



*Keywords: web application security, cybersecurity, 2fa, authentication*

Łukasz GUGAŁA <sup>1</sup>, Kamil ŁABA <sup>1</sup>, Magdalena DUL <sup>2\*</sup>

# PROTECTING WEB APPLICATIONS FROM AUTHENTICATION ATTACKS

## Abstract

*This paper explores the critical domain of safeguarding web-based applications against authentication attacks, recognizing the persistent challenges posed by evolving cyber threats. The project delineates the distinct objectives of such attacks, including data theft, identity theft, and service disruption, underlining their potential far-reaching implications, such as the compromise of sensitive corporate data and the execution of unauthorized administrative operations. It underscores the pivotal role of user awareness and education as the ultimate defense against authentication-related breaches. Robust security measures, encompassing the use of strong, intricate passwords, encrypted network communication, two-factor authentication, and the regulation of failed login attempts, are emphasized as essential safeguards. Additionally, the project underscores the significance of maintaining system components through regular updates and conducting comprehensive security audits. A holistic approach, integrating technical and human factors, underscores user awareness and ongoing training as indispensable elements in the endeavor to enhance security in an increasingly digital landscape. "Protecting Web Applications from Authentication Attacks" aims to equip its readers with a comprehensive understanding of authentication system security and offers practical directives for bolstering defense mechanisms in a professional and formal context.*

## 1. INTRODUCTION

Developments in technology in the spread of electronic services, including web-based applications, are also a constant battle against attacks on such applications [1]. The principle is simple - the more and more often users use such solutions, the more common attempts to break through the security of such systems become. The great majority of IT systems, which are available to users in the form of a web interface, allow users to authenticate their identity,

---

1. University of Information Technology and Management, Poland

2. Rzeszow University of Technology, Department of Complex Systems, Poland

through various types of functionality, which depends on the level of security required for the data in the created application [2]. Some applications will require a set username and password, while others will require the user to establish additional identity verification through the use of an external device - a fingerprint, SMS code or authentication key.

Technology enables us to relatively easily implement various solutions that should meet even the most stringent business requirements for the created software. However, as is known, the more implementation options for authentication mechanisms users have, the more types and attempts of attacks on the applied components they face. The objectives of attacks on internet systems vary, but they can be divided into three main categories:

1. Data Theft – If users have accounts on services where they can store our private and commercial data, through an attack on the authentication system, an attacker can gain access to this data. In an instant, the contents of network drives, digital libraries, e-learning services, audio and video materials, etc., can be taken over. In the case of commercially sensitive materials, the loss of important corporate data that could be used by competitors may occur, potentially leading to a loss of a competitive advantage.
2. Identity Theft - Increasingly, basic forms of communication are based on social media platforms. While phone numbers are registered, anyone can create an account with a specific name on social media. Impersonating someone through an account associated with their identity is a problem, and even worse, taking over and using someone else's account. Compromising access credentials through an attack on the authentication system can have very serious consequences when dealing with accounts on governmental platforms. Here, users essentially reach the possibility of carrying out administrative operations in someone else's name.
3. Disruption of Service - User authentication is one of the critical components of an application, and its proper functioning is essential for a service to work correctly. The target of attacks is the user authentication process, with the aim of making authentication impossible for the general user. When a large online platform no longer allows users to log into their accounts, it essentially becomes useless.

As users can see, the most critical aspect in information systems is the protection and access to the information they contain. Attacks on the authentication system are typically motivated by the desire to gain unauthorized access to the data within that system [3]. Possessing such data allows for financial gain in various ways, ranging from stealing commercially sensitive information to engaging in simple scams involving money from friends on social media platforms, all the way to professionally organized structures of stolen accounts used as intermediaries in perpetrating further frauds and utilizing someone else's identity [4].

