https://doi.org/10.31891/2307-5732-2025-359-108
УДК 004.056.55

ПАХОЛЮК ОЛЕГ
Міжнароднийуніверситет
https://orcid.org/0009-0002-8702-1324
E-mail: olehpah@proton.me

ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ ІНФОРМАЦІЙНИХ СИСТЕМ НА ОСНОВІ ПРОЦЕДУР ІДЕНТИФІКАЦІЇ ТА АВТЕНТИФІКАЦІЇ

У статті досліджено проблему логічних помилок у процесах автентифікації, що можуть призвести до критичних вразливостей у системах контролю доступу. Запропоновано формальну модель, яка демонструє типову помилку логіки перевірки облікових даних, коли автентифікація дозволяється за умови виконання хоча б однієї з умов: наявності правильного пароля або відсутності ідентифікатора користувача. Для верифікації моделі використано інструмент ProVerif, що дозволяє автоматично перевірити властивості безпеки протоколу. Результати моделювання підтверджують можливість обходу автентифікації, що свідчить про необхідність застосування формальних методів аналізу на етапі проєктування систем безпеки.

Ключові слова: *кібербезпека, автентифікація, вразливість доступу, логічна помилка, формальна верифікація*

PAKHOLIUK OLEH
International University

DETECTION OF INFORMATION SYSTEM VULNERABILITIES BASED ON IDENTIFICATION AND AUTHENTICATION PROCEDURES

The article investigates a critical aspect of cybersecurity—logical flaws in authentication mechanisms—that may result in severe vulnerabilities within access control systems. The focus is placed on a specific class of errors arising from incorrect formulation of credential verification logic, which can inadvertently allow unauthorized access. To illustrate this issue, the authors propose a formal model that encapsulates a common logical misconfiguration: authentication is granted if either a valid password is provided or the user identifier is missing. Such a condition, though seemingly benign, introduces a dangerous loophole that can be exploited by adversaries to bypass authentication entirely.

To rigorously analyze the security implications of this flaw, the model is subjected to formal verification using the ProVerif tool—a widely recognized framework for automated analysis of cryptographic protocols. ProVerif enables the simulation of various attack scenarios and the verification of key security properties such as confidentiality and authentication. The results of the analysis reveal that the flawed logic permits authentication bypass under certain conditions, confirming the presence of a vulnerability that could be exploited in real-world systems.

The findings underscore the importance of applying formal methods during the design and validation stages of security protocols. By modeling and verifying authentication logic before deployment, developers can identify and eliminate subtle errors that might otherwise compromise system integrity. The article advocates for the integration of formal verification tools into the development lifecycle of secure systems, emphasizing their role in enhancing resilience against logical attacks and ensuring robust access control.

Keywords: *cybersecurity, authentication, access vulnerability, logical flaw, formal verification*

Стаття надійшла до редакції / Received 20.10.2025
Прийнята до друку / Accepted 15.11.2025

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Інформаційна безпека є одним із пріоритетних напрямів сучасного цифрового суспільства. Захист інформаційних систем потребує постійного удосконалення засобів контролю доступу. Атаки, спрямовані на механізми автентифікації, роблять системи захисту даних вразливими. Процедури ідентифікації (розпізнавання) та автентифікації (підтвердження справжності) є базовими криптографічними заходами безпеки, де одним із ключових аспектів забезпечення інформаційної безпеки є коректна реалізація механізмів автентифікації та контролю доступу. На практиці значна частина вразливостей виникає не через криптографічну нестійкість, а внаслідок логічних помилок у структурі перевірок, умов доступу або обробці облікових даних. Такі помилки часто залишаються непоміченими при традиційному тестуванні, але можуть бути виявлені за допомогою формальних методів, які дозволяють моделювати поведінку системи на рівні логічних правил. Формальна верифікація забезпечує строгий математичний підхід до моделювання та аналізу таких сценаріїв, дозволяючи дослідникам виявляти суперечності та вразливості в проєктуванні протоколів ще до їх реалізації. Зокрема, інструмент ProVerif дозволяє автоматично перевіряти властивості безпеки (автентичність, конфіденційність та стійкість до атак), що робить його ефективним засобом для виявлення логічних помилок у протоколах автентифікації [1].

