Autenticação e Autorização em Sistemas Web

Capítulo 2. Autenticação HTTP

Prof. Angelo Assis

Autenticação e Autorização em Sistemas Web

Aula 2.1. Tipos de Autenticação

Prof. Angelo Assis

Nesta aula

- Tipos de Autenticação HTTP.

Tipos de Autenticação

- Basic.
- Bearer.
- Digest.
- Outras.

Basic

- RFC 7617
- Autenticação incluída no cabeçalho de cada requisição
  - `Authorization: Basic <credenciais>`
- Deve ser usado apenas com conexão HTTPS (TLS)

Bearer Authentication

- RFC 6750
- Também conhecido como token authentication
- Cabeçalho:
  - `Authorization: Bearer <token>`
- O “Bearer” identifica recursos protegidos
- O token deve ser uma string
  - Ele representa uma autorização do servidor emitida para o cliente.

Outros

- Digest
  - RFC 7616
- Mutual (two-way)
- AWS4-HMAC-SHA256
  - AWS Signature

Próxima aula

- Codificação Base64.

Autenticação e Autorização em Sistemas Web

Aula 2.2. Codificação Base64

Prof. Angelo Assis

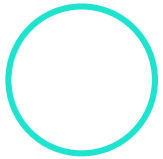
Nesta aula

- Codificação Base64.

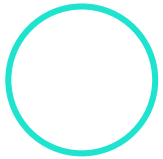
Base64

- É um método para codificação de dados.
- É constituído por 64 caracteres ([A-Z],[a-z],[0-9], "/" e "+")
- Exemplo:
  - Olá, mundo! ➔ T2zDoSwgbXVuZG8h

Codificação vs Criptografia



Criptografia: É a aplicação de um algoritmo que baseado em uma chave para transformar a estrutura da mensagem, de modo que, caso seja interceptada por um terceiro, não será compreendida.



Codificação: É a aplicação de um processo de conversão da mensagem com o intuito de que não seja compreendida. As regras de transformação serão definidas a fim de seja possível compreender o conteúdo da mensagem.

Codificação

- Tem o propósito de garantir que os dados possam ser consumidos por diferentes tipos de sistema
    - Exemplo: enviar dados binários em uma requisição.
- Transforma o formato da informação, usando métodos publicamente conhecidos e acessíveis.
- Deve ser facilmente revertido (decoded).
- O algoritmo não precisa de uma chave.
- O objetivo não é tornar a informação secreta.

Conclusão

- ✓ Codificação e Criptografia são diferentes.
- ✓ Ambos são fundamentais para o protocolo HTTP.

Próxima aula

- Autenticação Básica.

