Securitatea și confidențialitatea datelor în contextul aplicațiilor mobile

Ghimpu Lucian Eduard

April 7, 2019

Conducător științific
Dr. CZIBULA Istvan, Profesor Universitar

Contents

1 Introducere

1.1 Introducere

1.2 Ecosistemul aplicațiilor mobile

1.3 GDPR

2 Securitatea și confidențialitatea datelor în contextul aplicațiilor mobile

2.1 Autentificarea si înregistrarea

2.1.1 Ceva introducere *

Indiferent ca vorbim de aplicatii web, desktop sau mobile, majoritatea folosesc o metoda de autentificare. Auntentificarea și înregistrarea stau la baza problematicii securitatii datelor. In clasamentul OWASP Top 10 din 2017 [1], problemele legate de autentificare si gestiunea sesiunii, sunt clasate pe locul 2. Iar in clasamentul OWASP top 10 Mobile din 2016 [2], autentificare nesigura si autorizarea necurespunzatoare sunt clasate pe locul 4, respectiv 6.

Unicitatea aplicatilor mobile este data de faptul ca un dispozitiv mobil poate devenii accesibil oricarui persoane datorita portabilitatii lor. Un dispozitiv mobil poate fi furat, pierdut sau accesat temporal de o persoana necunoscuta fara permisiunea posesorului. Prin urmare nevoia de un sistem de autentificare robust este mandatorie atunci cand vorbim de aplicatii care gestioneaza date sensibile (aplicatii financiare, sociale, medicale, etc...).

In cadrul aplicatiilor mobile, autentificarea se poate face prin mai multe metode. De la simpla autentificare prin utilizator si parola, pana la utilizarea de sezori biometrici. Mitigari clasice precum impunerea unei parole sigure raman valabile si in contextul aplicatiilor mobile.

Pentru alegerea metodei de autentificare trebuie sa ne punem in prima instanta urmatoarele doua intrebari:

- Care este scopul aplicatiei? O aplicatie care notifica utilizatorul despre starea meteo poate ca nu ar avea nevoie de autentificare prin senzori biometrici.
- Aplicatia gestioneaza date confidentiale? Un exemplu potrivit ar fi o aplicatie precum BT pay, care gestioneaza contul curent al unui utilizatorul se foloseste de mai multe metode de autentificare, o data prin datele de logare, iar apoi prin senzori biometrici.

Dupa ce ne este clar care este scopul aplicatiei si cu ce fel de date lucreaza, putem include una sau mai multe metode de autentificare bazate dupa urmatorii factori:

1. Ceva ce utilizatorul stie (parola, pin etc. . .)
2. Ceva ce il defineste pe utilizator (amprenta, retina)
3. Ceva ce utilizatorul detine (parole generate temporal)

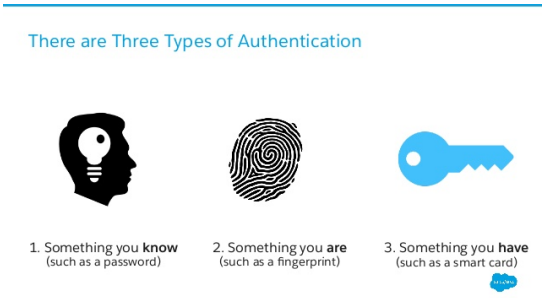

Figure 1: Metode de autentificare [3]

2.1.2 JWT si OAuth2

Majoritatea aplicatiilor de azi se folosesc de cea mai simpla forma de autentificare, prin folosirea de credentiale (ceva ce utilizatorul stie) si unui token de acces (ceva ce utilizatorul detine). Utilizatorul isi creeza cont pentru o anumita aplicatie, foloseste credentialele pentru a se loga, cererea de autentificare ajunge la server unde se verifica credentialele iar apoi se genereaza un token care va fi folosit de utilizator pentru a accesa diferite resurse in aplicatie.

Scenariu descris anterior se refera la folosirea de JWT (JSON Web Token), un standard (RFC 7519) [4] adoptat de multe aplicatii mobile in zilele noastre. JSON Web Token este o metoda sigura de autorizare a trasferului de informatii intre doua parti [5], deobicei clientul mobil si serverul la care se face cererea. Clientul revendica de la server o dovada, un token, care apoi este folosit de client pentru a accesa diferite resurse.

Din punct de vedere tehnic, un JWT are urmatoarea forma 11111.22222.33333 si este alcauit din 3 parti:

1. Antetul (Header)
2. Datele utile (Payload)
3. Semnatura (Signature)

Figure 2: Antetul unui JWT, alcatuit din tipul de algoritm de inregistrare (HS256) si tipul de token (JWT).

