Component Security no CakePHP 3

class `SecurityComponent`(*ComponentCollection $collection*, *array $config* =[])

O Componente security cria uma maneira fácil de integrar uma segurança mais reforçada em suas aplicações. Ele oferece métodos para várias tarefas como:

- Restringir que métodos HTTP sua aplicação aceita
- Proteção contra adulteração dos formulários
- Requerer que SSL seja usado
- Limitando a comunicação cruzada para controllers

Pode ser configurado através do método beforeFilter.

Usando o Componente Security você automaticamente recebe proteção contra adulteração de formulários. Token oculto deve ser automaticamente inserido nos forms e checados pelo componente Security.

Se você está usando a proteção do Componente Security
Se você estiver usando recursos de proteção de forms do componente Security e outros componentes que processam dados de formulários em seus callback startup(), não se esqueça de colocar componentes Security antes desses componentes em seu método initialize().

Quando usar o componente Security você deve usar o FormHelper para criar seus formulários. Além disso, você não deve sobrescrever qualquer um dos atributos "nome" dos campos. O componente Security procura por certos indicadores que são criados e gerenciados pelo FormHelper (especialmente aqueles created em View\Helper\ FormHelper::create() e View\Helper\FormHelper::end()). Dinamicamente alterar os campos que são submetidos em um pedido POST (por exemplo, desabling, deleting ou creating novos campos via JavaScript) é susceptível de causar o request para ser enviado para a callback blackhole. Veja a $validatePost ou o parâmetro de configuração $disabledFields.

Você deve sempre verificar o método HTTP a ser utilizado antes de executar efeitos colaterais. Você deve verificar o método HTTP ou usar Cake\Network\ Request::allowMethod() para garantir que o método HTTP correto foi usado.

Usando o Componente Security

O uso do componente Security geralmente é feito no método beforeFilter() dos Controllers. Você deve especificar as restrições de segurança que você deseja e o componente Security deve reforçar então em seu startup():

```
namespace App\Controller;

use App\Controller\AppController;
use Cake\Event\Event;

class WidgetsController extends AppController
{

    public function initialize()
```

```
    {
        parent::initialize();
        $this->loadComponent('Security');
    }

    public function beforeFilter(Event $event)
    {
        if (isset($this->request->params['admin'])) {
            $this->Security->requireSecure();
        }
    }
}
```

O exemplo acima deve forçar todos os actions que tem rota admin para exigir seguras request SSL:

```
namespace App\Controller;

use App\Controller\AppController;
use Cake\Event\Event;

class WidgetsController extends AppController
{

    public function initialize()
    {
        parent::initialize();
        $this->loadComponent('Security', ['blackHoleCallback' => 'forceSSL']);
    }

    public function beforeFilter(Event $event)
    {
        if (isset($this->params['admin'])) {
            $this->Security->requireSecure();
        }
    }

    public function forceSSL()
    {
        return $this->redirect('https://' . env('SERVER_NAME') . $this->request-
>here);
    }
}
```

Este exemplo poderia forçar todas as ações que tiverem admin routing para exigir solicitações SSL seguras. Quando o pedido é preto furado/black holed, ele irá chamar o callback denominado forceSSL(), que irá redirecionar solicitações não seguras para proteger os pedidos automaticamente.

Proteção CSRF

CSRF ou Cross Site Request Forgery é uma vulnerabilidade comum em aplicações web. Ela permite que um atacante capture e reproduza um pedido anterior, e às vezes enviar solicitações de dados usando tags ou recursos de imagem em outros domínios. Para habilitar os recursos de proteção CSRF usar o *Request Forgery Cross Site*: http://book.cakephp.org/3.0/en/controllers/components/csrf.html

Desabilitando o Componente Security para Actions Específicos

Pode haver casos em que você deseja desativar todas as verificações de segurança para um action (ex. AJAX requests). Você pode "unlock" essas ações, listando-os em $this->Security->unlockedActions em seu beforeFilter(). A propriedade unlockedActions não afetará outras características do SecurityComponent:

```
namespace App\Controller;

use App\Controller\AppController;
use Cake\Event\Event;

class WidgetController extends AppController
{

    public function initialize()
    {
        parent::initialize();
        $this->loadComponent('Security');
    }

    public function beforeFilter(Event $event)
    {
         $this->Security->config('unlockedActions', ['edit']);
    }
}
```

Este exemplo deve desabilitar todos os cheques de segurança para o action edit.

Mais
http://book.cakephp.org/3.0/en/controllers/components/security.html
http://book.cakephp.org/3.0/en/core-libraries/security.html
http://book.cakephp.org/3.0/en/controllers/components/csrf.html