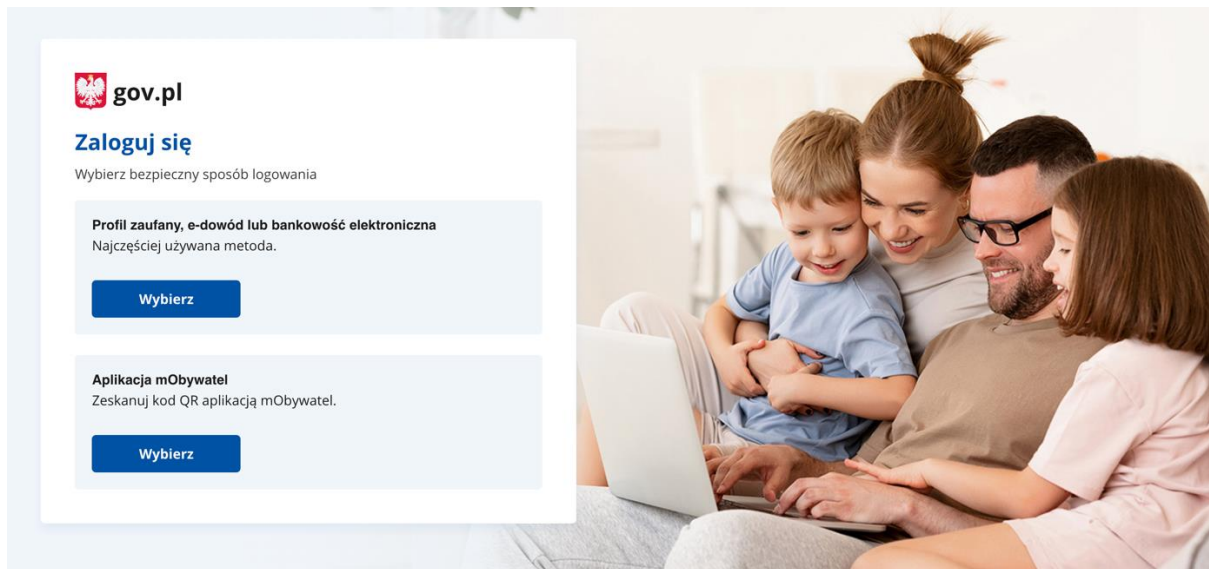




Figure 1 - Login Page for the Polish Government Service Providing Identity

It's worth noting that authentication systems are commonly found in services that provide user identity to external services through the Single Sign-On (SSO) mechanism. The goal of such services is to centralize the collection of user authentication data in one place, with other services typically receiving a token that they can validate with the identity provider service. On one hand, this provides convenience for users, as they only need to authenticate once in a single place. On the other hand, it poses a significant risk because the loss of an account in the identity provider service equates to losing control over actions in services that trust this identity provider.

Awareness that the resilience against authentication-related attacks largely depends on the end-user is becoming increasingly evident in mainstream information. This awareness is being disseminated through basic school education, internal cybersecurity training at workplaces and government offices, as well as general promotion of conscious internet usage through various media channels. The principle is simple – no matter how technologically advanced the information systems are, no matter how many security measures are implemented in a given information system to secure the authentication process, it ultimately depends on the end-user whether they do something that leads to the compromise of their account [5]. A specific example could be thoughtless entry of SMS codes used for two-factor authentication in an information system. Despite double verification in the service, an unaware user might enter a code that, in practice, serves cybercriminals to authorize a different operation.

 bezpiecznedane.gov.pl Wyloguj się



**kontakt@wsiz.edu.pl**

Wyszukiwarka nie znalazła moich danych w bazie wycieku. Co to oznacza?

Dane, które wpisałeś w wyszukiwarce, **nie zostały upublicznione podczas żadnego wykrytego wycieku**. Pamiętaj jednak, żeby sprawdzić wszystkie używane przez Ciebie loginy i adresy e-mail.

Sprawdź kolejne dane

## Sprawdź jak zadbać o bezpieczeństwo w sieci




-  [Zobacz jak włączyć dwuetapową weryfikację w najpopularniejszych serwisach?](#)
-  Poradnik: [Mechanizmy logowania i tworzenia dobrych haseł](#)
-  Poradnik: [Zabezpieczenie poczty i kont w mediach społecznościowych](#)

Figure 2 - Polish Service Providing Quick Verification of Whether a Given Email Address Exists in Databases with Stolen Internet Data

Awareness and informing the public, especially those who are less experienced in using the internet and web systems, about the risks associated with published data from past breaches and cyberattacks are crucial. Often, such data includes sets of usernames and passwords for various services, and in the case of some breaches, it may contain highly sensitive personal or medical information.

The most dangerous are credential sets in the form of usernames and passwords because such data is immediately used for attempting to log in to other popular services. As soon as information becomes available that specific credential data is present in a database from a breach, it is advisable to change (reset) the password for all other services where the user has used that same set of credentials.

For a long time, there have been foreign, private collections that allow searching through such databases of breaches. However, many less tech-savvy users have never heard of them. Therefore, government programs that reach out to all citizens through public awareness campaigns are commendable. An example from Poland is the government website [6] which is a government-run service that allows users to check if their data is present in a large database of breaches.

In summary, attacks on authentication systems are motivated by the desire to gain unauthorized access to information stored in computer systems. Defense against these attacks exists at the technological level but is also crucial in other aspects of using systems. Defensive measures include:

1. Using strong, hard-to-crack passwords.
2. Utilizing encrypted network communication.
3. Enabling two-factor authentication (2FA).
4. Limiting incorrect login attempts.
5. Monitoring the frequency and quality of logins to accounts.
6. Implementing secure user sessions.
7. Keeping systems up to date.
8. Conducting external security audits (not just the system itself).
9. Educating users.

Each of the above points will have further elaboration in the later stages of the project. Each aspect has different consequences in case of non-implementation, affects system and user account security differently, and has a different entry threshold for solution implementation in terms of time, costs, and complexity.

In history, many spectacular incidents related to the inadvertent and unintentional disclosure of credentials have been seen. There are cases where no theft or cyberattack on the authentication system occurs; rather, the user themselves compromises their login credentials [7].

