WPA2 with Decentralized Authentication servers

Devashish Dewangan(cs21m017)
Nagaraj Chandru(cs20m041)

Declaration

We declare that the work done in this project has not been and will be not used as-is for any other registered course at IITM; if this project code is extended further for any subsequent BTP/MTP project or MS/PhD thesis, this project work will be duly referenced in the reports/theses documents.

Abstract

WPA2 provides stronger security in WLANS, we added some more security features to wpa2 through this project. Instead of using a single Authentication server we used multiple servers and Blockchain technology will be used to maintain identical user credentials among servers to overcome a single point of failure. Kerberos protocol is used as a mechanism for identifying entities in the network.

Introduction

The goal of this term project is to implement WPA2 using Kerberos with decentralization user credentials using blockchain, to eliminate single point of failure.

Security protocols

- ## Authentication Server:
    1. After the AS starts running it will check if there is a psk(pre shared key) for the APs. If it is not there then the AS will generate a key for all the APs token authentication.
    2. The server will start listening to the authentication request.
    3. The userId and password can be added at the AS to the blockchain.
    4. For the authentication the request will be encrypted with the clients psk and the request ID will be in plaintext, the server will retrieve the password from the blockchain with the respective ID. Then decrypt the request using the key(AES CCM-128) if the request is decrypted without an error then the user is verified correctly.
    5. Else if an error occurs the authentication fails.
    6. If after the authentication successful AS will generate a token with the requested validity and send it back to the AP, to send it to the user.

- ## Authentication using token at AP:
    1. AP will see the request to connect by the user and check if the request consists of a token.
    2. If a token is absent then it will connect the user to the AS for authentication.

3. If it consists of a token then it will try to verify by decrypting with AES CCM -128 with psk shared by AS.
4. If the data is decrypted correctly then it will check for expiry time of the token, if it is a valid token then the user is authenticated and gets access.
5. If the token is invalid then the AP will connect the user to the AS.

- **STA/Client:**
    1. The client will check if it has token then it will use token to authenticate itself with AP
    2. If it doesn't have a token then it will look for the psk client. Then using the psk client it will send a request to the AP to connect to th AS.
    3. The request will be encrypted using the psk client.
    4. After getting verified the client will receive the token from AS and it will store it for future purpose.

Kerberos is used as an Access Server(AS) that provides mutual authentication and generates tokens to be used between the client and the Access Point(AP) over the wireless link.

● Authentication using kerberos:

➔ Client and AS prove their identities to each other. The AP blocks non-authentication traffic between the Client and AS until the authentication transaction is successful. The AP does not participate in the authentication transaction other than forwarding traffic between the client and AS.

Kerberos:

- Client & AS will have a secret key (password) shared between Client and AS, stored on blockchain.

C –> AS: ID || E[PSK(C), ID]

- Users can be authenticated at AS by UserID & password (secret key), after which AS will generate a token for the client.

AS –> C: E[PSK(C), (SK || ID || ExpiryTime)]
Token : E[PSK(AP), (SK || ID || ExpiryTime)]
SK(session key can used between AP & client to share data confidentially)

- At AP user can be authenticated using a token(certificate).

● Token generation and distribution:

The AS and Client perform several operations that cause cryptographic keys to be generated and placed on the AP and Client.

● Protected data transfer:

Frames are exchanged between client and the end user station through the AP.
Every user will have

● **Connection termination:**

The AP and Client exchange frames. During this phase, the secure connection is torn down and the connection is restored to the original state.

● **Blockchain:**

Blockchain technology is used to store the credentials of the users in the network. Blockchain is like blocks connected to one another like a chain. In a block users credentials(data) is stored along with timestamp(when the block is created), previous block hash, proof of work(IV), block hash.

Proof of work:

Computation effort put into creating a block in the blockchain network. Computational efforts means in order to create a block we use cryptographic hash function(sha256). By doing repeated hashing of a block till we get the required number of leading zeros in a hash value.

