Mecanisme de asigurare a identităţii digitale. Autentificarea prin factori multipli

Cornel Ion ŞERBAN[1], Ştefan MOCANU[2], Cosmin POPA[3]

[1] Academia de Studii Economice, Bucureşti, România,

[2] Facultatea de Automatică şi Calculatoare, UPB, Bucureşti, România,

[3] Ingenios.Ro, Str. George Constantinescu 2C, Bucureşti, România,

cornel.serban@tgssoftware.ro, smocanu@rdslink.ro, cosmin.popa@ingenios.ro

Rezumat: În vederea dezvoltării de tehnologii pentru a valorifica conceptul de identitate digitală este nevoie de creşterea securităţii sistemelor folosite prin elaborarea unor modele care au la bază date ale acestor sisteme. Astfel de date sunt acumulate în urma autentificărilor utilizatorilor. În prezent, procesul de autentificare a utilizatorilor pentru recunoaşterea identităţii unei persoane sau a unei organizaţii a devenit o problemă foarte importantă. Există doi termeni reprezentativi atunci când vrem să facem referire la identitatea unei persoane sau a unei entităţi: identificarea, care defineşte procesul de a atesta identitatea unei persoane sau a unui lucru şi autentificarea, care reprezintă procesul de confirmare a identităţii. Scopul acestui articol este de a prezenta autentificarea prin doi factori, ce are rolul de a creşte eficienţa autentificării pe bază de utilizator şi parolă. Pentru acest tip de autentificare au fost introduşi diferiţi factori care pot fi combinaţi pentru a imbunătăţi siguranţa accesului la diverse resurse. În acest context, lucrarea ce urmează a fi prezentată cuprinde atât avantajele, cât şi dezavantajele unei astfel de autentificări, dar şi modul în care aceasta poate fi implementată (pe baza unui token instalat pe telefonul mobil). Pentru a evidenţia aspectele teoretice menţionate cu referire la identitatea digitală şi autentificare, a fost proiectată şi implementată o aplicaţie de tip mobil ce va fi prezentată pe larg în Capitolul 3.

Cuvinte cheie: autentificare, securitate, identitate digitală, autentificare cu doi factori, token, autentificare cu factori multipli.

Digital Identity Assurance Mechanisms. Multiple Factors Based Authentication

Abstract: In order to develop new technologies capable to exploit the digital identity concept, increasing the security and safety of involved systems by elaborating models based on data from the systems is strongly required. The mentioned input data are gathered from users' authentication process. Currently, the authentication process for recognizing the digital identity of one person or organization is raising serious challenges. There are two key terms when someone is referring to digital identity of an entity: identification itself, which defines the process of identifying the person, and authentication, which represents the process of confirming the identity. The main purpose of this study is to present a two factors authentication approach which aims to increase the efficiency of a classical authentication based on username and password. For this type of authentication, different keys can be combined in order to offer a safer access to various resources. This paper presents both advantages and disadvantages of the described approach as well as a practical implementation based on token installed on a smartphone. In order to highlight the afore mentioned theoretical aspects related to digital identity and authentication, a mobile application was designed and implemented. The application is presented in Chapter 3.

Keywords: authentication, security, digital identity, two factor authentication, token, multi-factor authentication.

1. Introducere

Aplicațiile de tip "Machine Learning" pentru autentificarea utilizatorilor au în componență modele moderne care trec de la asigurarea identității unei persoane sau a unei entități la detectarea unei posibile fraude de identitate [3]. În timp, oamenii au ajuns la concluzia că pentru o mai bună securitate a unei aplicaţii nu este suficient să ştii doar numele de utilizator şi parola pentru o persoană sau o entitate. Astfel, nerespectarea anumitor tipare va duce la declanșarea unor mecanisme de avertizare sau blocare a accesului.

Cu alte cuvinte, nivelul de încredere într-un utilizator este cu atât mai ridicat cu cât acesta trece de mai multe etape succesive de autentificare. În vederea prevenirii fraudei de identitate, schimbarea comportamentului de conectare a unui utilizator trebuie semnalată prin canale diferite de canalul inițial (introducere parolă) cum ar fi: trimitere sms, utilizare de token-uri securizate sub formă de aplicații mobile, trimitere email.

Factorii de autentificare reprezintă o informație utilizată pentru a verifica identitatea unei persoane sau a unei organizații care solicită acces pe diferite platforme în diferite domenii: medical, educațional, birouri guvernamentale și mai ales domeniul financiar [4][9]. Pe baza acestor factori de autentificare putem defini trei tehnici de autentificare diferite: autentificarea cu un singur factor, autentificarea prin doi factori [2][13][14] și autentificarea cu mai mult de doi factori (multi-factor) [5][6][11][12].

