de la biblioteca s'activarà en temps d'execució i el procés es bloquejarà. [12]

7 REQUISITS

Pel que fa als requisits de la llibreria, aquests haurien de complir els següents punts:

1. L'usuari ha de ser capaç d'introduir un pin de llargada indeterminada per teclat.
2. L'usuari ha de visualitzar els números en diferents posicions cada vegada que hagi d'introduir el PIN en l'aplicació.

3. En cap moment s'ha de guardar la contrasenya introduïda per l'usuari en memòria per tal d'evitar vulnerabilitats de sistema.

4. Un cop introduït el PIN, si aquest no és correcte, s'ha de tornar a "shiftar" les posicions dels números.

5. La llibreria Secure PIN hauria de tenir protecció contra rooting i contra hooking, fet que s'aconsegueix implementant la llibreria T.A.K.

8 Documentació de la Secure PIN Library

Considerant l'amenaça humana de l'observació de les tecles en introduir el pin o la contrasenya, es considera necessària una funcionalitat que proporcioni a l'usuari una diferent posició dels números a la pantalla de manera que un atacant no pugui reproduir el mateix pin a causa que els números no es troben en la mateixa posició.

Per crear aquesta part del projecte es va iniciar amb la creació d'una aplicació Android per després exportar els mètodes a una llibreria i poder implementar la llibreria creada en una aplicació de demostració.

La primera part del projecte va ser desenvolupar el mètode de distribuir de manera aleatòria els números en el teclat. Per realitzar aquesta funcionalitat s'ha creat una vista a Android Studio on es desplega un text que ens indica que hem d'introduir la nostra contrasenya i després el teclat numèric.

En l'aplicació inicial, es va crear un atribut per poder visualitzar el teclat numèric convencional. L'atribut regularSequence es pot d'assignar a true dins de la classe ShifterPresenterImpl dins la funció getKeyboardSequence. En cas que aquest atribut estigui assignat a false, es cridarà a la funció generateShiftedKeyboardSequence, la qual ens permet aleatoritzar la posició dels números al teclat seguint el diagrama de la figura 1.

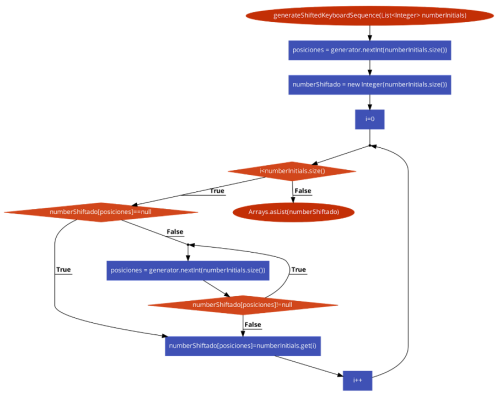

Fig. 1: Diagrama de fluxe de la funció de randomització.

Posteriorment, es va desenvolupar la vista del teclat per poder visualitzar el teclat i la introducció d'aquest.

La versió final de la visualització final del teclat és com es mostra en la figura 2.

En la figura 2 es pot observar que el teclat es troba en una representació de tres columnes i quatre files on les posicions de les tecles d'esborrar o de confirmar estan fixes en la primera i la tercera casella de la quarta fila.

