Laboratório - Atacando um banco de dados MySQL

# Objetivos

Neste laboratório, você visualizará um arquivo PCAP de um ataque anterior contra um banco de dados SQL.

Parte 1: Abra o Wireshark e carregue o arquivo PCAP.

Parte 2: Veja o Ataque de Injeção SQL.

Parte 3: O Ataque de Injeção SQL continua...

Parte 4: O Ataque de Injeção SQL fornece informações do sistema.

Parte 5: O Ataque de Injeção SQL e Informações da Tabela

Parte 6: O Ataque de Injeção SQL Conclui.

# Histórico/Cenário

Os ataques de injeção SQL permitem que hackers mal-intencionados digitam instruções SQL em um site e recebam uma resposta do banco de dados. Isso permite que atacantes adulterem dados atuais no banco de dados, identidades falsas e malícia diversa.

Um arquivo PCAP foi criado para você exibir um ataque anterior contra um banco de dados SQL. Neste laboratório, você visualizará os ataques de banco de dados SQL e responderá às perguntas.

# Recursos necessários

* Máquina Virtual CyberOps Workstation

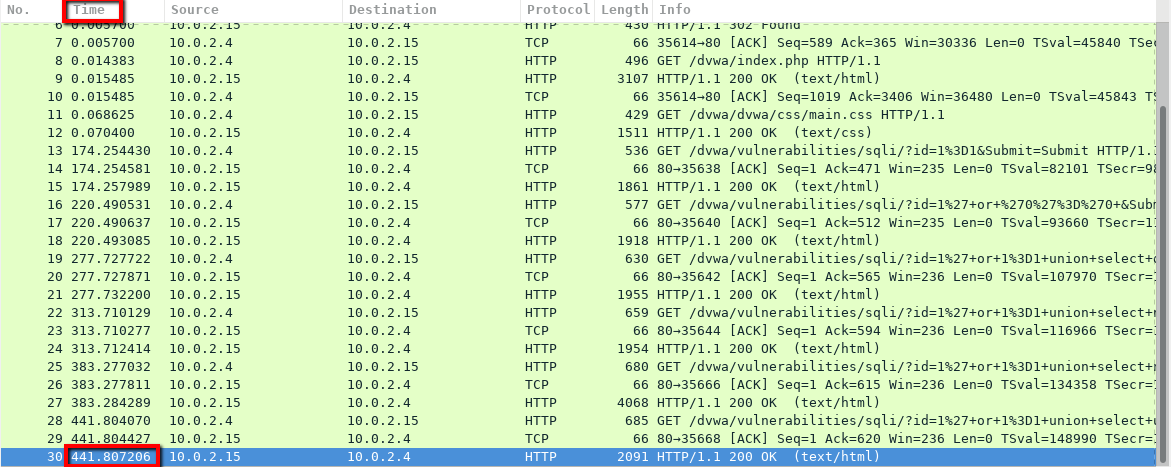
# Instruções

Você usará o Wireshark, um analisador de pacotes de rede comum, para analisar o tráfego de rede. Depois de iniciar o Wireshark, você abrirá uma captura de rede salva anteriormente e visualizará um ataque de injeção SQL passo a passo contra um banco de dados SQL.

## Abra o Wireshark e carregue o arquivo PCAP.

O aplicativo Wireshark pode ser aberto usando uma variedade de métodos em uma estação de trabalho Linux.

* + 1. Inicie o CyberOps Workstation VM.
    2. Clique em **Applications > CyberOps > Wireshark** na área de trabalho e navegue até o aplicativo Wireshark.
    3. No aplicativo Wireshark, clique em **Open** no meio do aplicativo em Arquivos.
    4. Navegue pelo diretório **/home/analyst/** e procure **lab.support.files**. No diretório **lab.support.files** e abra o arquivo **SQL\_Lab.pcap**.
    5. O arquivo PCAP é aberto no Wireshark e exibe o tráfego de rede capturado. Esse arquivo de captura se estende por um período de 8 minutos (441 segundos), a duração desse ataque de injeção SQL.



