



Quem se prepara, não para.

Arquitetura de Aplicações Web

5º período

Professora: Michelle Hanne

Sumário

- Confidencialidade e Integridade
- Vulnerabilidades mais exploradas na Web
- JSON WEB TOKEN (JWT)
- Autenticação e Autorização

Confidencialidade e Integridade

Conforme descrição feita pela norma ISO/IEC 17799, a proteção da informação é vital, sendo caracterizada pela trilogia CID, ou seja, Confidencialidade, Integridade e Disponibilidade.

Confidencialidade e Integridade

Confidencialidade

- Garante que somente pessoas autorizadas poderão acessar as informações.
- Trata-se da não permissão da divulgação de uma informação sem prévia autorização.

Disponibilidade

- Garante acesso a uma informação no momento desejado. Isso implica no perfeito funcionamento da rede e do sistema. Imagine você necessitando de umas informações para concluir um relatório e o sistema não está funcionando!

Integridade

- Garante que a exatidão e completeza das informações não sejam alteradas ou violadas. Um exemplo, vamos supor que um gerente de uma empresa determina aumento de salário de 2% aos funcionários, para isso, utilizou seu e-mail para o departamento financeiro. Alguém interceptou e alterou de 2% para 20% o aumento!!!

Ataque, *Exploit* e Código Malicioso

Ataque – qualquer tentativa, bem ou mal sucedida, de acesso ou uso não autorizado de um serviço, computador ou rede.

Exploit – programa ou parte de um programa malicioso projetado para explorar uma vulnerabilidade existente em um programa de computador.



Código Malicioso – termo genérico usado para se referir a programas desenvolvidos para executar ações danosas e atividades maliciosas em um computador ou dispositivo móvel.

- Tipos específicos são: vírus, worm, bot, spyware, backdoor, cavalo de troia e rootkit.



Tipos de Ataque

Engenharia social – técnica por meio da qual uma pessoa procura persuadir outra a executar determinadas ações.

Varredura em redes (*scan*) – consiste em efetuar buscas minuciosas em redes, com o objetivo de identificar computadores ativos e coletar informações sobre eles como, por exemplo, serviços disponibilizados.

Negação de serviço distribuída (DDoS) – atividade maliciosa, coordenada e distribuída, pela qual um conjunto de computadores e/ou dispositivos móveis é utilizado para tirar de operação um serviço, um computador ou uma rede conectada à Internet.

Força bruta – consiste em adivinhar, por tentativa e erro, um nome de usuário e senha de um serviço ou sistema.

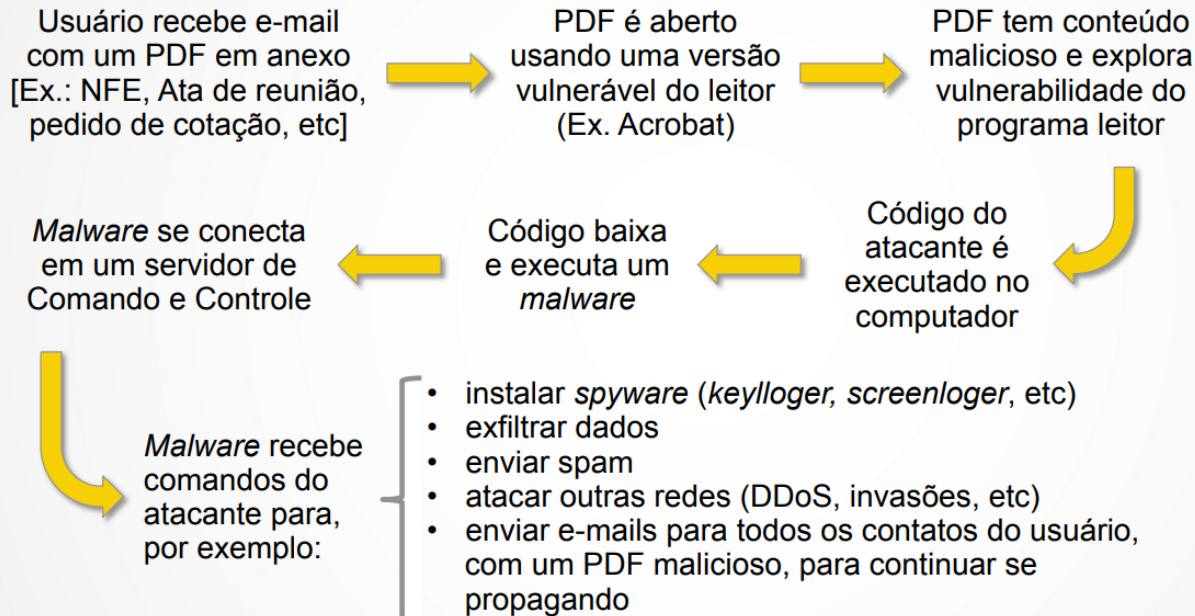
Invasão ou comprometimento – ataque bem sucedido que resulte no acesso, manipulação ou destruição de informações em um computador.

