2023

Secure Logon Security Plan (Governmental Inter-Departmental Bulletin Board)

APDS7311 PART 1

MICHAEL KRIEL (ST10085507)

Introduction

The creation of an inter-departmental bulletin board is a requirement of the National Government, which necessitates the creation of a strong and secure logon system. This proposal lays out a thorough security plan with many components and safeguards that will be built into the system. By limiting access to the bulletin board to authorized individuals, the main goal is to protect highly sensitive government data. This proposal will include the details required to consider such use cases, the reasoning for security safety measures, and references to back up the decisions that will be made prior to the creation of the strong bulletin board logon system.

Use Cases

To fulfil the government's requirements, the system will need to facilitate the following use cases:

- ✓ User Registration
  New users must be able to create accounts to access the bulletin board, necessitating a secure registration process.

- ✓ User Login
  Registered users will log in to access the system securely.

- ✓ Forgot Password
  Users should be able to reset their passwords through a secure email confirmation process.

- ✓ Session Management
  Efficient session management is vital to maintain user authentication state during their interactions with the system.

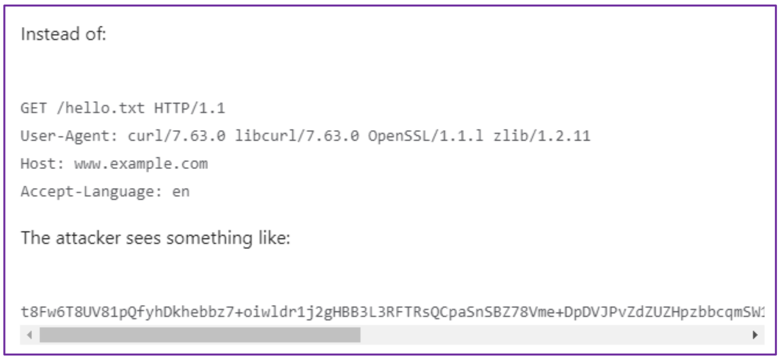
- ✓ User Access Control
  Access control mechanisms will be in place to ensure that only authorized users can utilize the system.

1.) Security Measures for User Logon/Registration

HTTP Requests and Traffic Security

The importance of HTTP requests and traffic security cannot be underestimated. The key idea to remember is the difference between HTTP and HTTPS as HTTPS (HyperText Transfer Protocol **Secure**), if set up and implemented correctly, can avoid cyber-attacks such as Man-in-the-middle attacks

(MITM) *(Soanes, 2021)*. These attacks can compromise data integrity and confidentiality, potentially leading to unauthorized access or user data theft *(Soanes, 2021).*



Instead of:

```
GET /hello.txt HTTP/1.1
User-Agent: curl/7.63.0 libcurl/7.63.0 OpenSSL/1.1.1 zlib/1.2.11
Host: www.example.com
Accept-Language: en
```

The attacker sees something like:

```
t8Fw6T8UV81pQfyhDkhebbz7+oiwldr1j2gHBB3L3RFTRsQCpaSnSBZ78Vme+DpDVJPvZdZUZHpzbbcqmSW
```

Figure 1 Source (Cloudflare.com)

What potential attacks are we defending against:

Eavesdropping/Spying: Attackers can intercept and read data being transmitted between the client/user and the server. This can expose very sensitive information such as the users' login credentials or confidential government data *(Frankenfield, 2022).*

Man-in-the-Middle Attacks: Malicious actors can place themselves between the client and the server, interrupting and potentially altering the data exchanged. This could lead to unauthorized access or data tampering *(Izquierdo, 2022).*

How the measures we propose will defend against these attacks:

To defend against these potential attacks, we propose implementing HTTPS, as mentioned earlier. HTTPS encrypts data transmission between the client and the server, making it unreadable to unauthorized parties during transit *(Soanes, 2021)*.

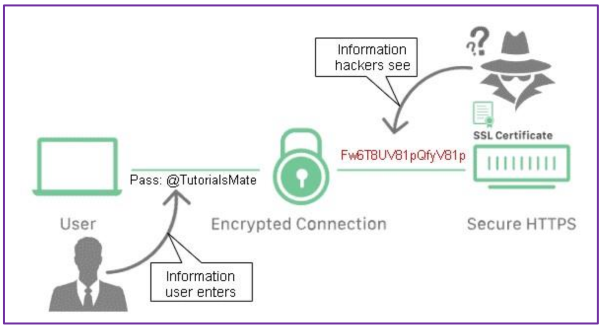


Figure 2 Source: (TutorialsMate.com)

Specifically, we plan to utilize modern cryptographic protocols such as TLS 1.3 and strong ciphers like AES-GCM or ChaCha20-Poly1305 to ensure robust traffic security *(Soanes, 2021)*. This encryption combats eavesdropping and ensures the confidentiality and integrity of the data.

