Universitatea Națională de Știință și Tehnologie POLITEHNICA București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Metode de autentificare multifactor pentru securitatea rețelelor

Lucrare de dizertație

prezentată ca cerință parțială pentru obținerea titlului de *Master* în domeniul *Inginerie electronică, telecomunicații și tehnologii informaționale*
programul de studii de masterat *Tehnologii multimedia în aplicații de biometrie și securitatea informației (BIOSINF)*

Conducător(i) științific(i):
Ș.L. dr. ing. Șerban MIHALACHE
C.S. II dr. ing. Alexandru CARANICA

Absolvent:
Diana-Roxana BRATU

2026

Universitatea Națională de Știință și Tehnologie POLITEHNICA București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației
Programul de masterat **BIOSINF**

Anexa 2

TEMA LUCRĂRII DE DISERTAȚIE

a masterandului **BRATU F.C. Diana-Roxana, 421-BIOSINF**

1. Titlul temei: Metode de autentificare multifactor pentru securitatea rețelelor

2. Descrierea temei:
Securitatea accesului la resursele critice din rețelele de calculatoare a devenit o prioritate zero, în contextul creșterii exponențiale a atacurilor de tip phishing și a compromiterii datelor de autentificare. În acest context, lucrarea abordează limitele metodelor tradiționale de autentificare și necesitatea implementării unor mecanisme robuste de tip MFA (Multi-Factor Authentication). Se va pune accent pe analiza vulnerabilităților metodelor curente (ex. SMS OTP) și pe tranziția către soluții moderne precum FIDO2 sau autentificarea biometrică.
Proiectul își propune implementarea și evaluarea unui ecosistem MFA, analizând impactul asupra securității și experienței utilizatorului. Se vor explora integrări cu protocoale standard și se vor testa scenarii de atac pentru a valida reziliența soluției propuse. Ulterior, se va implementa o aplicație web care va oferi un ecosistem MFA bazat pe protocoale standard și integrat cu servicii de autentificare existente.
Pentru dezvoltarea proiectului, se vor folosi tehnologii web moderne precum Node.js, React.js, Angular și baze de date open-source de tip SQL (MySQL, PostgreSQL).

3. Contribuția originală:
- Analiza comparativă a tehnologiilor MFA din perspectiva securității și a experienței utilizatorului.
- Arhitectura și implementarea unui sistem de autentificare centralizat care integrează factori multipli de verificare.
- Evaluarea securității soluției prin simularea unor vectori de atac specifici (Man-in-the-Middle, MFA Fatigue/Prompt Bombing).
- Măsurarea performanței și a latenței introduse de mecanismele de securitate adiționale.
- Elaborarea de concluzii detaliate privind performanțele și fezabilitatea soluției propuse.

4. Resurse și bibliografie:
- R. Kantipudi, A.S. Kumar Mallavarapu, S.M. Rajagopal, M. Kagolanu, "A Comprehensive Analysis on using Multifactor Authentication System for Three Level Security," 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2024.
- P. Grassi et al., "NIST Special Publication 800-63B-4: Digital Identity Guidelines - Authentication and Lifecycle Management," National Institute of Standards and Technology, rev. 4, 2024.
- FIDO Alliance, "FIDO2: Web Authentication (WebAuthn) Specification," W3C Recommendation, Level 2, 2021.
- A. Das, J. Bonneau, M. Caesar, N. Borisov, X. Wang, "The Tangled Web of Password Reuse," Network and Distributed System Security Symposium (NDSS), 2014.
- Duo Security, "py_webauthn - A Python implementation of WebAuthn," GitHub repository.

5. Data înregistrării temei: 2026-01-20 00:09:40

Conducător(i) lucrare,
Ș.L. dr. ing. Șerban Matei MIHALACHE

C.Ș.II dr. ing. Alexandru CARANICA

Student,
BRATU F.C. Diana-Roxana

Responsabil program master,
Prof. dr. ing. Dragoș BURILEANU

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **aa4c96497a**

Declarație de onestitate academică

Prin prezenta declar că lucrarea cu titlul *Metode de autentificare multifactor pentru securitatea rețelelor*, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității Naționale de Știință și Tehnologie POLITEHNICA București drept cerință parțială pentru obținerea titlului de *Master* în domeniul *Inginerie electronică, telecomunicații și tehnologii informaționale*, programul de studii de masterat *Tehnologii multimedia în aplicații de biometrie și securitatea informației (BIOSINF)*, este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referire la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referire la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurătorilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 16.06.2026

Absolvent: Diana-Roxana BRATU

(semnătura în original)

CUPRINS

LISTA TABELELOR vii
LISTA FIGURILOR ix
LISTA ABREVIERILOR xi

1. INTRODUCERE 1
1.1. Contextul actual al securității rețelelor informatice 1
1.2. Evoluția Cadrului Legislativ: NIS2 și OUG 155/2024 2
1.3. Motivația alegerii temei 3
1.4. Obiectivele lucrării si contribuțiile propuse 4
1.5. Structura lucrării 4

2. FUNDAMENTELE AUTENTIFICĂRII ÎN REȚELELE DE CALCULATOARE 5
2.1. Metode clasice de autentificare bazate pe parolă 5
2.2. Vulnerabilități ale autentificării tradiționale 7
2.2.1. Spargerea parolelor 7
2.2.2. Atacul de tip phishing 7
2.2.3. Atacuri Man-in-the-Middle 8

3. AUTENTIFICAREA MULTIFACTOR - CONCEPTE ȘI TEHNOLOGII . 11
3.1. Niveluri de identificare 11
3.2. Metode de Autentificare Multifactor utilizate în practică 12
3.2.1. Autentificarea Biometrică 12
3.2.2. Token-uir de Autentificare de tip Hardware 14
3.2.3. Token-uir de Autentificare de tip Software 15
3.2.4. Autentificarea fără parolă 15
3.3. Protocoale de federație: OAuth 2.0 și OpenID Connect (OIDC) 17
3.4. Limitări și riscuri ale sistemelor MFA clasice 18
3.4.1. Vulnerabilitățile sistemelor bazate pe SMS 18
3.4.2. MFA Fatigue 19
3.4.3. Riscurile autentificării biometrice 19

4. STADIUL CURENT AL TEHNOLOGIILOR DE AUTENTIFICARE . . . 21
4.1. Standardele NIST SP 800-63 21
4.1.1. Niveluri de Asigurare a Autentificării 21
4.1.2. Actualizarea Strategiei de Securitate a Parolelor 22
4.1.3. Rezistența la Phishing și Passkeys 23
4.1.4. Influența GenAI asupra Metodelor Biometrice 23

4.1.5. Gestiunea Sesiunilor și Recuperarea Identității 23
4.2. Analiza comparativă a metodelor MFA în literatura de specialitate 24

5. IMPLEMENTAREA PRACTICĂ A SISTEMULUI DE AUTENTIFICARE MULTIFACTOR 27
5.1. Arhitectura Sistemului MFA 27
5.2. Implementarea Componentelor Backend 30
5.2.1. Implementarea TOTP 30
5.2.2. Implementarea FIDO2 31
5.2.3. Implementarea OIDC 31
5.2.4. Gestionarea Identității și a Sesiunii 32
5.2.5. Considerații de secutitate în proiectare 33
5.3. Implementarea Interfeței Grafice 34

6. EVALUAREA SECURITĂȚII ȘI A PERFORMANȚEI SISTEMULUI . . 41
6.1. Simularea vectorilor de atac 41
6.1.1. Metodologia de evaluare 41
6.1.2. Atacul brute force în autentificarea cu parolă și cu TOTP 41
6.1.3. Atac de tip replay pe autentificarea prin codul TOTP 42
6.1.4. Falsificarea domeniului pentru autentificarea FIDO2 43
6.1.5. Interceptarea traficului prin MitM 43
6.2. Evaluarea performanței și a latenței 48
6.3. Impactul asupra experienței utilizatorului 50

7. CONCLUZII ȘI DIRECȚII VIITOARE 53
7.1. Concluziile rezultatelor experimentale și contribuțiile aduse 53
7.2. Limitări și direcții viitoare 54

BIBLIOGRAFIE 57

ANEXA A. ANEXA 1 61

LISTA TABELELOR

Tabelul 4.1 Sinteza cerințelor tehnice pe niveluri de asigurare conform NIST SP 800-63B-4. 22
Tabelul 4.2 Compromisul dintre securitate, utilizabilitate și implementare pentru metodele MFA. 25
Tabelul 4.3 Metodelor de autentificare în raport cu vectorii de atac 26
Tabelul 6.1 Latența principalelor procese de autentificare 49
Tabelul 6.2 Impactul proxy-ului de interceptare asupra latenței 49
Tabelul 6.3 Estimarea timpului suplimentar introdus de MFA 50
Tabelul 7.1 Sinteza rezultatelor evaluării pe vectori de atac 54

LISTA FIGURILOR

Figura 2.1 Atac MitM [11] 8
Figura 3.1 Exemple tipuri de autentificare[12] 11
Figura 3.2 Metode de autentificare biometrică [14] 13
Figura 3.3 Procesul de înregistrare al unui dispozitiv autentificator FIDO2/WebAuthn utilizând metoda „credentials.create”[18] 16
Figura 3.4 Procesul de autentificare al clientului FIDO2/WebAuthn utilizând metoda „credentials.get”[18] 17
Figura 5.1 Diagrama fluxului de autentificare al sistemului 28
Figura 5.2 Diagrama bazei de date 29
Figura 5.3 Pagina de autentificare 34
Figura 5.4 Pagina de înregistrare 34
Figura 5.5 Pagina principală 35
Figura 5.6 Configurarea autentificării cu TOTP 36
Figura 5.7 Pașii configurării aplicației Google Authenticator și codul QR generat . 36
Figura 5.8 Pagina principală după adăugarea TOTP 36
Figura 5.9 Configurarea autentificării cu FIDO2 37
Figura 5.10 Alegerea modalității de salvare a cheii de acces 37
Figura 5.11 Generarea codului QR pentru transmiterea cheii de acces 37
Figura 5.12 Pagina principală după configurarea metodelor de autentificare 38
Figura 5.13 Autentificare prin TOTP 38
Figura 5.14 Autentificare prin FIDO2 38
Figura 6.1 Interceptarea credențialelor 45
Figura 6.2 Interceptarea codului TOTP 45
Figura 6.3 Interceptarea datelor pentru FIDO2 46
Figura 6.4 Challenge-ul FIDO2 transmis de server 46
Figura 6.5 Datele WebAuthn interceptate de proxy 47
Figura 6.6 Interceptarea token-urilor 47

LISTA ABREVIERILOR

2FA	*Two-Factor Authentication* (lb. en.; autentificare în doi factori)
AAA	*Authentication, Authorization and Accounting* (lb. en.; autentificare, autorizare și auditare)
AAL	*Authentication Assurance Level* (lb. en.; nivel de asigurare a autentificării)
AitM	*Adversary-in-the-Middle*
API	*Application Programming Interface*
ARP	*Address Resolution Protocol*
BitM	*Browser-in-the-Middle*
BLE	*Bluetooth Low Energy*
BYOD	*Bring Your Own Device*
CER	*Crossover Error Rate* (lb. en.; rata de eroare egală)
CIA	*Confidentiality, Integrity, Availability* (lb. en.; confidențialitate, integritate, disponibilitate)
CORS	*Cross-Origin Resource Sharing*
CSRF	*Cross-Site Request Forgery*
CTAP	*Client-to-Authenticator Protocol*
CVE	*Common Vulnerabilities and Exposures*
DBSC	*Device Bound Session Credentials*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DNSC	Directoratul Național de Securitate Cibernetică
FAR	*False Acceptance Rate* (lb. en.; rata de acceptare falsă)
FER	*Failure to Enroll Rate* (lb. en.; rata de esec a înregistrării)
FIDO	*Fast IDentity Online* (standard de autentificare)
FRR	*False Rejection Rate* (lb. en.; rata de respingere falsă)
GenAI	*Generative Artificial Intelligence* (lb. en.; Inteligența Artificială Generativă)
HMAC	*Hash-based Message Authentication Code*
HOTP	*HMAC-based One-Time Password*
HSM	*Hardware Security Module*
HTTP	*HyperText Transfer Protocol*
HTTPS	*HyperText Transfer Protocol Secure*
IA	Inteligență artificială
IAM	*Identity and Access Management* (lb. en.; gestionarea identității și a accesului)
ID	*Identifier*
IP	*Internet Protocol*
JSON	*JavaScript Object Notation*
JSONB	*JSON Binary*
JSX	*JavaScript XML*
JWT	*JSON Web Token*
LAN	*Local Area Network*
MAC	*Media Access Control*

MFA	*Multi-Factor Authentication* (lb. en.; autentificare multifactor)
MIME	*Multipurpose Internet Mail Extensions*
MitM	*Man-in-the-Middle*
NFC	*Near Field Communication*
NIS2	*Network and Information Security Directive 2*
NIST	*National Institute of Standards and Technology*
OAuth	*Open Authorization* (protocol de autorizare)
OIDC	*OpenID Connect* (protocol de autentificare)
OTP	*One-Time Password*
OUG	Ordonanța de Urgență a Guvernului
PAD	*Presentation Attack Detection* (lb. en.; detectarea atacurilor de prezentare)
PAM	*Pluggable Authentication Modules*
PIN	*Personal Identification Number*
PKCE	*Proof Key for Code Exchange*
PKI	*Public Key Infrastructure* (lb. en.; infrastructură de chei publice)
QR	*Quick Response*
RBAC	*Role-Based Access Control*
RDP	*Remote Desktop Protocol*
REST	*Representational State Transfer*
RFC	*Request for Comments*
RP	*Relying Party*
SE	*Secure Element*
SIM	*Subscriber Identity Module*
SMS	*Short Message Service*
SQL	*Structured Query Language* (limbaj de programare)
SS7	*Signaling System No. 7*
SSL	*Secure Sockets Layer*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
TOTP	*Time-based One-Time Password*
TPM	*Trusted Platform Module*
U2F	*Universal 2nd Factor*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
USB	*Universal Serial Bus*
VLAN	*Virtual Local Area Network*
W3C	*World Wide Web Consortium*
WebAuthn	*Web Authentication*
XSS	*Cross-Site Scripting*

1. INTRODUCERE

Securitatea sistemelor informatice și a rețelelor de comunicații a traversat o perioadă de schimbări majore, evoluând odată cu tehnologia și cu creșterea amenințărilor cibernetice, de la protecția perimetrului fizic la arhitectura centrată pe identitate. În prezent, identitatea digitală este considerată elementul central al perimetrului de securitate, perspectivă impusă de evoluția serviciilor de tip cloud, dar și de adoptarea politicilor de lucru la distanță și de interconectivitatea globală. Astfel, bariera principală împotriva accesului neautorizat la resurse critice este procesul de autentificare. Cu toate acestea, compromiterea credențialelor reprezintă cel mai des întâlnit vector de atac utilizat, fapt datorat vulnerabilităților metodelor tradiționale de autentificare bazate pe parole.

Lucrarea de față abordează limitele mecanismelor actuale de autentificare și analizează necesitatea de a implementa soluții de tipul autentificării multifactor (MFA). Totodată, cercetarea se concentrează asupra vulnerabilităților metodelor curente și pe necesitatea tranziției catre soluții moderne, precum FIDO2, care sunt rezistente în fața atacurilor de tip phishing, Man-in-the-Middle (MitM) și a MFA Fatigue.

De asemenea, proiectul își propune implementarea și evaluarea unui ecosistem de autentificare multifactor, analizând impactul asupra securității și experienței utilizatorului. Se vor explora integrări cu protocoale standard (OIDC - OpenID Connect) și se vor testa scenarii de atac pentru a valida reziliența soluției propuse. Ulterior, se va implementa o aplicație web care va oferi un ecosistem de autentificare multifactor bazat pe protocoale standard și integrat cu servicii de autentificare existente.

Pentru dezvoltarea proiectului, se vor folosi tehnologii web moderne precum Node.js, React.js și baze de date de tip SQL (PostgreSQL) open source.

1.1. Contextul actual al securității rețelelor informatice

Raportul Verizon Data Breach Investigations Report 2025 [1] evidențiază faptul că numărul breșelor de date a atins un maxim istoric, fiind o consecință a exploatării vulnerabilităților și abuzul credențialelor. La nivel global, securitatea cibernetică nu ține doar de configurarea firewall-urilor, aceasta reprezintă un efort continuu de protejare a identităților într-un mediu dominat de criminalitate informatică industrializată și operațiuni de spionaj cibernetic.

Analiza datelor recente arată faptul că atacatorii vizează în special dispozitivele de la marginea rețelei (edge devices) și infrastructurile critice. Exploatarea vulnerabilităților ca vector de acces inițial a crescut cu 34% față de anul precedent, ajungând să reprezinte 20% din totalul breșelor confirmate. Această tendință este alimentată de viteza cu care vulnerabilitățile de tip zero-day sunt utilizate de grupările infracționale, timpul de exploatare în masă fiind de multe ori egal cu momentul publicării codului CVE [2].

În acest context, credențialele rămân cele mai des vizate în cazul unui atac. Aproximativ 22% din breșe sunt cauzate direct de abuzul de credențiale furate, iar în cazul atacurilor asupra

aplicațiilor web, această cifră ajunge la 88%[1][2]. Un fenomen nou observat este creșterea utilizării programelor de tip infostealer, care reușesc să extragă date de autentificare nu doar de pe dispozitive gestionate, dar și de pe terminale personale utilizate în regim BYOD (Bring Your Own Device)[3].

Inteligența Artificială Generativă (GenAI) a crescut complexitatea amenințărilor informatice. Atacatorii folosesc modele IA pentru a crea campanii de phishing mai convingătoare, textul generat sintetic dublându-se în ultimii doi ani[4]. Astfel, detectarea tentativelor de inginerie socială devine imposibilă pentru utilizatorul obișnuit, deoarece indiciile precum greșelile gramaticale sau tonul nefiresc, au fost eliminate.

În România, situația securității cibernetice reflectă trendurile europene, dar cu anumite particularități determinate de ritmul transformării digitale din sectorul public și privat. Directoratul Național de Securitate Cibernetică (DNSC) a clasificat nivelul amenințării pentru anul 2024 ca fiind unul ridicat, semnalând o intensificare a atacurilor asupra infrastructurii naționale civile.

Anul 2024 a fost marcat de două incidente majore care au testat reziliența sistemelor naționale:

a) Atacul asupra Camerei Deputaților, în care o breșă semnificativă a dus la exfiltrarea a aproximativ 300 MB de date, conținând 316 fișiere, printre care și documente scanate ale actelor de identitate ale parlamentarilor. Atacul a fost posibil prin utilizarea unui dispozitiv personal conectat la rețeaua instituției, subliniind deficiențele în controlul accesului și autentificarea punctelor terminale. Ca răspuns, s-au impus restricții severe de acces la internet și s-a eliminat posibilitatea utilizării conexiunilor neprotejate către intranet.
b) Atacul asupra Sistemului Hipocrate, de tip ransomware, a vizat serverele companiei Romanian Soft Company, blocând activitatea în 26 de spitale. Malware-ul utilizat, „Backmydata", din familia Phobos, s-a propagat prin conexiuni RDP (Remote Desktop Protocol) vulnerabile, criptând bazele de date și paralizând fluxurile medicale. Acest incident a demonstrat necesitatea a unui sistem național de alertare și a implementării autentificării multifactor pe toate punctele de intrare în rețelele de sănătate.

Statisticile furnizate de DNSC pentru anul 2024 indică o creștere cu 40,2% a fraudelor informatice și o majorare de 286,8% a incidentelor de tip malware. Aceste cifre confirmă faptul că România a devenit o țintă, nu doar pentru infractori financiari, dar și pentru grupări care urmăresc perturbarea serviciilor esențiale.

1.2. Evoluția Cadrului Legislativ: NIS2 și OUG 155/2024

Pentru a contracara aceste amenințări, Uniunea Europeană a introdus Directiva NIS2, menită să asigure un nivel ridicat de securitate cibernetică în toate statele membre. În România, transpunerea acestei directive s-a realizat prin Ordonanța de Urgență a Guvernului (OUG) nr. 155/2024, care stabilește un cadru legal riguros pentru entitățile esențiale.

Reglementările aduse de OUG 155/2024 transformă modul în care organizațiile trebuie să gestioneze securitatea rețelelor. Reglementările sunt următoarele:

- Membrii conducerii sunt direct responsabili pentru aprobarea măsurilor de securitate și pot fi sancționați în cazul nerespectării obligațiilor.
- Este înființat Corpul Auditorilor de Sisteme Informatice din România, responsabil cu monitorizarea continuă a conformității entităților cu standardele de securitate.

– Entitățile sunt obligate să raporteze incidentele semnificative către DNSC într-un interval de 24 de ore pentru alerta inițială și 72 de ore pentru raportul detaliat.
– Organizațiile trebuie să implementeze politici de securitate care să includă analiza riscurilor, managementul incidentelor și utilizarea soluțiilor de criptografie și autentificare multifactor robustă

Acest context legislativ pune o presiune constructivă asupra organizațiilor din România pentru a trece de la metodele de autentificare doar aparent conforme la soluții care oferă protecție reală în fața atacurilor moderne.

1.3. Motivația alegerii temei

Motivația alegerii temei „Metode de autentificare multifactor pentru securitatea rețelelor" rezultă din observarea nevoii unor sisteme de autentificare care să se ridice la nivelul noilor amenințări, prin utilizarea unor metode moderne, în detrimentul sau în completarea metodelor tradiționale. Parolele, deși există obligativitatea complexității și a schimbării periodice, s-au dovedit a fi imposibil de securizat la scară largă, fiind predispuse furtului prin *phishing*, *brute-force* sau prin intermediul programelor *malware* de tip *stealer*.

Un alt factorii în alegerea temei este vulnerabilitatea critică a metodelor MFA bazate pe SMS sau coduri OTP (One-Time Password). Deși reprezintă un progres față de parole, aceste metode se bazează în continuare pe secrete partajate, care pot fi interceptate sau manipulate, prin SIM Swapping sau Man-in-the-Middle.

O altă motivație este necesitatea protejării utilizatorii de propriile greșeli induse prin presiune psihologică, prin atacurile de tip MFA Fatigue sau „Prompt Bombing". Aceste atacuri se bazează pe transmiterea continuă a notificărilor de tip push către dispozitivul utilizatorului până când acesta, din oboseală sau neatenție, apasă butonul „Aprobă".

Motivația tehnică a lucrării este susținută de superioritatea demonstrată a standardelor FIDO2. Spre deosebire de metodele tradiționale, FIDO2 utilizează criptografia asimetrică, unde cheia privată nu părăsește niciodată dispozitivul hardware al utilizatorului. WebAuthn oferă rezistență la phishing prin crearea unei legături criptografice între domeniul web și răspunsul de autentificare, astfel încât dispozitivul refuză semnarea cererilor provenite de la site-uri frauduloase. În același timp, securitatea este dată și de faptul că serverul stochează doar cheia publică, astfel încât o eventuală breșă de date nu compromite conturile utilizatorilor, spre deosebire de sistemele bazate pe parole sau OTP.

Un ultim argument este reprezentat de impactul economic si operațional. O breșă de securitate poate avea consecințe financiare și reputaționale semnificative asupra organizațiilor afectate, incluzând pierderi de capital, costuri ridicate pentru investigarea incidentului și remedierea vulnerabilităților, amenzi de conformitate impuse de autorități, întreruperi operaționale și diminuarea încrederii clienților și a partenerilor de afaceri. În multe cazuri, aceste efecte pot conduce la pierderea avantajului competitiv și la deteriorarea pe termen lung a imaginii organizației.

1.4. Obiectivele lucrării si contribuțiile propuse

Obiectivul lucrării este dezvoltarea și evaluarea unui sistem de autentificare multifactor, care depășește limitele metodelor tradiționale, fiind rezilient în fața vectorilor de atac moderni.

Sistemul implementat îndeplinește cele trei condiții date de triada CIA. Confidențialitatea este asigurată de stocarea criptată a parolelor și de păstrarea cheilor private FIDO sau a codului TOTP exclusiv pe dispozitivul utilizatorului. Integritatea sesiunii este verificată pentru fiecare cerere prin JWT. Disponibilitatea este rezultatul prevenirii atacurilor de tip DDoS sau MitM, prin limitarea încercărilor de autetificare pentru o perioadă de timp.

Contribuția principală adusă este evaluarea rezistenței metodelor MFA implementate prin simularea a 5 vectori de atac des utilizați asupra sistemelor de autentificare. Rezultatul demonstrează diferența dintre cele două metode implementate. FIDO2 este rezsitent la phishing, prin legarea criptografică a cheii de domeniu, în timp ce codul TOTP, deși are o fereastră temporară de valabilitate, poate fi reutilizat și necesită protecție suplimentară, prin limitarea accesulului.