Аналіз досліджень та публікацій

У науковій літературі доведено, що неправомірне або зловмисне використання автентифікаційних механізмів — таких як Brute-force, Spoofing, Credential Stuffing та обхід багатофакторної автентифікації (MFA Bypass) — є одним із головних способів, якими кіберзлочинці здійснюють атаки на інформаційні системи [2;6]. Ці методи дозволяють зловмисникам отримати несанкціонований доступ до системи шляхом використання слабких реалізацій автентифікації, людського фактора або технічних вразливостей. [3]. Їх поширеність та ефективність обумовлюють необхідність впровадження додаткових механізмів захисту на рівні протоколів автентифікації та контролю доступу. Проте, стандарти (ISO/IEC 27001, OWASP, NIST SP800-63) визначають

загальні принципи захисту, але не враховують логічні сценарії взаємодії користувача [4;10]. Існуючі SIEM-рішення не завжди здатні виявити логічні порушення у схемі автентифікації. У зв'язку з цим дослідники та практики виділяють низку перспективних напрямів удосконалення засобів моніторингу, що передбачають більш глибокий контекстний аналіз схем автентифікації та поведінкових шаблонів користувачів [5,7].

Серед перспективних підходів удосконалення засобів моніторингу виділяють три напрями, що поєднують структурний, інтелектуальний та формальний аналіз:

— Графи автентифікаційних станів, які забезпечують візуалізацію та аналіз взаємозв'язків між компонентами системи автентифікації (Graph-based Security Models: Attack Graphs моделюють можливі шляхи атаки на систему, дозволяючи оцінити ризики та вразливості; Dependecy Graphs відображають залежності між автентифікаційними подіями, що важливо для виявлення каскадних порушень; Graphs for Botner Detection використовуються для виявлення аномальної поведінки, пов'язаної з ботнет-активністю; Probabilistic Graph Models дозволяють оцінювати ймовірність реалізації певних сценаріїв атаки, враховуючи невизначеність у поведінці системи) [8];

— ML-моделі машинного навчання для класифікації аномалій застосовуються для автоматичного виявлення нетипових шаблонів у поведінці користувачів або систем (Supervised Learning - навчання з учителем на основі розмічених даних; Logistic Regression проста модель для бінарної класифікації; Random Forest / XGBoost - ансамблеві методи, що забезпечують високу точність класифікації; Support Vector Machines, SVM - ефективні для виявлення меж між класами аномалій) [12] ;

— Формальні методи верифікації протоколів дозволяють математично довести коректність або виявити логічні суперечності у протоколах (Model Checking - автоматизована перевірка властивостей моделі протоколу на основі темпоральної логіки; Theorem Proving побудова формальних доказів коректності протоколу; Formal Grammars & E-networks - моделювання поведінки протоколу з використанням граматичних конструкцій та мережевих структур; Symbolic Execution - аналіз шляхів виконання протоколу з використанням символьних змінних; баньянові графи та моделі знання застосовуються для семантичного аналізу та виявлення прихованих залежностей між елементами протоколу) [9].

Формулювання цілей статті

Метою дослідження даної роботи є продемонструвати формальну модель логічної помилки з використанням інструменту автоматичної верифікації ProVerif, що дозволяє описувати протоколи автентифікації у вигляді логічних правил та перевіряти їх на відповідність заданим властивостям безпеки.