Modul în care cineva poate fi autentificat poate fi împărțit în următoarele categorii: ceva ce utilizatorul știe, ceva ce utilizatorul are și ceva ce utilizatorul este. Fiecare factor de autentificare acoperă unul sau mai multe astfel de elemente utilizate pentru identificarea și verificarea identității înainte de acordarea accesului, aprobarea unei solicitări de tranzacție, semnarea unui document sau a altui produs de lucru, acordarea autorității altor persoane și stabilirea unui grup de autoritate.

Cercetarea în domeniul securității recomandă, pentru o autentificare corectă, verificarea unui set de elemente din cadrul a cel puțin doi factori (de preferință toți trei).

2. Abordări curente uzuale

Autentificarea care se bazează pe un singur factor reprezintă cel mai slab nivel de autentificare și este bazat pe un singur element menționat deja anterior: ceva ce utilizatorul știe. De regulă acest „ceva” este reprezentat de o parolă însă sunt acceptate și utilizate și alte variante, cum ar fi: cod PIN, răspuns la o întrebare etc. Parola reprezintă elementul pe care utilizatorul îl știe și îl folosește în mod constant la autentificarea pe diferite aplicații, fiind prima metodă de autentificare apărută. De regulă, parolele sunt stocate în baze de date chiar și sub formă de text clar pentru a putea fi comparate mai departe cu datele introduse de utilizator. O astfel de metodă de autentificare (prezentată în Figura 1) nu mai este suficientă pentru a asigura un nivel adecvat de securitate în acest moment.

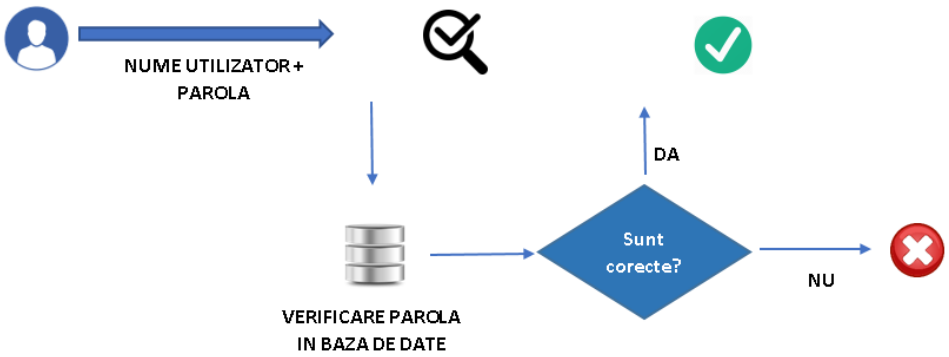

Figura 1. Autentificare cu nume utilizator și parolă

O îmbunătăţire a metodei prezentate anterior este reprezentată de utilizarea funcţiilor *hash* [1] pentru stocarea protejată a parolelor utilizate pentru autentificare. Deşi se încadrează în categoria metodelor de autentificare cu un singur factor, această metodă se bazează pe stocarea parolelor într-o formă criptată și oferă siguranță crescută împotriva atacurilor cibernetice. Funcțiile *hash* sunt neinversabile și fac aproape imposibilă reușita atacatorilor de a sparge conturile utilizatorilor unei anumite aplicații sau platforme. Totuşim există studii care indică existenţa unor probleme în implementările practice ale unor funcţii hash [8], prin urmare se recomandă atenţie sporită chiar și la stocarea credențialelor sub formă de hash-uri. Mecanismul de autentificare bazat pe utilizarea unei funcții *hash* este următorul: atunci când un utilizator își setează o anumită parolă,

această parolă nu este stocată în clar în baza de date, ci ii este aplicată o funcție *hash* cu ajutorul unui algoritm de criptare (MD5, SHA-1, SHA-256 - [15][16][17]), iar în baza de date se va salva rezultatul criptat al acelei parole. În acest fel, atacatorii care reușesc să aibe acces la o anumită bază de date, au șanse extrem de mici de a afla parolele utilizatorilor înregistrați. Sigur, există posibilitatea ca atacatorii să afle parolele utilizatorilor dacă intră în posesia algoritmului care a fost aplicat pentru a crea hash-urile parolelor, dar acest lucru necesită un timp destul de îndelungat pentru a ajunge la un rezultat final promițător.

Pentru a verifica parola introdusă de utilizator cu hash-ul parolei din baza de date este necesar ca platforma pe care utilizatorul se autentifică să calculeze după același algoritm hash-ul parolei introduse pe moment. Apoi hash-ul proaspăt obținut este căutat în baza de date și, dacă este găsită o înregistrare identică, utilizatorul reușește să se autentifice cu succes în aplicația dorită. În Figura 2 este detaliat procesul de autentificare în care parolele sunt stocate în baza de date sub formă criptată.

