International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 4

Authentication through Claims-Based Authentication

Pawan Patil, Ankit Ayyar, Vaishali Gatty
MCA, Vivekanand Education Society Institute of Technology,
Chembur East, Mumbai, India

ABSTRACT
Thinking as far as claims and issuers is an effective reflection that backs better approaches for securing your application. Claims have an understanding with the issuer and allow the claims of the user to be accepted only if the claims are issued by a trusted issuer. Authentication and authorization is explicit in CBAC as compared to other approaches. [1].

Keywords: *Claims; Authentication; Tokens; Identity Server; RBAC; Open ID; O Auth; Identity Tokens; Access Tokens;*

I. INTRODUCTION

To see the impact of claims, one may need to change their perspective of authentication. It's anything but difficult to give a specific authentication component a chance to compel your thinking. One can consider Identity as far as "Windows Authentication" of Ms. Windows or "username, passwords and roles" for ASP.Net. It is common that all the different authentication mechanisms are divided in two parts: "Claims and Issuer/Authority" [2]

Block Diagram :

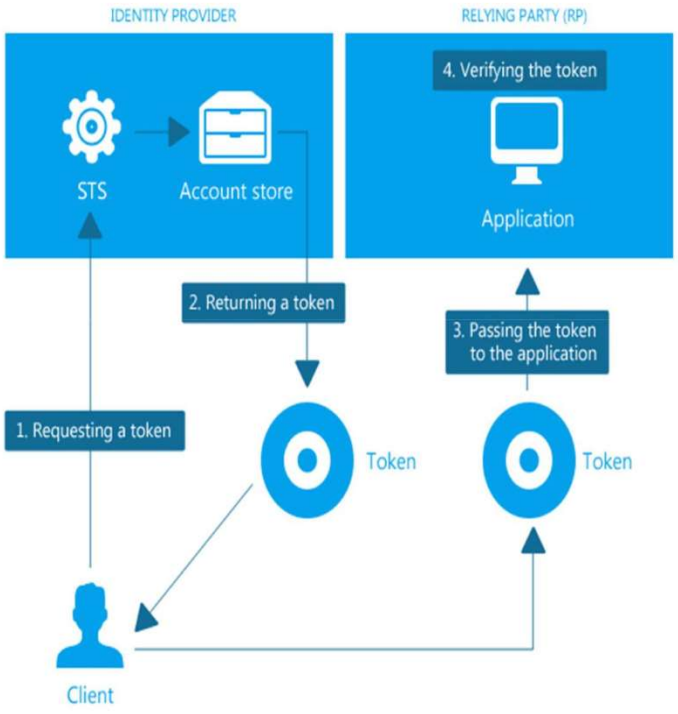

Fig 1: Token based Authentication

II. CURRENT SYSTEM:

Role-based access control (RBAC) is giving access to resources to users based on the roles in a given organization. Access can be anything like add/update records, etc. Roles are given on the bases of user's authorization, specialization, or responsibility. [5]

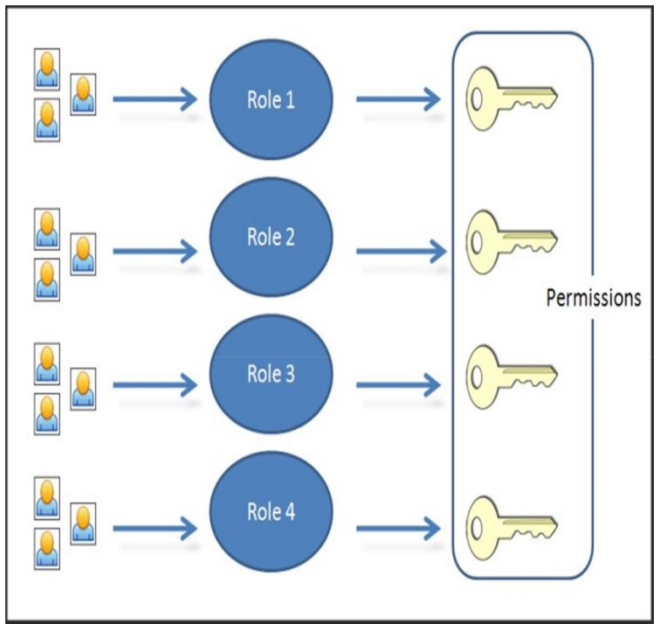

Fig 2: Role based Authentication

III. PROPOSED SYSTEM:

Claims-based access control (CBAC) is a process of authenticating access to the resources through claims via tokens issued by a legitimate issuer. This sort of access control does not contain any authentication rationale in itself but rather relies upon different administrations to give authentication to the application which all things considered lessens the multifaceted nature of the application itself. [3]

IV. HOW A TOKEN IS ISSUED:

There are numerous routes through which a token can be issued. As for the present situation we will look on Identity Server as the Middleware for authentication and issuing token.

