Tema: Securitatea platformelor mobile iOS și Android

Introducere

Informația este un produs care, ca și alte produse importante rezultate din activitatea umană, are valoare pentru o organizație și în consecință, este necesar să fie protejată corespunzător.

Fiecare organizație își poate implementa propriul său sistem de asigurare a securității informației. Un management al securității informației se realizează prin implementarea unui set corespunzător de acțiuni care cuprinde politici, practici, instrumente și proceduri, structuri organizaționale, precum și funcții software. Toate acțiunile trebuie prevăzute, definite și aplicate pentru a asigura că sunt întrunite obiectivele specifice de securitate a organizației.

Deoarece dezvoltarea tehnologiilor este în continuă avansare, telefoanele mobile, tabletele și laptopurile au devenit foarte răspîndite și comune. Puterea computațională a acestor dispozitive continuă să crească, pe cînd ele devin mai mici și mai ușoare. Un spectru nou de produse mobile care combină un șir de posibilități computaționale într-un singur dispozitiv este în curs de dezvoltare. Aceste posibilități convergente mobile permit utilizatorilor să rămînă online, accesînd toate datele de care dispun, ca poșta electronică și cotații la bursă chiar și în timpul mișcării.

Deși există multe avantaje evidente ale acestor dispozitive, ele de asemenea afectează procesele tradiționale de business și din acest considerent este nevoie de acordat o importanță prioritară securității. Organizațiile trebuie să fie preocupate de securitate, ce include furtul sau pierderea dispozitivelor mobile folosite de angajați, scurgerea informațiilor corporative prin intermediul acestor dispozitive, probabilitatea infectării cu viruși și o posibilă interceptare neautorizată a traficului. În timp ce beneficiem de convinienețele și efecienețele aduse de noile tehnologii mobile, trebuie să fie elaborate și implementate măsuri corespunzătoare de securitate pentru a contracara orice amenințări asupra detelor senzitive introduse de utilizatorii dispozitivelor mobile.

În cadrul acestei lucrări de curs vor fi prezentate două dintre cele mai importante noțiuni care caracterizează tehnologiile mobile moderne și anume platformele iOS și Android. Lucrarea își propune să descrie cum asigură aceste platforme normele de securitate, ce măsuri sunt întreprinse pentru a asigura confidențialitatea utilizatorilor și datelor acestora.

Securitatea iOS

Compania Apple a proiectat platforma iOS avînd la bază conceptul de securitate. Păstrarea informației securizată pe dispozitivele mobile este foarte importantă pentru fiecare utilizator, indiferent dacă el accesează informații corporative și personale sau păstrează fotografii personale, informații bancare și adrese. Deoarece informația fiecărui utilizator este importantă, dispozitivele iOS sunt proiectate să mențină un nivel înalt de securitate fără a compromite experiența utilizatorului.

Dispozitivele iOS oferă caracteristici stricte de securitate a tehnologiilor, dar totuși sunt ușor de folosit. Dispozitivele sunt proiectate să asigure o transparență cît mai mare a securității. Multe caracteristici de securitate sunt activate implicit și astfel departamentul IT nu trebuie să execute configurări adăugătoare. Și unele caracteristici cheie, cum ar fi criptarea dispozitivului, nu sunt posibile de configurat, deci utilizatorii nu le pot dezactiva din greșeală.

Pentru organizații, luarea în considerație a securității dispozitivelor iOS, este de mare ajutor să înțeleagă cum lucrează caracteristicile încorporate de securitate pentru a oferi o platformă mobilă sigură.

iPhone, iPad și iPod touch sunt proiectate cu nivele de securitate. Caracteristicile hardware și firmware de nivel jos protejează împotriva programelor malițioase și a virușilor, pe cînd caracteristicile de nivel înalt permit accesul securizat la informația personală și corporativă, previne utilizarea neautorizată și ajută la contracararea atacurilor.

Modelul de securitate iOS protejează informațiile totodată permițînd utilizarea mobilă, aplicații străine și sincronizarea. În mare parte sistemul se bazează pe principiile securizate de proiectare și în multe cazuri, Apple a realizat lucru de proiectare adițional pentru a spori gradul de securitate, fără a compromite utilizarea.

iOS se bazează pe aceleați tehnologii ca și OS X și beneficiază de ani de dezvoltare a securității. Înbunătățirile continue și caracteristicile adiționale de securitate au permis departamentelor IT din întreprinderile din lumea întreagă să adopte rapid și să suporte dispozitivele iOS în rețelele lor.

1.1 Arhitectura sistemului

Integrarea strînsă a hardware și software în dispozitivele iOS permit validarea activităților pentru toate nivelele. De la pornirea inițială pînă la instalarea unui soft iOS, fiecare pas este analizat și verificat pentru a asigura că fiecare activitate este de încredere și că folosește resursele în mod corespunzător.

O dată ce sistemul lucrează, această arhitectură de securitate integrată depinde de integritatea și fiabilitatea XNU, kernel-ul iOS. XNU impune caracteristici de securitate în timpul rulării și este esențial în sporirea încrederei în funcțiile și aplicațiile de nivel înalt.

1.1.1 Pornire securizată

Fiecare pas al procesului de pornire (boot-up) conține componente ce sunt semnate criptografic de către Apple pentru a asigura integritatea și purcede doar după verificarea lanțului de încredere. Acesta include aplicații bootloader, kernel, extensiile kernel și banda firmware.

Cînd un dispozitiv iOS este conectat, procesorul de aplicații execută imediat codul de pe memoria read-only, cunoscută ca Boot ROM. Acest cod imuabil este prevăzut în timpul fabricării cipurilor și este implicit de încredere. Codul de Boot ROM conține cheia publică a Apple - Root CA, care este folosită pentru a verifica dacă Bootloader (LLB) este semnat de către Apple înainte de a-i permite să se încarce. Acesta este primul pas în lanțul de încredere, unde fiecare pas asigură faptul că următorul este semnat de către Apple. Cînd termină LLB sarcinile sale, acesta verifică și rulează următoarea etapă, iBoot, care, la rîndul său, verifică și rulează kernel-ul iOS.

Acest lanț de pornire securizat asigură că cele mai mici niveluri ale softului nu sunt alterate și permite iOS să ruleze doar pe dispozitivele Apple validate.

Dacă un pas al acestui proces de pornire nu poate încărca sau verifica următorul, procesul de boot-up este oprit și dispozitivul afișează ecranul de "Conectare la iTunes". Acesta este modul de recuperare. Dacă Boot ROM nu este capabil de a încărca sau a verifica nici LLB, acesta pornește modul DFU (Device Firmware Upgrade). În ambele cazuri, dispozitivul trebuie să fie conectat la iTunes prin USB și trebuie să fie restabilite setările implicite.

1.1.2 Personalizarea softului de sistem

Apple lansează regulat actualizări de software pentru a răspunde preocupărilor de securitate ce apar; aceste actualizări sunt furnizate pentru toate dispozitivele acceptate simultan. Utilizatorii primesc notificările de actualizări iOS pe dispozitive și prin intermediul iTunes, și actualizările sunt livrate în mod wireless, încurajînd adoptarea rapidă a remedierilor recente de securitate.

Procesul de boot descris mai sus asigură faptul că numai codul semnat de Apple poate fi instalat pe un dispozitiv. Pentru a preveni dispozitivele de a fi retrogradate la versiuni mai vechi, care nu dispun de cele mai recente actualizări de securitate, iOS utilizează un proces numit System Software Personalization. Dacă retrogradările au fost posibile, un atacator care cîștigă posesia unui dispozitiv ar putea instala o versiune mai veche de iOS și să exploateze o vulnerabilitate care a fost fixată în versiunea mai nouă.

Actualizările softului iOS pot fi instalate folosind iTunes sau over-the-air (OTA) pe dispozitiv. Cu iTunes, o copie completă a iOS este descărcată și instalată. Actualizările de soft OTA sunt oferite ca delta pentru eficiența rețelei.

În timpul unei instalări sau modernizări iOS, iTunes (sau dispozitivul însuși, în cazul actualizărilor de soft OTA) se conectează la serverul de instalare autorizat al Apple și trimite o listă de măsurători criptografici pentru fiecare parte a instalației, care urmează să fie instalate (de exemplu LLB, iBoot, kernel-ul, imagine OS), o valoare anti-replay aleatorie (Nonce), și ID-ul unic al dispozitivului (ECID).

Serverul verifică lista prezentată de măsurători față de versiunile pentru care se permite instalarea, iar în cazul în care se găsesc similarități, adaugă ECID la măsuratori și semnează rezultatul. Setul complet de date semnate de pe server este trecut la dispozitiv ca parte a procesului de instalare sau upgrade. Adăugarea ECID "personalizează" autorizația pentru dispozitivul solicitant. Autorizînd și semnînd doar măsurătorile cunoscute, serverul asigură că actualizarea este exact așa cum a fost furnizată de Apple.

Aceste etape asigură că autorizația este pentru un anumit dispozitiv și că o versiune veche a iOS de la un dispozitiv nu poate fi copiată la altul. Nonce împiedică un atacator de a salva răspunsul serverului și de a-l folosi pentru a retrograda un dispozitiv în viitor.