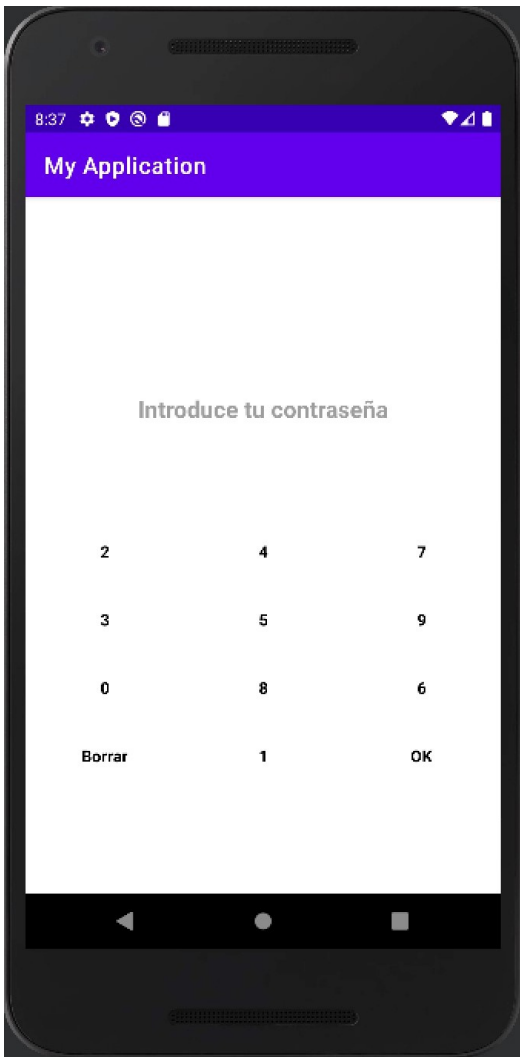

Fig. 2: Representació del teclat i l'organització de les tecles en pantalla.

El fet de mantenir les tecles esborrar i confirmar en la mateixa posició s'ha aconseguit amb la funció onBindViewHolder seguint el diagrama de flux de la figura del [Apendice 7] 7.

Posteriorment, es va desenvolupar la introducció la contrasenya utilitzant el teclat. Per tal que la contrasenya no sigui visible un cop introduïda, s'ha fet servir la funció de binding a cada caràcter introduït per l'usuari. D'aquesta manera la contrasenya introduïda està ofuscada de cara a l'usuari i de cara a altres persones al voltant seu. Aquesta característica es pot observar en la figura 4.

Així mateix, en aquesta part del projecte també s'ha considerat l'emmagatzematge de les dades entrades per l'usuari, ja que aquestes no s'han de guardar en memòria permanent, per tal de minimitzar el risc que un atacant pugui accedir a memòria i aconseguir aquesta informació.

Finalment, com a últim pas de la primera part del projecte, era la implementació de la llibreria T.A.K al projecte creat. De la llibreria T.A.K s'ha implementat el mètode de register, checkIntegrity i rootDetection. Aquesta part es pot observar en la figura **??** on en iniciar l'aplicació es mostra un toast el qual ens mostra si el dispositiu està en mode root o no.

Pel que fa a la segona part del projecte, en aquesta la

Fig. 3: Representació de la introducció de contrasenya.

Fig. 4: Representació de la implementació de la llibreria T.A.K

proposta era exportar el que s'havia realitzat fins al moment com una llibreria, és a dir, com un arxiu ".aar" i implementar-la en una aplicació Android de demostració.

Primer de tot, per tal d'exportar el que s'ha dut a terme en la primera part del projecte s'han de tenir diverses coses en compte, primerament hem de tenir en compte la importació de la llibreria T.A.K. Aquesta ha d'estar en una carpeta i aquesta carpeta s'ha de referenciar en l'arxiu settings.gradle per tal de poder exportar el projecte. Seguonament, en l'arxiu build.gradle s'ha d'actualitzar la configuració del projecte, ja que deixarà de ser una aplicació per ser una llibreria, i per tal de portar a terme això, s'ha de canviar "com.android.application" a "com.android.library", a més d'implementar el projecte que hi ha en la carpeta on hem guardat els arxius de la llibreria T.A.K.

Amb aquestes dues referències, podem recompilar el projecte amb els canvis en build.gradle i exportar el projecte com un ".aar" el qual ens permetrà utilitzar el projecte desenvolupat en altres projectes.