De asemenea, o altă contribuție este dezvoltarea unui cadru de evaluare a rezilienței MFA, care corelează fiecare tip de atac cu mecanismul de protecție corepsunzător și cu rezultatul observabil al sistemului, precum codurile HTTP. Cadrul construit poate fi reutilizat pentru evaluarea oricărui sistem care combină factori de atentificare bazați pe cunoaștere și posesie.

Un rezultat dorit este și validarea recomandărilor NIST SP 800-63B-4 privind tranziția de la mecanisme bazate pe secrete partajate către autentificatori rezistenți la phishing.

Sistemul propus este dezvoltat cu ajutorul tehnologiilor menționate anterior, respectiv React.js, Node.js și PostgreSQL, și reprezintă infrastructura experimentală pentru evaluarea și validarea ipotezelor cu privire la rezilienta metodelor MFA.

1.5. Structura lucrării

Lucrarea este alcătuită din șapte capitole.

- **Capitolul 1** introduce contextul securității rețelelor, cadrul legislativ (NIS2 și OUG 155/2024), motivația temei, obiectivele și contribuțiile propuse.
- **Capitolul 2** prezintă fundamentele autentificării, metodele clasice bazate pe parolă și vulnerabilitățile lor, precum spargerea parolelor, phishing și atacurile Man-in-the-Middle.
- **Capitolul 3** definește autentificarea multifactor, tipurile de factori, metodele folosite în practică (biometrie, token-uri hardware și software, autentificare fără parolă) și limitările sistemelor MFA clasice.
- **Capitolul 4** sintetizează stadiul actual, respectiv standardele NIST SP 800-63, analiza comparativă a metodelor MFA din literatură și protocoalele de federație OAuth 2.0 și OpenID Connect.
- **Capitolul 5** descrie implementarea practică și este alcătuit din arhitectura client-server, componentele backend (TOTP, FIDO2, OIDC, gestionarea sesiunii, măsuri de securitate) și interfața grafică.
- **Capitolul 6** conține evaluarea securității prin scenarii de atac automatizate, analiza rezilienței, evaluarea performanței și a latenței, impactul asupra experienței utilizatorului și limitările metodologice.
- **Capitolul 7** formulează concluziile și propune direcții viitoare de cercetare.

2. FUNDAMENTELE AUTENTIFICĂRII ÎN REȚELELE DE CALCULATOARE

Proiectarea unui sistem de autentificare presupune alegerea tehnologiilor necesare pentru verificarea identității unui utilizator prin compararea datelor de autentificare prezentate cu cele stocate pe un server de autentificare. În cazul unei potriviri reușite, contul este autentificat. Un sistem robust de autentificare trebuie să îndeplinească cerințele de securitate legate de confidențialitate, integritate și disponibilitate (triada CIA):

- **Confidențialitatea:** protejarea datelor de autentificare ale contului împotriva scurgerilor de informații, pentru a preveni impersonarea neautorizată;
- **Integritatea:** asigurarea faptului că metoda de autentificare este de încredere și rezistentă la falsificare sau la încercările de ocolire a sistemului;
- **Disponibilitatea:** facilitarea unei autentificări rapide și ușor de utilizat, fără a împiedica fluxurile de lucru[5].

Controlul accesului reprezintă procesul care reglementează modul în care subiecții (un utilizator sau proces, entitatea activă) interacționează cu obiectele (resurse specifice, precum fișiere, dispozitive hardware), prin atribuirea unor drepturi și permisiuni specifice, respectând triada CIA[6].

Contolul accesului este implementat prin sisteme de Gestionare a Identității și a Accesului (IAM). Componentele de bază sunt:

- Identitatea utilizatorului - etapa inițială în care este creat contul sau un identificator unic (ID) care să reprezinte utilizatorul, dispozitivul sau procesul în cadrul rețelei;
- Autentificarea - reprezintă validarea identității pretinse de un subiect și utilizează diferiți factori de autentificare, precum parole sau certificate digitale;
- Autorizarea - determină privilegiile si drepturile de acces pe care un subiect le deține asupra un obiect. Modul de acordare al drepturilor depinde de modelul ales, de exemplu în funcție de rolul utilizatoruli (RBAC - Role Based Access Control);
- Monitorizarea si Auditarea - presupune trasabilitatea acțiunilor efectuate de subiecți asupra obiectelor, pentru a putea identifica activitățile ilegitime[5].

Acest capitol analizează metodele clasice de autentificare bazate pe parolă și vulnerabilitățile acestora. Obiectivul este de a demonstra limitările sistemelor tradiționale cu un singur factor de autentificare și de a motiva importanța autentificării multifactor în contextul amenințărilor actuale.

2.1. Metode clasice de autentificare bazate pe parolă

Parolele reprezintă cea mai răspândită metoda de autentificare și de determinare a drepturilor de access. o parolă este definită ca fiind un șir de caractere utilizat pentru a permite accesul la sisteme informatice, aplicații software și alte sisteme securizate. Acestea au rolul de a preveni

accesul neautorizat, solicinând informații specifice persoanelor care doresc accesul. O parolă poate fi alcătuită din o combinație de cifre, litere și simboluri.

Eficiența parolelor depinde de complexitatea acestora, astfel principala problemă este tendința utilizatorilor de a alege parole ușor de reținut, în detrimentul complexității, Această abordare generează vectori de atac predictibili, cele mai frecvente greșeli, care expun conturile la riscuri de securitate, sunt:

- Folosirea informațiilor personale - utilizarea numelui propriu, a datei de naștere sau a numelor persoanelor apropiate, care pot fi aflate cu ușurință de către atacatori;
- Utilizarea secvențelor simple - alegerea unor șiruri logice de cifre sau litere (precum „1234" ori „abcd") sau a unor combinații alăturate de taste („qwerty"), care sunt primele testate în atacuri de tip dictionary sau brute-force;
- Cuvinte din dicționar - folosirea unor cuvinte comune, care pot fi identificate prin atacuri automatizate care rulează baze de date lexicale;
- Substituția previzibilă a literelor cu simboluri - înlocuirea literelor cu simboluri asemănătoare vizual, precum „P@ssw0rd". Aceste variații par sigure, dar deoarece această metodă este foarte cunoscută, face parte din algoritmii standard de atac.

Pentru a crește nivelul de securitate, utilizatorii trebuie să opteze pentru parole „puternice". O parolă este considerată sigură atunci când are o lungime suficientă și combină cel puțin două din următoarele tipuri de caractere:

- Litere mici și mari (a-z si A-Z);
- Cifre (0-9);
- Caractere speciale și simboluri.

Cu cât parola conține mai multe tipuri de caractere și este mai lungă, cu atât devine mai greu de descifrat de către persoane neautorizate[7].

Gestionarea defectuasă a credențialelor reprezintă unul dintre cei mai utilizați vectori în atacurile cibernetice asupra rețelelor. Pentru a crește nivelul securității sistemelor informatice, este esențială implementarea unor politici stricte, care să reglementeze modul de creare și de schimbare a parolelor, însoțite de instruirea corespunzătoare a utilizatorilor.

O politică de parole eficientă reprezintă baza oricărui plan de securitate, însă provocarea principală constă în a gasi o balanță între complexitatea si ușurința memorării unei parole. În consecință, NIST (The National Institute of Standards and Technology) recomandă în cel mai recent ghid publicat [8] o parolă dificil de descifrat, dar care să fie ușor de reținut, astfel încât utilizatorii să nu recurgă la practici riscante, precum notarea pe suport de harti sau într-un format electronic nesigur.

O astfel de politică reprezintă prima linie de apărare împotriva atacatorilor care vizează rețele vulnerabile, unde accesul poate fi obținut cu un efort minim datorat neglijenței de gestionare a parolelor[7].

O primă configurație tehnică prezentă în politici vizează **lungimea parolei**, prin stabilirea unui număr minim de caractere, dar și complexitatea acesteia. În plus, securitatea este menținută prin gestionarea ciclului de viață al parolei. Reglementarea privind vechimea parolei, sau „password age", obligă utilizatorul să își schimbe parola după un număr prestabilit de zile. Totodată, cerințele de istoric și reutilizare împiedică folosirea unor parole utilizate anterior[5].

Fenomenul de reutilizare a parolelor nu se limitează la sistemul intern, ci include și folosirea aceleiași parole pe diferite site-uri, o practică ce nu poate fi controlată tehnic, doar prin politici administrative și educarea utilizatorilor.

Este important de menționant ca procesul nu se oprește la verificarea credențialelor. După ce sistemul confirmă identitatea utilizatorului, urmează etapa de autorizare, conform AAA.

2.2. Vulnerabilități ale autentificării tradiționale

2.2.1. Spargerea parolelor

Cea mai eficientă cale de a accesa un sistem informatic este prin utilizarea unor credențiale valide, obținute in urma unor metode de spargere a parolelor. Conform [7], principalele astfel de atacuri sunt:

- Atacul de tip dicționar - presupune utilizarea unui fișier predefinit, care conține o listă extinsă de cuvinte, termeni din dicționar sau parole compromise anterior. Instrumentul de atac testează fiecare cuvânt din listă până când găsește parola corectă;
- Atacul de tip Brute Force - implică testarea sistematică a tuturor combinațiilor posibile de caractere până la identificarea parolei corecte. Această metodă necesită timp și resurse suficiente de calcul. Succesul unui astfel de atac depinde direct de lungimea și complexitatea parolei.
- Atacul prin tabele Rainbow (Rainbow Table Attack) - necesită pre-calcularea valorilor de tip hash pentru toate combinațiile posibile dintr-un set de caractere și stocarea lor în tabele optimizate. Ulterior, atacatorul folosește algoritmul Rainbow pentru a găsi corespondența între aceste tabele. Această metodă este rapidă, dar necesită un volum mare de spațiu de stocare.

O modalitate de a combate atacurile de tip tabele Rainbow este tehnica de „salting", care presupune adăugarea unor date aleatorii unice la parola fiecărui utilizator înaintea procesului de hashing, obținând astfel amprente digitale diferite. Aceasta metodă este utilă și în cazul atacurilor de tip brute force, nu oprește acest tip de atac, dar îl încetinește, necesitând mai mult timp și resurse de calcul[9].

În auditul de securitate și analiză juridică, se utilizează aplicații destinate testării rezistenței parolelor, precum John the Ripper, pentu atacuri de tip dicționar sau brute force, sau Cain&Abel, pentru interceptarea traficului și utilizarea tabelelor Rainbow. Existența acestor soluții software capabile de a compromite cu ușurință credențiale subliniază importanța politicilor stricte de securitate și a utilizării autentificării multifactor[7].

2.2.2. Atacul de tip phishing

În prezent, phishing-ul a devenit una dintre cele mai răspândite și periculoase modalități de criminalitate cibernetică. Definit ca o formă de „înșelăciune electronică", phishing-ul combină tehnici de inginerie socială cu metode de *spoofing*, având scopul de a manipula victimele în a dezvălui informații confidențiale, impersonând entități de încredere[10].

Atacul utilizează, în majoritatea cazurilor, e-mailul drept vector de atac principal, încercând să convingă utilizatorul să acceseze link-uri malițioase. O abordare frecventă implică utilizarea unor site-uri web clonate, care imită instituții bancare sau platforme de e-commerce[10]. Atacatorii trimit atenționări false sau solicitări urgente de actualizare a contului. Odată ce utilizatorul introduce datele de autentificare pe site-ul clonă, credențialele sunt preluate de atacator. Acest tip de atac duce la pierderi financiare substanțiale, furt de identitate și breșe severe de securitate.

Phishing-ul este într-o continuă evoluție, deoarece atacatorii utilizează în prezent Inteligența Artificială (IA) pentru a genera mesaje realiste, ceea ce reduce drastic eficiența sistemelor de detecție bazate pe șabloane fixe[10]. Identificarea originii reale a unui e-mail se poate face prin analiza antetelor MIME (Multipurpose Internet Mail Extentions), dar sunt necesare cunoștințe

tehnice, pe care majoritatea persoanelor nu le posedă. Astfel, este necesară implementarea unor politici riguroase și educarea continuă a utilizatorilor cu privire la tipurile de date care nu sunt niciodată solicitate prin e-mail, precum parolele[7].

Asadar, securitatea în prezent îmbină soluțiile tehnice cu creșterea vigilenței utilizatorilor.

2.2.3. Atacuri Man-in-the-Middle

Atacul de tip Man-in-the-Middle, redenumit în prezent atac de tip On-Path, reprezintă o categorie importantă de amenințări în domeniul securității informației, caracterizată prin interceptarea și, în anumite cazuri, alterarea comunicațiilor dintre două entități. Acest tip de atac compromite confidențialitatea, integritatea și autenticitatea datelor transmise, fiind o consecință a comunicațiilor nesecurizate sau slab protejate. Figura 2.1 ilustrează modalitatea de atac.

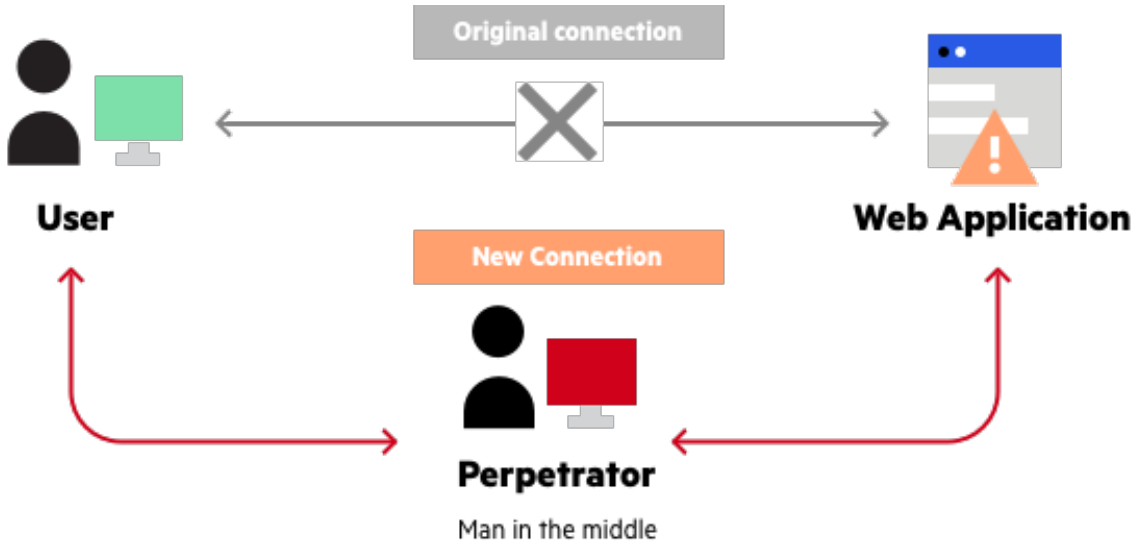

Figura 2.1: Atac MitM [11]

Scenariul de atac al MitM presupune ca atacatorul să capteze traficul de date dintre doua părți legitime, precum un client și un server, fără a fi detectat. Acest proces implică două etape principale: interceptarea și, daca este posibl, decriptarea. Interceptarea poate fi realizată prin tehnici precum compromiterea rețelelor Wi-Fi publice sau prin manipularea protocoalelor de rutare, în timp ce decriptarea depinde de vulnerabilitățile mecanismenlor criptografice utilizate.

Atacurile MitM exploatează lipsa autentificării mutuale și implementarea defectuasă a protocoalelor de securitate.

Cele mai des întâlnite tipologii de atacuri sunt[11]:

- ARP Spoofing – adresa MAC a atacatorului este asociată în mod fraudulos cu adresa IP a unui ultizator legitim dintr-o rețea locală, prin transmiterea de mesaje ARP falsificate.
- DNS Spoofing – implică compromiterea unui server DNS prin modificarea înregistrărilor corespunzătoare adreselor unor site-uri web. Astfel, utilizatorii care încearcă să acceseze site-ul legitim sunt redirectionați către un site controlat de atacator.
- SSL Hijacking – atacator introduce chei de autentificare falsificate în timpul stabilirii conexiunii TCP, creând impresia unei conexiuni securizate
- Session Hijacking – deturnarea unei sesiuni autentificate prin interceptarea identificatorilor de sesiune.

Aceste tehnici pot fi folosite împreună pentru a crește eficiența atacului și pentru a reduce probilitatea de detecție. Astfel, adoptarea unor mecanisme avansate de securitate și creșterea nivelului de conștientizare rămân esențiale pentru protejarea comunicațiilor digitale.

3. AUTENTIFICAREA MULTIFACTOR - CONCEPTE ȘI TEHNOLOGII

Sistemele de autentificare bazate exclusiv pe parole sau pe un singur factor de tip cunoaștere (knowledge factor sau **Something You Know**) sunt considerate vulnerabile în contextul actual al securității cibernetice. Caracterul confidențial al parolelor este expus compromiterii, ceea ce le face insuficiente pentru garantarea integrității accesului. Pentru a mitiga aceste riscuri, se impune utilizarea altor tipuri de factori de autentificare care să suplimenteze sau să înlocuiască mecanismele tradiționale. Tehnologia de Autentificare Multifactor (MFA) integrează utilizarea a cel puțin doi factori de tip diferit pentru a consolida procesul de verificare a identității[5]. Nivelurile de identificare sunt ilustrate in Figura 3.1.

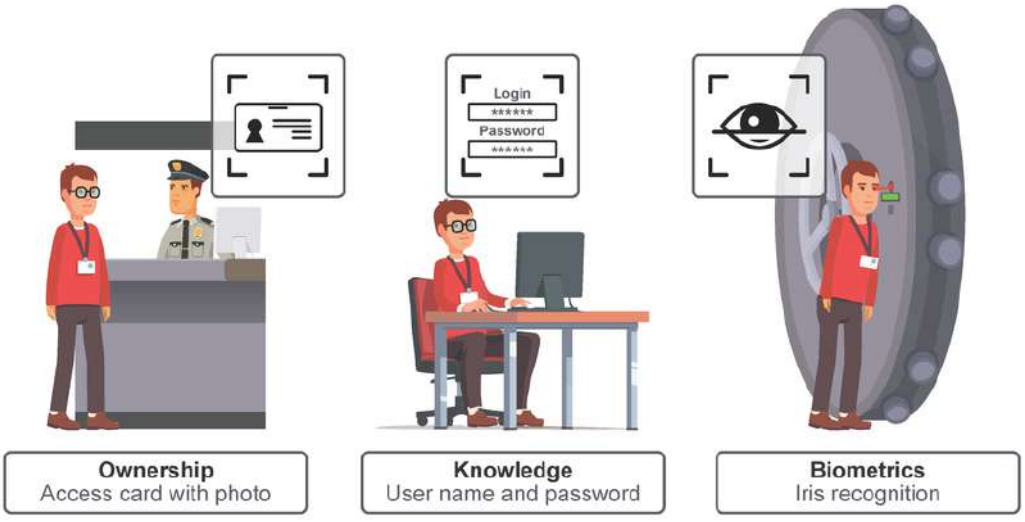

Figura 3.1: Exemple tipuri de autentificare[12]

3.1. Niveluri de identificare

Un prim factor este cel de posesie (**Something You Have**). Factorul de posesie (ownership factor) atestă faptul că titularul contului deține un obiect fizic sau digital unic, inaccesibil altor entități. Exemplele relevante includ cardurile inteligente (smart cards), dispozitivele de tip token (key fob) sau terminalele mobile capabile să genereze ori să recepționeze coduri criptografice. Această metodă adaugă un strat de securitate fizică, deoarece accesul depinde de prezența dispozitivului respectiv.[5]

Factorul biologic sau de inerență (**Something You Are**) se bazează pe biometrie și se referă la caracteristicile intrinseci ale utilizatorului. Autentificarea biometrică utilizează:

– Identificatori fiziologici: Amprente digitale sau palmare, scanări retiniene, recunoaștere facială sau bătăile inimii;
– Identificatori comportamentali: Tipare specifice de mișcare, cum ar fi mersul (gait analysis), ritmul tastării, vocea sau scrisul/semnătura. [13]

Trăsaturile fiziologice sunt considerate mai sigure față de cele comportamentale, fiind aproape imposibil de replicat.

Procesul implică scanarea și înregistrarea acestor identificatori sub forma unui șablon de referință (template). În momentul autentificării, o nouă scanare este comparată cu șablonul stocat pentru a confirma validitatea identității solicitantului.[13]

Un alt criteriu de autentificare poate fi reprezentat și de locația geografică (**Somewhere You Are**). Aceasta se poate referi la o locație geografică determinată de serviciul de localizare al unui dispozitiv sau de adresa sa de protocol internet (IP). O adresă IP a unui dispozitiv poate indica un segment de rețea logic sau poate fi asociată cu o locație geografică prin intermediul unui serviciu de geolocalizare. Autentificarea bazată pe locație poate fi bazată pe locația fizică a portului, rețeaua LAN virtuală (VLAN) sau rețeaua Wi-Fi din cadrul unei rețele locale. Autentificarea bazată pe locație nu este utilizată ca factor principal de autentificare. Cu toate acestea, poate servi ca mecanism de autentificare continuă sau ca funcție de control al accesului. Dacă un utilizator introduce date de autentificare valide, dar adresa sa IP indică o locație care nu corespunde așteptărilor, pot fi implementate restricții de acces pentru a limita privilegiile sau pentru a refuza accesul în totalitate. Un alt exemplu este atunci când un utilizator pare să se conecteze din mai multe locații geografice, ceea ce ar fi fizic imposibil având în vedere timpul de deplasare[5].

Autentificarea multifactorială necesită integrarea mai multor tehnologii. De exemplu, solicitarea unui cod PIN împreună cu data nașterii poate oferi un nivel sporit de securitate în comparație cu introducerea doar a codului PIN, însă aceasta nu constituie o autentificare multifactorială.

În plus, se pot întâlni mențiuni despre autentificarea în doi factori (2FA). Aceasta indică faptul că sunt implicate două componente, cum ar fi un card inteligent bazat pe proprietate sau un factor biometric combinat cu un factor de cunoaștere, cum ar fi o parolă sau un cod PIN[5].

3.2. Metode de Autentificare Multifactor utilizate în practică

3.2.1. Autentificarea Biometrică

Dupa cum este menționat in secțiunea anterioară, prima etapă a implementării autentficării biometrice, datele brute ale utilizatorului sunt colectate pentru a crea un șablon de referință. Ulterior, scanarea este efectuată din nou de fiecare dată când subiectul solicită acces la o resursă, iar datele colectate în timp real sunt comparate cu șablonul care a fost stocat anterior. Gradul de asemănare dintre cele două șabloane trebuie să se încadreze într-o marjă de toleranță predefinită pentru ca accesul să fie permis.[5]

Eficiența și fiabilitatea acestor sisteme pot fi evaluate prin intermediul unor indicatori specifici, exprimați în procente, care determină acuratețea și fezabilitatea soluției[5]:

– Rata de respingere falsă, sau FRR(**False Rejection Rate**) – cunoscută și sub numele de eroare de Tip I, aceasta reprezintă situația în care un utilizator legitim este respins de sistem. Deși cauzează inconveniente utilizatorilor, este considerată mai puțin critică decât eroarea de Tip II.

– Rata de acceptare falsă, sau FAR(**False Acceptance Rate**) – denumită eroare de Tip II, aceasta apare atunci când un intrus este autentificat eronat ca fiind utilizator legitim. Fiind direct responsabilă pentru potențiale breșe de securitate, FAR este considerată cea mai importantă metrică în evaluarea riscului.

Rata de eroare egală (**Crossover Error Rate**, CER) reprezintă punctul în care valorile FRR și FAR coincid. Un nivel scăzut al CER indică o tehnologie mai eficientă și mai precisă. Calibrarea sistemului se face, de regulă, prin ajustarea sensibilității până la atingerea acestui echilibru.

Capacitatea de procesare (**Throughput**) se referă la viteza de creare a șabloanelor și la timpul necesar pentru autentificare, factor esențial în punctele de acces cu trafic intens, precum aeroporturile.

Rata eșecului la înrolare (**Failure to Enroll Rate**, FER) indică procentul de cazuri în care sistemul nu reușește să creeze un șablon valid pentru un utilizator în faza inițială.

Implementarea sistemelor biometrice trebuie să ia în considerare nu numai caracteristicile tehnice, dar și factori externi, precum costurile de achiziție, dificultatea integrării pe dispozitive mobile sau scepticismul utilizatorilor legat confidențialitate. Totodată, tehnologia poate prezenta provocări legate de accesibilitate pentru persoanele cu dizabilități, necesitând o abordare incluzivă în faza de proiectare.

