MITIGATION OF CYBER THREADS WITH NEXT-GEN AUTHENTICATION

Mr. R Punithavel(Assistant Professor), Ms. Tharunya Amirtham[2] (MCA)
Department of MCA,
KIT- Kalaignarkarunanidhi Institute of Technology,
Coimbatore-641402
punithavel.R@gmail.com, kit.25.23mmc057@gmail.com

Abstract— A lot of security primitives rely on hard mathematical problems. Security using AI hard problems is becoming a new and exciting paradigm but it has not been explored much. In this paper, we introduce a new security primitive based on hard AI problems, i.e., a new family of graphical password systems that resist large-scale online guessing attacks, while being both cost-effective and user-friendly: Next-Generation Security Based on Captcha (NGS-Captcha). It improves security against various attacks including online guessing attacks, relay attacks and with proper use of such dual-view technologies shoulder-surfing attacks. Remarkably, the only way to encounter a password in the system described is to expose it through automatic online guessing attacks, assuming it is in a search set. Furthermore, it resolves the so-called image hotspot issue in conventional graphical password systems like Pass Points that usually encourage weak password selection. For securing the functionality of the system, we have included a secure file storage module just after login. The Authenticated File System module enables authenticated users to upload and browse files on a site. The file storage system provides security of data through encryption and access controls. This added feature further enhances the security level, as it protects not only user's authentication credentials but also sensitive data. It is not a panacea, but it looks like a usable tradeoff between security and usability in the next generation of authentication. The solution is ideal for use in enterprise settings where additional user authentication and secure file access is required to bolster defences against cyber-attacks Plus, the system is integrated with file storage and retrieval options that guarantee end-to-end security. This initiative provides a smooth and secure method for authentication accelerating the move away from the status quo in today's protection against cyber threats.

Keywords— Graphical Authentication, CAPTCHA, CaRP, Secure File Storage, Cyber Threat Mitigation.

I. INTRODUCTION

Conventional password-based systems cannot provide enough protection as cyber threats get in scope and complexity. Many times, users choose weak passwords or reuse credentials across multiple platforms, so allowing unwanted access and data theft. This project aims to create a system using dynamically produced CAPTCHA images in combination with graphical password techniques redefining authentication security.

The system uses CaRP—a visual password model that shows various CAPTCHA images every login session—so rendering brute force and automated attacks useless. Within these CAPTCHA images, the user's password turns into a recognition challenge adding a level of complexity machines find difficult to break. Apart from authentication, the system supports encrypted file uploads and safe downloads, accessible only to authorised users. This project thus closes the usability gap in safe systems and conforms with contemporary needs for strong authentication mechanisms. It becomes especially helpful in applications where confidential file handling and safe login are absolutely vital. Apart from authentication, this project combines a safe file storage system to let users manage and keep their private information following successful login. The module for file storage uses access control systems and encryption to stop illegal access, so guaranteeing data integrity and confidentiality.

This system provides a complete solution for reducing cyber risks and handling typical authentication vulnerabilities by combining modern authentication methods with safe file storage. The project is a good security improvement for many platforms needing safe authentication and data management since it is meant to be scalable and flexible for practical uses.

II. RELATED RESEARCH

The evolving world of cyber threats has generated considerable scholarly and industrial study towards more secure and convenient methods of user authentication. While password-based systems are perhaps the most popular forms of authentication, they are slowly and increasingly viewed as insufficient due to their inadequacy in protecting against complex attacks such as phishing, brute-forcing, keylogging, and shoulder surfing. In order to combat these risks, a number of researchers began looking into the combination of graphical authentication, CAPTCHA, and multi-factor authentication, resulting in hybrid models such as Captcha as Graphical Passwords (CaRP).