Desfiguração de página (*Defacement*) – consiste em alterar o conteúdo da página *Web* de um *site*.

Escuta de tráfego – consiste em inspecionar os dados trafegados em redes de computadores, por meio do uso de programas específicos.



Cenário: Ataque Contra Usuários de Internet



Cenário: Ataque Contra Servidores Web

Atacante instala ferramentas em um *site* já comprometido



Varre a Internet em busca de *sites* com sistemas CMS (Wordpress, Joomla, etc)

Em cada *site* realiza um ataque de força bruta de *logins* e senhas



Constrói uma lista de *sites* a serem atacados



Ao conseguir acesso ao *site* pode, entre outras coisas:

- alterar o seu conteúdo (*defacement*)
- desferir ataques contra outros sistemas ou redes (como DDoS, enviar *spam*, tentar invadir outros sistemas, etc)
- levantar páginas de *phishing*
- inserir *scripts* maliciosos, que exploram vulnerabilidades dos navegadores dos visitantes do *site*, com o objetivo de infectar os usuários (ataques de *drive-by*)
- instalar suas ferramentas e iniciar a busca por outros *sites* com CMS para reiniciar o ciclo do ataque

Vulnerabilidades de aplicações web

O tema da segurança em aplicações web é vasto e requer muito conhecimento. Nesse sentido, este infográfico irá apresentar as cinco principais vulnerabilidades dessas aplicações e suas características.



Cross-site Scripting (XSS)

O Cross-site Scripting (XSS) consiste em uma vulnerabilidade que tem como objetivo explorar a falta de verificação de dados de entrada e saída da aplicação web.

Esse tipo de vulnerabilidade permite aos atacantes executarem scripts no navegador da vítima de modo a roubar sessões dos usuários e redirecioná-los a sites maliciosos.



SQL injection

A vulnerabilidade SQL injection consiste em um tipo de ameaça de segurança que explora falhas em sistemas que integram bancos de dados. Essa vulnerabilidade ocorre quando o atacante realiza a inserção de uma série de instruções SQL dentro de uma consulta por meio da manipulação das entradas de dados de uma aplicação.

Após a injeção, o atacante pode manipular a base de dados, excluindo e alterando dados de forma maliciosa.





Cross-site Request Forgery

O Cross-site Request Forgery consiste em uma técnica em que o atacante engana o usuário com **falsas requisições HTTP** por meio de imagens, ou outra técnica, com o objetivo de **induzi-lo** a ser redirecionado ao seu site.

Muitas vezes, o usuário não está ciente de que uma requisição foi enviada. Dessa forma, o atacante pode fazer uso do Cross-site Request Forgery para obter **informações confidenciais** do usuário e utilizá-las para diversos fins.



Falha de autenticação e gerenciamento de sessão

A vulnerabilidade de Falha de autenticação e gerenciamento de sessão tem como objetivo explorar falhas de implementação dos mecanismos de autenticação e gerenciamento de sessão.