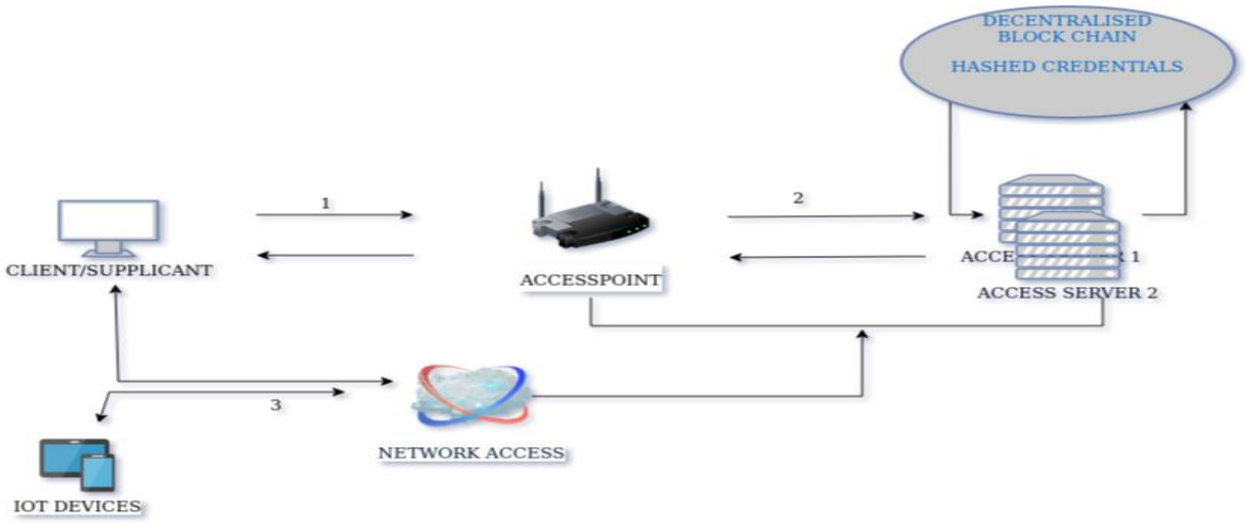
Miners:

Inorder to maintain the consensus problem in distributed systems miners are used. Miners will collect data from users and create a block and distribute it to users. Then users and miners will create their blockchain.

- We used the json format to store the blocks in the blockchain.
- Miners are used to get consensus between the servers(to get identical blocks in servers).
- Servers will send data credentials to miners, it will generate a block using the data and send it to every server.
- Four leading zeros are considered in the generation of pow.

Main use of blockchain in these decentralized authentication servers is that we can verify which server is adding the user. If we use a sql database any one can add data, we cannot verify who is the one added that data and we can distinguish between server and attacker.
It will be very hard for third parties to modify the data by using the blockchain. If he/she tries to modify a single block, he/she has to change the entire blockchain starting with the block.

- **Authentication (at AS)** : Is achieved by userID & password at AS.
- **Authentication (at AP)** : Is achieved by a token at AP.
- **Confidentiality (b/w user & AP)** : Is achieved by secret key shared between the AP and client.
- **Confidentiality (user credentials)** : we are using blockchain technology to store user credentials. A block consists of “PrevBlockHash” + “userID” + “password” + “BlockHash”.



Development environment and implementation

Development environment:

- Language : Python
- Major Components : Authentication Server & Blockchain Database(JSON).
- Packages : Cryptography.

Conclusions

- The Token using the kerberos reduces the overhead of getting authenticated by credentials.
- The synchronization of authentication servers to maintain the blockchain has very large overhead. Thus alteration to the blockchain i.e. to add a user requires a large amount of time.
- Learned how to work as a team.

Learning-Experience

- We have learned how to work with blockchain, json..
- We have learned how simple the blockchain concept is.
- Learned multithreading, multiprocessing and maintaining synchronization among them.

Suggestions

- Add more assignments as they help in understanding how the protocol works in practice. Providing balancing between theory and practice. It would be much more interesting if we have more weightage to assignments.