1.1.3 Semnarea codului aplicațiilor

Odată ce kernel-ul iOS s-a pornit, acesta controlează care dintre procesele și aplicațiile utilizatorilor pot fi rulate. Pentru a se asigura că toate aplicațiile provin dintr-o sursă cunoscută și aprobată și nu au fost modificate, iOS impune ca tot codul executabil să fie semnat utilizînd un certificat emis de Apple. Aplicațiile prevăzute cu dispozitivul, cum ar fi Mail și Safari, sunt semnate de către Apple. Semnarea obligatorie a codului extinde conceptul de lanț de încredere de la sistemul de operare la aplicații și previne aplicațiile să încărce resurse nesemnate sau să folosească cod ce se automodifică.

În scopul de a dezvolta și instala aplicații pe dispozitive iOS, dezvoltatorii trebuie să se înregistreze la Apple și să se alăture programului iOS Developer. Identitatea reală a fiecărui dezvoltator, fie el individ sau întreprindere, este verificată de Apple înainte de eliberarea certificatului lor. Acest certificat permite dezvoltatorilor să semneze aplicații și să le prezinte la App Store pentru

distribuție. Ca urmare, toate aplicațiile din App Store au fost prezentate de către o persoană sau organizație identificabilă, servind ca un factor de descurajare pentru crearea de aplicații malițioase. Ei au fost, de asemenea, verificați de către Apple pentru a se asigura că aceștia operează așa cum este descris și nu conțin erori evidente sau alte probleme.

Întreprinderile au, de asemenea, posibilitatea de a scrie aplicații pentru folosirea în cadrul organizației lor și să le distribuie angajaților lor. Întreprinderile și organizațiile pot aplica la iOS Developer Enterprise Program (iDEP), cu un număr DUNS. Apple aprobă aplicanții după verificarea identității și eligibilității lor. Odată ce o organizația devine membră a iDEP, se poate înregistra pentru a obține un profil de provizionare care permite aplicațiilor să ruleze pe dispozitive autorizate. Utilizatorii trebuie să aibă profilul de provizionare instalat în scopul de a rula aplicațiile in-house.

Spre deosebire de alte platforme mobile, iOS nu permite utilizatorilor să instaleze aplicații potențial malițioase nesemnate de pe site-uri web, sau să ruleze cod ce nu este de încredere. În timpul rulării, semnătura cod verifică dacă toate paginile de memorie executabile sunt făcute ca acestea să asigure că o aplicație nu a fost modificată de cînd a fost instalată sau actualizată ultima dată.

1.1.4 Securitatea procesului de runtime

Odată ce se verifică dacă o aplicație provine de la o sursă autorizată, iOS impune măsuri de securitate pentru a se asigura că aceasta nu poate compromite alte aplicații sau restul sistemului.

Toate aplicațiile de la părți terțe sunt limitate de la accesul fișierelor stocate de alte aplicații sau de la efectuarea de modificări pe dispozitiv. Acest lucru previne aplicațiile să colecteze sau să modifice informațiile stocate de către alte aplicații. Fiecare aplicație are un director unic pentru fișierele sale, care este atribuit aleatoriu atunci cînd aplicația este instalată. În cazul în care o aplicație de la părți terțe are nevoie să acceseze informații altele decît cele proprii, face acest lucru doar prin utilizarea de API și a serviciilor oferite de iOS.

Fișierele de sistem și resursele sunt protejate de aplicațiile utilizatorilor. Instrumentele care nu sunt necesare, cum ar fi serviciile de autentificare de la distanță, nu sunt incluse în software-ul sistemului, și API nu permite aplicațiilor să escaladeze privilegiile lor pentru a modifica alte aplicații sau iOS în sine.

Accesul către aplicații de la părți terțe la informațiile și caracteristicile utilizatorului, cum ar fi iCloud este controlată utilizînd drepturi declarate. Drepturile sunt perechi cheie / valoare, care sunt conectate la o aplicație și permite autentificarea dincolo de factori de runtime, cum ar fi ID-ul unix al utilizatorului. Deoarece drepturile sunt semnate digital, acestea nu pot fi schimbate. Drepturile sunt utilizate pe scară largă de aplicațiile de sistem pentru a efectua operațiuni specifice privilegiate care în caz contrar ar impune procesul de a rula ca root. Acest lucru reduce foarte mult potențialul de escaladare privilegiată de către o aplicație de sistem compromisă.

În plus, aplicațiile pot efectua numai procesare de fond prin intermediul sistemului furnizat de API. Acest lucru permite aplicațiilor să continue să funcționeze fără a dăuna performanței sau a a avea un impact dramatic asupra bateriei. Aplicațiile nu pot partaja date direct unele cu altele; partajarea poate fi implimentată doar de către ambele aplicații simultan, utilizînd scheme personalizate.

Address space layout randomization (ASLR) protejează împotriva exploatării de bug-uri ce corup memoria. Aplicațiile implicite folosesc ASLR pentru a asigura că toate regiunile de memorie sunt atribuite aleatoriu la pornire. În plus, locațiile librăriilor partajate de sistem sunt atribuite aleatoriu la fiecare pornire a dispozitivului. Xcode, mediul de dezvoltare iOS, în mod automat compilează programele de la părți terțe, cu sprijinul ASLR pornit.

Protecția ulterioară este asigurată de iOS folosind caracteristica ARM Execute Never (XN), care marchează paginile de memorie ca neexecutabile. Paginile de memorie marcate atît ca inscriptibile cît și ca executabile pot fi utilizate numai de către aplicații, în condiții strict controlate.

1.2 Criptarea și protejarea datelor

Lanțul de pornire securizat, semnarea codului și securitatea procesului de execuție, toate contribuie la asigurarea faptului că doar aplicații și cod de încredere pot rula pe un dispozitiv. iOS are caracteristici suplimentare de securitate pentru a proteja datele utilizatorului, chiar și în cazurile în care alte părți ale infrastructurii de securitate au fost compromise (de exemplu, pe un dispozitiv cu modificări neautorizate). Ca arhitectura sistemului în sine, aceste capabilități de criptare și protecție a datelor folosesc nivele de hardware și tehnologii software integrate.

1.2.1 Caracteristici de securitate hardware

Pe dispozitive mobile, viteza și eficiența energiei sunt critice. Operațiunile criptografice sunt complexe și pot introduce performanță sau probleme pentru baterie în cazul în care nu sunt concepute și implimentate în mod corect.

Fiecare dispozitiv iOS are un motor de criptare AES 256 construit în calea DMA între stocarea flash și memorie principală de sistem, determinînd criptarea fișierelor să fie extrem de eficientă. Împreună cu motorul AES, SHA-1 este implementat în hardware, reducînd în continuare operațiile de criptografie suplimentare.

ID-ul unic al dispozitivului (UID) și ID-ul de grup al dispozitivului (GID) sunt cheile AES 256-bit fuzionate în procesorul aplicației în timpul fabricării. Nici un soft sau firmware nu le pot citi direct, ei pot vedea doar rezultatele operațiunilor de criptare sau decriptare efectuate utilizîndu-le. UID este unic pentru fiecare dispozitiv și nu este înregistrat de Apple sau de oricare dintre furnizorii săi. GID este comun pentru toate procesoarele într-o clasă de dispozitive (de exemplu, toate dispozitivele care folosesc Apple A5 cip), și este folosit ca un nivel suplimentar de protecție atunci cînd se livrează soft-ul sistemului în timpul instalării și a restabilirii. Scrierea acestor chei în siliciu previne falsificarea sau ocolirea lor și garantează că acestea pot fi accesate doar de motorul AES.

UID permite ca datele să fie criptografic legate de un anumit dispozitiv. De exemplu, ierarhia cheilor ce protejează fișierele sistemului include UID, deci dacă cipurile de memorie sunt fizic mutate de la un dispozitiv la altul, fișierele sunt inaccesibile. UID nu este legat de nici un alt identificator pe dispozitiv.

În afară de UID și GID, toate celelalte chei criptografice sunt create de către generatorul de numere aleatoare al sistemului (RNG), folosind un algoritm bazat pe Yarrow. Entropia sistemului este colectată de la momentul întreruperii în timpul procesului de pornire și, în plus de la senzorii interni odată ce aparatul s-a conectat.

Ștergerea sigură a cheilor salvate este la fel de importantă ca și generarea lor. Este deosebit de dificil de a face acest lucru pe stocare flash, unde numărul mare de nivele ar putea să însemne mai multe copii ale datelor ce trebuie să fie șterse. Pentru a rezolva această problemă, dispozitivele iOS includ o caracteristică menită să securizeze ștergerea datelor numită Effaceable Storage. Această caracteristică accesează tehnologia de stocare de bază (de exemplu, NAND) pentru a aborda în mod direct și a șterge un număr mic de blocuri de la un nivel foarte scăzut.

