

Security Considerations Regarding Access to Information Systems

PhD candidate Petar Dimitrov
University of Economics - Varna, Varna, Bulgaria
p.d.dimitrov@ue-varna.bg

PhD candidate Dimitrios Simeonidis
University of Economics - Varna, Varna, Bulgaria
simeonidis@ue-varna.bg

Abstract

By implementing computer and network security, the IT industry strives to provide users with a high level of trust and reliability. The main goal is to ensure confidentiality, integrity and availability of data (the so-called CIA triad). However, this process is invariably accompanied by a parallel one - constant attempts to detect weaknesses in the architecture, software or hardware implementation of a given solution or system. Malicious actors often have an advantage, since it is enough to find a single vulnerability, while protection must cover all possible attack vectors. In recent decades, a number of techniques have been developed to overcome established protections and exploit systems. The history of information security abounds with cases of mass data leaks, in which weak protection led to adverse consequences for organizations. The publication examines some security issues when accessing information systems.

Keywords: network security, attacks, brute-force, sniffing, keylogger

JEL Code: L86

DOI: 10.56065/IJUSV-ESS/2025.14.2.135

Въведение

Използването на локални и облачни системи налага постоянното надграждане (Jordanov & Petrov, 2023; Jordanov et al., 2024) на мерките за компютърна и мрежова сигурност, която да предоставя на потребителите висока степен на доверие и надеждност. Основната цел е да се осигури конфиденциалност, цялостност и достъпност на данните (т.нар. CIA триада) (Stallings, 2017; Lundgren & Moller, 2019). Този процес обаче неизменно се съпровожда от паралелен - постоянни опити за откриване на слабости в архитектурата, софтуерната и хардуерната реализация на дадено решение или система. Атакуващите често имат предимство, тъй като е достатъчно намирането на една единствена уязвимост, за да успеят, докато защитата се счита за успешна, ако покрива всички възможни варианти на уязвимости и съответно – на атака. Разработени са редица техники за преодоляване на изградени защити и злоупотреба със системи. Най-често срещаните методи за атака, във връзка с автентификацията на потребителите при достъп до информационни системи, включват:

- Brute-force и dictionary атаки – масово изпробване на комбинации от пароли и използване на предварително съставени речници с често срещани пароли;
- Rainbow table атаки – използване на предварително изчислени хеш стойности за ускорено разбиване на пароли;
- Sniffing и Man-in-the-Middle атаки - прихващане и манипулиране на трафик в мрежата;
- Keylogger атаки – зловреден софтуер или хардуерни устройства, които записват натисканията на клавиши;
- Phishing, Smishing, Vishing – използване на човешкия фактор чрез подвеждащи имейли, SMS съобщения или телефонни обаждания;
- Изтичане на данни – публикуване на големи масиви от данни след при компрометиране на информационни системи.

Всеки от тези методи може успешно да се приложи при системи, използващи класическите подходи за автентификация, базирани единствено на пароли, което означава, че те

вече не са достатъчни за надеждна защита. Съвременни техники като двуфакторна автентикация (2FA) и нови стандарти като U2F (Petrov et al., 2020) и WebAuthn целят да премахнат зависимостта от паролите¹ и да елиминират възможностите за горепосочените атаки. Въпреки това, не са редки случаите, при които дори големи ИТ компании не прилагат достатъчни мерки за защита².

1. Характеристики на базови методи за атака срещу автентикационни данни

Brute-force атаката е може би най-лесния за реализиране, но и най-сигурен метод за познаване на пароли или криптографски ключове. При този подход атакуващият изпробва всяка възможна комбинация от символи, докато не открие правилната. Макар на теория този метод да постига целта си при достатъчно време и изчислителна мощ, неговата практическа приложимост зависи от сложността на паролата и от използваните механизми за защита. Колкото по-дълга е паролата и колкото повече разнообразни символи съдържа, толкова по-голямо е множеството от възможни комбинации, което прави атаката експоненциално по-бавна. Развитието на хардуера, особено появата на графични процесори (GPU) и специализирани ASIC (Application-specific integrated circuit) устройства, позволяват извършването на милиарди опити за секунда, което прави дори сравнително сложни пароли уязвими (Bonneau et al., 2012).