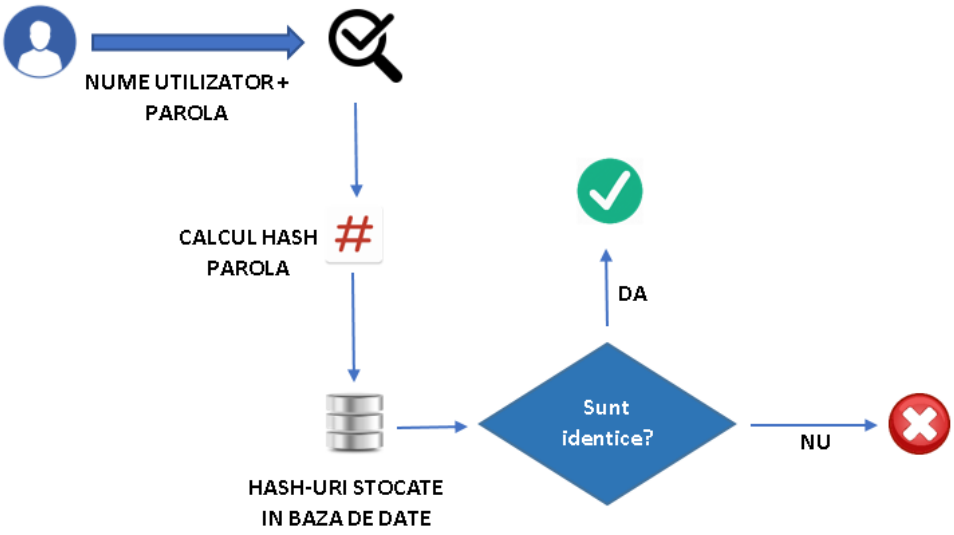

Figura 2. Utilizarea funcțiilor *hash* pentru stocarea și verificarea parolei unui utilizator

Nivelul următor de securitate este asigurat de autentificarea cu doi factori. Acesta oferă o creștere semnificativă a siguranței [13][14] și poate oferi combinația ideală pentru fiecare platformă în parte în funcție de necesități. Cele mai importante elemente ale autentificării prin doi factori sunt: parola, funcțiile hash aplicate pentru o anumită parolă, conceptul de *salting*, puterea parolei și OTP-urile (*One Time Password*).

Conceptul de *salting* [7][10] definește procedeul de adăugare a unui șir de caractere în mod aleator la fiecare parolă stocată în baza de date a unei entități. Șirul de caractere generat aleator este adăugat la parolele în text clar ale utilizatorilor. După crearea noilor șiruri de caractere, funcțiile hash sunt aplicate în mod direct pe acestea [10], rezultând astfel șiruri criptate care vor fi stocate în baza de date. Atunci când utilizatorul își dorește acces la o anumită aplicație, sistemul extrage șirul generat aleator, îl adaugă la parola introdusă de utilizator, aplică funcția hash cu același algoritm folosit la salvarea parolei în baza de date și compară rezultatul obținut cu cel stocat în bază. Dacă cele două șiruri criptate sunt identice, atunci utilizatorul este autentificat cu succes.

Acest concept a fost creat pentru a evita situațiile în care atacatorii sunt siguri de algoritmul de criptare utilizat de entitatea țintă și sunt dispuși să încerce pentru o perioadă mai mare de timp toate combinațiile posibile pentru a afla parolele utilizatorilor acelei companii. Atacatorii nu pot afla șirul generat aleator ce a fost adăugat la fiecare parolă în parte, astfel că, această metodă, reprezintă o barieră suplimentară împotriva atacurilor cibernetice.

