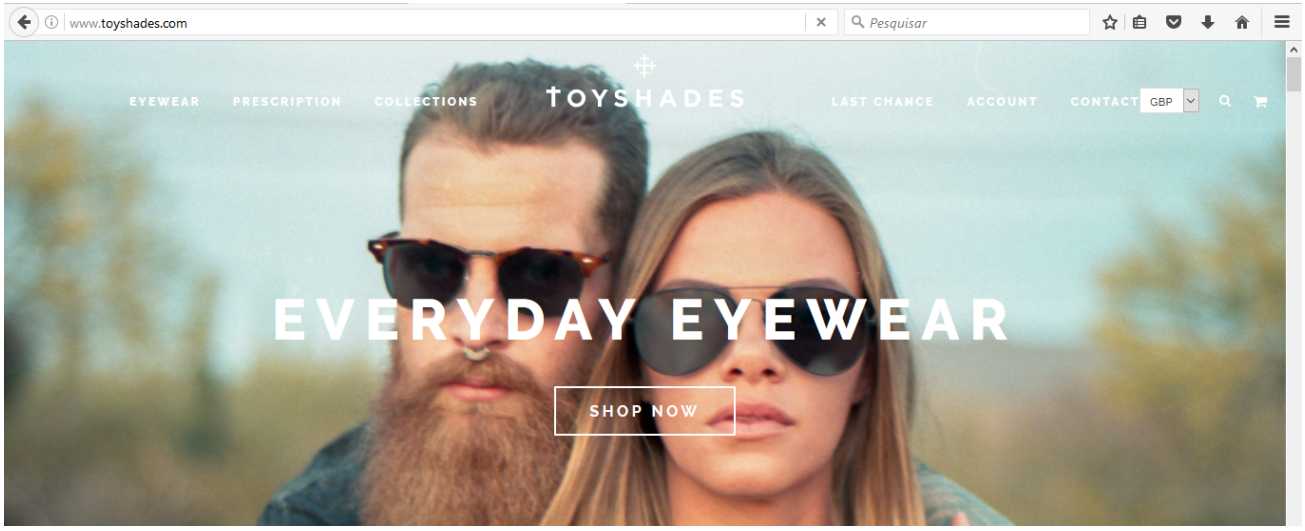
01

Falhas de segurança

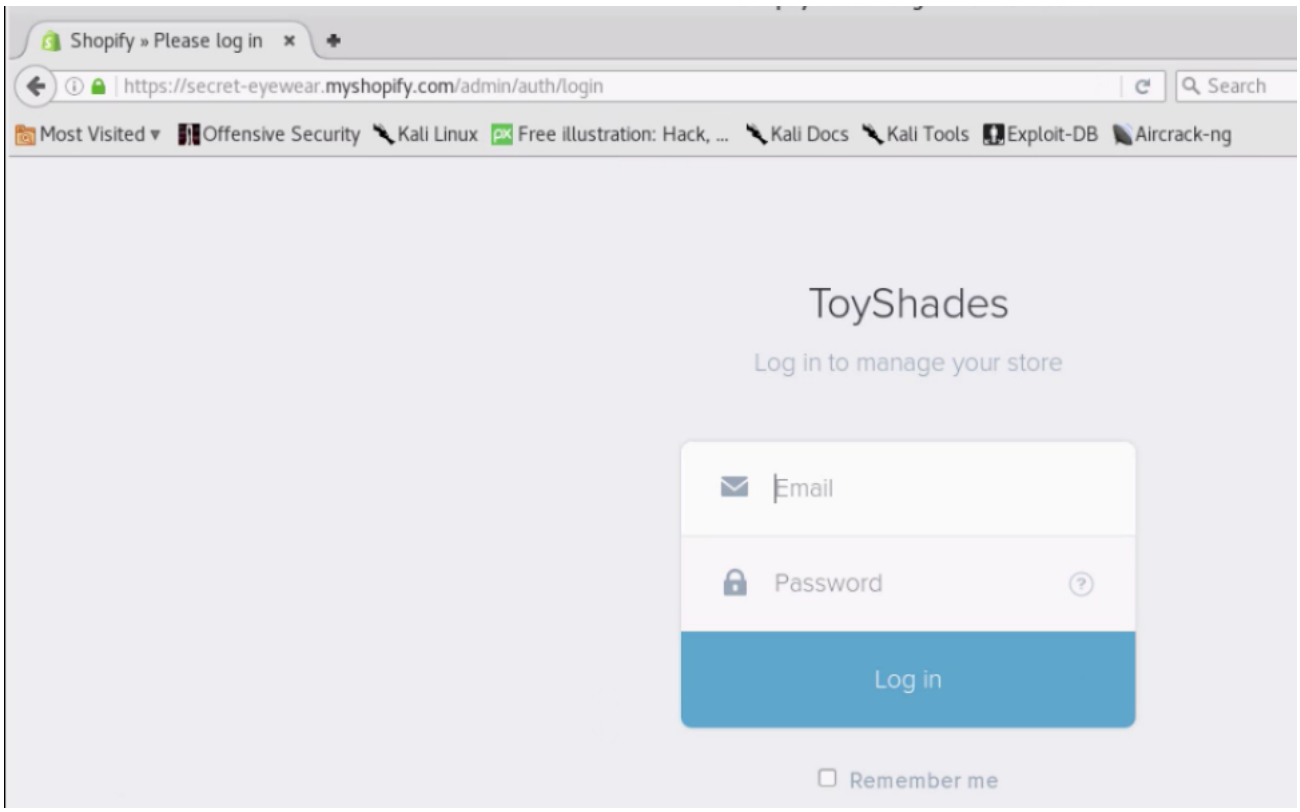
Transcrição

Nesta aula, vamos analisar outros tipos de ataques que podem ser realizados em ambientes da web. Vamos utilizar como exemplo uma loja virtual que faz uso da plataforma do *Shopify*. Primeiro, observamos o seguinte site:



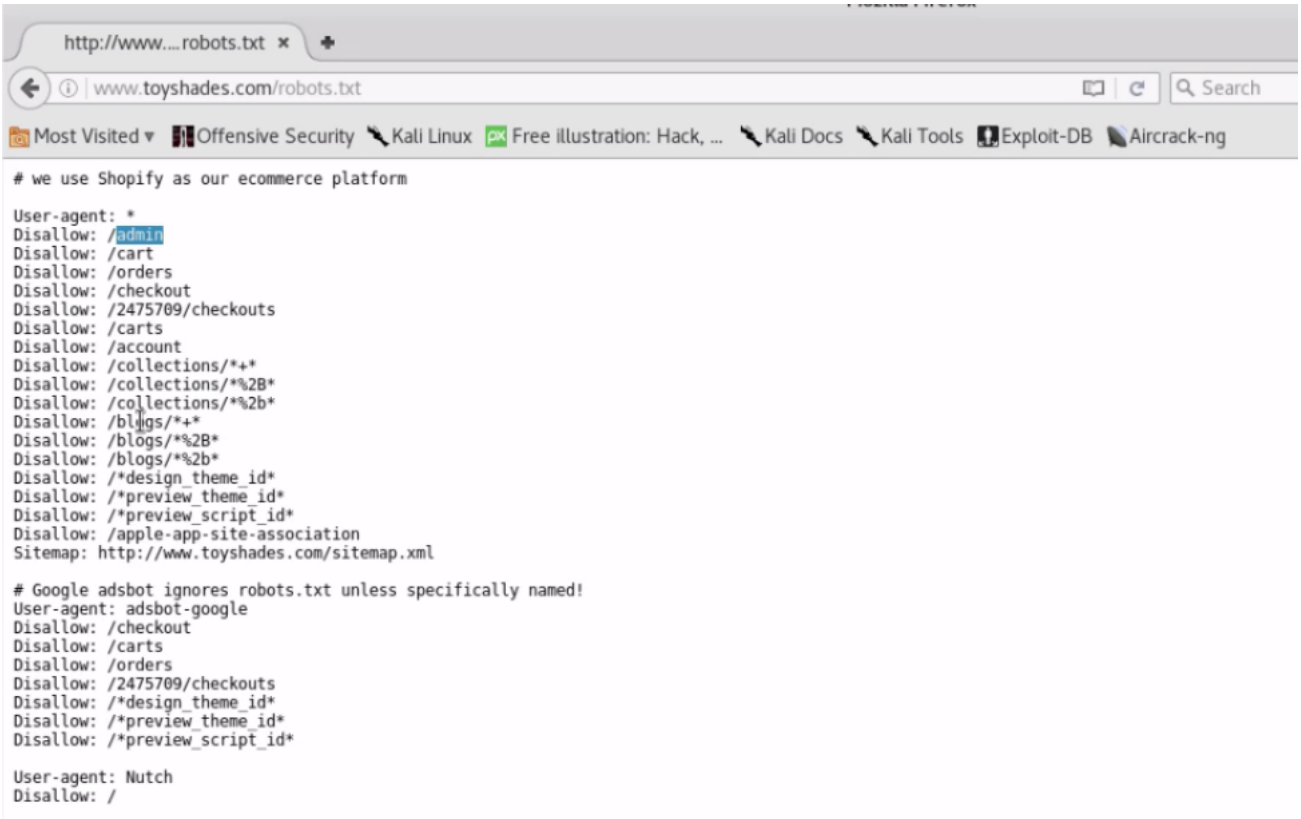
Este link corresponde a uma loja especializada em vender óculos. Para que o usuário acesse produtos e seus respectivos preços é preciso existir alguém que disponibilize e arrume essas informações para nós.

