Università di Pisa

PENETRATION TESTING ON 5G CORE WEB TECHNOLOGIES: OPEN5GS CASE STUDY

Based on paper: Penetration Testing of 5G Core Network Web Technologies

CONTI SIMONE, MAT. 675682
COPELLI FRANCESCO, MAT 675686
LEPORE NICOLA, MAT. 678038

PAPER PRESENTATION

AIM

Use penetration testing to identify security gaps in the 5G core and their network implications.

Objective

Analyze key protocols, simulate attacks, and propose countermeasures to mitigate risks in 5G networks.

ATTACKS IN DETAILS

01 ✓	02 ✗	03 ✓	04/05 ✓✓
Database permission leakage	SQL Injection	NoSQL Injection	Dictionary / Bruteforce Attack

ACTUALLY...

ATTACKS IN DETAILS

06 ✗

(D)DoS Attack

07 ✗

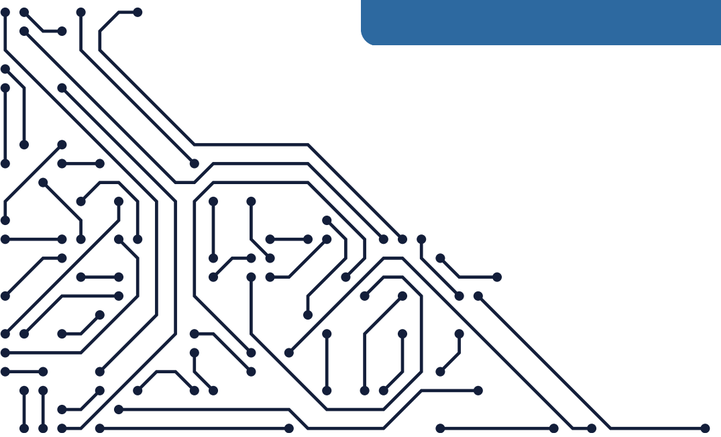
Directory traversal

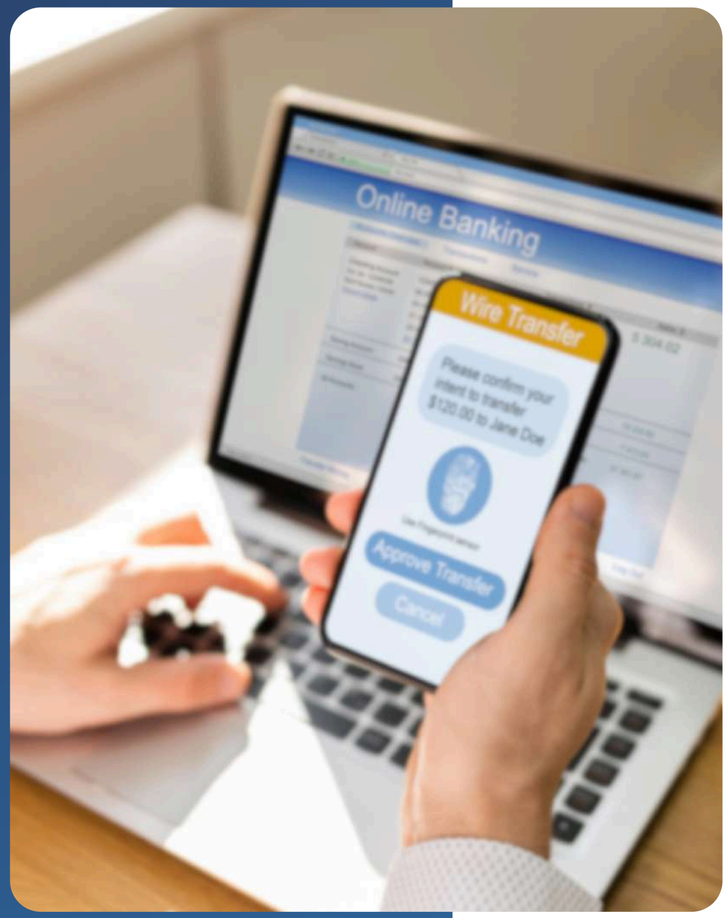
08 ✗

Clickjacking

09 ✓

JWT Robustness





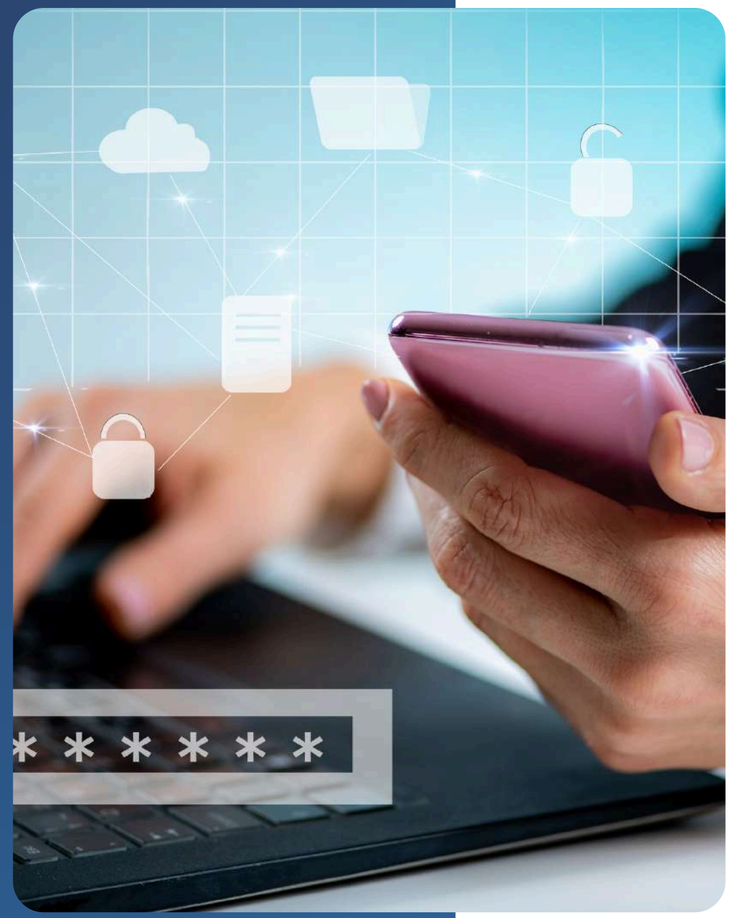
MONGODB ACCESS

- **Authentication missing**

  No password asked when authenticating on MongoDB

- **Permission**

  Everyone is allowed to view and/or edit database entries



COOKIE(S)

- **connect.sid**
  - s:<session_id>.<signature>
  - shared secret ‘change-me’
  - hmac signature sha256
- **csrf**
  - No integrity checks between session and user
  - Anyone can ask for a valid csrf

JSON WEB TOKEN (JWT) ATTACKS

Attack	Description	Relevance for Open5GS
Brute-force of the secret	If the secret used to sign the JWT is weak or known (e.g., 'change-me'), the attacker can guess the secret and sign new valid tokens.	✅ Weak secret ➔ forged admin token.
Algorithm Confusion Attack	The server accepts the algorithm (alg) value specified in the token. An attacker can change alg from HS256 to none and bypass the signature.	⚠ We don't know if Open5GS is vulnerable, but it would be a threat if the server doesn't properly verify alg.
Token Replay Attack	Reuse a valid stolen token, e.g., sniffed over an insecure network, to authenticate elsewhere.	🟠 Low risk, but possible if sessions don't expire quickly.
Token Manipulation (tampering)	If the server doesn't properly verify the signature, an attacker can change the payload (e.g., roles) without recalculating the signature.	❌ Not relevant if the signature is always correctly checked.
Long-Lived Tokens	Token with exp (expiration) too long ➔ even if stolen or forged, it remains valid for a long time.	✅ Mitigation possible: short-lived JWT.
Hardcoded Keys	The secret is hardcoded in the code or easily recoverable	✅ 'change-me' secret known.