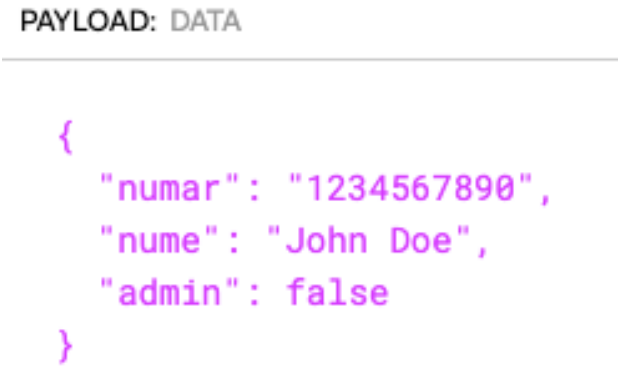

Figure 3: Partea utila al unui JWT contine date sau permisiuni pe care clientul le are.

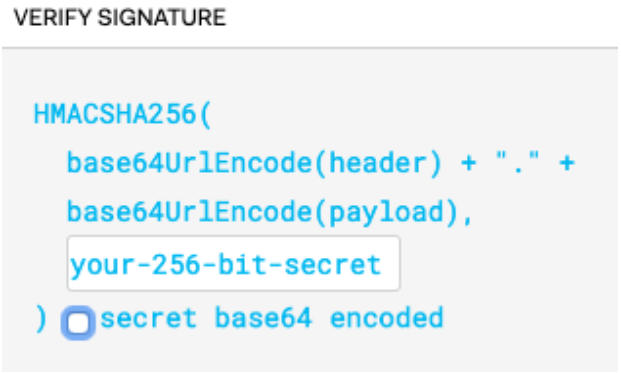

Figure 4: Semnatura unui JWT este alcatuita din antetul encodat, datele encodate, algoritmul folosit in antet si un secret. Semnatura are rolul de a oferi o metoda de verificare pentru a asigura ca continutul nu a fost modificat

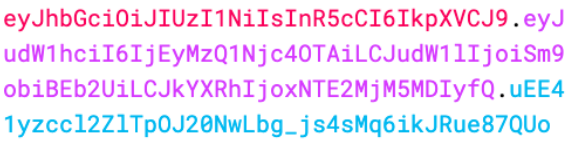

Figure 5: JWT va fi format in final de 3 siruri de caractere de tip Base64-URL separate prin punct.

O data ce clientul detine un token, acesta trebuie tratat cu multa grija in cadrul unei aplicatii. Acesta poate fi folosit mai apoi in antetul tuturor cererilor de tip HTTP sub forma Authorization: Bearer ¡token¿. Din acest motiv cel mai probabil se doreste salvarea token-ului in memoria locala a dispozitivului mobil pentru a putea fi apoi folosit in viitor. Acesta poate fi criptat iar la randul lui la nivelul clientului, desi in mod nativ atat pe android cat si pe ios exista metode sigura de stocare a date de tip token (SharedPreferences in mod private pe Android si keychain pe iOS).

Pentru un nivel si mai mare de siguranta, se poate limita durata de timp pe care este valabil un token. Spre exemplu aplicatia BT Pay foloseste un token care este valid tip de 10-15 minute. Dupa ce token-ul expira, utilizatorul este nevoit sa se autentifice din nou.

Avantajul principal pe care il ofera JWT este facilitatea prin care se demareaza tot procesul prin care un client isi revendica datele sau drepturile de la un server. Un alt aspect important il reprezinta faptul ca in spate, totul se produce folosit obiecte de tipul JSON, obiecte care sunt extrem de raspandite in orice limbaj de programare si in tot internetul.

List of Figures

References

[1] OWASP Top 10 2017
https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf

[2] OWASP Top 10 Mobile 2016
https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10

[3] 3 types of Authentication
https://www.slideshare.net/awesomeadmin/secure-your-salesforce-org-with-twofactor-authe

[4] RFC 7519 JWT
https://tools.ietf.org/html/rfc7519

[5] JSON Web Token (JWT)
https://tools.ietf.org/html/rfc7519

[6] ENISA, Privacy and data protection in mobile applications, 29.01.2019
https://www.enisa.europa.eu/publications/privacy-and-data-protection-in-mobile-applicat

[7] ENISA, Smartphone Secure Development Guidelines, 27.02.2017,
https://www.enisa.europa.eu/publications/smartphone-secure-development-guidelines-2016

[8] OWASP Mobile Application Security Verification Standard,
https://github.com/OWASP/owasp-masvs