Um usuário mal intencionado pode tentar acessar a parte do site onde as informações são inseridas, isto é, uma página administrativa. Junto a URL principal, a `www.toyshades.com/` , podemos inserir palavras que normalmente são utilizadas para indicar links administrativos. Por exemplo, depois da barra acrescentamos: `administrador` , `administrator` e `adm` . Apenas a última alternativa funciona e acessando a `www.toyshades.com/admin` temos o seguinte:



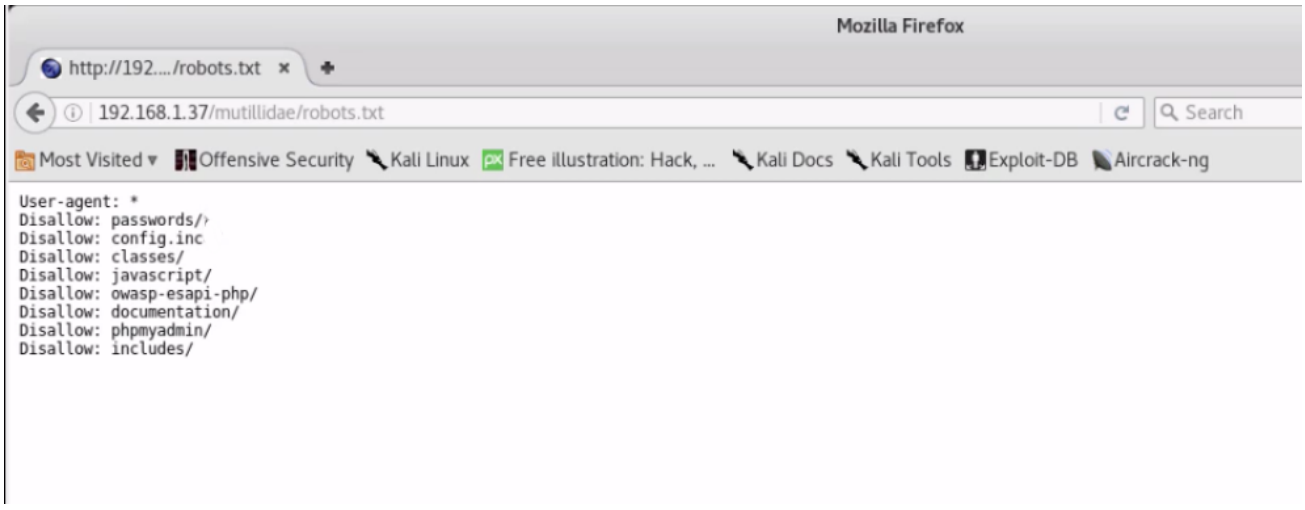
Para acessar essa parte administrativa é preciso um usuário e senha cadastrados e assim o sistema verifica permissão de acesso da página. Se não houver checagem qualquer pessoa pode entrar na parte administrativa e fazer o que quiser com ela. Como desenvolvedores podemos escolher esconder algumas páginas dos mecanismos de busca.