1.2.2 Protecția fișierului de date

În plus față de funcțiile de criptare hardware integrate în dispozitive iOS, Apple folosește o tehnologie numită Protecție de date pentru a proteja în continuare datele stocate în memoria flash pe dispozitiv. Această tehnologie este proiectată cu dispozitive mobile, luîndu-se în considerare faptul că acestea pot fi întotdeauna pornite și conectate la Internet, și pot primi apeluri telefonice, text sau e-mailuri în orice moment.

Protecție de date permite unui dispozitiv să răspundă la evenimente cum ar fi apeluri telefonice primite fără a decripta datele sensibile și a descărca informații noi în timp ce e blocat. Aceste comportamente individuale sunt controlate prin atribuirea fiecărui fișier la o clasă.

Protecția datelor protejează datele din fiecare clasă în baza timpului cînd datele trebuie să fie accesate. Accesibilitatea este determinată și de faptul că tastele de clasă au fost deblocate. Protecția de date este pusă în aplicare prin construirea și gestionarea ierarhiei de chei, și se bazează pe tehnologii de criptare hardware.

Prezentarea generală a arhitecturii

De fiecare dată cînd este creat un fișier pe partiția de date, Protecția datelor creează o cheie nouă de 256-bit și o transmite către motorul hardware AES, care utilizează cheia pentru a cripta fișierul așa cum este scris pe memorie flash folosind modul AES CBC.

Metadata tuturor fișierelor în sistemul de fișiere sunt criptate cu o cheie aleatoare, care este creată atunci cînd iOS este instalat pentru prima dată sau cînd dispozitivul este distrus de către un utilizator. Cheia fișierului de sistem este stocat în Effaceable Storage. Din moment ce este stocat pe dispozitiv, această cheie nu este utilizată să păstreze confidențialitatea datelor, în schimb, este concepută astfel ca să poată fi rapid ștearsă la cerere (de către utilizator, cu opțiunea "Erase all content and settings" sau de către un utilizator sau administrator, care cer ștergerea de la distanță de pe un server Mobile Device Management, Exchange ActiveSync, sau iCloud). Ștergerea cheiei în acest mod face ca toate fișierele criptografice să fie inaccesibile.

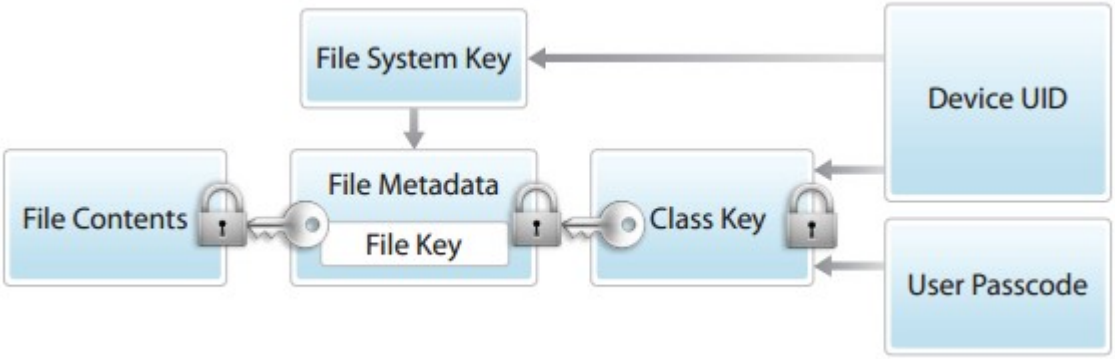


Figura 1.2.2.1

1.2.3 Cod de acces

Prin stabilirea unui cod de acces pentru un dispozitiv, utilizatorul activează în mod automat Protecția datelor. iOS suportă codurile de acces de patru cifre și de lungime arbitrară alfanumerică. În plus față de deblocarea dispozitivului, un cod de acces prevede entropia pentru chei de criptare, care nu sunt stocate pe dispozitiv. Acest lucru înseamnă că un atacator ce se află în posesia unui dispozitiv nu poate obține accesul la date din anumite clase de protecție, fără coduri de acces.

Codul de acces este "încurcat" cu UID-ul dispozitivului, astfel încercările trebuie să fie efectuate pe dispozitivul ce se află sub atac. Un număr mare de iterații este folosit pentru a face fiecare încercare mai lentă. Numărul iterațiilor este calibrat astfel încît o încercare durează aproximativ 80 milisecunde. Acest lucru înseamnă că va dura mai mult de 5 ani și jumătate pentru a încerca toate combinațiile de șase caractere a codului de acces alfanumeric cu litere mici și numere, sau 2 ani și jumătate pentru un cod de acces de nouă cifre cu numere numai.

Pentru a descuraja atacurile asupra codului de acces, interfața iOS impune escaladarea întîrzierii de la intrarea unui cod de acces invalid la ecranul de blocare. Utilizatorii pot alege de a șterge dispozitivul automat după 10 încercări eșuate de introducere a codului de acces. Această setare este de asemenea, disponibilă ca o politică administrativă prin Mobile Management Device (MDM) și Exchange ActiveSync, și poate fi, de asemenea, setat la un prag inferior.

Crearea parolelor Apple ID puternice

ID-urile Apple sunt utilizate pentru conectare la un număr de servicii, inclusiv iCloud, FaceTime, și iMessage. Pentru a ajuta utilizatorii să creeze parole puternice, toate conturile noi trebuie să conțină următoarele atribute pentru parolă:

- Cel putin opt caractere
- Cel puțin o literă
- Cel puțin o literă majusculă
- Cel puțin un număr
- Nu mai mult de trei caractere identice consecutiv
- Să nu fie la fel cu numele de cont

Considerații pentru parolă

Dacă este introdusă o parolă lungă care conține numai numere, pe ecranul de Lock este afișată o tastatură numerică în loc de tastatură completă. Un cod de acces mai lung poate fi introdus mai ușor decît un cod de acces alfanumeric mai scurt, în timp ce asigură același nivel de securitate.

1.2.4 Clase

Atunci cînd este creat un fișier nou pe un dispozitiv iOS, lui îi este atribuită de către aplicație o clasă care îl creează. Fiecare clasă utilizează politici diferite pentru a determina cînd datele sunt accesibile. Clasele de bază și politicile sunt, după cum urmează:

Complete Protection

(NSFileProtectionComplete): Cheia clasei este protejată cu o cheie derivată din codul de acces al utilizatorului și dispozitivul UID. La scurt timp după ce utilizatorul blochează un dispozitiv (10 secunde, dacă setarea Require password este Immediately), cheia clasei decriptată este eliminată, făcînd toate datele din această clasă inaccesibile pînă utilizatorul introduce parola din nou. Aplicația de Mail pune în aplicare Complete Protection pentru mesaje și atașamente. Aplicațiile care lansează imagini și date de localizare sunt, de asemenea, stocate cu Complete Protection.

Protejat pînă este deschis

(NSFileProtectionCompleteUnlessOpen): Unele fișiere pot avea nevoie să fie scrise în timp ce dispozitivul este blocat. Un bun exemplu în acest sens este un atașament e-mail ce se descarcă în background. Acest comportament este realizat prin utilizarea criptografiei asimetrice curbe eliptice (ECDH pe Curve25519). Împreună cu obișnuita cheie per-fișier, protecția datelor generează o pereche de chei publice/private pentru fișier. Un secret partajat este calculat folosind cheia privată a fișierului și

cheia publică a clasei Protected Unless Open, a cărui cheie privată corespunzătoare este protejată cu codul de acces al utilizatorului și UID-ul dispozitivului. Cheia per-file este împachetată cu hash-ul acestui secret partajat și stocată în metadata fișierului, împreună cu cheia publică a fișierului, cheia privată corespondentă este apoi ștearsă din memorie. Îndată ce fișierul este închis, cheia per-file este, de asemenea, ștearsă din memorie. Pentru a deschide fișierul din nou, secretul partajat este re-creat folosind cheia privaă a clasei Protected Unless Open și cheia publică efemeră a fișierului; hash-ul este folosit pentru a desface cheia per-file, care este apoi folosită pentru a decripta fișierul.

Protected Until First User Authentication

(NSFileProtectionCompleteUntilFirstUserAuthentication): Această clasă se comportă în același mod ca și Complete Protection, cu excepția faptului că cheia decriptată a clasei nu este ștearsă din memorie atunci cînd dispozitivul este blocat. Protecția în această clasă are proprietăți similare pentru decriptarea desktop full-disk și protejează datele de atacuri care implică o restartare.

No protection

(NSFileProtectionNone): Cheia acestei clase este protejată doar cu UID, și se păstrează în Effaceable Storage. Aceasta este clasa implicită pentru toate fișierele care nu sunt atribuite unei clase Data Protection. Din moment ce toate cheile necesare pentru a decripta fișiere în această clasă sunt stocate pe dispozitiv, criptarea permite numai beneficiul de ștergere rapidă la distanță. Dacă unui fișier nu îi este atribuită o clasă Data Protection, este încă stocată în formă criptată (așa cum sunt toate datele pe un dispozitiv iOS).

iOS Software Development Kit (SDK) oferă o suită completă de API-uri, care fac mai ușor adoptarea clasei Data Protection de către terțe părți și dezvoltatori in-house și asigurarea nivelului cel mai înalt de protecție a datelor pentru aplicațiile lor. Data Protection este disponibilă pentru API de fișier și baze de date, inclusiv NSFileManager, CoreData, NSData, și SQLite.