Implementarea sistemelor biometrice se bazează pe trăsături fiziologice (amprentă, față, iris) sau comportamentale (voce, mers, semnătură). Cele mai utilizate metode includ:

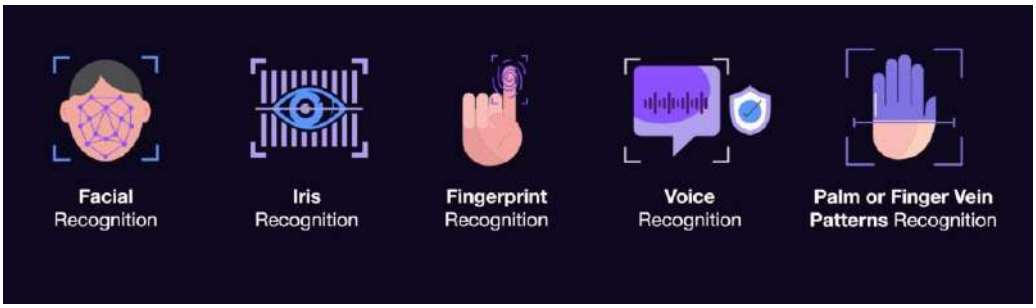

Figura 3.2: Metode de autentificare biometrică [14]

Recunoașterea amprentelor digitale este cea mai răspândită metodă datorită costului redus. Se bazează pe analiza modelelor unice formate de creste (ridges) și văi (valleys) papilare[15]. Identificarea se realizează prin localizarea punctelor caracteristice numite minuții sau minutiae, definite drept punctele în care crestele se termină sau se bifurcă, fiind cele mai importante caracteristici locale.

Recunoașterea facială implică extragerea trăsăturilor cheie, precum ochi, nas, gură, și a distribuției geometrice a acestora pe suprafața feței[15]. Sistemele moderne utilizează adesea camere cu infraroșu pentru a preveni tentativele de fraudare prin fotografii (anti-spoofing)[5]. Deși este o metodă foarte accesibilă și non-invazivă, poate fi influențată de variații de iluminare sau postură.

Recunoașterea irisului este considerată una dintre cele mai precise metode, fiind dificil de replicat. Irisul reprezintă regiunea inelară dintre pupilă și scleră, având o textură complexă care rămâne stabilă pe tot parcursul vieții[13].

În ceea ce privește recunoașterea vocală, aceasta se află la granița dintre biometria fiziologică și cea comportamentală. Sistemele analizează caracteristici unice precum tonul, ritmul și gama de frecvențe, utilizând coeficienți cepstrali în frecvență Mel (MFCC) pentru a extrage calitățile

tonale ale vorbitorului. Un avantaj major este posibilitatea autentificării non-invazive și de la distanță, de exemplu, prin intermediul telefonului, metodă adoptată deja cu succes de numeroase instituții bancare[13].

Recunoașterea vasculară,și anume a venelor degetului sau palmei reprezintă o metodă recentă care analizează modelul unic al vaselor de sânge de sub piele. Deoarece trăsătura este internă, este rezistentă la replicare sau falsificare[13].

Sistemele multimodale, care combină două sau mai multe trăsături, de exemplu fața și irisul, oferă o acuratețe semnificativ mai mare și o robustețe sporită comparativ cu sistemele unimodale, reducând considerabil ratele de eroare FAR și FRR.

3.2.2. Token-uir de Autentificare de tip Hardware

Factorul de posesie reprezintă un mecanism de securitate în care utilizatorul demonstrează controlul asupra unui dispozitiv fizic unic, numit autentificator. Acest dispozitiv are capacitatea de a genera sau de a primi un token, care permite verificarea identității utilizatorului. În arhitecturile moderne, există trei tehnologii principale pentru generarea acestor token-uri[5]:

- Autentificarea bazată pe certificate – utilizatorul poate crea un token semnat unic folosind o cheie privată. Semnătura este verificată de furnizorul de identitate folosind cheia publică corespunzătoare. Deși este o abordare extrem de sigură, pentru a fi utilizată necesită o infrastructură complexă de chei publice (PKI).
- One Time Code (OTP) –token-ul este generat prin aplicarea unei funcții hash asupra unui secret partajat și a unui element de sincronizare. Acesta poate fi un indicator temporal (TOTP) sau un contor bazat pe HMAC (HOTP). Avantajul major este că nu necesită o infrastructură PKI, deși depinde de existența unui secret partajat între dispozitiv și server.
- Fast Identity Online (FIDO) Universal 2nd Factor (U2F)– Această tehnologie utilizează perechi de chei publice/private pentru fiecare cont în parte, eliminând riscurile asociate transmiterii unui secret partajat. Cheia privată rămâne securizată pe dispozitivul U2F, în timp ce cheia publică este înregistrată pe server pentru verificarea semnăturilor.

Un token de tip hardware utilizează un procesor criptografic securizat pentru a preveni extragerea datelor sensibile. Dispozitivele care sunt cele mai frecvent utilizate pentru a implementa acest nivel de securitate sunt[5]:

- Cardurile inteligente – păstrează cheia privată și certificatul digital al utilizatorului. În timp ce un cod PIN permite accesul la card, utilizarea acestuia necesită un cititor fizic, prin contact sau prin tehnologie NFC;
- Generatoarele OTP portabile – sunt dispozitive autonome care afișează un cod numeric generat intern. Acestea nu necesită o conexiune directă cu computerul, codul fiind introdus manual de către utilizator;
- Cheile de securitate (Security keys)– reprezintă module hardware portabile (HSM) care se conectează prin USB sau NFC.Deși sunt asociate în principal cu standardul U2F, ele pot susține și alte metode de autentificare. Pentru a preveni utilizarea neautorizată, aceste chei necesită confirmarea prezenței fizice, adesea prin intermediul unui cititor de amprente integrat sau al unui buton de activare, având și un cod PIN ca metodă de rezervă.

3.2.3. Token-uir de Autentificare de tip Software

Token-urile de autentificare de tip software reprezintă mecanisme de generare a parolelor cu utilizare unică, numite OTP sau One-Time Password, emise de un furnizor de identitate și transmise către entitatea solicitantă (supplicant). Deși aceste sisteme utilizează frecvent token-uri bazate pe contor (counter-based), ele încorporează și o perioadă de expirare temporală pentru limitarea ferestrei de expunere. Pentru acestea, se disting două categorii principale de implementare[5].

Prima categorie este transmiterea prin Canale Out-of-Band. Utilizarea mesajelor de tip SMS sau Email pentru transmiterea codurilor OTP este considerată o metodă deficitară din perspectiva securității. Din punct de vedere conceptual,această metodă este clasificată drept verificare în doi pași (2-Step Verification), nu autentificare multifactorială, deoarece nu îndeplinește pe deplin criteriul factorului de posesie[16].

O alternativă superioară o reprezintă generarea codurilor OTP prin intermediul unor aplicații dedicate instalate pe telefonul mobil. Procesul de configurare implică trimiterea unui secret partajat, realizată prin scanarea unui cod QR furnizat de entitatea verificatoare. Securitatea acestui model este consolidată prin protecția aplicației folosind credențialele dispozitivului, iar codul este generat local, eliminând riscul interceptării[5].

Cu toate acestea, aplicația rulează pe un dispozitiv cu utilizări multiple, astfel prezența unor programe de tip malware la nivelul sistemului de operare gazdă poate compromite integritatea aplicației de autentificare și, implicit, securitatea întregului proces.

3.2.4. Autentificarea fără parolă

Spre deosebire de sistemele de autentificare multifactor convenționale, unde parola rămâne adesea mecanismul principal sau de rezervă, conceptul de passwordless elimină complet utilizarea factorilor de cunoaștere din arhitectura de securitate. Acest sistem se bazează pe specificațiile FIDO2 și WebAuthn, oferind un cadru în care identitatea este validată prin posesia unui dispozitiv și verificarea locală a utilizatorului[5].

Standardul FIDO2 este un cadru tehnic rezultat din colaborarea dintre Alianța FIDO și W3C, conceput pentru a standardiza accesul securizat în web. FIDO2 nu este un protocol unic, acesta cuprinde două specificații fundamentale[17]:

- Web Authentication (WebAuthn)– este un API JavaScript standardizat de W3C care permite aplicațiilor web să interacționeze direct cu browserul pentru a solicita operațiuni criptografice;
- Client-to-Authenticator Protocol (CTAP)– reprezintă un protocol binar care guvernează comunicarea între sistemul gazdă și un autentificator extern (roaming), precum o cheie de securitate USB, NFC sau BLE.

Pentru securizarea accesului, utilizatorul poate alege între următoarele două categorii[5]:

- Roaming Authenticator– este un dispozitiv extern, portabil, precum o cheie de securitate USB/NFC sau smartphone conectat prin Bluetooth, care poate fi utilizat pe multiple terminale;
- Platform Authenticator– este o componentă integrată direct în sistemul de operare al dispozitivului, spre exemplu Windows Hello, Face ID sau Touch ID.

Spre deosebire de parole, WebAuthn stochează pe server doar o cheie publică. Rezistența sa nativă la phishing este garantată prin origin binding, prin care browserul verifică identitatea DNS a site-ului și refuză să utilizeze o acreditare creată pentru site-ul înregistrat pe un domeniu malițios.

Securitatea materialului criptografic se bazează pe radăcini de încredere, sau Roots of Trust. Cheile private sunt generate în medii izolate, asigurând proprietatea de non-exportabilitate.

Atestarea reprezintă mecanismul prin care un dispozitiv de autentificare, cum ar fi o cheie de securitate FIDO sau modulul TPM al unui sistem de calcul, demonstrează că reprezintă o rădăcină de încredere. Fiecare dispozitiv este fabricat cu un certificat de atestare și un identificator de model unic. În etapa de înregistrare, dacă entitatea verificatoare, sau relying party, solicită atestarea, dispozitivul utilizează această cheie pentru a transmite un raport de conformitate. Astfel, entitatea verificatoare poate valida raportul pentru a confirma marca și modelul dispozitivului, asigurându-se totodată că acesta îndeplinește proprietățile criptografice impuse[5].

Procesul de autentificare este definit prin următorul set de pași, alcătuind un cadru robust pentru acest tip de arhitectură[5]:

a) Utilizatorul optează pentru o categorie de autentificator, roaming sau platform authenticator;
b) Se stabilește o metodă securizată sau un gest local pentru a confirma prezența utilizatorului și pentru a debloca dispozitivul, precum amprentă, PIN sau recunoaștere facială, informație validată exclusiv local de către autentificator, fară a părăsi dispozitivul;
c) Pentru fiecare serviciu nou, autentificatorul generează o pereche unică de chei criptografice de tip public/privat. Cheia publică este trimisă serverului (Relying Party, RP), care o asociază contului, finalizând procesul de **înregistrare**;
d) În momentul autentificării, utilizatorul efectuează gestul local pentru a accesa cheia privată. Aceasta este utilizată pentru a semna digital o confirmare a succesului verificării locale, care este ulterior transmisă către RP sub forma unui răspuns la „provocarea” (**challenge**) primită.
e) RP utilizează cheia publică stocată anterior pentru a verifica semnătura digitală, validând astfel identitatea utilizatorului și inițiind sesiunea.

În figurile următoare sunt reprezentate procesele, numite și ceremonii, de înregistrare și autentificare:

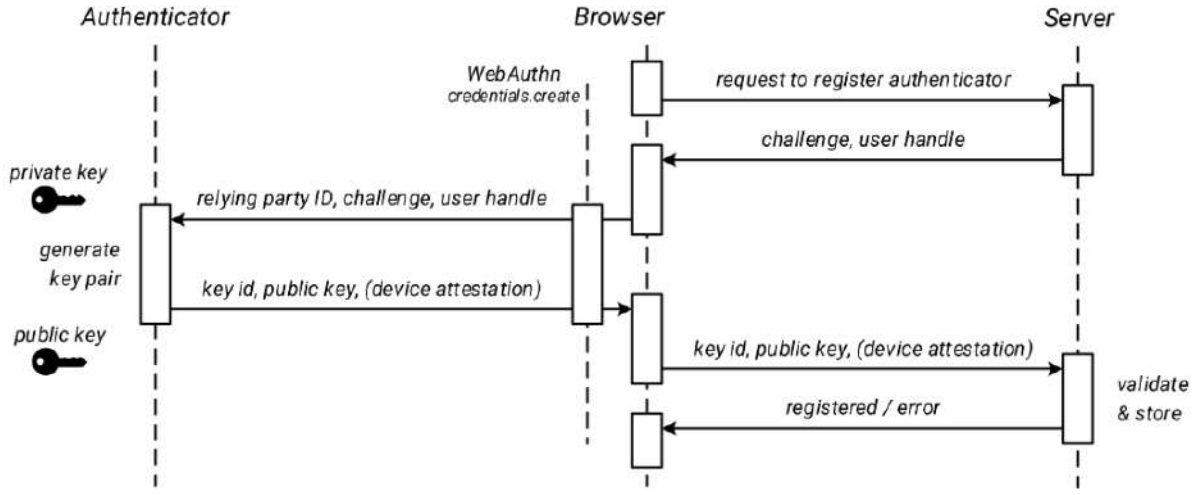

Figura 3.3: Procesul de înregistrare al unui dispozitiv autentificator FIDO2/WebAuthn utilizând metoda „credentials.create”[18]

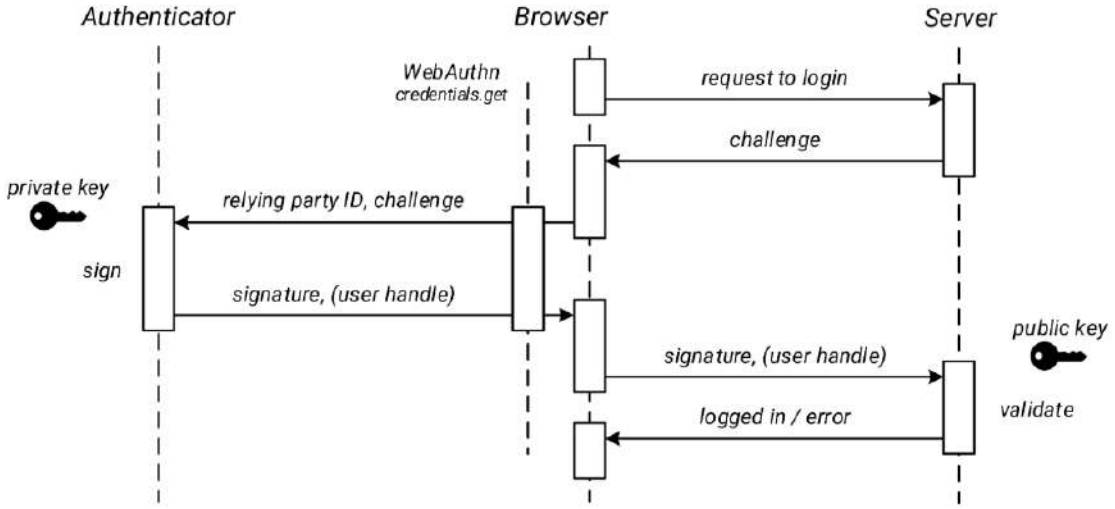

Figura 3.4: Procesul de autentificare al clientului FIDO2/WebAuthn utilizând metoda „credentials.get"[18]

Standardele FIDO2 și WebAuthn oferă o soluție eficientă pentru eliminarea vulnerabilităților cauzate de parole, reducând riscurile de phishing și furt de date prin utilizarea criptografiei asimetrice. Deși stocarea securizată pe hardware crește semnificativ protecția, succesul implementării într-o organizație depinde de echilibrarea securității cu procese clare de recuperare a accesului în caz de pierdere a dispozitivelor.

3.3. Protocoale de federație: OAuth 2.0 și OpenID Connect (OIDC)

Federația de identitate permite unei aplicații să delege verificarea identității unui furnizor extern de încredere, eliminând nevoia de a stoca și de a gestiona parole proprii pentru fluxul respectiv. Cele două protocoale care stau la baza acestui model sunt OAuth 2.0, pentru autorizare, și OpenID Connect, pentru autentificare.

OAuth 2.0 (RFC 6749) este un cadru de autorizare, nu de autentificare. Scopul lui este să acorde unei aplicații, care este clientul în acest scenariu, acces delegat și limitat la resursele unui utilizator, respectiv deținătorul resursei, fără ca utilizatorul să își partajeze parola. Protocolul definește patru roluri:

- deținătorul resursei
- clientul
- serverul de autorizare
- serverul de resurse

Accesul este efectuat printr-un token de acces, cu domeniu de aplicabilitate delimitat prin scopuri (scopes).

Pentru aplicațiile web cu backend, fluxul recomandat este cel care utilizează cod de autorizare. În acest scenariu, clientul redirecționează utilizatorul către serverul de autorizare, transmițând identificatorul de client, adresa de redirecționare, scopurile cerute și un parametru de stare. Utilizatorul se autentifică pe serverul de autorizare și își acordă consimțământul pentru partajarea datelor. Ulterior, serverul de autorizare redirecționează utilizatorul înapoi la client, atașând un cod de autorizare de unică folosință. Clientul schimbă codul, printr-o cerere

server-la-server autentificată cu secretul clientului, pe un token de acces. Ca ultim pas, clientul folosește token-ul de acces pentru a interoga serverul de resurse.

Codul intermediar evită expunerea token-ului în bara de adrese a browserului, motiv pentru care fluxul cu cod l-a înlocuit pe cel implicit. Clienții confidențiali, și anume cu backend, folosesc secretul clientului, iar clienții publici, reprezentați de aplicații mobile sau de tip single-page, folosesc extensia PKCE, care leagă cererea inițială de schimbul de cod printr-o valoare derivată criptografic. Token-urile de acces sunt temporare, iar reînnoirea lor se face printr-un token de reîmprospătare, cu rotație la fiecare folosire.

Protocolul OpenID Conect este responsabil cu autentificarea, ulterior autorizării oferite de OAuth. Diferența este dată de token-ul de identitate (ID Token), emis sub forma unui JSON Web Token semnat semnat, care conține emitentul (iss), subiectul (sub), destinatarul (aud), momentul emiterii (iat), expirarea (exp) și, opțional, un nonce care leagă token-ul de cererea inițială pentru a preveni reutilizarea. Solicitarea acestui token se face prin includerea scopului OpenID în cererea de autorizare.

De asemenea, OpenID Connect definește un punct final *UserInfo*, de la care clientul obține atribute suplimentare despre utilizator, și un document de descoperire publicat la adresa standard *.well-known/openid-configuration*, care expune cheile publice și capabilitățile furnizorului. Validarea autentificării presupune decodificarea token-ul de identitate, verificarea semnăturii cu certificatele publice ale furnizorului și confirmarea revendicărilor iss, aud și exp.

Securitatea fluxului depinde de verificările parametrilor. Parametrul state previne atacurile *Cross-Site Request Forgery* în timpul redirecționărilor, fiind o valoare aleatorie verificată la întoarcerea de la furnizor. Parametrul *nonce* previne reutilizarea unui token de identitate interceptat. De asemenea, toate adresele de redirecționare trebuie înregistrate într-o listă aprobată, pentru a împiedica redirecționarea către un domeniu controlat de atacator. Validarea revendicărilor iss și aud asigură că token-ul a fost emis de furnizorul așteptat și pentru clientul corect.

3.4. Limitări și riscuri ale sistemelor MFA clasice

Tranziția către digitalizare a dus la o reevaluare a securității identității, marcând trecerea de la sistemele bazate pe un singur factor, în special parole, catre autentificarea multifactor. Cu toate acestea, implementarea MFA nu reprezintă sfârsitul amentințărilor, deoarece atacatorii au perseverat în a identifica vulnerabilități și noi metode de exploatare.

În subcapitolul următor sunt prezentate principalele vulnerabilități ale MFA tradiționale, subliniind nevoia trecerii la metode moderne, rezistente la phishing.

3.4.1. Vulnerabilitățile sistemelor bazate pe SMS

Utilizarea SMS-ului ca al doilea factor de autentificare este din ce în ce mai contestată, atât în literatura de specialitate, cât și în ultimele publicații ale NIST [16], din cauza riscului generat de infrastructura de telecomunicații globală.

Protocolul Signaling System No.7 (SS7), baza rețelelor de telefonie, a fost creat in anii '70, atunci când doar operatorii naționali de telecomunicații aveau acces la rețea, iar singurele considerente de securitate erau pentru securitatea fizică. SS7 gestionează semnalele de control, transmise separat de comunicațiile de voce sau date [19].

Prin urmare, atacatorii pot exploata lipsa de securitate a SS7 pentru un atac. Aceștia pot obține identificatorul unic al abonatului și locația curenta a acestuia, pentru ca ulterior să

modifice parametrul corespunzător locației, astfel redirecționând orice SMS către propria rețea, inclusiv coduri OTP[19].

Un alt tip de atac care vizează codurile trimise prin SMS este SIM Swapping. Atacatorul utilizează informații obținute din brese de date anterioare pentru a impersona victima în fața serviciului de asistență pentru clienți al unui operator de telefonie. Prin tehnici de inginerie socială, acesta convinge operatorul să porteze numărul de telefon pe o nouă cartelă SIM aflată în posesia sa. Victima pierde astfel accesul la rețea, în timp ce atacatorul primește toate mesajele SMS și apelurile destinate proceselor de resetare a parolelor și autentificare multifactor[20].

3.4.2. MFA Fatigue

O metodă considerată mai sigură decât codurile transmise prin SMS a fost introducerea notificărilor push. În cazul acestui tip de autentificare, utilizatorul primeste o notificare printr-o aplicație dedicată, precum Microsoft Authenticator, pentru a aproba accesul prin apăsarea unui buton pe dispozitivul personal. Această simplificare a autentificării a introdus riscul atacurilor MFA Fatigue.

În cadrul acestui atac, actorul malițios, deținând deja parola victimei, în urma unui furt prin phishing sau malware, declanșează repetat cereri de autentificare de tipul notificărilor push. Obiectivul este de a copleși utilizatorul prin volumul mare de notificări, adesea trimise în timpul nopții, până cand acesta, din greșeală sau frustrare, aprobă accesul[21].

În consecință, standardele actuale [16] recomandă implementarea „number matching", unde utilizatorul trebuie să introducă un cod afișat pe ecran direct în aplicație, restabilind legătura dintre contextul de autentificare și dispozitivul de aprobare.

3.4.3. Riscurile autentificării biometrice

Biometria este percepută ca fiind lipsită de vulnerabilități în ceea ce privește autentificarea, deoarece se bazează pe factorul de inerență. Totuși, vulnerabilitățile sistemelor biometrice pot duce la consecințe considerabile, mult mai grave decât in cazul sistemelor tradiționale, din cauza naturii permanente a datelor biometrice.

O primă amenințare este reprezentată de atacurile de prezentare, sau *spoofing*. Acestea utilizează artefacte fizice, precum fotografii 2D, măști 3D sau mulaje de silicon, pentru a păcăli senzorii. Deși sistemele moderne includ detecția vitalității, sau *liveness detection*, acestea pot fi depășite de artefacte complexe [22][23].

O vulnerabilitate majoră a telefoanelor mobile este *MasterPrint*. Din cauza dimensiunii reduse, senzorii captează un procent redus din amprentă, generând o pierdere de entropie [24]. Acest fapt permite atacuri de tip dicționar folosind amprente sintetice, numite *MasterPrints*, care posedă trăsături comune ale unei mari părți din populație, având o rată de succes de până la 27% în anumite configurații[25].

Atacurile de injecție video reprezintă o amenințare unde atacatorul nu prezintă un obiect fizic camerei. Acesta introduce un flux video sintetic (Deepfake) direct în API-ul sistemului de autentificare[26]. În 2024, aceste atacuri au crescut de 9 ori, fiind capabile să ocolească tehnicile tradiționale de detecție a vitalității, deoarece elimină complet verificările prin interfața hardware a camerei[27].

Cel mai mare risc este reprezentat de faptul că, spre deosebire de parole, datele biometrice sunt irevocabile. Odată ce un șablon biometric este furat dintr-o bază de date centralizată, identitatea utilizatorului este compromisă pe viață fără posibilitatea de resetare. Din acest motiv,

standardele moderne, precum FIDO2, impun stocarea datelor biometrice exclusiv pe dispozitivul local, în enclave securizate, folosindu-le doar pentru a debloca o cheie criptografică[28].

Este important de menționat că este posibil ca atacatorii să ocoleasca MFA prin furtul token-ului de sesiune rezultat după autentificarea reușită. Aceștia pot folosi un proxy pentru a intercala o pagină de phishing între victimă și site-ul real. Proxy-ul interceptează în atât parola, cât și codul MFA, obținând ulterior cookie-ul de sesiune prin care va avea acces nelimitat. Acest tip de atac este clasificat ca MitM. Cookie-urile de sesiune pot fi preluate și printr-un malware de tip Infostealer[29].