Por questões estratégicas da empresa podemos escolher, por exemplo, que a página do `admin` não seja indexada. Para fazer isso utilizamos o `robots.txt` . Quando digitamos na URL `www.toyshades.com/robots.txt` temos o seguinte:



Esta lista mostra endereços não indexados. O problema disso é que algumas destas páginas não possuem sistemas de autenticação tornando-as alvos de possíveis invasões. Portanto, como estratégia de segurança é preciso que páginas restritas sempre possuam sistemas de autenticação.

Agora, vamos acessar o site da *Multillidae* e verificar se existem páginas a ela vinculadas, as quais os desenvolvedores não indexaram na busca. Para observar isso podemos digitar no navegador `192.168.1.37/multillidae/robots.txt` e teremos :

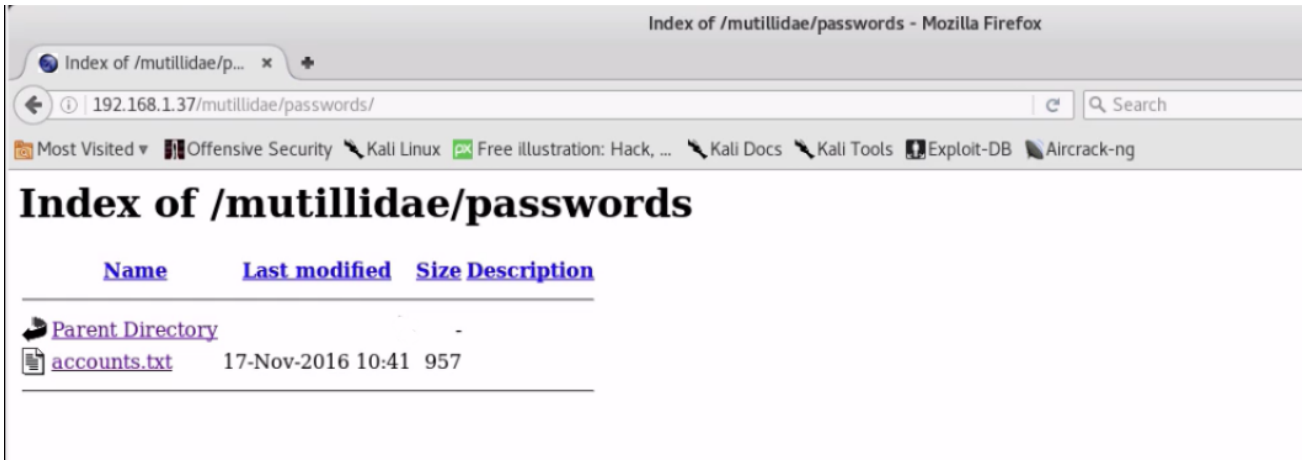


O primeiro item que aparece na lista é o `/passwords` que significa, literalmente, senhas. É provável que o usuário não tenha acesso a essa parte do site e que o desenvolvedor tenha inserido nela algum tipo de verificação. Mas, será que o