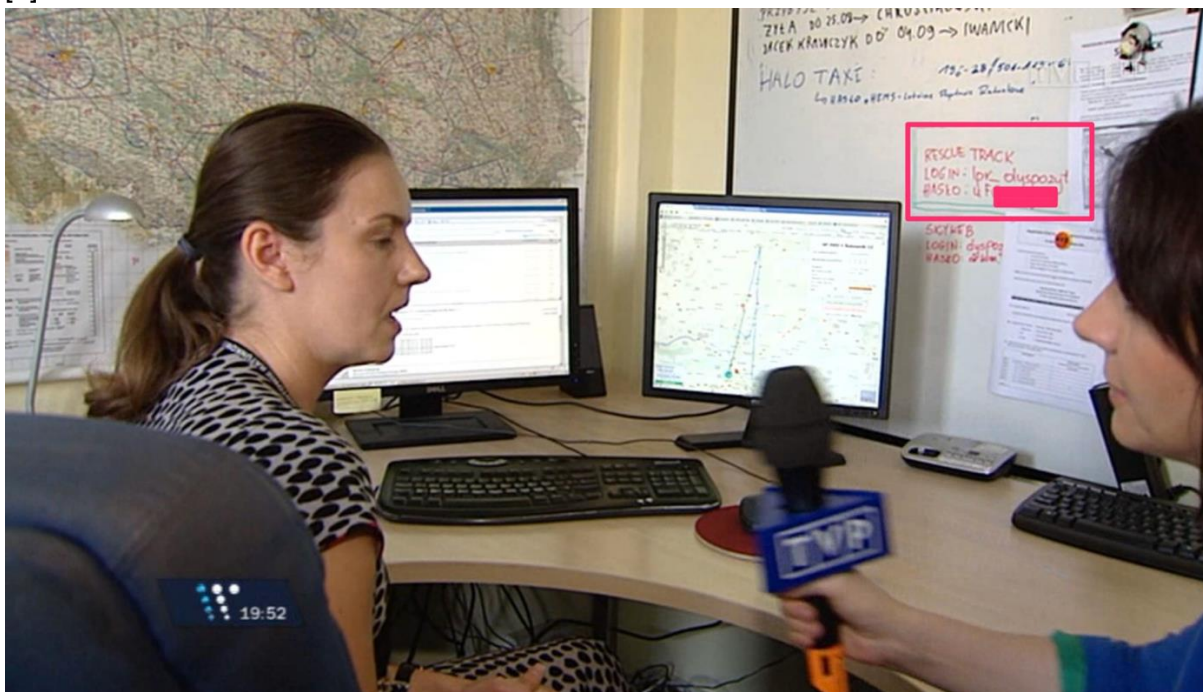


Figure 3 - TVP (Polish Television) Broadcast in Which Login Credentials for the Rescue Helicopter Monitoring System in Poland Were Inadvertently Disclosed

## 2.BASIC TERMS AND CONCEPTS

As part of the project, to better understand the overall topic, it's important to present and explain the meanings of certain concepts and outline the concepts commonly encountered when protecting web application authentication systems from attacks.

**Web System:** It is a computer program running on a server that is accessible to end-users through a web browser. A web system, also known as a web application, can be used on desktop computers, laptops, tablets, or mobile phones. Nowadays, it's standard practice to adapt the system's graphical interface based on the type of device being used. On the server-side, applications can be developed using various technologies, with the most popular being PHP, .NET, Java, or JavaScript with Node.js. On the user's browser side, the technology stack typically includes HTML, CSS, JavaScript, along with images and other multimedia content, depending on the nature and purpose of the system.

**Server:** This is a computer, usually with specialized hardware, designed for providing continuous and uninterrupted operation. Servers are used to ensure constant access to the software installed and running on them, simplifying this concept for now.

**Authentication:** In the context discussed, it's a mechanism of a web application that must identify the user using predetermined methods to grant them access to protected and personalized resources within the information system. The sophistication of authentication mechanisms is determined by business needs, which arise from the type of system and the required level of data security. The most common form of security is the combination of a username and password, which must be provided to gain access to the data. Technically, authentication solutions are implemented through headers sent with HTTP requests, ranging from simple methods like Basic Auth to more advanced mechanisms like OAuth or JWT-based systems. Implementation details are unique to each application and depend on the overall architecture of the specific solution.

**Authorization:** In this context, authorization is a process carried out on the server. Once a user has been authenticated (verified through successful login), the server grants access to specific resources within the running application. This is done based on permissions – an administrator typically has greater permissions than a regular user. Authorization also involves determining the scope of data access; a specific user should only see data they have permission to access. For example, one user should not have access to another user's data unless the system is designed to allow it. The structure of the system depends on individual needs, as determined by the business requirements that led to the creation of the specific information system.

**User Session:** This is a period during which a user interacts with the resources of an information system. It can involve activities such as browsing a website without prior login, editing documents, uploading files, or any actions related to using the system, depending on the available functionality. The primary purpose of initiating and maintaining a session is to identify a specific user for the server. This is usually achieved by creating a cookie on the client-side and transmitting an identifier with each HTTP request. User identification becomes particularly important when authentication and authorization mechanisms are in place, as the server needs to know who is currently requesting resources and whether they have the right to access them.

**User Password:** It is a string of characters that allows a user to confirm their identity by providing the required string when attempting an operation that requires it, such as logging into an account or gaining access to other protected resources.

**2FA:** It is a method of securing access to systems and networks where two different types of information are required to confirm one's identity. This can be a combination of a password with verification through SMS, a code from a mobile application, or the use of a special hardware token. The use of hardware tokens, such as Yubico's YubiKey, has become increasingly popular due to their high level of security. Possessing the physical key is the only way to successfully complete the authentication process.