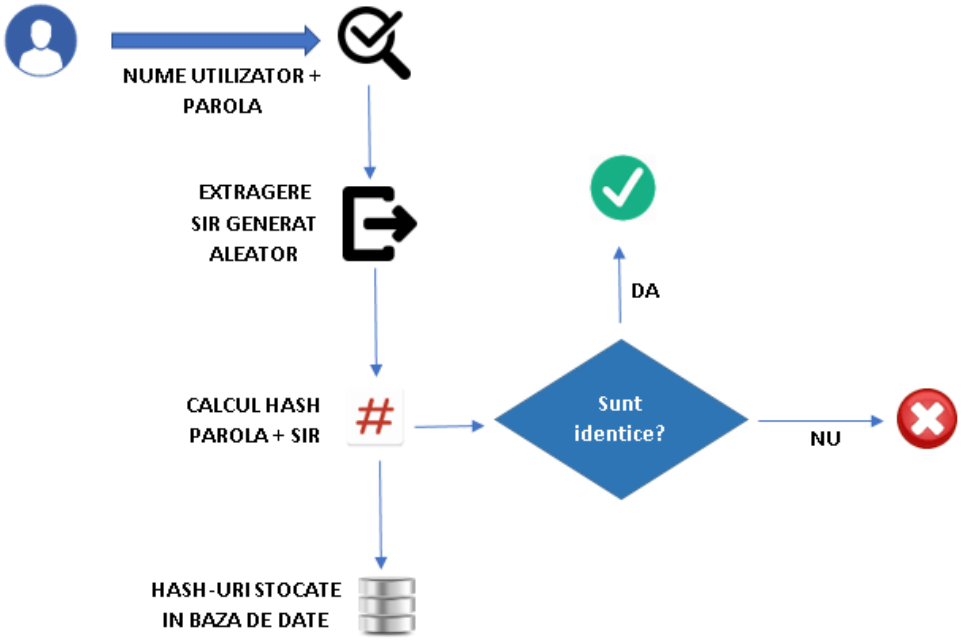

Figura 3. Exemplificarea conceptului de *salting*

Platformele pe care utilizatorii se conectează trebuie să ofere o siguranță foarte mare, mai ales când este vorba de sisteme financiare unde un atac cibernetic ar produce pagube majore [4][9]. De asemenea, este de datoria utilizatorului să aleagă parole puternice, greu de intuit şi care să ridice nivelul de securitate al platformelor respective. Astfel, utilizatorii trebuie să fie informați despre cum arată o parolă puternică pentru a putea ajuta sistemele să funcționeze în mod sigur şi stabil. De cele mai multe ori platformele au implementate validări care nu permit utilizatorilor să seteze parole slabe, indicând ce elemente ar trebui să conțină parolele lor pentru a fi sigure.

Conceptul OTP reprezintă, în fapt, o parolă care este folosită o singură dată. Există mai multe modalități de a aplica acest concept: fie există o listă de parole pe care utilizatorii o au la dispoziție și sunt informați că, în momentul în care folosesc una din parolele de pe listă, la următoarea autentificare acea parolă nu va mai fi disponibilă, fie utilizatorii sunt de acord ca, la un interval de timp stabilit, parolele lor să fie actualizate, fie sunt de acord cu utilizarea unui token ce generează parole aleator care sunt valabile o perioadă limitată de timp.

Există mai multe variante de a actualiza parolele, astfel încât atacatorii să nu poată ajunge la parolele reale ale utilizatorilor: fie sunt refolosite vechile parole drept cheie pentru criptarea noilor parole aplicând funcții hash, fie se are în vedere aplicarea multiplă a unor funcții hash pentru aceeași nouă parolă (daca un atacator identifică algoritmul de criptare utilizat pentru calculul hash-ului unei parole, atunci el va avea primul hash generat, dar nu va ști de câte ori a mai fost aplicat același algoritm pe aceeași parolă pentru a obține hash-ul final stocat în baza de date).

Token-urile securizate au rolul de a genera coduri de acces pentru un utilizator, fiind întotdeauna cel de-al doilea factor al unei autentificări. Utilizarea unui astfel de token prezintă atât avantaje, cât și dezavantaje. El reprezintă elementul pe care utilizatorul il are și, împreună cu elementul pe care utilizatorul îl știe, se creează o barieră sigură împotriva atacurilor hackerilor.

3. Implementare practică

Pentru a implementa o aplicație pe modelul unui token securizat trebuie luate în calcul mai multe aspecte: mecanismele care vor fi implementate, alegerea mediului în care vor avea loc implementarile și în care se vor face testele, analiza altor modele asemănătoare pentru a putea identifica avantajele și dezavantajele unei astfel de aplicații.

Mediul de implementare ales a fost Android Studio 3.0.1 conceput pentru telefoanele mobile. Aplicația a fost creată urmărind modelul token-ului virtual des utilizat în mecanismul de internet banking. Acest token reprezintă o modalitate sigură față de securitatea altor aplicații web, generând coduri unice pentru utilizatorul care încearcă să se autentifice în altă aplicație. Utilizatorul va fi recunoscut pe baza acestor coduri unice.

Ca prim pas, utilizatorul unui telefon cu sistem de operare Android trebuie să își instaleze aplicația pe dispozitivul mobil. În urma instalării, se va efectua procesul de inițializare a token-ului. Se va introduce cheia furnizată de sistem printr-un canal securizat sau alternativ precum și parola cu care se va securiza cheia. O aceeași cheie este folosită aât pentru criptare și pentru decriptare. Prin urmare, sistemul folosit este unul simetric, ambele părți cunoscând atât algoritmul cât și cheia de criptare. Menținerea unei chei în secret este o sarcină importantă pentru stabilirea și menținerea unui canal de comunicare securizat. În această privință există o problemă a transferului inițial al cheii (sincronizarea cheii), soluția aleasă bazâdu-se pe un canal alternativ securizat sau predare personală în plic securizat.

Dacă utilizatorul este deja înregistrat în aplicație, acesta va fi redirecționat în activitatea de logare.