Graphical Passwords and Cognitive Psychology in Security Design: A substantial amount of literature demonstrates that people remember images rather than alphanumeric strings better, which serves as the basis of graphical passwords. Cognitive science and usability studies like Biddle et al. (2012) have found that users tend to remember visual patterns, especially association with object location. This feature improves the usability of systems like CaRP, where users identify and click on certain areas of images rather than typing passwords. Other studies found that graphical password systems lessens password fatigue and input errors which improves the adoption and satisfaction rates of the system.

Captcha as Graphical Passwords (CaRP): A Dual-Security Model: The concept of CaRP was first introduced as a novel solution to combat automated attacks and pattern-based password guessing. Unlike static graphical passwords, CaRP dynamically generates CAPTCHA images, ensuring that each login attempt presents a unique visual challenge. Research by B. Zhu et al. (2014) demonstrated that CaRP systems resist dictionary attacks and significantly lower the chances of bots cracking authentication routines due to the unpredictable nature of the CAPTCHA generation process. Furthermore, CaRP can be extended to support different CAPTCHA types, such as object recognition and image reCAPTCHA, improving adaptability across platforms.

Human Interaction and Authentication Design: Usability studies on authentication focus on the balance between security and user experience. Chiasson et al. (2009) conducted research on the design and use of secure graphical passwords and found that their complex visual interfaces do offer additional security measures; however, those interfaces must be engineered for quick user engagement. In our project, the graphical authentication system allows clicks on images that depict the required patterns as intuitive images, instead of distorting CAPTCHA images, thus protecting the users from unnecessary complications.

CAPTCHA Evolution and Resistance to Automation: CAPTCHA technology, designed to distinguish between human users and automated bots, has significantly advanced over time. Initially, text-based CAPTCHAs were effective, but they soon fell prey to machine learning and Optical Character Recognition (OCR) methods. This prompted the move towards image-based CAPTCHAs and led to the introduction of CaRP. Recent studies in adversarial AI highlight a critical insight: while many bots can bypass traditional CAPTCHAs, they face challenges with dynamic pattern recognition tasks found in CaRP systems. This unique aspect positions CaRP as a robust solution in today's cybersecurity environment..

Secure File Handling and Encrypted Storage Systems: Research on secure file storage backs up the backend part of this project. Work on end-to-end encrypted storage systems (e.g., Boxcryptor Cryptomator) shows how crucial it is to encrypt files when they're stored and when they're sent. Our system follows this idea by using AES encryption and allowing file access when graphical authentication succeeds. Also, studies on audit trails and access logs point out how important it is to record all file interactions, which helps with both openness and responsibility.

Multifactor Authentication and Adaptive Security Models : New research also looks into the use of flexible and situation-aware multi-factor authentication systems. Studies by Komanduri et al. (2015) talk about how situation-based authentication—relying on things like what device you're using, your IP address, and your access history—can change security layers on the fly. While this project zeroes in on single-session visual authentication, it sets the stage to add more layers of security down the road.

III. DEVELOPMENT

Setting up the Environment: Our secure login system starts with creating a complete web setup using the latest internet tools. We build the part you see with React.js, which is great for making pieces you can reuse and screens that change . The behind-the-scenes stuff runs on Node.js and Express.js giving us a strong base to make APIs and handle server tasks. We use MongoDB to keep user login info, CAPTCHA designs, and file details. This type of database helps us grow and get to our data fast.

Graphical Authentication (CaRP) Implementation: At the heart of the system is the Captcha as a Recognition Password (CaRP) model. At registration time, the system dynamically creates CAPTCHA images with numerous visual objects. The user picks certain objects or areas in the image as his/her graphical password. These point coordinates are hashed and saved safely in the database. At login time, a fresh CAPTCHA is drawn, and the user is required to duplicate his/her pattern of selection. The back end validates such inputs with the stored ones based on coordinate mapping logic and opens access only if there's an exact match.

Captcha Generator Module:
We've developed a custom CAPTCHA generator that creates unique visual patterns every time you load the page. This generator distorts letters, adds some noise, and even overlaps different shapes to make it tougher for bots to get through. Each CAPTCHA is designed to be one-of-a-kind, ensuring security while still being easy for humans to read. It's seamlessly integrated into the login module using JavaScript, generating everything in real time for a more engaging user experience.