**Encryption of Network Communication:** This is the process of securing information transmitted within a computer network. This network can be a home network or even the entire Internet, which is essentially one massive computer network. Thanks to protocol standardization, various elements of the global network enable encryption of communication and data protection from the server to the client and vice versa. Encryption ensures that data is stored in a way that unauthorized individuals (e.g., intermediate servers) cannot access it. Achieving such data protection is possible through encryption using one of the available protocols and methods. An example of encrypting information is the HTTPS protocol, where data, such as user login information, is transmitted from the client to the server in a confidential manner while ensuring data integrity.

**IT Security Audit:** This is a process that involves assessing the resilience of systems and operations against cyberattacks. It includes checking physical access to buildings and rooms where system data is stored, as well as the architecture of applications and their security measures against various types of attacks. Depending on the type of audit, real security tests of systems and networks may also be conducted, where a penetration tester attempts to break through the security measures of the tested system. Additionally, all legal aspects related to applied procedures and regulations are examined. In summary, a comprehensive audit of systems and networks involves legal, personnel, and primarily technical actions [8].

### 3.USING STRONG, DIFFICULT-TO-CRACK PASSWORDS

The basic protection for data access during the authentication process is the password. In the vast majority of cases, it's a combination of the username and password. That's why it's so important to use complex passwords to prevent their potential automatic or even manual cracking within a reasonable amount of time.

A strong password is a string of characters that consists of a combination of lowercase and uppercase letters, numbers, and special characters. Additionally, the longer the password, the harder it is to crack. This is based on fundamental mathematical principles—more characters and diversity lead to a larger password cracking dictionary and a greater number of attempts required to guess it [8][9].

When it comes to passwords, it's also important to have a different password set for each service associated with a username. Even the strongest password can be compromised if the internet service it's associated with is successfully attacked, and the password database is stolen. In such a situation, the compromised data set is almost immediately tested for login attempts on other popular online services, including social media accounts, email providers, and hosting services—basically, any important online services [10].

```
5721 = WEBEdition-GRS
7600 = Redmine Project Management Web App
root@mybox:~#
root@mybox:~# hashcat -m 0 -a 1 /root/Desktop/hashes.txt /root/Desktop/password
Initializing hashcat v0.49 with 1 threads and 32mb segment-size...

Added hashes from file /root/Desktop/hashes.txt: 5 (1 salts)

NOTE: press enter for status-screen

bfd59291e825b5f2bbf1eb76569f8fe7:asd123

Input Mode: Dict (/root/Desktop/password)
Index.....: 1/1 (segment), 4 (words), 23 (bytes)
Recovered..: 1/5 hashes, 0/1 salts
Speed/sec.: - plains, - words
Progress..: 4/4 (100.00%)
Running...: --:--:--:--
Estimated.: --:--:--:--

Started: Tue May 5 10:01:48 2015
Stopped: Tue May 5 10:01:48 2015
root@mybox:~# █
```

Figure 4 – Appearance in the Command Line Interface (CLI) of a password cracking application called Hashcat.

The screenshot shown above is from a popular application for automatically cracking passwords. Essentially, you just need to provide a dictionary range, and the computer will automatically start cracking the hash of the password you obtained. This demonstrates that even hashed passwords are not secure. The simpler the password, the faster a potential attacker can gain access to its value before obtaining its hash using functions like sha1, md5, and similar ones.

Of course, it should be understood that using passwords directly related to the user, simple names, common phrases, and other straightforward formulations also expose users to being victims of even manual attempts, where attackers use phrases associated with the user.

## 4.UTILIZING ENCRYPTED NETWORK COMMUNICATION

Even if you have the most complicated password, it can be compromised if the communication between the client and the authenticating server is not encrypted. The process of encrypting information aims to transform a given string of characters in a way that it can only be read by the authorized party for data transmission. In a scenario where a computer user tries to log in to a website and enters the password it should be readable only by the target server. All other intermediate network services involved in data transmission should not be able to read the entered data.

Encryption is achieved through various protocols, most commonly SSL or TLS. Encryption protocols are widely standardized solutions widely supported by network devices. Naturally, encryption can be used when both parties in the data transmission support it.

The entire encryption on the Internet is based on the HTTPS protocol. For the average internet user, a connection made over HTTPS is indicated by a padlock icon in the address bar. However, in practice, it's a complex process that ultimately provides security, confidentiality, and data integrity [11].