Dictionary атаките са следващата стъпка към оптимизация на процеса по разбиване на пароли. Вместо да се опитват всички възможни комбинации, се използват списъци с най-често използвани думи и изрази, които потребителите избират за пароли. Тези списъци обикновено включват елементарни низове като „123456“, „password“, „qwerty“ или имена на любими хора и домашни любимци. Реалната опасност произтича от факта, че голям процент от потребителите продължават да използват прости и предвидими пароли, дори когато системите им предоставят препоръки за по-сложни такива. С времето dictionary атаките се усложняват като се включват комбинации - към реални думи се добавят числа, символи или промени на регистъра - например „Summer2025!“ или „Pa\$\$word“. Тази адаптивност прави dictionary атаките ефективни срещу големи масиви от изтекли данни (Alkhwaja et al., 2023).

Rainbow атаките представляват друг по своята същност метод за разбиване на пароли, спрямо предходните два, при който се използват предварително изчислени таблици с хеширани стойности. Тези таблици, наричани „rainbow tables“, позволяват на атакуващия бързо да намери съответствие между даден хеш и неговата оригинална парола, без да се налага всеки път да изчислява хеша отново. Предимството на този метод е значителното намаляване

¹ В областта на информационната сигурност има редица случаи на масови изтичания на данни, при които слабата защита на пароли е довела до значими последици. Например, през 2009 г. сайтът RockYou претърпява пробив, при който изтичат 32 милиона пароли, съхранявани в plaintext, без никаква защита. През 2013 г. Adobe съобщава за компрометиране на 153 милиона потребителски акаунта, като значителна част от паролите са били криптирани по неподходящ начин и бързо разбити. През 2013-2014 г. Yahoo съобщава, че са компрометирани над 1 милиард акаунта.

² През май 2019 г. Google съобщава, че част от паролите на потребители на Google Workspace (тогава GSuite) са били временно съхранявани в plain text (Frey, 2019). През 2005 г. в Google Workspace (тогава Google Apps) е въведена функционалност, позволяваща на администраторите да създават временни пароли за потребителите. Тези пароли са били съхранявани на сървърите на Google до първото влизане на потребителя, когато той е принуден да смени временната парола. Въпреки че информацията не е била умишлено изложена на публичност, това е пример как plain text съхранението може да създаде риск за потребителите. Подобен случай се наблюдава и във Facebook, където Педро Канахуати, главен инженер по сигурността на компанията, изнася информация, че във вътрешна система са били съхранявани данните на над 600 милиона потребители на Facebook и Instagram в plain text (Sanahuati, 2019). Според него, това не е довело до изтичане на информация, но проблемът произтича от неправилна конфигурация на системата за анализ на грешки. Тези примери показват, че plain text съхранението на пароли, макар и рядко, все още съществува и представлява значителен риск за сигурността на информационните системи и дори големи компании могат да бъдат жертви на атаки, ако не прилагат адекватни средства за защита.

на времето за атака, особено когато става дума за слаби или често срещани пароли. Проблемът, от гледна точка на защитата е, че веднъж изчислени, rainbow таблиците могат да се използват многократно срещу различни системи (Beullens, 2022), стига те да използват един и същ хеш алгоритъм³. Използването на salt стойности (произволни низове, добавени към паролата преди хеширането) прави rainbow таблиците почти безполезни, тъй като за всяка уникална комбинация от парола и salt би трябвало да се генерира нова таблица.