Виклад основного матеріалу

Вразливість доступу часто виникає внаслідок помилок в процедурі автентифікації або авторизації. Логічна помилка в цьому контексті означає хибу в структурі або умовах прийняття рішень (приміром, коли система дозволяє доступ за неповної перевірки), що може бути використано для обходу захисту системи. Значна частина вразливостей виникає через логічні помилки в реалізації умов перевірки, незважаючи на широке застосування криптоалгоритмів. Тому процедури ідентифікації та автентифікації все частіше піддаються атакам, які не потребують складної технічної експлуатації — лише логічного маніпулювання. Під час традиційних процедур тестування програмних або інформаційних систем часто не вдається виявити значну частину потенційних помилок, особливо тих, що мають складний логічний чи структурний характер і можуть проявлятися лише при специфічних умовах функціонування. Для вирішення цієї проблеми застосовуються формальні методи моделювання та перевірки, які базуються на математичному описі системи з використанням логічних схем. Завдяки такому підходу, можна визначити системні суперечності, структурні недоліки та інші критичні вади ще на стадії проектування, що забезпечить підвищення надійності, цілісності та безпеки кінцевого програмного продукту ще до його впровадження в експлуатацію. Зокрема, інструмент ProVerif дозволяє автоматично перевіряти властивості безпеки (автентичність, конфіденційність та стійкість до атак), що робить його ефективним засобом для виявлення логічних помилок у протоколах автентифікації.

Вразливості, що виникають на основі процедур ідентифікації та автентифікації, є критичними, адже саме ці механізми контролюють доступ до ресурсів системи. Такі вразливості можуть бути класифіковані за характером порушення безпеки, типом автентифікації, а також за джерелом виникнення — технічним, організаційним або людським. До першої групи належать вразливості, що виникають унаслідок використання слабких або передбачуваних облікових даних. Це охоплює як недостатню складність паролів, так і відсутність політик щодо їх регулярної зміни. Друга група охоплює вразливості, пов'язані з передачею інформації, яка підтверджує справжність у відкритому (незашифрованому) вигляді, що створює ризик перехоплення даних через атаки типу «людина посередині».

Наступну категорію становлять вразливості, пов'язані з відсутністю або неправильною реалізацією багатофакторної автентифікації. У таких випадках навіть компрометація одного фактора (наприклад, пароля) може призвести до повного доступу до системи. До цієї ж категорії належать вразливості, що виникають унаслідок використання застарілих або ненадійних протоколів перевірки справжності (NTLM або HTTP Basic без використання протоколу TLS), які не забезпечують належного рівня захисту від атак повторного використання [2;11].

Вразливості, що виникають у механізмах відновлення доступу є важливими для організаційної безпеки. Якщо процедура скидання пароля базується на слабких контрольних запитаннях або не передбачає належної перевірки особи, це відкриває шлях до атак типу «перехоплення облікового запису».

У формуванні вразливостей значну роль відіграє ще людський фактор. Компрометації протоколу автентифікації сприяє недостатня обізнаність користувачів щодо безпечного поводження з обліковими даними, схильність до використання однакових паролів на різних платформах, а також сприйнятливість до соціальної інженерії.

Також вразливості можуть виникати внаслідок неправильної реалізації протоколів єдиного входу (SSO), зокрема OAuth, SAML або OpenID Connect. Якщо система не перевіряє джерело токена або не здійснює його криптографічну верифікацію, це створює ризик несанкціонованого доступу до системи.

Вразливості автентифікації не завжди пов'язані з криптографічною слабкістю. Часто вони виникають через логічні помилки в реалізації умов доступу. Такі помилки часто не виявляються класичним тестуванням, але формальні методи дозволяють їх виявити на рівні моделі протоколу.

Для демонстрації логічних помилок доцільно використовувати формальні методи, зокрема ProVerif, Tamarin чи NuSMV.

Для моделювання логічної помилки буде використано інструмент автоматичної верифікації ProVerif, що дозволяє описувати протоколи автентифікації у вигляді логічних правил та перевіряти їх на відповідність заданим властивостям безпеки. На Рис.1 представлено модель, що демонструє ситуацію, коли автентифікація користувача реалізована з логічною помилкою.

```
(* Типи *)
type user.
type password.
type token.
(* Функції *)
fun valid : user * password -> bool.
fun allow_login : bool -> bool.
(* Атака: обхід автентифікації *)
let attacker_attempt(u: user, p: password) =
  if (u = null_user || valid(u, p)) then
    event login_success
  else
    event login_failure.
(* Запит безпеки *)
query attacker: login_success.
```

Рис.1. Моделювання логічної помилки з інструментом ProVerif.