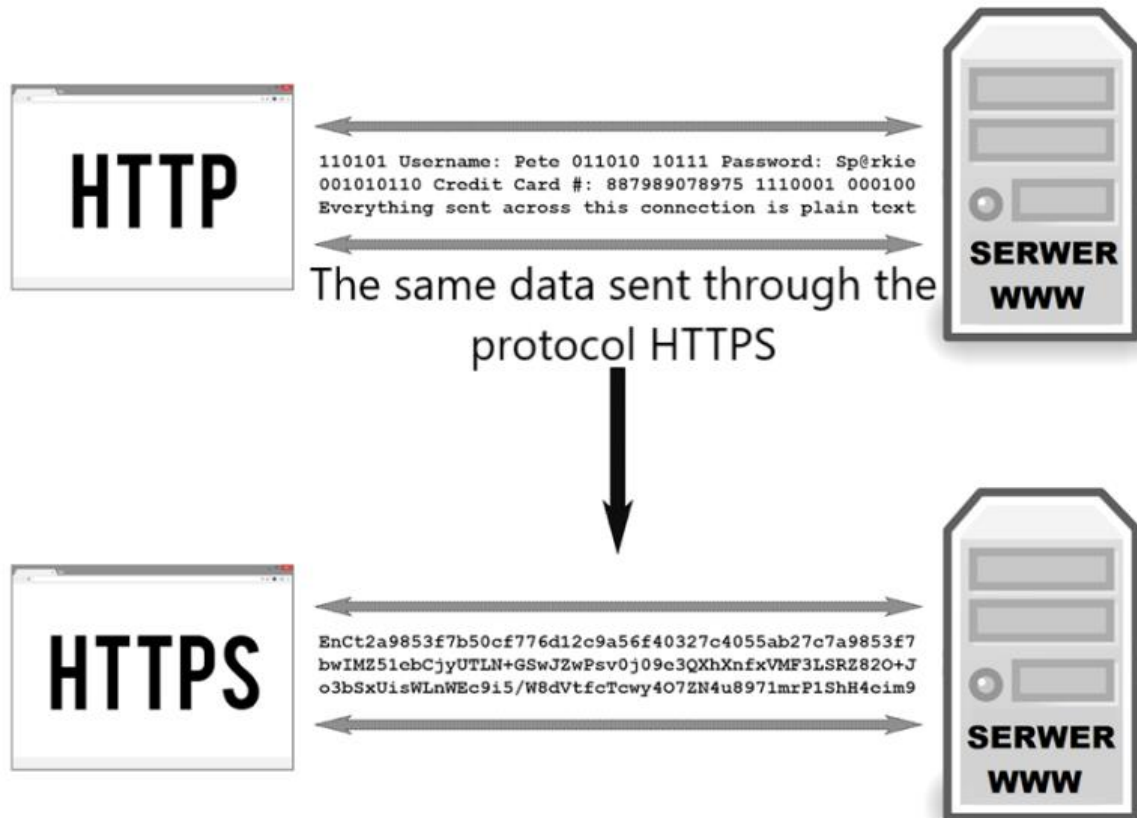


Figure 5 – Comparison of data transmitted via the HTTP and HTTPS protocols

## 5.TWO-FACTOR AUTHENTICATION (2FA)

Two-factor authentication is a method of securing access to IT system resources by using an additional device in the authentication process, which confirms the operation with a token. Adding an additional device makes it less of a problem even if a complete set of data, such as a username and password, is leaked. If the IT system requires an additional token for access, whether it's from a mobile app, an SMS code, or the best possible solution - a hardware key, a potential attacker would need to capture not only the login credentials but also the ability to use the additional device.

A hardware key is the best solution because, unlike a mobile application or SMS, it cannot be remotely used. While one can imagine both the victim's computer and phone being infected with malicious software simultaneously, unauthorized use of a hardware key would require physically intercepting the device. Interestingly and very positively, the use of security keys is becoming increasingly common [12].



Figure 6 – Example of a popular hardware key from YUBICO

Hardware keys should be required for almost every important web service as part of the authentication process. They should also be added by users to all the places on the web where their use is available. This significantly enhances the security of the authentication process. Using keys is convenient, as you don't always have to insert them into a USB port, as some versions fully support NFC technology, which is useful for laptops and mobile phones.

Furthermore, it's a good practice to purchase at least two keys and add this pair of keys to any service that supports them. In case of physical key loss, the user will still have access to their account quickly using the backup key. It's important to be aware that once the requirement for a key is enabled, most online services will require an identity verification process in the event of the previous device being lost, which can take weeks.

## 6.UTILIZING ENCRYPTED NETWORK COMMUNICATION

Another important issue is automatically limiting failed login attempts. If a user provides 3 consecutive incorrect passwords during authentication, the administrator can require them to take a break before attempting further logins, such as waiting for 30 minutes. This protects the user's account from brute force attacks where an automated tool may try to find the password corresponding to a given username using a dictionary.

Furthermore, it's worth noting that protection against brute force attacks in the form of temporarily locking out the authentication operation for a user also has performance benefits for the server. Each processing of an HTTP request consumes server computing resources in terms of CPU time and memory allocation. Allowing automated attempts of this nature is simply a waste of server computational resources, which are not free.

Many popular platforms that provide ready-made solutions, such as complete office suites like Google Workspace or Microsoft Office 365, allow users with administrative privileges to set up advanced rules regarding what should happen with user accounts that have experienced incorrect login attempts. Such incidents are logged in the event log, enabling appropriate analysis that can reveal patterns of mass attacks. This may lead to implementing additional

security measures, such as requiring CAPTCHA verification for all logins from the start, to ensure the system's continuity during an attack and protect against future attempts [7].