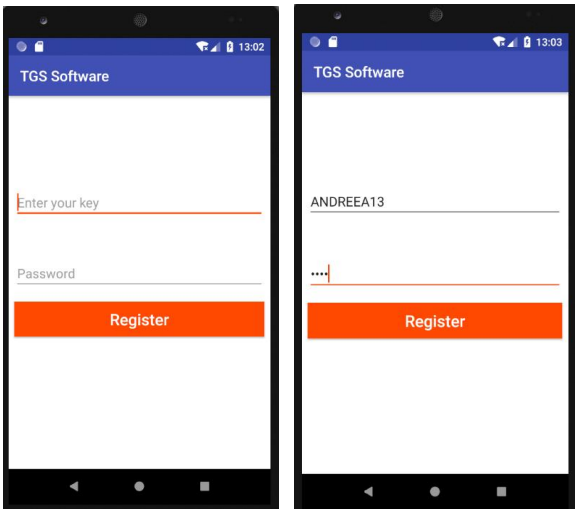

Figura 4. Activitatea de înregistrare a aplicației Android

În cadrul formularului de logare utilizatorul va trebui să își introducă parola (cea setată în momentul înregistrării). Dacă parola este validă, utilizatorul va fi redirecționat în activitatea principală a aplicației unde își poate alege opțiunea dorită. Dacă parola nu este validă și a fost introdusă deja de 3 ori în cadrul aplicației, atunci aceasta va fi blocată, nepermițându-i utilizatorului să se mai logheze. Acesta este unul dintre mecanismele de protecție împotriva fraudelor de identitate și care, din păcate, poate genera neplăceri în cazul în care utilizatorul legitim comite erori repetate la introducerea parolei.

De asemenea, dacă utilizatorul dorește să își reseteze cheia și parola din cauza unor motive personale, acesta poate apăsa opțiunea “Reset your credentials” din activitatea de logare.

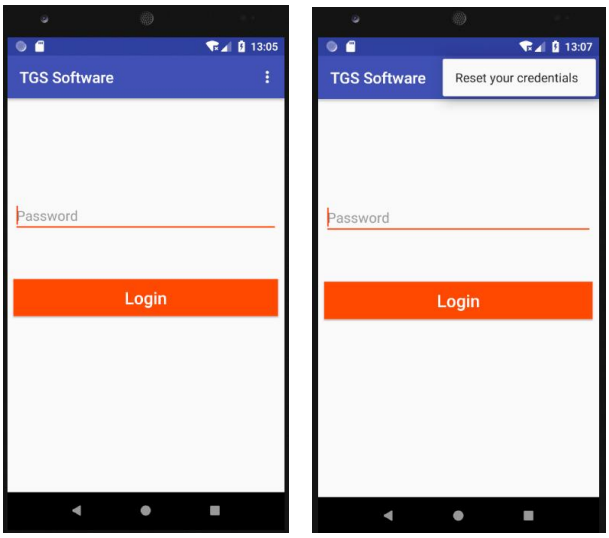

Figura 5. Activitatea de logare a aplicației Android

Cele trei opțiuni posibile pe care un utilizator le poate avea folosind aplicația sunt afișate în activitatea principală a acesteia:

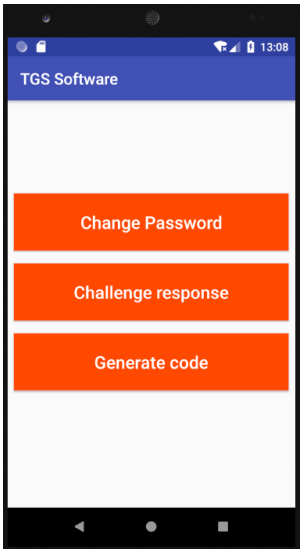

Figura 6. Activitatea principală a aplicației Android cu trei opțiuni

Opțiunea 1: Challenge response

Opțiunea 2: Generate code

Opțiunea 3: Change password

A. Opțiunea 1: Obținerea unui cod pe baza altui cod (Challenge response)

Utilizatorul primește un cod după sau înaintea conectării în cadrul altei aplicații web și trebuie să introducă ca raspuns un alt cod, cod generat de token-ul personal. Astfel, pe dispozitivul Android va introduce codul dat de platforma pe care vrea să se conecteze. Dacă respectivul cod introdus este valid (valid înseamnă un cod care conține numai numere și are o lungime fixă de 6 caractere), utilizatorul va primi un alt cod valid pentru ziua curentă. Acest cod generat este creat prin aplicarea algoritmului MD5 asupra concatenării codului introdus cu cheia utilizatorului și cu data în formatul "dd-mm-yyyy". Codul este valabil pe o durată limitată de timp, respectiv numai în ziua în care a fost generat, și are o lungime de 6 caractere. Acest cod a fost realizat conform mecanismului *salting*.