Autenticação e Autorização em Sistemas Web

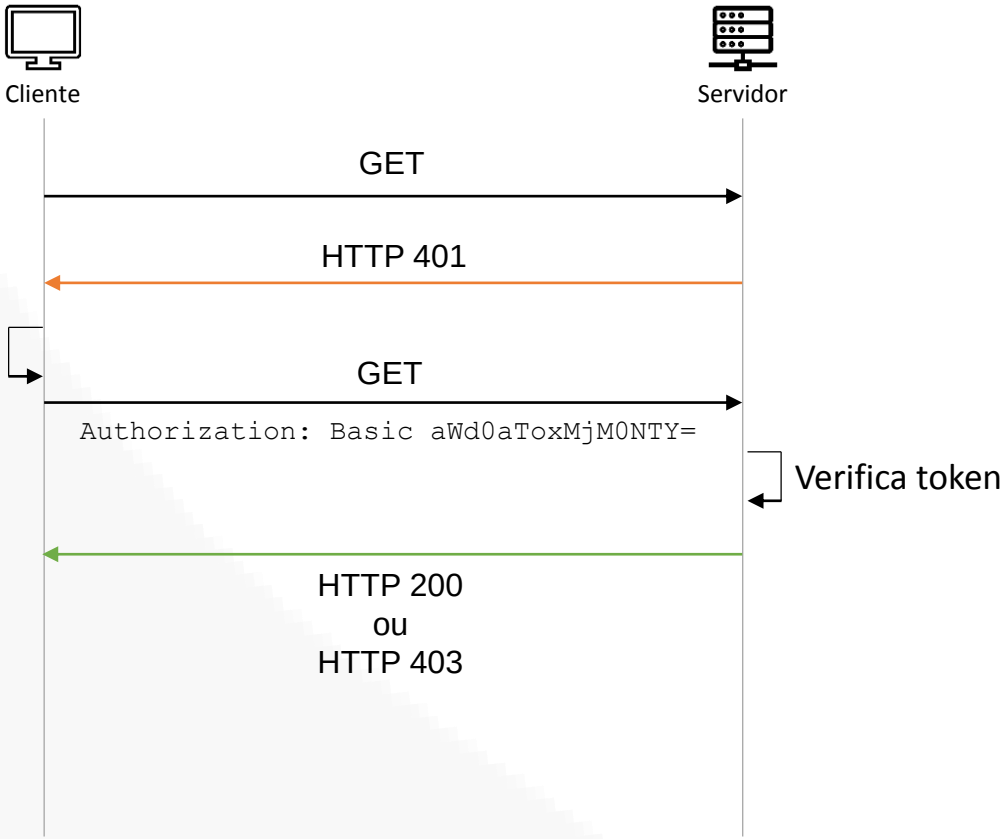
Aula 2.3. Autenticação Básica

Prof. Angelo Assis

Nesta aula

- Autenticação Básica.

Autenticação Básica



Autenticação Básica

- O Basic Authentication é o sistema de autenticação mais comum do protocolo HTTP.
- Header da requisição HTTP:
  - `Authorization: Basic <credenciais>`
- Credenciais:
  - Em Base64 no formato “usuário:senha”
- Importante: Base64 é um esquema de codificação e não criptografia. Deve ser usado somente com uma conexão HTTPS (TLS).

Conclusão

- ✓ É a técnica mais simples de controle de acesso.
- ✓ Não requer cookies.
- ✓ Em vez disso usa campos padrão no cabeçalho HTTP.
- ✓ Também não garante muita segurança ao aplicativo.

Próxima aula

- JSON Web Tokens.

Autenticação e Autorização em Sistemas Web

Aula 2.4. JSON Web Tokens

Prof. Angelo Assis

Nesta aula

- JSON Web Tokens.

JSON Web Token (JWT)

- O JWT é padrão da indústria para autenticação entre duas partes por meio de um token assinado.
- https://jwt.io/
- Amplamente usado para Autenticação e Autorização.

Estrutura do JWT

JWT possui três partes:

1. Header
2. Payload
3. Signature

O que gera um token similar a:

- xxxxx.yyyyy.zzzzz

Header do JWT

O cabeçalho do token possui duas informações:

1. O tipo do token (JWT)

2. O algoritmo utilizado (Ex.: HMAC, SHA256, RSA)

Exemplo:

```
{
    "alg": "HS256",
    "typ": "JWT"
}
```

O JSON é codificado em Base64 para formar a primeira parte do JWT.

Payload do JWT

É um objeto JSON com as informações.

As informações podem ser de 3 tipos:

1. Reserved claims: atributos não obrigatórios (mas recomendados) usados na validação do token.
2. Public claims: atributos que usamos em nossas aplicações. Geralmente são as informações do usuário autenticado.
3. Private claims: atributos definidos especialmente para compartilhar informações entre aplicações.

Payload do JWT

- Exemplos de “reserved claims”:
  - sub (subject): Entidade a quem o token pertence, normalmente o id do usuário;
  - iss (issuer): Emissor do token;
  - exp (expiration): Timestamp de quando o token irá expirar;
  - iat (issued at): Timestamp de quando o token foi criado;
  - aud (audience): Destinatário do token, representa a aplicação que irá usá-lo.
- Exemplo de payload:

```
{
    "sub": "1234567890",
    "name": "John Doe",
    "admin": true
}
```

Signature do JWT

É a concatenação dos hashes gerados a partir do Header e Payload usando base64UrlEncode, com uma chave secreta

```
HMACSHA256(
    base64UrlEncode(header)
    + "."
    + base64UrlEncode(payload),
    "secret"
)
```

É utilizada para garantir a integridade do token.