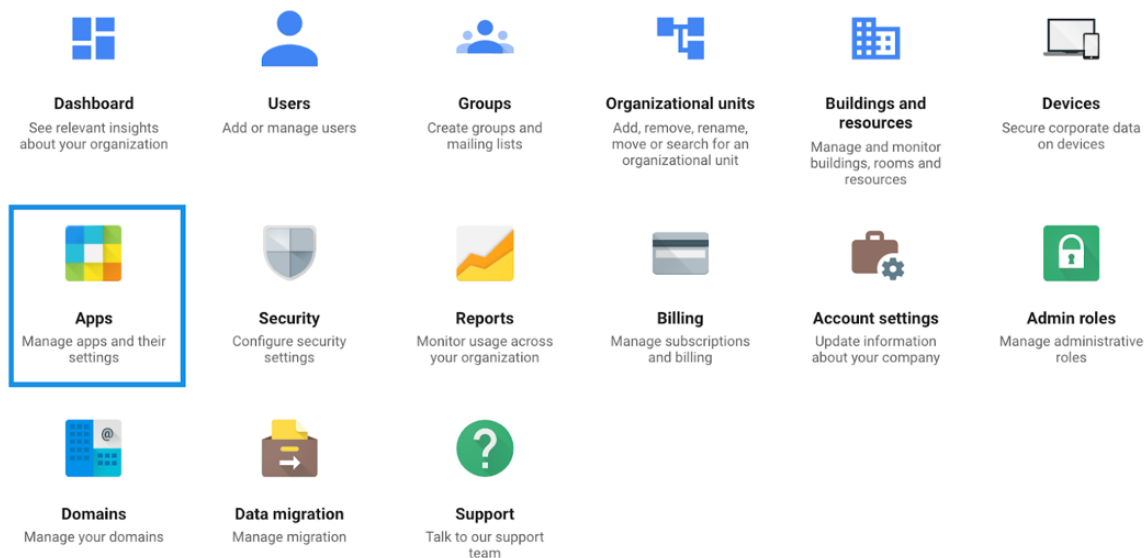


Figure 7 - Google Workspace Admin Console

Within the context of a typical Google Workspace setup, you can define various levels of authentication restrictions based on the currently detected situation related to attacks. This can include whether a single or multiple accounts are being targeted, or if it's a widespread attack aimed at gaining mass access to the organization's accounts.

## 7.UTILIZING ENCRYPTED NETWORK COMMUNICATION

When it comes to user sessions, the topic is highly technical, and the end-user doesn't have much to do with it. A user session and its associated session key are crucial elements of the interaction between the authentication server and the user. The session key allows the server to identify which user is associated with a specific HTTP request. In more modern applications, you might encounter other methods of user identification, such as JSON Web Tokens (JWTs). However, the principle remains simple: you must not allow the interception and substitution of values associated with one user's identification during a request made by someone else. In other words, you must prevent the manipulation of user identification data and impersonation.

When it comes to sessions based on cookies, the most important principles include:

- Cookie creation and reading should happen exclusively on the server-side. The use of the `httpOnly` flag prevents access to cookies from JavaScript in the browser.
- Use cookies only over encrypted connections by adding the `secure` flag to cookies. This ensures that sensitive information is transmitted securely.

- Regenerate session identifiers periodically or even with every HTTP request. This helps mitigate the risk of someone stealing and reusing a session key generated once after successful login.
- Associate user sessions with their IP address and browser fingerprint. While this can enhance security, it can also be risky, especially on mobile devices where IP addresses can change frequently. However, in specific situations, it can significantly improve security.

It's also important to configure Cross-Origin Resource Sharing (CORS) settings on the server side appropriately. This ensures that specific domains can handle specific types of HTTP requests, which is crucial for large web applications [13] [14].

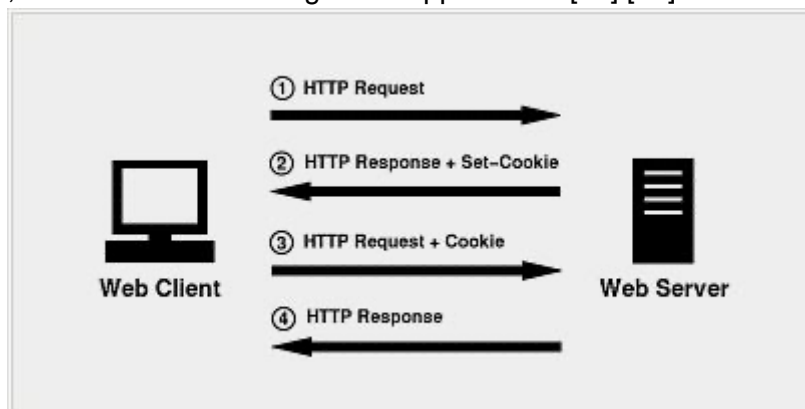


Figure 8 - View of the process of creating and using cookies in the authentication process.

## 8.SYSTEM COMPONENTS UPDATE

Even the most securely designed IT system loses its protective qualities when its components become outdated. Every day, IT specialists, operating on the less legal side of the IT security barricade, actively seek vulnerabilities and flaws in currently used software. If an authentication system is built upon components that contain vulnerabilities, the entire system is vulnerable to password or user session interception.

Software updates apply to literally every element of an IT system. Starting from the BIOS software of devices, through the operating system, down to web server libraries and dependencies, as well as the application software itself. Given the vast range of potential errors and their types, and the practically impossible task of manually determining whether the system currently has dependencies that contain security flaws, there is a need to focus on automating the detection of such vulnerabilities [15].

