Resultados detalhados

Nessa sessão, serão relatados, de forma detalhada, os resultados obtidos na execução dos testes em cada iteração planejada. Serão apresentados os passos utilizados para detectar possíveis vulnerabilidades, as vulnerabilidades encontradas e quais danos estas podem causar caso sejam exploradas.

1. Iteração 1

**1.1. Tela Login - Exploração via Força Bruta**

Tarefa realizada dia 01/04/16.

Duração 1hora e 20 minutos.

**Objetivo**

Utilizar a ferramenta Hydra para adivinhar, por tentativa e erro, um login de um usuário e senha válidos para efetuar login no Sistema de Vacinas para verificar se a tela de Login não é vulnerável a ataques de força bruta.

**Resultados**

**Tentativa 1**

Configurar a Hydra com 3 parâmetros para que ela realize os testes de acesso com força bruta na tela de Login.

**Impedimento**: Hydra não está reclamando de 3 parâmetros quando, a tela de Login só utiliza 2.

**Resultado**: Indefinido.

**Tentativa 2**

Inserir um login de usuário válido e possíveis senhas mais comuns utilizadas para login de acesso e configurar número de 16 tentativas de acesso na Hydra.

*Comando utilizado: hydra -l admin -P forcaBruta 192.168.0.58 http-post-form “vacina/login.php:login=^USER^:senha=^PASS^:logado=false”*

No comando acima:

1. “-l ” significa o login utilizado no momento. Nesse caso, deve ser sempre “admin”, pois é o único usuário válido fornecido para o acesso ao Sistema de Vacinas;
2. “-P ” aponta para o arquivo contendo as senhas que a Hydra tentará utilizar para fazer o login;
3. 192.168.0.58 é o IP onde está hospedada a aplicação alvo;
4. “http-post-form” é o método utilizado, no caso , POST do HTTP;
5. *“vacina/login.php:login=^USER^:senha=^PASS^:logado=false”* é a URL utilizada para fazer a requisição.
   1. USER é onde a Hydra deve inserir o login válido usado na requisição;
   2. PASS é onde a Hydra deve inserir a senha válida usada na requisição

**Resultado:** Passou no teste.

**Tentativa 3**

Conseguir configurar a ferramenta para executar 16 testes por minuto realizando tentativas com o login “admin”, cadastrado na base de dados do sistema, e com um dicionário de senhas retiradas da *Lista de Piores Senhas de 2015* da SplashData ¹.

*Comando utilizado: hydra -l admin -P forcaBruta 192.168.0.58 http-post-form “vacina/login.php:login=^USER^:senha=^PASS^:logado=false” -V*

No comando acima:

1. “ -l ” significa o login utilizado no momento. Deve ser sempre admin, pois é o único usuário válido fornecido para acessar o Sistema de Vacinas;
2. “ -P ” aponta para o arquivo contendo as senhas que a Hydra tentará utilizar;
3. 192.168.0.58 é o IP onde está hospedada a aplicação;
4. “http-post-form” é o método utilizado, no caso , POST do HTTP;
5. *“vacina/login.php:login=^USER^:senha=^PASS^:logado=false”* é a URL utilizada para fazer a requisição;
   1. USER é onde a Hydra deve inserir o login válido usado na requisição;
   2. PASS é onde a Hydra deve inserir a senha válida usada na requisição.
6. “-V” é para exibir no console o resultado da operação.

**Resultado:** Não passou no teste. A execução do teste mostrou que o sistema é vulnerável ao acesso via força bruta. A Figura 2 mostra que embora várias tentativas tenham sido realizadas sem sucesso de login, o sistema permite que várias requisições sejam feitas ao servidor por tempo indeterminado. Como não há mecanismo de bloqueio para inúmeras requisições de login, o sistema abre uma brecha para que senhas de acesso ao sistema possam ser descobertas após várias tentativas.