Analiza vulnerabilităților prezentată anterior demonstrează că metodele MFA tradiționale, precum SMS OTP sau notificările Push, sunt insuficiente în fața atacurilor menționate. Autentificarea biometrică trebuie utilizată cu prudență, preferabil în modele descentralizate, unde datele nu părăsesc dispozitivul utilizatorului. Pentru o securitate sporită, standardul NIST[16] recomandă migrarea către implememtări FIDO2 și passkeys, eliminând dependența de canalele de comunicare vulnerabile și de factorii umani predispuși la eroare.

4. STADIUL CURENT AL TEHNOLOGIILOR DE AUTENTIFICARE

4.1. Standardele NIST SP 800-63

Evoluția securității cibernetice a influențat o reevaluare a modului în care entitățile digitale sunt identificate și verificate. Astfel, National Institute of Standards and Technology (NIST) a publicat Special Publication 800-63B-4 [16], reprezentând nu doar o actualizare a ghidurilor anterioare, dar și o schimbare de paradigmă către un model bazat pe riscuri, rezistență la phishing și centrat pe experiența utilizatorului. Această secțiune analizează specificațiile tehnice ale NIST SP 800-63B-4 ca fiind punctul de referință pentru tehnologiile de autentificare moderne, subliniind modul în care aceste standarde răspund amenințărilor.

4.1.1. Niveluri de Asigurare a Autentificării

NIST SP 800-63B-4 grupează procesele de autentificare în trei niveluri de asigurare a autentificării (AAL - Authentication Assurance Levels), fiecare reflectând creșterea graduală a încrederii în faptul că solicitantul (claimant) este într-adevăr abonatul (subscriber) legitim care controlează secretele de autentificare. Această structură modulară permite organizațiilor să selecteze nivelul adecvat de securitate pe baza unei analize de risc, evitând implementarea unor măsuri excesive pentru servicii cu risc scăzut, dar asigurând protecție maximă pentru activele critice.

Nivelul de Asigurare a Autentificării 1 (AAL1) oferă o încredere de bază în controlul subiectului asupra autentificatorului, fiind destinat serviciilor cu profil de risc scăzut. Din punct de vedere tehnic, acest nivel permite utilizarea autentificării cu un singur factor, acceptând parole memorate și secrete de tip look-up, dispozitive out-of-band sau coduri OTP de tip single-factor. Cu toate acestea, este recomandată utilizarea autentificării multifactor unde este posibil. Managementul sesiunii este flexibil, impunând o limită maximă recomandată de 30 de zile pentru durata totală a sesiunii înainte de a fi necesară o reautentificare, fără a obliga aplicarea unui prag de inactivitate.

Nivelul de Asigurare a Autentificării 2 (AAL2) asigură un grad ridicat de securitate în verificarea identității utilizatorului și reprezintă standardul utilizat în majoritatea aplicațiilor critice, prin folosirea a doi factori de autentificare. Această cerință poate fi îndeplinită fie prin utilizarea unui singur autentificator multifactor, precum un token hardware activat biometric, fie prin combinarea a doi factori independenți, cum ar fi o parolă utilizată împreună cu un cod OTP generat de software. O specificație tehnică definitorie a AAL2 este obligativitatea ca cel puțin un factor să fie rezistent la atacurile de tip replay, precum și necesitatea ca verificatorul să ofere cel puțin o opțiune de autentificare rezistentă la phishing. Controlul sesiunii este considerabil mai strict decât în cazul AAL1, limitând durata unei sesiuni la 24 de ore și impunând un timeout de inactivitate de maximum 60 de minute.

Nivelul de Asigurare a Autentificării 3 (AAL3) asigură cel mai ridicat nivel de securitate în procesul de autentificare, oferind o încredere ridicată prin eliminarea dependenței de secrete care pot fi partajate sau furate. Autentificarea la acest nivel se bazează pe protocoale criptografice cu chei publice, asimetrice, unde utilizatorul trebuie să demonstreze controlul asupra unei chei private stocate exclusiv într-un mediu hardware protejat și izolat, precum TPM sau token USB securizat. Specificațiile tehnice interzic în mod explicit utilizarea cheilor exportabile sau sincronizabile în cloud, utilizându-se chei *device-bound*. Rezistența la phishing și la atacurile de tip replay este obligatorie, iar managementul sesiunii este riguros, cu o durată maximă a sesiunii de 12 ore și un prag de inactivitate de 15 minute, asigurând o prezență continuă și verificată a utilizatorului.

Principalele diferențe între cele trei niveluri sunt prezentate în Tabelul 4.1:

Caracteristică Tehnică	**AAL1**	**AAL2**	**AAL3**
Număr Factori necesari	1 (Single-Factor)	2 (MFA)	2 (MFA cu Protocoale Criptografice)
Rezistență la Phishing	Nu	Da, pentru min. 1 factor	Da, pentru toți factorii
Rezistență la Replay	Nu	Da, pentru min. 1 factor	Da, pentru toți factorii
Controlul Cheii Criptografice	Permisă exportarea	Permisă exportarea	Strict Neexportabilă
Mediu de Stocare	Software / Hardware	Software / Hardware	Hardware Izolat (TPM/SE)
Timeout Sesiune	30 zile	24 ore	12 ore
Timeout Inactivitate	Nu este menționat	Maxim 60 min.	Maxim 15 min.

Tabelul 4.1: Sinteza cerințelor tehnice pe niveluri de asigurare conform NIST SP 800-63B-4.

4.1.2. Actualizarea Strategiei de Securitate a Parolelor

Având în vedere tendința utilizatorilor de a crea parole previzibile atunci când sunt impuse reguli complexe, NIST a eliminat obligativitatea utilizării caractereor speciale, în combinație cu cifre și litere mari. Această decizie se bazează pe cercetări care demonstrează că lungimea parolei este o caracteristică mult mai importantă pentru rezistența în fața atacurilor de tip brute-force față de complexitate.

De asemenea, parolele utilizate ca factor unic pentru nivelul AAL1 trebuie să aibă o lungime minimă de 15 caractere. Pentru sistemele care utilizează autentificarea multifactor, lungimea minimă poate fi de 8 caractere, deși se recomandă un număr mai mare de caractere pentru conturile cu privilegii ridicate. Sunt acceptate toate caracterele ASCII imprimabile, inclusiv spațiul și caractere Unicode, facilitând astfel utilizarea frazelor de acces, sau *passphrases*, acestea fiind mai ușor de reținut, dar dificil compromis prin metode computaționale.

O altă schimbare important de menționat este interzicerea resetării periodice forțate a parolelor. Studiile indică faptul că această practică determină utilizatorii să aleagă parole ușor de ghicit prin modificări minore. Conform NIST, parolele trebuie schimbate doar atunci când există dovezi clare de compromitere a contului sau a bazei de date. Pentru a contracara utilizarea parolelor slabe, standardul impune obligativitatea verificării fiecărei parole noi împotriva unei liste (blocklist). Această listă trebuie să includă parole implicate în breșe de date anterioare,

cuvinte din dicționar, secvențe repetitive și termeni specifici contextului aplicației, cum ar fi numele serviciului sau al organizației.

4.1.3. Rezistența la Phishing și Passkeys

Rezistența la phishing reprezintă un aspect important al autentificării moderne și este definită ca fiind capacitatea unui protocol de a preveni dezvăluirea secretelor către un verificator impostor fără a se baza pe vigilența utilizatorului. Această măsură de protecție este realizată prin două mecanisme de legare criptografică. Primul, numit *Verifier Name Binding*, utilizează numele domeniului DNS pentru a se asigura că autentificatorul eliberează credențialele doar către serverul legitim pentru care au fost create. Al doilea, *Channel Binding*, creează o legătură matematică între protocolul de autentificare și canalul de comunicare TLS securizat, făcând imposibilă interceptarea sau retransmiterea datelor de către un atacator situat la mijlocul conexiunii.

Tehnologia *Passkeys*, bazată pe standardele FIDO2 și WebAuthn, implementează nativ mecanismele de legare menționate anterior. NIST permite utilizarea acestor autentificatori sincronizabili la nivelul AAL2. O condiție tehnică este ca aceste chei să fie protejate prin criptare utilizând un secret controlat exclusiv de utilizator, pe care furnizorul de cloud nu îl poate accesa.

4.1.4. Influența GenAI asupra Metodelor Biometrice

Ca răspuns la evoluția tehnologiilor GenAI, NIST reevaluează integrarea biometriei în procesele de autentificare. Astfel, au fost redefinite regulile privind utilizarea biometriei în cadrul sistemelor de autentificare multifactor, subliniind că aceasta nu reprezintă un autentificator de sine stătător și trebuie să fie asociată cu posesia unui dispozitiv.

Important de menționat este decizia de a interzice utilizarea biometriei vocale pentru autentificare, fapt datorat vulnerabilității acestei metode în fața clonării audio prin IA. Pentru metodele biometrice precum amprenta, recunoașterea facială sau scanarea irisului, NIST impune utilizarea mecanismelor de detectare a atacurilor de prezentare și a atacurilor de injecție. Sistemele PAD verifică dacă datele biometrice provin de la o persoană reală, prevenind utilizarea fotografiilor, măștilor sau a altor copii, în timp ce mecanismele de detecție a injecției urmăresc să împiedice introducerea directă a datelor biometrice în sistem prin intermediul unui software malițios sau al unui emulator de cameră.

De asemenea, se impune o rata de potivire falsă de cel mult 1 la 10.000 pentru toate grupurile demografice și se recomandă ca verificarea biometrică să aibă loc pe dispozitivul utilizatorului, limitând expunerea datelor sensibile.

4.1.5. Gestiunea Sesiunilor și Recuperarea Identității

Protecția identității digitale nu se limitează la momentul autentificării, aceasta se extinde pe întreaga durată a activității utilizatorului.

Device Bound Session Credentials, sau DBSC, este responsabil cu legarea criptografică a cookie-urilor de sesiune de *hardware-protected environment*, transformând token-urile de tip *bearer* în dovezi de posesie care nu pot fi compromise. Dacă un software malițios de tip *infostealer* reușește să extragă cookie-ul de sesiune din browser, acesta este inutilizabil pe alt dispozitiv, deoarece serverul va solicita periodic semnătura cheii private neexportabile stocate în hardware-ul original.

În ceea ce privește procesul de recuperare a conturilor, NIST descurajează metodele bazate pe întrebări de securitate din cauza vulnerabilității lor în fața ingineriei sociale. În schimb, sunt impuse alternative precum codurile de recuperare salvate, notificările *out-of-band* trimise pe canale verificate anterior sau procesele de reverificare a identității prin metode biometrice pentru conturile cu grad ridicat de risc. Astfel de acțiuni asupra contului se recomandă să fie însoțite de notificări către o adresă de contact a utilizatorului, pentru a permite detectarea rapidă a unei tentative de fraudă.

4.2. Analiza comparativă a metodelor MFA în literatura de specialitate

Evaluarea comparativă a metodelor de autentificare are la bază modelul propus de Bonneau et al.[30], a căror analiză a schemelor se bazează pe trei axe complementare:

- utilizabilitatea - efortul cognitiv și fizic cerut utilizatorului
- implementabilitatea - costul, infrasturctura și compatibilitatea cu browserele
- securitatea - rezistenta la phishing, la interceptare, la ghicire, la atacuri asupra severului și la reply

Prin evaluarea a 35 de scheme în funcție de 25 de proprietăți, a fost demonstrat că nicio metodă nu oferă simultan toate beneficiile și că avantajele parolei clasice nu pot fi păstrate integral odată ce se adaugă noi măsuri de securitate. Așadar, alegerea unei metode este un compromis didctate de profilul de risc al aplicației.

Pe axa securității, parola și verificarea prin SMS sunt cele mai slabe metode. Codurile trimise prin SMS sunt expuse interceptării prin protocolul SS7 și prin atacurile de tip SIM swapping. Studiul [31] a vizat cinci operatori de telefonie din Statele Unite și a arătat că toti foloseau metode de verificare ușor de ocolit prin inginerie socială. Au fost identificate 17 servicii web pe care un cont putea fi compromis prin SIM swap, fără a cunoaște parola. Aceste rezultate susțin decizia NIST [16] de a descuraja utilizarea acestei metode ca factor de posesie, reflectată și în excluderea acesteia din implementarea sistemului de față.

TOTP, generat local de o aplicație, elimină canalul de transmisie vulnerabil și protejează împotriva interceptării pasive. El rămâne însă un secret partajat și poate fi introdus de un utilizator pe un site de phishing, iar un cod interceptat poate fi reutilizat în fereastra de validitate. În plus, notificările push reduc efortul utilizatorului, dar sunt vulnerabile în fața atacurilor de tip MFA fatigue, mitigate în standardele recente prin mecanismul number matching [16] [21].

Autentificarea biometrică oferă utilizabilitate ridicată, însă se confruntă cu irevocabilitatea datelor, deoarece un șablon compromis nu poate fi resetat. Un alt dezavantaj este compromisul dintre rata de respingere falsă și rata de acceptare falsă, care este ajustat prin mecanisme de detectare a atacurilor de prezentare. Un studiu asupra percepției utilizatorilor față de această metodă [32] a arătat că mulți cred, în mod eronat, că datele biometrice sunt trimise către server în autentificarea FIDO2, deși acestea rămân pe dispozitiv. Astfel, corectarea acestei percepții poate crește încrederea în metodă, având în vedere că abordările recente propun modele descentralizate, în care datele biometrice nu părăsesc dispozitivul [28].

FIDO2 și cheile de acces sunt singura clasă rezistentă la phishing, la reply și la posibila compromitere a bazei de date conclomitent, datorită utilizării criptografiei asimetrice și a legării răspunsului de domeniu [18]. Utilizabilitatea ridicată a metodei a fost demonstrată prin implementarea la scară largă a Google, pentru o perioadă de 2 ani [33]. Studiul a fost desfășurat într-o organizație alcătuită din 50.000 de angajați, pentru care cheile de securitate au redus timpul de autentificare cu două treimi față de codurile OTP prin SMS, au scăzut suportul tehnic

și au crescut nivelul de securitate. Dezavantajul metodei este dat de complexitatea procesului de recuperare în urma pierderii dispozitivului utilizatorului [16].

Studiile de utilizabilitate arată că diferențele în ceea ce privește securitatea nu sunt percepute de utilizatori. Într-o evaluare comparativă a cinci metode 2FA pe un esantion de 72 de participanți [34], a fost demonstrat că nu dificultățile principale în rândul utilizatorilor sunt date de procesul de configurare și riscul de blocare al accesului, nu de folosirea ulterioară a metodei. Asadar, alegerea celui de-al doilea factor de autentificare pentru conturile critice nu poate fi lăsată preferinței utilizatorului, întrucât acesta nu ia în considerare riscurile de securitate ale acestora, ci utilizabilitatea.

Tabelul 4.2 sintetizează compromisul descris în literatură, pe o scară calitativă.

Metodă	**Securitate**	**Utilizabilitate**	**Implementare**	**Rezistență phishing**
Parolă	Scăzută	Ridicată	Ridicată	Nu
SMS OTP	Scăzută	Ridicată	Ridicată	Nu
TOTP	Medie	Medie	Ridicată	Nu
Push + number matching	Medie	Ridicată	Medie	Parțială
Biometrie	Medie	Ridicată	Medie	Parțială
FIDO2 / passkeys	Ridicată	Medie	Medie	Da

Tabelul 4.2: Compromisul dintre securitate, utilizabilitate și implementare pentru metodele MFA.

A doua parte a analizei se concentrează pe evaluarea metodelor de autentificare în raport cu vectorii de atac în fața cărora sunt vulnerabile și rezultatul posibil al unui atac reușit.

În ceea ce privește FIDO2, Catalano et al. [35] discută posibili vectori de atac care să conteste rezistența la phishing a standardului. În condiții optime, legarea raspunsului de domeniu oferă protecție în fața unui atac de tipul Browser-in-the-Middle, în care traficul este retransmis de către atacatorul aflat într-un punct intermediar între victimă și serviciul real.

Studiul demonstrează că, dacă pagina conține o vulnerabilitate de tip Reflected XSS, atacul menționat combinat cu exploatarea acestei vulnerabilități reprezintă o amenințare pentru FIDO2. Rezultatul arată că rezistența FIDO2 la atacuri depinde de integritatea mediului în care sunt utilizate credențialele. Odată ce un atacator poate executa cod în aplicația legitimă, acesta acționează în interiorul unei sesiuni autentificate, diminuând beneficiile mecanismului de legare la domeniu. În aceste condiții, impactul unui atac poate deveni comparabil cu cel asociat altor metode MFA. Recomandarea este implementarea unor măsuri de protecție suplimentare la nivelul aplicației, preum protecție impotriva XSS și verificarea sesiunii.

O vulnerabilitate greu de combătut este dată de factorul uman. Un studiu ce a vizat o organizație de dimensiune medie [36] a constatat că 40% dintre cheile de securitate erau coduri PIN simple, ușor de ghicit, astfel riscul este creat de alegerile utilizatorului, asemnea alegerii unei parole slabe.

Studiul realizat de Rivera-Dourado et al. [37] a evidențiat diferențe dintre implementările WebAuthn și FIDO2 pe diverse platforme. De exemplu, Windows utilizează componente native ale sistemului de operare pentru gestionarea protocolului CTAP, în timp ce Linux se bazează în pe browser și pe module PAM. Autorii au observat, de asemenea, că unele servicii continuă să utilizeze credențiale non-rezidente, ceea ce limitează posibilitatea unei autentificări complet fără

parolă. Astfel, adoptarea FIDO2 este influențată nu doar de proprietățile sale criptografice, dar și de nivelul de suport oferit de platformele și serviciile utilizate.

La nivelul arhitecturilor enterprise, Bhushan et al. [38] propun combinarea FIDO2 cu autentificarea bazată pe certificate pentru conformitatea cu NIST SP 800-63-4 și pentru o apărare stratificată împotriva atacurilor de tip adversary-in-the-middle, iar Ramcharan et al. [39] integrează autentificarea multifactor în modelul Zero Trust, unde se implementează verificarea continuă. Concluzia studiilor vizate este că autentificarea multifactor reduce riscul de compromitere față de autentificarea cu un singur factor, însă eficacitatea ei depinde de metoda aleasă, de implementare și de comportamentul utilizatorului.

Tabelul 4.3 sintetizează tehnologiile evaluate, vectorii de atac și rezultatele relevante.

Studiu	**Tehnologii**	**Vectorul de atac**	**Rezultatul principal**
Catalano et al. [35]	FIDO2, CTAP2, WebAuthn	Browser-in-the-Middle combinat cu Reflected XSS	BitM aplicat singur nu afectează FIDO2, doar combinat cu o vulnerabilitate XSS
Bhushan et al. [38]	FIDO2, certificate (PKI)	Phishing, furt de credențiale, AiTM	Integrarea dată de certificate și apărarea stratificată asigură conformitatea cu NIST
Ramcharan et al. [39]	MFA în model Zero Trust	Acces neautorizat, breșe de date	MFA este o componentă esențială a Zero Trust, împreună cu aplicarea dinamică a politicilor de securitate
Kantipudi et al. [40]	OTP, biometrie, token-uri hardware	Tentative de ocolire a autentificării	O schemă pe trei niveluri îmbunătățește securitatea, dacă este avută în vedere și utilizabilitatea
Ravilla et al. [36]	FIDO2, chei de securitate	PIN-uri slabe, inginerie socială	40% din cheile de securitate prin PIN-uri ușor de ghicit, rezultă un risc dat de factorul uman
Rivera-Dourado et al. [37]	FIDO2, chei de securitate	Diferențe de implementare, phishing	Vulnerabilități date de fragmentarea între platforme (Windows față de Linux) și credențialele non-rezidente

Tabelul 4.3: Metodelor de autentificare în raport cu vectorii de atac

Așadar, stadiul actual indică o repoziționare a metodelor, nu o înlocuire completă. TOTP rămâne un factor secundar accesibil, iar FIDO2 devine factorul preferat pentru conturile cu privilegii ridicate.

5. IMPLEMENTAREA PRACTICĂ A SISTEMULUI DE AUTENTIFICARE MULTIFACTOR

Acest capitol este dedicat prezentării implementării practice a sistemului de autentificare multifactor. Obiectivul sistemului este demonstrarea eficienței metodelor moderne MFA, în detrimentul metodelor tradiționale, aliniindu-se cu standardele de securitate cibernetică. În urma analizei stadiului curent domeniului, au fost integrate doua metode menționate în literatura de specialitate, în completarea utilizării e-mailului și a parolei sau a protocolului OpenID Connect. Prima metodă este TOTP, sau Time-based One-Time Password, accesibilă prin aplicații precum Google Authenticator. A doua metodă este FIDO2/WebAuthn, un standard recunoscut pentru reziliența în fața atacurilor de tip phising.

Arhitectura inițială prevedea integrarea SMS OTP, însă această metodă a fost exclusă din implementare conform recomandărilor NIST privind identitatea digitală. Utilizarea SMS OTP este descurajată din cauza vulnerabilităților asociate atacurilor de tip SIM swapping și interceptării mesajelor SMS. Prin urmare, în proiect sunt utilizate exclusiv metodele TOTP și FIDO2/WebAuthn.

5.1. Arhitectura Sistemului MFA

Arhitectura sistemului este bazată pe o abordare client-server, separând componentele frontend și backend, iar comunicarea este realizată prin intermediul unui API RESTful. Această structură asigură scalabilitate orizontală prin posibilitatea adăugării unor instanțe suplimentare de server. Separarea garantează mentenabilitate, deoarece modificările pot fi aplicate independent fiecărei componente. Sistemul oferă flexibilitate, deoarece componentele frontend și backend pot fi găzduite în medii separate, iar integrarea cu sisteme terțe este facilitată de interfața API.

Pentru backend au fost alese platforma Node.js și framework-ul Express.js, datorită ecosistemului de librării și modului de gestionare a cererilor concurente. Pentru stocarea datelor este utilizat PostgreSQL, o bază de date relațională preferată pentru capacitățile de indexare și fiabilitatea în mediile de producție.

Dezvoltarea funcționalităților MFA a necesitat integrarea mai multor librării. A fost utilizat pachetul *@simplewebauthn/server* pentru implementarea standardului WebAuthn Level 2, librăria *speakeasy* pentru generarea și validarea codurilor TOTP conform specificațiilor RFC 6238, și librăria *qrcode* pentru generarea codurilor QR necesare în configurarea metodei TOTP. Pentru protocolul OAuth 2.0 și OpenID Connect este integrat pachetul *openid-client*.

Arhitectura procesului de autentificare este prezentată în Figura 5.1. Diagrama surprinde toate scenariile de autentificare, prin adresa de email și parolă sau prin Google, cu sau fără autentificare multifactor activă.