1.3 Securitate de rețea

În plus față de măsurile ce au fost luate de către Apple pentru a proteja datele stocate pe dispozitive iOS, există multe măsuri de securitate de rețea pe care organizațiile le pot lua pentru a proteja informații ce se deplasează la și de la un dispozitiv iOS.

Utilizatorii mobili trebuie să fie în măsură să acceseze rețele de informații corporative de oriunde în lume, așa că este important să se asigure că sunt autorizați și că datele lor sunt protejate în timpul transmisiei. iOS folosește și oferă acces de dezvoltator la protocoalele standard de rețea pentru comunicări autentificate, autorizate, și criptate. iOS oferă tehnologii verificate și cele mai recente standarde de securitate pentru a realiza aceste obiective de securitate, atît pentru conecții Wi-Fi cît și pentru cele mobile.

Pe alte platforme, software-ul firewall este necesar pentru a proteja numeroasele porturi de comunicare deschise. Deoarece iOS atinge o suprafață redusă de atac prin limitarea porturilor de ascultare și eliminare a utilităților de rețea inutile, cum ar fi telnet, shells, sau un server web, nu are nevoie de software-ul firewall. În plus, comunicarea folosind iMessage, FaceTime, și Apple Push Notifications Server este complet criptată și autentificată.

1.3.1 SSL, TLS

iOS sprijină Secure Socket Layer (SSL v3), precum și Transport Layer Security (TLS v1.1, TLS v1.2) și DTLS. Safari, Calendar, Mail, și alte aplicații Internet folosesc automat aceste mecanisme pentru a activa un canal de comunicare criptat între dispozitive și servicii de rețea. API-uri de nivel înalt (cum ar fi CFNetwork) fac mai ușor pentru dezvoltatori să adopte TLS în aplicațiile lor, în timp ce API-uri de nivel scăzut (SecureTransport) furnizează control bun.

1.3.2 WI-FI

iOS sprijină protocoalele Wi-Fi, inclusiv WPA2 Enterprise, pentru a oferi acces autentificat la rețele corporative fără fir. WPA2 Enterprise folosește criptare AES pe 128-bit, oferindu-le utilizatorilor cel mai înalt nivel de asigurare că datele lor rămîn protejate atunci cînd se trimit și se primesc comunicații printr-o conexiune Wi-Fi. Cu suport pentru 802.1X, dispozitivele iOS pot fi integrate într-o amplă gamă de medii de autentificare RADIUS. Metodele de autentificare fără fir 802.1X acceptate de iPhone și iPad includ EAP-TLS, EAP-TTLS, EAP-FAST, EAP-SIM, PEAPv0, PEAPv1, și LEAP.

1.4 Acces la dispozitiv

iOS sprijină politicile flexibile de securitate și configurațiile care pot fi ușor puse în aplicare și gestionate. Acest lucru permite întreprinderilor să protejeze informațiile corporative și să se asigure că angajații îndeplinesc cerințele, chiar dacă ei utilizează dispozitive personale.

1.4.1 Protecția codului de acces

În plus față de protecția criptografică, codurile de acces previn accesul neautorizat la interfața dispozitivului. Interfata iOS impune controlul întîrzierilor după introducerea unui cod de acces nevalid, reducînd dramatic eficacitatea atacurilor prin intermediul Lock screen. Utilizatorii pot opta pentru ștergerea automată a dispozitivului după 10 încercări eșuate de introducere a parolei. Această setare este disponibilă ca o politica administrativă și poate fi, de asemenea, setată la un prag inferior prin MDM și Exchange ActiveSync.

În mod implicit, codul de acces al utilizatorului poate fi definit ca un cod PIN de patru cifre. Utilizatorii pot specifica un cod de acces alfanumeric mai lung prin activarea Settings > General > Passcode > Complex Passcode. Parolele mai lungi și mai complexe sunt mai greu de ghicit sau atacat și sunt recomandate pentru utilizare în întreprinderi.

Administratorii pot impune cerințe complexe față de codul de acces și alte politici utilizînd MDM sau Exchange ActiveSync, sau prin a cere utilizatorilor să instaleze manual profiluri de configurare. Următoarele politici față de codul de acces sunt disponibile:

- Valoare simplă
- Necesită o valoare alfanumerică
- Lungime minimă a parolei
- Număr minim de caractere complexe
- Vîrstă maximă a codului de acces
- Istoria codului de acces
- Auto-Lock timeout
- Perioada de grație pentru blocarea dispozitivului
- Numărul maxim de încercări eșuate

1.4.2 Realizarea configurației

Un profil de configurare este un fișier XML care permite unui administrator să distribuie informații de configurare pentru dispozitive iOS. Setările care sunt definite de un profil de configurare instalat nu pot fi modificate de către utilizator. Dacă utilizatorul șterge un profil de configurare, toate setările definite de profil sunt, de asemenea, eliminate. În acest fel, administratorii pot impune setări prin legarea politicii de acces. De exemplu, un profil de configurare care oferă o configurație de e-mail poate, de asemenea, să specifice o politică de acces la dispozitiv. Utilizatorii nu vor putea accesa e-mail-ul dacă codurile lor de acces nu îndeplinesc cerințele administratorului.

Profilele de configurare pot fi semnate și criptate pentru a valida originea lor, asigura integritatea lor, și proteja conținutul lor. Profilele de configurare sunt criptate folosind CMS (RFC 3852), ce suportă 3DES și AES-128.

Profilele de configurare pot fi, de asemenea, blocate la un dispozitiv pentru a preîntîmpina complet ștergerea lor, sau pentru a permite ștergerea doar cu un cod de acces. Deoarece mulți utilizatori din întreprinderi dețin personal dispozitivele lor iOS, profile de configurare care leagă un dispozitiv de un MDM server poate fi eliminat, dar acest lucru va elimina, de asemenea, toate informațiile de configurare gestionate, date și aplicații.

Utilizatorii pot instala profilele de configurare direct pe dispozitivele lor folosind iPhone Configuration Utility. Profilele de configurare pot fi descărcate prin e-mail sau wireless folosind un server MDM.

1.4.3 Managementul dispozitivelor mobile

iOS suport pentru MDM permite companiilor să configureze în mod sigur și să gestioneze dezvoltările scalate pentru iPhone și iPad din organizațiile lor. Capabilitățile MDM sunt construite pe tehnologiile iOS existente, cum ar fi Configuration Profiles, Over-the-Air Enrollment și serviciul Apple Push Notification. Folosind MDM, departamentele IT pot înscrie dispozitivele iOS într-un mediu de întreprindere, configura și actualiza setările în mod wireless, monitoriza respectarea politicilor corporative, și chiar șterge sau bloca dispozitivele gestionate de la distanță.

1.4.4 Ștergere la distanță

Dispozitivele iOS pot fi șterse de la distanță de către un administrator sau utilizator. Ștergerea instantă de la distanță se realizează prin eliminarea sigură a cheii de criptare a blocului de păstrare din Effaceable Storage, făcînd toate datele imposibil de citit. Ștergerea de la distanță poate fi inițiată de MDM, Exchange, sau iCloud.

Cînd este declansată ștergerea de la distanță de MDM sau iCloud, dispozitivul trimite o confirmare și efectuează ștergerea. Pentru ștergerea de la distanță prin intermediul Exchange, dispozitivul verifică utilizînd Exchange Server înainte de a efectua ștergerea.

Utilizatorii pot șterge, de asemenea, dispozitivele aflate în posesia lor folosind setările aplicației. Și, după cum a fost menționat, dispozitivele pot fi setate pentru a șterge automat după o serie de încercări eșuate de a introduce codul de acces.

2. Securitate Android

Android este o platformă mobilă modernă, care a fost proiectată pentru a fi cu adevărat deschisă. Aplicațiile Android fac uz de hardware și software avansat, precum și date locale, expuse prin intermediul platformei pentru a aduce inovație și valoare pentru consumatori. Pentru a proteja această valoare, platforma trebuie să ofere un mediu de aplicație care asigură securitatea utilizatorilor, datelor, aplicațiilor, dispozitivelor, și rețelelor.

Asigurarea unei platforme deschisă necesită o arhitectură de securitate robustă și programe riguroase de securitate. Android a fost proiectat cu mai multe nivele de securitate, care oferă flexibilitatea necesară pentru o platformă deschisă, oferind în același timp protecție pentru toți utilizatorii platformei.

Android a fost proiectat ținînd cont de dezvoltatori. Controale de securitate au fost concepute pentru a reduce povara asupra dezvoltatorilor. Dezvoltatorii pot lucra cu ușurință și se pot baza pe controalele de securitate flexibile. Dezvoltatorii mai puțin familiarizați cu securitatea vor fi protejați de valori implicite sigure.