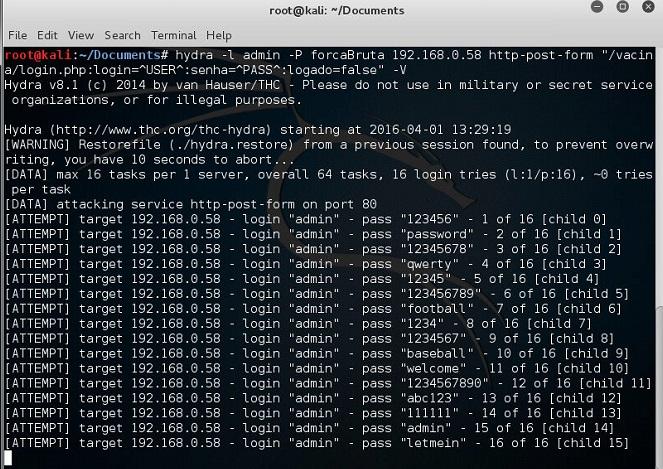


Figura 2 - Resultado da operação

**Referências**

1 - [Piores Senhas 2015](https://www.teamsid.com/worst-passwords-2015/) - <https://www.teamsid.com/worst-passwords-2015/>

<https://www.teamsid.com/worst-passwords-2015/>

**1.2. Tela Login - Exploração via Path Transversal**

Tarefa realizada entre os dias 31/03/16 e 01/04/16.

Duração: 2 horas e 30 minutos.

Um scanner foi realizado através da ferramenta Burp Suite para identificar a estrutura de pastas do sistema dentro do servidor. Os resultados obtidos conforme mostra a figura 3 [anexar figura], foram utilizados para executar tentativas de acesso a partes do sistema operacional do servidor que não deveriam ser acessíveis.

**Objetivo**

Através dos parâmetros utilizados para realizar login enviados pela URL, tentar acessar as páginas do Sistema de Vacinas ou do sistema operacional do servidor que não deveriam ser acessadas pela aplicação.

Todos os testes a executados nessa atividade têm como objetivo navegar em diretórios da aplicação dentro do servidor e dessa forma conseguir acesso a locais não autorizados. “Serão utilizados os seguintes comandos “../../” para tentar realizar a navegação de pastas.

**Resultados**

**Tentativa 1**

Quando o sistema não consegue efetuar um login, o seguinte parâmetro é exibido na url: *vacina/login.php?****logado****=false.*

A primeira tentativa foi fazer uma requisição ao servidor alterando a url citada anteriormente para: v*acina/login.php?logado=../../dir,* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “../../” foi utilizado para tentar a navegação de forma forçada dentro da estrutura de pastas do sistema operacional do servidor no qual a aplicação está hospedada.
2. O comando *dir*  no cmd do Windows serve para listar os arquivos e diretórios dentro da pasta atual.

**Resultado**: Passou no teste.

**Tentativa 2**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?logado= ../../cat /etc/passwd,* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “../../” foi utilizado para tentar a navegação de forma forçada dentro da estrutura de pastas do sistema operacional do servidor no qual a aplicação está hospedada.
2. O comando *cat /etc/passwd* , no sistema operacional Linux, serve para listar o conteúdo de passwd (local onde ficam armazenados informações de senha dos usuários). Como o sistema operacional do servidor em questão é Windows, então o esperado seria um erro.

**Resultado**: Passou no teste.

**Tentativa 3**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?logado=%2e%2e%2f%2e%2e%2f (código equivalente a ../../),* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “../../” servem para tentar a navegação de forma forçada dentro da estrutura de pastas do sistema operacional do servidor que hospeda o Sistema de Vacinas.

**Resultado**: Passou no teste.

**Tentativa 4**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?logado=../data.js*

para acessar o arquivo “data.js”, a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “../” serve para tentar navegar de forma forçada dentro da estrutura de pastas do Sistema Operacional.
2. “data.js” é um arquivo descoberto através do escaneamento utilizado para descobrir a estrutura de pastas da aplicação citado anteriormente**.**

**Resultado**: Passou no teste.

**Tentativa 5**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?logado=./././ ,* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “./.././” serve para tentar navegar de forma forçada dentro da estrutura de pastas do sistema operacional que hospeda a aplicação.

**Resultado**: Passou no teste

**Tentativa 6**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?login=teste&senha=./././,* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “./.././” serve para tentar navegar de forma forçada dentro da estrutura de pastas do sistema operacional que hospeda a aplicação.

**Resultado**: Passou no teste

**Tentativa 7**

Fazer uma requisição ao servidor utilizando a url *vacina/login.php?login=teste&senha=.. ,* a fim de verificar se o sistema não permite a navegação dentro das pastas armazenadas no servidor que hospeda o Sistema de Vacinas.