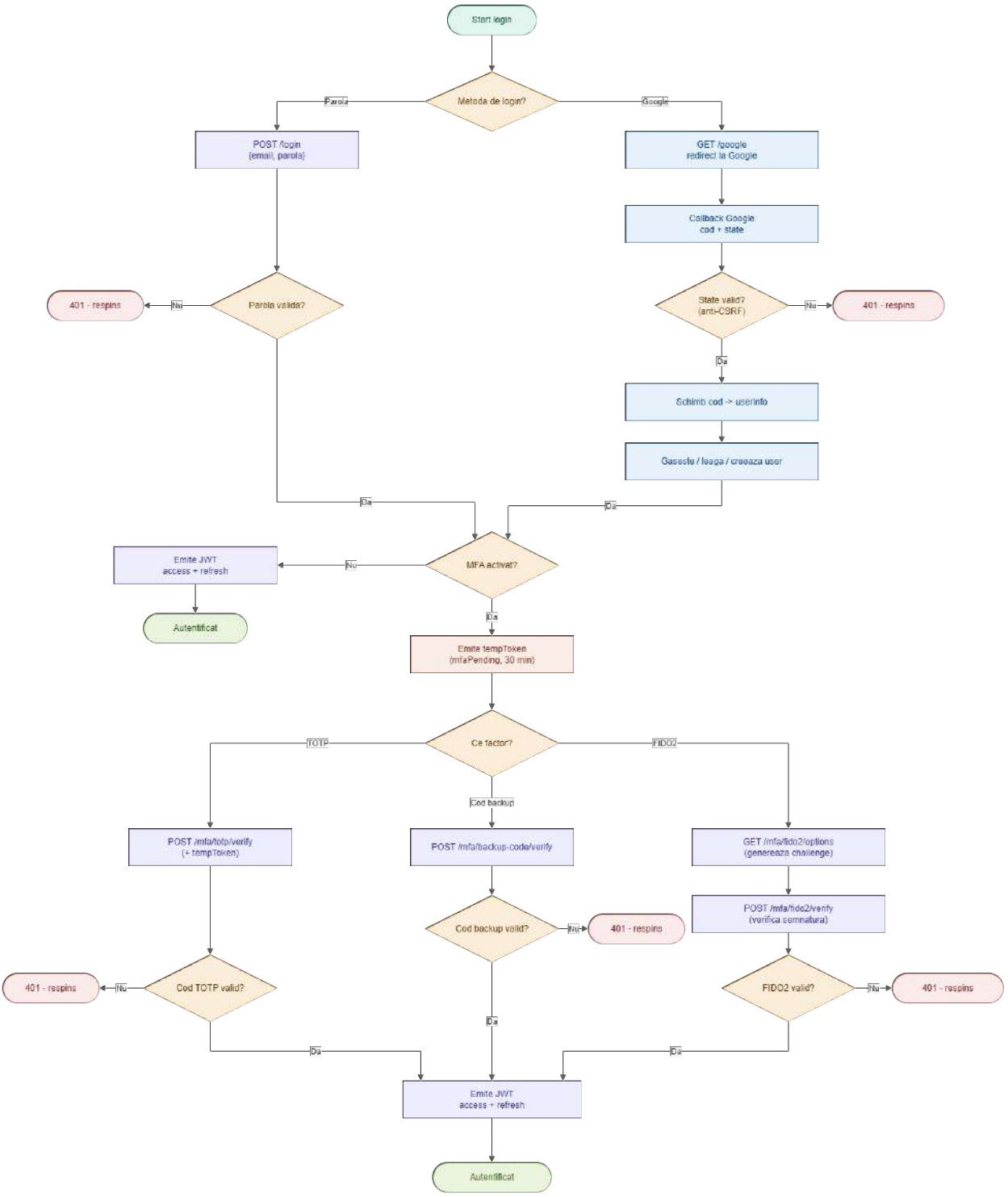

Figura 5.1: Diagrama fluxului de autentificare al sistemului

Aspectele de securitate sunt gestionate prin pachete dedicate. A fost implementat *bcryptjs* pentru obținerea hash-urilor aferente parolelor prin algoritmului Blowfish cu 12 runde de procesare și *jsonwebtoken* pentru manipularea și validarea token-urilor JWT. Protecția împotriva vulnerabilităților web este asigurată de librăria *helmet*, care securizează headerele HTTP. În plus, *express-rate-limit* limitează rata cererilor pentru a diminua riscul atacurilor DDoS, iar *express-validator* este utilizat pentru validarea și sanitizarea datelor introduse de utilizatori.

Aplicația frontend este dezvoltată prin intermediul React.js, o librărie JavaScript utilizată pentru construirea interfețelor grafice pe baza unor componente reutilizabile. Navigarea între paginile aplicației este realizată prin React Router, iar cererile HTTP către backend sunt gestionate prin clientul Axios. Acesta este configurat cu interceptori, contribuind la adăugarea automată a datelor de autentificare la fiecare cerere. Starea globală a aplicației, în special componenta de autentificare, este administrată prin React Context API.

Schema bazei de date permite alocarea mai multor metode MFA fiecărui utilizator, asigurând totodată auditul acțiunilor și gestiunea sesiunilor. Baza de date este alcătuită din 5 tabele principale, reprezentate în Figura 5.2.

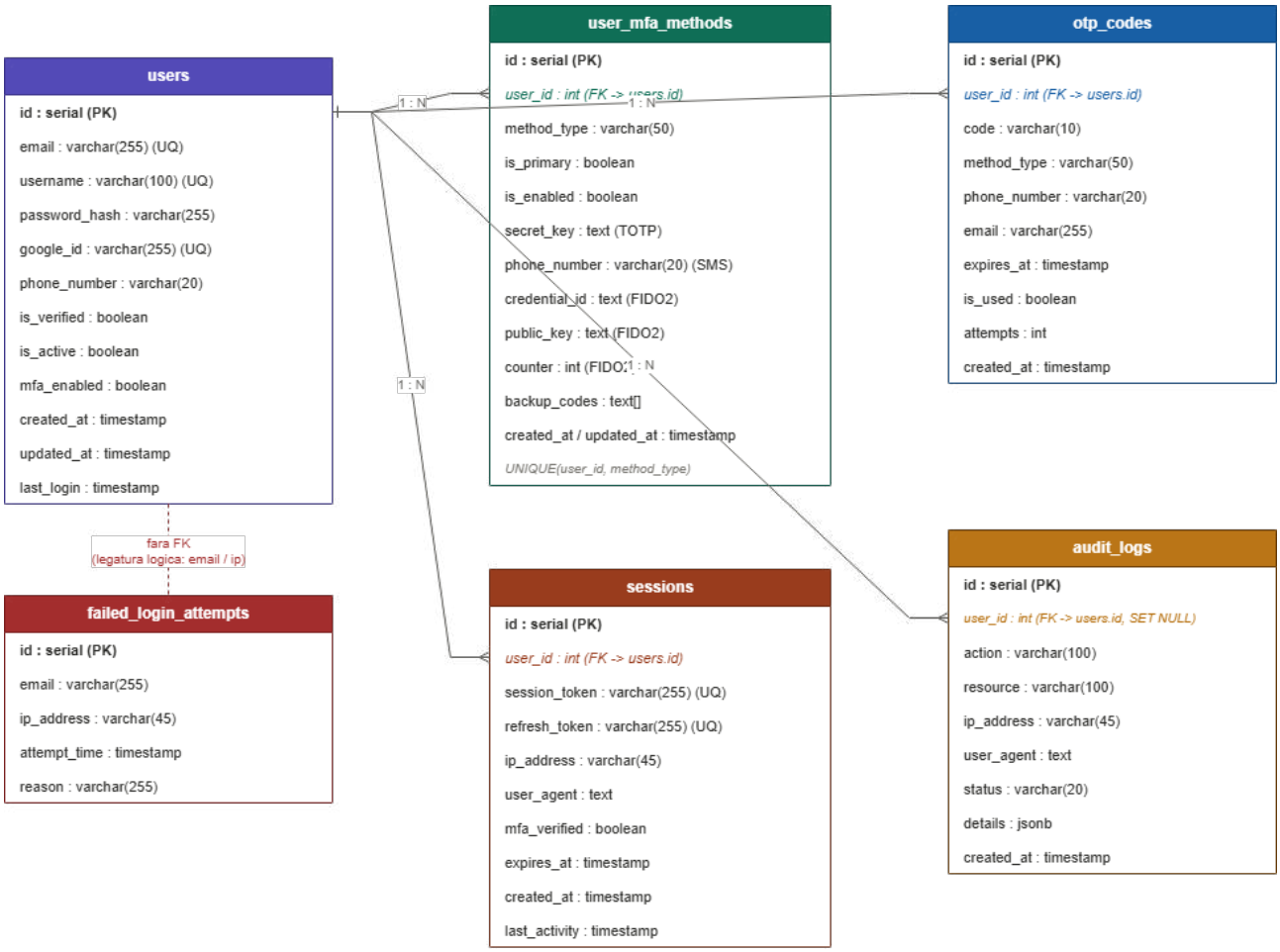

Figura 5.2: Diagrama bazei de date

Tabelul *users* stochează informațiile de bază, precum adresa de email, numele de utilizator, hash-ul parolei, identificatorul Google pentru fluxul OAuth și indicatorul privind starea MFA. În tabelul *user_mfa_methods* sunt înregistrate particularitățile metodelor MFA configurate, incluzând tipul metodei (TOTP sau FIDO2), cheia secretă pentru TOTP, credențialele, cheia publică pentru FIDO2 și codurile de backup. Sesiunile active sunt stocate în tabelul *sessions*,

care include token-urile aferente, marcajul de verificare MFA, adresa IP și agentul utilizatorului. În tabelul *audit_logs* sunt salvate toate acțiunile utilizatorului asupra contului, reprezentând date importante în posibile investigații de securitate. Tabelul *failed_login_attempts* contribuie la detecția și prevenirea atacurilor de tip brute force prin monitorizarea încercărilor eșuate de autentificare.

Optimizarea bazei de date este realizată prin definirea indexurilor pe coloanele des interogate, precum email, username, user_id și session_token. De asemenea, sunt configurate trigger-uri PostgreSQL pentru actualizarea marcajelor de timp (updated_at) și funcții stocate destinate ștergerii automate a datelor expirate (sesiuni și coduri OTP). Structura JSONB este utilizată în stocarea detaliilor de audit, înlocuind eficient stocarea pe coloane multiple.

5.2. Implementarea Componentelor Backend

Implementarea componentei de backend a fost realizată conform principiilor arhitecturii RESTful, dorindu-se o separare a responsabilităților între modulele aplicației. Modelele de date, precum *User.js* și *MFA.js*, gestionează interacțiunea cu baza de date prin intermediul interogărilor SQL parametrizate. Astfel, este asigurată organizarea logicii de acces la date și protecția împotriva vulnerabilităților.

Controllerele, precum *authController.js* și *mfaController.js*, coordonează fluxurile de business ale aplicației, realizând legătura dintre serviciile interne și modelele de date. Serviciile dedicate, asemenea *totpService.js*, încapsulează logica tehnică a autentificării multifactor. Această separare a responsabilităților contribuie la menținerea unei structuri modulare, ușor de extins.

În ceea ce privește middleware-urile, precum *auth.js*, acestea au rolul de a intercepta cererile către API, dar și de a verifica validitatea token-urilor JWT și a drepturilor de acces ale utilizatorilor. Endpoint-urile aplicației sunt definite în componenta de rutare, unde sunt aplicate mecanismele de validare și filtrare a datelor primite.

Componentele utilitare, *auditLogger.js* și *challengeStore.js*, sunt utilizate pentru auditarea acțiunilor, gestionarea *challenge-urilor* și procesarea operațiunilor de sistem.

5.2.1. Implementarea TOTP

Principiul TOTP (Time-based One-Time Password) se bazează pe generarea unor parole temporare prin utilizarea a doi factori: un secret comun și timpul curent. Algoritmul este definit în standardul RFC 6238 și reprezintă o extensie a mecanismului HOTP.

Pentru acest algoritm, o cheie secretă comună este stocată pe server și în aplicația de autentificare a utilizatorului, împreună cu timpul curent segmentat în intervale de 30 de secunde. Pentru fiecare interval temporal, este aplicată o funcție criptografică HMAC și este generat un cod numeric unic, format din șase cifre, validat ulterior în procesul de autentificare.

Implementarea metodei în aplicație este realizată în serviciul *totpService.js*. Primul pas al fluxului de configurare constă în generarea unui secret de 32 de caractere în format Base32 utilizând librăria *speakeasy*. Acest secret reprezintă elementul partajat între server și aplicația de autentificare a utilizatorului, Google Authenticator în acest caz. Ulterior, este generat un URL de tip OTPAuth, care este transformat într-un cod QR prin intermediul librăriei *qrcode*. Codul QR este transmis către interfața browserului sub forma unui Data URL, pentru configurarea a aplicației de autentificare pe dispozitivul mobil al utilizatorului.

În etapa inițială, secretul este stocat temporar în baza de date cu parametrul *is_enabled=false*, până la validarea primului cod generat de utilizator. Verificarea codului introdus este realizată prin funcția *speakeasy.totp.verify()*, fiind utilizată o fereastră de toleranță de 30 de secunde

pentru a compensa diferențele de sincronizare dintre server și dispozitivul mobil. La fiecare autentificare MFA ulterioară, este solicitată introducerea codului temporar generat de aplicația utilizatorului.

Un avantaj al utilizării mecanismului TOTP este prevenirea atacurilor de tip replay. Deoarece codul este unul temporar, posibilitatea reutilizării unui cod interceptat este redusă, contribuind la creșterea securității procesului de autentificare.

5.2.2. Implementarea FIDO2

Pentru a obține un sistem aliniat cu standardele actuale și rezistent în fața amenințărilor moderne, sistemul utilizează standardele FIDO2 și WebAuthn. Acestea sunt recomandate de NIST și W3C pentru rezistența la atacurile de tip phising. În acest scop, este utilizat pachetul *@simplewebauthn/server*, compatibil cu specificația WebAuthn Level 2. Acesta asigură comunicarea dintre server, browser și autentificatorul hardware ales de utilizator, precum YubiKey, Touch ID, Face ID sau Windows Hello.

În cadrul arhitecturii WebAuthn, serverul are rolul de *Relying Party* (RP), iar browserul client intermediază schimbul de informații prin API-ul dedicat. În etapa de înregistrare, serverul generează opțiunile criptografice prin funcția *generateRegistrationOptions()*. Acestea includ datele de identificare ale utilizatorului, domeniul și un parametru de tip *challenge*, stocat temporar pentru a fi validat ulterior. Datele sunt transmise catre frontend, unde este inițiat apelul *navigator.credentials.create()*, prin care este declanșată interacțiunea cu dispozitivul de autentificare.

Autentificatorul hardware generează local o pereche de chei asimetrice și trimite către server cheia publică și identificatorul unic al acreditării (*credentialID*). Cheia privată nu va părăsi dispozitivul utilizatorului, fiind stocată în mod securizat și izolat. Răspunsul primit este verificat pe server prin funcția *verifyRegistrationResponse()*. Astfel, este validat *challange-ul* pentru a preveni atacurile de tip replay, este verificată originea domeniului pentru a combate tentativele de phishing și este confirmată veridicitatea semnăturii criptografice. Ulterior validării, cheia publică și contorul inițial de utilizare sunt stocate în baza de date.

Procesul de autentificare urmează pași similari. Serverul generează un *challenge* nou, iar frontend-ul initiază apelul funcției *navigator.credentials.get()*. *Challenge-ul* este semnat prin cheia privată asociată, iar semnătura este verificată pe server prin cheia publică. De asemenea, sunt validate originea cererii și valoarea contorului de utilizate, care trebuie să fie strict crescătoare, astfel detectându-se tentativele de clonare a autentificatorului. După efectuarea verificărilor menționate, utilizatorul este autentificat.

Implementarea procesului de autentificare evidențiază avantajele utilizării standardului FIDO2/WebAuthn, precum eliminarea parolelor și a secretelor partajate, protecția cheilor private la nivel hardware, rezistența la phishing prin asocierea criptografică cu domeniul aplicației și detectarea dispozitivelor clonate prin mecanismul bazat pe contoare de utilizare.

5.2.3. Implementarea OIDC

Autentificarea prin Google reprezintă o alternativă mai rapidă comparativ cu introducerea adresei de e-mail și a parolei, integrată în majoritatea site-urilor moderne. Această funcționalitate utilizează protocolul OpenID Connect sau OIDC, structurat peste protocolul de autorizare OAuth 2.0. Fluxul este initiat atunci când utilizatorul apasă butonul „COntinuă cu Google”. Interfața frontend redirecționează procesul către serverul de autentificare Google. Această redirecționare include o configurație formată din 3 parametri: identificatorul de client (client

ID), adresa URL de redirecționare (redirect URI) și parametrul *response_type=code* utilizat pentru solicitarea accesului la profilul de bază și la adresa de e-mail.

După autentificarea utilizatorului pe platforma externă și acordarea consimțământului pentru partajarea datelor, Google redirecționează utilizatorul înapoi către aplicația interna la adresa URL specificată. Simultan, în parametri URL-ului, este transmis un cod de autorizare. Serverul backend primește acest cod și transmite o cerere HTTP de tip POST către serverul de token-uri Google. Cererea include codul de autorizare și cheia secretă a clientului (*client_secret*). Astfel, sunt obținute un token de acces (access token) și un token de identitate (ID Token), care este sub formă de JSON Web Token (JWT) și stochează datele de profil ale utilizatorului.

Validarea token-ului de identitate este posibilă în backend prin decodificarea formatului și verificarea semnăturii digitale utilizând certificatele publice emise de Google. Sunt verificate 3 elemente specifice din structura token-ului: emitentul (iss), destinatarul (aud) și valabilitatea temporală (exp). După confirmarea datelor, este realizată identificarea sau înregistrarea utilizatorului. Sistemul interoghează baza de date pe baza identificatorului google_id. Dacă identificatorul există, utilizatorul este autentificat în contul asociat. În caz contrar, este creat un cont nou pe baza datelor de contact furnizate și sunt generate token-urile de sesiune.

Pentru securizarea fluxului este utilizat parametrul state, cu scopul de a preveni atacurile de tip Cross-Site Request Forgery (CSRF) în timpul redirecționărilor. Validarea procesului se bazează pe verificarea ID Token-ului primit. Toate adresele URL de redirecționare sunt configurate într-o listă aprobată în consola Google Cloud.

5.2.4. Gestionarea Identității și a Sesiunii

Gestionarea identității și a sesiunilor este bazată pe standardul JSON Web Tokens (JWT), definit prin RFC 7519 și implementat cu ajutorul librăriei *jsonwebtoken*. Pentru asigurarea securității și separării responsabilităților, sunt utilizate 3 tipuri de token-uri.

Primul este token-ul temporar (*temp token*), generat după validarea credențialelor de autentificare (e-mail și parolă) pentru utilizatorii cu autentificare multifactor activată. Acesta este de forma userId, mfaPending: true , are o valabilitate de 10 minute și restricționează accesul la resursele aplicației până la finalizarea verificării MFA.

După finalizarea procesului de autentificare este generat token-ul de acces (*Access Token*), care include identificatorul utilizatorului (userId) și este valid timp de 24 de ore. Acest token este transmis în antetul cererilor HTTP sub forma *Authorization: Bearer ¡token¿* pentru accesarea endpoint-urilor API-ului.

În paralel, este emis un *Refresh Token* cu o valabilitate de 7 zile, utilizat pentru obținerea unor noi token-uri de acces fără a solicita reluarea procesului de autentificare. În mediile de producție, acesta este stocat sub formă de cookie *httpOnly*, pentru o securitate sporită.

Controlul accesului către resursele protejate este realizat prin middleware-ul *authenticateToken*, care interceptează cererile către rutele securizate. Acesta extrage token-ul din antetul cererii, verifică semnătura folosind cheia secretă a serverului, validează perioada de valabilitate și identifică utilizatorul asociat. Datele utilizatorului sunt încărcate în contextul cererii curente(*req.user*) pentru a fi accesibile controllerelor aplicației. În cazul unui token lipsă, modificat sau expirat, accesul este blocat, iar serverul returnează codurile HTTP corespunzătoare, *401 Unauthorized* sau *403 Forbidden*.

5.2.5. Considerații de secutitate în proiectare

Pentru a asigura trasabilitatea acțiunilor și pentru a facilita investigațiile tehnice de secuitate, este integrat un mecanism de înregistrare a jurnalelor de audit destinat monitorizării activității utilizatorilor. Evenimentele înregistrate acoperă acțiunile utilizatorului de autentificare și administrare a securității aplicației.

Printre acțiunile monitorizate se numără procesul de înregistrare, încercările de autentificare reușite sau eșuate, stările de așteptare a validării MFA și rezultatele verificărilor multifactor. De asemenea, sunt incluse în loguri și configurarea metodelor TOTP și FIDO2, starea de activare a autentificării multifactor, generarea codurilor de rezervă și deconectarea utilizatorilor.

Fiecare înregistrare din jurnalul de audit conține informații despre acțiunea efectuată, precum identificatorul utilizatorului, tipul evenimentului, resursa vizată (*auth*, *mfa* sau *user*) și statusul operațiunii (succes, eroare sau avertisment). În plus, sunt stocate atribute tehnice destinate analizei incidentelor, precum adresa IP a cererii, clientul utilizat (browser și sistem de operare), marcajul temporal și date suplimentare structurate în format JSONB.

Componenta de backend include două măsuri de securitate destinate să prevină atacurile cibernetice. Prima este limitarea ratei cererilor (*Rate Limiting*), realizată cu ajutorul pachetului *express-rate-limit*. Sistemul permite 100 de cereri la fiecare 15 minute pentru o adresă IP, iar pentru rutele de autentificare limita este de 5 încercări în același interval, diminuând riscul atacurilor de tip *brute force*. Limita cererilor este completată de monitorizarea tentativelor consecutive de autentificare eșuate la nivelul bazei de date.

Protecția împotriva atacurilor de tip SQL Injection este asigurată prin utilizarea interogărilor parametrizate din clientul *pg*, evitând concatenarea directă a instrucțiunilor SQL malițioase. Datele primite sunt validate și sanitizate prin librăria *express-validator*, înaintea procesării în logica aplicației sau în baza de date.

Riscul atacurilor Cross-Site Scripting (XSS) este diminuat prin implementarea librăriei *Helmet.js*, care configurează antetele HTTP de securitate, precum *Content-Security-Policy*. Componenta frontend include mecanisme native pentru escaparea conținutului randat în JSX. De asemenea, vulnerabilitățile Cross-Site Request Forgery (CSRF) sunt controlate prin plasarea token-urilor JWT strict în antetul de autorizare, fiind oprită stocarea în cookie-uri expuse browserului. Configurația Cross-Origin Resource Sharing (CORS) este restricționată la originile autorizate, fără utilizarea setărilor de tip wildcard.

Securitatea acreditărilor este asigurată prin algoritmul *bcrypt* pentru hashing-ul parolelor, configurat cu 12 runde de procesare și un *salt* unic generat aleatoriu pentru fiecare parolă.

În plus, procesele de verificare sunt implementate pentru a opera într-un timp constant, reducând riscul atacurilor de tip *timing attack*.

La nivel de infrastructură, comunicația este securizată prin protocolul HTTPS. Politicile de securitate sunt consolidate prin configurarea antetului *Strict-Transport-Security* prin *Helmet.js* și prin redirecționarea traficului HTTP către conexiuni securizate în mediile de producție.

5.3. Implementarea Interfeței Grafice

Interfața grafică este alcătuită dintr-un număr de pagini web intuitive și plăcute vizual, astfel încât utilizatorul, indiferent de nivelul de cunoștințe tehnice, să seteze și să înțeleaga necesitatea autentificării multifactor. Componenta frontend a fost realizată cu ajutorul bibliotecii React.

Prima pagină este cea de autentificare, prezentată in Figura 5.3. Utilizatorul poate alege să se autentifice cu adresa de email și parola sau prin contul Google. Dacă acesta nu dispune de un cont deja existent, își poate crea unul prin accesarea paginii de înregistrare, Figura 5.4, prin intermediul butonului „Înregistrează-te".

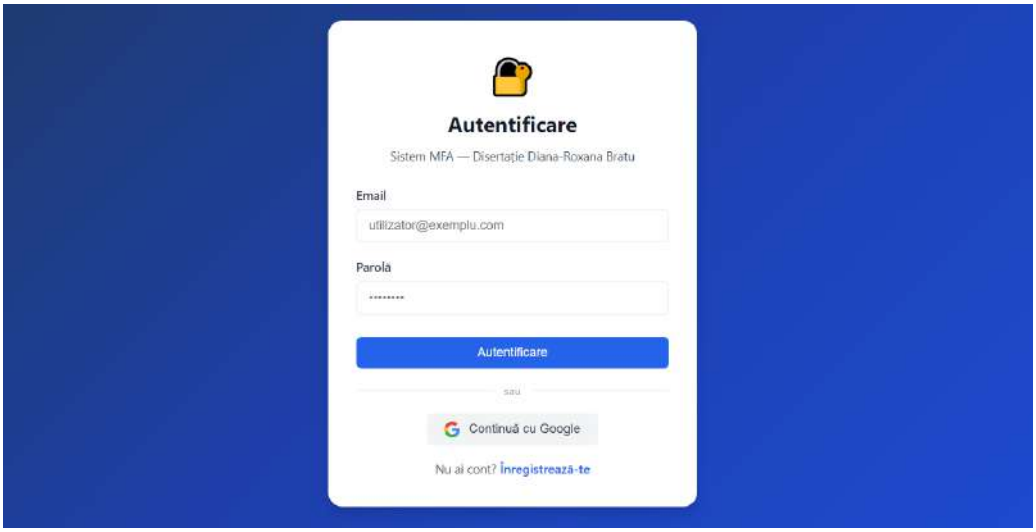

Figura 5.3: Pagina de autentificare

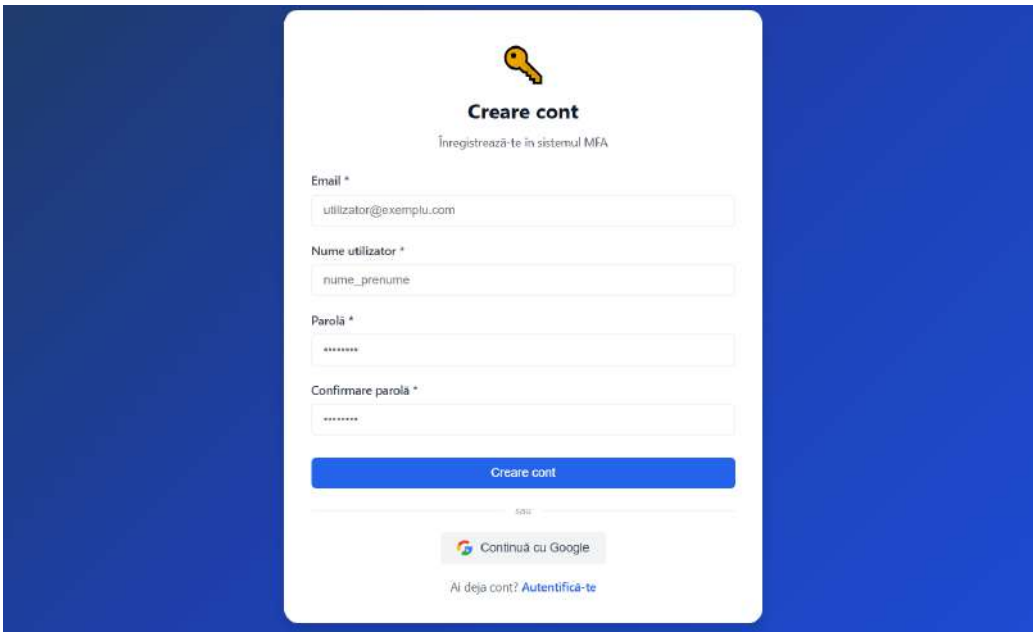

Figura 5.4: Pagina de înregistrare

Astfel, utilizatorul poate alege să își creeze contul prin furnizarea informațiilor din formularul de înregistrare sau cu ajutorul contului Google. Dacă este aleasă varianta din urmă, acesta este redirecționat către pagina *AuthCallbackPage.jsx*, utilizată pentru procesarea răspunsurilor asincrone generate în urma autentificării prin Google OAuth, unde va alege contul dorit și va urma pașii necesarii verificării identității sale.