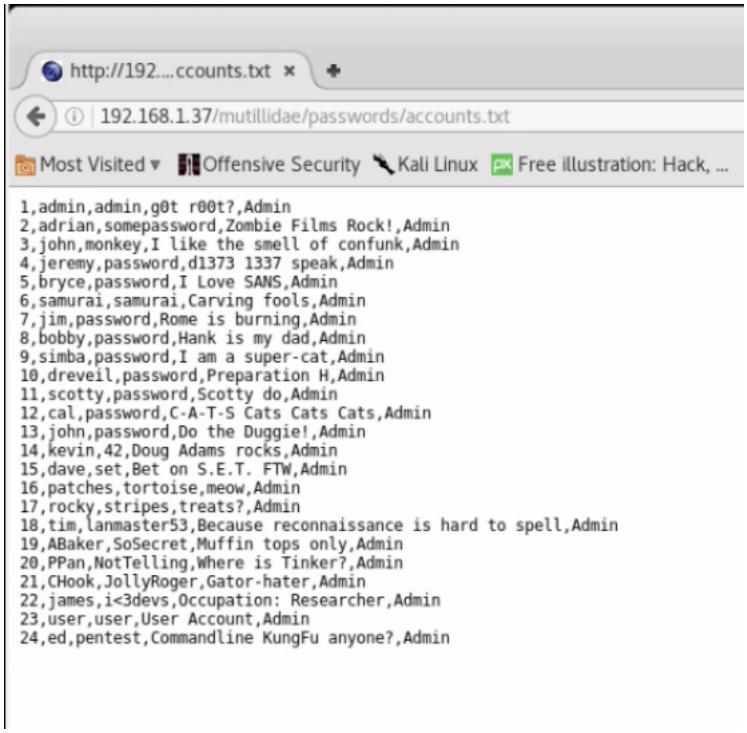
desenvolvedor realmente adotou essa precaução? Para verificar isso podemos tentar acessar o seguinte:

```
192.168.1.37/multillidae/passwords
```

Abrirá a seguinte página:



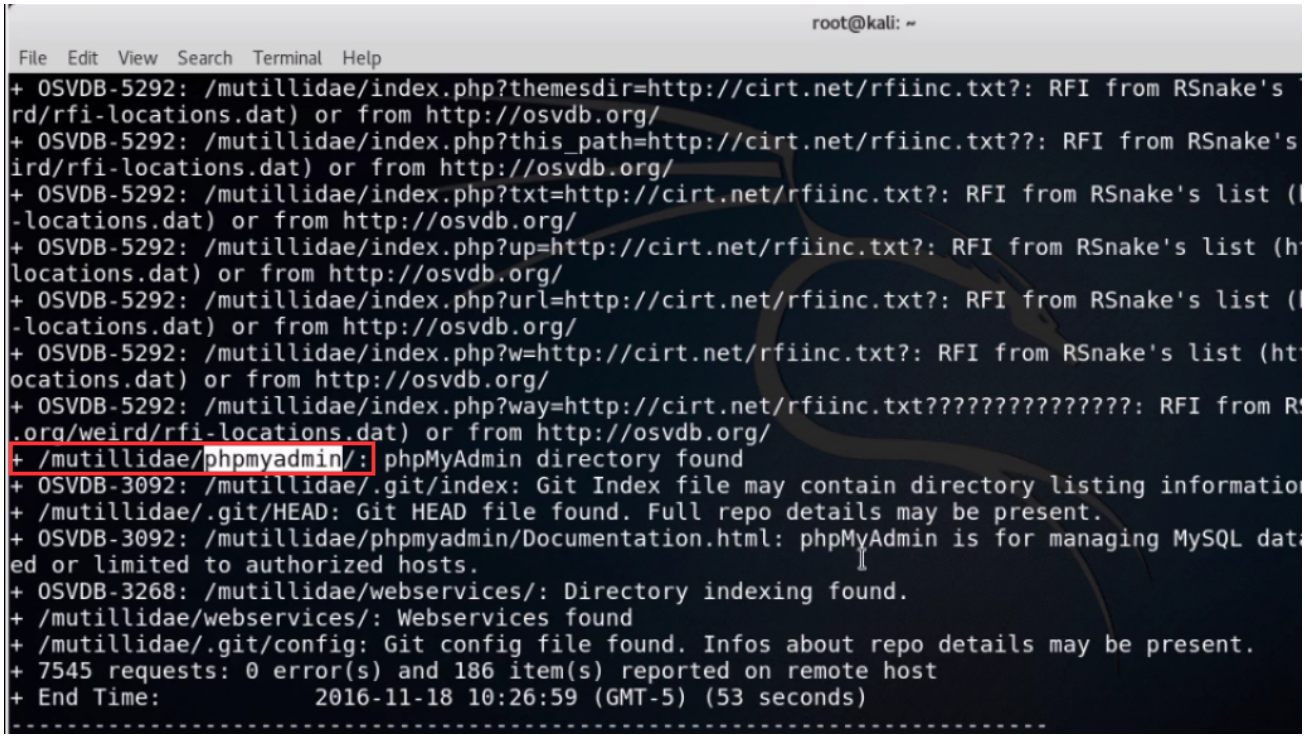
Ou seja, o desenvolvedor não se precaveu, pois temos acesso as senhas. Ele pensou que não seria necessário inserir uma autenticação nesse trecho do site. Ainda, clicando no `accounts.txt`, temos acesso a diversos dados privados dos clientes:



Existe uma ferramenta que nos auxilia neste tipo de mapeamento, a **nikto**. Ela é utilizada para verificar o `host`, portanto, no Terminal, vamos escrever `nikto -h` e inserimos a URL da página sobre a qual desejamos fazer uma busca:

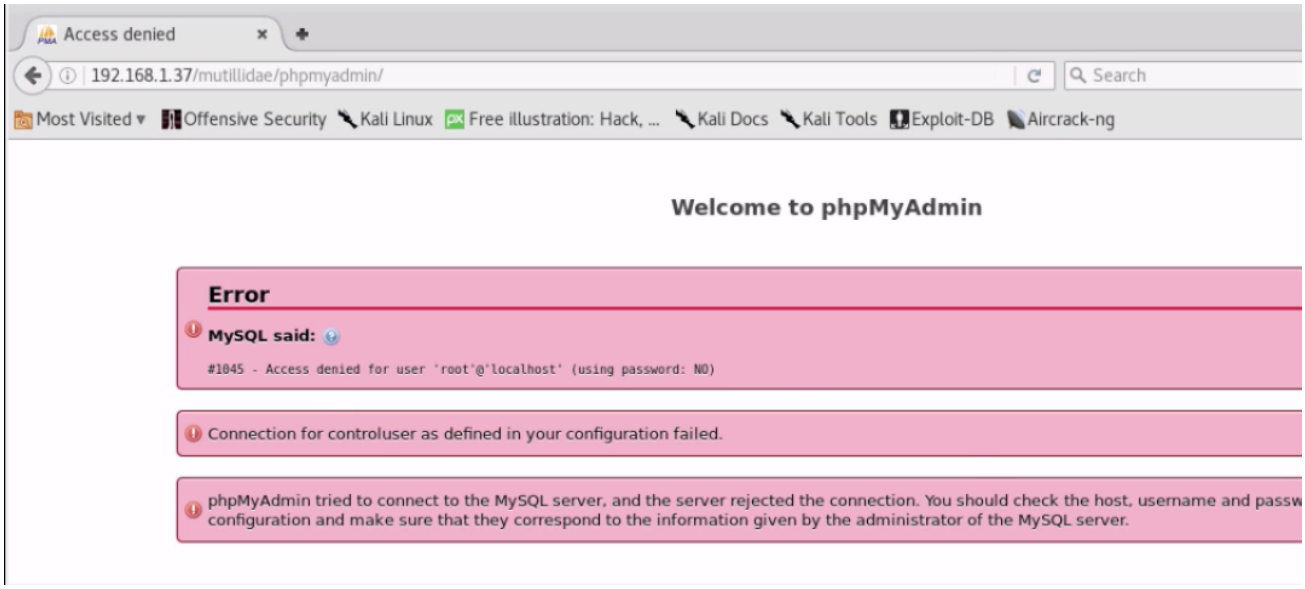
```
> nikto -h "http://192.168.1.37/multillidae/"
```

Dando um "Enter" nisso a `nikto` começa a realizar um mapeamento e assim que é finalizado temos todas as páginas encontradas pela ferramenta:

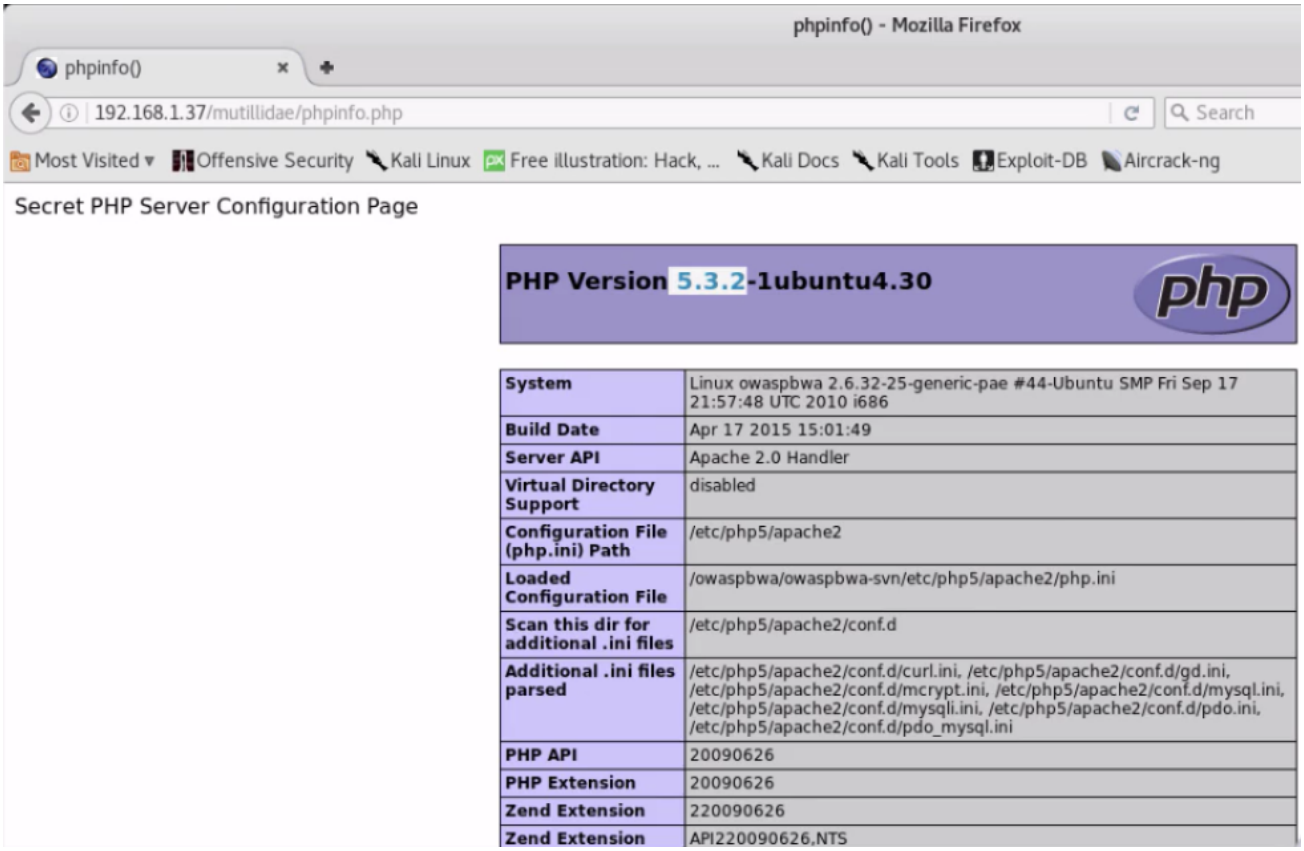


Uma das páginas mapeada chama atenção, a `multillidae/phpmyadmin/` . Ao lado dela aparece um pequeno resumo sobre o link, trata-se de uma página administrativa voltada para o Banco de dados do MySQL. Teoricamente, devido a sua importância, essa página não deve estar disponível para acesso de qualquer pessoa. Vamos testar!

Ao digitarmos no navegador `192.169.1.37/multillidae/phpmyadmin` , veremos o seguinte resultado:



O acesso é de fato negado, portanto, o desenvolvedor realmente se precaveu. Mas, ao acessarmos o `192.169.1.37/multillidae/phpinfo.php` conseguimos entrar na página:



Neste outro link temos acesso a diversas informações relevantes, pois, o desenvolvedor acabou não se dando ao trabalho de esconder o que está aí descrito. Essas informações, por exemplo, a **Apache Version** são o suficiente para um futuro ataque!