1. O comando “./.././” serve para tentar navegar de forma forçada dentro da estrutura de pastas do sistema operacional que hospeda a aplicação.

**Resultado**: Passou no teste

--------------------------------------------------------------------------------------------------------------------

**Tela Home - Exploração via Mapeamento**

Tarefa realizada no dia 05/04/16

Duração dos testes: 5 min.

Duração total da atividade: 15min.

**Objetivo**

Utilizar o Burp Intruder, funcionalidade da ferramenta Burp Suite, para mapear os parâmetros recebidos pela página Home para facilitar as futuras tentativas de invasão da página em questão.

**Resultados**

Não foi possível realizar mapeamento na Home pois ela é uma página estática e não faz uso de parâmetros no envio e resposta de requisições ao servidor. Essa página também não possui campos que permitam passagem de parâmetros. Os parâmetros são fundamentais para que o Burp Intruder consiga enxergar as demais partes do sistema e monte a hierarquia de pastas da aplicação.

**Tela Login - Mapeamento da hierarquia de pastas do sistema**

Tarefa realizada dia 30/03/16.

Duração dos testes:1 h.

Duração total da atividade: 1h 30 min.

**Objetivo**: Utilizar de forma automatizada o módulo Burp Scanner da Ferramenta Burp Suite para realizar um escaneamento de vulnerabilidades a partir da URL do Login (192.168.0.58/vacina/login.php). A partir dos resultados, verificar se foi possível ter acesso à hierarquia de pastas do sistema mesmo sem efetuar a requisição do login. Além disso, verificar se é possível ter acesso aos códigos da aplicação.

**Resultados**

**Tentativa 1**

Acessar a página do Login conforme URL informada anteriormente e tentar enxergar a hierarquia de pastas do sistema.

**Resultado**: Passou no teste

Conforme mostra imagem abaixo, não foi possível escanear estrutura de pastas do sistema apenas acessando o página do Login.

**Figura [numero] - Resultado tentativa mapeamento pastas do sistema**

**Tela Login - Exploração Via SQL Injection**

Tarefa realizada dia 07/04/16.

Duração dos testes:1 h.

Duração total da atividade: 1h 40 min.

**Objetivo**

Submeter entradas maliciosas no sistema através de injeção de códigos SQL na página de Login na tentativa de forçar ações não autorizadas no sistema.

**Resultados**

**Tentativa 1**

Verificar se o campo Login é vulnerável a injeção de código SQL fazendo uso da técnica de concatenação de aspas simples.

Espera-se que o sistema trate o caractere aspa simples como um parâmetro regular da consulta.

Entradas: Login = admin’

**Resultado**: Passou no teste. A aplicação está fazendo tratamento de dados de entrada não permitindo a injeção de instruções SQL através do campo de Login.

**Tentativa 2**

Inserir uma string com uma distribuição adequada de aspa simples e comentários inline como entrada no campo Login, a fim de modificar a instrução SQL utilizada pelo sistema para consulta de dados referentes ao login de usuários cadastrados. Comentários inline serão utilizados para tentar forçar o interpretador de SQL da aplicação a ignorar todas as intruções SQL que vierem após o comentário inserido.

Entradas:

admin’ or 1=1 #

admin’ or 1=1 --

O envio desses dados como entrada no campo do Login, tentará forçar o sistema a consultar não apenas os dados do usuário admin, mas qualquer usuário cadastrado na base de dados, uma vez que a instrução “1=1” será sempre verdadeira.

Para que a consulta alcance o objetivo proposto, será necessário utilizar possíveis instruções de que comentam linhas de código SQL em determinados SGBDs: “#” e “--” . Essas instruções podem ignorar qualquer instrução posterior presente na mesma linha em que elas foram inseridas.

**Resultado**: Passou no teste.

**Tentativa 3**

Inserir uma string fazendo uso de uma única aspa simples para completar a entrada de uma possível instrução SQL executada no momento em que função logar da página Login for invocadas. Nesse caso, será possível que uma instrução SQL de consulta semelhante a essa seja executada SELECT login from usuarios WHERE login = ‘admin’ or ‘a’=’a’.

Entradas:

admin’ or ‘a’=’a

**Resultado:** Passou no teste