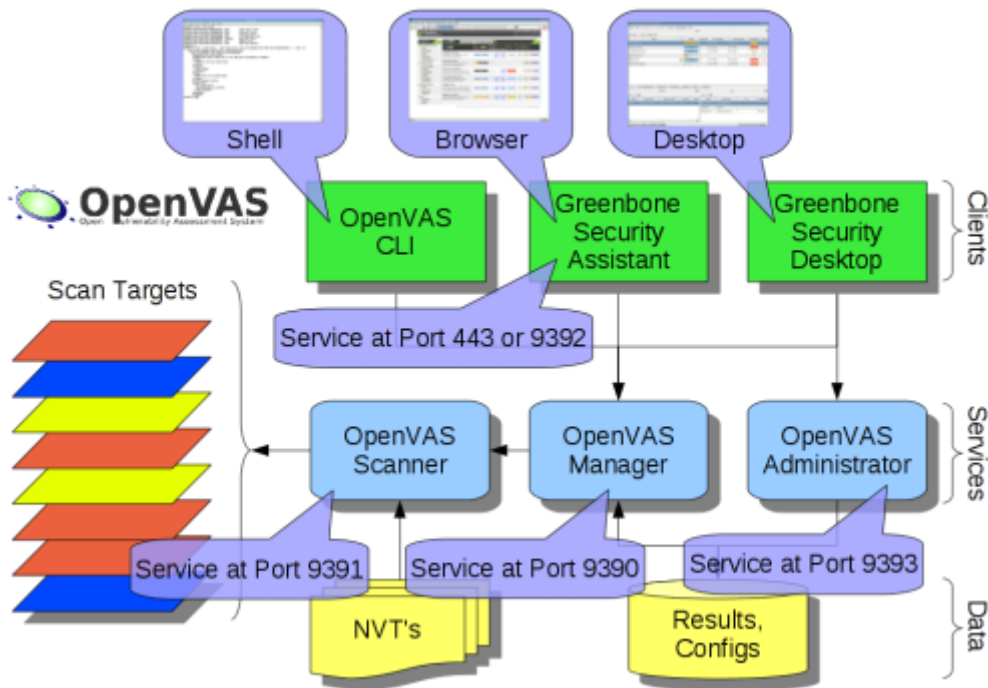


Figure 9 - Diagram of the OpenVAS software package dependencies [16].

Due to the widespread awareness of this issue, many ready-made software packages have been developed to work on various types of network endpoints, from servers to workstations. The purpose of this software package is to continuously monitor the components installed on specific machines, alert about the need for updates, and most importantly, alert about situations where a component with a known vulnerability from a threat database is detected.

The most dangerous vulnerabilities are 0-day vulnerabilities. This means that one day, a system administrator may find out that their system has a critical vulnerability for which there is no patch available yet, and its impact on the system is significant. In systems requiring a very high level of security, such as those providing authentication and authorization mechanisms, these types of vulnerabilities are extremely dangerous.

## 9. EDUCATING USERS

Regardless of how advanced the protection measures for an IT system's resources are, including regularly updated components or source code written in a way that leaves no chances for attackers, unfortunately, humans are the most vulnerable element who can unintentionally lose a password or be deceived in some other way during the authentication process [17].

Such incidents are not directly related to the authentication system itself, but awareness of threats and the potential for secure practices while using authentication systems on the Internet is crucial. Depending on the type of workplace, the nature of activities, and other specific factors, each person is exposed to different types of attacks. Someone who primarily logs into social media should approach this topic differently than an accountant who makes large electronic transfers through online banking. Data protection, especially in government

offices, where a loss of control over login data for systems can be a significant incident, is also very important [5].

