# A01:2021-Broken Access Control

## Exceto para recursos públicos, negar por padrão.

O middleware Authentication, responsável pelo sistema de login, já faz a maioria do trabalho. Os recursos que não precisam de autenticação, estão listados na constante src/Controller/UsersController.php::ACTIONS\_SEM\_AUTENTICACAO

## Implemente mecanismos de controle de acesso uma vez e reutilize-os em todo o aplicativo, incluindo a minimização do uso de Cross-Origin Resource Sharing (CORS).

Somente para o CRUD de Users é necessário um controle de autorização.

Código que aplica o controle: src/Controller/UsersController.php->beforeFilter().

CORS não é utilizado.

## Os controles de acesso ao modelo devem impor a propriedade do registro em vez de aceitar que o usuário possa criar, ler, atualizar ou excluir qualquer registro.

O KAW tem dois tipos de usuário, o comum e o root. A única diferença entre eles é que o root pode executar ações consideradas de configuração do sistema. O restante das permissões são iguais.

## Os requisitos de limite de negócios de aplicativos exclusivos devem ser impostos por modelos de domínio.

As validações de autenticação e autorização são feitas no controller, sempre levando em consideração o usuário autenticado no servidor. Nunca confiando nas informações enviadas do frontend.

## Desative a lista de diretórios do servidor da web e certifique-se de que os metadados do arquivo (por exemplo, o .git) e os arquivos de backup não estejam presentes nas raízes da web (web roots).

Em README > Configuração inicial da aplicação > Ambiente de produção > Item 2.1, ensina a configurar o Apache2 para considerar a pasta kaw/webroot/ como a raiz da aplicação.

O arquivo kaw/.htaccess faz o trabalho de redirecionar para a pasta webroot/

## Registrar falhas de controle de acesso e alertar os administradores quando apropriado (por exemplo, falhas repetidas).

Criada a Atividades Suspeitas que com base nos logs registrados, mostra para o usuário atividades danosas na aplicação.

Arquivos:

* Sistema de logs: kaw/src/Log/\*
* Atividades Suspeitas: kaw/templates/Pages/home.php (Home do sistema)

## Limite de taxa o acesso da API e do controlador para minimizar os danos do conjunto de ferramentas de ataque automatizado.

Foi criado o módulo de IPs Bloqueados. Quando um usuário erra consecutivamente as credenciais de acesso o IP é bloqueado.

## Os identificadores de sessão com estado devem ser invalidados no servidor após o logout. Os tokens JWT sem estado devem ter vida curta, para que a janela de oportunidade para um invasor seja minimizada. Para JWTs de longa duração, é altamente recomendável seguir os padrões OAuth para revogar o acesso.

Melhorado a segurança da Session e Cookies. Ler o arquivo kaw/config/app.examplo.php > ‘Session’

[**A02:2021-Cryptographic Failures**](https://owasp.org/Top10/A02_2021-Cryptographic_Failures/)

[falta implementar]

[**A03:2021-Injection**](https://owasp.org/Top10/A03_2021-Injection/)

[falta implementar]

[**A04:2021-Insecure Design**](https://owasp.org/Top10/A04_2021-Insecure_Design/)

[falta implementar]

[**A05:2021-Security Misconfiguration**](https://owasp.org/Top10/A05_2021-Security_Misconfiguration/)

[falta implementar]

[**A06:2021-Vulnerable and Outdated Components**](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

[falta implementar]

[**A07:2021-Identification and Authentication Failures**](https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/)

[falta implementar]

[**A08:2021-Software and Data Integrity Failures**](https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/)

[falta implementar]

[**A09:2021-Security Logging and Monitoring Failures**](https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/)

[falta implementar]

[**A10:2021-Server-Side Request Forgery**](https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/)

[falta implementar]