User Interface (UI) Development: The user interface is built with React.js components, providing a sleek and responsive design. Some of the main screens include registration, a CAPTCHA-based login, secure file uploads, and a download dashboard. To maintain a consistent look across all devices, Tailwind CSS is utilized for styling. Plus, interactive elements like modals, loading spinners, and tooltips are included to enhance user feedback and accessibility..

File Storage & Encryption Module Once users have successfully logged in, they can dive into the secure file storage module. Files can be uploaded right through the browser interface, and as soon as they're uploaded, they're encrypted with the AES-256 encryption algorithm. The encrypted files,

along with important details like the file name, type, user ID, and upload timestamp, are safely stored in MongoDB's GridFS. Every time a file retrieval request is made, it goes through an authentication check to make sure that only the rightful users can access their data.

Download Logging & Access Control: To boost accountability, we log every download with timestamps, IP addresses, and user credentials. This way, we can spot any suspicious activity or unauthorized access attempts. The system employs token-based access control (JWT) to ensure session integrity, which helps prevent token hijacking or replay attacks.

Session & Token Management: User sessions are handled using JSON Web Tokens (JWT). When you log in, these tokens are created and they have a set expiration time, which helps ensure that access is limited to a specific period. To keep things secure, all tokens are stored in HTTP-only cookies, which helps reduce the chances of XSS attacks. When you log out, the token is removed from the client, effectively invalidating the session.

Error Handling & Feedback: We've put in place strong error handling on both the front and back ends. Users will receive notifications for failed CAPTCHA attempts, incorrect login details, expired sessions, and any upload problems. Our backend validation works to block any invalid or tampered requests before they can be executed, keeping the system safe from injection and logic attacks.

Testing and Debugging: During the development process, we carried out thorough unit and integration testing using tools like Jest and Postman. We made sure to test front-end components for their responsiveness and functionality across various browsers, while also validating backend endpoints for security and performance. To tackle any unexpected system behaviors during runtime, we relied on error logs and event trackers to help us debug effectively

IV. WORKING PROCESS

SECURE AUTHENTICATION ARCHITECTURE:

The Mitigation of Cyber Threads with Next-Gen Authentication system is designed with a secure and modular framework that guarantees smooth logins, safe file storage, and strong authentication. The front-end is crafted using React.js, which allows for a responsive and engaging user experience, while the back-end runs on Node.js and Express.js, managing request routing, authentication processes, and file handling. For data storage, we utilize a MongoDB database that keeps user credentials, CAPTCHA patterns, and file metadata encrypted, ensuring flexibility, speed, and scalability. At the heart of this system is the CaRP (Captcha as Recognition Password) mechanism, which merges CAPTCHA with graphical password selection. Unlike the usual static login credentials, CaRP creates visual challenges that change with each login session. This innovative approach significantly boosts resistance against automated threats like bots, brute force attacks, and shoulder surfing.

CAPTCHA & GRAPHICAL PASSWORD SYSTEM: When users sign up, they encounter a dynamically created CAPTCHA image filled with various characters or objects. They need to pick a specific pattern—like clicking on three characters in a particular order—which then becomes their graphical password. These click coordinates are carefully mapped, encrypted, and securely saved in the database. When it's time to log in, a fresh CAPTCHA image is generated on the spot, and users must identify and click the same pattern again. The system then checks the coordinates against what's stored using a tolerance-based matching algorithm. If everything lines up, the user is authenticated and granted access. This approach guarantees that the CAPTCHA is unique for each session, making it tough for bots to replicate or for automated attacks to succeed.