Esses elementos são de difícil construção e nem sempre são abordados de forma correta pelos desenvolvedores. Assim, esse tipo de falha permite ao invasor ter acesso a algumas contas ou até mesmo a todas as contas dos usuários.



Referência insegura direta a objeto

A vulnerabilidade de Referência insegura direta a objeto ocorre quando o desenvolvedor expõe uma **referência de um objeto de implementação interna**, tal como um arquivo, diretório, uma URL ou um parâmetro de um formulário.

Desse modo, o atacante explora uma dessas vulnerabilidades e acessa objetos sem a necessidade de qualquer autorização.



JSON WEB TOKEN (JWT)

O JSON Web Token (JWT) é um padrão aberto (RFC 7519), que tem por objetivo definir um modo compacto e independente, que pode ser enviado dentro de um cabeçalho HTTP e conter as informações do usuário, para a transmissão segura de informações entre cliente e servidor, através de um objeto JSON.

Várias bibliotecas que implementam o padrão JWT estão disponíveis para as mais variadas linguagens de programação, dentre elas: .NET, Python, NodeJs, Java, JavaScript, Perl, Ruby e PHP

JSON WEB TOKEN (JWT)

Algorithm

HS256



Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) ☐ secret base64 encoded
```

As bibliotecas podem ser usadas tanto para autenticação de usuários, que é o cenário mais comum, **sendo que a cada solicitação de uma rota, serviço ou recurso o usuário envia junto o token**, permitindo seu acesso ou não com base nesse token, quanto para transmitir informações de forma segura, uma vez que com base em sua **assinatura criptografada é possível verificar se o conteúdo não foi adulterado**.

Um token gerado pela aplicação tem **sua estrutura dividida em três partes, cada parte é separada por um ponto final (".")** e contém informações de diferentes tipos. A primeira parte é identificada como **header**, que é o **cabeçalho do token e contém informações a respeito da criptografia** usada para assinar o token.

```
{"alg": "HS256" XIII Encontro Anual de Computação - EnAComp 2017 - UFG 188 3. }
```

JSON WEB TOKEN (JWT)

A segunda parte é conhecida como **payload**, que é a parte onde ficam as informações no token, geralmente **informações do usuário que aquele token pertence e a sua validade**.

No token gerado as informações encontradas foram a **“iat”**, que é o momento em que foi gerado, contado em segundos desde às 00:00 de 01/01/1970. A informação de **“sub”** que define do subject do token, isso é, o número de identificação do sujeito para quem o token pertence. Já a informação de **“iss”** é o issuer que foi definido para identificar o nível de acesso daquele usuário à aplicação. A última informação encontrada nessa parte é a **“exp”** que define o tempo de expiração do token, usando a contagem de tempo da mesma maneira que o anterior.

Quadro 4. Payload do Token em JSON

```
1. {  
2.   "iat": 1491609751,  
3.   "sub": "2",  
4.   "iss": "client",  
5.   "exp": 1491717751  
6. }
```

Quadro 5. Payload do Token em base64