В тази връзка „Hive Systems Password Table“ е популярен и редовно актуализиран ресурс, който илюстрира колко време е необходимо да се разбие дадена парола, използвайки съвременни методи за brute-force атака. Тя показва времето за разбиване в зависимост от дължината на паролата и нейната сложност (кои типове символи съдържа – малки букви, главни букви, цифри и специални символи) и служи да образова потребителите и организациите относно значението на използване на дълги и сложни пароли – видно е, че добавянето на няколко допълнителни символи към паролата или включването на нов тип символи (напр. специален символ) може да увеличи времето за разбиване от секунди на хиляди години (фиг. 1).

Hardware: 12 x RTX 5090 Password hash: bcrypt (10)					
Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	57 minutes	2 hours	4 hours
6	Instantly	46 minutes	2 days	6 days	2 weeks
7	Instantly	20 hours	4 months	1 year	2 years
8	Instantly	3 weeks	15 years	62 years	164 years
9	2 hours	2 years	791 years	3k years	11k years
10	1 day	40 years	41k years	238k years	803k years
11	1 weeks	1k years	2m years	14m years	56m years
12	3 months	27k years	111m years	917m years	3bn years
13	3 years	705k years	5bn years	56bn years	275bn years
14	28 years	18m years	300bn years	3tn years	19tn years
15	284 years	477m years	15tn years	218tn years	1qd years
16	2k years	12bn years	812tn years	13qd years	94qd years
17	28k years	322bn years	42qd years	840qd years	6qn years
18	284k years	8tn years	2qn years	52qn years	463qn years

Фигура 1. Визуално представяне на „2025 Hive Systems Password Table“ – време, необходимо за отгатване на паролата.

Източник: Neskey, 2025, <https://www.hivesystems.com/password>

За да осъществят успешно атаки от разглежданите по-горе типове, атакуващите разчитат на софтуерни инструменти, които автоматизират процеса. Един от популярните е Hashcat, който използва графични процесори за бързо изчисляване на хешове, като по този начин позволява тестване на милиарди комбинации за кратко време. Друг широко използван инструмент е John the Ripper, който съчетава brute-force и dictionary атаки и поддържа множество хеш алгоритми (Marchetti & Vodily, 2022). Съществуват и решения като Cain &

³ През 2012 г. социалната мрежа LinkedIn губи над 117 милиона акаунта, хеширани със SHA-1 без salt, което позволява на атакуващите с rainbow таблици и GPU кълстери да открият голяма част от паролите.

Abel (Gautam, 2024), насочени основно към Windows среди, които предлагат широк спектър от атаки - от класически brute-force до криптоанализ на мрежов трафик. Тези инструменти са свободно достъпни и често използвани не само от злонамерени лица, но и от специалисти по киберсигурност при провеждане на тестове за устойчивост. Това поставя въпроса доколко лесният достъп до подобен софтуер улеснява киберпрестъпността.

Една от основните мерки срещу brute-force, dictionary и rainbow атаки е използването на съвременни криптографски алгоритми за хеширане на пароли, като bcrypt, scrypt или Argon2, които са проектирани да бъдат бавни и устойчиви срещу масови опити (NIST, 2017). Прилагането на salt техники също е от съществено значение, тъй като те правят предварително изчислените таблици неизползваеми и значително увеличават времето, необходимо за атака. Друг важен механизъм е прилагането на ограничения за брой неуспешни опити и временно блокиране на акаунти (т.нар. throttling), което намалява ефективността на автоматизирани атаки. Въвеждането на многофакторна автентикация, при която паролата е само една от няколко необходими стъпки за достъп е също ефективна мярка. Добра практика е организациите да прилагат системи за мониторинг и анализ на поведението на потребителите, които могат засичат необичайна активност и по този начин предотвратят успешни атаки още в зародиш. Прилагането на тези мерки може значително да намалят риска (Petrov et al., 2021) от неототоризирано проникване, те не влияят на друг вектор на атака, който е свързан с преноса на данни, тъй като дори добре защитени акаунти могат да бъдат компрометирани, ако информацията в мрежата бъде прихваната.