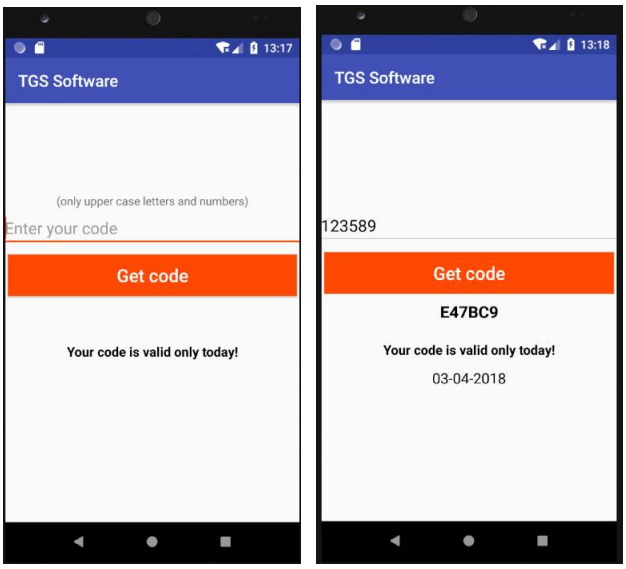

Figura 7. Activitatea Challenge Response a aplicației Android

B. Opțiunea 2: Generează codul (Generate code)

Această opțiune permite utilizatorului generarea unui cod unic care poate fi folosit timp de zece minute. Timpul poate varia în funcție de minutul la care utilizatorul a apăsat butonul de generare a codului în aplicația Android, astfel că acesta va avea la dispoziție, în mod real, mai puțin de zece minute. Aplicația de Android contorizează şi afișează timpul disponibil pentru a introduce codul în cealaltă aplicație web.

Când perioada de timp specificată expiră, cealaltă aplicație nu va mai accepta codul generat și va fi necesară generarea unui nou cod. De această dată, codul care este generat este reprezentat de concatenarea cheii și a datei cu formatul "dd-mm-yyyy mm:ss". Minutele sunt importante în acest caz pentru că utilizatorul are la dispoziție doar 10 minute (chiar mai puțin, așa cum s-a arătat anterior) pentru a introduce codul generat.

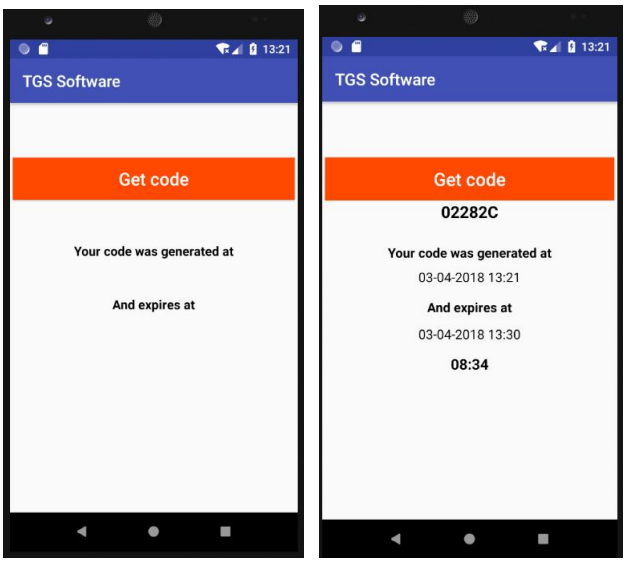

Figura 8. Activitatea de generare cod a aplicației Android

C. Opțiunea 3: Schimbarea parolei (Change Password)

Dacă utilizatorul vrea să își schimbe parola din motive personale, acesta poate folosi activitatea de schimbare a parolei. Utilizatorul va fi nevoit să completeze trei câmpuri: parola curentă, noua parolă și o confirmare a noii parole. Dacă toate acestea sunt valide, parola va fi schimbată.

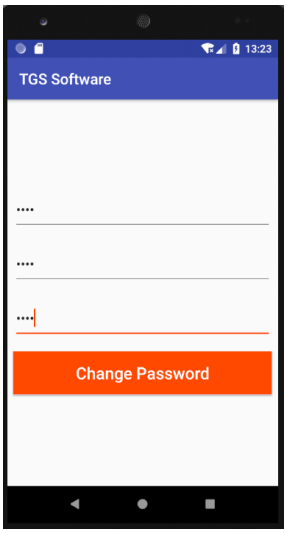

Figura 9. Activitatea de schimbare a parolei aplicației Android

Codul aplicației Android este generat prin aplicarea unei funcții hash (reprezentată de algoritmul MD5) pe un șir constând în concatenarea datei din ziua în care codul s-a cerut o generare cu o coloană din baza de date care reprezintă un anumit cod ales de fiecare utilizator (cheia utilizatorului). MD5 este o funcție criptografică unică și este utilizată des în domeniul semnaturilor electronice. În utima vreme, a fost inlocuit cu algoritmi de criptare mai puternici.

4. Concluzii legate de analiza comportamentală a identitaţii digitale

Autentificarea reprezintă un aspect cheie al atribuirii identității, oferind o asigurare codificată a identității unei entități. Metodologiile de autentificare includ prezentarea unui obiect unic, furnizarea de informații confidențiale, confirmarea dreptului de proprietate asupra unei adrese de e-mail, unui telefon sau certificat digital.