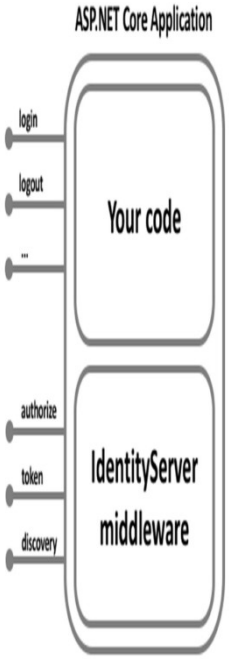

Fig3: Identity Server in Asp.Net Application

"Identity Server is middleware that includes the spec agreeable Open ID Connect and O Auth 2.0 endpoints to a subjective application". This means that the user only has to build login and logout page and all the authentication part will be taken care off by adding all the important protocols by the Identity Server middleware.

IdentityServer has a number of jobs and features including:

- Secure all the resources.
- Authentication of user via external identity providers or local database.
- Provide single sign-on and session management.
- Verify and manage clients
- Issue "Identity and Access tokens" to clients
- Authenticate tokens

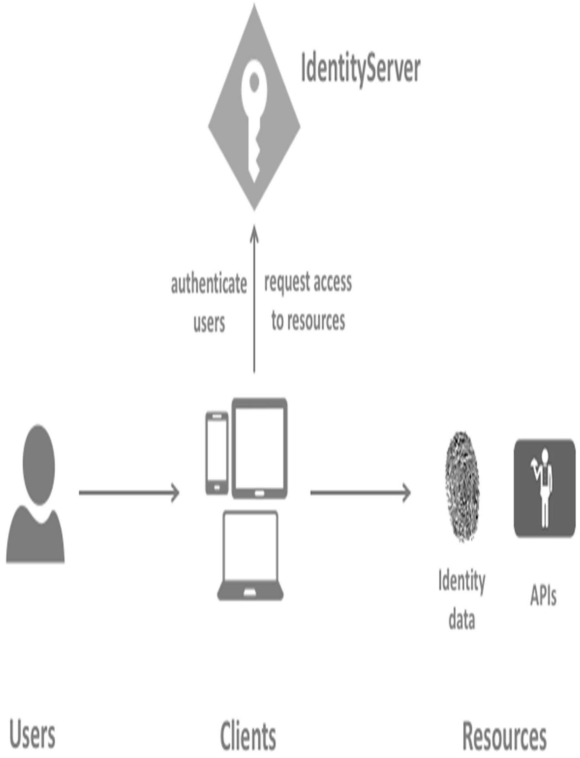

Fig4: IdentityServer

- **User:**

A user is anyone who requests resource through legitimate client.

- **Client:**

"A client is any application/software that requests tokens from Identity Server - either for user authentication or to request a service (requesting an access token)". For the client to request tokens it must be first registered and identified by the Identity Server.

A client can be anything from web applications to mobile applications, etc.

- **Resources:**

A resource is what the user wants to protect with Identity Server. Resource can be anything from user information or API's

Every resource is identified by a unique name and clients use this name to identify the requested resource.

- **Identity Token:**

An identity token represents all the information of user and how and when the user was authenticated. An Identity token can also contain other information regarding the user.

- **Access Token:**

Access token is used by the API to authorize access to their data. Access token is forwarded to the API after Identity token is authorized. [4]

V. ADVANTAGES

- **Outsourcing Authentication:**

By Outsourcing authentication CBAC removes all the authentication complexities from the application. This means that all the authentication logic and data of the users are stored and managed by external identity providers.

- **Extensibility:**

CBAC offers multiple attributes to be added to the claims to add more information to the claims.

- **Single Sign-On:**

CBAC uses single sign-on which allows users to signin only once and then the token is used among various applications to verify the user.

- **Federation Gateway:**

Federation gateway allows authentication through external identity providers like Google and Facebook.

VI. CONCLUSION:

CBAC enables new clients to use resources through Authorization rather than changing/making new roles each time another client needs to get to the resource. CBAC is broader authentication framework than old and insecure username-secret thing. Instead of saying yes or no in regard to authentication endeavor CBAC is more extensive – outside framework which can give out more data about client by making claims and placing these into marked tokens. Claims can be stacked over each other. This means various clients with various types of authentication can partake on a similar web application. Claims are Internet prepared. Since they utilize algorithms, for example, RSA, they are to a great degree secure and trustable.

VII. REFERENCES:

1. https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff359101(v=pandp.10)
2. https://en.wikipedia.org/wiki/Claims-based_identity
3. http://gunnarpeipman.com/aspnet/what-is-claims-based-authentication/
4. https://identityserver4.readthedocs.io/en/release/
5. https://en.wikipedia.org/wiki/Role-based_access_control