We will set up the "Strict-Transport-Security" (HSTS) header in the HTTP response headers. When connecting to the application, HSTS instructs web browsers to only use HTTPS *(Soanes, 2021)*. As a result, users who attempt to access the application over an insecure connection will be automatically redirected to the secure HTTPS version, further protecting the data being transferred.

We'll also make sure that trusted Certificate Authorities (CAs) are used to authenticate the website that is being accessed, increasing user confidence *(Coclin, 2021)*.This enhances security while also benefiting the website's search engine ranking, which strengthens its legitimacy *(Coclin, 2021)*. Additionally, to reassure users of the secure HTTPS connection, modern browsers offer visual indicators like padlock icons in the address bar *(Soanes, 2021)*.

Figure 3 Source: (ChemiCloud.com)

Input Validation

Data input validation is very important because it acts as the first line of defence against a variety of security threats such as injecting code into user input fields (SQL Injection) etc. Improper error handling can expose sensitive information or functionality to unauthorized users *(McCarvill, 2022)*. Attackers might exploit error messages to gain insights into system vulnerabilities.

What potential attacks are we defending against:

SQL Injection: SQL injection is an attack where malicious SQL statements are inserted into input fields to manipulate the database *(Yahya, 2023)*. Proper input validation prevents attackers from injecting harmful SQL queries by ensuring that user inputs do not contain special characters or unauthorized keywords.

Cross-Site Scripting (XSS): XSS attacks involve injecting malicious scripts into web applications that are then executed by other users' browsers *(Banach, 2022)*. Input validation safeguards against XSS attacks by filtering out or escaping potentially harmful characters in user inputs *(McCarvill, 2022)*.

How the measures we propose will defend against these attacks:

Data Type Checking: Attackers cannot inject SQL code or other malicious scripts by ensuring that input data matches the expected data types *(Yahya, 2023)*. By making sure that user inputs adhere to preset data types, this validation mechanism helps in maintaining the database's integrity *(Yahya, 2023)*.

Format Checking: Ensuring that input data matches specific formats, such as valid email addresses, helps prevent attackers from injecting script tags that could lead to XSS vulnerabilities *(Yahya, 2023)*.

Storing and Hashing of Passwords

A key component of cybersecurity is secure password storage, which is committed to protecting user credentials and confidential information. In this section, we'll go into detail about how crucial it is to store passwords securely and explain how specific security precautions, like salting and hashing passwords, offer a strong defence against potential attacks. Due to its role in preventing and reducing potential security threats, secure password storage is crucial.

What potential attacks are we defending against:

Password Database Breaches: Passwords stored in plaintext represent a substantial security risk *(Jacobson, 2020)*. In the unfortunate event of a breach, attackers can easily access user accounts, thereby compromising sensitive information, financial data, or personal records.

Password Reuse: Users frequently reuse passwords across multiple services, exacerbating the security threat *(Jacobson, 2020)*. A breach in one service can swiftly cascade into unauthorized access to other accounts across various platforms, magnifying the potential damage.

How the measures we propose will defend against these attacks:

Password Hashing: Password hashing stands as a foundational measure in secure password storage. It involves the transformation of user passwords into irreversible, fixed-length hash values *(Jacobson, 2020)*. This process ensures that even if malicious actors gain access to the hashed passwords, they cannot reverse-engineer them to obtain the original plaintext passwords.

Use of Strong Hashing Algorithms: The choice of hashing algorithm significantly impacts the security of password storage *(Jacobson, 2020)*. Employing modern, slow hashing algorithms, such as Bcrypt and Argon2, enhances security measures. Slower hashing increases the time and computational resources required for brute force attacks, rendering such attacks practically infeasible. Moreover, strong hashing algorithms are designed to resist collision attacks, making it exceedingly challenging for attackers to find different inputs that result in the same hash *(Arias, 2019)*.