У моделі реалізовано сценарій, де сервер дозволяє автентифікацію користувача, якщо виконується хоча б одна з умов:

— користувач не визначений (*null_user*);

— пароль користувача є правильним (*valid (u, p)),* де *u:* user*, p:* password.

Це суперечить принципам автентифікації, оскільки вхід має бути дозволений лише при одночасному виконанні обох умов: наявності валідного користувача та коректного пароля.

Компонентами моделі є:

— типи, що визначають об'єкти автентифікації: user, password, token;

— функції: *valid (u, p),* яка перевіряє, чи є пароль правильним для заданого користувача; *allow_login*, яка визначає, чи дозволено вхід;

— події, що фіксують результат автентифікаці: *login_success*, *login_failure*;

— запит безпеки, що перевіряє можливості обходу автентифікації: *query attacker: login_success*

Інструмент верифікації ProVerif дозволяє автоматично перевірити, чи може зловмисник досягти події login_success без валідного користувача. Якщо запит підтверджується, це свідчить про наявність логічної вразливості, яку можна використовувати для обходу процедури автентифікації.

Отримані результати демонструють, що логічні помилки можуть бути критичними навіть за умови використання стійких криптографічних механізмів. Формальна верифікація дозволяє виявити такі помилки на ранніх етапах проєктування системи, що надає можливість значно підвищити надійність систем безпеки.

У таблиці 1 наведено порівняння двох варіантів логіки автентифікації: коректної, що базується на одночасній перевірці ідентичності користувача і пароля, та некоректної, яка допускає автентифікацію при виконанні лише однієї з умов. Така помилка є типовою для реалізацій, де логічні оператори використовуються без належного аналізу сценаріїв доступу. Модель у ProVerif дозволяє формально перевірити наслідки кожного варіанту, зокрема можливість досягнення події зловмисником.

Таблиця 1

Порівняння логік автентифікації

№	Критерій	Варіант логіки	
		Правильна	**Неправильна**
1	Умова входу	user ≠ null Λ valid (u, p)	user = null V valid (u, p)
2	Тип перевірки	одночасна перевірка ідентичності та пароля (Λ - кон'юнкція)	достатньо виконання хоча б однієї умови (V - диз'юнкція)
3	Рівень безпеки автентифікації	Високий (доступ можливий лише для валідного користувача з правильним паролем)	Низький (можливий обхід автентифікації без існуючого користувача)
4	Ризик обходу процедури автентифікації	Відсутній (при коректній реалізації)	Високий (зловмисник може без ідентифікатора пройти автентифікацію з правильним паролем
5	Результат моделювання в ProVerif	*login_success* для зловмисника недосяжний	*login_success* для зловмисника досяжний (без валідного користувача)

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Формальне моделювання у ProVerif є ефективним засобом виявлення логічних помилок у протоколах автентифікації. Запропонована модель демонструє, як некоректна логіка перевірки може призвести до критичних вразливостей, що не завжди виявляються традиційними методами тестування. Застосування формальних методів є доцільним для забезпечення високого рівня безпеки в сучасних інформаційних системах.

Література

1. Blanchet B. Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif // Foundations and Trends in Privacy and Security. — 2016. — Т. 1, № 1. — С. 1–135. — DOI: 10.1561/3300000004.

2. Dolev D., Yao A. On the security of public key protocols // IEEE Transactions on Information Theory. — 1983. — Т. 29, № 2. — С. 198–208.