2. Характеристики на базови методи за атака чрез използване на слабости в комуникационните канали

Техниката sniffing (често срещан още като packet sniffing) е процес на прихващане и анализиране на мрежовите пакети, които се пренасят между клиент и сървър в една компютърна мрежа. При нормални условия пакетите са видими само за изпращача и получателя, но при уязвими или неправилно конфигурирани системи, недоброжелател може да получи достъп до тях. Това е възможно благодарение на особеностите на мрежовата архитектура, където данните се предават във вид на пакети, преминаващи през множество устройства като рутери, суичове и междинни сървъри (фиг.2). Основната цел на атакуващия е да се сдобие с чувствителна информация - пароли, потребителски имена, данни за банкови карти, идентификатори на сесии и друга поверителна информация. Sniffing методите може да се използват от администратори за диагностика и мониторинг на мрежи (Malkawi et al., 2021; Simeonidis et al., 2024), но техниката се използва също толкова успешно и от недоброжелатели, което я прави инструмент с двойно предназначение.

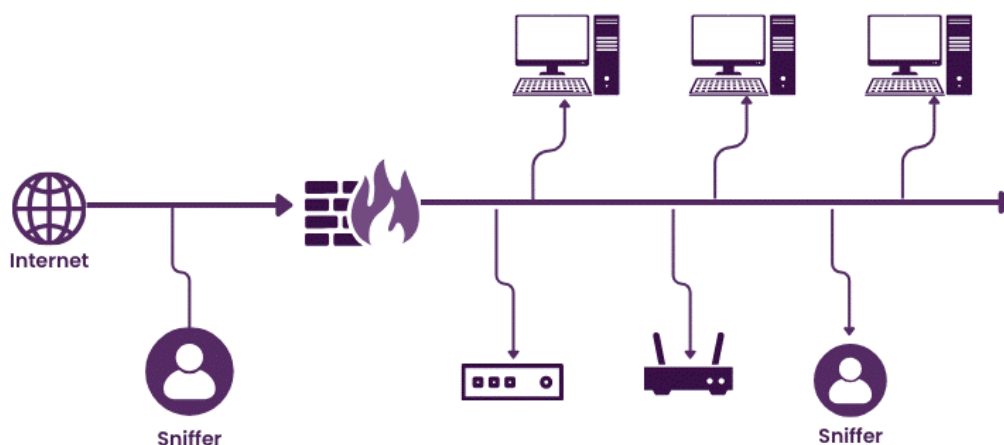
Man-in-the-Middle (MitM) е сравнително по-усъвършенствана форма на sniffing атака, при която атакуващият не само прихваща данни, но и активно ги модифицира. При MitM атаката злонамереното лице се позиционира между две комуникиращи страни - например потребител и сървър - без те да знаят за това. От гледна точка на потребителя, връзката изглежда нормална, докато в действителност трафикът минава през „посредник“. Тази намеса позволява на нападателя да подменя съдържанието на пакетите, да инжектира злонамерен код, да пренасочва жертвата към фалшиви уебсайтове или дори да открадне цяла сесия чрез т.нар. session hijacking. Подобни атаки са особено опасни в публични Wi-Fi мрежи, където криптирането на връзката е слабо или липсва (Simeonidis et al., 2023; Kostadinova et al., 2023).

Съществуват различни методи, чрез които може да се реализира Man-in-the-Middle атака:

- ARP Spoofing - атакуващият изпраща фалшиви ARP (Address Resolution Protocol) съобщения в локалната мрежа, като убеждава устройствата, че неговият MAC адрес принадлежи на легитимния рутер. По този начин целият трафик преминава през неговото устройство.