FILE ENCRYPTION AND STORAGE

Once you're logged in, users can easily access a secure dashboard where they can upload or download files. Any files you upload are automatically encrypted with the AES-256 algorithm, which is one of the most robust symmetric encryption methods out there. This encryption takes place on the server side before the file is stored in the MongoDB GridFS system. Along with the file itself, we also keep track of important details like the file type, size, timestamp, and user ID. When it's time to retrieve a file, the system decrypts it in real-time and asks the user for confirmation before allowing the download. This way, we make sure that only authorized and verified users can access the stored data

SESSION & TOKEN MANAGEMENT

Session control is managed through JWT (JSON Web Tokens) to keep user authentication secure across different routes. When a user logs in successfully, a token is created and saved in an HTTP-only cookie, which helps shield it from client-side attacks. These sessions are time-sensitive, and the token will expire after a period of inactivity, leading to an automatic logout. When a user logs out, the token is invalidated, and any session traces are cleared, ensuring that access to the system remains secure and temporary.

SECURITY LAYERS & VALIDATION

To keep up with the ever-changing landscape of cyber threats, the system is built with several layers of security that strengthen both the authentication process and how files are handled. One of the key defenses is rate limiting, which controls how many login attempts can be made from a single IP address within a certain time period. This approach effectively reduces the risk of brute-force and dictionary attacks by stopping automated scripts from quickly guessing passwords. In addition to rate limiting, there's a CAPTCHA refresh feature that lets users request a new visual challenge during login, making sure that security and accessibility go hand in hand, especially when the original CAPTCHA is hard to read. The system also uses input sanitization techniques for all input fields and API endpoints. This means that any data provided by users is cleaned and validated before it reaches the backend services, helping to neutralize threats like SQL injection, cross-site scripting (XSS), and code injection attacks. By enforcing strict content-type checks and escaping special characters, the application guarantees that all inputs meet the expected formats and behaviors. Integrated throughout the back-end infrastructure are comprehensive error handling modules. These modules are crafted to capture and log any anomalies without revealing internal logic or stack traces to the end user. This way, malicious actors can't gain insights into the system's architecture or take advantage of unhandled exceptions. Errors are logged with timestamps and contextual metadata, allowing developers and administrators to review incidents and respond proactively to any potential vulnerabilities. Additionally, validation pipelines are in place for all sensitive interactions within the system, especially for file uploads and downloads. These pipelines check file types, scan for any executable or harmful content, enforce size limits, and ensure that only authenticated users can access the relevant resources. All file transactions are carefully monitored to maintain security and integrity.

INTEGRATION WITH WEB TECHNOLOGIES:

Front-End Integration: The front end interacts with the server through RESTful APIs, enabling smooth asynchronous communication. CAPTCHA images are retrieved from a secure endpoint, and all responses are rendered dynamically.

Back-End Operations: The back end guarantees that authentication, encryption, and database interactions happen securely and efficiently. Our file handling services are fine-tuned to manage various file types and sizes without slowing down the system. Cloud Compatibility: Thanks to the modular design of the application, it can easily integrate with cloud platforms like AWS S3 or Google Cloud for secure cloud storage and scalability.

FEATURES AND USER EXPERIENCE: The system is built with a strong focus on security and user-friendly design, making sure that users have a smooth and intuitive experience. One standout feature is the graphical login flow, which swaps out traditional text-based passwords for an image-based authentication method. This not only boosts security but also makes it easier for users, as they engage with visual patterns that are simpler to remember and less susceptible to automated attacks. The file management interface boasts a clean and responsive layout, allowing users to easily upload, view, and download files in a secure environment. It's designed to be straightforward and functional, so even those who aren't tech-savvy can navigate file operations without any hassle. To promote accountability and traceability, the system has a download logging feature that records every file download, complete with timestamps, file IDs, and user credentials. This creates an audit trail that can be reviewed for security or administrative needs, adding a layer of transparency to file access. Moreover, there's a refreshable CAPTCHA feature on the login screen, which lets users request a new CAPTCHA image if the current one is hard to read. This ensures that everyone can access the system easily, enhances the login experience, and keeps the authentication process secure. All these features work together to create a secure, efficient, and user-friendly system, striking a perfect balance between top-notch protection and a smooth, engaging user experience**.**