Dupa ce procesul de înregistrare este complet, se revine la pagina de autentificare și se continuă cu metoda preferată. Dacă este prima autentificare a utilizatorului, cea de-a doua metodă nu este configurată, așadar acesta poate accesa direct pagina principală, ilustrată în Figura 5.5.

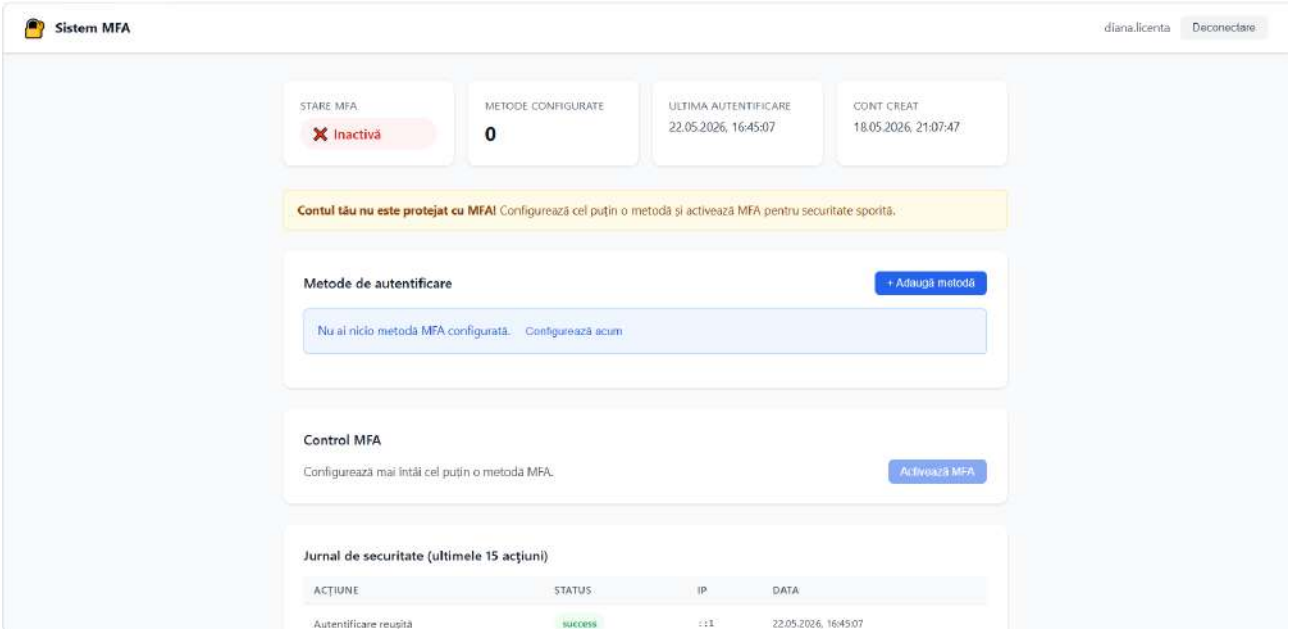

Figura 5.5: Pagina principală

Pagina principală este alcătuită din:

- Informații privind starea securității contului (MFA activă/inactivă), numărul metodelor configurate, data ultimei autentificări și data creării contului;
- Metode de autentificare – prezintă metodele deja configurate sau posibilitatea configurării acestora;
- Control MFA – activarea și dezactivarea metodelor configurate;
- Jurnalul de securitate – informații despre acțiunile inițiate asupra contului, precum autentificare, configurarea MFA, dar și rezultatul acestora, IP-ul utilizat și data efectuării acțiunii menționate.

În continuare, pentru a seta o a doua metodă de autentificare, utilizatorul foloseste butonul „Adaugă metodă” din secțiunea „Metode de autentificare”.

Acesta este redirecționat catre pagina de configurare MFA, de unde poate alege dintre TOTP și FIDO2. Pentru fiecare metodă există informații cu privire la beneficiile pe care le aduc în ceea ce privește securitatea contului, pentru ca utilizatorul să ia o decizie informată, dar si pentru a întelege nevoia unui pas suplimentar în procesul de autentificare.

Pentru configurarea TOTP, se apasă butonul „Generează codul QR”, prezent în Figura 5.6. Figura 5.7 ilustrează rezultatul, respectiv codul QR generat și pașii pentru configurarea aplicației Google Authenticator.

Figura 5.6: Configurarea autentificării cu TOTP

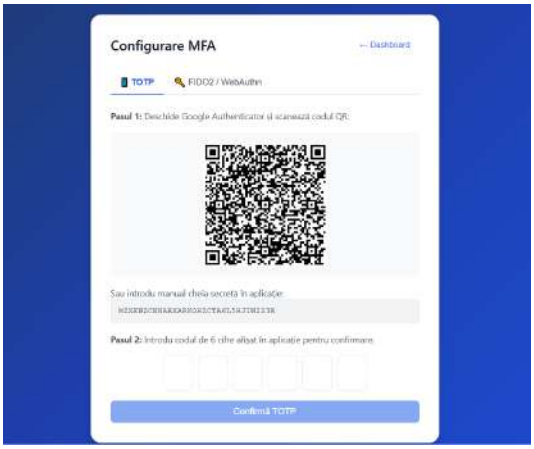

Figura 5.7: Pașii configurării aplicației Google Authenticator și codul QR generat

După ce metoda este configurată, aceasta va fi solicitată pentru următoarea autentificare. De asemenea, pagina principală va fi actualizată în consecință, dupa cum se observă în Figura 5.8.

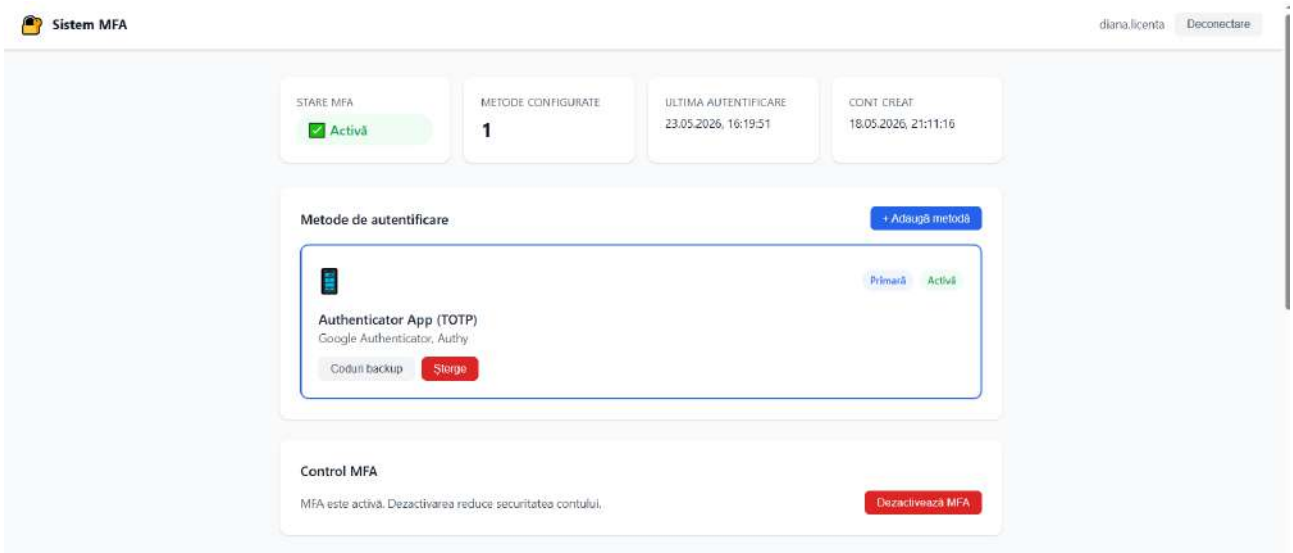

Figura 5.8: Pagina principală după adăugarea TOTP

Autentificarea prin FIDO2 se setează, de asemenea, din pagina de configurări (Figura 5.9). Butonul „Înregistrează dispozitivul” declanșează întregul proces de înregistrare a cheii de acces, prin intermediul API-urilor browserului. Sistemul solicită utilizatorului să aleagă modalitatea de salvare a acesteia (Figura 5.10). Se alege optiunea de a folosi un dispozitiv mobil sau tabletă. Astfel, un cod QR este generat (Figura 5.11). Prin scanarea codului, cheia este transmisă și stocată în siguranță pe dispozitivul ales și este deblocată utilizând metoda biometrică sau codul de deblocare a acestuia.

Figura 5.9: Configurarea autentificării cu FIDO2

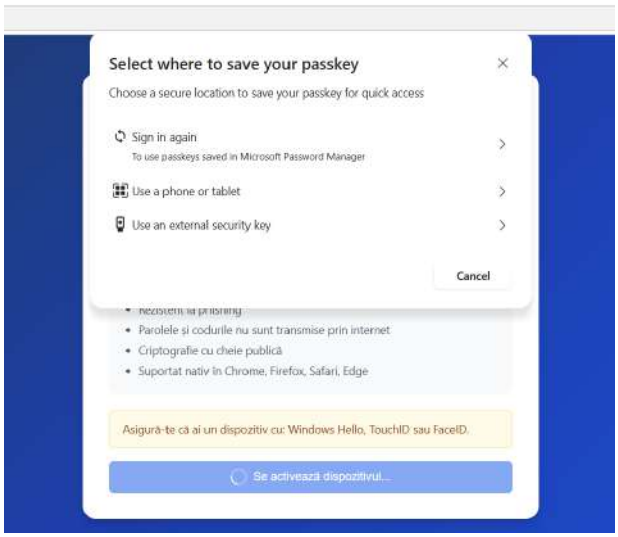

Figura 5.10: Alegerea modalității de salvare a cheii de acces

Figura 5.11: Generarea codului QR pentru transmiterea cheii de acces

În cazul în care metodele principale de autentificare nu sunt disponibile, este posibilă generarea unor coduri de backup pentru recuperarea accesului.

După configurarea ambelor metode de autentificare disponibile, pagina principală este actualizată automat după cum este ilustrat în Figura 5.12.

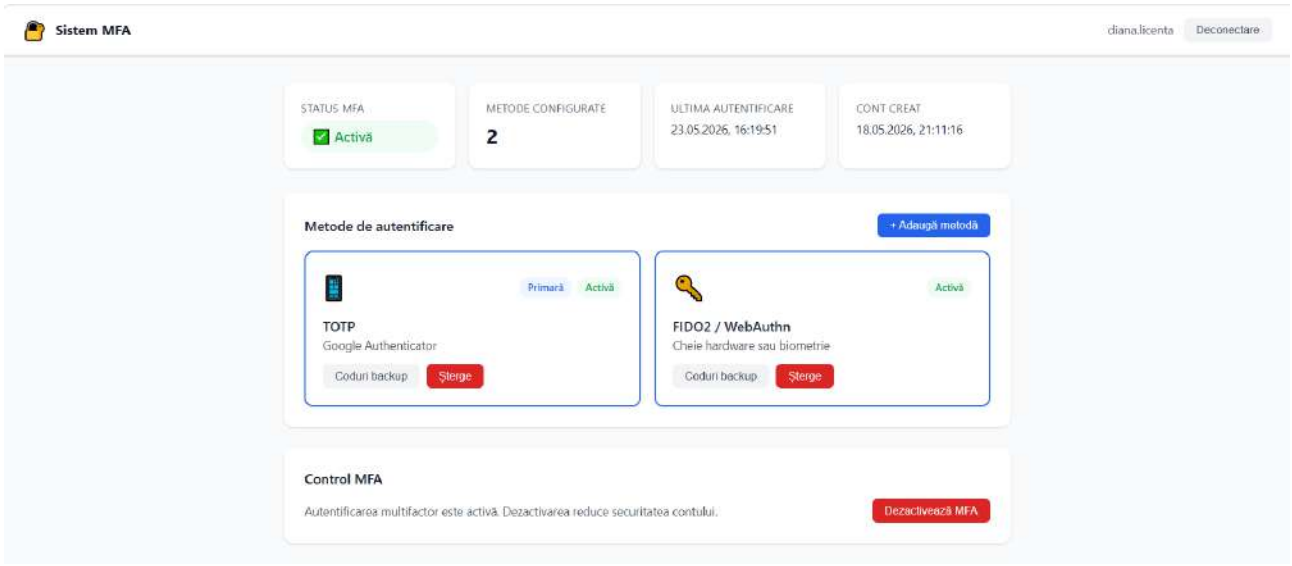

Figura 5.12: Pagina principală după configurarea metodelor de autentificare

Pentru urmatoarea autentificare, după ce utilizatorul va introduce email-ul și parola sau contul Google, va fi redirecționat către cea de-a doua pagina de autentificare. Dacă acesta are ambele metode configurate, va putea alege dintre TOTP (Figura 5.13) și FIDO (Figura 5.13). În caz contrar, va fi disponibilă doar metoda deja configurată. După ce utilizatorul fie introduce codul de 6 cifre din aplicația Google Authenticator, fie alege să folosească cheia stocată pe dispozitivul mobil, acesta va fi autetentificat cu succes și va fi redirecționat către pagina principală.

Figura 5.13: Autentificare prin TOTP

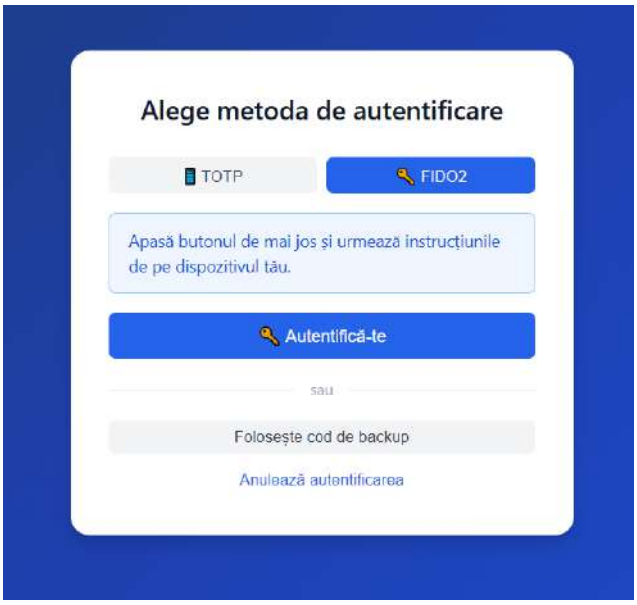

Figura 5.14: Autentificare prin FIDO2

Gestiunea stării globale în aplicația frontend este realizată prin React Context API. Acest mecanism arhitectural previne transferul parametrilor între componente și centralizează datele identității utilizatorului activ. Starea globală încapsulează 3 elemente principale:

- datele de profil (identificatorul, adresa de e-mail, numele de utilizator și starea MFA);
- statutul de autentificare (valoare booleană utilizată pentru redirecționările condiționate);
- stadiul de încărcare a setărilor din memoria locală.

Modulul are 5 tipuri de funcții operaționale. Funcția de înregistrare gestionează crearea entităților, funcția de autentificare procesează cererile și redirecționarea către validarea MFA, funcțiile de validare rezolvă provocările criptografice, iar funcțiile de deconectare și actualizare execută curățarea instanțelor locale și reîmprospătarea stării cu atributele noi.

Interacțiunile de rețea dintre client și server sunt administrate prin serviciul api.js, prin instanța clientului Axios configurată cu 2 interceptori. Primul interceptor, cel de cerere, este executat înaintea transmiterii pachetelor de date către rețea și are rolul de a extrage valoarea accessToken din localStorage, pentru a o adăuga ulterior în header-ul de autorizare. Interceptorul de răspuns monitorizează fluxul de date retrunat de server. Dacă este primit codul HTTP 401, asociat unui token expirat, este extras token-ul de reîmprospătare și este inițiat un apel către endpoint-ul /auth/refresh-token. În cazul imposibilității de reîmprospătare, este curățată memoria și utilizatorul este redirecționat către pagina de autentificare.

6. EVALUAREA SECURITĂȚII ȘI A PERFORMANȚEI SISTEMULUI

Evaluarea sistemului constă în observarea comportamentului aplicației împotriva unui set de vectori de atac derivați din amenințările analizate în capitolele 2 și 3. Scopul este atât demonstrarea importanței mecanismenelor de apărare implementate, cât și delimitarea clasei de atac pentru fiecare metodă, TOTP și FIDO2.

Se ia în considerare în evaluarea sistemului și experiența utilizatorului, influențată de timpul de autentificare adăugat de al doilea factor.

6.1. Simularea vectorilor de atac

6.1.1. Metodologia de evaluare

Mediul de testare cuprinde intreaga aplicație, alcătuită din backend, frontend și baza de date, executată local. Pentru implementarea vectorilor de atac sunt utilizate scripturi Python autonome, care folosesc biblioteca *urllib*. Folosirea exclusivă a bibliotecii menționate rezultă în posibilitatea de a reproduce scripturile cu orice versiune Python.

Vectorii de atac considerați sunt cei mai des întâlniți în atacurile reale asupra sistemelor de autentificare și derivă din amenințările numite în capitolele anterioare. Atacurile implementate sunt următoarele:

- Atac brute force pe autentificarea cu parolă;
- Atac brute force pe autentificarea prin codul TOTP;
- Atac de tip replay pe autentificarea prin codul TOTP;
- Falsificarea originii pentru autentificarea FIDO2;
- Interceptarea traficului prin MitM;

Succesul împotriva tentativelor de atac este definit prin imposibilitatea de a obține accesul în aplicație, fie prin respingerea cererilor printr-un cod de eroare de tipul 4xx sau prin limitarea traficului. Rezultatul este returnat dupa fiecare rulare a scripturilor în terminal, alături de codul HTTP returnat.

6.1.2. Atacul brute force în autentificarea cu parolă și cu TOTP

Atacul de tip brute force are ca scop spargerea unui cod sau a unei parole. Atacatorul încearcă să ghicească parola unui cont prin trimiterea automată a unui număr mare de combinații, fie dintr-o listă de parole uzuale (atac de tip dicționar), fie prin parcurgerea sistematică a caracterelor. Succesul depinde de complexitatea parolei și de existența limitării numărului de încercări la nivelul serverului.

Pentru testarea rezilienței în fața acestui tip de atac, a fost alcătuită o listă de 56 de parole posibile, care include combinații des utilizate sau derivate din numele de utilizator. Scriptul trimite lista către endpoint-ul de autentificare. Limitatorul dedicat rutelor de autentificare permite maxim 5 încercări eșuate consecutive într-un interval de 15 minute pentru o adresă IP. Opțiunea *skipSuccessfulRequests* este activă, astfel doar încercările nereușite de autentificare sunt contorizate.

Primele 5 încercări de autentificare nereușite primesc ca răspuns codul *401 Unauthorized*. De la incercarea a 6-a până la epuizarea tuturor celor 56 de parole, codul returnat este *429 Too Many Requests*, iar atacatorul este blocat și nu mai are posibilitatea de autentificare pentru 15 minute. În acest scenariu siguranța contului este dată atat de limita de 5 încercări, dar și de complexitatea parolei. Este impusă o parolă de minim 8 caractere și cel puțin o literă mică, o literă mare, o cifră și un caracter special, rezultatul fiind un alfabet de 70 de caractere posibile. Pentru o parolă de lungime minimă, există 70^8, respectiv $5,76 \times 10^14$ combinații posibile. Timpul de spargere depinde de scenariu. Limitatorul permite 5 încercări în 15 minute, astfel se pot testa 480 parole pe zi. Așadar, epuizarea tuturor combinațiilor de caractere are o durată de $1,6 \times 10^9$ ani, rezultând un scenariu nerealist. În cazul unei parole slabe, precum un cuvânt din dicționar, durata este de ordinul secundelor sau minutelor.

Contextul acestui atac subliniază importanța unui al doilea factor de autentificare, deoarece, dacă atacatorul reușește să treacă de autentificarea prin parolă, fie prin brute force sau prin inginerie socială, accesul este conditionat de posesia unui dispozitiv, pentru TOTP sau FIDO2.

În ceea ce privește posibilitatea unui atac brute force offline, în care atacatorul obține hash-urile parolelor din baza de date, sistemul este protejat de implementarea bcrypt cu cost 12, indiferent de puterea de procesare.

Atacul brute force pentru codul TOTP este asemănător din punct de vedere al implementării. Diferența este dată structura codului, și anume de numărul de caractere și de variabila temporală. Lungimea codului este de 6 cifre, rezultând 10^6 combinații posibile. Pentru a parcurge toate codurile posibile, ținând cont de limita de 5 coduri în decursul a 15 minute, este nevoie de aproximativ 5 ani. Adăugarea variabilei temporale face scenariul de identificare imposibil.

Scriptul vizează 15 incercari (000100 - 000114), pentru a demonstra limitarea încercărilor de autentificare la a 6-a încercare. Codurile de eroare sunt *401 Unauthorized* pentru primele 5 coduri și *429 Too Many Requests* pentru restul.

6.1.3. Atac de tip replay pe autentificarea prin codul TOTP

Atacul de tip replay presupune interceptarea unui cod de autentificare valid, prin metode precum *keylogger*, *phising* sau MitM, și retrimiterea lui imediată pentru a obtine acces. Scenariul testează posibilitatea utilizării unui cod de două ori in fereastra de 30 de secunde.

Verificarea TOTP folosește *speackeasy.totp.verify* cu parametrul window:1 (toleranță de aproximativ 30 de secunde) și nu păstrează o listă a codurilor utilizate anterior. Timpul de validare este de 90 de secunde, interval în care același cod poate fi reutilizat.

Scriptul propus reproduce atacul și demonstrează vulnerabilitatea metodei. Primul pas este autentificarea prin email și parolă, pentru a obține un token temporar. Ulterior, codul TOTP valid din aplicația Google Authenticator este intordus în terminal, pentru a fi trimis către endpoint-ul de verificare, simulând autentificarea legitimă a victimei. Pentru a simula comportamentul atacatorului, codul este retrimis, utilizând același token temporar. Pentru ambele cereri de autentificare răspunsul este 200 OK, ceea ce înseamnă că același cod a fost acceptat de două ori și atacul a avut succes. Rezultatul confirmă vulenrabilitatea metodei în fața atacurilor de tip replay, respectiv lipsa invalidării codurilor deja utilizate. Protecția în fața

acestui vector de atac este limitată la expirarea codului și limitarea numărului de autentificări nereușite.

În schimb, FIDO2 leagă fiecare raspuns de un challenge unic și de un contor strict crescător, oferind protecție criptografică la replay și reprezintă o alternativă mai sigură.

O soluție posibilă pentru a continua utilizarea TOTP este crearea unei liste de coduri utilizate pentru fiecare utilizator în fereastra de timp curentă.

6.1.4. Falsificarea domeniului pentru autentificarea FIDO2

Rezistența la phising a standardului FIDO2 se bazează pe legarea criptografică a raspunsului de autentificare de domeniul aplicației. Pentru fiecare autentificare, browserul include în obiectul *clientDataJSON* domeniul care a inițiat cererea și o semnează împreună cu un *challenge* unic. Serverul are rolul de Relaying Party și acceptă raspunsul doar dacă domeniul coincide cu valoarea înregistrată. Pentru un atac de phishing, atacatorul creează pe un domeniu propriu o pagină care să o imite pe cea reală, iar acțiunile victimei sunt retransmise către serverul legitim.