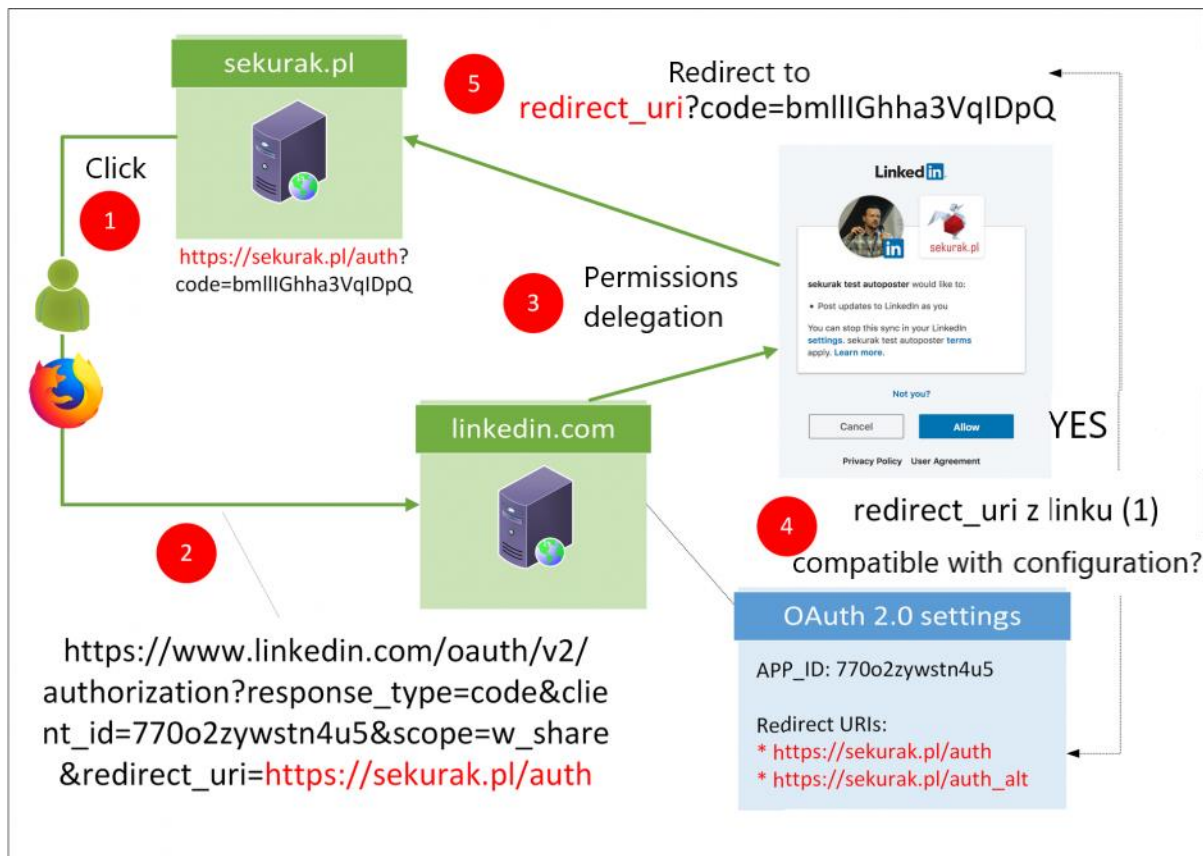


Figure 10 - Screen capture from Securium company presentation - REST API security training [18]

As part of education, it is possible to engage the services of training companies that specialize in IT security and offer training programs tailored to various industries, whether it's a large corporation, a government agency, or a medical institution. In Poland, popular companies like Niebezpiecznik and Securium are capable of conducting IT security training for virtually any organization.

## 10.CONCLUSION

Within the scope of this project, technical and substantive issues related to authentication system security and potential attack vectors have been addressed. However, it's important to recognize that the types of attacks presented are just the tip of the iceberg in the realm of cybersecurity. In terms of attacks on authentication systems, one can conclude that the technology itself, along with the correct implementation from a coding perspective, is just the beginning.

In order to enhance information security, special attention should be given to:

1. Analyzing the nature of information stored within the system - whether it's low-value data, sensitive personal information, trade secrets, or access to financial transactions in banking systems.
2. Implementing security measures in the authentication system that are appropriate for the specific needs. While more security is generally better, it's impractical to require two-factor authentication for a children's story website, and conversely, a simple password login may not be sufficient for a bank serving large corporations.
3. Implementing proper procedures related to monitoring the cybersecurity aspects of authentication system usage. This includes rules for automatic account locking, traffic analysis for authentication attempts to trigger alerts in case of a mass attempt to breach access.
4. Training the personnel responsible for the system on the potential threats associated with the loss of account access, passwords, and other authentication data.

All of the above, when appropriately balanced and implemented in a timely manner, will contribute to an increase in the security of the authentication system. In summary, the technical structure and attention to detail alone are not enough.

### **Author Contributions**

*All authors declare equal contribution to this research paper.*

### **Conflicts of Interest**

*The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

*The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:*

.....

## REFERENCES

- [1] Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Reports*, 7, 8176–8186.  
doi:10.1016/j.egy.2021.08.126
- [2] Usmonov, M. (2021). Identification and Authentication.
- [3] Khan, H. (2013). Comparative Study of Authentication Techniques.
- [4] Erickson, J. (2010). Hacking: the art of exploitation
- [5] Hadnagy C. & Wilson P. A. (2011). Social engineering : the art of human hacking. Wiley.
- [6] Bezpieczne Dane, [bezpiecznedane.gov.pl](http://bezpiecznedane.gov.pl)
- [7] Stallings, W. (2016). *Network Security Essentials*. Pearson
- [8] <https://doi.org/10.6028/NIST.SP.800-63-3>
- [9] <https://www.security.org/how-secure-is-my-password/>
- [10] Anderson, R. J. (2021). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley John + Sons.
- [11] Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice*. Pearson.
- [12] Smith, R. E. (2001). *Authentication: From Passwords to Public Keys*. Addison-Wesley Professional.
- [13] Stutz, D., Pinto, M., & Inni. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley.
- [14] Cross-Site Request Forgery Prevention Cheat Sheet, OWASP Cheat Sheet Series, [https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site\\_Request\\_Forgery\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html)
- [15] Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution Press.
- [16] Compile OpenVAS 7 on CentOS 6, Github, <https://elatov.github.io/2014/06/compile-openvas-7-on-centos-6/>
- [17] Maher Alsharif, M. A., Shailendra Mishra. (2022). Impact of Human Vulnerabilities on Cybersecurity. *Computer Systems Science and Engineering*, 40(3), 1153–1166.  
doi:10.32604/csse.2022.019938
- [18] Bezpieczeństwo API REST – szkolenie, Securitum - bezpieczeństwo systemów IT, <https://securitum.pl/szkolenia/bezpieczenstwo-api-rest-szkolenie/>