Sistemele de autentificare folosesc, în scenariile de detectare a fraudei, instrumente de învățare a programelor prin care se analizează seturi mari de utilizatori și se raportează încercările de fraudă a autentificărilor (utilizări neautorizate ale identității digitale).

Utilizând informațiile colectate de-a lungul timpului, modelul de date a fost îmbunătățit constant. În contextul dat, evoluția are la bază faptul că oamenii raportează anomaliile legate de diversele conturi pe care le dețin (conturile bancare, conturi de email, conturi de comerț on-line sau activități suspecte ale propriului cont în sistem). Această analiză este utilă și pentru autentificarea utilizatorilor, deoarece nu toți utilizatorii știu să recunoască anumite comportamente anormale și, prin urmare, nu pot raporta încercări neautorizate de a accesa serviciile de afaceri și aplicațiile pe care le utilizează. Evidențierea comportamentului anormal îl face pe utilizator să conștientizeze accesul neautorizat sau tentativa de furt de identitate.

Analiza în dinamica datelor de conectare a dus la concluzia că utilizatorii respectă un anumit tipar datorat următorilor factori:

- factorilor comportamentali: program de lucru, program de somn – conectarea se face într-o anumită perioadă de timp. Reprezintă o anomalie conectarea într-o perioadă nespecifică de timp, conectarea din două locuri în același timp sau o mișcare nenaturală în spațiu;

- factori de proprietate – desktop, laptop, tabletă, telefon mobil. Un comportament normal este că un utilizator foloseste aceleași calculatoare, telefoane, aceleași adrese IP. Reprezintă o anomalie conectarea de la adrese IP aflate teoretic la distanțe considerabile, conectarea dintr-o țară/oraș noi sau neuzuale, folosirea unor sisteme de operare diferite, neuzuale sau noi, folosirea de browsere neuzuale;

- factori corelați - de exemplu, se va conecta seara de acasă și ziua de la locul de muncă.

Deşi aplicaţia practică nu pare a avea o complexitate foarte ridicată, în realitate ea are menirea de a scoate în evidenţă aspecte practice legate de autentificarea cu doi factori. Aceasta reprezintă o formă simplificată a autentificarii cu factori multipli care, cel mai probabil, va deveni o metodă curentă de autentificare în viitorul apropiat. Deja multe platforme online (în special cele din domenii critice, cum ar fi domeniul financiar-bancar) folosesc în mod curent autentificarea cu doi factori și au anunțat migrarea, în viitorul apropiat, la autentificarea cu factori multipli.

Mulţumiri

Studiul a fost finanţat şi suţinut prin proiectele de cercetare COOPID mySMIS-115656 şi ADSELECT mySMIS-115646.

BIBLIOGRAFIE

1. Alkandari, A. A., Al-Shaikhli, I. F., Alahmad , M. A. (2013). Cryptographic Hash Function: A High Level View, *2013 International Conference on Informatics and Creative Multimedia*, Kuala Lumpur, Malaysia, DOI: 10.1109/ICICM.2013.29.

2. Aloul, F., Syed Zahidi, S., Wassim El-Hajj, W. (2009). Two factor authentication using mobile phones, *2009 IEEE/ACS International Conference on Computer Systems and Applications*, Rabat, Morocco, DOI: 10.1109/AICCSA.2009.5069395.

3. Camp, J. L. (2004). Digital identity, *IEEE Technology and Society Magazine*, Vol. 23, Issue: 3, 34 - 41, DOI: 10.1109/MTAS.2004.1337889.

4. Cox, D. (2014). Handbook of Anti-Money Laundering. Publisher: Wiley, United Kingdom, ISBN 978-0470065747.

5. Jacomme, C., Kremer, S. (2018). An Extensive Formal Analysis of Multi-factor Authentication Protocols, *IEEE 31st Computer Security Foundations Symposium (CSF)*, Oxford, UK, DOI: 10.1109/CSF.2018.00008.

6. Kennedy, W., Olmsted, A. (2017). Three factor authentication, *12th International Conference for Internet Technology and Secured Transactions (ICITST)*, Cambridge, UK, DOI: 10.23919/ICITST.2017.8356384.

7. Kharod, S., Sharma, N., Sharma, A. (2015). An improved hashing based password security scheme using salting and differential masking, *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, Noida, India, DOI: 10.1109/ICRITO.2015.7359225.

8. Mouha, N., Raunak, M. S., Kuhn, D. R., Kacker, R. (2018). Finding Bugs in Cryptographic Hash Function Implementations, *IEEE Transactions on Reliability*, Vol. 67, Issue: 3, Sept. 2018, 870-884, DOI: 10.1109/TR.2018.2847247.

9. Mutua, M. (2018). Mastering Anti-Money Laundering: What I have Learned About Dirty Money and What You Need to Know. *CreateSpace Independent Publishing Platform*, ISBN 978-1720466185.