Finalment, per tal d'implementar la llibreria creada en una nova aplicació, haurem de crear un nou projecte i un cop creat importar un nou mòdul. Seleccionarem el directori on es troba l'arxiu ".aar" i procedirem amb la importació. Haurem de tenir en compte que per tal d'importar el projecte que s'ha de realitzar en aquest treball s'haurà d'afegir aquesta línia de codi en la classe desitjada: "import com.example.myapplication.keyboardview.view. fragment.ShiftedKeyboardFragment;"

9 Conclusions

En el món de la seguretat informàtica hi ha moltes variables de les quals hem de tenir cura. Si tenim en compte les vulnerabilitats del sistema, però no reparem en possibles amenaces externes que podria patir la nostra aplicació, estem obviant una part important de la seguretat del nostre sistema. I és que tenint en compte que avui gran part de la població utilitza el seu telèfon mòbil en llocs públics, l'accés a aplicacions sensibles és cada vegada més crític. La importància de tenir un inici de sessió segur, no només en el fet que estigui protegit de vulnerabilitats de sistema, sinó també d'amenaces externes, és molt necessari.

La realització d'aquest treball es basa en la creació d'un inici de sessió que no només estigui protegit de vulnerabilitats, fet que s'aconsegueix aplicant funcionalitats de la llibreria T.A.K desenvolupada per l'empresa Build38, sinó que també estigui protegit per amenaces externes, com per exemple, un usuari que "espiï" quan s'introdueixi el PIN i robi la informació. Aquesta última part és la que s'ha dut a terme al projecte: l'aleatorització de les tecles del teclat per introduir el PIN d'un inici de sessió. D'aquesta manera,

si qualsevol persona espia la introducció del codi PIN no serà capaç de reproduir-lo, ja que les tecles no estaran en la localització convencional.

Així i tot, s'ha de tenir en compte que aquesta llibreria no protegeix de cara a transmissió de dades al servidor, per tant, si els desenvolupadors volen implementar aquesta llibreria, el treball a futur hauria de ser implementar l'encriptació de dades en la comunicació amb el servidor.

AGRAÏMENTS

Primer de tot voldria agrair al meu tutor que m'ha aconsellat i ajudat durant tot aquest procés i sobretot per la magnífica idea proposada.

A continuació m'agradaria agrair a la Universitat i a totes les persones que han format part dels meus anys universitaris perquè sense cap mena de dubte no hi seria on soc sense els professors que m'han ensenyat i sense els meus companys amb els quals he après d'ells i també han après de mi.

Finalment, voldria agrair a la meva família, que sense la seva paciència i bones paraules, no hi seria aquí.

REFERÈNCIES

[1] Pagina web de la empresa Build 38 [Online]. Disponible: https://build38.com/

[2] Build38 developers. *Build38's Top 10 most effective procedures for App Protection* Barcelona, 2022. [Online]. Disponible: https://build38.com/effective-procedures-app-protection/

[3] Explicació de la llista CVE [Online]. Disponible: https://es.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures

[4] Definició de Vulnerabilitat. [Online]. Disponible: https://www.ambit-bst.com/blog/diferencias-entre-amenaza-vulnerabilidad-y-riesgo

[5] Definició de Amenaça. [Online]. Disponible: https://www.ambit-bst.com/blog/diferencias-entre-amenaza-vulnerabilidad-y-riesgo

[6] Definició de Risc. [Online]. Disponible: https://www.ambit-bst.com/blog/diferencias-entre-amenaza-vulnerabilidad-y-riesgo

[7] 8 vulnerabilidades que hacen tu app insegura. [Online]. Disponible: https://seguridad.prestigia.es/vulnerabilidades-app-insegura/

[8] ¿Qué es un certificado SSL? [Online]. Disponible: https://www.websecurity.digicert.com/es/es/security-topics/what-is-ssl-tls-https

[9] Hooking de funciones en apps móviles (o cómo alterar su comportamiento) [Online]. Disponible: welivesecurity.com/la-es/2017/06/07/hooking-aplicaciones-alterar-comportamiento/

[10] What rooting is and how root detection can be done on Android [Online]. Disponible: https://promon.co/security-news/root-detection-android/

[11] ¿En qué consiste la vulnerabilidad de Apache Log4J (Log4Shell)? [Online].https://www.trendmicro.com/es_es/what-is/apache-log4j-vulnerability.html

[12] T.A.K documentation, Build38 Developers, 2022. Documentació no disponible al públic.