Android a fost proiectat ținînd cont de utilizatorii de dispozitive. Utilizatorilor li se furnizează vizibilitate asupra modului în care funcționează aplicațiile, precum și controlul asupra acestor aplicații. Acest design include speranța că atacatorii vor încerca să efectueze atacuri comune, cum ar fi atacurile de inginerie socială pentru a convinge utilizatorii să instaleze dispozitive malware, precum și atacurile asupra aplicații terțe pe Android. Android a fost proiectat pentru a reduce atît probabilitatea acestor atacuri cît și pentru a limita foarte mult impactul atacului, în cazul în care acesta a fost reușit.

2.1 Programe de securitate Android

Încă de la începutul dezvoltării, echipa de dezvoltare a Android a recunoscut că un model de securitate robust e necesar pentru a permite un ecosistem viguros de aplicații și dispozitive construite pe și în jurul platformei Android și susținute de servicii de cloud. Ca urmare, prin intermediul întregului ciclului de dezvoltare, Android a fost supus unui program de securitate profesional. Echipa Android a avut ocazia de a observa modul în care celelalte mobile, desktop, și platforme server împiedică și reacționează la problemele de securitate și de a construi un program de securitate pentru a aborda punctele slabe observate în alte oferte.

Componentele cheie ale Programului de Securitate Android includ:

Design Review: procesul de securitate Android începe devreme în ciclul de dezvoltare, cu crearea unui model de securitate bogat și configurabil. Fiecare caracteristică majoră a platformei revizuie resusele de inginerie și securitate, cu controale de securitate corespunzătoare integrate în arhitectura sistemului.

Testarea și prezentarea codului: în timpul dezvoltării platformei, Android și componentele open-source create sunt supuse unor evaluări de securitate viguroase. Aceste evaluări sunt efectuate de către echipa de securitate Android, echipa Google Information Security Engineering, și consultanți independenți de securitate. Scopul acestor evaluări este de a identifica punctele slabe și punctele vulnerabile posibile precum și pentru a simula tipuri de analiză, care vor fi efectuate de către experți externi.

Open Source și opinia comunității: Proiectul Android Open Source permite să fie efectuată evaluarea securității de către orice parte interesată. Android folosește, de asemenea, tehnologiile open-source, care au fost supuse evaluărilor externe semnificative, cum ar fi kernel-ul Linux. Google Play oferă un forum pentru utilizatorii și întreprinderile unde furnizează informații cu privire la aplicații specifice direct la utilizatori.

Reacția la incidente: Chiar și cu toate aceste măsuri de precauție, problemele de securitate pot apărea dupa transportare, motivul pentru care proiectul Android a creat un proces de securitate cuprinzător ca răspuns. O echipa de securitate Android monitorizează în mod constant comunitatea de securitate specifică Android și generală pentru discutarea potențialelor vulnerabilități. La descoperirea problemelor legitime, echipa Android are un proces de răspuns care permite atenuarea rapidă a vulnerabilităților pentru a se asigura că riscul potențial pentru toți utilizatorii Android este minimizat. Aceste răspunsuri pot include actualizarea platformei Android (actualizări over-the-air), eliminarea aplicațiilor de la Google Play, precum și eliminarea aplicațiilor de la dispozitive din domeniu.

2.2 Arhitectura de securitate a platformei Android

Android urmărește să fie cel mai sigur și ușor de utilizat sistem de operare pentru platforme mobile prin modificarea scopului controalelor de securitate a sistemelor de operare pentru a:

- Proteja datele utilizatorului
- Proteja resurselor de sistem (inclusiv de rețea)
- Oferi izolarea aplicației.

Pentru a atinge aceste obiective, Android oferă aceste caracteristici cheie de securitate:

- Securitate robustă la nivelul sistemului de operare prin intermediul kernel-ului Linux
- Sandbox-ul aplicației obligatoriu pentru toate aplicațiile
- Comunicare între procese securizată
- Semnătura aplicației
- Permisiuni definite pe aplicație și permise de utilizator.

În figura 2.2 sunt sumarizate componentele de securitate și considerentele diferitor niveluri ale stivei software-ul Android. Fiecare componentă presupune că componentele de mai jos sunt securizate corespunzător. Cu excepția unei cantități mici de cod Android OS rulează ca root, tot codul de mai sus de kernel-ul Linux este restricționat de Application Sandbox.

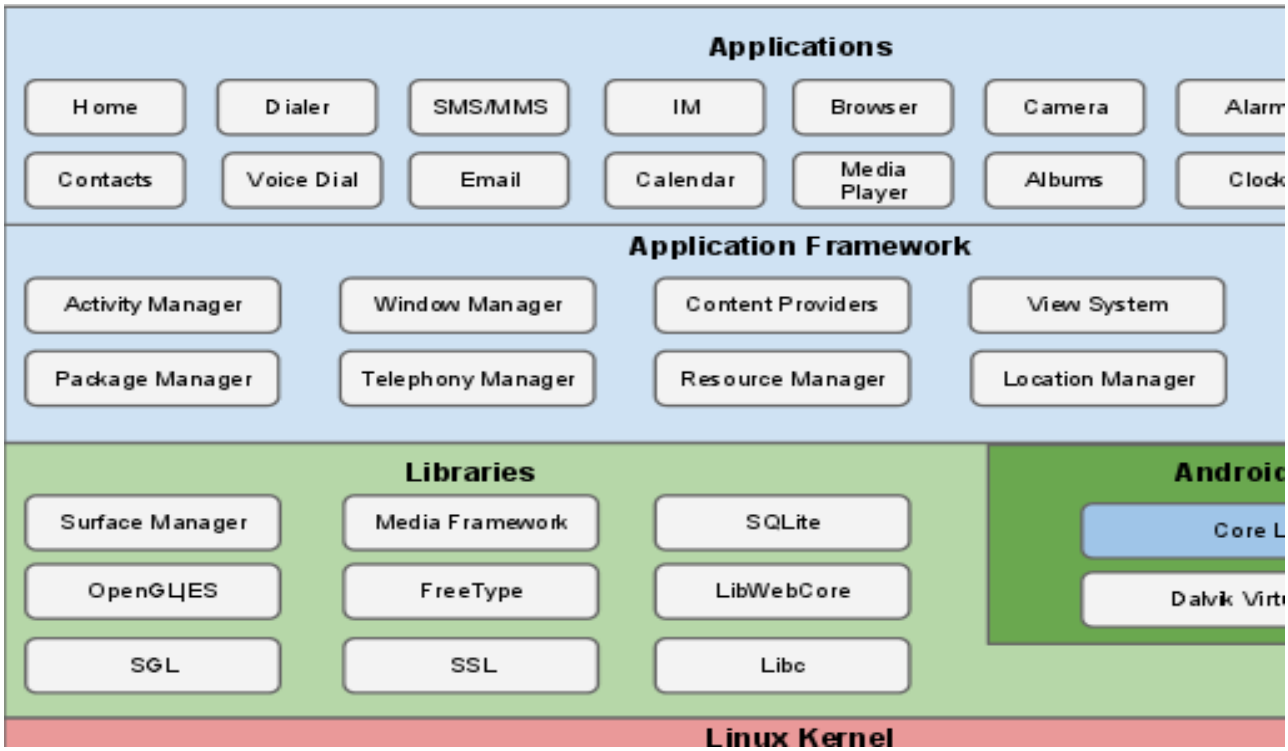


Figura 2.2. Stiva software Android

2.3 Nivelul de securitate a sistemului și Kernel-ului

La nivel de sistem de operare, platforma Android oferă securitatea kernel-ului Linux, precum și facilitatea de comunicare între procese securizat (IPC) pentru a activa comunicarea securizată între aplicațiile care rulează în diferite procese. Aceste caracteristici de securitate la nivel de sistem de operare asigură că și codul nativ este constrîns de Sandbox Application. Dacă acest cod este rezultatul comportamentului aplicației incluse sau o exploatare a unei vulnerabilități a aplicației, sistemul ar împiedica aplicația să afecteze alte aplicații, sistemul Android, sau însuși dispozitivul.

2.3.1 Securitate Linux

Fundația platformei Android este kernel-ul Linux. Kernel-ul Linux în sine a fost în uz pe scară largă de ani de zile, și este utilizat în milioane de medii sensibile de securitate. Prin istoria sa fiind cercetat în mod constant, atacat, și fixat de mii de dezvoltatori, Linux a devenit un kernel stabil și sigur, în care au încredere multe companii și profesioniști în securitate.

Ca bază pentru un mediu de calcul mobil, kernel-ul Linux oferă platformei Android cu mai multe caracteristici cheie de securitate, inclusiv:

- Modelul de permisiuni bazat pe utilizator
- Proces de izolare
- Mecanism extensibil pentru IPC securizat
- Capacitatea de a elimina piese inutile și potențial nesigur ale kernel-ului

2.3.2 Application Sandbox

Platforma Android profită de protecția Linux bazată pe utilizator ca mijloc de identificare și izolare a resurselor de aplicare. Sistemul Android atribuie un ID unic de utilizator (UID) pentru fiecare aplicație Android și îl rulează ca acel utilizator într-un proces separat. Această abordare este diferită de alte sisteme de operare (inclusiv configurația tradițională Linux), în cazul în care mai multe aplicații rulează cu aceleași permisiuni de utilizator.