10. Nohe, P. (2018). The difference between Encryption, Hashing and Salting, December 19, 2018, online: https://www.thesslstore.com/blog/difference-encryption-hashing-salting/

11. Ometov, A., Bezzateev, S. (2017). Multi-factor authentication: A survey and challenges in V2X applications, *9th International Congress on Ultra Modern Telecommunications and*

Control Systems and Workshops (ICUMT), Munich, Germany, DOI: 10.1109/ ICUMT.2017.8255200.

12. Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., Koucheryavy, Y. (2018). Multi-Factor Authentication: A Survey, *Cryptography*, 2(1), 1, https://doi.org/10.3390/ cryptography2010001.

13. Petsas, T., Tsirantonakis, G., Athanasopoulos, E., Ioannidis, S. (2015). Two-factor Authentication: Is the World Ready? Quantifying 2FA Adoption, *EuroSec'15,* April 21–24, Bordeaux, France, ACM 978-1-4503-3479-2/15/04, http://dx.doi.org/10.1145/ 2751323.2751327.

14. Zhang , J., Tan, X., Wang, X., Yan, A., Qin, Z. (2018). T2FA: Transparent Two-Factor Authentication, *IEEE Access*, Vol. 6, 32677-32686, 15 June, DOI: 10.1109/ACCESS.2018. 2844548.

15. *** The MD5 Message-Digest Algorithm, online: https://www.ietf.org/rfc/rfc1321.txt

16. *** US Secure Hash Algorithm 1 (SHA1), online: https://tools.ietf.org/html/rfc3174

17. *** Re-Hashed: The Difference Between SHA-1, SHA-2 and SHA-256 Hash Algorithms, online: https://www.thesslstore.com/blog/difference-sha-1-sha-2-sha-256-hash-algorithms

Cornel ȘERBAN a absolvit Facultatea de Cibernetică, Statistică și Informatică Economică în anul 1999 și în prezent urmează Școala Doctorală din cadrul Acadamiei de Studii Economice București. Deține o diplomă în economie și în prezent este directorul TGS Software, o firmă de IT din București. Este implicat în coordonarea activităților din cadrul firmei TGS Software și în coordonarea proiectelor pe care aceasta le implementează. Are experiență atât în domeniul IT, cât și în implementarea și validarea soluțiilor tehnice privind raportări pentru BNR, Oficiul Național de Spălare a Banilor și ANAF.

Cornel SERBAN graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 1999. He holds a diploma in Economics and he is currently PhD student at the Doctoral School of Bucharest University of Economic Studies. Cornel is the General Manager of TGS Software, an IT company located in Bucharest. He is involved in coordinating the activity within TGS Software and coordinating the projects that the company implements. He has experience in IT field and also in validating and implementing technical solutions regarding the BNR, ONPCSB and ANAF reports.

Ştefan MOCANU a absolvit Facultatea de Automatică şi Calculatoare, direcţia Automatică şi Informatică Industrială şi masterul Sisteme cu Arhitectură Deschisă din cadrul aceleiaşi facultăţi. Din 2005 deţine titlul de doctor în urma stagiului de doctorat efectuat tot în cadrul Facultăţii de Automatică şi Calculatoare. În prezent este cadru didactic la Facultatea de Automatică şi Calculatoare unde desfăşoară activităţi didactice şi de cercetare în domeniul IT&C.

Ştefan MOCANU graduated the Faculty of Automatic Control and Computer Science, Department of Automatic Control and Industrial Informatics and the Open Architecture Systems Master within the same faculty. Since 2005 he holds a PhD degree based on a doctoral stage at the Faculty of Automatic Control and Computer Science. Currently he is tenured associate professor inside the Department of Automatic Control and Industrial Informatics where he is in charge with teaching and research activities related to IT&C.

Cosmin POPA a absolvit Facultatea de Automatică și Calculatoare, direcția Automatică și Informatică Industrială și masterul Sisteme cu Arhitecturi deschise din cadrul Universității Politehnica din București. În prezent este doctorand la Școala Doctorală a Facultății de Automatică și Calculatoare, Universitatea Politehnica din București. Are o experiență de aproape 20 de ani în diverse domenii performante IT&C, fiind implicat în proiecte de dezvoltare de software pentru managementul publicității, administrație publică locală sau centrală, telecom, medical sau banking. Domeniile de interes actuale sunt axate pe servicii distribuite și analize complexe de date care implică de asemenea și identitate digitală.

Cosmin POPA graduated the Faculty of Automatic Control and Computer Science, Department of Automatic Control and Industrial Informatics and the Open Architecture Systems Master of the Politehnica University of Bucharest. Currently he is PhD student at the Doctoral School of Faculty of Automatic Control and Computer Science, Politehnica University of Bucharest. He has an experience for almost 20 years in different performant domains in IT&C, being involved in software development projects for the management of advertising, local and central public administration, telecom, medical or banking. His interest domains are focused on distributed services and complex data analysis that implies also digital identity.