APÈNDIX

1.1 Gantt

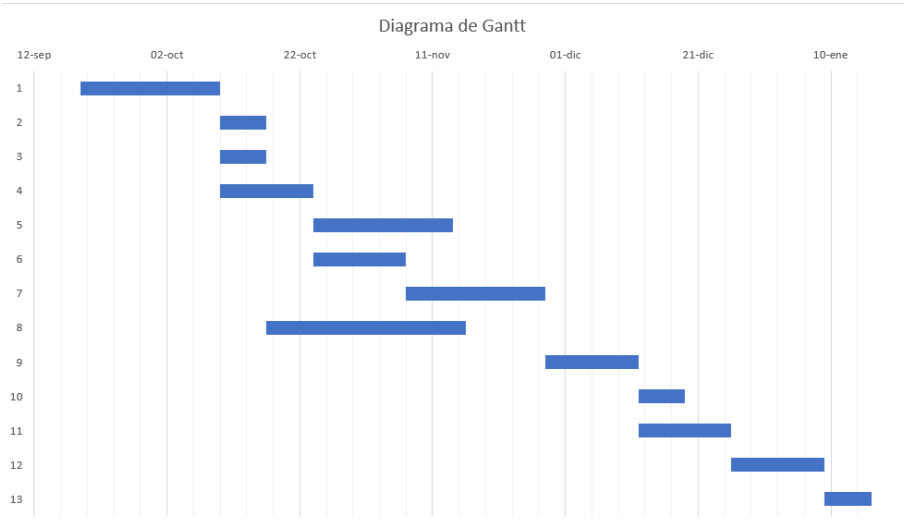

Fig. 5: Diagrama de Gantt per a la planificació inicial.

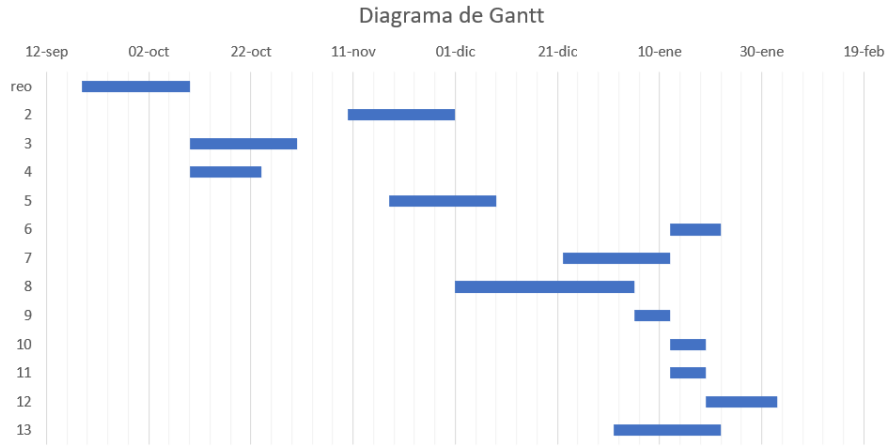

Fig. 6: Diagrama de Gantt per a la planificació realitzada.

1.2 Fluxe

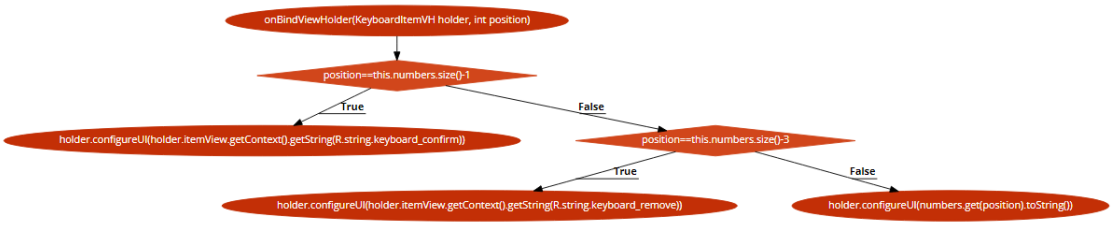

Fig. 7: Diagrama de fluxe de l'estabilització de les tecles Esborrar i Ok