Aceasta stabilește un Application Sandbox la nivelul de kernel. Kernel-ul impune securitatea între aplicații și sistem la nivel de proces prin facilități standarde Linux, cum ar fi ID-urile de utilizator și de grup, care sunt atribuite aplicațiilor. În mod implicit, aplicațiile nu pot interacționa unele cu altele și au acces limitat la sistemul de operare. Dacă aplicația A încearcă să facă ceva rău intenționat cum ar fi să citească datele aplicației B sau să apeleze telefonul fără permisiune (care este o aplicație separată), atunci sistemul de operare protejează împotriva acestei aplicații, deoarece aplicația A nu are privilegiile corespunzătoare de utilizator.

Deoarece Application Sandbox este în kernel, acest model de securitate se extinde la cod nativ și la aplicații ale sistemului de operare. Tot software-ul de mai sus de kernel în Figura 1, inclusiv bibliotecile sistemului de operare, cadru de aplicare, cererea de rulare, precum și toate aplicațiile rulează în cadrul Application Sandbox. Pe unele platforme, dezvoltatorii sunt constrînși la un cadru de dezvoltare specific, set de API-uri, sau de limbă, cu scopul de a impune securitate. Pe Android, nu există restricții privind modul în care o aplicație poate fi scrisă; în acest sens, codul nativ este la fel de sigur ca și codul interpretat.

În unele sisteme de operare, erori de corupere a memoriei duc, în general, la compromiterea completă a securității dispozitivului. Acesta nu este cazul în Android datorită tuturor aplicațiilor și resurselor acestora. O eroare de corupere a memoriei va permite doar executarea codului arbitrar, în cadrul aplicației specifice, cu permisiunile stabilite de sistemul de operare.

Ca toate caracteristicile de securitate, Application Sandbox nu este indestructibil. Cu toate acestea, pentru a îl sparge într-un dispozitiv configurat corect, trebuie să fie compromisă securitatea kernel-ului Linux.

2.3.3 Partiție de sistem și Safe Mode

Partiția de sistem conține kernel-ul Android, precum și bibliotecile sistemului de operare, timpul de execuție a aplicației, cadrul de aplicare, și aplicațiile. Această partiție este setată la modul read-only. Cînd un utilizator conectează dispozitivul în Safe Mode, doar aplicațiile Android de bază sunt disponibile. Acest lucru asigură faptul că utilizatorul poate porni telefonul într-un mediu care este liber de software-ul părților terțe.

2.3.4 Permisiuni FileSystem

Într-un mediu în stil UNIX, permisiunile sistemului de fișiere asigură că un utilizator nu poate modifica sau citi fișierele altui utilizator. În cazul Android, fiecare aplicație rulează ca utilizator proprie. Cu excepția cazului în care dezvoltatorul expune în mod explicit fișierele altor aplicații, fișierele create de o aplicație nu pot fi citite sau modificate de către o altă aplicație.

2.3.5 Criptare FileSystem

Android 3.0 și versiunile mai tîrzii, asigură criptarea completă a sistemului de fișiere, astfel toate datele utilizatorului pot fi criptate în kernel-ul folosind imlementarea dmcrypt a AES128 cu CBC și ESSIV: SHA256. Cheia de criptare este protejată de AES128 folosind o cheie derivată din parola utilizatorului, prevenind accesul neautorizat la datele stocate, fără parola dispozitivului utilizatorului. Pentru a oferi rezistență față de atacurile de ghicire sistematică a parolei, parola este combinată la întîmplare și trunchiată în mod repetat cu SHA1 folosind algoritmul standard PBKDF2 înainte de a fi utilizat pentru a decripta cheia sistemului de fișiere. Pentru a oferi rezistență față de atacuri de ghicire a parolei din dicționar, Android prevede reguli de complexitate a parolei, care pot fi stabilite de către administratorul de dispozitiv și puse în aplicare de către sistemul de operare. Criptarea sistemului de fișiere necesită utilizarea unei parole de utilizator, modelul bazat pe ecran de blocare nu este acceptată.

2.3.6 Protejarea parolei

Android poate fi configurat pentru a verifica o parolă de utilizator furnizată înainte de a oferi acces la un dispozitiv. În plus, pentru a preveni utilizarea neautorizată a dispozitivului, această parolă protejează cheia criptografică pentru criptarea completă a sistemului de fișiere.

Utilizarea unei parole și / sau a regulilor de complexitate a parolei poate fi cerută de către un administrator al dispozitivului.

2.3.7 Administrarea dispozitivului

Android 2.2 și versiunile mai tîrzii furnizează Android Device Administration API, care oferă caracteristici de administrare ale dispozitivului la nivel de sistem. De exemplu, aplicația integrată Android Emai utilizează API-uri pentru a îmbunătăți suportul de schimb. Prin aplicația Email, administratorii Exchange pot aplica politici de parole - inclusiv parole alfanumerice sau PIN-uri numerice - pe tot dispozitivul. Administratorii pot, de asemenea, șterge de la distanță (restabili setările implicite) dispozitivele pierdute sau furate.

2.3.8 Root pentru dispozitiv

În mod implicit, pe Android doar kernel-ul și un subset mic de aplicații de bază se execută cu permisiuni de root. Android nu împiedică un utilizator sau o aplicație cu permisiuni de root de la modificarea sistemului de operare, kernel-ului, precum și oricărei alte aplicație. În general, root are acces deplin la toate aplicațiile și datele lor. Utilizatorii care schimbă permisiunile pe un dispozitiv Android pentru a acorda acces root aplicațiilor, sporesc expunerea de securitate față de aplicații malware și potențiale defecte ale aplicațiilor.

Capacitatea de a modifica un dispozitiv Android pe care le dețin, este important pentru dezvoltatorii care lucrează cu platforma Android. Pe mai multe dispozitive Android utilizatorii au posibilitatea de a debloca bootloader-ul pentru a permite instalarea unui sistem de operare alternativ.

Aceste sisteme de operare alternative pot permite unui proprietar să cîștige acces root pentru scopuri de depanare a aplicațiilor și componentelor de sistem sau pentru a accesa funcții care nu sunt prezentate la aplicații de către Android API.

La unele dispozitive, o persoană cu un control fizic al unui dispozitiv și un cablu USB este capabil de a instala un sistem de operare nou, care oferă privilegii de root pentru utilizator. Pentru a proteja toate datele utilizatorilor existente de la compromitere, mecanismul de deblocare bootloader necesită ca bootloader-ul să șteargă orice date ale utilizatorilor existente ca parte a etapei de deblocare. Accesul root cîștigat prin exploatarea unei erori de kernel sau unei breșe de securitate poate ocoli această protecție.

Criptarea datelor cu o cheie stocată pe dispozitiv nu protejează datele aplicației de utilizatorii cu acces root. Aplicațiile pot adăuga un nivel de protecție a datelor folosind criptarea cu o cheie stocată pe dispozitiv, cum ar fi pe un server sau o parolă de utilizator. Această abordare poate oferi o protecție temporară în timp ce cheia nu este prezentă, dar la un moment dat cheia trebuie să fie furnizată aplicației și apoi devine accesibilă pentru utilizatorii cu acces root.

În cazul pierderii sau furtului dispozitivului, criptarea completă a sistemului de fișiere pe dispozitivele Android utilizează parola dispozitivului pentru a proteja cheia de criptare, astfel modificînd bootloader-ul sau sistemul de operare nu este suficientă pentru a accesa datele utilizatorului fără parola dispozitivului utilizatorului.

2.4 Securitatea aplicațiilor Android

În mod implicit, o aplicație Android poate accesa doar o gamă limitată de resurse de sistem. Sistemul gestionează accesul la resursele aplicațiilor Android care, dacă sunt folosite incorect sau cu rea intenție, ar putea avea un impact negativ asupra experienței utilizatorului, rețelei, sau a datelor de pe dispozitiv.

Aceste restricții sunt puse în aplicare într-o varietate de forme diferite. Unele capacități sunt restricționate de o lipsă intenționată de API-uri pentru funcționalitate (de exemplu, nu există nici un API pentru manipularea directă a cartelei SIM). În unele cazuri, separarea rolurilor oferă o măsură de securitate. În alte cazuri, API-urile sunt destinate utilizării de către aplicații de încredere și protejate printr-un mecanism de securitate cunoscut sub numele de Permissions.

Aceste API-uri protejate includ:

- Funcțiile camerei
- Datele de localizare (GPS)
- Funcțiile bluetooth
- Funcțiile telefoniei
- Funcții SMS / MMS
- Conexiuni la rețea / date

Aceste resurse sunt accesibile numai prin intermediul sistemului de operare. Pentru a face uz de API-uri protejate de pe dispozitiv, o aplicație trebuie să definească capacitățile de care are nevoie. Cînd se pregătește să instaleze o aplicație, sistemul afișează o casetă de dialog utilizatorului, care indică permisiunile solicitate și întreabă dacă să continue instalarea. Dacă utilizatorul continuă instalarea, sistemul acceptă faptul că utilizatorul a acordat toate permisiunile solicitate. Utilizatorul nu poate acorda sau refuza permisiuni individuale - utilizatorul trebuie să acorde sau să refuze toate permisiunile solicitate în formă de bloc.