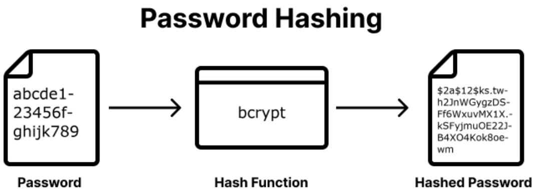


Figure 4 Source: (AuthGear.com)

Salting Passwords: Password salting constitutes an additional layer of defence. It involves adding a unique random value (known as a salt) to each user's password before hashing *(Arias, 2019)*. This practice ensures that even identical passwords yield different hash values. This complexity increases the effort required for password cracking *(Arias, 2019)*.

Maintaining Authentication State

Maintaining authentication state is a critical component of cybersecurity, ensuring that once a user is authenticated after the logon process, they retain their authorized access throughout their current session.

What potential attacks are we defending against:

Session Hijacking: We can safeguard our users from Session hijacking, often referred to as "session fixation", is a security attack where an unauthorized user gains access to an authenticated user's session *(Arampatzis, 2023)*. This can result in identity theft, unauthorized data access, or malicious actions on behalf of the real or actual user.

Session Timeout: Session timeout attacks occur when an attacker abuses an users' session that has been left unattended/abandoned for an extended period *(Arampatzis, 2023)*. Attackers can take control of abandoned sessions and misuse them.

How the measures we propose will defend against these attacks:

Use of Secure Tokens: Utilizing secure tokens, such as JSON Web Tokens (JWT) or session cookies, is fundamental in maintaining authentication state securely *(N-able, 2020)*. These tokens serve as a way to check user authentication/authorization and are even encrypted to ensure their integrity and confidentiality remains *(N-able, 2020).*

Session Management Headers: Specific HTTP headers, like HTTP-Only and Secure Flags for the Cookies, can be used to protect against session hijacking and session timeout attacks. Setting this up makes sure that they are only transmitted over secure (HTTPS) connections and are inaccessible via client-side scripts, as stated in the HTTP Requests and Traffic Security *(OWASP, 2021).* This stops hackers from altering session data or stealing cookies while they are being transferred. To reduce the risk of session hijacking, the SameSite attribute for cookies can also be set to "Strict" or "Lax" *(OWASP, 2021).* It aids in improving session security by regulating when cookies are sent with cross-origin requests. To ensure that inactive sessions are ended after a period of user inactivity, we can implement session timeout policies *(OWASP, 2021).*

Credential Security

Credential security is essential to protect sensitive data and prevent unauthorized access to accounts, systems, and data. The chances of identity theft, financial loss, and damage to one's reputation on both a personal and professional level are reduced as a result.

What potential attacks are we defending against:

Brute Force Attacks: These attacks involve systematically trying all possible combinations of characters, typically starting with the simplest and progressing to more complex ones, to guess a user's password *(Hanna, 2021).*

Dictionary Attacks: An attack using a predefined list of commonly used words, phrases, or passwords (often from dictionaries) to steadily guess a user's password *(TechTarget, 2021).*

Phishing Attacks: Phishing attacks involve tricking users into revealing their login credentials, often by impersonating a trustworthy entity through deceptive emails, websites, or messages (TechTarget, 2021).

How the measures we propose will defend against these attacks:

Use Hashing Algorithms: We can store passwords using hashing algorithms (e.g., bcrypt) to prevent plain text storage and enhance security *(Jacobson, 2020)*.

Utilize HTTPS: Transmitting user credentials over HTTPS to encrypt data during transmission, can prevent interception by attackers and protecting against man-in-the-middle attacks *(Soanes, 2021).*

Implement Strong Password Policies: Enforce password policies that define length, complexity, expiration, and reuse rules. This deters brute force and password guessing attacks and encourages secure password choices *(Srinivasan, 2023)*.

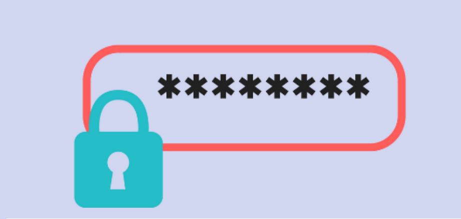


Figure 5 Source: (Webwise)

Overall Flow of Login Process

The overall flow of the login process significantly impacts user experience and security. A poorly designed login process could lead to vulnerabilities or confusion, potentially compromising user accounts or the integrity of the system.