JWT

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkFuZ
2VsbyBBc3NpcyIsImlhdCI6MTYwOTQ3MDAwMH0.
EoPaAKUbtu_JGaquCmSu0UlFBSlkO4yR4pFXxZN
qB7c
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

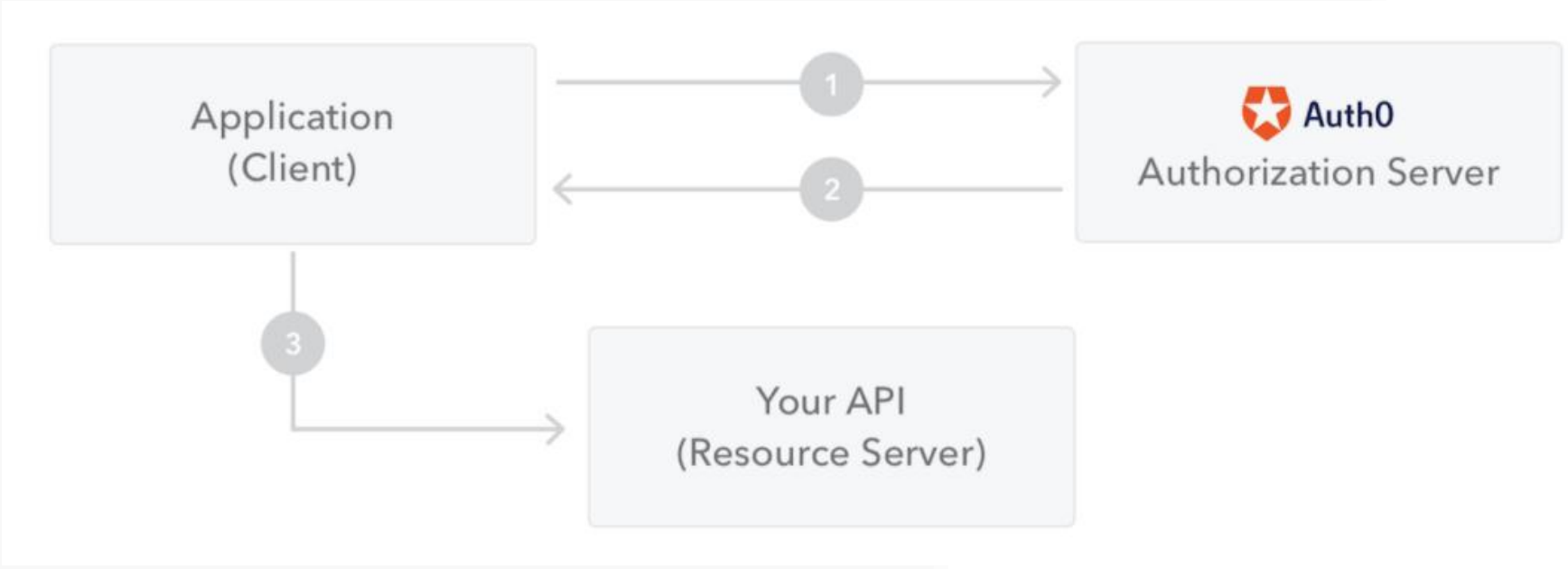
PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "Angelo Assis",
  "iat": 1609470000
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  IGTI
) ☐ secret base64 encoded
```

Fluxo de autenticação com JWT



JSON Web Token

Conclusão

- ✓ Basic Authentication e Bearer Authentication são técnicas de controle de acesso.
- ✓ JWT é um token seguro que pode ser usado em Bearer Auth.

Próxima aula

- Criptografia.