Odată acordate, permisiunile se aplică aplicației atît timp cît este instalată. Pentru a evita confuzia utilizatorului, sistemul nu notifică utilizatorul din nou de permisiunile acordate aplicației, și aplicațiile care sunt incluse în sistemul de operare de bază nu solicită permisiuni de la utilizator. Permisiunile sunt eliminate în cazul în care o aplicație este dezinstalată, deci o ulterioară re-instalare va afișa din nou permisiunile.

În cadrul setărilor dispozitivului, utilizatorii au posibilitatea de a vedea permisiunile pentru aplicațiile pe care le-au instalate anterior. Utilizatorii pot dezactiva anumite funcționalități la nivel global atunci cînd aleg, cum ar fi dezactivarea GPS, radio, sau Wi-Fi.

În cazul în care o aplicație încearcă să utilizeze o caracteristică protejată, care nu a fost declarată în manifestul aplicației, eșecul permisiunii tipic va rezulta într-o excepție de securitate fiind aruncată înapoi la aplicație. Controalele de permisiuni API protejate sunt puse în aplicare la cel mai mic nivel posibil pentru a preveni eludarea. Un exemplu de mesagerie a utilizatorului atunci cînd o aplicație este instalată în timp ce solicită accesul la API-uri protejate este prezentat în Figura 2.4.1.

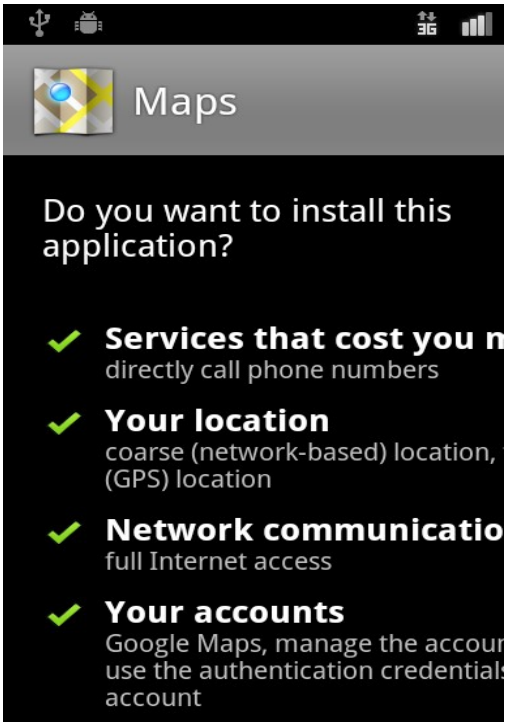
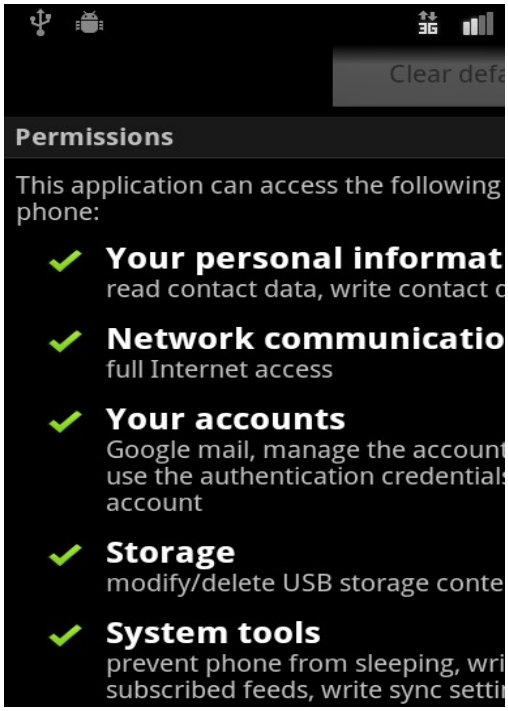
Permissions at Application Install -- Google Maps	**Permissions of an Installed Application – gMail**
	

Figura 2.4.1. Prezentarea permisiunilor pentru aplicații

2.4.1 Cum înțeleg utilizatorii aplicațiile părților terțe

Android încearcă să realizeze interacțiunea utilizatorilor cu aplicațiile părților terțe cît mai clară și să informeze utilizatorul ce capacități au aceste aplicații. Înainte de instalarea oricărei aplicații, utilizatorului îi este indicat un mesaj clar cu privire la diverse permisiuni pe care le solicită aplicația. După instalare, utilizatorul nu mai este rugat să confirme orice permisiune.

Există multe motive pentru a prezenta permisiunile înainte de instalare. Este momentul cînd utilizatorul revizuiește activ informații despre aplicație, dezvoltator, și funcționalitate pentru a stabili dacă acestea corespund nevoilor și așteptărilor lui. De asemenea, este important că acesta încă nu a stabilit un angajament mental sau financiar față de aplicație, și o poate compara cu ușurință cu alte aplicații alternative.

Unele platforme folosesc o abordare diferită a notificării utilizatorului, solicitînd permisiunea la începutul fiecărei sesiuni sau în timp ce aplicațiile rulează. Viziunea Android este că utilizatorii să aibă libertatea de a schimba aplicațiile fără probleme atunci cînd doresc. Fiind nevoiți să confirme de fiecare dată, ar încetini utilizatorul și nu va permite platformei Android să îi livreze o experiență excelentă.

Avînd permisiunile de revizuire la instalare, utilizatorul are opțiunea de a nu instala aplicația dacă se simte inconfortabil.

De asemenea, multe studii a interfeței de utilizator au arătat că prezența a multor ferestre de confirmare îl determină pe utilizator să înceapă să spună "OK" pentru fiecare dintre ele. Unul din obiectivele de securitate ale Android este de a transmite în mod eficient informațiile de securitate importante pentru utilizator, care nu poate fi realizat cu ajutorul ferestrelor de dialog pe care utilizatorul va fi instruit să le ignore. Prin prezentarea informațiilor importante o dată, și numai atunci cînd este important, utilizatorul este mult mai probabil să gîndească cu ce este de acord.

Unele platforme aleg să nu prezinte nici o informație despre funcționalitatea aplicațiilor. Această abordare împiedică utilizatorii să înțeleagă ușor și să discute capacitățile aplicației. În timp ce nu este posibil pentru toți utilizatorii să facă întotdeauna decizii bazate pe cunoștințe, modelul de permisiuni Android face informația despre aplicații ușor accesibilă pentru o gamă largă de utilizatori. De exemplu, cererile neprevăzute de permisiuni pot solicita utilizatorilor mai sofisticați să pună întrebări critice despre funcționalitatea aplicației și să împărtășească preocupările lor în locuri cum ar fi Google Play unde sunt vizibili pentru toți utilizatorii.

2.4.2 Acces la cartela SIM

Accesul de nivel scăzut la cartela SIM nu este disponibilă pentru aplicații terțe. Sistemul de operare gestionează toate comunicările cu cartela SIM, inclusiv accesul la informații personale (contacte) pe memoria cartelei SIM. Aplicațiile, de asemenea, nu pot accesa comenzi AT, astfel cum acestea sunt gestionate exclusiv de Radio Interface Layer (RIL). RIL nu furnizează API-uri de nivel înalt pentru aceste comenzi.

2.4.3 Informații personale

Android a pus API-uri care permit accesul la datele utilizatorilor în setul de API-uri protejate. Cu o utilizare normală, dispozitivele Android vor acumula, de asemenea, datele de utilizator în cadrul aplicațiilor terțe instalate de către utilizatori. Aplicațiile care aleg să împărtășească această informație pot folosi controalele de permisiune a sistemului de operare Android pentru a proteja datele de aplicații terțe.

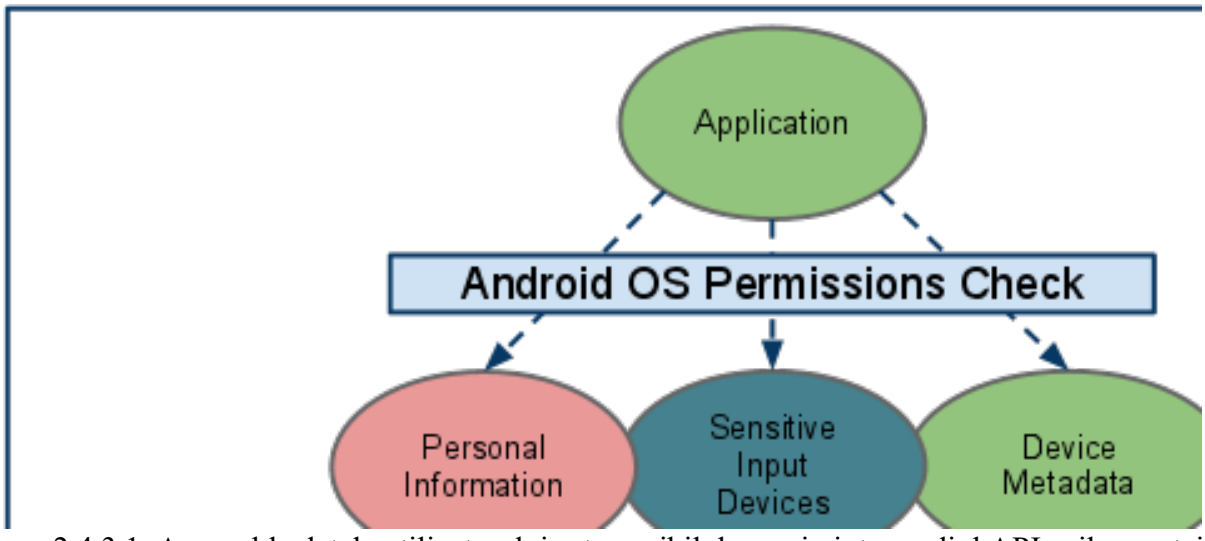


Figura 2.4.3.1. Accesul la datele utilizatorului este posibil doar prin intermediul API-urilor protejate

Furnizorii de conținut ai sistemului, care pot conține informații personale sau de identificare personală, cum ar fi contactele și calendarul, au fost creați cu permisiuni clar identificate. Această granularitate oferă utilizatorului indicații clare a tipurilor de informații care pot fi furnizate aplicației. În

timpul instalării, o aplicație terță poate cere permisiunea de a accesa aceste resurse. Dacă permisiunea este acordată, aplicația poate fi instalată și va avea acces la datele solicitate în orice moment.