- DNS Spoofing - чрез подмяна на DNS отговорите потребителите биват пренасочвани към злонамерени сайтове, въпреки че въвеждат правилния адрес.
- SSL Stripping - при този метод атакуващият понижава защитата на връзката, като замества HTTPS връзката с обикновен HTTP, без потребителят да разбира за това.
- Wi-Fi Evil Twin - злонамерено лице създава фалшива точка за достъп с идентично SSID като легитимна мрежа, като по този начин подвежда потребителите да се свържат към нея.
- Session Hijacking - при прихващане на токени за сесии атакуващият може да се представи за легитимния потребител, без да е необходимо да знае паролата му.

Всяка от тези техники има специфични изисквания, за да се изпълни на практика, но всички те са показали своята ефективност в реални кибер инциденти⁴.



Фигура 2. Визуално представяне на процес по осъществяване на sniffing атака.

Източник: *The Knowledge Academy*, 2025, <https://www.theknowledgeacademy.com/blog/packet-sniffing/>

Съществуват множество софтуерни инструменти, които позволяват извършването на sniffing и MitM атаки. Един от често използваните е Wireshark, използван както от администратори, така и от атакуващи за анализ на мрежовия трафик. Ettercap е друг класически инструмент за ARP spoofing и MitM атаки, позволяващ прихващане и модифициране на пакети в реално време. Инструментът Cain & Abel предлага функционалност за прихващане на пароли, декриптиране на VoIP разговори и други техники. Съществуват и по-съвременни решения като Bettercap, които предоставят пълноценна платформа за провеждане на комплексни атаки и автоматизация на процеса. Достъпността на тези инструменти, включително като open-source софтуер, увеличава риска, тъй като дори потребители с ограничени технически познания могат да се възползват от тях.

Защитата от sniffing и MitM атаки изисква комбинация от технологични и организационни мерки.

- Криптиране на връзките - използването на HTTPS, TLS 1.3 и VPN тунели са някои от основните методи за защита срещу прихващане и модификация на данни.

⁴ През 2011 г. се регистрира един от най-известните случаи на MitM атака, когато компанията DigiNotar, която издава цифрови сертификати, е компрометирана. Атакуващите успяват да издадат фалшиви SSL сертификати за домейни на Google, което позволява извършването на MitM атаки срещу ирански потребители. Друг пример е атаката срещу услугата Superfish, предварително инсталирана на някои лаптопи на Lenovo през 2015 г. Софтуерът инжектира реклами чрез подмяна на SSL връзки, което създава огромен риск за сигурността на потребителите. В публичните Wi-Fi мрежи също са документирани множество случаи на MitM атаки, при които потребителите са били пренасочвани към фалшиви страници за вход, откъдето паролите им са били откраднати.

- Защита на DNS - внедряването на DNSSEC предотвратява подмяна на DNS записи.
- Сертификати и PKI - правилното управление на цифровите сертификати гарантира автентичността на връзките и предпазва от фалшиви сертификати.
- Public Wi-Fi защита - потребителите трябва да избягват достъп до чувствителни данни през публични мрежи или да използват VPN.
- Системи за засичане на аномалии - IDS/IPS решенията могат да идентифицират и блокират подозрителен трафик в реално време.

Комбинацията от тези механизми може значително да намали вероятността от успешна атака.

3. Характеристики на базови методи за атака чрез използване на слабости в крайно устройство

В устройствата, с които работят крайните потребители на мрежова услуга, може да се използват техники които регистрират натиснатите клавиши или прихващат локални входни събития и по този начин може да се получи достъп до чувствителна информация преди тя да бъде защитена. Това, например, може да са пароли и потребителски имена, лични и служебни съобщения, номера на кредитни карти, или друга информация, въведена във вътрешни или външни системи. Докато други атаки изискват специфични условия като достъп до мрежата или прихващане на сесии, keylogger атаката се осъществява директно върху крайното устройство, което я прави понякога трудна за откриване. Keylogger атаки може да се осъществят с хардуерни или софтуерни средства (фиг.3).



Фигура 3. Предложена, от някои автори, таксономия за keylogger устройства.

Източник: Bhardwaj & Goundar, 2020.