Scopul scenariului propus este de a testa dacă un răspuns WebAuthn generat pentru un alt domeniu față de cel înregistrat este acceptat de server.

Primul pas este obținerea unui token de acces valid și a unui challenge FIDO2 generat de server. Ulterior, sunt construite patru raspunsuri WebAuthn, în care obiectul clientDataJSON este generat manual, cu domenii diferite de cel legitim, precum:

- http://evil-site.com;
- http://localhost:3001;
- https://phishing-mfa.com;
- http://127.0.0.1:3000.

Răspunsurile create sunt trimise serverului și este înregistrat codul HTTP returnat pentru fiecare. În final, este trimis un al cincilea răspuns, cu domeniul legitim, dar cu date criptografice false.

Toate cele patru răspunsuri cu domenii diferite sunt respinse, deoarece acestea nu coincid cu valoarea *expectedOrigin* configurată pe server. În ceea ce privește utilimul răspuns testat, deși acesta conține domeniul real, este respins, deoarece semnătura criptogtafică este invalidă. Astfel, experimentul demonstrează atât importanta legării de domeniu, cât și a verificării autenticității semnăturii.

Rezultatul obținut subliniază diferența dintre FIDO2 și alte metode, precum TOTP, pentru care pot fi interceptate datele de acces printr-o pagină malițioasă, pentru a fi reutilizate de un atacator, chiar și în fereastra de 30 de secunde specifică introducerii codului.

6.1.5. Interceptarea traficului prin MitM

Pentru evaluarea gradului de expunere a informațiilor de autentificare în lipsa protecției oferite de protocolul HTTPS, a fost realizată o simulare controlată a unui atac de tip Man-in-the-Middle (MitM). Scopul experimentului nu a fost compromiterea unui sistem real, a fost vizată vizualizarea informațiilor ce pot fi intercepate de un atacator aflat între client și server atunci când traficul este transmis prin protocol HTTP.

Un atac de tip Man-in-the-Middle presupune poziționarea unui intermediar între două entități care comunică. Acesta poate observa, modifica sau retransmite datele trimise între cele două părți fără știința acestora. În practică, astfel de atacuri sunt întâlnite în rețele nesecurizate sau în infrastructuri compromise.

Experimentul a fost construit cu ajutorul conceptului de proxy. Un proxy reprezintă un serviciu intermediar care primește cereri de la client și le retransmite către destinația finală. Din perspectiva clientului, proxy-ul apare ca fiind serverul cu care comunică, iar din perspectiva serverului acesta apare ca fiind clientul care inițiază conexiunea. Proxy-urile sunt utilizate pentru monitorizare, caching, filtrarea traficului sau controlul accesului. Un reverse proxy, spre deosebire de un proxy clasic, care acționează în numele clientului, acționează în numele serverului. Clientul comunică exclusiv cu reverse proxy-ul, iar acesta distribuie ulterior cererile către unul sau mai multe servere interne. Acest model este utilizat frecvent pentru echilibrarea încărcării, terminarea conexiunilor TLS și protecția infrastructurii backend. În cadrul experimentului prezentat a fost utilizat un proxy interceptiv bazat pe *mitmproxy*, poziționat între browser și aplicația web analizată.

Pentru a evidenția rolul protocolului HTTPS, demonstrația a fost realizată utilizând HTTP. Alegerea nu reflectă o configurație recomandată, pentru că urmărește evidențierea diferenței dintre traficul protejat și cel transmis în clar. Fără criptarea TLS, informațiile trimise între client și server pot fi obținute de orice entitate care are posibilitatea de a intercepta comunicația. Astfel, adresele de e-mail, parolele, token-urile de acces și răspunsurile mecanismelor MFA devin vizibile în conținutul pachetelor transmise în rețea.

Într-un scenariu de atac real, atacatorul nu cunoaște structura aplicației. Pentru a fi posibilă interceptarea exactă a datelor de interes, procesul începe cu o etapă de observare, sau de recunoaștere, în cadrul căreia sunt identificate endpoint-urile relevante și formatul mesajelor schimbate între client și server. Această etapă include observarea traficului, analiza documentației publice, inspectarea codului JavaScript transmis către browser sau examinarea răspunsurilor API. În urma analizei, atacatorul poate determina endpointul care gestionează autentificarea, câmpurile utilizate pentru transmiterea credențialelor și mecanisme MFA implementate de aplicație.

Scenariul considerat pornește de la premisa că etapă de recunoaștere a fost realizată cu succes. Astfel, scriptul de interceptare folosește reguli specifice pentru endpoint-urile de autentificare și verificare MFA identificate. Această abordare reflectă modul în care un atacator adaptează metodele utilizate la o aplicație cunoscută, nu este realizat un interceptor universal pentru orice aplicație web.

Implementarea experimentului a fost realizată prin dezvoltarea unui *addon* pentru mitmproxy. Acesta inspectează atât cererile HTTP transmise de client, cât și răspunsurile returnate de server. Pentru fiecare tip de operație au fost definite reguli de procesare diferite.

Pentru autentificarea bazată pe e-mail și parolă, scriptul monitorizează cererile de autentificare și extrage câmpurile care conțin credențialele utilizatorului. Informațiile identificate sunt salvate în memorie și afișate în jurnalul aplicației. Rezultatul prezentat în Figura 6.1 arată că, în lipsa protecției oferite de HTTPS, credențialele sunt vizibile în format text. De asemenea, în partea dreaptă se observă traficul interceptat pentru fiecare endpoint.

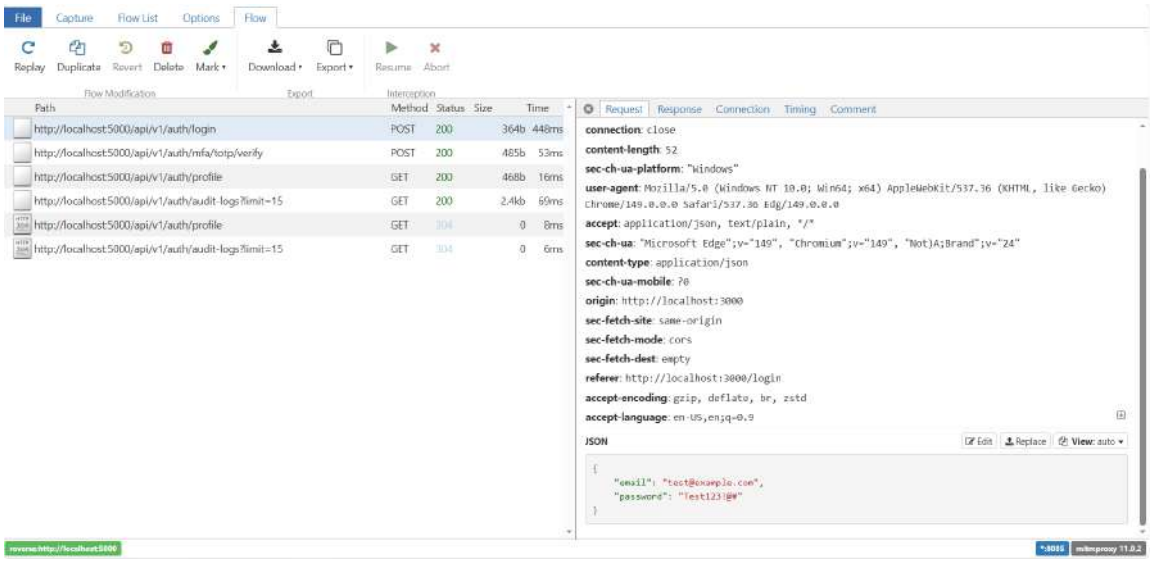

Figura 6.1: Interceptarea credențialelor

În ceea ce privește autentificarea multifactor bazată pe coduri TOTP, este interceptată cererea de verificare MFA și este extras atât codul TOTP introdus de utilizator, cât și token-ul temporar asociat sesiunii de autentificare. Se demonstrează, după cum se vede în Figura 6.2, că prezența unui al doilea factor de autentificare nu elimină riscul interceptării atunci când comunicația nu este protejată.

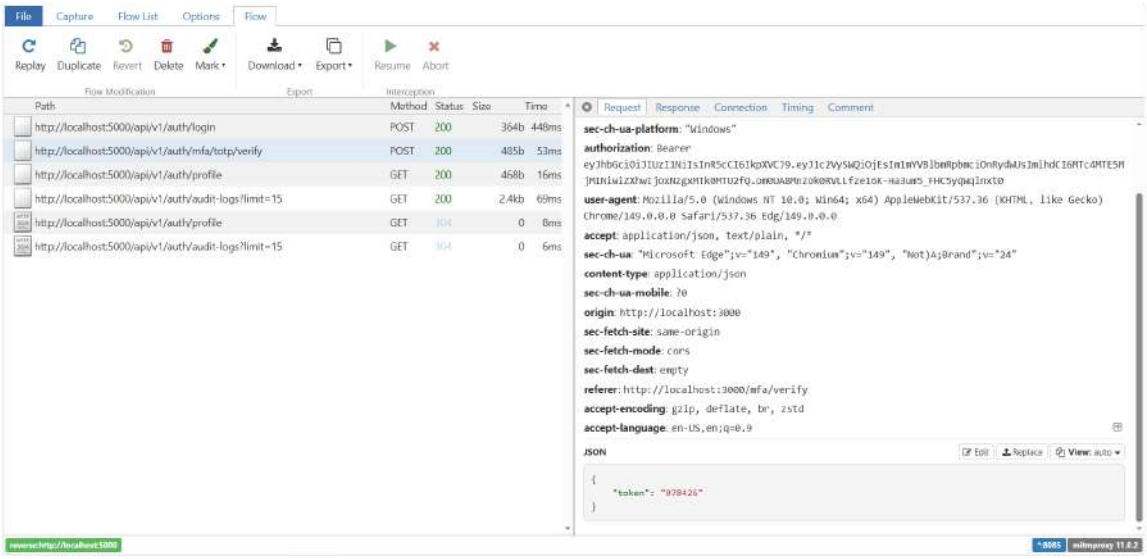

Figura 6.2: Interceptarea codului TOTP

Pentru a evidenția importanța mecanismelor anti-replay, scriptul include o componentă care încearcă reutilizarea codului TOTP. În cazul unei implementări corecte, serverul trebuie să respingă reutilizarea codului după validarea inițială. În funcție de implementarea serverului, răspunsul poate indica expirarea codului, invalidarea acestuia după prima utilizare sau activarea mecanismelor de limitare a numărului de încercări.

Spre deosebire de autentificarea bazată pe parole și coduri TOTP, FIDO2/WebAuthn utilizează perechi de chei criptografice și *challenge-uri* generate pentru fiecare sesiune. Scriptul interceptează mesajele schimbate între browser și server (Figura 6.3) și afișează informații

precum challenge-ul transmis de server (Figura 6.4), câmpurile din obiectul *clientDataJSON*, datele autentificatorului și semnătura digitală generată în timpul autentificării (Figura 6.5).

În cazul autentificării FIDO2, a fost utilizat un cont configurat pentru autentificare prin intermediul furnizorului de identitate Google, utilizând protocolul OpenID Connect (OIDC). În captura de trafic din Figura 6.3 poate fi observată cererea către endpoint-ul */auth/google*, care reprezintă etapa de redirecționare a utilizatorului către furnizorul extern de identitate. Figura evidențiază, de asemenea, alte evenimente observate în timpul sesiunii de testare, precum o încercare eșuată de autentificare (401 Unauthorized) și operația de delogare a utilizatorului (/auth/logout), ambele fiind înregistrate de proxy alături de fluxul de autentificare FIDO2.

Path	Method	Status	Size	Time
http://localhost:5000/api/v1/auth/logout	POST	200	50b	552ms
http://localhost:5000/api/v1/auth/login	POST	401	121b	80ms
http://localhost:5000/api/v1/auth/google	GET	302	342b	409ms
http://localhost:5000/api/v1/auth/mfa/fido2/options	GET	200	229b	48ms
http://localhost:5000/api/v1/auth/mfa/fido2/verify	POST	200	1.1kb	808ms
http://localhost:5000/api/v1/auth/profile	GET	200	485b	91ms
http://localhost:5000/api/v1/auth/audit-logs?limit=15	GET	200	2.4kb	211ms
http://localhost:5000/api/v1/auth/profile	GET	304	0	24ms
http://localhost:5000/api/v1/auth/audit-logs?limit=15	GET	304	0	26ms

Figura 6.3: Interceptarea datelor pentru FIDO2

Request | Response | Connection | Timing | Comment

X-RateLimit-Limit: 5
X-RateLimit-Remaining: 3
Date: Thu, 11 Jun 2026 20:01:56 GMT
X-RateLimit-Reset: 1781208946
Content-Type: application/json; charset=utf-8
Content-Length: 229
ETag: W/"e5-kAjSByysLShc1MJW7m17kGhIMxs"
Connection: close

JSON — Edit | Replace | View: auto


```
{
    "success": true,
    "options": {
        "challenge": "im6BMovOiov7y5fPtskwSscQtTLe_JFYIzbDucjG9sw",
        "allowCredentials": [
            {
                "id": "wz6MjUg9M0KWeB82odMccInTArw",
                "type": "public-key"
            }
        ],
        "timeout": 60000,
        "userVerification": "required",
        "rpId": "localhost"
    }
}
```


Figura 6.4: Challenge-ul FIDO2 transmis de server

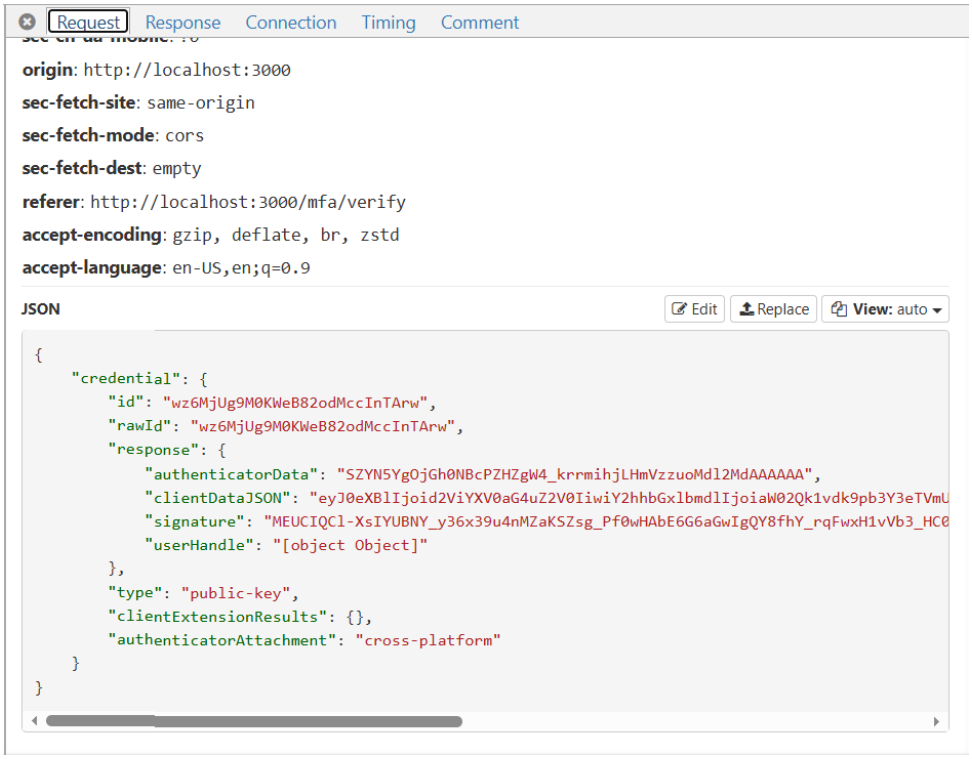

Figura 6.5: Datele WebAuthn interceptate de proxy

Cu toate acestea, informațiile obținute nu permit generarea unei noi autentificări valide. Challenge-ul este unic pentru fiecare sesiune, semnătura digitală este legată criptografic de *challenge* și de domeniul aplicației, iar cheia privată necesară generării semnăturii nu părăsește dispozitivul utilizatorului. În consecință, datele observate în trafic nu pot fi reutilizate pentru autentificări ulterioare.

În plus, răspunsul serverului conține tokenul de acces și tokenul de reîmprospătare, care sunt de asemenea interceptate și afișate în acest caz în secțiunea dedicată, respectiv „Response” (Figura 6.6).

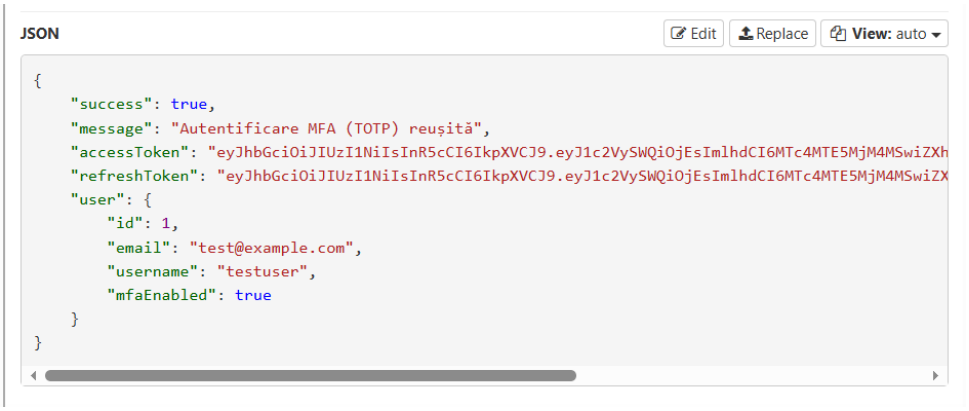

Figura 6.6: Interceptarea token-urilor

Experimentul evidențiază diferențele dintre mecanismele moderne de autentificare și soluțiile tradiționale bazate exclusiv pe parole sau coduri. În cazul comunicațiilor HTTP, informațiile de autentificare sunt expuse și pot fi interceptate de atacatori. Utilizarea HTTPS elimină această vulnerabilitate prin criptarea traficului între client și server. În plus, mecanismele FIDO2/WebAuthn oferă protecție suplimentară împotriva atacurilor de tip phishing și Man-in-the-Middle, deoarece autentificarea se bazează pe dovezi criptografice generate local de autentificator, nu pe secrete reutilizabile transmise prin rețea.

6.2. Evaluarea performanței și a latenței

Evaluarea mecanismelor de autentificare nu se limitează la rezistența în fața atacurilor. Următorul pas este analiza timpului suplimantar introdus de metodele MFA în procesul de autentificare. Dacă timpul de raspuns creste sesizabil, utilizatorul poate percepe mecanismul de securitate drept un impediment, fără a lua în considerare beneficiile din punct de vedere al securității. În acest scop, au fost măsurate separat operațiile implicate în procesul de autentificare.

Măsurătorile au fost realizate cu ajutorul unui script Python. Scriptul trimite cereri HTTP către backend și măsoară durata fiecărei operații cu *time.perf_counter()*. Pentru fiecare cerere, momentul inițial este salvat înainte de transmiterea *request-ului*, iar momentul final este înregistrat după primirea răspunsului de la server. Diferența dintre cele două valori este convertită în milisecunde și adăugată în lista de rezultate a testului respectiv.

Implementarea folosește funcții separate pentru cererile de tip POST și GET. Corpul cererilor este serializat în format JSON, iar pentru endpoint-urile protejate este adăugat antetul Authorization, în format *Bearer* token.

Scriptul calculează pentru fiecare set de măsurători media, mediana, valoarea minimă, valoarea maximă, percentila 95 și abaterea standard. Acești indicatori permit compararea comportamentului mediu, dar și observarea variațiilor.

Media reprezintă costul mediu al procesului și este utilă pentru comparația directă între metode, însă este sensibilă în cazul valorilor extreme. Mediana indică valoarea de mijloc, iar comparația dintre medie și mediană arată gradul de asimetrie al distribuției, întrucât o medie mai mare decât mediana semnalează existența unor cereri mai lente, care ridică valoarea medie. Minimul și maximul delimitează cazul cel mai favorabil și cel mai nefavorabil observat, oferind intervalul în care variază timpul de răspuns. Percentila 95 reprezintă pragul sub care se încadrează 95% dintre cereri, fiind un indicator des utilizat în evaluarea performanței sistemelor. Abaterea standard măsoară dispersia valorilor în jurul mediei, o valoare mică indicând un comportament stabil și predictibil.

Au fost definite două adrese de testare. Prima adresă accesează backend-ul direct. A doua adresă accesează backend-ul prin mitmproxy. Această împărțire permite măsurarea separată a latenței aplicației și a latenței introduse de proxy-ul folosit pentru atacul Man-in-the-Middle.

Primul scenariu considerat măsoară autentificarea fără MFA, folosit pentru comparație. În acest caz, serverul validează parola și returnează direct tokenurile de sesiune. Ulterior, se măsoară primul pas al autentificării cu MFA, în care serverul validează parola și returnează un tempToken, fără să se finalizeze autentificarea. Este măsurat și timpul de verificare al tokenului JWT pentru cererile protejate, prin accesarea endpoint-ului */auth/profile*.

Pentru TOTP, scriptul măsoară latența endpoint-ului de verificare folosind coduri invalide. Scopul nu este autentificarea cu succes. Este urmărită măsurarea timpului necesar serverului pentru procesarea unei cereri TOTP. Testul este limitat la patru iterații pentru a nu activa mecanismul de *rate limiting*. Pentru FIDO2, scriptul construiește o cerere de test, cu valori

generate aleator. Serverul procesează structura primită și returnează un răspuns de eroare. Este surprins costul din partea serverului pentru procesarea cererii FIDO2, nu și timpul în care utilizatorul își folosește dispozitivul personal pentru autentificare.

Rezultatele obținute sunt prezentate în Tabelul 6.1.

Proces	**Medie (ms)**	**Mediană (ms)**	**Minim (ms)**	**Maxim (ms)**	**P95 (ms)**	**Abatere standard (ms)**
Autentificare fără MFA	343,1	327,8	272,5	511,6	407,7	69,5
Autentificare cu MFA, pasul 1 (parola)	320,9	326,9	296,3	332,1	331,9	13,2
Verificare JWT, per cerere	29,5	30,1	20,9	36,3	32,2	3,1
Verificare TOTP, la nivelul serverului	25,6	22,6	21,7	35,5	22,9	6,6
Verificare FIDO2, la nivelul serverului	13,0	11,8	4,3	23,4	20,6	8,8

Tabelul 6.1: Latența principalelor procese de autentificare

Autentificarea fără MFA ajunge la un timp mediu de 343,1 ms. Acesta include validarea parolei și generarea tokenurilor de sesiune. Verificarea parolei în cadrul autentificării cu MFA are o medie de 320,9 ms. Valoarea în al doilea caz este mai mică deoarece autentificarea nu este finalizată în acest pas, iar serverul returnează doar un token temporar pentru a fi utilizat în etapa a doua.

Verificarea TOTP are o durată medie de 25,6 ms. Această valoare indică un cost redus. În utilizarea efectivă, timpul perceput de utilizator este mai mare, deoarece include deschiderea aplicației de autentificare, citirea codului și introducerea acestuia.

Pentru FIDO2, timpul mediu este de 13,0 ms. Rezultatul nu include timpul necesar utilizatorului pentru confirmarea autentificării pe dispozitiv, doar procesarea efectuată de server după primirea răspunsului WebAuthn. Valoarea arată că verificarea FIDO2 nu introduce un cost ridicat.

Verificarea JWT are o durată medie de 29,5 ms pentru fiecare cerere protejată. Această operație este relevantă, deoarece fiecare endpoint protejat trebuie să valideze tokenul primit înainte de returnarea datelor către client.

Pentru evaluarea mediului folosit în simularea atacului MitM, a fost comparată autentificarea directă către backend cea realizată prin mitmproxy. Rezultatele sunt prezentate în Tabelul 6.2.

Scenariu	**Medie (ms)**	**Mediană (ms)**	**Minim (ms)**	**Maxim (ms)**	**P95 (ms)**	**Abatere standard (ms)**
Autentificare direct către backend	317,6	319,4	309,4	328,9	319,8	7,9
Autentificare prin proxy	324,7	325,5	312,1	334,8	328,0	8,3

Tabelul 6.2: Impactul proxy-ului de interceptare asupra latenței

Compararea celor două scenarii arată că proxy-ul a introdus un timp suplimentar mediu de **7,1 ms**, valoare redusă în raport cu durata totală a autentificării. Prin urmare, proxy-ul folosit în experimentul MitM nu modifică semnificativ timpul de răspuns al aplicației.