### Pergunta:

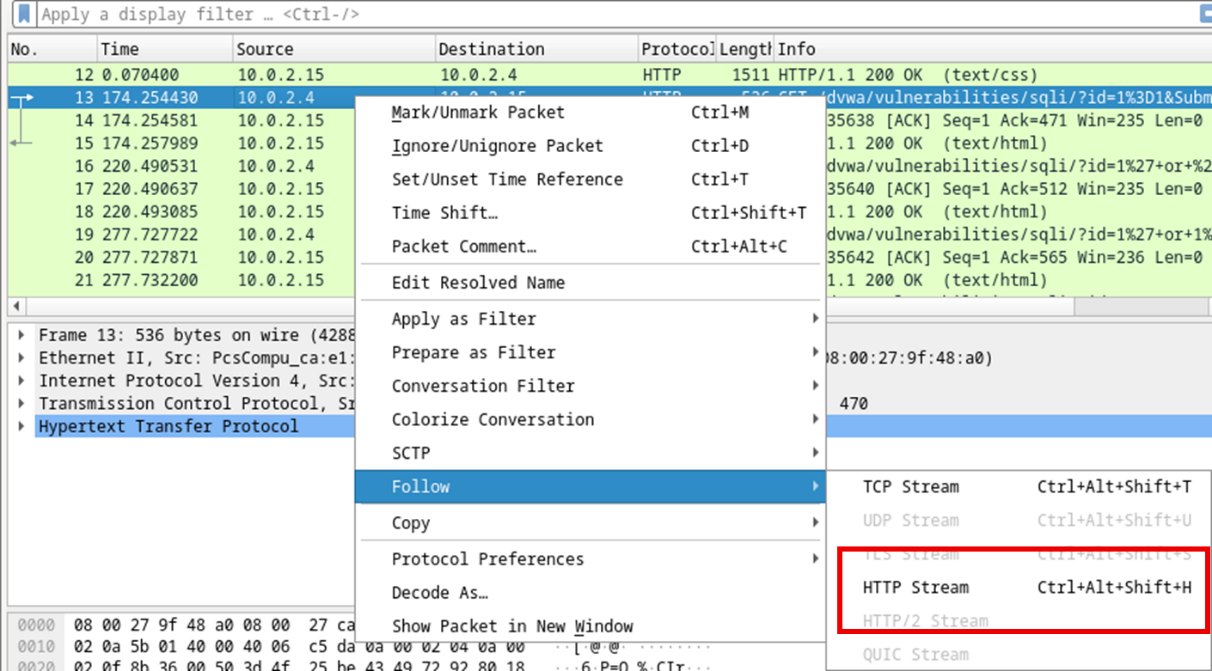
Quais são os dois endereços IP envolvidos neste ataque de injeção SQL com base nas informações exibidas?

Digite suas respostas aqui.

## Exibir o Ataque de Injeção SQL.

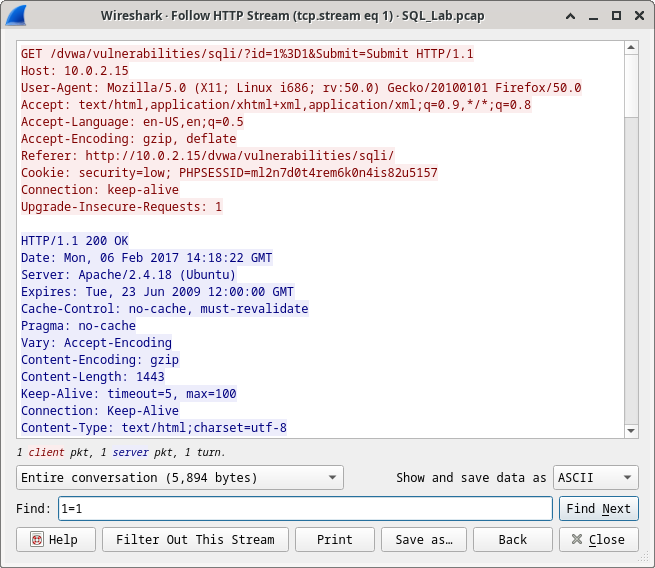
Nesta etapa, você estará visualizando o início de um ataque.