LOCAL STORAGE

Valid CSRF

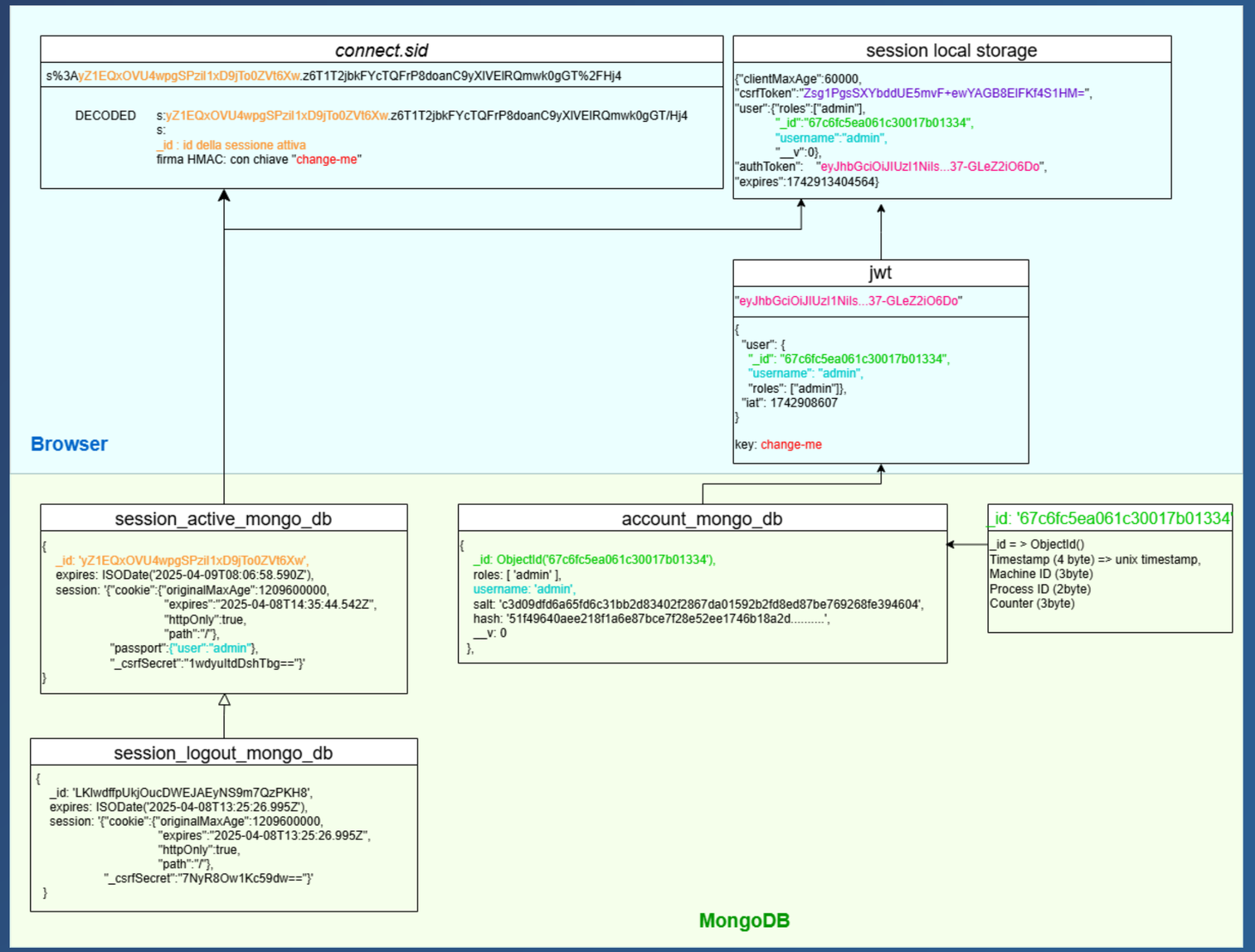
Same issue as explained before

Valid JWT token

Forged from leaked data from db thanks to known secret key

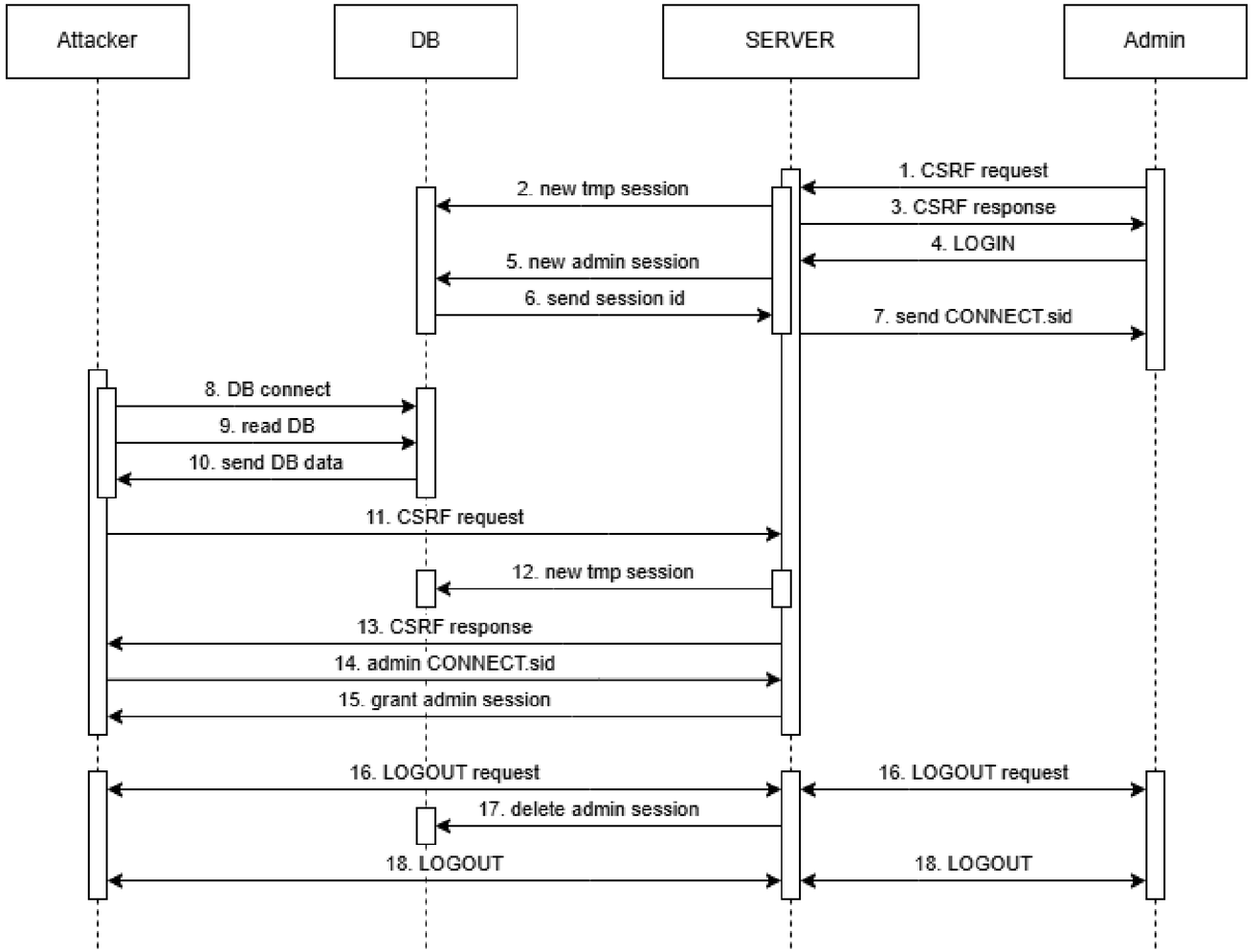
User Data

Retrieve a valid user with proper authorization from DB



INFORMATION DISTRIBUTION

ATTACK FLOW


Attacker
DB
SERVER
Admin
1. CSRF request
2. new tmp session
3. CSRF response
4. LOGIN
5. new admin session
6. send session id
7. send CONNECT.sid
8. DB connect
9. read DB
10. send DB data
11. CSRF request
12. new tmp session
13. CSRF response
14. admin CONNECT.sid
15. grant admin session
16. LOGOUT request
16. LOGOUT request
17. delete admin session
18. LOGOUT
18. LOGOUT

HOW TO PREVENT ATTACKS?

1. Change default secret

Add in environment a different secret key before deploying the network

2. MongoDB protections

- Add authentication protocols for connections
- Do not expose DB publicly

3. Token validations

- Validate valid tokens signature
- Validate subject field

BIBLIOGRAPHY

- Paper: Penetration Testing of 5G Core Network Web Technologies
- Project attack flow on Github
- Open5GS Documentation
- RFC 7519: JSON Web Token (JWT)
- CWE-639: Authorization Bypass Through User-Controlled Key

UNIVERSITÀ DI PISA

5G

THANK YOU!

CONTI SIMONE, MAT. 675682
COPELLI FRANCESCO, MAT 675686
LEPORE NICOLA, MAT. 678038