3. Corewin. How to Avoid Authentication and Authorization Vulnerabilities [Електронний ресурс]. — 2024. — Режим доступу: https://corewin.ua

4. OWASP Foundation. Authentication Cheat Sheet [Електронний ресурс]. — 2023. — Режим доступу: https://owasp.org

5. Alshamrani A., Myneni S., Chowdhary A., Huang D. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities // IEEE Communications Surveys & Tutorials. — 2019. — Т. 21, № 2. — С. 1851–1877. —DOI: 10.1109/COMST.2018.2887285

6. Mottasec. How Attackers Bypass Multi-Factor Authentication (MFA) [Електронний ресурс]. — 2024.– Режим доступу: https://www.mottasec.com/article/industry-insight-how-attackers-bypass-multi-factor-authentication-mfa

7. Zhang Y., Wang J. Graph models for cybersecurity: A survey [Електронний ресурс] // arXiv preprint arXiv:2311.10050. — 2023. — Режим доступу: https://doi.org/10.48550/arXiv.2311.10050

8. Singh R., Kumar R. Graph-based approaches for network security monitoring and anomaly detection: A review // International Journal of Information Security. — 2023. — Т. 22, № 4. — С. 567–589. — Режим доступу: https://link.springer.com/article/10.1007/s10207-023-00742-7#citeas

9. Shivakumar M. D., Mamatha N. Applications of graph theory in cybersecurity: Network defense models // World Journal of Advanced Research and Reviews. — 2022. — Т. 14, № 2. — С. 735–743. — DOI: https://doi.org/10.30574/wjarr.2022.14.2.0467.

10. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. — Geneva: ISO, 2022. — 36 p.

11. Digital Identity Guidelines: Authentication and Lifecycle Management. — Gaithersburg, MD: NIST, 2017. — 80 с. — Режим доступу: https://pages.nist.gov/800-63-3/sp800-63b.html

12. Supervised Learning Classification Models [Електронний ресурс] // MLJourney. – Режим доступу: https://mljourney.com/supervised-learning-classification-models/

References

1. Blanchet B. Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif // Foundations and Trends in Privacy and Security. — 2016. — Vol. 1, No. 1. — P. 1–135. — DOI: 10.1561/3300000004.

2. Dolev D., Yao A. On the security of public key protocols // IEEE Transactions on Information Theory. — 1983. — Vol. 29, No. 2. — P. 198–208.

3. Corewin. How to Avoid Authentication and Authorization Vulnerabilities [Electronic resource]. — 2024. — Access mode: https://corewin.ua
4. OWASP Foundation. Authentication Cheat Sheet [Electronic resource]. — 2023. — Access mode: https://owasp.org
5. Alshamrani A., Myneni S., Chowdhary A., Huang D. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities // IEEE Communications Surveys & Tutorials. — 2019. — Vol. 21, No. 2. — P. 1851–1877. – DOI: 10.1109/COMST.2018.2887285
6. Mottasec. How Attackers Bypass Multi-Factor Authentication (MFA) [Electronic resource]. — 2024. — Access mode: https://www.mottasec.com/article/industry-insight-how-attackers-bypass-multi-factor-authentication-mfa
7. Zhang Y., Wang J. Graph models for cybersecurity: A survey [Electronic resource] // arXiv preprint arXiv:2311.10050. — 2023. — Access mode: https://doi.org/10.48550/arXiv.2311.10050
8. Singh R., Kumar R. Graph-based approaches for network security monitoring and anomaly detection: A review // International Journal of Information Security. — 2023. — Vol. 22, No. 4. — P. 567–589. — Access mode:https://link.springer.com/article/10.1007/s10207-023-00742-7#citeas
9. Shivakumar M. D., Mamatha N. Applications of graph theory in cybersecurity: Network defense models // World Journal of Advanced Research and Reviews. — 2022. — Vol. 14, No. 2. — P. 735–743. — DOI: https://doi.org/10.30574/wjarr.2022.14.2.0467.
10. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. — Geneva: ISO, 2022. — 36 p.
11. Digital Identity Guidelines: Authentication and Lifecycle Management. — Gaithersburg, MD: NIST, 2017. — 80 p. — Access mode: https://pages.nist.gov/800-63-3/sp800-63b.html.
12. Supervised Learning Classification Models [Electronic resource] // MLJourney. — Access mode: https://mljourney.com/supervised-learning-classification-models/