* + 1. Na captura Wireshark, clique com o botão direito do mouse na linha 13 e selecione **Follow** > **HTTP Stream**. A linha 13 foi escolhida porque é uma solicitação HTTP GET. Isso será muito útil em seguir o fluxo de dados como as camadas de aplicativo vê-lo e leva até o teste de consulta para a injeção SQL.

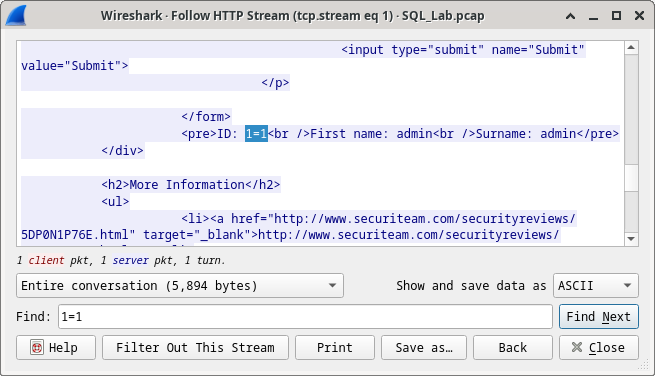


O tráfego de origem é mostrado em vermelho. A origem enviou uma solicitação GET para o host 10.0.2.15. Em azul, o dispositivo de destino está respondendo de volta à origem.

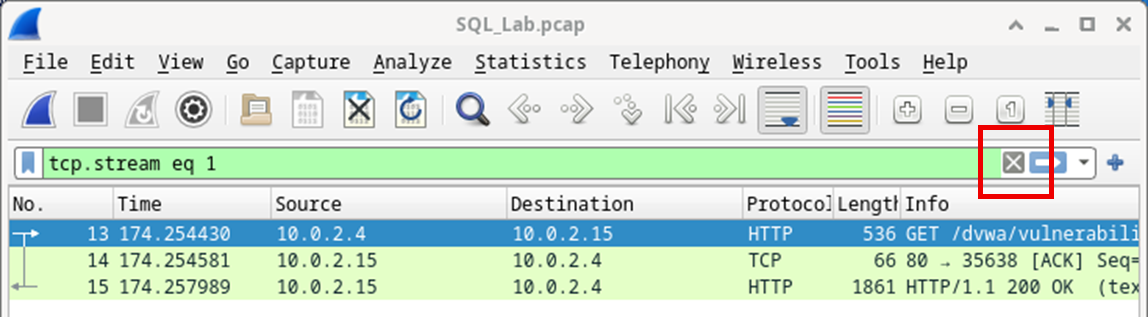
* + 1. No campo **Localizar**, insira **1=1**. Clique em **Localizar próxima**.



* + 1. O invasor inseriu uma consulta (1=1) em uma caixa de pesquisa UserID no destino 10.0.2.15 para ver se o aplicativo está vulnerável à injeção de SQL. Em vez de o aplicativo responder com uma mensagem de falha de logon, ele respondeu com um registro de um banco de dados. O invasor verificou que pode inserir um comando SQL e o banco de dados responderá. A string de pesquisa 1=1 cria uma instrução SQL que será sempre verdadeira. No exemplo, não importa o que é inserido no campo, sempre será verdade.



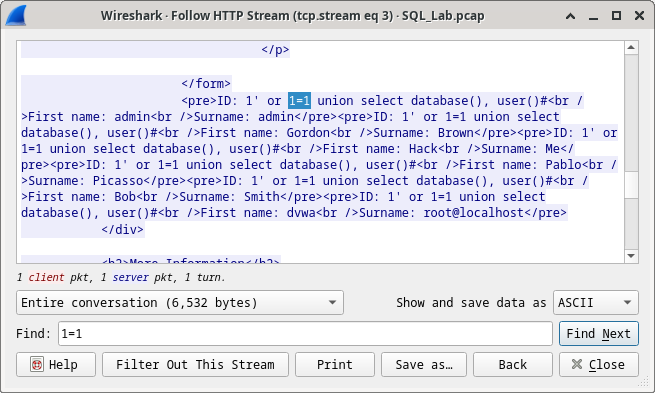
* + 1. Feche a janela Follow HTTP Stream
    2. Clique em **Limpar filtro** de exibição para exibir toda a conversa Wireshark.