Orice aplicație care colectează informații cu caracter personal în mod implicit vor avea aceste date limitate numai pentru aplicații specifice. În cazul în care o aplicație alege să facă datele disponibile pentru alte aplicații prin IPC, aplicația care acordă acces poate aplica permisiuni pentru mecanismul IPC, care este pus în aplicare de către sistemul de operare.

2.4.4 Dispozitive de introducere a datelor sensibile

Dispozitivele Android oferă frecvent dispozitive de introducere a datelor sensibile, care permit aplicațiilor să interacționeze cu mediul înconjurător, cum ar fi camera foto, microfon sau GPS. Pentru o aplicație terță ca să acceseze aceste dispozitive, trebuie mai întâi furnizate în mod explicit accesul de către utilizator prin utilizarea de Android OS Permissions.

Dacă o aplicație dorește să știe locația utilizatorului, aplicația necesită o permisiunea de a accesa locația utilizatorului. După instalare, programul de instalare va întreba utilizatorul dacă aplicația poate accesa locația utilizatorului. În orice moment, în cazul în care utilizatorul nu dorește nici o aplicație să acceseze locația, poate rula aplicația "Settings", să meargă la "Location & Security", și să debifeze "Use wireless networks" și "Enable GPS satellites". Acest lucru va dezactiva serviciile bazate pe localizare pentru toate aplicațiile de pe dispozitivul utilizatorului.

2.4.5 Metadata dispozitivului

Android, de asemenea, încearcă să limiteze accesul la date care nu sunt intrinsec sensibile, dar pot dezvălui indirect caracteristicile legate de utilizator, preferințele utilizatorului, precum și modul în care acesta folosește un dispozitiv.

Aplicațiile implicite nu au acces la log-urile sistemului de operare, istoricul browser-ului, număr de telefon, sau informația de identificare hardware / rețea. Dacă o aplicație cere acces la această informație în momentul instalării, programul de instalare va întreba utilizatorul dacă aplicația poate accesa informațiile. În cazul în care utilizatorul nu permite accesul, aplicația nu va fi instalată.

2.4.6 Semnătura aplicațiilor

Semnarea codului permite dezvoltatorilor să identifice autorul aplicației și să actualizeze aplicarea acesteia, fără a crea interfețe complicate și permisiuni. Fiecare aplicație care rulează pe platforma Android trebuie să fie semnată de către dezvoltator. Aplicațiile care încearcă să instaleze, fără a fi semnate sunt respinse fie de către Google Play fie de programul de instalare pe dispozitivul Android.

Pe Google Play, semnarea aplicației identifică încrederea pe care Google o are față de dezvoltator și încrederea dezvoltatorului în aplicația lui. Dezvoltatorii știu că aplicația lor este livrată nemodificată la dispozitivul Android, și dezvoltatorii pot fi trași la răspundere pentru comportamentul aplicației lor.

Pe Android, semnarea aplicației este primul pas pentru introducerea unei aplicații în Application Sandbox. Certificatul aplicației semnate definește care ID de utilizator este asociat cu care aplicație; diferite aplicații rulează sub ID-uri de utilizator diferite. Semnarea aplicației garantează că o aplicație nu poate accesa orice altă aplicație decît prin IPC bine definit.

Atunci cînd o aplicație (fișier APK) este instalat pe un dispozitiv Android, Package Manager verifică dacă APK a fost corect semnat cu certificatul inclus. În cazul în care certificatul (sau, mai exact, cheia publică în certificat) se potrivește cu cheia utilizată pentru a semna orice alt APK pe dispozitiv, noul APK are opțiunea de a specifica în manifest că va împărți un UID cu alte APK semnate similar.

Aplicațiile pot fi semnate de către o parte terță (OEM, operator, piața alternativă) sau auto-semnate. Android prevede semnarea codului folosind certificate auto-semnate, pe care dezvoltatorii le

pot genera fără asistența externă sau permisiune. Aplicațiile nu trebuie să fie semnate de către o autoritate centrală. Android în prezent, nu efectuează verificarea certificatelor CA pentru aplicații.

Aplicațiile sunt, de asemenea, capabile să declare permisiunile de securitate la nivelul protecției semnăturii, limitînd accesul numai la aplicații semnate cu aceeași cheie menținînd în același timp UID-uri distincte.

2.4.7 Management-ul drepturilor digitale

Platforma Android oferă un cadru extensibil DRM care permite aplicațiilor să gestioneze conținutul protejat de drepturi în conformitate cu constrîngerile licenței care sunt asociate conținutului. Cadrul DRM suportă multe scheme DRM.

Cadrul Android DRM este implementat în două nivele arhitecturale (Figura 2.4.7.1):

• Un API al cadrului DRM, care este expus la aplicații prin cadrul de aplicații Android și rulează prin Dalvik VM pentru aplicații standarde.

• Un manager de cod nativ DRM, care implementează cadrul DRM și expune o interfață pentru plug-inurile (agenți) DRM să gestioneze drepturile de management și decriptarea pentru diferite scheme DRM.

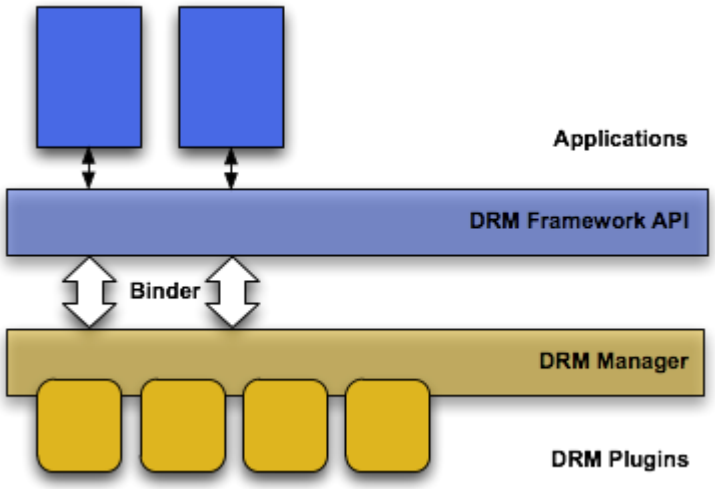


Figure 2.4.7.1: Architectura Digital Rights Management pe platforma Android

2.5 Actualizări Android

Android oferă actualizări de sistem, atît pentru securitate cît și pentru scopuri legate de caracteristici.

Există două moduri de a actualiza codul pe majoritatea dispozitivelor Android: over-the-air (actualizări OTA) sau actualizări side-loaded. Actualizările OTA pot fi rulate pe o perioadă de timp definită sau propuse la toate dispozitivele dintr-o dată, în funcție de modul în care OEM și / sau transportatorul ar dori să propună actualizările. Actualizările side-loaded pot fi furnizate de o locație centrală pentru utilizatori să descarce un fișier zip pe desktop-ul lor sau direct pe telefonul lor. Odată ce actualizarea este copiată sau descărcată pe cardul SD de pe dispozitiv, Android va recunoaște actualizarea, va verifica integritatea și autenticitatea, și va actualiza automat dispozitivul.

În cazul în care o vulnerabilitate periculoasă este descoperită în interior sau raportată în mod responsabil la Google sau Android Open Source Project, echipa de securitate Android va începe următorul proces.

1. Echipa Android va notifica companiile care au semnat NDA cu privire la problemă și va începe discutarea soluției.
2. Proprietarii de cod vor începe să-l fixeze.
3. Echipa Android va stabili problemele de securitate legate de Android.
4. Atunci cînd un patch este disponibil, fixarea este oferită de companiile NDA.
5. Echipa Android va publica patch-uri în cadrul Proiectului Open Source Android
6. OEM / transportatorul va propune o actualizare clienților.

NDA trebuie să asigure că problema securității nu devine publică înainte de disponibilitatea unei soluții.