Софтуерните keylogger-и могат да бъдат внедрени по множество начини - чрез заразен USB носител, зловреден прикачен файл в електронна поща или чрез инсталиране на приложение, маскирано като легитимен софтуер. След като бъде стартиран, софтуерният keylogger работи във фонов режим, като прихваща натиснатите клавиши и често изпраща събраните данни към отдалечен сървър. Някои по-усъвършенствани версии дори могат да заснемат екрана или да записват копираното съдържание от клипборда.

За разлика от софтуерните, хардуерните keylogger-и представляват физически устройства, които се свързват между клавиатурата и компютъра. Те често изглеждат като преходник или удължител и могат сравнително лесно да останат незабелязани. Хардуерните keylogger-и са практически невидими за антивирусния софтуер и софтуерните системи за защита, защото не работят с операционната система на устройството. За използването на подобни устройства атакуващият трябва да притежава физически достъп до хардуера - както при първоначалното поставяне на устройството, така и при неговото премахване с цел извличане на информация.

Единствената ефективна защита срещу keylogger атака е комбинация от ограничаване на физическия достъп до устройствата, внимателно поведение онлайн (Kuyumdzhiiev & Petrov, 2024) и използване на модерни механизми за автентикация, които не разчитат на въвеждане на чувствителни данни от клавиатурата.

Заклучение

Съвременните стандарти за автентикация като FIDO2 и WebAuthn, евентуално прилагани съвместно с биометрия (Dimitrov et al., 2020), предоставят високи нива на защита срещу различни атаки. Вместо потребителят да въвежда парола или друг вид код, тези технологии използват криптографски ключове, съхранявани директно върху устройството или специален хардуерен токен. Частният ключ никога не напуска устройството и не може да бъде прихванат. Така, дори при пълно компрометиране на клавиатура и мрежова връзка, атакуващият не може да се автентичира без притежание на хардуерния ключ. Това прави подхода на FIDO2 и WebAuthn по-сигурен спрямо други подходи за защита. В този контекст развитието и внедряването на FIDO/WebAuthn протоколите представлява съществена крачка напред в борбата срещу различни видове атаки и осъществяват по-високо ниво на сигурност както за отделните потребители, така и за организациите.

References

1. Alkhwaja, I., Albugami, M., Alkhwaja, A., Alghamdi, M., Abahussain, H., Alfawaz, F., ... & Min-Allah, N. (2023). Password cracking with brute force algorithm and dictionary attack using parallel programming. *Applied Sciences*, 13(10), 5979.
2. Beullens, W. (2022). Breaking rainbow takes a weekend on a laptop. In *Annual International Cryptology Conference*, pp.464-479. DOI: https://doi.org/10.1007/978-3-031-15979-4_16
3. Bhardwaj, A., & Goundar, S. (2020). Keyloggers: silent cyber security weapons. *Network Security*, 2020(2), pp.14-19. DOI: [https://doi.org/10.1016/S1353-4858\(20\)30021-0](https://doi.org/10.1016/S1353-4858(20)30021-0)
4. Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE symposium on security and privacy*, pp.553-567.
5. Canahuati, P. (2019). Keeping Passwords Secure. *Meta Newsroom*. [Online] Available from: <https://about.fb.com/news/2019/03/keeping-passwords-secure/> [Accessed 11/10/2025]
6. Dimitrov, G. , Bychkov, O., Petrova, P., Merkulova, K., Zhabska, Y., ... (2020). Creation of Biometric System of Identification by Facial Image. In *IEEE 2020 3rd International Colloquium on Intelligent Grid Metrology (SMAGRIMET)*, pp.29-34. DOI: <https://doi.org/10.23919/SMAGRIMET48809.2020.9263995>
7. Frey, S. (2019). Notifying administrators about unhashed password storage. *Inside Google Cloud*. [Online] Available from: <https://cloud.google.com/blog/products/g-suite/notifying-administrators-about-unhashed-password-storage> [Accessed 11/10/2025]
8. Gautam, T. (2024). Study of password cracking methodologies. *Internafional Journal of Fuzzy Logic and Design*, 9(1), pp.26-36.
9. Jordanov, J., & Petrov, P. (2023). Domain driven design approaches in cloud native service architecture. *TEM Journal*, 12(4), pp.1985-1994. DOI: <https://doi.org/10.18421/TEM124-09>