## O Ataque de Injeção SQL continua...

Nesta etapa, você estará visualizando a continuação de um ataque.

* + 1. Na captura Wireshark, clique com o botão direito do mouse na linha 19 e clique em **Follow** > **HTTP Stream**.
    2. No campo **Find** , entre **1=1**. Clique em **Find next**.
    3. O invasor inseriu uma consulta (1' ou 1=1 união select database (), user () #) em uma caixa de pesquisa UserID no destino 10.0.2.15. Em vez de o aplicativo responder com uma mensagem de falha de logon, ele respondeu com as seguintes informações:



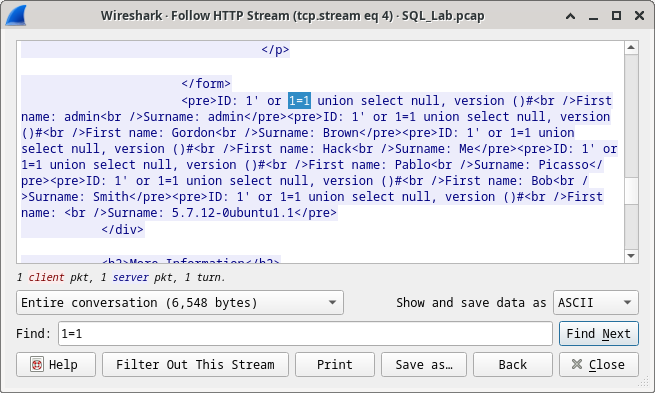
O nome do banco de dados é **dvwa** e o usuário do banco de dados é **root @localhost**. Há também várias contas de usuário sendo exibidas.

* + 1. Feche a janela Follow HTTP Stream.
    2. Clique em **Clear display filter** para exibir toda a conversa do Wireshark.

## O SQL Injection Attack fornece informações do sistema.

O invasor continua e começa a segmentar informações mais específicas.

* + 1. Na captura Wireshark, clique com o botão direito do mouse na linha 22 e selecione **Seguir** > **Fluxo HTTP**. Em vermelho, o tráfego de origem é mostrado e está enviando a solicitação GET para o host 10.0.2.15. Em azul, o dispositivo de destino está respondendo de volta à fonte.
    2. No campo **Find** , entre **1=1**. Clique em **Find next**.
    3. O invasor inseriu uma consulta (1' ou 1=1 união select null, version () #) em uma caixa de pesquisa UserID no destino 10.0.2.15 para localizar o identificador de versão. Observe como o identificador de versão está no final da saída logo antes do </pre>.</div> fechando código HTML.



### Pergunta:

Qual é a versão?

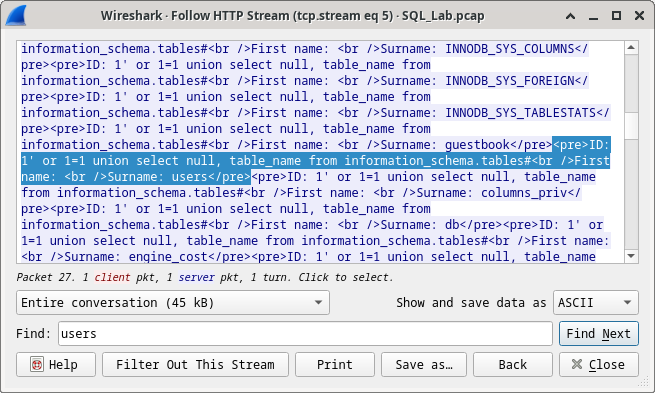
Digite suas respostas aqui.

