Patrick Cheng - pc2720

Meenakshi Madhu - mm14029

Adria Orenstein - alo278

Raheem Warriach - raw9846

Group 18

Securing Gaming Accounts:

Addressing Login Vulnerabilities

Domain Introduction

The research and project covered in this document explore the domain of gaming. Gaming is a popular pastime for many people, but it can involve many security risks. These include Account Compromise where there is a theft of sensitive personal data such as passwords; Denial of Service attacks, where servers are inundated with false requests by bad actors that result in the game crashing for everyone; Doxing, which involves the leaking of personal data publicly without the user's permission; and Data Breaches, where unauthorized access and transfer of data occurs.

Problem Statement

We aimed to address the issue of Account Compromisation in gaming, specifically during the log-in phase. When a user logs in, they enter their credentials, and the information is sent to the game server to verify. The credential data, in either plaintext or hashed form, is stored on the server side. However, an account can be compromised if a malicious third party intercepts that

connection, or if there is a password data leak. In these cases, the attacker can steal sensitive information like login credentials and gain unauthorized access to the user's accounts, leading to a compromised account.

Our Approach

Our plan to circumvent the leaking of sensitive login information during the connection phase is to establish a new authentication method between the server and the client. This will be achieved using the Open Quantum Safe library, specifically the Dilithium algorithm for digital signature, the Kyber algorithm for key generation, and the AES algorithm for message encryption. The intent is to develop a password-less system for login, which avoids storing any kind of user credentials on the server side, providing security against password leaks. The key encapsulation method for encryption gives additional security and privacy against malicious listeners on the connection.

Our Project and Functionality

Our system would be initiated when a user client attempts to log in, prompting the server to begin initial authentication. Using 'Kyber512' as the key encapsulation mechanism (KEM) algorithm, the server encapsulates a secret using the client's public key and sends the ciphertext to the client. Then, the client can decapsulate the ciphertext with its private key. With this, the server and client should have a unique shared secret known only to them. The client's private key is required to recreate the shared secret which is never transmitted. Each party's private key should only be known to themself so this helps mitigate the risk of an attacker intercepting the secret over the network.

Next, the server generates a “challenge” and encrypts it using AES encryption with the shared secret as the symmetric key. This encrypted challenge is sent to the client who can decrypt the ciphertext using the shared secret or the symmetric key. Using ‘Dilithium’ as the digital signature algorithm, the client signs the decrypted challenge with the client’s private key and sends it back to the server. The server can verify the signature using the client’s public key. The combination of AES encryption and digital signing acts as a double layer of protection to help verify and authenticate the user. The shared secret acts as the symmetric key for AES encryption ensuring that the challenge being signed and sent back is valid. This is because only the two involved parties should be able to know the unencrypted challenge used. Digital signatures also help validate that the message comes from the expected client untampered with.

Once all checks have been passed, the user is authenticated and logged into the game.

Conclusion

This assignment outlines the design and implementation of a password-less authentication system for gaming platforms incorporating the quantum-resistant cryptographic algorithms from the Open Quantum Safe library and the AES encryption algorithm, to provide security against password leaks and malicious attacks during the game login phase. The security of the system depends on the shared secret and private keys of the client and server entities. The shared secret is transferred using the Key Encapsulation Mechanism which helps to transmit keys securely over a channel, and the private keys are never transmitted; thus ensuring the confidentiality and security of the user accounts.

Citations

https://www.e2encrypted.com/posts/kyber-algorithm-post-quantum-champion-revolutionizing-cryptography/

Official Reference Implementation:

https://github.com/pq-crystals/kyber.git

PQCLean:

https://github.com/pqclean/pqclean

Liboqs:

https://github.com/open-quantum-safe/liboqs

Digital Signatures:

https://www.geeksforgeeks.org/digital-signatures-certificates/