10. Jordanov, J., Simeonidis, D., & Petrov, P. (2024). Containerized Microservices for Mobile Applications Deployed on Cloud Systems. *International Journal of Interactive Mobile Technologies*, 18(10), pp.48-58. DOI: <https://doi.org/10.3991/ijim.v18i10.45929>
11. Kostadinova, I., Dimitrov, G., Martsenyuk, V., Rancic, D., Dirgova-Luptakova, I., ... (2023). Research and Analysis of Different Real Cases, with use AAI. In *IEEE 2023 16th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, pp.291-298. DOI: <https://doi.org/10.1109/TELSIKS57806.2023.10316177>
12. Kuyumdzhiev, I., & Petrov, P. (2024). Virtualization and Online Engineering of the Administrative Services in Universities. *International Journal of Online & Biomedical Engineering*, 20(1), pp.150-159. DOI: <https://doi.org/10.3991/ijoe.v20i01.45415>
13. Lundgren, B., & Moller, N. (2019). Defining information security. *Science and engineering ethics*, 25(2), pp.419-441.
14. Malkawi, R., Alsmadi, I., Aleroud, A., & Petrov, P. (2021). A firewall-adversarial testing approach for software defined networks. *Journal of Theoretical and Applied Information Technology*, 99(1), pp.227-241.
15. Marchetti, K., & Bodily, P. (2022). John the ripper: An examination and analysis of the popular hash cracking algorithm. In *IEEE 2022 Intermountain Engineering, Technology and Computing (IETC)*, pp.1-6. DOI: <https://doi.org/10.1109/IETC54973.2022.9796671>
16. Neskey, C. (2025). Are Your Passwords in the Green, [Online] Available from: <https://www.hivesystems.com/password> [Accessed 11/10/2025]
17. NIST. (2017). *Digital Identity Guidelines*. NIST Special Publication 800-63B.
18. Petrov, P., Dimitrov, P., Stoev, S., Dimitrov, G. P., & Bulut, F. (2020). Using the universal two factor authentication method in web applications by software emulated device. *International Multidisciplinary Scientific GeoConference: SGEM*, 20(2.1), pp.403-410. DOI: <https://doi.org/10.5593/sgem2020/2.1/s07.052>
19. Petrov, P., Stoev, S., Radev, M., Sergeev, A., & Dimitrov, G. (2021). Risk management processes in information systems development. *International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM*, 21(7), pp.467-473. DOI: <https://doi.org/10.5593/sgem2021/2.1/s07.18>
20. Simeonidis, D., Petrov, P., & Jordanov, J. (2023). Network Intrusion Detection Through Classification Methods and Machine Learning Techniques. In *IEEE 2023 International Conference Automatics and Informatics (ICAI)*, pp.409-413. DOI: <https://doi.org/10.1109/ICAI58806.2023.10339029>
21. Simeonidis, D., Petrov, P., Penchev, G., Petrova, S., Dimitrov, G., & Petrivskiy, V. (2024). Performance and Accuracy Assessment of Detecting Network Intrusions with eSOM-Based Techniques. In *IEEE 2024 International Conference Automatics and Informatics (ICAI)*, pp.586-591. DOI: <https://doi.org/10.1109/ICAI63388.2024.10851699>
22. Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice*. Pearson.
23. The Knowledge Academy, (2025). What is Packet Sniffing? An Essential Guide. *IT Security & Data Protection Resources Blog*. [Online] Available from: <https://www.theknowledgeacademy.com/blog/packet-sniffing/> [Accessed 11/10/2025]