Pentru a estima timpul total de autentificare în cazul MFA, au fost însumate duratele corespunzătoare primului pas, validarea parolei, și al doilea pas, verificarea TOTP. Rezultatul a fost comparat cu autentificarea fără MFA și este prezentat în Tabelul 6.3.

Scenariu	**Timp mediu (ms)**
Autentificare fără MFA	343,1
Autentificare cu MFA (parolă + TOTP)	346,5
Timp suplimentar introdus de MFA	3,4
Timp suplimentar JWT	29,5

Tabelul 6.3: Estimarea timpului suplimentar introdus de MFA

Astfel, diferența de timp adăugată de MFA este de doar 3,4 ms la nivelul serverului. Această valoare arată că verificarea factorului suplimentar nu este cauza principală a întârzierii percepute de utilizator. Timpul suplimentar apare din acțiunile manuale ale acestuia.

Sunt important de menționat și limitele întâlnite. Testele au fost executate într-un mediu local, nu într-o infrastructură distribuită. De asemenea, pentru FIDO2 a fost măsurată doar procesarea la nivelul serverului, nu întregul flux, dar rezultatele sunt suficiente pentru comparația mecanismelor implementate.

Rezultatele arată că introducerea MFA nu afectează semnificativ timpul de răspuns al serverului. Diferențele de performanță sunt mici, în timp ce beneficiile de securitate aduse în fața atacurilor sunt semnificative. Astfel, folosirea autentificării multifactor este justificată prin creșterea nivelului de protecție, fără un cost substanțial asupra latenței aplicației.

6.3. Impactul asupra experienței utilizatorului

Alegerea unei metode de autentificare depinde atât de nivelul de securitate oferit, cât și de gradul de acceptare din partea utilizatorilor. Un sistem dificil de utilizat poate duce la evitarea funcționalităților de securitate sau la apariția unor practici nesigure. De asemenea, un motiv pentru care metodele MFA sunt văzute ca un impediment este datorat faptului că nu le este înțeleasă utilitatea. În acest scop, în cadrul sistemului propus, utilizatorul primește o descriere a beneficiilor de securitate, astfel încât alegerea dintre TOTP și FIDO2 este informată, nu impusă.

Evaluarea de față se bazează pe observarea proiectării interfeței și a fluxului de autentificare, nu pe un studiu formal cu participanți, însă concluziile sunt în acord cu rezultatele din literatura de specialitate. Studiile discutate anterior au arătat că dificultățile percepute de utilizatori provin în principal din etapa de configurare și din riscul de blocare a accesului, nu din folosirea curentă a metodei. Această observație se reflectă și în sistemul propus, în care efortul suplimentar este concentrat în etapa inițială de înrolare, în timp ce autentificările ulterioare presupun un număr redus de acțiuni.

Autentificarea prin parolă este cea mai cunoscută metodă de majoritatea utilizatorilor. Cu toate acestea, simplitatea oferită este însoțită de riscurile prezentate anterior. Utilizatorii aleg parole ușor de memorat, reutilizează aceleași credențiale pentru mai multe platforme sau le notează pe suport de hartie, astfel reducând nivelul de securitate.

Metoda TOTP introduce un compromis între securitate și utilizabilitate. Utilizatorul trebuie să dețină un dispozitiv suplimentar și să introducă manual codul generat. Procesul este simplu, însă poate deveni incomod în situațiile în care dispozitivul nu este disponibil sau când codul expiră înainte de finalizarea autentificării.

În cazul autentificării prin FIDO2, autentificarea se realizează prin confirmarea identității pe dispozitivul asociat, utilizând mecanisme precum amprenta, recunoașterea facială sau codul de

deblocare al dispozitivului. Din perspectiva utilizatorului, procesul este mai rapid și presupune mai puține acțiuni decât introducerea unui cod TOTP. Totodată, măsurătorile din secțiunea 6.2 confirmă că diferența de timp la nivelul serverului este neglijabilă, verificarea FIDO2 fiind chiar mai rapidă decât cea a codului TOTP. O limitare observată este etapa inițială de configurare, deoarece utilizatorii care nu sunt familiarizați cu conceptele FIDO2 sau cu cheile de acces pot considera procesul mai complex decât configurarea unei aplicații TOTP. La aceasta se adaugă fragmentarea suportului între platforme și browsere, care poate influența experiența de configurare în funcție de sistemul de operare folosit.

Un aspect important al experienței utilizatorului este recuperarea accesului în situațiile în care factorul suplimentar nu este disponibil, de exemplu la pierderea dispozitivului sau la expirarea codului. Pentru a reduce riscul de blocare, sistemul oferă coduri de rezervă, generate la configurarea metodelor MFA, care permit autentificarea atunci când metoda principală nu poate fi folosită. Prezența unei astfel de soluții diminuează reticența utilizatorilor față de activarea autentificării multifactor.

Analiza arată că FIDO2 oferă cel mai bun echilibru între securitate și experiența utilizatorului. Metoda elimină necesitatea transmiterii secretelor partajate în rețea, fără a introduce întârzieri semnificative în procesul de autentificare.

7. CONCLUZII ȘI DIRECȚII VIITOARE

7.1. Concluziile rezultatelor experimentale și contribuțiile aduse

Creșterea numărului de incidente de securitate care au la bază compromiterea identității digitale a determinat reevaluarea mecanismelor tradiționale de autentificare. Parolele rămân metoda cel mai des utilizată, însă eficiența lor este limitată de comportamentul utilizatorilor și de varietatea atacurilor disponibile în prezent, precum phishing, brute force sau programe de tip infostealer. În acest context, autentificarea multifactor este una dintre principalele măsuri de reducere a riscului asociat accesului neautorizat, iar trecerea către factori rezistenți la phishing este încurajată de standardele NIST SP 800-63B-4 și de cadrul legislativ introdus prin OUG 155/2024.

Obiectivul lucrării a fost dezvoltarea și evaluarea unui ecosistem de autentificare multifactor care depășește limitele metodelor cu un singur factor și care este rezilient în fața vectorilor de atac moderni. Obiectivul a fost atins printr-o aplicație web funcțională, dezvoltată cu Node.js, React.js și PostgreSQL, care integrează TOTP, FIDO2/WebAuthn și autentificare federată prin OpenID Connect. SMS OTP a fost exclus din implementare conform recomandărilor NIST, din cauza vulnerabilităților la SIM swapping și la interceptare. Aplicația a constituit ulterior infrastructura experimentală prin care a fost măsurat nivelul de protecție oferit de fiecare metodă.

Sistemul îndeplinește cele trei proprietăți de securitate derivate din triada CIA. Confidențialitatea este asigurată prin stocarea parolelor cu bcrypt și prin păstrarea cheilor private FIDO2 sau a secretului TOTP exclusiv pe dispozitivul utilizatorului. Integritatea sesiunii este verificată la fiecare cerere prin semnătura JWT. Disponibilitatea controlată provine din limitarea numărului de autentificări care blochează abuzul fără a refuza utilizatorii legitimi, datorită contorizării exclusive a încercărilor eșuate.

Tabelul 7.1 reunește vectorii evaluați și concluziile evaluării.

Rezultatele confirmă clasificarea descrisă în literatura de specialitate. Vectorii de atac evaluați au evidențiat o separare a caracteristicilor pentru doua categorii de metode. Atacurile prin *brute force*, asupra parolei și asupra codului TOTP, au fost oprite prin limitarea ratei, complexitatea parolei și reînnoirea periodică a codului.

Diferența a apărut între cele două metode multifactor. FIDO2 a respins toate tentativele de phishing și a rămas inutilizabil pentru un atacator și atunci când traficul a fost interceptat, deoarece cheia privată nu părăsește dispozitivul, iar răspunsul este legat criptografic de domeniu și de un *challenge* unic. TOTP, în schimb, a oferit doar o protecție temporală. Codul a putut fi folosit de două ori în fereastra de validitate și a putut fi observat în clar în lipsa criptării TLS. Așadar, evaluarea arată că FIDO2 elimină clase întregi de atacuri prin construcția sa, în timp ce TOTP doar le îngreunează, distincție care corespunde nivelurilor de asigurare definite de NIST.

Vector de atac	Metoda vizată	Rezultat observat	Concluzia evaluării
Brute force pe parolă	Parolă	Blocat la a 6-a încercare	Limitarea ratei și bcrypt opresc atacul online și offline
Brute force pe codul TOTP	TOTP	Blocat	Spațiul redus și expirarea la 30 s fac atacul nerealist
Reluarea codului TOTP	TOTP	Cod acceptat de două ori	Vulnerabilitatea este protecția doar temporală, fără invalidarea codului
Falsificarea domeniului FIDO2	FIDO2	4 din 4 domenii false respinse, semnătura falsă respinsă	Rezistență criptografică la phishing prin legarea de domeniu
Interceptare prin MitM (pentru HTTP)	Parolă, TOTP, FIDO2	Parola, codul TOTP și token-urile sunt vizibile și pot fi reutilizate; răspunsul FIDO2 deși vizibil, este inutilizabil	Secretele partajate sunt expuse fără TLS; FIDO2 este rezistent interceptării

Tabelul 7.1: Sinteza rezultatelor evaluării pe vectori de atac

Măsurătorile arată că introducerea autentificării multifactor nu afectează semnificativ timpul de răspuns al serverului. Cel mai mare cost al întregului proces revine autentificării cu parolă, care atinge în medie 343,1 ms, valoare dată în principal de algoritmul bcrypt cu 12 runde, ales astfel pentru a încetini atacurile asupra parolelor. Prin comparație, verificarea factorului suplimentar este rapidă, de 25,6 ms pentru TOTP și 13,0 ms pentru FIDO2 la nivelul serverului, astfel încât timpul suplimentar adăugat de MFA este de doar 3,4 ms, iar verificarea token-ului JWT, efectuată la fiecare cerere protejată, este de 29,5 ms. Proxy-ul folosit în simularea MitM a introdus la rândul său un cost redus, de 7,1 ms, ceea ce confirmă că rezultatele nu sunt influențate de mediul de testare. Așadar, întârzierea percepută de utilizator nu provine din procesarea serverului, provine din acțiunile manuale, precum deschiderea aplicației de autentificare, citirea codului sau confirmarea pe dispozitiv. Cu toate că măsurătorile au fost realizate local, iar pentru FIDO2 a fost evaluată doar procesarea de pe server, rezultatele sunt suficiente pentru compararea mecanismelor implementate și demonstrează că beneficiile de securitate aduse de autentificarea multifactor sunt obținute fără un cost real asupra performanței aplicației.

Contribuția principală adusă este evaluarea rezistenței metodelor MFA prin simularea a cinci vectori de atac. Rezultatul obținut accentuează diferența dintre rezistența criptografică a FIDO2 și rezistența temporală a TOTP.

A doua contribuție este dezvoltarea unui cadru de evaluare a rezilienței MFA, care leagă fiecare tip de atac de mecanismul de protecție corespunzător și de rezultatul observabil al sistemului. Cadrul poate fi reutilizat pentru orice sistem care combină factori bazați pe cunoaștere și pe posesie.

O altă contribuție este validarea recomandărilor NIST SP 800-63B-4 privind trecerea de la secrete partajate către autentificatori rezistenți la phishing. Sistemul propus alcătuiește infrastructura experimentală care a făcut posibilă evaluarea metodelor vizate.

7.2. Limitări și direcții viitoare

Concluziile prezentate sunt valabile în limitele mediului de testare folosit. Evaluarea s-a desfășurat într-un mediu de dezvoltare local, astfel încât rezultatele privind confidențialitatea presupun un mediu de producție care utilizează HTTPS.

De asemenea, scripturile de atac validează comportamentul mecanismelor de apărare în

condiții fixe, dar nu acoperă atacurile adaptive sau distribuite, precum un atac de tip *brute force* distribuit, care ar utiliza mai multe adrese IP, ocolind limitarea pentru o singură adresă.

În ceea ce privește măsurarea latenței, aceasta este realizată pentru un singur utilizator, într-un mediu local și nu trateză cazul în care mai mulți utilizatori s-ar autentifica simultan, prin metode diferite. Totodată, vectorii de atac testați nu acoperă complet suprafața de atac, întrucât există o mulltitudine de atacuri posibile, care au fost analizate doar teoretic, în capitolul 3.

Impactul asupra experienței utilizatorului a fost evaluat având în vedere modul de proiectare al interfeței, fără un studiu cu utilizatori reali, iar validarea FIDO2 a folosit fluxul WebAuthn standard, deoarece utilizarea Windows Hello nu a fost posibilă din cauza unor limitări tehnice.

O direcție viitoare a lucrării este remedierea vulenrabilității date de reutilizarea codurilor TOTP. Aceasta poate fi realizată prin păstrarea temporară a codurilor deja validate pentru fiecare utilizator în baza de date. Astfel, codul ar fi respins după prima utilizare, chiar dacă fereastra temporală nu a expirat.

Evaluarea performanței poate fi extinsă prin reluarea testelor pentru un număr variabil de utilizatori. Rezultatul ar fi observarea comportamentului endpoint-urilor TOTP, FIDO2 și JWT, dar și evaluarea latenței în condiții de producție.

O altă direcție este realizarea unui studiu de utilizabilitate. Pot fi măsurați timpul de configurare, timpul de autentificare și rata de eroare pentru TOTP și FIDO2. În acest mod, comparația dintre metode ar include efortul real al utilizatorului, nu doar securitatea.

De asemenea, pot fi testate scenarii de atac suplimentare, precum atacurile asupra fluxului de recuperare a contului și scenariile de MFA fatigue pentru metode bazate pe notificări push.

În ceea ce privește FIDO2, se pot testa modalitățile de autentificare neutilizate în cadrul proiectului, precum YubiKey, Windows Hello sau Touch ID. În plus, testarea a mai multor browsere și sisteme de operare ar permite evaluarea compatibilității WebAuthn între platforme.

Rezultatele susțin folosirea metodelor MFA. TOTP este o soluție accesibilă pentru conturi cu risc moderat, deoarece este ușor de implementat și compatibilă cu majoritatea aplicațiilor de autentificare. Pentru accesul la resurse critice, FIDO2/WebAuthn oferă un nivel mai ridicat de protecție, prin eliminarea secretelor partajate și prin legarea autentificării de dispozitiv și de domeniul aplicației.

BIBLIOGRAFIE

[1] Verizon Business, *2025 Data Breach Investigations Report*, Technical Report, Accessed: 2026-04-26, Verizon, 2025, Disponibil: https://www.verizon.com/business/resources/T16f/reports/2025-dbir-data-breach-investigations-report.pdf.

[2] Saeed Abbasi, *The Verizon 2025 Data Breach Investigations Report (DBIR): Six Trends You Can't Ignore*, Accessed: 2026-04-26, Qualys, Apr. 2025, Disponibil: https://blog.qualys.com/qualys-insights/2025/04/22/the-verizon-2025-data-breach-investigations-report-dbir-six-trends-you-cant-ignore.

[3] Directoratul Național de Securitate Cibernetică (DNSC), *Buletin de indicatori, statistici și tendințe de securitate cibernetică. H1 2025*, rap. teh., Accessed: 2026-04-26, DNSC, Aug. 2025, Disponibil: https://www.dnsc.ro/vezi/document/buletin-de-indicatori-statistici-si-tendinte-de-securitate-cibernetica-h1-2025.

[4] European Union Agency for Cybersecurity (ENISA), *ENISA Threat Landscape 2025*, rap. teh., Accessed: 2026-04-26, ENISA, Oct. 2025, DOI: 10.2824/2445233, Disponibil: https://www.enisa.europa.eu/sites/default/files/2025-11/ENISA%20Threat%20Landscape%202025.pdf.

[5] James Pengelly, Gareth Marchant, *The Official CompTIA Security+ Student Guide (Exam SY0-701)*, United States: Comptia, INC International Concepts, 2024.

[6] A. Caranica, *IAA și mijloace de securizare a rețelelor (guvernanța sistemelor)*, București, România: Curs Securitatea în Rețelele de Calculatoare, 2025.

[7] Littlejohn Shinder, Michael Cross, „Chapter 11 - Passwords, Vulnerabilities, and Exploits", în Littlejohn Shinder, Michael Cross (Eds.) *Scene of the Cybercrime (Second Edition)*, Syngress, Burlington, pp. 467–503 2008.

[8] P. Grassi, E. Newton, J. Fenton et al., *NIST Special Publication 800-63B-4: Digital Identity Guidelines - Authentication and Lifecycle Management*, National Institute of Standards și Technology, 2024, Disponibil: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63B-4.pdf.

[9] James Messer, *Professor Messer's SY0-701 CompTIA Security+ Course Notes*, United States: Messer Studios, LLC, 2023.

[10] Tianyu Bell Pan, Qiangeng Yang, Alexa Jordyn Cole, Ronald Wilson, Damon L Woodard, „Psychology of Phishing Emails: Quantifying Persuasion Principles and Simulating Detection with Large Language Models", *Expert Systems with Applications*, vol. nr. Elsevier, p. 131507 2026.

[11] Imperva, *What is a Man-in-the-Middle Attack? MitM Types & Prevention*, Accessed: 2026-04-27, Imperva, n.d. Disponibil: https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/.

[12] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen, Yevgeni Koucheryavy, „Multi-Factor Authentication: A Survey", *Cryptography*, vol. 2, nr. Ian. 2018.

[13] D. Burileanu, *Tehnologii biometrice. Analiza semnalului vocal și a semnalelor biologice*, București, România: Curs Tehnologii Biometrice, 2025.

[14] Fraud.com, *Biometric Authentication*, Accessed: 2026-05-03, Fraud.com, n.d. Disponibil: `https://www.fraud.com/post/biometric-authentication`.

[15] Sheetal Chaudhary, Rajender Nath, „A multimodal biometric recognition system based on fusion of palmprint, fingerprint and face", *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 596–600 2009.

[16] Paul A. Grassi, James L. Fenton, Elaine M. Newton, et al., *Digital Identity Guidelines: Authentication and Lifecycle Management*, NIST Special Publication 800-63B Revision 4, Accessed: 2026-04-27, National Institute of Standards și Technology (NIST), 2024, DOI: 10.6028/NIST.SP.800-63B-4, Disponibil: `https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63B-4.pdf`.

[17] Alex Brown, *WebAuthn vs FIDO2: Understanding the Differences*, Accessed: 2026-05-05, Descope, Sept. 2024, Disponibil: `https://www.descope.com/blog/post/webauthn-vs-fido2`.

[18] Anna Angelogianni, Ilias Politis, Christos Xenakis, „How Many FIDO Protocols Are Needed? Surveying the Design, Security and Market Perspectives", *ACM Computing Surveys*, vol. nr. Association for Computing Machinery 2021.

[19] Terrazone, *SS7 Security Vulnerabilities, Attacks, and Prevention*, Accessed: 2026-04-30, Terrazone, n.d. Disponibil: `https://terrazone.io/ss7-security-vulnerabilities-attacks-prevention/`.

[20] Lucie Cardiet, *The Vulnerabilities of SMS Two-Factor Authentication*, Accessed: 2026-04-30, Vectra AI, Ian. 2024, Disponibil: `https://www.vectra.ai/blog/the-hidden-risks-of-sms-based-multi-factor-authentication`.

[21] Trout Team, *Top 5 MFA Methods Compared: SMS, TOTP, Biometrics, Hardware Keys & Push Notifications*, Accessed: 2026-04-30, Trout Software, Feb. 2026, Disponibil: `https://www.trout.software/blog/top-5-mfa-methods-compared-sms-totp-biometrics-hardware-keys-and-push-notifications`.

[22] Elísabet Líf Birgisdóttir, Michał Ignacy Kunkel, Lukáš Pleva, Maria Papaioannou, Gaurav Choudhary, Nicola Dragoni, „Exploring the Security of Mobile Face Recognition: Attacks, Defenses, and Future Directions", *Applied Sciences*, vol. 15, nr. 24, MDPI, p. 13232 2025.

[23] I Wayan Widi Pradnyana, Kayla Alisha Rania, Fendi Permadi, Rizky Maradika Saputra, „Biometric Authentication Systems in the Deepfake Era: A Systematic Review of ISO 15408 Compliance and Presentation Attack Mitigation", *2025 International Conference on ICT for Smart Society (ICISS)*, pp. 1–6 2025.

[24] Aditi Roy, Nasir Memon, Arun Ross, „Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems", *IEEE Transactions on Information Forensics and Security*, vol. 12, nr. 9, IEEE, pp. 2013–2025 2017.

[25] Man Zhou, Shuao Su, Qian Wang, Qi Li, Yuting Zhou, Xiaojing Ma, Zhengxiong Li, „PrintListener: Uncovering the vulnerability of fingerprint authentication via the finger friction sound", *arXiv preprint arXiv:2404.09214*, vol. nr. 2024.

[26] Daniyar Kurmankhojayev, Andrei Shadrikov, Dmitrii Gordin, Mikhail Shkorin, Danijar Gabdullin, Aigerim Kambetbayeva, Kanat Kuatov, „Virtual camera detection: Catching video injection attacks in remote biometric systems", *arXiv preprint arXiv:2512.10653*, vol. nr. 2025.

[27] Brendan Klare, *Next-Gen Liveness Detection for Deepfake and Injection Attacks*, Accessed: 2026-05-02, ROC, Mai 2025, Disponibil: `https://roc.ai/2025/05/13/next-gen-liveness-detection-for-deepfake-and-injection-attacks/`.

[28] Imen Rjab, Layth Sliman, „Survey on biometric authentication for decentralized identity management: trends, challenges, and future directions", *Future Internet*, vol. 18, nr. 3, MDPI, p. 126 2026.

[29] Josh Amishav, *MFA Bypass Attacks: How Attackers Get Past Multi-Factor Authentication*, Accessed: 2026-05-03, Breachsense, Apr. 2026, Disponibil: `https://www.breachsense.com/blog/mfa-bypass-attacks/`.

[30] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, Frank Stajano, „The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes", *2012 IEEE Symposium on Security and Privacy*, IEEE, pp. 553–567 2012.

[31] Kevin Lee, Ben Kaiser, Jonathan Mayer, Arvind Narayanan, „An Empirical Study of Wireless Carrier Authentication for SIM Swaps", *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*, USENIX Association, pp. 61–80 2020.

[32] Leona Lassak, Annika Hildebrandt, Maximilian Golla, Blase Ur, „It's Stored, Hopefully, on an Encrypted Server: Mitigating Users' Misconceptions About FIDO2 Biometric WebAuthn", *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, pp. 91–108 2021.

[33] Juan Lang, Alexei Czeskis, Dirk Balfanz, Marius Schilder, Sampath Srinivas, „Security Keys: Practical Cryptographic Second Factors for the Modern Web", *Financial Cryptography and Data Security (FC 2016)* Lecture Notes in Computer Science, vol. 9603, Springer, pp. 422–440 2016.

[34] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, Kent Seamons, „A Usability Study of Five Two-Factor Authentication Methods", *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, USENIX Association, pp. 357–370 2019.

[35] Christian Catalano, Andrea Chezzi, Vita Santa Barletta, Franco Tommasi, „Defeating FIDO2/CTAP2/WebAuthn using browser in the middle and reflected cross site scripting", *Journal of Computer Virology and Hacking Techniques*, vol. 21, nr. 1, p. 11 2025.

[36] Harshavardhan Ravilla, Rishi Sayal, Pooja Kulkarni, „Study and Analysis of FIDO2 Passwordless Web Authentication", *International Conference on Advances in Computational Intelligence and Informatics*, pp. 375–386 2023.

[37] Martiño Rivera-Dourado, Marcos Gestal, Alejandro Pazos, José M. Vázquez-Naya, „An Analysis of the Current Implementations Based on the WebAuthn and FIDO Authentication Standards", *Engineering Proceedings (4th XoveTIC Conference)*, vol. 7, MDPI, p. 56 2021.

[38] Badal Bhushan, „Bridging Identity Assurance Gaps: Integrating FIDO2 and Certificate-Based Authentication for Phishing-Resistant, Scalable Enterprise Security", *International Journal of Data Science and Machine Learning*, vol. 5, nr. 2, pp. 9–24 2025.

[39] Harold Ramcharan, „The Effective Integration of Multi-Factor Authentication (MFA) with Zero Trust Security", *American Journal of Mathematical and Computer Modelling*, vol. 10, nr. 1, pp. 1–5 2025.

[40] Revathi Kantipudi, Ajay Sai Kumar Mallavarapu, Shinu M Rajagopal, Madhav Kagolanu, „A comprehensive analysis on using multifactor authentication system for three level security", *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pp. 498–503 2024.

A. ANEXA 1

Codul pentru sistemul de autentificare, alcătuit din interfața utilizatorului, server, baza de date, dar și scripturile pentru scenariile de atac și calculul latenței se pot accesa la următoarea adresă:

`https://github.com/dianaabratu/SistemMFA`.