What potential attacks are we defending against:

Brute Force Attacks: These attacks involve systematically trying all possible combinations of characters, typically starting with the simplest and progressing to more complex ones, to guess a user's password *(Hanna, 2021).*

Credential Stuffing: Attackers use previously leaked username-password pairs to gain access to multiple accounts. Secure login flows, along with password policies and MFA, can stop these attempts *(Cloudflare, 2023).*

Session Hijacking: Without proper authentication and session management, attackers can hijack users' active sessions. Secure login flows establish and manage user sessions securely, reducing the risk of hijacking *(Arampatzis, 2023).*

Phishing Attacks: Phishers can create fake login pages to steal user credentials. Secure login flows, through visual indicators (e.g., HTTPS) and user education, help users verify the legitimacy of login pages *(TechTarget, 2021).*

How the measures we propose will defend against these attacks:

HTTPS (SSL/TLS Encryption): We can implement HTTPS for secure data transmission during login. HTTPS encrypts data in transit, preventing eavesdropping by attackers *(Kothmayr et al., 2015)*.

Strong Password Policies: We can do this by enforcing password complexity requirements, password length, and password expiration policies *(National Institute of Standards and Technology, 2019)*. Encouraging users to create strong passwords is a basic but effective approach as well.

2. Protection Against Attacks

Username Harvesting

Definition: Username harvesting is the process of collecting valid usernames from a target system or application. Attackers gather usernames for various malicious purposes, such as launching future attacks, including phishing or brute force attacks *(Overby, 2022).*

How it works: Attackers may use techniques like web scraping, phishing, social engineering, or analysing publicly available information to compile a list of valid usernames *(Overby, 2022)*.

Protection Measures

Implement Account Lockout: Enforce an account lockout policy that temporarily locks an account after a certain number of failed login attempts. This deters attackers from performing username enumeration via brute force *(Sindhuja & Vaidehi, 2015).*

Use User Enumeration Mitigation Techniques: Employ security mechanisms to prevent user enumeration. Customize error messages to avoid revealing whether a username is valid or not (OWASP, 2021).

Brute Force Attacks

Definition: These attacks involve systematically trying all possible combinations of characters, typically starting with the simplest and progressing to more complex ones, to guess a user's password *(Hanna, 2021).*

How it works: Attackers repeatedly guess usernames and passwords until they find the correct combination, exploiting weak or easily guessable credentials *(Hanna, 2021).*

Protection Measures

Account Lockout: Implement account lockout mechanisms after a specified number of failed login attempts to slow down or stop brute force attacks *(Sindhuja & Vaidehi, 2015)*.

Strong Password Policies: By enforcing strong password policies that require complex and long passwords, reducing the likelihood of successful brute force attacks *(National Institute of Standards and Technology, 2019)*.

Session Jacking

Definition: Session hijacking (session jacking) occurs when an attacker captures or steals a user's active session, enabling them to impersonate the user and gain unauthorized access *(Arampatzis, 2023)*.

How it works: Attackers exploit vulnerabilities or weak session management practices to capture or manipulate session identifiers, gaining control over the user's session *(Arampatzis, 2023)*.

Protection Measures

HTTPS (SSL/TLS Encryption): Implement HTTPS to encrypt the session data and protect it from eavesdropping during transmission *(Kothmayr et al., 2015).*

Secure Session Management: Adhere to secure session management practices, including generating strong session tokens and associating them securely with users *(OWASP, 2021).*

Session Fixation

Definition: a type of attack where the attacker accesses a valid user session by taking advantage of a flaw in the way that the web app manages its session ID *(OWASP, 2012)*.

How it works: Attackers trick users into using a session identifier that the attacker knows, either by sharing a link or through other means *(Banach, 2021)*. When the user logs in, the attacker can hijack the session.

Protection Measures

Use HttpOnly Cookies: is an attribute applied to cookies to restrict access to them. When session cookies are marked as HttpOnly, it means that JavaScript running on web pages cannot access or modify these cookies *(Banach, 2021)*.

Use Secure Cookies: Secure is another attribute applied to cookies. When session cookies are marked as Secure, they are only transmitted over secure (HTTPS) connections. This ensures that the session data in the cookies is encrypted during transmission, enhancing its security *(Banach, 2021)*.