DEVELOPMENT TOOLS & TECHNOLOGIES: The Mitigating Cyber Threats with Next-Gen Authentication system is built on a cutting-edge full-stack technology suite designed to deliver top-notch security, performance, and scalability. For the front end, we're using React.js, a well-loved JavaScript library that makes it easy to create dynamic and responsive user interfaces thanks to its component-based architecture. On the back end, we rely on Node.js paired with Express.js to handle the server logic and API routing, ensuring smooth communication between the client interface and server processes. When it comes to storing data, we've chosen MongoDB as our database solution. It offers great flexibility for managing user credentials, session data, and encrypted file metadata through its document-based structure. To keep user sessions secure, we implement JSON Web Tokens (JWT), which allow for stateless authentication and time-limited access control. To safeguard user files, we incorporate AES-256 encryption, a robust symmetric encryption algorithm that's well-regarded for its ability to protect sensitive data during both storage and transmission. We also add extra security layers with libraries like Helmet.js, which secures HTTP headers, and Bcrypt.js, used for hashing sensitive user credentials such as passwords. For testing and validation, we utilize tools like Postman to check API endpoints and data flows, while Jest is our go-to for unit and integration testing across various components. All these technologies come together to create a cohesive and powerful development environment that ensures the secure and efficient implementation of the system.

V. CONCLUSION

The Mitigating Cyber Threats with Next-Gen Authentication project isn't just about improving traditional security measures; it's about rethinking how authentication can adapt to the challenges we face in today's digital world. With data breaches, identity theft, and cyber-attacks becoming more sophisticated and common, this initiative offers a forward-looking solution that swaps out vulnerable text-based logins for dynamic, image-based graphical passwords, all while incorporating CAPTCHA technology. By blending cognitive science with security engineering, this system allows users to engage with authentication in a way that feels more intuitive and visually appealing. The use of CaRP (Captcha as Recognition Password) not only helps to block automated attacks but also adds an element of unpredictability and uniqueness to each login attempt—something you don't often see in traditional systems. Plus, the smooth integration of secure file storage and encryption protocols guarantees that sensitive information stays protected from unauthorized access throughout the entire user experience. Moreover, this project highlights the incredible potential of secure-by-design principles. It shows how cybersecurity can be both strong and user-friendly by combining solid back-end encryption with an accessible front-end design. As our digital environments continue to grow, the need for adaptable, intelligent, and user-focused security models will only become more pressing. In summary, Mitigating Cyber Threats with Next-Gen Authentication sets a new standard for secure access control by focusing on both usability and security. It represents a forward-thinking approach to authentication—not just as a barrier, but as a smart, responsive gateway to secure digital spaces. This system not only protects against current threats but also paves the way for future innovations, ultimately creating a safer, more reliable, and trustworthy online experience

VI- REFERENCES

1. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 891–904, 2014.
2. William Stallings, *Network Security Essentials: Applications and Standards*, 6th ed., Pearson Education, 2023.
3. Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 20th Anniversary Edition, Wiley, 2022.
4. Charles J. Brooks et al., *Cybersecurity Essentials*, Wiley, 2022.
5. Richard E. Smith, *Authentication: From Passwords to Public Keys*, Addison-Wesley, 2021.
6. Kevin D. Mitnick, *The Art of Deception: Controlling the Human Element of Security*, Wiley, 2020.
7. National Institute of Standards and Technology (NIST), "Digital Identity Guidelines," [Online]. Available: https://pages.nist.gov/800-63-3/.
8. OWASP Foundation, "Top 10 Web Application Security Risks," [Online]. Available: https://owasp.org/www-project-top-ten/.
9. Cloudflare, "What is Multi-Factor Authentication (MFA)?" [Online]. Available: https://www.cloudflare.com/learning/access-management/multi-factor-authentication/.
10. Auth0 Blog, "Graphical Authentication: Security Through User Cognition," [Online]. Available: https://auth0.com/blog/.