* + 1. Feche a janela Follow HTTP Stream.
    2. Clique em **Clear display filter** para exibir toda a conversa do Wireshark.

## O Ataque de Injeção SQL e Informações da Tabela.

O invasor sabe que há um grande número de tabelas SQL que estão cheias de informações. O atacante tenta encontrá-los.

* + 1. Na captura Wireshark, clique com o botão direito do mouse na linha 25 e selecione **Follow** > **HTTP stream**. A fonte é mostrada em vermelho. Ele enviou uma solicitação GET para o host 10.0.2.15. Em azul, o dispositivo de destino está respondendo de volta à fonte.
    2. No campo **Find**, insira **users**. Clique em **Find next**.
    3. O invasor inseriu uma consulta (1'ou 1=1 union select null, table\_name from information\_schema.tables #)em uma caixa de pesquisa de ID de usuário no destino 10.0.2.15 para visualizar todas as tabelas no banco de dados. Isso fornece uma enorme saída de muitas tabelas, como o invasor especificou “nulo” sem quaisquer especificações adicionais.



### Pergunta:

O que o comando modificado de (**1' OU 1=1 UNION SELECT null, column\_name FROM INFORMAtTION\_schema.columns WHERE table\_name='users'**) faria para o invasor?

Digite suas respostas aqui.

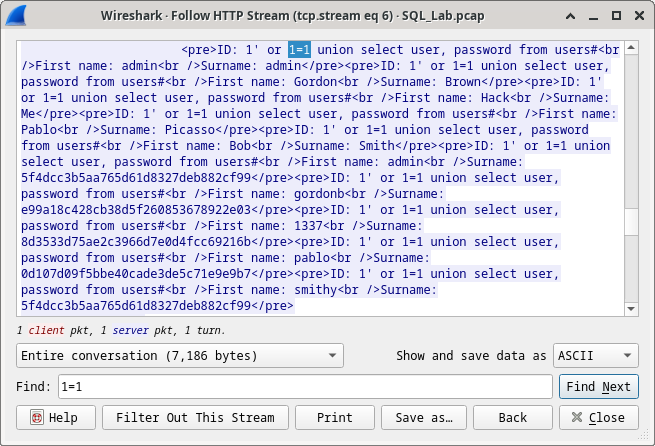
* + 1. Feche a janela Follow HTTP Stream.
    2. Clique em **Clear display filter** para exibir toda a conversa do Wireshark.

## O ataque de injeção de SQL é concluído.

O ataque termina com o melhor prêmio de todos; hashes de senha.

* + 1. Na captura Wireshark, clique com o botão direito do mouse na linha 28 e selecione **Follow** > **HTTP Stream**. A fonte é mostrada em vermelho. Ele enviou uma solicitação GET para o host 10.0.2.15. Em azul, o dispositivo de destino está respondendo de volta à fonte.
    2. Clique em **Find** e digite **1=1**. Procure por esta entrada. Quando o texto estiver localizado, clique em **Cancel** na caixa de pesquisa Localizar texto.

O invasor inseriu uma consulta (1'ou 1=1 union select user, password from users#) em uma caixa de pesquisa UserID no destino 10.0.2.15 para obter nomes de usuário e hashes de senha!



### Pergunta:

Qual usuário tem o hash de senha de 8d3533d75ae2c3966d7e0d4fcc69216b?

Digite suas respostas aqui.

* + 1. Usando um site como <https://crackstation.net/>, copie o hash de senha no cracker de hash de senha e comece a crackear.

### Pergunta:

Qual é a senha de texto simples?

Digite suas respostas aqui.

* + 1. Feche a janela Follow HTTP Stream. Feche todas as janelas abertas.

# Perguntas para reflexão

* 1. Qual é o risco de as plataformas usarem o langauge SQL?

Digite suas respostas aqui.

* 1. Navegue na internet e faça uma pesquisa em “prevenir ataques de injeção SQL”. Quais são os dois métodos ou etapas que podem ser tomadas para evitar ataques de injeção SQL?

Digite suas respostas aqui.