References and Sources:

- Cloudflare (2023) Why is HTTP not secure? | HTTP vs. HTTPS . Available at: https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/ (Accessed: 13 September 2023).
- Arampatzis, A. (2023) What is session hijacking &amp; how does it work?, Venafi. Available at: https://venafi.com/blog/what-session-hijacking/ (Accessed: 13 September 2023).
- Arias, D. (2019) How to hash passwords: One-way road to enhanced security, Auth0. Available at: https://auth0.com/blog/hashing-passwords-one-way-road-to-security/ (Accessed: 13 September 2023).
- Banach, Z. (2022) Input validation errors: The root of all evil in web application security, Invicti. Available at: https://www.invicti.com/blog/web-security/input-validation-errors-root-of-all-evil/ (Accessed: 13 September 2023).
- Coclin, D. (2021) What is a Ca? certificate authorities explained, SSL Digital Certificate Authority. Available at: https://www.digicert.com/blog/what-is-a-certificate-authority (Accessed: 13 September 2023).
- Frankenfield, J. (2022) What is an eavesdropping attack?, Investopedia. Available at: https://www.investopedia.com/terms/e/eavesdropping-attack.asp (Accessed: 13 September 2023).
- Hanna, K.T. (2021) What is a brute-force attack? - definition from TechTarget, Security. Available at: https://www.techtarget.com/searchsecurity/definition/brute-force-cracking (Accessed: 13 September 2023).
- Ilinca (2017) SSL encryption for everyone, ChemiCloud Blog. Available at: https://chemicloud.com/blog/ssl-encryption-for-everyone/ (Accessed: 13 September 2023).
- Izquierdo, R. (2022) 5 ways to prevent man-in-the-middle (MITM) attacks, The Motley Fool. Available at: https://www.fool.com/the-ascent/small-business/endpoint-security/articles/mitm/ (Accessed: 13 September 2023).
- Jacobson, K. (2020) Hashing: What you need to know about storing passwords, Security Boulevard. Available at: https://securityboulevard.com/2020/05/hashing-what-you-need-to-know-about-storing-passwords/ (Accessed: 13 September 2023).
- Kothmayr, T., Schmitt, J., & Hu, W. (2015). Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN. IEEE Transactions on Information Forensics and Security, 10(9), 1851-1863.
- McCarvill, A. (2023) An introduction to the importance of input validation in preventing security vulnerabilities, Bright Security. Available at: https://brightsec.com/blog/an-introduction-to-the-importance-of-input-validation-in-preventing-security-vulnerabilities/ (Accessed: 13 September 2023).
- N-able (2020) How does token-based authentication work? - N-able, How Does Token-Based Authentication Work? Available at: https://www.n-able.com/blog/how-does-token-based-authentication-work (Accessed: 13 September 2023).
- National Institute of Standards and Technology. (2019). Digital Identity Guidelines: Authentication and Lifecycle Management. Special Publication 800-63B. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf
- OWASP (2012) Session fixation, Session fixation | OWASP Foundation. Available at: https://owasp.org/www-community/attacks/Session_fixation#:~:text=Session%20Fixation%20is%20an%20attack,specifically%20the%20vulnerable%20web%20application. (Accessed: 13 September 2023).

- OWASP (2021) Session Management Cheat Sheet, Session Management - OWASP Cheat Sheet Series. Available at: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (Accessed: 13 September 2023).
- Sindhuja, V., & Vaidehi, V. (2015). An intelligent account lockout mechanism to mitigate brute-force password attacks. Procedia Computer Science, 47, 41-46.
- Soanes, R. (2021) HTTPS vs http: The vital importance of using HTTPS, Securiwisers RSS. Available at: https://www.securiwiser.com/blog/https-vs-http-the-vital-importance-of-using-https/#:~:text=HTTP%20requests%20are%20sent%20in,is%20important%20this%20remains%20private. (Accessed: 13 September 2023).
- Srinivasan, S. (2023) Weak passwords are the biggest threat to organisational cybersecurity, Express Computer. Available at: https://www.expresscomputer.in/guest-blogs/weak-passwords-are-the-biggest-threat-to-organisational-cybersecurity/97828/#:~:text=On%20the%20other%20hand%2C%20weak,%2C%20dictionary%20attacks%2C%20and%20phishing. (Accessed: 13 September 2023).
- TechTarget (2021) What is a dictionary attack? - definition from whatis.com, Security. Available at: https://www.techtarget.com/searchsecurity/definition/dictionary-attack#:~:text=A%20dictionary%20attack%20is%20a,an%20encrypted%20message%20or%20document. (Accessed: 13 September 2023).
- What is HTTPS? definition and working (2023) TutorialsMate. Available at: https://www.tutorialsmate.com/2020/09/what-is-https.html (Accessed: 13 September 2023).
- Yahya, H. (2023) The importance of input validation and error handling, Medium. Available at: https://bootcamp.uxdesign.cc/the-importance-of-input-validation-and-error-handling-5359a4cd7a80#:~:text=Input%20validation%20is%20checking%20that,accurate%20and%20free%20from%20errors. (Accessed: 13 September 2023).

END OF DOCUMENT