```
eyJpYXQiOiE0OTE2MDk3NTesInN1YiI6IjIiLCJpc3MiOiJjbGllbnQiLCJleHA  
iOiE0OTE3MTc3NTF9
```


JSON WEB TOKEN (JWT)

A terceira e última parte do token é a **signature** que guarda uma assinatura criptografada com base no algoritmo especificado no cabeçalho. Para criar a assinatura, **o algoritmo usa como base os dados das duas primeiras partes do token**, que são os dados do tipo da criptografia, **os dados de payload**, e uma chave definida ao gerar o token. Essa chave em especial é definida em texto puro e o algoritmo a converte para **base64**, o resultado da assinatura em binário também é convertido para base64

Quadro 6. Assinatura do Token criptografado em SHA256 e transformado em *base64*

VnEwyZdBzekCIycPoezPp_Hzi0sD4BmM_KOFnMyMhBo

Autenticação Usando JWT Web Token

A autenticação do usuário se dará por meio de um **envio de uma requisição HTTP**, do tipo **POST**, ao **servidor de aplicação**, o qual no corpo da requisição enviará o **usuário e senha para login**. Os dados serão **validados pelo banco de dados** e se coincidirem será gerado um **token** para aquele **usuário**. Esse **token** será enviado de **volta para o usuário**, o qual pode armazená-lo em **cookies** ou no armazenamento de sessão do dispositivo.

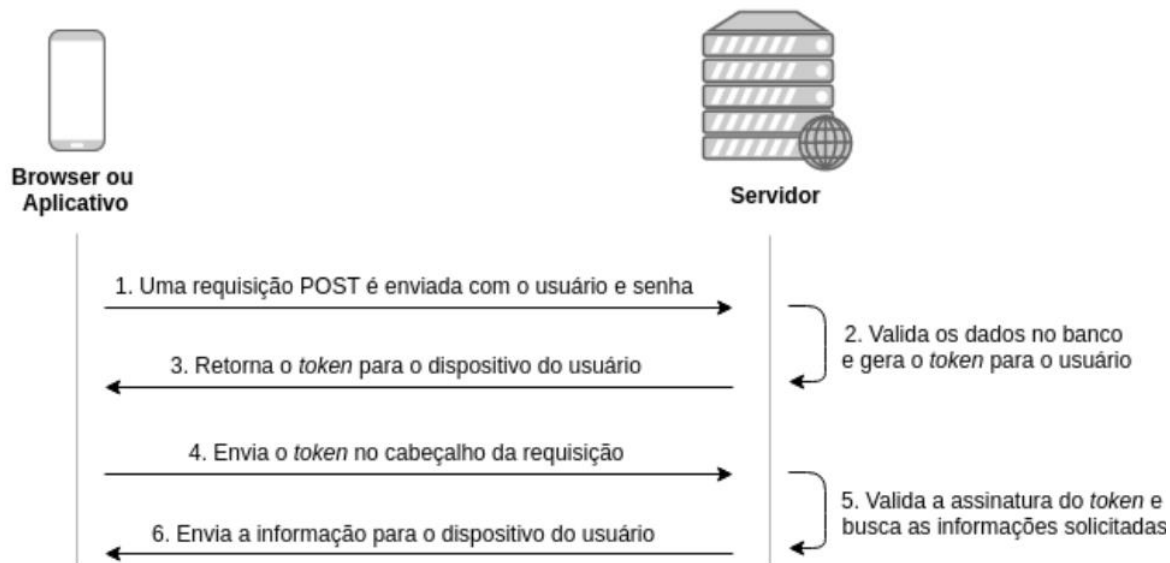


Figura 1. Diagrama de autenticação e validação de token

É possível implementar o **JSON Web Token** em uma aplicação através do serviço chamado **Auth0**, que implementa o JWT em seu código e libera à seus clientes uma API na qual o cliente se cadastra por meio do serviço, podendo inclusive usar contas de terceiros como Google, Facebook e Twitter, e é enviado para a aplicação apenas as informações do usuário, uma vez que a senha e outros dados são guardados com segurança pelo Auth0.

O serviço é utilizado por grandes empresas, como Nvidia, AMD, Mozilla, Dow Jones, entre outras [Auth0 2017]

- FERNANDES, Nélia O. Campo. Segurança da Informação, rede e-Tec Brasil, 2014. Disponível em: <http://proedu.rnp.br/bitstream/handle/123456789/1538/15.6_versao_Finalizada_com_Logo_IFRO-Seguranca_Informacao_04_04_14.pdf?sequence=1&isAllowed=y> . Acesso em: 15 mar. 2022.
- MONTANHEIRO, Lucas Souza, Carvalho, Ana Maria Martins, RODRIGUES, Jackson Alves. Utilização de JSON Web Token na Autenticação de Usuários em APIs REST, XIII Encontro Anual de Computação - EnAComp 2017 – UFG. Disponível em: <https://www.enacomp.com.br/2017/docs/json_web_token_api_rest.pdf>. Acesso em: 15 mar. 2022.